# ALE Vertical Coordinate: Requirements and Design

MPAS Development Team

October 11, 2012

# Contents

**1 Summary** **2**

**2 Requirements** **3**

2.1 Requirement: A vertical coordinate that allows layers that are thinner than the SSH deviations. . . . . 3

2.2 Requirement: A vertical coordinate that reduces vertical mixing by moving with high frequency motions. . . . . 3

**3 Algorithmic Formulations** **4**

3.1 ALE formulation . . . . . 4

3.2 Introduction to z-tilde coordinate system . . . . . 6

3.3 Algorithm for additional z-tilde tendencies . . . . . 7

3.4 Geopotential gradient in momentum equation . . . . . 8

3.5 Isopycnal Coordinate transition from Montgomery Potential . . . . . 8

3.6 Revised Runga-Kutta algorithm . . . . . 10

3.7 Revised Split Explicit Algorithm . . . . . 11

**4 Design and Implementation** **15**

4.1 New variables . . . . . 15

**5 Testing** **16**

# Chapter 1

# Summary

An Arbitrary Lagrangian-Eulerian (ALE) vertical coordinate provides greater freedom than the traditional fixed z-level vertical coordinate system to reduce vertical mixing, allow thin top layers, and improve accuracy over steep topography. This ALE implementation has two components: it responds to the external mode and to high frequency internal models. The first allows all layers to expand and contract in proporation to the external mode, ie. changes in sea surface height (SSH). This alone has been called a $z^*$ coordinate system [1, 2]. In addition, layer interfaces move with high frequency motions in the water column. This was called a $\tilde{z}$ coordinate system, as described by [3]. The purpose of both is to reduce spurious vertical mixing associtaed with fixed z-level coordinates. The $z^*$ part allows the top layers to be very thin, and bypass the constraint of a ten meter thick top layer that is required when all SSH variations must be within the top layer. The algorithm presented here draws heavily from [3], but is translated into the thickness variables of MPAS-Ocean.

# Chapter 2

# Requirements

## 2.1 Requirement: A vertical coordinate that allows layers that are thinner than the SSH deviations.

Date last modified: 11/28/2011
Contributors: Mark, Todd

SSH in a global ocean model is a maximum of 2-3m. The minimum layer thickness typically used in a z-level model is 10m, because all the SSH deviations must be encompassed by the top layer, the only one with variable thickness. If all layers are allowed to vary in thickness, the top layers can be arbitrarily thin.

## 2.2 Requirement: A vertical coordinate that reduces vertical mixing by moving with high frequency motions.

Date last modified: 11/28/2011
Contributors: Mark, Todd

Much of the diapycnal mixing in fixed z-level ocean models is due to internal waves that advect fluid vertically with high temporal frequency, but do not transport water masses on longer time scales. One way to avoid this spurious vertical mixing is an ALE vertical coordinate that moves with the high frequency oscillations. Thus, the coordinate acts as a Lagrangian coordinate at high frequencies in order to limit diapycnal mixing and acts as an Eulerian coordinate for low frequency motions associated with water mass transport.

# Chapter 3

# Algorithmic Formulations

Date last modified: 2/3/2012
Contributors: Mark, Todd

## 3.1 ALE formulation

Here we present a conceptual description of how the ALE vertical coordinate fits into a typical timestepping algorithm. The specifics of implementing ALE in our Runge-Kutta and split explicit formulation is given in later sections.

The MPAS-ocean z-level formulation solves the following equations for thickness, momentum, and tracers at layer $k$:

$$\frac{\partial h_k}{\partial t} + \nabla \cdot \left( h_k^{edge} \mathbf{u}_k \right) + h_k \frac{\partial w_k}{\partial z} = 0, \tag{3.1}$$

$$\frac{\partial \mathbf{u}_k}{\partial t} + \frac{1}{2} \nabla |\mathbf{u}_k|^2 + (\mathbf{k} \cdot \nabla \times \mathbf{u}_k) \mathbf{u}_k^{\perp} + f \mathbf{u}_k^{\perp} + w_k^{edge} \frac{\partial \mathbf{u}_k}{\partial z} = -\frac{1}{\rho_0} \nabla p_k - \frac{\rho g}{\rho_0} \nabla z_k^{mid} + \nu_h \nabla^2 \mathbf{u}_k + \frac{\partial}{\partial z} \left( \nu_v \frac{\partial \mathbf{u}_k}{\partial z} \right), \tag{3.2}$$

$$\frac{\partial h_k \varphi_k}{\partial t} + \nabla \cdot \left( h_k^{edge} \varphi_k^{edge} \mathbf{u}_k \right) + h_k \frac{\partial}{\partial z} \left( \varphi_k w_k \right) = \nabla \cdot \left( h_k^{edge} \kappa_h \nabla \varphi_k \right) + h_k \frac{\partial}{\partial z} \left( \kappa_v \frac{\partial \varphi_k}{\partial z} \right) \tag{3.3}$$

For the numerical formulation, we rewrite these equations with tendencies,

$$\frac{\partial h_k}{\partial t} = T^h(h^*, u^*, w^*) \tag{3.4}$$

$$\frac{\partial \mathbf{u}_k}{\partial t} = \mathbf{T}^u(h^*, u^*, w^*, p^*) \tag{3.5}$$

$$\frac{\partial h_k \varphi_k}{\partial t} = T^{\varphi}(h^*, u^*, w^*, \varphi^*) \tag{3.6}$$

For this general timestepping description, we proceed from time $t_n$, where all variables are known, to time $t_{n+1}$, using $*$ variables for the fluxes that appear in the tendencies of the prognostic equations. Typically, the $*$ variables are from time $n$ for the first iteration, and time $n+1/2$ or $n+1$ for subsequent iterations.

**Single iteration of the timestepping scheme for z-star**

$$\text{known: all } n \text{ and } * \text{ variables except } w_*^{top} \tag{3.7}$$
$$\text{ALE step: compute vertical transport } w_*^{top} \text{ (branch for different grid types)} \tag{3.8}$$
$$\text{compute tendencies } \mathbf{T}^u, T^h, T^\varphi \text{ using } * \text{ variables} \tag{3.9}$$
$$\text{boundary update on all tendencies: } \mathbf{T}^u, T^h, T^\varphi \tag{3.10}$$
$$\mathbf{u}_{k,n+1} = \mathbf{u}_{k,n} + \Delta t \mathbf{T}^u_k \tag{3.11}$$
$$h_{k,n+1} = h_{k,n} + \Delta t T^h_k \tag{3.12}$$
$$\varphi_{k,n+1} = \varphi_{k,n} + \Delta t T^\varphi_k \tag{3.13}$$
$$\text{compute diagnostics based on } \mathbf{u}_{k,n+1}, h_{k,n+1}, \varphi_{k,n+1} \tag{3.14}$$

There is only one step that determines the vertical grid–the ALE step where $w_*^{top}$ is computed. It is important to note that $w_*^{top}$ is the transport of fluid through the layer interface. If the grid is fixed (z-level) then it is the vertical velocity of the fluid. The critical part of the ALE step is that $w_*^{top}$ is specifically chosen in order to get the desired vertical grid behavior. The thickness $h$ is then stepped forward with the continuity equation using this $w_*^{top}$.

The algorithm for the ALE step determination of $w^{top}$ can be summarized as follows:

$$\text{thickness-weighted divergence} \qquad D_{k,*} = \nabla_h \cdot \left( h^{edge}_{k,*} \mathbf{u}_{k,*} \right) \tag{3.15}$$
$$\text{barotropic divergence} \qquad \overline{D}_* = \sum_{k=1}^{kmax} D_{k,*} \tag{3.16}$$
$$\textbf{branch for grid type:} \tag{3.17}$$
$$\text{z-level} \qquad \tilde{T}^h_k = \begin{cases} -\overline{D}_*, & k=1 \\ 0, & k>1 \end{cases} \tag{3.18}$$
$$\text{z-star} \qquad \tilde{T}^h_k = \frac{W_k h_k}{\sum_{k'=1}^{kmax} W_{k'} h_{k'}} (-\overline{D}_*) \tag{3.19}$$
$$\text{z-tilde} \qquad \tilde{T}^h_k = \frac{W_k h_k}{\sum_{k'=1}^{kmax} W_{k'} h_{k'}} (-\overline{D}_*) + T^{hhf}_k \tag{3.20}$$
$$\text{isopycnal} \qquad \tilde{T}^h_k = -D_{k,*} \tag{3.21}$$
$$\text{compute } w_*^{top} \text{ from } k = k_{max} \text{ to } k = 2 \qquad w^{top}_{k,*} = w^{top}_{k+1,*} - D_{k,*} - \tilde{T}^h_k \tag{3.22}$$

The z-tilde branch is explained in the next sections. The weights, $W_k$, determine how variations in SSH are distributed among the layers. The choices are as follows:

- z-level: Set $W(1) = 1$ (top layer), $W(2 : kmax) = 0$. Like POP 2.0, SSH deviations are all contained in the top layer.

- z-star: Set $W(k) = 1$ at all layers. This spreads the thickness change due to SSH throughout the layers in proportion to their initial thickness.
- weighted z-star: Weights may be set in any fashion. A useful weighted z-star set-up is where $W(k) = 1$ in upper layers, taper off from 1 to 0 in mid-layers, and zero in lower layers. This distributes the SSH variations in upper and mid-layers only.

## 3.2 Introduction to z-tilde coordinate system

Here we write the full equation set for the z-tilde system. Two new prognostic equations are added, for $h^{hf}$, the high frequency variations in thickness, and $D^{lf}$, the low frequency divergence variable. Definitions are as follows:

$$\text{thickness-weighted divergence} \qquad D_k = \nabla_h \cdot \left( h_k^{edge} \mathbf{u}_k \right) \tag{3.23}$$

$$\text{barotropic divergence} \qquad \overline{D} = \sum_{k=1}^{kmax} D_k \tag{3.24}$$

$$\text{barotropic and baroclinic divergence} \qquad D_k = \overline{D} + D'_k = \overline{D} + D_k^{hf} + D_k^{lf} \tag{3.25}$$

$$\text{low frequecy baroclinic divergence} \qquad \frac{\partial D_k^{lf}}{\partial t} = -\frac{2\pi}{\tau_{Dlf}} \left[ D_k^{lf} - D'_k \right] \tag{3.26}$$

$$\text{decomposed thickness} \qquad h_k = h_k^{init} + h_k^{hf} + h_k^{ext} \tag{3.27}$$

$$\text{decomposed thickness tendency} \qquad \frac{\partial h_k}{\partial t} = \frac{\partial h_k^{hf}}{\partial t} + \frac{\partial h_k^{ext}}{\partial t} \tag{3.28}$$

$$\text{high frequency thickness tendency} \qquad \frac{\partial h_k^{hf}}{\partial t} = -D_k^{hf} - \frac{2\pi}{\tau_{hhf}} h_k^{hf} + \nabla_h \cdot \left( \kappa_{hhf} \nabla_h h_k^{hf} \right) \tag{3.29}$$

$$\text{external mode thickness response} \qquad \frac{\partial h_k^{ext}}{\partial t} = -\frac{W_k h_k}{\sum_{k'=1}^{kmax} W_{k'} h_{k'}} \overline{D} \tag{3.30}$$

$$\text{SSH} \qquad \frac{\partial \zeta}{\partial t} = -\overline{D} \tag{3.31}$$

$$\text{continuity equation (solve for } w) \qquad \frac{\partial h_k}{\partial t} + D_k + \frac{\partial}{\partial z} \left( h_k w_k \right) = 0. \tag{3.32}$$

These equations are discrete in the vertical, indexed by layer $k$, but continuous in the horizontal and in time. The thickness variables $h$, divergence variables $D$, velocity $\mathbf{u}$, and SSH $\zeta$ are functions of $(\mathbf{x}, t)$, and vertical dependance is explicitly labelled with a subscript $k$. External thickness weights $W_k$ do not depend on $\mathbf{x}$ or $t$, and constant parameters include $\tau_{Dlf}$, $\tau_{hhf}$, and $\kappa_{hhf}$.

The divergence in each cell, $D$, is partitioned (3.25) into its barotropic component $\overline{D}$ and its baroclinic component $D'$, which is further subpartitioned into the low frequency baroclinic component $D^{lf}$ and the remaining high frequency component $D^{hf}$. The low frequency baroclinic divergence (3.26) is computed using a first-order low-pass filter, as suggested by [3], with timescale $\tau_{Dlf}$. The new prognostic variable $D^{lf}$ should be initialized as zero and saved to the restart file.

The thickness $h$ (3.27) is initialized with the 3D field $h^{init}$, and then evolves (3.28) with contributions due to high frequency divergence $h^{hf}$ (the $\tilde{z}$-coordinate part) and external mode oscillations $h^{ext}$

(the $z^*$-coordinate part). The tendency for $h^{hf}$ (3.29) is primarily due to the high frequency divergence term $D^{hf}$, but also includes a restoring term with timescale $\tau_{hhf}$ to prevent any longterm drift coordinate drift, and a thickness diffusion term with diffusion coefficient $\kappa_{hhf}$ to prevent gridscale noise in the surface interfaces, as explained in [3]. The new prognostic variable $h^{hf}$ should be initialized as zero and saved to the restart file.

The prognostic equation for $h_k^{ext}$ (3.30) simply changes the layer thickness in proportion to the change in SSH $\zeta$, where the constant of proportionality $W$ may vary by level (this differs from the formulation in [3]). In a pure $z^*$-coordinate system, one could compute the thickness simply as

$$h_k^{new} = dz_k \left( 1 + \zeta \frac{W_k}{\sum_{k'=1}^{kmax} W_{k'} dz_{k'}} \right), \tag{3.33}$$

at each timestep, where $dz_k$ is the original reference thickness. But here in the $\tilde{z}$-coordinate system, a prognostic equation for $h_k^{ext}$ is required so that it can be combined with the high-frequency divergence effects. The continuity equation (3.32) is then used to solve diagnostically for $w$, the vertical transport through the layer interface.

## 3.3 Algorithm for additional z-tilde tendencies

In the tendency computations that must be added to the code for z-tilde are:

low freq. baroclinic div. $$\frac{\partial D_k^{lf}}{\partial t} = -\frac{2\pi}{\tau_{Dlf}} \left[ D_k^{lf} - (D_k - \overline{D}) \right] \tag{3.34}$$

high freq. thickness tend. $$\frac{\partial h_k^{hf}}{\partial t} = -(D_k - \overline{D} - D_k^{lf}) - \frac{2\pi}{\tau_{hhf}} h_k^{hf} + \nabla_h \cdot \left( \kappa_{hhf} \nabla_h h_k^{hf} \right) \tag{3.35}$$

Note that the variables $D'$ and $D^{hf}$ do not appear in the algorithm and thus do not need to be stored or explicitly computed. For the numerical algorithm, these equations are rewritten as

$$\frac{\partial D_k^{lf}}{\partial t} = T^{Dlf}(D_{k,*}, \overline{D}_*, D_{k,*}^{lf}) \tag{3.36}$$

$$\frac{\partial h_k^{hf}}{\partial t} = T^{hhf}(D_{k,*}, \overline{D}_*, D_{k,*}^{lf}, h_{k,*}^{hf}) \tag{3.37}$$

where * indicates the time-level of the tendency variables. A simple timestepping algorithm, like forward Euler or backwards Euler, is as follows. Details of the Runge-Kutta and split explicit timestepping are given in later sections.

**Single iteration of the timestepping scheme for z-tilde**

$$\text{known: all } n \text{ and } * \text{ variables except } w_*^{top} \tag{3.38}$$
$$\text{compute tendencies } T^{hhf},\ T^{Dlf} \text{ using } * \text{ variables} \tag{3.39}$$
$$\text{ALE step: compute vertical transport } w_*^{top} \text{ (branch for different grid types)} \tag{3.40}$$
$$\text{compute tendencies } \mathbf{T}^u, T^\varphi \text{ using } * \text{ variables} \tag{3.41}$$
$$\text{boundary update on all tendencies: } \mathbf{T}^u, T^h, T^{hhf}, T^{Dlf}, T^\varphi \tag{3.42}$$
$$\mathbf{u}_{k,n+1} = \mathbf{u}_{k,n+1} + \Delta t \mathbf{T}_k^u \tag{3.43}$$
$$h_{k,n+1} = h_{k,n} + \Delta t T_k^h \tag{3.44}$$
$$\varphi_{k,n+1} = \varphi_{k,n} + \Delta t T_k^\varphi \tag{3.45}$$
$$h_{k,n+1}^{hf} = h_{k,n}^{hf} + \Delta t T_k^{hhf} \tag{3.46}$$
$$D_{k,n+1}^{lf} = D_{k,n}^{lf} + \Delta t T_k^{Dlf} \tag{3.47}$$
$$\text{compute diagnostics based on } \mathbf{u}_{k,n+1}, h_{k,n+1}, \varphi_{k,n+1} \tag{3.48}$$

## 3.4 Geopotential gradient in momentum equation

In order to account for hydrostatic pressure differences in tilted layers, an additional term is required in the momentum equation:

$$\begin{aligned}\frac{\partial \mathbf{u}_k}{\partial t} + \frac{1}{2}\nabla\left|\mathbf{u}_k\right|^2 + (\mathbf{k}\cdot\nabla\times\mathbf{u}_k)\mathbf{u}_k^\perp + f\mathbf{u}_k^\perp + w_k^{edge}\frac{\partial \mathbf{u}_k}{\partial z} \\ = -\frac{1}{\rho_0}\nabla p_k - \frac{\rho g}{\rho_0}\nabla z_k^{mid} + \nu_h\nabla^2\mathbf{u}_k + \frac{\partial}{\partial z}\left(\nu_v\frac{\partial \mathbf{u}_k}{\partial z}\right)\end{aligned} \tag{3.49}$$

The variable $z_k^{mid}$ does not require a permanent variable, but can be computed as needed with:

$$z_k^{mid} = \zeta - \sum_{k'=1}^{k-1} h_{k'} - \frac{1}{2}h_k \tag{3.50}$$

## 3.5 Isopycnal Coordinate transition from Montgomery Potential

The current formulation for the momentum equation in isopycnal coordinates,

$$\frac{\partial \mathbf{u}}{\partial t} + q(h\mathbf{u}^\perp) = -\nabla M - \nabla K + \nu(\nabla\delta + k\times\nabla\eta) \tag{3.51}$$

uses a Montgomery Potential, defined as

$$M(x,y,z,t) = \alpha p(x,y,z,t) + gz \tag{3.52}$$

where $\alpha = 1/\rho$ is the specific volume. The hydrostatic condition, $\partial p/\partial z = -\rho g$, implies that $M$ is independent of $z$ in a layer of constant density (See [4] p. 401 for derivation of $M$). For the shallow water equations, fix $z = z_{top} = h + b$ in (3.52) and assume pressure at the top of the fluid has a constant value, so that $\nabla M = g\nabla(h+b)$. Similarly, for the primitive equations $\nabla M = \alpha\nabla p$.

Both $\rho$ and $M$ are constant within each layer, but the pressure and z vary continuously in the vertical. Taking (3.52) in the limit as one approaches a layer interface, $\Delta M = p_{int}\Delta\alpha$. In the top layer, evaluating (3.52) at $z = 0$ and using the hydrostatic condition, $M = g\eta$ where $\eta$ is the sea surface height. These two equations allow us to compute the Montgomery Potential in each column:

$$M_1 = g\eta \tag{3.53}$$

$$M_{k+1} = M_k + \left(\frac{1}{\rho_{k+1}} - \frac{1}{\rho_k}\right) pbot_k \tag{3.54}$$

where $pbot_k$ is the pressure at the bottom of layer $k$.

We would like the pressure computation to be uniform for isopycnal and all z-type vertical coordinates. In isopycnal, by definition, there is no vertical fluxes between layers, and the density in each layer remains constant. The question for the pressure gradient is whether $\nabla M$ can be replaced with

$$\frac{1}{\rho_0}\nabla p_k + \frac{\rho g}{\rho_0}\nabla z_k^{mid}. \tag{3.55}$$

From the definition above, and with constant density in each layer,

$$\nabla M = \frac{1}{\rho}\nabla p + g\nabla z \tag{3.56}$$

There are only two differences: (3.55) uses a constant $\rho_0$ in the first term, due to the Boussinesq approximation; and the second term in (3.55) includes a $\rho/\rho_0$.

## 3.6 Revised Runga-Kutta algorithm

For any timestepping method, the additional tendencies and diagnostic variables outlined in section 3.3 above would be implemented in the same way, using the same subroutines. For the Runge-Kutta algorithm, prognostic equations for $h^{hf}$ and $D^{lf}$ must be added as follows. Here $*$ denotes intermediate variables.

**Prep variables before first stage**

$$\mathbf{u}_{k,n+1} = \mathbf{u}_{k,n},\ \ h_{k,n+1} = h_{k,n},\ \ h^{hf}_{k,n+1} = h^{hf}_{k,n},\ \ D^{lf}_{k,n+1} = D^{lf}_{k,n},\ \ \varphi_{k,n+1} = \varphi_{k,n} h_{k,n} \tag{3.57}$$

$$\mathbf{u}^*_k = \mathbf{u}_{k,n},\ \ h^*_k = h_{k,n},\ \ h^{*hf}_k = h^{hf}_{k,n},\ \ D^{*lf}_k = D^{lf}_{k,n},\ \ \varphi^*_k = \varphi_{k,n},\ \ p^*_k = p_{k,n} \tag{3.58}$$

$$a = \left(\frac{1}{2}, \frac{1}{2}, 1, 0\right),\ \ b = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}\right), \tag{3.59}$$

**Iteration**

do j=1,4 (3.60)

compute tendencies $T^{hhf},\ \ T^{Dlf}$ (3.61)

ALE step: compute vertical transport $w^{top}_*$ (branch for different grid types) (3.62)

compute tendencies $\mathbf{T}^u(\mathbf{u}^*_k, w^*_k, p^*_k),\ \ T^\varphi(\mathbf{u}^*_k, w^*_k, \varphi^*_k)$ (3.63)

$$T^h_k = -D^*_k + w^*_{k+1} - w^*_k \tag{3.64}$$

boundary update on all tendencies: $\mathbf{T}^u, T^h, T^{hhf}, T^{Dlf}, T^\varphi$ (3.65)

$$\mathbf{u}^*_k = \mathbf{u}_{k,n} + a_j \Delta t \mathbf{T}^u_k \tag{3.66}$$

$$h^*_k = h_{k,n} + a_j \Delta t T^h_k \tag{3.67}$$

$$h^{*hf}_k = h^{hf}_{k,n} + a_j \Delta t T^{hhf}_k \tag{3.68}$$

$$D^{*lf}_k = D^{lf}_{k,n} + a_j \Delta t T^{Dlf}_k \tag{3.69}$$

$$\varphi^*_k = \frac{1}{h^*_k}\left[h_{k,n}\varphi_{k,n} + a_j \Delta t T^\varphi_k\right] \tag{3.70}$$

compute diagnostics based on $\mathbf{u}^*_k, h^*_k, \varphi^*_k$ (3.71)

$$\mathbf{u}_{k,n+1} = \mathbf{u}_{k,n+1} + b_j \Delta t \mathbf{T}^u_k \tag{3.72}$$

$$h_{k,n+1} = h_{k,n+1} + b_j \Delta t T^h_k \tag{3.73}$$

$$h^{hf}_{k,n+1} = h^{hf}_{k,n+1} + b_j \Delta t T^{hhf}_k \tag{3.74}$$

$$D^{lf}_{k,n+1} = D^{lf}_{k,n+1} + b_j \Delta t T^{Dlf}_k \tag{3.75}$$

$$\varphi_{k,n+1} = \varphi_{k,n+1} + b_j \Delta t T^\varphi_k \tag{3.76}$$

enddo (3.77)

**End of step**

$$\varphi_{k,n+1} = \varphi_{k,n+1}/h_{k,n+1} \tag{3.78}$$

revise $\mathbf{u}_{k,n+1},\ \ \varphi_{k,n+1}$ with implicit vertical mixing (3.79)

compute diagnostics based on $\mathbf{u}_{k,n+1}, h_{k,n+1}, \varphi_{k,n+1}$ (3.80)

## 3.7 Revised Split Explicit Algorithm

Additional tendencies and diagnostic variables outlined in section 3.3 are the same here. The additional prognostic equations for $h^{hf}$ and $D^{lf}$ do not affect the barotropic subcycling, and are similar to the timestepping of tracer variables. The wrinkle here is to ensure that the new SSH computed from the barotropic solver (Stage 2) is used correctly for $h_k^{ext}$.

**Prepare variables before first iteration**
Always use most recent available for forcing terms. The first time, use end of last timestep.

$$\mathbf{u}_k^* = \mathbf{u}_{k,n}, \;\; w_k^* = w_{k,n}, \;\; p_k^* = p_{k,n}, \;\; \varphi_k^* = \varphi_{k,n}, \tag{3.81}$$

$$h_{k,*} = h_{k,n}, \;\; h_{k,*}^{edge} = h_{k,n}^{edge}, \zeta_* = \zeta_n, \;\; \mathbf{u}_{k,*}^{bolus} = \mathbf{u}_{k,n}^{bolus} \tag{3.82}$$

$$h_k^{*hf} = h_{k,n}^{hf}, \;\; D_k^{*lf} = D_{k,n}^{lf} \tag{3.83}$$

$$\overline{\mathbf{u}}_n = \textstyle\sum_{k=1}^{N^{edge}} h_{k,n}^{edge} \mathbf{u}_{k,n} \Big/ \sum_{k=1}^{N^{edge}} h_{k,n}^{edge}, \text{ on start-up only.} \tag{3.84}$$

$$\text{Otherwise, } \overline{\mathbf{u}}_n \text{ from previous step.} \tag{3.85}$$

$$\mathbf{u}'_{k,n} = \mathbf{u}_{k,n} - \overline{\mathbf{u}}_n \tag{3.86}$$

$$\mathbf{u}'_{k,n+1/2} = \mathbf{u}'_{k,n} \tag{3.87}$$

The full algorithm, Stages 1–3 are iterated. This is typically done using two iterations, like a predictor–corrector timestep. The flag for the number of these large iterations is `config_n_ts_iter`.

**Stage 1: Baroclinic velocity (3D), explicit with long timestep**
Iterate on linear Coriolis term only.

$$\text{compute } \mathbf{T}^u(\mathbf{u}_k^*, w_k^*, p_k^*) + g\nabla\zeta^* \tag{3.88}$$

$$\text{compute weights, } \omega_k = h_{k,*}^{edge} \Big/ \textstyle\sum_{k=1}^{N^{edge}} h_{k,*}^{edge} \tag{3.89}$$

$$\left\{ \begin{array}{l} \text{compute } \mathbf{u}'^{\perp}_{k,n+1/2} \text{ from } \mathbf{u}'_{k,n+1/2} \\ \tilde{\mathbf{u}}'_{k,n+1} = \mathbf{u}'_{k,n} + \Delta t \left( -f \mathbf{u}'^{\perp}_{k,n+1/2} + \mathbf{T}^u(\mathbf{u}_k^*, w_k^*, p_k^*) + g\nabla\zeta^* \right) \\ \overline{\mathbf{G}} = \frac{1}{\Delta t} \sum_{k=1}^{N^{edge}} \omega_k \tilde{\mathbf{u}}'_{k,n+1} \text{ (unsplit: } \overline{\mathbf{G}} = 0 \text{ )} \\ \mathbf{u}'_{k,n+1} = \tilde{\mathbf{u}}'_{k,n+1} - \Delta t \overline{\mathbf{G}} \\ \mathbf{u}'_{k,n+1/2} = \frac{1}{2} \left( \mathbf{u}'_{k,n} + \mathbf{u}'_{k,n+1} \right) \\ \text{boundary update on } \mathbf{u}'_{k,n+1/2} \end{array} \right. \qquad l = 1 \ldots L \tag{3.90}$$

(3.91)

The bracketed computation is iterated $L$ times. The default method is to use $L = 1$ on the first time through Stages 1–3, and $L = 2$ the second time through. This is set by the flags `config_n_bcl_iter_beg = 1, config_n_bcl_iter_mid = 2, config_n_bcl_iter_end = 2`. In the case of two iterations through Stages 1–3, the flag `config_n_bcl_iter_mid` is not used.

**Stage 2: Barotropic velocity (2D), explicitly subcycled**
Advance $\overline{\mathbf{u}}$ and $\zeta$ as a coupled system through $2J$ subcycles, ending at time $t + 2\Delta t$.

**velocity predictor step:**

compute $\overline{\mathbf{u}}^{\perp}_{n+(j-1)/J}$ from $\overline{\mathbf{u}}_{n+(j-1)/J}$ (3.92)

$$\tilde{\overline{\mathbf{u}}}_{n+j/J} = \overline{\mathbf{u}}_{n+(j-1)/J} + \frac{\Delta t}{J}\left(-f\overline{\mathbf{u}}^{\perp}_{n+(j-1)/J} - g\nabla\zeta_{n+(j-1)/J} + \overline{\mathbf{G}}_j\right) \tag{3.93}$$

boundary update on $\tilde{\overline{\mathbf{u}}}_{n+j/J}$ (3.94)

**SSH predictor step:**

$$\zeta^{edge}_{n+(j-1)/J} = Interp(\zeta_{n+(j-1)/J}) \tag{3.95}$$

$$\tilde{\mathbf{F}}_j = \left((1-\gamma_1)\overline{\mathbf{u}}_{n+(j-1)/J} + \gamma_1\tilde{\overline{\mathbf{u}}}_{n+j/J}\right)\left(\zeta^{edge}_{n+(j-1)/J} + H^{edge}\right) \tag{3.96}$$

$$\tilde{\zeta}_{n+j/J} = \zeta_{n+(j-1)/J} + \frac{\Delta t}{J}\left(-\nabla\cdot\tilde{\mathbf{F}}_j\right) \tag{3.97}$$

boundary update on $\tilde{\zeta}_{n+j/J}$ (3.98)

**velocity corrector step:**

compute $\tilde{\overline{\mathbf{u}}}^{\perp}_{n+j/J}$ from $\tilde{\overline{\mathbf{u}}}_{n+j/J}$ (3.99)

$$\overline{\mathbf{u}}_{n+j/J} = \overline{\mathbf{u}}_{n+(j-1)/J} + \frac{\Delta t}{J}\left(-f\tilde{\overline{\mathbf{u}}}^{\perp}_{n+j/J} - g\nabla\left((1-\gamma_2)\zeta_{n+(j-1)/J} + \gamma_2\tilde{\zeta}_{n+j/J}\right) + \overline{\mathbf{G}}_j\right) \tag{3.100}$$

boundary update on $\overline{\mathbf{u}}_{n+j/J}$ (3.101)

**SSH corrector step:**

$$\tilde{\zeta}^{edge}_{n+j/J} = Interp\left((1-\gamma_2)\zeta_{n+(j-1)/J} + \gamma_2\tilde{\zeta}_{n+j/J}\right) \tag{3.102}$$

$$\mathbf{F}_j = \left((1-\gamma_3)\overline{\mathbf{u}}_{n+(j-1)/J} + \gamma_3\overline{\mathbf{u}}_{n+j/J}\right)\left(\tilde{\zeta}^{edge}_{n+j/J} + H^{edge}\right) \tag{3.103}$$

$$\zeta_{n+j/J} = \zeta_{n+(j-1)/J} + \frac{\Delta t}{J}\left(-\nabla\cdot\mathbf{F}_j\right) \tag{3.104}$$

boundary update on $\zeta_{n+j/J}$ (3.105)

Repeat $j = 1 \ldots 2J$ to step through the barotropic subcycles. There are two predictor and two corrector steps for each subcycle. At the end of $2J$ subcycles, we have progressed through two baroclinic timesteps, i.e. through $2\Delta t$. In the code, `config_n_btr_subcycles` is $J$, while `config_btr_subcycle_loop_factor=2` is the "2" coefficient in $2J$.

The input flag `config_btr_solve_SSH2=.true.` runs the algorithm as shown, while `.false.` does not include the SSH corrector step. Here $H^{edge}$ is the total column depth without SSH perturbations, that is, from the higher cell adjoining an edge to $z = 0$.

The velocity corrector step may be iterated to update the velocity in the Coriolis term. The number of iterations is controlled by `config_n_btr_cor_iter`, and is usually set to two.

**Stage 2 continued**

The coefficients $(\gamma_1, \gamma_2, \gamma_3)$ control weighting between the old and new variables in the predictor velocity, corrector SSH gradient, and corrector velocity, respectively. These are typically set as $(\gamma_1, \gamma_2, \gamma_3) = (0.5, 1, 1)$, but this is open to investigation. These flags are `config_btr_gam1_uWt1`, `config_btr_gam2_SSHWt1`, `config_btr_gam3_uWt2`.

The baroclinic forcing $\overline{\mathbf{G}}_j$ may vary over the barotropic subcycles, as long as $\frac{1}{2J}\sum_{j=1}^{2J} \overline{\mathbf{G}}_j = \overline{\mathbf{G}}$. This option is not currently implemented in the code.

$$\overline{\mathbf{u}}_{avg} = \frac{1}{2J+1}\sum_{j=0}^{2J} \overline{\mathbf{u}}_{n+j/J} \tag{3.106}$$

$$\overline{\mathbf{F}} = \frac{1}{2J}\sum_{j=1}^{2J} \mathbf{F}_j \tag{3.107}$$

boundary update on $\overline{\mathbf{F}}$ (3.108)

$$\mathbf{u}^{corr} = \left(\overline{\mathbf{F}} - \sum_{k=1}^{N^{edge}} h_{k,*}^{edge}\left(\overline{\mathbf{u}}_{avg} + \mathbf{u}'_{k,n+1/2} + \mathbf{u}_{k,*}^{bolus}\right)\right) \Big/ \sum_{k=1}^{N^{edge}} h_{k,*}^{edge} \tag{3.109}$$

$$\mathbf{u}_k^{tr} = \overline{\mathbf{u}}_{avg} + \mathbf{u}'_{k,n+1/2} + \mathbf{u}_{k,*}^{bolus} + \mathbf{u}^{corr} \tag{3.110}$$

where $\mathbf{u}_{k,*}^{bolus}$ is the GM bolus velocity computed at the end of stage 3, and $\mathbf{u}^{tr}$ is the transport velocity used in the advection terms for for thickness and tracers.

For unsplit explicit, skip all computations in stage 2. Instead, set:

$$\overline{\mathbf{u}}_{avg} = 0 \tag{3.111}$$

$$\mathbf{u}_k^{tr} = \mathbf{u}'_{k,n+1/2} + \mathbf{u}_{k,*}^{bolus} \tag{3.112}$$

**Discussion:** The derivation of $\mathbf{u}^{corr}$ is as follows. For consistency between the barotropic and summed baroclinic thickness flux, we must enforce

$$\overline{\mathbf{F}} \;=\; \sum_{k=1}^{N} h_{k,*}^{edge}\mathbf{u}_k^{tr}. \tag{3.113}$$

on each edge. To do that, introduce a velocity correction $\mathbf{u}^{corr}$ that is vertically constant, and add it to the velocity as a barotropic correction. The total transport velocity used in the thickness and tracer tendency terms is then given by (3.110). Substitute (3.110) into (3.113), note that $\mathbf{u}^{corr}$ moves outside of the vertical sum, and solve for $\mathbf{u}^{corr}$ to obtain (3.109).

**Stage 3: update thickness, tracers, density and pressure**

$$\text{compute tendencies } T^{hhf},\ T^{Dlf} \tag{3.114}$$

$$\text{boundary update on tendencies: } T^{hhf}, T^{Dlf} \tag{3.115}$$

$$\text{ALE step: compute vertical transport } w_*^{top} \text{ (branch for different grid types)} \tag{3.116}$$

$$\text{based on } u^{tr}, h^* \tag{3.117}$$

$$T_k^h = \left( -\nabla \cdot \left( h_k^{*\,edge} \mathbf{u}_k^{tr} \right) - \frac{\partial}{\partial z} \left( h_k^* w_k^* \right) \right) \tag{3.118}$$

$$T^{\varphi} = \left( -\nabla \cdot \left( h_k^{*\,edge} \varphi_k^* \mathbf{u}_k^{tr} \right) - \frac{\partial}{\partial z} \left( h_k^* \varphi_k^* w_k^* \right) + \nabla \cdot \left( h_k^* \kappa_h \nabla \varphi_k^* \right) + h_k^* \frac{\partial}{\partial z} \left( \kappa_v \frac{\partial \varphi_k^*}{\partial z} \right) \right) \tag{3.119}$$

$$\text{boundary update on tendencies: } T^h, T^{\varphi} \tag{3.120}$$

$$h_{k,n+1} = h_{k,n} + \Delta t T_k^h \tag{3.121}$$

$$\varphi_{k,n+1} = \frac{1}{h_{k,n+1}} \left[ h_{k,n} \varphi_{k,n} + \Delta t T_k^{\varphi} \right] \tag{3.122}$$

$$h_{k,n+1}^{hf} = h_{k,n}^{hf} + \Delta t T_k^{hhf} \tag{3.123}$$

$$D_{k,n+1}^{lf} = D_{k,n}^{lf} + \Delta t T_k^{Dlf} \tag{3.124}$$

**Reset variables**

| **if iterating** | **after final iteration** |
|---|---|
| $\mathbf{u}'_{k,*} = \mathbf{u}'_{k,n+1/2}$ | $\mathbf{u}'_{k,n+1}$ from stage 1 |
| $\overline{\mathbf{u}}_* = \overline{\mathbf{u}}_{avg}$ from stage 2 | $\overline{\mathbf{u}}_{n+1} = \overline{\mathbf{u}}_{avg}$ from stage 2 |
| $\mathbf{u}_{k,*} = \overline{\mathbf{u}}_* + \mathbf{u}'_{k,*}$ | $\mathbf{u}_{k,n+1} = \overline{\mathbf{u}}_{n+1} + \mathbf{u}'_{k,n+1}$ |
| $h_{k,*} = \frac{1}{2}\left(h_{k,n} + h_{k,n+1}\right)$ | $h_{k,n+1}$ from stage 3 |
| $\varphi_{k,*} = \frac{1}{2}\left(\varphi_{k,n} + \varphi_{k,n+1}\right)$ | $\varphi_{k,n+1}$ from stage 3 |
| $h_{k,*}^{hf} = \frac{1}{2}\left(h_{k,n}^{hf} + h_{k,n+1}^{hf}\right)$ | $h_{k,n+1}^{hf}$ from stage 3 |
| $D_{k,*}^{lf} = \frac{1}{2}\left(D_{k,n}^{lf} + D_{k,n+1}^{lf}\right)$ | $D_{k,n+1}^{lf}$ from stage 3 |
| diagnostics: | diagnostics: |
| $\rho_k^* = EOS(T_k^*, S_k^*)$ | $\rho_{k,n+1} = EOS(T_{k,n+1}, S_{k,n+1})$ |
| $p_k^* = g\sum_{k'=1}^{k-1} \rho_{k'}^* h_{k'} + \frac{1}{2} g \rho_k^* h_k$ | $p_{k,n+1} = g\sum_{k'=1}^{k-1} \rho_{k',n+1} h_{k',n+1} + \frac{1}{2} g \rho_{k,n+1}^* h_{k,n+1}$ |
| $h_{k,*}^{edge} = interp(h_{k,*})$ | $h_{k,n+1}^{edge} = interp(h_{k,n+1})$ |
| $\zeta_* = \sum_{k=1}^{kmax} h_{k,*} - H$ | $\zeta_{n+1} = \sum_{k=1}^{kmax} h_{k,n+1} - H$ |
| compute $\mathbf{u}_{k,*}^{bolus}$ | compute $\mathbf{u}_{k,n+1}^{bolus}$ |

(3.125)

where H is the total column height with zero sea surface height.

Question: Check [3]: eqns 17-18 shows conservative vs nonconservative schemes. This is the method of vertical advection for us - does it effect our coordinate scheme?

# Chapter 4

# Design and Implementation

Date last modified: 11/28/2011
Contributors: Mark, Todd

## 4.1 New variables

```
namelist character vert_grid config_vert_grid_type      isopycnal
 also: 'zlevel', 'zstar', 'ale'
```

$\tau_{Dlf}$, $\tau_{hhf}$, and $\kappa_{hhf}$:

```
namelist real      grid     config_Dlf_restore_time    432000.0
namelist real      grid     config_hhf_restore_time    432000.0
namelist real      grid     config_hhf_diff            100
```

note: should config_h_diff be in list hmix or vert_grid?

Arrays for $h^{hf}$, $D^{lf}$, $W$.

```
var persistent real    hHighFreq      ( nVertLevels nCells Time ) 2 r hHighFreq state - -
var persistent real    tend_hHighFreq ( nVertLevels nCells Time ) 1 - hHighFreq tend - -
var persistent real    divergenceLowFreq      ( nVertLevels nCells Time ) 2 r divergenceLowFre
var persistent real    tend_divergenceLowFreq ( nVertLevels nCells Time ) 1 - divergenceLowFre
```

# Chapter 5

# Testing

Date last modified: 11/28/2011
Contributors: Mark, Todd

Domain: 60km quasiuniform global mesh.
Verify that config_vert_grid_type = zlevel and isopycnal match between trunk and branch.
Verify conservation of tracers (pointwise) and total volume.

Domain: periodic channel with forced internal oscillations, see [3].
Verify that $\tilde{z}$ produces less vertical mixing than $z*$, as in fig. 6-9.

Domain: overflow, see [1].

# Bibliography

[1] A Adcroft and JM Campin. Rescaled height coordinates for accurate representation of free-surface flows in ocean circulation models. *Ocean Modelling*, 7(3-4):269–284, 2004.

[2] JM Campin, A Adcroft, C Hill, and J Marshall. Conservation of properties in a free-surface model. *Ocean Modelling*, 6(3-4):221–244, 2004.

[3] M Leclair. -coordinate, an Arbitrary Lagrangian-Eulerian coordinate separating high and low frequency motions. *Ocean Modelling*, 2011.

[4] R. L. Higdon. Numerical modelling of ocean circulation. *Acta Numerica*, 15:385–407, 2006.