Synthetic Floats, Drifters, and Moorings in HYCOM

By George Halliwell and Zulema Garraffo, 20 February 2002

General Remarks

HYCOM has been equipped with code designed to deploy and track synthetic floats and drifters, or to seed the model with synthetic moorings, during model run time. Floats and drifters are tracked during the model run because the relatively short temporal sampling interval required to accurately advect the float in the presence of energetic diurnal, inertial, and synoptic variability requires large model fields to be archived far too frequently for off-line execution to be feasible. Even when monthly forcing drives the model, velocity fields should be sampled every one or two hours to accurately advect the float.

The horizontal and temporal interpolation schemes used to advect the floats were adapted from the algorithms developed for MICOM (Garraffo *et al*, 2001a; 2001b). Horizontal interpolation is performed using a sixteen-point grid box surrounding the float. Two-dimensional polynomial interpolation is performed if a sufficient number of good grid points are available, while bilinear interpolation from the four grid points surrounding the float is performed otherwise. Temporal interpolation is performed using a fourth-order Runga-Kutta algorithm.

The capabilities of the HYCOM synthetic float code have been substantially extended from MICOM. Floats in MICOM were released in a specified model layer and always remained within that layer (e.g. Fratantoni, 1996). This works well in the absence of diapycnal mixing, or for relatively short integrations. It also works well in the MICOM surface mixed layer if the intent is to compare model surface horizontal Lagrangian tracks to observed tracks obtained from surface drifters (Garraffo *et* al., 2001a; 2001b). Beneath the mixed layer, the advection of MICOM floats confined to one layer is along isopycnic surfaces, so they track water parcels fairly accurately when diapycnal mixing is not present. Since diapycnal mixing alters the density of a water parcel, water does not flow along isopycnic surfaces when this mixing is present and MICOM floats confined to a layer cannot track water parcels. The mixing acts to slowly relocate MICOM model interfaces in the vertical even when local vertical velocity is zero. As a result, the position of a water parcel relative to model interfaces will change over time, eventually to the point where the parcel (and a float that tracks it) enters a new model layer. This is also true in HYCOM, where the selected vertical mixing algorithms and the hybrid coordinate adjustment algorithm will relocate vertical coordinates relative to the depth of a water parcel. If a float located below the surface mixed layer is to follow a water parcel, it cannot simply remain within the model layer where it was initially released.

In HYCOM, three types of floats/drifters are supported based on the specified vertical advection rules. To follow water parcels, the diagnosed vertical velocity field is used to advect three-dimensional Lagrangian floats. Isobaric floats remain at prescribed pressure levels while isopycnic floats remain at the depths of prescribed density surfaces. These latter two float types are included because actual drifters released in the ocean are usually one of these two types. For example, surface drifters can be specified by setting the type to isobaric and setting the initial drifter depth to be within model layer one. The user can also specify that the model be seeded with synthetic moorings, in which case no advection is performed. These moorings can provide high-frequency sampling of the water column at prescribed locations, which can be at the locations of observational moorings to facilitate model-data comparison. Dynamical and thermodynamical water properties are interpolated to the location of each float. Time series of float position and depth, along with the interpolated water properties, are archived for further analysis. If synthetic moorings are specified, velocity components u, v, w are output instead of float position and depth.

Algorithms

Vertical Interpolation Algorithm

Horizontal interpolation is performed for both layer and interface variables. Horizontal interpolation of layer variables is straightforward because values from the model layer containing the float at the surrounding grid points are used as input. For interface variables, however, vertical interpolation must be performed first at the surrounding grid points. Vertical interpolation is performed as follows: Given a float located at pressure p_d within model layer k, which is bounded by upper interface k and lower interface $k+1$, the quotient q_d is first determined using

$$q_d = \frac{p^{k+1} - p_d}{p^{k+1} - p^k}. \tag{1}$$

The virtual pressure surface $\hat{p}$ shown in the Figure is defined as the surface within model layer k where q_d is constant; i.e., where

$$\hat{p} = q_d p^k + (1 - q_d) p^{k+1} \tag{2}$$

At each input grid point i, j, interface variables are vertically interpolated to the pressure depth $\hat{p}^{i,j} = q_d p^{i,j,k} + (1 - q_d) p^{i,j,k+1}$. The vertically interpolated value of interface variable A at grid point i, j then becomes

$$A^{i,j} = q_d A^{i,j,k} + (1 - q_d)\, A^{i,j,k+1}. \tag{3}$$

The resulting input array $A^{i,j}$ is then horizontally interpolated to the float location.

Horizontal Interpolation Algorithm

The first step of the horizontal interpolation algorithm is to select the sixteen-point input box. The model grid point i_0, j_0 located immediately to the southwest of the float location is identified. The sixteen-point grid box then consists of points $i_0 - 1$ to $i_0 + 2$ and $j_0 - 1$ to $j_0 + 2$. If a sufficient number of water grid points are available, a two-dimensional polynomial surface is fit to the data to perform the interpolation. If too few water grid points are available, a bilinear scheme using the four grid points i_0 to $i_0 + 1$, j_0 to $j_0 + 1$ is invoked to perform the interpolation. If none of these four points are water points, the float is assumed to have run aground. All model variables are interpolated from their native grid (p, u, or v); e.g. for zonal velocity u, sixteen u grid points are selected to perform the interpolation. In addition to land points, other points in the 16-point box are masked from the interpolation if the layer containing the float is a zero thickness layer at the bottom.

There are minor differences in the HYCOM implementation of horizontal interpolation in comparison to MICOM. The two-dimensional polynomial fit is performed using the same routine that Mariano and Brown (1992) employed for the large-scale trend surface fit in their parameter matrix objective analysis algorithm. The minimum number of surrounding water grid points required to perform the polynomial interpolation instead of the bilinear interpolation is presently set at 12, determined as a balance between using the more-accurate polynomial interpolation whenever possible, but having a sufficient number of grid points to resolve the two-dimensional structure of the field.

Estimation of Vertical Velocity for Three-Dimensional Lagrangian Floats

In the absence of diabatic processes, model interfaces are material surfaces, and three-dimensional flow in the model will be parallel to model interfaces. In practice, the model continuity and horizontal advection/diffusion equations are solved under the assumption that model interfaces are material surfaces. A horizontally stationary float will therefore move with the local vertical velocity determined from the continuity equation. In the Figure, this is illustrated by the vertical arrows at the float and at the model interfaces located immediately above and below the float. If the local vertical velocity is zero and the float is moving horizontally, the float motion must have a vertical component if it moves parallel to a sloping model layer as illustrated in the Figure. In the absence of diabatic processes, the float will remain on the virtual pressure interface $\hat{p}$ such that the quotient q_d in (1) remains constant.

Both components of the vertical velocity can be important. The local vertical velocity w_l contains the influence of Ekman pumping/suction and equatorial upwelling. The advective vertical velocity w_a is important, for example, at the Equator. At the depth of the Equatorial Undercurrent, model interfaces slope upward toward the east. For a float located at the Equator and at the core depth of the undercurrent, w_l is small while w_a is significant and positive as the eastward-propagating float moves parallel to the upward-sloping model interfaces.

The local vertical velocity is diagnosed from continuity as follows:

$$\frac{\partial}{\partial t}(\Delta p) = -\Delta p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) - \left(u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y}\right)(\Delta p), \tag{4}$$

where Δp is layer pressure thickness. Local vertical velocity is estimated at model interfaces by integrating (4) downward from the surface. The vertical velocity at interface $k+1$ is:

$$w_l^{k+1} = w_l^k - \Delta t\left[(\Delta p)^k\left(\frac{du^k}{dx} + \frac{dv^k}{dy}\right) - \left(u^k\frac{d}{dx} + v^k\frac{d}{dy}\right)(\Delta p)^k\right], \tag{5}$$

where w_l is the pressure change per unit time. Given the array of local vertical velocities at model interfaces, this variable is horizontally interpolated to the float location after vertical interpolation is performed at surrounding grid points using (3). The estimate of w_l is performed in the float subroutine. These estimates were compared to the vertical interface motion produced by the HYCOM continuity equation in subroutine cnuity.f, and found to be extremely close, as expected.

To calculate advective vertical velocity as pressure change per unit time, the float is assumed to remain on the virtual pressure surface $\hat{p}$ (see the Figure):

$$w_a = u_d \frac{\partial \hat{p}}{\partial x} + v_d \frac{\partial \hat{p}}{\partial y}, \tag{6}$$

where $\mathbf{v}_d$ is the horizontal float velocity. This component of vertical velocity is estimated at the float location by first calculating $\partial p / \partial x$ and $\partial p / \partial y$ on model interfaces at u and v grid points, respectively. These horizontal gradient components are vertically interpolated to the depth of $\hat{p}$ using (3). The vertically interpolated zonal gradients at u points and meridional gradients at v points are then horizontally interpolated to the float location. They are used along with the layer velocity components u and v that were previously horizontally interpolated to the float location to estimate w_a using (6).

The presence of diabatic processes will tend to induce fluxes across model interfaces and the relocation of model vertical coordinates ($ds / dt \neq 0$, where s is a generalized vertical coordinate). This occurs due to the combined influence of the selected vertical mixing algorithms and the hybrid vertical coordinate generator. As a result, model interfaces will move vertically relative to the float depth and q_d in (1) must be recalculated each time the float is updated.

Vertical Adjustment of Isopycnic Floats

After horizontal advection is performed on isopycnic floats, their pressure depth is set to the depth of the reference isopycnic surface as follows: The first step is to calculate the pressure depth of the reference isopycnic surface at each of the selected pressure grid points surrounding the float. This is accomplished by first identifying the two model layers whose densities bracket the isopycnic reference density, then linearly vertically interpolating between the middle pressure depths of each layer. The scalar estimates of the isopycnic surface depth at the surrounding grid points are then horizontally interpolated to the float location.

Runga-Kutta Time Interpolation

The fourth-order Runga-Kutta time interpolation algorithm is a well-known procedure widely available in numerical methods references and texts. The procedure requires estimates of the model velocity at the present time and at two earlier times. If three-dimensional Lagrangian floats are selected, the time interpolation is performed on all three velocity components; otherwise, it is performed only on the horizontal velocity components. Ideally, the time interval separating each of the velocity component fields used in the interpolation should be between one and two hours. In HYCOM, the user selects this time interval as an integer number of baroclinic time steps and is warned if the time interval does not fall between one and two hours.

At each model time when the float is advected, the velocity component fields at the earliest of the three times represent the previous float advection time. When the float update subroutine is called at the intermediate time, the only action taken is to store the intermediate fields required to advect the float at the next advection time. Thus, the floats are update only every second time the subroutine is called. In addition to horizontal velocity components, previous fields of w_l, $\partial p / \partial x$, and $\partial p / \partial y$ are saved so that the time interpolation of vertical velocity can be performed for three-dimensional Lagrangian floats.

Implementation of the Float Algorithm

Float Initialization

The user must first choose the velocity sampling time interval for the Runga-Kutta time interpolation. The model baroclinic time interval Δt_{bc} is set prior to starting the model run. The velocity sampling time interval Δt_{vel} should ideally be between one and two hours. When HYCOM is run at low horizontal resolution, Δt_{bc} is typically set to order one hour. In this case, the velocity sampling time interval is set to $\Delta t_{vel} = 2\Delta t_{bc}$. At higher resolution, the velocity-sampling interval is set to $\Delta t_{vel} = n_{vel}\Delta t_{bc}$, where n_{vel} is set to a value that insures that Δt_{vel} is between one and two hours.

With this selection made, the float position will be updated every $2n_{vel}$ baroclinic time steps using velocity fields at the present time t and at the two earlier times $t - n_{vel}\Delta t_{bc}$ and $t - 2n_{vel}\Delta t_{bc}$. The user must also choose the time interval between which float information is output for further analysis. The options are to output this information every $2n_{vel}$ baroclinic time steps (each time the drifter is updated, or to output it once every n_{out} days. For the float code to work properly, Δt_{bc} must be chosen so that there are an integer number (n_{bc}) of baroclinic time steps per day, while n_{vel} must be chosen so that $n_{bc} / 2n_{vel}$ is an integer.

The user must select the float type (Lagrangian, isopycnic, isobaric, mooring) prior to starting the model. Float deployment information is provided in the file float.inp, which contains four columns of data. The first column specifies the float/drifter/mooring type, with 0 for three-dimensional Lagrangian, 1 for isobaric, 2 for isopycnic, and 3 for mooring. The second column is deployment time in days from the start of the model run. If the value is zero, the float is released instantaneously. The third column is termination time in days from the start of the model run. If the value is zero, the float lasts until it runs aground. The fourth and fifth columns provide the initial longitude and latitude. Information in column six varies by float type. For Lagrangian floats, isobaric floats/drifters, and moorings, it contains the initial depth in meters. For isopycnic floats, it contains the reference isopycnic surface value.

Preliminary Calculations

The float subroutine is called every n_{vel} time steps. Float advection is performed every $2n_{vel}$ time steps. At the intermediate time steps where the floats are not advected, the subroutine stores only the old velocity and other fields required for the Runga-Kutta time interpolation. Interpolation of water properties to the float location is performed only every n_{out} time steps when float information is saved for further analysis.

Prior to executing the update loop for each individual float, several three-dimensional fields are calculated if necessary. If the floats are Lagrangian, fields required to estimate vertical velocity are calculated. Fields of w_l are estimated at model interfaces on pressure grid points. For the later interpolation of w_a to the float location, fields of $\partial p / \partial x$ are estimated on u grid points and fields of $\partial p / \partial y$ are estimated on v grid points. For all floats, the relative vorticity field is calculated on p grid points at the times when float information is output.

Identification of the Model Layer Containing the Float

Following the preliminary calculations, the main float loop is executed once for each float. The first step is to determine the model layer that contains the float. This is initially assumed to be the model layer k that contained the float during the previous time step. This is tested by interpolating the pressure depths of interfaces k and $k+1$ to the location of the float and determining if the previously determined pressure depth of the float falls between those limits. If the float is no longer within this layer (or if the float has just been released), a special algorithm is executed to determine the model layer containing the float. Moving down from the surface, interface pressure depths are interpolated to the float location until the new layer containing the float is identified. Once the model layer is identified, the ratio q_d is calculated using (1).

Float Advection

Once the model layer is known, the Runga-Kutta time interpolation algorithm is executed to move the float horizontally. For three-dimensional Lagrangian floats, the Runga-Kutta algorithm is also executed to move the float vertically. The depth of isopycnic floats is set to the depth of the reference isopycnic surface. For isobaric floats, the float is not moved vertically. For moorings, of course, neither horizontal nor vertical advection is performed.

Interpolation of Water Properties to the Float Location

Several water properties are interpolated to the floats and output for further analysis in the file float.out. The variables are longitude, latitude, float depth, water depth, temperature, salinity, density, relative vorticity, potential vorticity, vertical viscosity, and vertical temperature diffusivity. For moorings, water depth is replaced by vertical velocity while viscosity and temperature diffusivity are replaced by horizontal velocity components.

References

Garraffo, Z., D. A. J. Mariano, A. Griffa, C. Veneziani, and E. P. Chassignet, 2001a: Lagrangian data in a high resolution numerical simulation of the North Atlantic. I: Comparison with in-situ float data. J. Mar. Sys., 29, 157-176.

Garraffo, Z. D., A. Griffa, A. J. Mariano, and E. P. Chassignet, 2001b: Lagrangian data in a high resolution numerical simulation of the North Atlantic. II: On the pseudo-Eulerian averaging of Lagrangian data. J. Mar. Sys., 29, 177-200.

Mariano, A. J. and O. B. Brown, 1992: Efficient objective analysis of dynamically heterogeneous and nonstationary fields via the parameter matrix. Deep Sea Res., 39, 1255-1292.

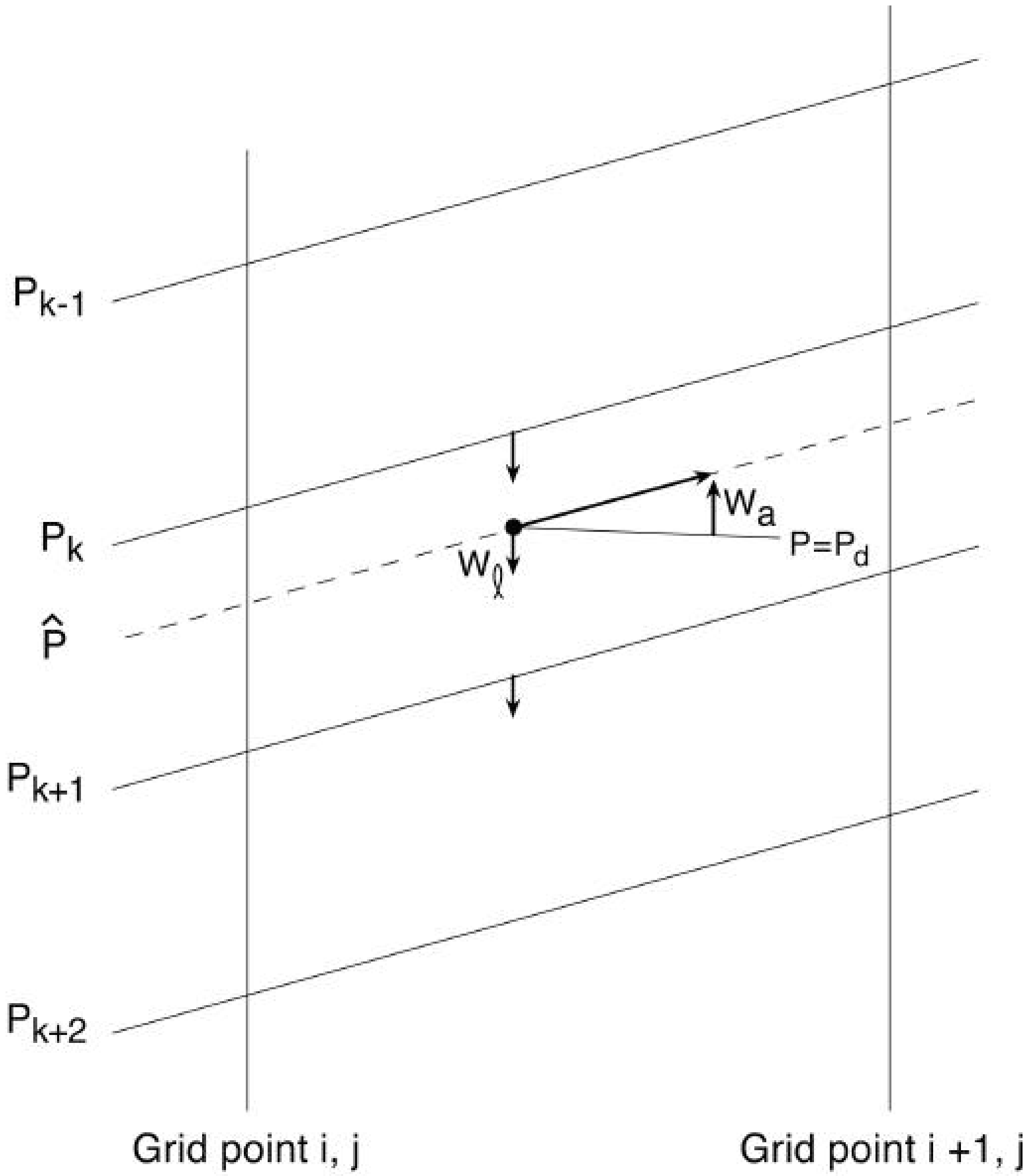

Figure. Diagram illustrating the estimation of float vertical velocity. This is a two dimensional case where the float is located at the latitude of the j row of grid points. The float is located in model layer k bounded by interfaces k and $k+1$. Local vertical velocity w_l obtained from the continuity equation is estimated using (4) at model interfaces, which is illustrated by the vertical arrows at the float location. When a float moves along a sloping model layer as illustrated above, the resulting advective vertical velocity w_a is estimated using (6).