USERS GUIDE for

A THREE-DIMENSIONAL, PRIMITIVE EQUATION, NUMERICAL OCEAN MODEL

George L. Mellor
Program in Atmospheric and Oceanic Sciences
Princeton University, Princeton, NJ 08544-0710

This revision: June 2004

Notes on the 1998 Revision This version of the users guide recognizes changes that have occurred since 1991. The code itself incorporates some recent changes. the fortran names, tmean, smean have been changed (globally) to tclim, sclim in oder to distiquish the function and treatment of these variables from that of rmean. the names, trnu, trnv, have been changed to drx2d, dry2d and the names, advuu, advvv, to adx2d, ady2d to more clearly indicate their functions. Instead of a wind driven closed basin, pom97.f now solves the problem of the flow through a channel which includes an island or a seamount at the center of the domain. Thus, subroutine bcond contains active open boundary conditions. These illustrative boundary conditions, however, are one set of many possibilities and, consequently, open boundary conditions for regional models pose difficult choices for users of the model. This 1998 revision contains a fuller discussion of open boundary conditions in section 16.

Notes on the 2002 revision The basic code, now labeled pom2k.f results from extensive tidying by John Hunter which includes more comments and lower case fortran variables, a move which apparently renders the code "modern". However the basic – we believe, well conceived - structure of the code remains unchanged.

As of this revision date, June 2004, there are over 1900 POM users of record.

Sponsor Acknowledgment: The development and application of the program has had many sponsors since 1977. They include the Geophysical Fluid Dynamics Laboratory/NOAA, Princeton University, Sea Grant/NOAA through the New Jersey Marine Sciences Consortium, the Department of Energy, Minerals Management Services/DOI, the National Ocean Services/NOAA, the Institute of Naval Oceanography and the Office of Naval Research/DOD.

Web site: http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom/

Title Page Illustration: North Atlantic velocity field on the 32.45 potential density surface. Courtesy Dr. Sirpa Häkkinen.

CONTENTS

1. INTRODUCTION

This report is documentation for a numerical ocean model created by Alan Blumberg and me around 1977. Subsequent contributions were made by Leo Oey, Jim Herring, Lakshmi Kantha and Boris Galperin and others. In recent years Tal Ezer has been an important force in research using the model and in helping others to use it. He has created and maintains the POM web site cited below. Institutionally, the model was developed and applied to oceanographic problems in the Atmospheric and Oceanic Sciences Program of Princeton University, the Geophysical Fluid Dynamics Laboratory of NOAA and Dynalysis of Princeton. Many sponsors, as acknowleged above, have supported the effort. Papers that either describe the numerical model (Blumberg and Mellor, 1987) or made use of the model are contained in the Reference Section and a more complete list is available on the POM home page at
http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom.

The model is oftentimes referenced as the Princeton Ocean Model (POM). The principal attributes of the model are as follows:

o It contains an imbedded second moment turbulence closure sub-model to provide vertical mixing coefficients.
o It is a sigma coordinate model in that the vertical coordinate is scaled on the water column depth.
o The horizontal grid uses curvilinear orthogonal coordinates and an "Arakawa C" differencing scheme.
o The horizontal time differencing is explicit whereas the vertical differencing is implicit. The latter eliminates time constraints for the vertical coordinate and permits the use of fine vertical resolution in the surface and bottom boundary layers.
o The model has a free surface and a split time step. The external mode portion of the model is two-dimensional and uses a short time step based on the CFL condition and the external wave speed. The internal mode is three-dimensional and uses a long time step based on the CFL condition and the internal wave speed.
o Complete thermodynamics have been implemented.

The turbulence closure sub-model is one that I introduced (Mellor, 1973) and then was significantly advanced in collaboration with Tetsuji Yamada (Mellor and Yamada,1974; Mellor and Yamada,1982). It is often cited in the literature as the Mellor-Yamada turbulence closure model (but, it should be noted that the model is based on turbulence hypotheses by Rotta and Kolmogorov which we extended to stratified flow cases). Here, the Level 2.5 model is used together with a prognostic equation for the

turbulence macroscale. The closure model is contained in subroutines PROFQ and ADVQ. A list of papers pertaining to the closure model is also included in the Reference section. A much more extensive list of references by user of POM is on the web site.

By and large, the turbulence model seems to do a fair job simulating mixed layer dynamics although there have been indications that calculated mixed layer depths are a bit too shallow (Martin, 1985). A recent paper (Mellor and Blumberg 2004) suggests ameliorative changes which are incorporated in this version. Also, wind forcing may be spatially smoothed and temporally smoothed. It is known that the latter process will reduce mixed layer thicknesses (Klein, 1980). Further study is required to quantify these effects.

The sigma coordinate system is probably a necessary attribute in dealing with significant topographical variability such as that encountered in estuaries or over continental shelf breaks and slopes. Together with the turbulence sub-model, the model produces realistic bottom boundary layers which are important in coastal waters (Mellor, 1985) and in tidally driven estuaries (Oey et al., 1985a, b) which the model can simulate since it does have a free surface. More recently, we find that bottom boundary layers are important for deep water formation processes (Zavatarelli and Mellor, 1995; Jungclaus and Mellor, 1996; Baringer and Price, 1996; Ezer and Mellor, 2004) and for the maintenance of the baroclinicity of oceans basins (Mellor and Wang, 1996).

The horizontal finite difference scheme is staggered and, in the literature, has been called an Arakawa C-grid. The horizontal grid is a curvilinear coordinate system, or as a special case, a rectilinear coordinate system may be easily implemented. The advection, horizontal diffusion and, in the case of velocity, the pressure gradient and Coriolis terms are contained in subroutines advt, advq, advct, advu, advv and advave. The horizontal differencing could be changed without affecting the overall logic of the program or the remaining subroutines. The vertical diffusion is handled in subroutines, proft, profq, profu and profv.

The specific program that is now supplied to outside users (as of June 1996) simulates the flow, east to west across a seamount with a prescribed vertical temperature stratification, constant salinity, zero surface heat and salinity flux and a zero wind stress distribution although wind stress may be easily applied. The program should run with no additional data requirements. The open boundary conditions specified in subroutine bcond for this problem are a sampling of many possible open boundary conditions. I leave it to users to invent their own problems, defined by topography, horizontal grid (rectilinear, where dx(i, j) is specified as a function of i and dy(i, j) as a function of j, or a more general orthogonal curvilinear grid in which case dx and dy are both functions of i and j), vertical sigma grid and boundary conditions. Users may need to alter program

pom2k and subroutine bcond; in principal, there should be no need to alter any of the other subroutines.

The present program code is written in standard FORTRAN 77. There are other versions in existence such as a non-Boussinesq version and a more general vertical coordinate version of which the sigma coordinate is a special case. However, we only support and maintain the sigma version.

Provision has been made so that the 2-D (external mode) portion of the model can be run *cum sole*. In this case, the bottom shear stress, normally a consequence of the 3-D calculation and the turbulence mixing coefficient, is replaced by a quadratic drag relation. The code may also be run in a diagnostic mode where the thermodynamic properties are invariant in time.

Users will need to write their own code to set up their own problem dependent, initial conditions and lateral and surface boundary conditions; see Appendix B. We can, however, supply simple subroutines that convert data at constant z-levels to sigma coordinates and vice versa.

To access pom2k.f and other files through the Internet, type **ftp ftp.aos.princeton.edu**; when prompted for your name, type **anonymous**; when prompted for a password, type your iternet address; after receiving a guest login ok, type **cd pub/pom**. You may list filenames with the **ls** command. You may download with the command **get filename**. Type **quit** to terminate.

Alternately, check the POM web page on
http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom

The current code is called pom2k.f. To run the code, transfer pom2k.f and pom2k.c to a directory, compile and run. A netCDF utility, pom2k.n, is also available and may be downloaded to create netCDF output.

2. THE BASIC EQUATIONS

The basic equations have been cast in a bottom following, sigma coordinate system which is illustrated in Figure 1. The reader is referred to Phillips (1957), Blumberg and Mellor (1980,1987) or Appendix D for a derivation of the sigma coordinate equations which are based on the transformation,

$$x^* = x,\ y^* = y,\ \sigma = \frac{z - \eta}{H + \eta},\ t^* = t \qquad \text{(1a, b, c, d)}$$

where *x,y,z* are the conventional cartesian coordinates; $D \equiv H + \eta$ where $H(x, y)$ is the bottom topography and η (*x, y, t*) is the surface elevation. Thus, σ ranges from σ = 0 at $z = \eta$ to σ = -1 at $z = \tilde{H}$. After conversion to sigma coordinates and deletion of the asterisks, the basic equations may be written (in horizontal cartesian coordinates),

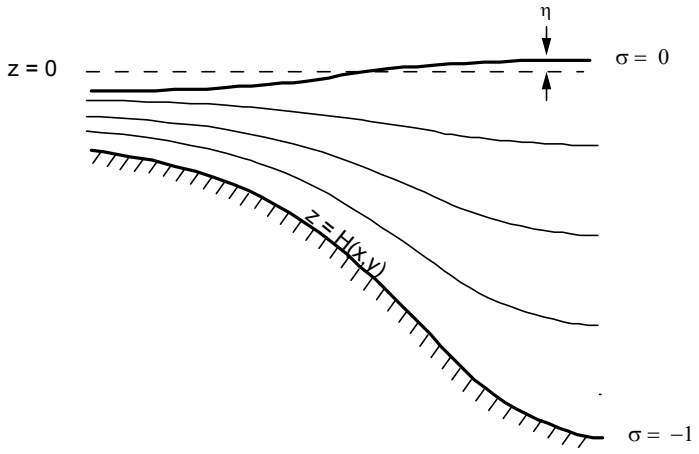


Figure 1. The sigma coordinate system.

$$\frac{\partial DU}{\partial x} + \frac{\partial DV}{\partial y} + \frac{\partial \omega}{\partial \sigma} + \frac{\partial \eta}{\partial t} = 0 \tag{2}$$

$$\frac{\partial UD}{\partial t} + \frac{\partial U^2 D}{\partial x} + \frac{\partial UVD}{\partial y} + \frac{\partial U\omega}{\partial \sigma} - fVD + gD\frac{\partial \eta}{\partial x}$$

$$+ \frac{gD^2}{\rho_o}\int_\sigma^o \left[\frac{\partial \rho'}{\partial x} - \frac{\sigma'}{D}\frac{\partial D}{\partial x}\frac{\partial \rho'}{\partial \sigma'}\right] d\sigma' = \frac{\partial}{\partial \sigma}\left[\frac{K_M}{D}\frac{\partial U}{\partial \sigma}\right] + F_x \tag{3}$$

$$\frac{\partial VD}{\partial t} + \frac{\partial UVD}{\partial x} + \frac{\partial V^2 D}{\partial y} + \frac{\partial V\omega}{\partial \sigma} + fUD + gD\frac{\partial \eta}{\partial y}$$

$$+ \frac{gD2}{\rho_o}\int_\sigma^o \left[\frac{\partial \rho'}{\partial y} - \frac{\sigma'}{D}\frac{\partial D}{\partial y}\frac{\partial \rho'}{\partial \sigma'}\right] d\sigma' = \frac{\partial}{\partial \sigma}\left[\frac{K_M}{D}\frac{\partial V}{\partial \sigma}\right] + F_y \tag{4}$$

$$\frac{\partial TD}{\partial t} + \frac{\partial TUD}{\partial x} + \frac{\partial TVD}{\partial y} + \frac{\partial T\omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_H}{D}\frac{\partial T}{\partial \sigma}\right] + F_T - \frac{\partial R}{\partial z} \tag{5}$$

$$\frac{\partial SD}{\partial t} + \frac{\partial SUD}{\partial x} + \frac{\partial SVD}{\partial y} + \frac{\partial S\omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_H}{D}\frac{\partial S}{\partial \sigma}\right] + F_S \tag{6}$$

$$\frac{\partial q^2 D}{\partial t} + \frac{\partial Uq^2 D}{\partial x} + \frac{\partial Vq^2 D}{\partial y} + \frac{\partial \omega q^2}{\partial \sigma} = \frac{\partial}{\partial \sigma}\left[\frac{K_q}{D}\frac{\partial q^2}{\partial \sigma}\right]$$

$$+\frac{2K_M}{D}\left[\left(\frac{\partial U}{\partial\sigma}\right)^2+\left(\frac{\partial V}{\partial\sigma}\right)^2\right]+\frac{2g}{\rho_o}K_H\frac{\partial\tilde{\rho}}{\partial\sigma}-\frac{2Dq^3}{B_1\ell}+F_q \tag{7}$$

$$\frac{\partial q^2\ell D}{\partial t}+\frac{\partial Uq^2\ell D}{\partial x}+\frac{\partial Vq^2\ell D}{\partial y}+\frac{\partial\omega q^2\ell}{\partial\sigma}=\frac{\partial}{\partial\sigma}\left[\frac{K_q}{D}\frac{\partial q^2\ell}{\partial\sigma}\right]$$

$$+E_1\ell\left(\frac{K_M}{D}\left[\left(\frac{\partial U}{\partial\sigma}\right)^2+\left(\frac{\partial V}{\partial\sigma}\right)^2\right]+E_3\frac{g}{\rho_o}K_H\frac{\partial\tilde{\rho}}{\partial\sigma}\right)-\frac{Dq^3}{B_1}\tilde{W}+F_\ell \tag{8}$$

Definitions of the variables are contained in section 3. Note that ω is the transformed vertical velocity; physically, ω is the velocity component normal to sigma surfaces. The transformation to the Cartesian vertical velocity is

$$W=\omega+U\left(\sigma\frac{\partial D}{\partial x}+\frac{\partial\eta}{\partial x}\right)+V\left(\sigma\frac{\partial D}{\partial y}+\frac{\partial\eta}{\partial y}\right)+\sigma\frac{\partial D}{\partial t}+\frac{\partial\eta}{\partial t}$$

The so-called wall proximity function is prescribed according to $\tilde{W}=1+E_2(\ell/kL)$ where $L^{-1}=(\eta-z)^{-1}+(H-z)^{-1}$. Also, $\partial\tilde{\rho}/\partial\sigma\equiv\partial\rho/\partial\sigma-c_s^{-2}\partial p/\partial\sigma$ (see discussion of static stability in Appendix A) where c_s is the speed of sound. Note that T is potential temperature (see Appendix A).

In equations (3) and (4), ρ_{MEAN} should be subtracted from ρ to form ρ' before the integration is carried out in subroutine BAROPG. ρ_{MEAN} is generally the initial density field which is area averaged on z-levels and then transferred to sigma coordinates in the exact same way as the initial density field. This procedure should reduce the truncation errors associated with the calculation of the pressure gradient term in sigma coordinate over steep topography (see Mellor et al., 1994 and Mellor et al. 1998 for evaluation of this error in POM).

The horizontal viscosity and diffusion terms are defined according to:

$$F_x\equiv\frac{\partial}{\partial x}\left(H\tau_{xx}\right)+\frac{\partial}{\partial y}\left(H\tau_{xy}\right) \tag{9a}$$

$$F_y\equiv\frac{\partial}{\partial x}\left(H\tau_{xy}\right)+\frac{\partial}{\partial y}\left(H\tau_{yy}\right) \tag{9b}$$

where

$$\tau_{xx} = 2A_M \frac{\partial U}{\partial x}, \quad \tau_{xy} = \tau_{yx} = A_M\left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right), \quad \tau_{yy} = 2A_M \frac{\partial V}{\partial y} \qquad (10a,b,c)$$

Also,

$$F_\phi \equiv \frac{\partial}{\partial x}\left(Hq_x\right) + \frac{\partial}{\partial y}\left(Hq_y\right) \qquad (11)$$

where

$$q_x \equiv A_H \frac{\partial \phi}{\partial x}, \quad q_y \equiv A_H \frac{\partial \phi}{\partial y} \qquad (12a,b)$$

and where ϕ. represents T, S, q^2 or $q^2\ell$. It should be noted that these horizontal diffusion terms are not what one would obtain by transforming the conventional forms to the sigma coordinate system. Justification for the present forms will be found in Mellor and Blumberg (1985) and relate to the fact that we wish to maintain a valid bottom boundary layer simulation in the face of horizontal diffusion which may be large. The penalty for this is that (12a,b) in sigma coordinates can introduce vertical fluxes even when isotherms and isohalines are flat in cartesian coordinates. The remedy for this is, first, the use of a Smagorinsky diffusivity (see below) so that, at least when velocities are small or nil, so are the values of q_x and q_y. The second remedy is that, before executing (12a, b) for temperature or salinity, we first subtract T_{CLIM} and S_{CLIM} which are "climatologies" of T and S. The latter may be true climatologies (e.g.; Levitus) or approximations such as temperature and salinities which are area averaged prior to transfer to sigma coordiates (in which case, they are treated the same as ρ_{MEAN}). If something like a Levitus climatology is used, then most of the vertical component of the diffusion is removed; furthermore, the diffusion terms tend to slowly drive the scalars back to climatology rather than to a horizontally homogeneous state as in the case of z - level models. The third remedy is make use of a small diffusivity relative to viscosity. Thus, the value, $TPRNI \equiv A_H / A_M$, can generally set to a small number, say 0.2, or even zero in some cases.

It should be noted that the treatment in (9a,b), (10a,b), (11) and (12a,b)allows for a realistic treatment of bottom boundary layers. The bottom boundary layer is important in tidally driven regions, in wind driven coastal regions and according to Mellor and Wang (1996), in deep ocean basins.

In (9a, b) and (11), H is used in place of D for the small algorithmic simplification it offers for terms whose physical significance is questionable.

The Smagorinsky Diffusivity

We generally use the Smagorinsky diffusivity for horizontal diffusion although a constant or biharmonic diffusion can and has been used instead. The Smagorinsky formula is,

$$A_M = C\Delta x\Delta y \frac{1}{2}\left|\nabla \mathbf{V} + (\nabla \mathbf{V})^T\right|$$

where $\left|\nabla \mathbf{V} + (\nabla \mathbf{V})^T\right| = [(\partial u/\partial x)^2 + (\partial v/\partial x + \partial u/\partial y)^2/2 + (\partial v/\partial y)^2]^{1/2}$. Values of C (the HORCON parameter) in the range, 0.10 to 0.20 seem to work well, but, if the grid spacing is small enough (Oey *et al*, 1985a,b), C can be nil. An advantage of the Smagorinsky relation is that C is non-dimensional; related advantages are that A_M decreases as resolution improves and that A_M is small if velocity gradients are small.

Vertical Boundary Conditions.

The vertical boundary conditions for (2) are

$$\omega(0) = \omega(-1) = 0 \qquad (13a,b)$$

However, if there is to be surface throughflow of (usually fresh) water, $\omega(0) \neq 0$.

The surface boundary conditions for (3) and (4) are

$$\frac{K_M}{D}\left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma}\right) = -\left(< wu(0) >, < wv(0) >\right),\ \sigma \rightarrow 0 \qquad (14a,b)$$

where the right hand side of (14a,b) is the input values of the surface turbulence momentum flux (the stress components are opposite in sign). The bottom boundary conditions are

$$\frac{K_M}{D}\left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma}\right) = C_z\left[U^2 + V^2\right]^{1/2}(U, V),\ \sigma \rightarrow -1 \qquad (14c,d)$$

where

$$C_z = MAX\left[\frac{\kappa^2}{\left[\ln\{(1+\sigma_{kb-1})H/z_o\}\right]^2}, 0.0025\right] \qquad (14e)$$

κ = 0.4 is the von Karman constant and z_o is the roughness parameter. Equations (14c,d,e) can be derived by matching the numerical solution to the "law of the wall". Numerically, they are applied to the first grid points nearest the bottom. Where the

bottom is not well resolved, $(1+\sigma_{kb-1})H/z_o$ is large and (14e) reverts to a constant drag coefficient, 0.0025. The boundary conditions on (5) and (6) are

$$\frac{K_H}{D}\left(\frac{\partial T}{\partial \sigma},\frac{\partial S}{\partial \sigma}\right) = -\left(<w\theta(0)>\right) \ , \ \sigma \rightarrow 0 \qquad (15a,b)$$

$$\frac{K_H}{D}\left(\frac{\partial T}{\partial \sigma},\frac{\partial S}{\partial \sigma}\right) = 0 \ , \qquad \sigma \rightarrow -1 \qquad (15c,d)$$

The boundary conditions for (7) and (8) are

$$\left(q^2(0), q^2\ell(0)\right)=\left(B_1^{2/3}u_\tau^2(0), 0\right) \qquad (16a,b)$$

$$\left(q^2(-1), q^2\ell(-1)\right)=\left(B_1^{2/3}u_\tau^2(-1), 0\right) \qquad (16c,d)$$

where B_1 is one of the turbulence closure constants and u_τ is the friction velocity at the top or bottom as denoted in (16a) and (16c). In pom97.f and later versions, (16a) has been replaced by $q^2\ell(\sigma_1) = q^2(\sigma_1)\kappa D\sigma_1$ where σ_1 is the value of σ corresponding to k =1, it is believed that this averts some numerical noise in some applications.

The Vertically Integrated Equations

The equations, governing the dynamics of coastal circulation, contain fast moving external gravity waves and slow moving internal gravity waves. It is desirable in terms of computer economy to separate the vertically integrated equations (external mode) from the vertical structure equations (internal mode). This technique, known as mode splitting (Simons, 1974; Madala and Piacsek, 1977) permits the calculation of the free surface elevation with little sacrifice in computational time by solving the velocity transport separately from the three-dimensional calculation of the velocity and the thermodynamic properties.

The velocity external mode equations are obtained by integrating the internal mode equations over the depth, thereby eliminating all vertical structure. Thus, by integrating Equation (2) from $\sigma = -1$ to $\sigma = 0$ and using the boundary conditions (13a,b), an equation for the surface elevation can be written as

$$\frac{\partial \eta}{\partial t} + \frac{\partial \overline{U}D}{\partial x} + \frac{\partial \overline{V}D}{\partial y} = 0 \qquad (17)$$

After integration, the momentum equations, (3) and (4), become

$$\frac{\partial \overline{U}D}{\partial t}+\frac{\partial \overline{U}^2 D}{\partial x}+\frac{\partial \overline{U}\overline{V}D}{\partial y}-\tilde{F}_x-f\overline{V}D+gD\frac{\partial \eta}{\partial x}=-<wu(0)>+<wu(-1)> \\ +G_x-\frac{gD}{\rho_o}\int_{-1}^{o}\int_{\sigma}^{o}\left[D\frac{\partial \rho'}{\partial x}-\frac{\partial D}{\partial x}\sigma'\frac{\partial \rho'}{\partial \sigma}\right]d\sigma' d\sigma \qquad (18)$$

$$\frac{\partial \overline{V}D}{\partial t}+\frac{\partial \overline{U}\overline{V}D}{\partial x}+\frac{\partial \overline{V}^2 D}{\partial y}-\tilde{F}_y+f\overline{U}D+gD\frac{\partial \eta}{\partial y}=-<wv(0)>+<wv(-1)> \\ +G_y-\frac{gD}{\rho_o}\int_{-1}^{o}\int_{\sigma}^{o}\left[D\frac{\partial \rho'}{\partial y}-\frac{\partial D}{\partial y}\sigma'\frac{\partial \rho'}{\partial \sigma}\right]d\sigma' d\sigma \qquad (19)$$

The overbars denote vertically integrated velocities such as

$$\overline{U}\equiv\int_{-1}^{o}U\,d\sigma\ . \qquad (20)$$

The wind stress components are $-<wu(0)>$ and $-<wv(0)>$, and the bottom stress components are $-<wu(-1)>$ and $-<wv(-1)>$. The quantities $\tilde{F}_x$ and $\tilde{F}_y$ are defined according to

$$\tilde{F}_x=\frac{\partial}{\partial x}\left[H2\overline{A}_M\frac{\partial \overline{U}}{\partial x}\right]+\frac{\partial}{\partial y}\left[H\overline{A}_M\left(\frac{\partial \overline{U}}{\partial y}+\frac{\partial \overline{V}}{\partial x}\right)\right] \qquad (21a)$$

and

$$\tilde{F}_y=\frac{\partial}{\partial y}\left[H2\overline{A}_M\frac{\partial \overline{V}}{\partial y}\right]+\frac{\partial}{\partial x}\left[H\overline{A}_M\left(\frac{\partial \overline{U}}{\partial y}+\frac{\partial \overline{V}}{\partial x}\right)\right] \qquad (21b)$$

The so-called dispersion terms are defined according to

$$G_x=\frac{\partial \overline{U}2D}{\partial x}+\frac{\partial \overline{U}\overline{V}D}{\partial y}-\tilde{F}_x-\frac{\partial \overline{U2}D}{\partial x}-\frac{\partial \overline{UV}D}{\partial y}+\overline{F}_x \qquad (22a)$$

$$G_y=\frac{\partial \overline{U}\overline{V}D}{\partial x}+\frac{\partial \overline{V}2D}{\partial y}-\tilde{F}_y-\frac{\partial \overline{UV}D}{\partial x}-\frac{\partial \overline{V2}D}{\partial y}+\overline{F}_y \qquad (22b)$$

Note that, if A_M is constant in the vertical, then the "F" terms in (22a) and (22b) cancel. However, we account for possible vertical variability in the horizontal diffusivity; such is the case when a Smagorinsky type diffusivity is used. As detailed below, all of the terms on the right side of (18) and (19) are evaluated at each internal time step and then held constant throughout the many external time steps. If the external mode is executed *cum sole*, then $G_x = G_y = 0$.

3. FORTRAN SYMBOLS

In the following table, we list the FORTRAN symbols followed by their corresponding analytical symbols in parentheses and a brief description of the symbols. Not explicitly tabulated are the suffixes b, blank and f which are appended to many of the variables to denote the time levels n - 1, n and n + 1 respectively.

Indices

i, j (i, j)	horizontal grid indexes
im, jm	outer limits of i and j
k (k)	vertical grid index; k = 1 at the top and k = kb at the bottom
iint (n)	internal mode time step index
iext	external mode time step index

Constants

days	Specifies runtime (days)
dte (Δt_e)	external mode time step, (s)
dti (Δt_i)	internal mode time step, (s)
horcon(C)	the coefficient of the Smagorinsky diffusivity
iend	total internal mode time steps
iprint	the interval in iint at which variables are printed
isplit	dti/dte
kappa (κ)	von Karman's constant = 0.4
mode	if mode = 2, a 2-D calculation is performed if mode = 3, a 3-D prognostic calculation is performed if mode = 4, a 3-D diagnostic calculation is performed
nread (0 or 1)	(does not or does) expect an beginning restart file
rfe, rfw, rfn, rfs	= 1 or 0 on the four open boundaries; for use in BCOND
smoth (α)	parameter in the temporal smoother
tprni (A_H/A_M)	inverse, horizontal, turbulence Prandtl number
r, ad1, ad2	Constants in the radiative penetrative equation, dependent on Jerlov type
umol	background vertical diffusivity
tbias, sbias	temperature, salinity bias: for 32 bit arithmetic, may reduce roudoff error.

One-dimensional Arrays

z(σ)	sigma coordinate which spans the domain, z = 0 (surface) to z = -1 (bottom)
zz	sigma coordinate, intermediate between z
dz($\delta\sigma$)	= z(k)~ - z(k+1)
dzz	= zz(k)~ - zz(k+1)

Two-dimensional Arrays

aam2d	vertical average of aam(m^2 s^{-1})
art, aru, arv	cell areas centered on the variables, T, U and V respectively (m^2)
advua, advva	sum of the second, third and fourth terms in equations (18) and (19)
adx2d, ady2d	vertical integrals of advx, advy; also the sum of the fourth, fifth and sixth terms in equations (22a,b)
cor (*f*)	the Coriolis parameter (s^{-1})
curv2d	the vertical average of curv
dum	Mask for the *u* component of velocity; = 0 over land; = 1 over water
dvm	Mask for the *v* component of velocity; = 0 over land; = 1 over water
fsm	Mask for scalar variables; = 0 over land; = 1 over water
dx (h_x or δx)	grid spacing (m)
dy (h_y or δy)	grid spacing (m)
el (η)	the surface elevation as used in the external mode (m)
et (η)	the surface elevation as used in the internal mode and derived from el (m)
eg (η)	the surface elevation also used in theinternal mode for the pressure gradient and derived from el (m)
d (*D*)	= h + el (m)
dt (*D*)	= h + et (m)
drx2d, drx2d	vertical integrals of drhox and drhoy
e_atmos	Atmospheric pressure in equivalent meters of water (m)

h (H)	the bottom depth (m)
swrad	short wave radiation incident on the ocean surface (m s^{-1}K)
ua, va ($\overline{U},\overline{V}$)	vertical mean of u,v (m s^{-1})
ut, vt ($\overline{U},\overline{V}$)	ua,va time averaged over the interval, DT = dti (m s^{-1})
wusurf, wvsurf	($<wu(0)>$, $<wv(0)>$) momentum fluxes at the surface (m^2s^{-2})
wubot, wubot	($<wu(-1)>$, $<wv(-1)>$) momentum fluxes at the bottom (m^2s^{-2})
wtsurf, wssurf	($<wt(0)>$, $<ws(0)>$) temperature and salinity fluxes at the surface (ms^{-1} K, ms^{-1} psu)
vflux	Volume flux through water column surface (ms^{-1})

Three-dimensional Arrays

advx, advy	horizontal advection and diffusion terms in equations (3) and (4)
aam (A_M)	horizontal kinematic viscosity (m^2 s^{-1})
aah (A_H)	horizontal heat diffusivity = TPRNI*AAM
curv ($\tilde{f}$)	curvature terms; see equation (28)
l (ℓ)	turbulence length scale
km (K_M)	vertical kinematic viscosity (m^2s^{-1})
kh (K_H)	vertical diffusivity (m^2s^{-1})
drhox $\left(gDh_y\rho_o^{-1}\left[-D\int_\sigma^0 \delta_x\rho'\,\delta\sigma' + \delta_x D\int_\sigma^0 \sigma'\delta\rho'\right]\right)$	x-component of the internal baroclinic pressure gradient subtract rmean from density before integrating
drhoy $\left(gDh_x\rho_o^{-1}\left[-D\int_\sigma^0 \delta_y\rho'\,\delta\sigma' + \delta_y D\int_\sigma^0 \sigma'\delta\rho'\right]\right)$	y-component of the internal baroclinic pressure gradient subtract rmean from density before integrating
rad (R)	short wave radiation flux (ms^{-1}K). Sign is the same as wtsurf
q2 (q^2)	twice the turbulence kinetic energy (m^2s^{-2})
q2l ($q^2\ell$)	q2 x the turbulence length scale (m^3s^{-2})
t (T)	potential temperature (K)
s (S)	salinity (psu)
rho (ρ -1000.)/rhoref	density (non-dim.). Default rhoref=1025.

u, v (*U, V*)	horizontal velocities (m s^{-1})
w (ω)	sigma coordinate vertical velocity (m s^{-1})
rmean	density field which is horizontally averaged before transfer to sigma coordinates.
tclim	a stationary temperature field which approximately has the same vertical structure as T.
sclim	a stationary salinity field which approximately has the same vertical structure as S.

The variables, uf and vf, are used to denote the n+1 time level for u and v respectively. However, in order to save memory they are also used to represent the n+1 time level for t and s and for q2 and q2l respectively. As soon as uf, vf are calculated for each representation, the time level is reset.

4. THE NUMERICAL SCHEME

Figure 2 is the flow chart for the program in simplified form. The external mode calculation is contained in program pom2k.

External-Internal Mode Interaction.

The external mode calculation in MAIN results in updates for surface elevation, el, and the vertically averaged velocities, ua, va. The internal mode calculation results in updates for u,v,t,s and the turbulence quantities.

Fig. 3 illustrates the time stepping process for the external and internal mode. Assume everything is known at t^{n-1} and t^n (the previous leap frog time step having just been completed). Integrals involving the baroclinic forcing and the advective terms are supplied to the external mode along with the bottom stress, a process which is labeled "Feedback" in Fig. 3; their values are held constant during $t^n < t < t^{n+1}$. The external mode "leap frogs" many times, with the time step, dte, until $t = t^{n+1}$. The vertical and time averaged velocities, utf, vtf, and those from the previous time step, utb,vtb, are time averages of the external variables, ua, va. The internal and external modes have different truncation errors so that the vertical integrals of the internal mode velocity may depart slightly from (ua,va) during the course of a long integration. We therefore adjust the internal velocity, u so that its vertical integrals are the mean of utb and utf. Similarly, v is adjusted to the mean of vtb and vtf.

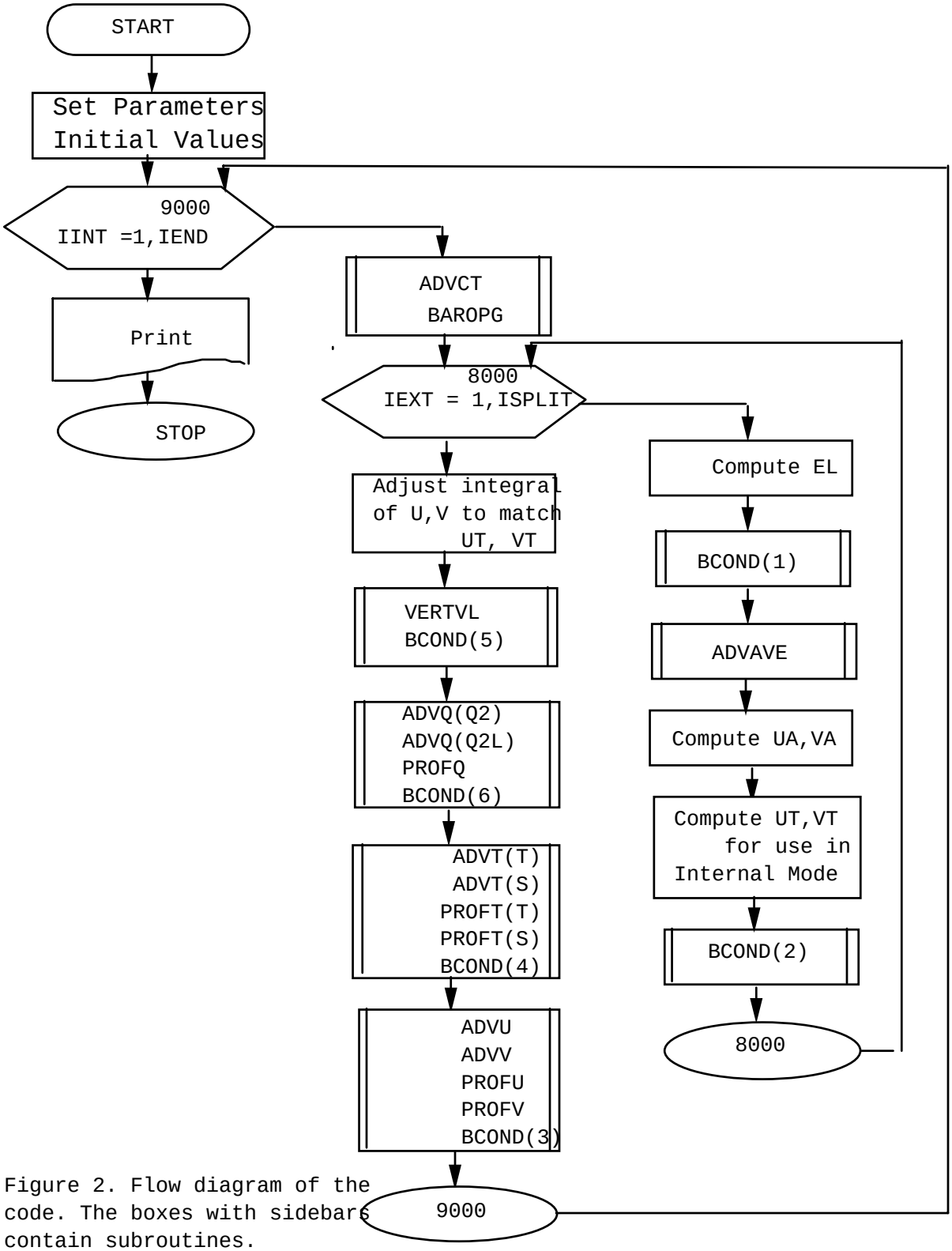


Figure 2. Flow diagram of the code. The boxes with sidebars contain subroutines.

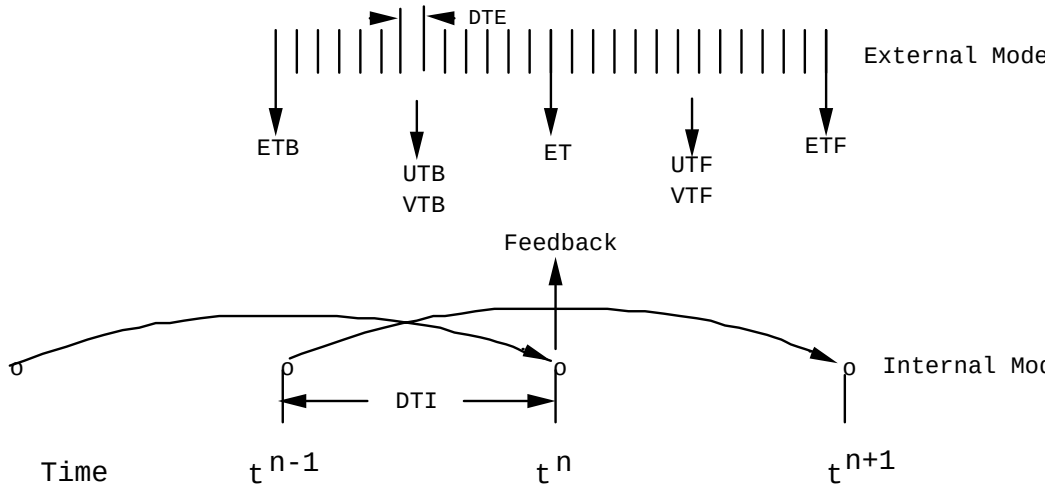


Figure 3. A simplified illustration of the interaction of the External Mode and the Internal Mode. The former uses a short time step, dte, whereas the latter uses a long time step, dti. The external mode primarily provides the surface elevation to the internal mode whereas, as symbolized by "Feedback", the internal mode provides intergrals of momentum advection, density integrals and bottom stress to the external mode.

Care is taken to relate etf to elf so that together with etb, saved from a previous time step, the internal velocities and etf and etb correctly satisfy the continuity equation, (17). Otherwise, the sigma coordinate equations for t, s will not be conservative.

Aside from the above, numerically important details, 0.5*(egf + egb) is used to obtain the elevation gradients for the internal mode "leap frog" from t^{n-1} to t^{n+1}. egb and egf are el, averaged over the intervals, t^{n-1} to t^{n} and t^{n} to t^{n+1} respectively. It is this maneuver that renders the internal mode immune to the CFL condition based on the barotropic wave speed. The governing wave speed is the baroclinic wave speed.

Structure of the Internal Mode Calculation.

The calculation of the three-dimensional (internal) variables is separated into a vertical diffusion time step and an advection plus horizontal diffusion time step. The former is implicit (to accommodate small vertical spacing near the surface) whereas the latter is explicit. To illustrate, consider the temperature equation,

$$\frac{\partial DT}{\partial T} + Adv(T) - Dif(T) = \frac{1}{D}\frac{\partial}{\partial \sigma}\left(K_H \frac{\partial T}{\partial \sigma}\right) - \frac{\partial R}{\partial \sigma} \qquad (23)$$

$Adv(T)$ and $Dif(T)$ represents the advection and horizontal diffusion terms. The solution is carried out in two steps. Thus, the advection and horizontal diffusion parts are differenced according to

$$\frac{\widetilde{D}\widetilde{T} - D^{n-1}T^{n-1}}{2\Delta t} = - Adv(T^n) + Dif(T^{n-1}) \tag{24}$$

and is solved by subroutine advt. The vertical diffusion part is differenced according to

$$\frac{D^{n+1}T^{n+1} - \widetilde{D}\widetilde{T}}{2\Delta t} = \frac{1}{D^{n+1}}\frac{\partial}{\partial\sigma}\left(K_H \frac{\partial T^{n+1}}{\partial\sigma}\right) - \frac{\partial R}{\partial\sigma} \tag{25}$$

and is solved by subroutine proft as detailed in section 9. Note that $\widetilde{D}\widetilde{T}$ can be any three-dimensional variable since it cancels in (24) and (25). Note that, in this subroutine, T^{n-1} is stored in tb, T^n in t and T^{n+1} in uf.

In the "leap frog" time differencing scheme, the solutions at odd time steps can diverge slowly from the solutions at the even time steps. This time splitting is removed by a weak filter (Asselin, 1972) where the solution is smoothed at each time step according to

$$T_s = T^n + \frac{\alpha}{2}\left(T^{n+1} - 2T^n + T^{n-1}\right)$$

where T_S is the smoothed solution; frequently, we use α = 0.05. This technique introduces less damping than either the Euler-backward or forward stepping techniques. After smoothing, T_S is reset to T^{n-1} and T^{n+1} to T^n.

Grid Arrangement

The staggered grid arrangement for the external mode is depicted in Fig. 4 and 5 for the external and internal grid respectively. These diagrams will be useful in understanding the coding in pom2k and in the "prof" and "adv" subroutines. Although the fortran nomenclature in the code may appear to be cartesian coordinates, the grid can be an orthogonal curvilinear grid. One merely needs to specify h_x(=dx(i, j)) and h_y(=dy(i, j)) as that associated with a particular grid. The advective operators in equations (2) to (8) and (17) to (19) are then described in a finite volume sense; i.e. Equation (5) or, rather, the *Adv* operator in (24), is written

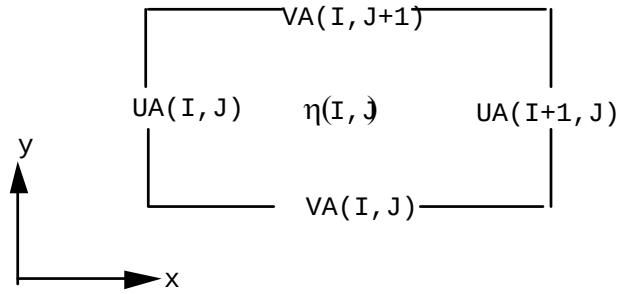


Figure 4. The two-dimensional external mode grid.

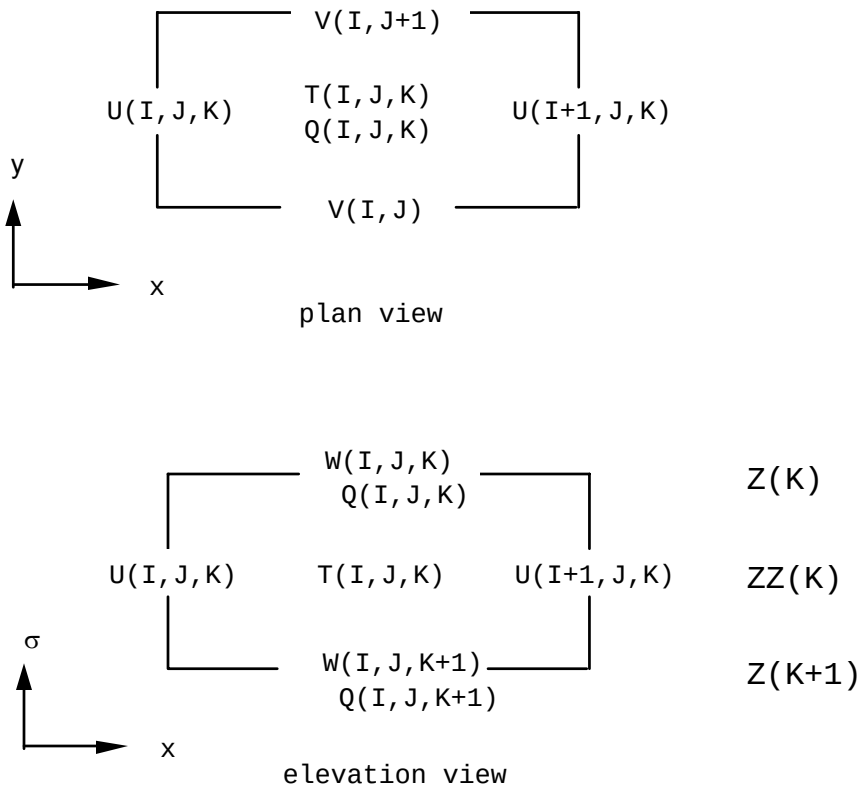


Figure 5. The three-dinensional internal mode grid. Q represents km, kh, q2, or q2l. t represents t,s or rho.

$$-Adv(T)h_x h_y \;=\; \delta_x\left(Dh_y UT\right) + \delta_y(Dh_x VT) \;+\; h_x h_y \frac{\delta_\sigma(\omega T)}{\delta\sigma} \qquad (26)$$

(where it might be more consistent to multiply through by $\delta\sigma$, but this has not been effected in the code). Thus Dh_yUT represents the transport of T and δ_x represents the difference in this quantity through the opposing faces of the volume element. We leave it to the code listing in subroutine advt to describe the exact method of differencing.

The differencing for the velocity is accomplished in a similar way but involves Coriolis and curvature terms. The advective term for U in equation (3) is

$$-Adv(U)h_x h_y \;=\; \delta_x\left(Dh_y UU\right) + \delta_y(Dh_x UV) \;+\; h_x h_y \frac{\delta\sigma(\omega U)}{\delta\sigma} - \tilde{f}VD\, h_x h_y \qquad (27)$$

where

$$\tilde{f} = \frac{V\delta_x(h_y)}{h_x h_y} \;-\; \frac{U\delta_y(h_x)}{h_x h_y} \qquad (28)$$

is the curvature term. In subroutine advct, the horizontal advection, diffusion and curvature terms are calculated (and stored in subroutines advx and advy) well in advance of subroutines advu and advv so that their vertical averages can be used in the external mode calculation. In subroutines advu and advv, the pressure gradient, Coriolis and vertical advection are included along with the terms imported from subroutine advct.

Time Step Constraints.

The Courant-Friedrichs-Levy (CFL) computational stability condition on the vertically integrated, external mode, transport equations limits the time step according

$$\Delta t_E \;\le\; \frac{1}{C_t}\left|\frac{1}{\delta x^2} \;+\; \frac{1}{\delta y^2}\right|^{-1/2} \qquad (29)$$

where $C_t = 2(gH)^{1/2} + U_{max}$; U_{max} is the expected, maximum velocity. There are other restrictions but in practice the CFL limit is the most stringent. The model time step is usually 90% of this limit. The internal mode has a much less stringent time step since the fast moving external mode effects have been removed. The time step criteria is analogous to that for the external mode given by Equation (26) and is

$$\Delta t_I \;\le\; \frac{1}{C_T}\left|\frac{1}{\delta x^2} \;+\; \frac{1}{\delta y^2}\right|^{-1/2} \qquad (30)$$

where $C_T = 2C + U_{max}$; C_T is the maximum internal gravity wave speed based on the gravest mode, commonly of order 2m/s, and U_{max} is the maximum advective speed. For typical coastal ocean conditions the ratio of the time steps, $\Delta t_I / \Delta t_E$ = dti/dte = isplit, is often a factor of 30 - 80 or larger. For more information on the sensitivity of POM to time steps, see Ezer et al. (2002)

Additional limits are imposed by horizontal diffusion of momentum or scalars are, for $A = A_M$ or $A = A_H$

$$\Delta t_I \leq \frac{1}{4A} \left| \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right|^{-1} \tag{31}$$

A limit imposed by rotation is

$$\Delta t_I < \frac{1}{f} = \frac{1}{2\Omega \sin \Phi} \tag{32}$$

where A_H is the horizontal diffusivity, Ω is the angular velocity of the earth Φ is the latitude. (31) and (32) are, however, not restrictive compared to (29) and (30).

5. pom2k.c

Common block definitions are contained in the file, pom2k.c. The file is then "include"d in each subroutine.

6. program pom2k and the external mode

The main program, pom2k, contains model initialization and, subsequently, internal mode time stepping via the index, iint. All of the internal mode (three-dimensional) subroutines, advq, profq, advu, profu, advv, profv, advt (for temperature or salinity), proft (for temperature or salinity) and dens are called once for each value of iint = 1 to iend.

Imbedded in the iint loop (which terminates at statement 9000) is the external mode iext loop which cycles isplit times. Note that dti/dte = isplit. The external mode solves for the vertically mean velocities and surface elevation using the density, dispersion terms and bottom stress created by the internal mode all of which are held constant throughout the many external mode, time steps.

The advective and horizontal diffusion terms in the external model are calculated by vertical integration of the corresponding internal terms, advx and advy, created in subroutine advct. The latter are available every internal time step. However, they are

updated by virtue of similar terms (but derived from the mean velocity) in subroutine advave. We find that this need not be done every external time step to maintain a stable calculation. subroutine advave which calculates these terms are called at intervals of ispadv; a typical value is ispadv = 5.

7. subroutine advave

This subroutine calculates the advective and horizontal diffusion terms for the external mode calculation contained in equations (18) and (19). If mode = 2, it also calculates the bottom friction from a quadratic drag equation; otherwise, in the standard three dimensional calculation, the bottom friction is determined by profu and profv quite naturally as a byproduct of the bottom boundary layer.

8. subroutine advt

This subroutine solves equation (24) for temperature or salinity (or any other scalar variable) which are labeled f internally. The operator, *Adv*(*f*), is written in the form of equation (26). As shown in the code listing, horizontal advective transports through the faces of the grid elements are computed in the form, d*u*f*dy and d*v*f*dx, using appropriate cell averages. Note that, in the code listing, dt is simply the external value, d, averaged over the internal time step. To the advective fluxes are added the horizontal diffusion fluxes. Before this occurs, tclim is subtracted from the actual temperature [see Mellor and Blumberg, 1986, and discussion after equation (12a, b)]. Then, the diffusion terms will slowly (or not if tprni =0) drive the calculated field back to climatology. As resolution improves, the diffusion terms decrease as dx*dy decreases. The vertical advective flux divergence is determined (and temporally stored in ff) and then combined with the horizontal transport divergence. Finally, the time step is executed and the new value is stored in ff.

9. subroutine proft

This subroutine solves (25) for temperature and salinity. We use the method described on p.198-201 of Richtmeyer and Morton (1967). The procedure described below will be also be used for u,v,q2 and q2l in which case the radiation term in (25) is either null or is replaced by source/sink turbulence terms. Subroutine proft as well as advt can be used to solve for other geochemical constituents besides temperature and salinity.

First, finite difference (25) with respect to σ [We note that $\widetilde{D} = D^{n+1}$ = dh; the choice is irrelevant so long as the same value of $\widetilde{D}$ is used in (24) and (25)]. Thus, with reference to the elevation view of Fig. 5,

$$f_k - \tilde{f}_k = \frac{dti2}{dh**2*dz_k}\left[\frac{kh_k}{dzz_{k-1}}*(f_{k-1}-f_k) - \frac{kh_{k+1}}{dzz_k}*(f_k - f_{k+1})\right] - \frac{dt2}{dh*dz_k}[rad_k - rad_{k+1}] \quad (33)$$

where $dz_k = z_k - z_{k+1}$, $dzz_k = zz_k - zz_{k+1}$ and f_k represents either temperature or salinity. In the above, we use subscripts for k instead of parenthetical enclosure to save space; we also omit the i, j indicies.

Solution Technique

Equation (33) can now be written as

$$-f_{k+1}*a_k + f_k*(a_k + c_k - 1) - f_{k-1}*c_k = d_k \quad (34)$$

where

$$a_k = -\frac{dti2*kh_{k+1}}{dh**2*dz_k*dzz_k} \quad (35a)$$

$$c_k = -\frac{dti2*kh_k}{dh**2*dz_k*dzz_{k-1}} \quad (35b)$$

$$d_k = -\tilde{f}_k + \frac{dti2}{dh*dz_k}[rad_k - rad_{k+1}] \quad (35c)$$

Now assume a solution of the form

$$f_k = ee_k * f_{k+1} + gg_k \quad (36)$$

Inserting f_{k+1} directly from (36) and f_{k-1} ,obtained from (36), into (34) and collecting coefficients of f_k and 1 yields

$$ee_k = \frac{a_k}{a_k + c_k*(1 - ee_{k-1}) - 1} \quad (37a)$$

$$gg_k = \frac{c_k*gg_{k-1} + d_k}{a_k + c_k*(1 - ee_{k-1}) - 1} \quad (37b)$$

The way the system works is as follows: All a_k's, c_k's and d_k's are calculated from (35a,b,c). Surface boundary conditions, discussed below, provide ee_1 and gg_1 and all of

the necessary ee_k's and gg_k's are obtained from the descending (as k increases towards the bottom) recursive relations (37a,b). Bottom boundary conditions provide f_{kb-1} where kb-1 is the grid point nearest the bottom. Thereafter all of the f_k's may be obtained from the ascending recursive relation (36).

Short Wave Radiation

To specify the short wave radiation, we use the classification of Jerlov(1976) as interpreted by Paulson and Simpson (1977), Thus,

$$rad_k = swrad * (r * \exp(z_k * dh/ad1) + (1 - r) * \exp(z_k * dh/ad2)) \tag{38}$$

where r, ad1 and ad2 are functions of ntp such that

ntp	1	2	3	4	5
Jerlov type	I	Ia	Ib	II	III
r	0.58	0.62	0.67	0.70	0.78
ad1 (m)	0.35	0.60	1.0	1.5	1.4
ad2 (m)	23.0	20.0	17.0	14.0	7.9

Surface and Bottom Boundary Conditions

To apply the surface boundary conditions where the surface flux is prescribed (prescribing the surface temperature is much simpler since $ee_1 = 0$, $gg_1 = f_1$) begins with (33) where for k=1

$$f_1 - \tilde{f}_1 = -\frac{dti2}{dh * dz_1}(wtsurf + rad_1 - rad_2) + a_1(f_1 - f_2)$$

where f_1 is the time step at n+1. Using (36) to eliminate f_2 and collecting coefficients of f_1 and 1 yields

$$ee_1 = \frac{a_1}{a_1 - 1} \tag{39a}$$

$$gg_1 = \left[\frac{dt2}{d * dz_1} * (wtsurf + rad_1 - rad_2) - \tilde{f}_1\right] * \left[\frac{1}{a_1 - 1}\right] \tag{39b}$$

At the bottom, we specify zero heat flux. A repeat of the above procedure leads to

$$f_{kb-1} = \frac{c_{kb-1} * gg_{kb-2} - \tilde{f}_{kb-1} + dti2 * (rad_1 - rad_2)/(dh * dz_1)}{c_{kb-1} * (1 - ee_{kb-2}) - 1} \qquad (40)$$

Four different surface boundary conditions can be selected by choosing the appropriate nbc parameter when calling proft:

nbc=1 - surface BC is wtsurf or wssurf (heat or salt flux BC)
nbc=2 - surface BC is wtsurf and swrad (heat flux and short wave radiation penetration)
nbc=3 - surface BC is tsurf or ssurf (SST or SSS BC)
nbc=4 - surface BC is tsurf and swrad (SST and short wave radiation penetration)

Note that wtsurf and swrad are *negative* when water column is *warming.* (To transfer values of heat flux given in Wm^{-2} to wtsurf in K m s^{-1}, divide by the factor $4.1876 \times 10^{6)}$.

10. subroutine baropg

This subroutine calculates the baroclinic, vertical integrals involving density in equation (3) and (4) after the equations have been written in finite volume form.

We note the fact that, in the code, $\rho_{MEAN} = \text{rmean}$ has been subtracted from ρ before the integrals are calculated. ρ_{MEAN} is the basin area average density which is mapped onto the sigma-grid just as the initial conditions were similarly mapped This procedure removes most of the truncation error in the transformed baroclinic terms which arise due to the subtraction of the two large terms involving $\partial\rho / \partial x$ and $D^{-1}(\partial D/\partial x)\sigma\partial\rho / \partial\sigma$ in (3) and similarly in (4).

11. subroutines advct, advu and advv

advct calculates the horizontal advection (including curvature terms) and the diffusion parts of (3) and (4) which are differenced in the manner of equation (27) and saved as advx and advy. These terms are vertically integrated and saved as adx2d and ady2d for use in the external mode calculation in program main. Originally, advct had been incorporated in advu and advv. However, it was determined by Oregon State University colleagues that advancing the calculation of horizontal advection terms (see Figure 2) for use in the external mode increased the model's intrinsic stability.

12. subroutines profu and profv

These subroutines are virtually identical to subroutine proft. However, the bottom boundary conditions are obtained from equations (14c,d,e).

13. subroutine advq

This subroutine is very similar to all the other "adv-" subroutines in that it calculates the advective terms for the the turbulence quantities, q2 and q2l.

14. subroutine profq

This subroutine first solves for the vertical part of the equations (7) and (8) for q2 and q2l in the manner of equation (25). The numerical procedure is the same as subroutine proft. The turbulence closure scheme as described by Mellor and Yamada(1982) is contained in this subroutine. A somewhat simplified version of the level 2 1/2 model is used here and is discussed in Galperin *et al* (1988) and Mellor (1989). A recent correction to the model is presented by Mellor (2001)

The vertical diffusivities, K_M and $K_{H,}$ are defined according to

$$K_M = q\ell S_M \tag{41a}$$

$$K_H = q\ell S_H \tag{41b}$$

The coefficients, S_M and S_H, are functions of a Richardson number given by

$$S_H[1-(3A_2B_2 + 18A_1A_2)G_H] = A_2[1-6A_1/B_1] \tag{42a}$$

$$S_M[1-9A_1A_2G_H] - S_H[(18A_12 + 9A_1A_2)G_H] = A_1[1 - 3C_1 - 6A_1/B_1] \tag{42b}$$

where

$$G_H = \frac{\ell^2}{q^2}\frac{g}{\rho_o}\left[\frac{\partial\rho}{\partial z} - \frac{1}{c_s^2}\frac{\partial p}{\partial z}\right] \tag{43c}$$

is a Richardson number. The five constants in (42a,b) are evaluated from neutral homogeneous and near surface turbulence data (law-of-the-wall region) and are found (Mellor and Yamada, 1982) to be (A_1, B_1, A_1, B_2, C_1) = (0.92, 16.6, 0.74, 10.1, 0.08). The stability functions limit to infinity as G_H approaches the value, 0.0288, a value larger than one expects to find in nature. The quantity, c_s^2, in the square brackets of (42c) is the speed of sound squared. In the code the vertical pressure gradient is obtained from the hydrostatic relation, of course, but here, the density is taken as a constant consistent with the pressure determination in subroutine dens; i.e., $\partial p/\partial z = -\rho_o g$

15. subroutine vertvl

This short subroutine integrates equation (2) to obtain the sigma coordinate transformed "vertical velocity" which, actually, is the velocity normal to sigma surfaces. Occasionally, check w(i, j, kb); if all is well, the code should yield very small values ($\sim 10^{-11}$). If there is to be a surface through-flow of (usually fresh) water, then w(i,j,1) = vflux(i,j) $\neq 0$

16. subroutine bcond

Lateral boundary conditions contiguous to coastlines are handled automatically by the masks dum, dvm and fsm. They set to zero the velocities normal to land boundaries. The landward tangential velocities in the horizontal friction terms are also set to zero. For a sigma coordinate system, the latter is of little importance since the minimum water depth next to the coast can be quite shallow so that bottom friction dominates over lateral friction. We often set the minimum depth at 10 m, but smaller values are possible. For example, where tides are present, negative values of d are to be avoided

Open boundaries are considerably more demanding and uncertain and there is a need for boundary condition specification for both the external and internal modes.

Table A collects a variety of open boundary conditions; they are by no means inclusive. If (A - 1) is used around <u>all</u> open boundaries, then it is necessary to insure that the horizontal integral of BC around the boundary is zero; otherwise, the average basin elevation can increase or decrease, possibly disastrously. This can also happen with the exclusive use of (A - 4).

Calculations do not seem overly sensitive to the velocity component tangential to the boundary, at least for low Rossby number flows. We often set it to zero; alternatively advective boundary conditions similar to (B - 3) have been used.

Table A: A list of possible external mode open boundary conditions. In the formulations, $c_e = \sqrt{gH}$. The variable *BC* is user specified and may be equated to the left sides of (A-1) to (A-3) where $\overline{U}$ and η are known *a priori*. The right sides of (A-4) and (A-5) need not necessarily be zero. This table has been greatly augmented from the original by Peter Holloway (School of Geography and Oceanography, University College, University of New South Wales, Australian Defence Force Academy, Australia) and edited by George Mellor. The table does not exhaust the list of possible boundary conditions. Please report errors.

Formula	Boundary	Code
Inflow condition: $D\overline{U} = BC$ (A -1)	EAST	uaf(im,j) = 2*bc(j)/(h(im,j)+elf(im,j) + h(imm1,j +elf(imm1,j))) elf(im,j) = elf(imm1,j) vaf(im,j) = set[1]
	WEST	uaf(2,j) = 2*bc(j)/(h(1,j)+elf(1,j) + h(2,j)+elf(2,j)) elf(1,j) = elf(2,j) vaf(1,j) = set
	NORTH	vaf(i,jm) = 2*bc(i)/(h(i,jm)+elf(i,jm) + h(i,jmm1) + elf(i,jmm1)) elf(i,jm) = elf(i,jmm1) uaf(i,jm) = set
	SOUTH	vaf(i,2) = 2*bc(i)/(h(i,1)+elf(i,1) + h(i,2)+elf(i,2)) elf(i,1) = elf(i,2) uaf(i,1) = set
Elevation condition: η = *BC* (A - 2)	EAST	elf(imm1,j) = bc(j) elf(im,j) = elf(imm1,j) cosmetic uaf(im,j) = uaf(imm1,j) vaf(im,j) = set
	WEST	elf(2,j) = bc(j) uaf(2,j) = uaf(3,j) vaf(1,j) = set
	NORTH	elf(i,jmm1) = bc(i) elf(i,jm) = elf(i,jmm1) cosmetic vaf(i,jm) = vaf(i,jmm1) uaf(i,jm) = set
	SOUTH	elf(i,2) = bc(i) vaf(i,2) = vaf(i,3) uaf(i,1) = set

[1] We use "set" to denote the prescription for the along-boundary component of velocity. If it is a known value then that value can be used. More often it is not known and the value, 0, is used.

Radiation: $H\bar{U} \pm c_e \eta = BC$[2] (A-3)	EAST	uaf(im,j) = sqrt(grav/h(imm1,j))* el(imm1,j) + bc(j) elf(im,j) = elf(imm1,j) vaf(im,j) = set
	WEST	uaf(2,j) = - sqrt(grav/h(2,j))* el(2,j)+bc(j) elf(1,j) = elf(2,j) vaf(1,j) = set
	NORTH	vaf(i,jm) = sqrt(grav/h(i,jmm1))* el(i,jmm1) + bc(i) elf(i,jm) = elf(i,jmm1) uaf(i,jm) = set
	SOUTH	vaf(i,2) = - sqrt(grav/h(i,2))* el(i,2)+bc(i) elf(i,1) = elf(i,2) uaf(i,1) = set
Radiation: $\frac{\partial \bar{U}}{\partial t} \pm c_e \frac{\partial \bar{U}}{\partial x} = 0$ (A-4)	EAST	gae = dte*sqrt(grav*h(im,j))/dx(im,j) uaf(im,j) = gae*ua(imm1,j) + (1.-gae)*ua(im,j) elf(im,j) = elf(imm1,j) vaf(im,j) = set
	WEST	gae = dte*sqrt(grav*h(2,j))/dx(2,j) uaf(2,j) = gae*ua(3,j) + (1.-gae)*ua(2,j) elf(1,j) = elf(2,j) vaf(1,j) = set
	NORTH	gae = dte*sqrt(grav*h(i,jm))/dy(i,jm) vaf(i,jm) = gae*va(i,jmm1) + (1.-gae)*va(i,jm) elf(i,jm) = elf(i,jmm1) uaf(i,jm) = set
	SOUTH	gae = dte*sqrt(grav*h(i,2))/dy(i,2) vaf(i,2) = gae*va(i,3) + (1.-gae)*va(i,2) elf(i,1) = elf(i,2) uaf(i,1) = set

Table B are open boundary conditions for the internal mode.As in the external mode, the choice for the normal velocities is unclear. One might presume that (B - 2) is to be preferred over (B - 1) since internal waves can pass through the boundary with little reflection. In some applications, that may be the case. However, we have seen cases (open boundaries with substantial inflows) where the "freedom" of (B - 2) can set up unphysical, but numerically valid, baroclinic structures interior to the boundary.

[2] The boundary forcing can be set to known values approximately balancing the left side; e. g., on the east, bc(j) = uabe(j)-sqrt(grav/h(imm1,j))* ele(j) where uabe(j) and ele(j) are specified values.

Radiation: $\frac{\partial \eta}{\partial t} \pm c_e \frac{\partial \eta}{\partial x} = 0$ (A-5)	EAST	gae = dte*sqrt(grav*h(imm1,j))/dx(imm1,j) elf(imm1,j) = gae*el(imm2,j) + (1.-gae) *el(imm1,j) elf(im,j) = elf(imm1,j) uaf(im,j) = uaf(imm1,j) vaf(im,j) = set
	WEST	gae = dte*sqrt(grav*h(2,j))/dx(1,j) elf(2,j) = gae*el(3,j) + (1.-gae)*el(2,j) uaf(2,j) = uaf(3,j) vaf(2,jm) = set
	NORTH	gae = dte*sqrt(grav*h(i,jmm1))/dy(i,jmm1) elf(i,jmm1) = gae*el(i,jmm2) + (1.-gae) *el(i,jmm1) elf(i,jm) = elf(i,jmm1) vaf(i,jm) = vaf(i,jmm1) uaf(i,jm) = set
	SOUTH	gae = dte*sqrt(grav*h(i,2))/dy(i,2) elf(i,2) = gae*el(i,3) + (1.-gae)*el(i,2) vaf(i,2) = vaf(i,3) uaf(i,1) = set
Cyclic (A-6)	EAST (i=im)	elf(im,j) = elf(3,j) uaf(im,j) = uaf(3,j) vaf(im,j) = vaf(3,j)
	WEST (i=1)	elf(1,j) = elf(imm2,j) elf(2,j) = elf(imm1,j) uaf(2,j)=uaf(imm1,j) vaf(2,j)=vaf(imm1,j)
	NORTH (j=jm)	elf(i,jm) = elf(i,3) uaf(i,jm) = uaf(i,3) vaf(i,jm) = vaf(i,3)
	SOUTH (j=1)	elf(i,1) = elf(i,jmm2) elf(i,2) = elf(i,jmm1) uaf(i,2)=uaf(i,jmm1) vaf(i,2)=vaf(i,jmm1)

The finite difference expression one gets for the EAST version of (B - 2) is

$$U_{im}^{n+1} = \gamma U_{im-1}^{n} + (1-\gamma) U_{im}^{n}; \quad \gamma \equiv c_i \Delta t_i / \Delta x$$

where one might like c_i to be the gravest mode, baroclinic phase speed. However, it is assumed that: *a)* the user has found and is using a Δt_i such that the maximum value of γ is near unity, corresponding approximately to the maximum depth and *b)* that c_i is proportional to $\sqrt{H}$. This is a seemingly crude approximation, but may perform fairly well; it at least guarantees that $0 < \gamma \leq 1$.

TABLE B: A list of internal mode variables to be set on open lateral boundaries and example boundary conditions. Note that UF and VF are used for the forward time step of U and V, T and S, and Q2 and Q2L. The variables TBE, TBW, TBN, TBS (and similar variables for salinity) are supplied by the user

Formula	Boundary	Code
Inflow condition: $U = BC$ (B-1)	EAST	uf(im,j,k) = bc(j,k) vf(im,j,k) = set
	WEST	uf(2,j,k) = bc(j,k) vf(1,j,k) = set
	NORTH	vf(i,jm,k) = bc(i,k) uf(i,jm,k) = set
	SOUTH	vf(i,2,k) = bc(i,k) uf(i,1,k) = set
Radiation: $\frac{\partial U}{\partial t} \pm c_i \frac{\partial U}{\partial x} = 0$ (B-2)	EAST	gai = sqrt(h(im,j)/hmax) uf(im,j,k) = gai*u(imm1,j,k) + (1.-gai)*u(im,j,k) vf(im,j,k) = set
	WEST	gai = sqrt(h(2,j)/hmax) uf(2,j,k) = gai*u(3,j,k) + (1.-gai)*u(2,j,k) vf(1,j,k) = set
	NORTH	gai = sqrt(h(i,jm)/hmax) vf(i,jm,k) = gai*v(i,jmm1,k) + (1.-gai)*v(i,jm,k) uf(i,jm,k) = set
	SOUTH	gai = sqrt(h(i,2)/hmax) vf(i,2,k) = gai*v(i,3,k) + (1.-gai)*v(i,2,k) uf(i,1,k) = set

Upstream advection on T or S: $\frac{\partial T}{\partial t} + U\frac{\partial T}{\partial x} = 0$ (B-3)	EAST	uf(im,j,k) = t(im,j,k) -dti/(dx(im,j)+dx(imm1,j)) * ((u(im,j,k) + abs(u(im,j,k))) * (t(im,j,k)-t(imm1,j,k)) + (u(im,j,k) - abs(u(im,j,k))) * (tbe(j,k)-t(im,j,k)))
	WEST	uf(1,j,k) = t(1,j,k) -dti/(dx(1,j)+dx(2,j)) * ((u(1,j,k) + abs(u(1,j,k))) * (t(1,j,k)-tbw(j,k)) + (u(1,j,k) - abs(u(1,j,k))) * (t(2,j,k)-t(1,j,k)))
	NORTH	uf(i,jm,k) = t(i,jm,k) -dti/(dy(i,jm)+dy(i,jmm1)) * ((v(i,jm,k) + abs(v(i,jm,k))) * (t(i,jm,k)-t(i,jmm1,k)) + (v(i,jm,k) - abs(v(i,jm,k))) * (tbn(i,k)-t(i,jm,k)))
	SOUTH	uf(i,1,k) = t(i,1,k) -dti/(dy(i,1)+dy(i,2)) * ((v(i,1,k) + abs(v(i,1,k))) * (t(i,1,k)-t(i,2,k)) + (v(i,1,k) - abs(v(i,1,k))) * (tbs(i,k)-t(i,1,k)))
Cyclic (B-4)		Much the same as (a - 6) except replace uaf with uf, etc. and t, s, q2 and q2l are handled similar to elf.

17. subroutine dens

The UNESCO equation of state, as adapted by Mellor(1991) is used. The *in situ* density is determined as a function of salinity, potential temperature and pressure; the latter is approximated by the hydrostatic relation and constant density. Initially, the values tbias and sbias are subtracted from temperature and salinity to reduce round-off error. With 32 bit arithmatic, a suggestion is tbias = 10. and sbias = 35. for open ocean models; with 64 bits, zero values are appropriate. In dens, these values are added again before the density is calculated. The actual density is normalized on 1025 kg/m^2. Since only gradients are needed (in subroutines baropg and profq), the value 1.025 is subtracted to reduce round-off error. APPENDIX A includes some discussion of thermodynamics.

18. subroutine slpmin

This subroutine examines the topography and adjusts h(i,j) so that the difference of the depths of any two adjacent cells divided by the sum of the depths is less than or equal to the parameter, slmin. In the process, volume is preserved. What generally happens is that the topography in deeper water is not changed whereas the shallower regions are altered depending on resolution.

19. utililty subroutines

There are a number of utility subroutines supplied with the program. For the most part they can be understood by reference to comments written into the code. All of the printing subroutines print out numbers in floating point or integer format. They accept a scale factor in the argument list which is either zero, in which case the code generates its own scale factor, or a finite value which is then used to scale the printed numbers. If the scale factor is negative, the output is floating point.

20. PROGRAM CURVIGRID

This program is relatively simple and may not perform as well as the more sophisticated program grid.f which is available on the POM web site.

The program is set up to accept values of longitude and latitude, here denoted by *x* and *y* to define the four edges of the gridded domain. This can be altered to accommodate rectilinear coordinates by setting the cosine of the latitude, CS = 1, in subroutine ORTHOG or by expunging the variable completely.

The border of the domain is determined by NB, NR or NL points on the j = 1, i = 1 and j = jm borders respectively. In this version of the program, DATA statements

contain this information. Cubic splines are then used to fill in the missing border coordinates.

The program is comprised of two steps:

I. The interior grid points ($1 < j < jm$) are filled such that the values at every i column is distributed proportionately to the y-values at i = 1; the interior x value are similarly distributed.

II. Subroutine ORTHOG is called to render the $x_{i,j}$ and $y_{i,j}$ an orthogonal coordinate system. Then, use is made of the orthogonality conditions

$$\left(\frac{\partial x}{\partial s}\right)_j = -\left(\frac{\partial y}{\partial s}\right)_i, \quad \left(\frac{\partial y}{\partial s}\right)_j = \left(\frac{\partial x}{\partial s}\right)_i \tag{44a,b}$$

or

$$\frac{\delta_j x}{\delta_j s} = -\frac{\delta_i y}{\delta_i s}, \quad \frac{\delta_j y}{\delta_j s} = \frac{\delta_i x}{\delta_i s} \tag{45a,b}$$

With reference to Fig. 6, (45a,b) are solved according to

$$x_{i,j} - x_{i,j-1} = \frac{\delta_j s}{\delta_i s}[y_{i+1,j} - y_{i-1,j} + y_{i+1,j-1} - y_{1,j-1}] \tag{46a}$$

$$y_{i,j} - y_{i,j-1} = \frac{\delta_j s}{\delta_i s}[x_{i+1,j} - x_{i-1,j} + x_{i+1,j-1} - x_{1,j-1}] \tag{46b}$$

where

$$\delta_i s = \frac{1}{4}[(x_{i+i,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i1,j})^2]^{1/2} + \frac{1}{4}[(_{i+1,j-1} - x_{i-1,j-1})^2 + (y_{i+1,j-1} - y_{i-1,j-1})^2]^{1/2} \tag{47a}$$

$$\delta_j s = [(x_{i,j} - x_{i,j-1})^2 + (y_{i,j} - y_{i,j-1})^2]^{1/2} \tag{47b}$$

The factor, CS, the cosine of the latitude, is not included in (46a,b) and (47a,b) but is included in the corresponding code in ORTHOG. Now, the above equations are interated many times during which $\delta_j s$ is fixed; i.e., $\delta_j s$, $x_{i,1}$ and $y_{i,1}$ are data of the initial field specified in step I which are retained. In the course of iteration, $\delta_j s$, $x_{i,j}$ and $y_{i,j}$ are reevaluated. The shape of the original domain does change but not greatly. During this iteration, CS is held fixed. In fact, CS changes very little so that ORTHOG is called only twice to converge on this factor.

It should be noted that, if the border points contain too much curvature, then the curves normal to the i = constant curves can focus to a point at some j row after which the calculation is nonsense. Some trial and error is therefore required. A way to avoid this is to call POISSON after step I which solves for $y_{i,j}$ according to $\partial^2 y / \partial^2 i + \partial^2 y / \partial^2 j = 0$. This avoids the focusing problem but may not yield the most desirable grid.

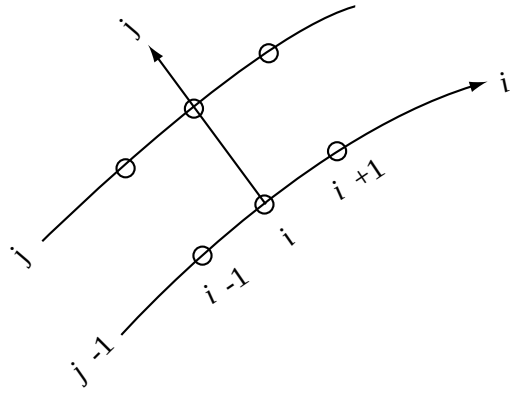


Figure 6. The orthogonal curvilinear grid system.

A good practice is to map the bottom topography on to the grid, then calculate the CFL limiting time step for each grid point; one wishes, of course, to avoid overly small steps.

An alternate and more sophisticated grid generating code can be found in grid.f and sepelli.f in the directory, contrib_code, in our ftp site. However, there is as yet no documentation for grid.f.

APPENDIX A: Note on the Equation of State, Potential Temperature and Static Stability

Two equations of state for density are

$$\rho = \rho_1(T, S, p) \tag{A1}$$

$$\rho = \rho_2(\Theta, S, p) \tag{A2}$$

where T is *in situ* temperature and Θ is the potential temperature. In the model, (A2) is used. To relate potential temperature, Θ, to *in situ* temperature, T, recall the thermodynamic relation for entropy.

$$Td\eta = dh - \frac{dp}{\rho} - \mu dS \tag{A3}$$

where η is the entropy, h, enthalpy and μ the chemical potential for salt taken here as a single average constituent. Furthermore,

$$dh = C_p dT + (1 - \alpha T)\frac{dp}{\rho} \tag{A4}$$

where we have set

$$\left(\frac{\partial h}{\partial T}\right)_{p,S} \equiv C_p \ , \quad \left(\frac{\partial h}{\partial p}\right)_{T,S} = \frac{(1-\alpha T)}{\rho} \tag{A5a, b}$$

and where the coefficient of thermal expansion is

$$\alpha \equiv - \frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_p \tag{A5c}$$

We note that (A5b) has been obtained from (A3) and one of Maxwell's relations. Combining (A3) and (A4), we have

$$d\eta = C_p \frac{dT}{T} - \alpha \frac{dp}{\rho} - \frac{\mu dS}{T} \tag{A6}$$

The definition of potential temperature in oceanography* is

$$C_{po} \frac{d\Theta}{\Theta} \equiv C_p \frac{dT}{T} - \alpha \frac{dp}{\rho} \tag{A7}$$

where $C_p = C_p\,(T,S,p)$ and $C_{po} = C_p\,(T,S,0)$. Combining (A6) and (A7),

---

* as contrasted to meteorology, where, for a perfect gas, we have $\alpha T = 1$ and $p = \rho RT$. Potential temperature is then defined as $d\Theta/\Theta = d\eta/C_p = dT/T - (R/C_p)dp/p$ which can be integrated exactly to give $\Theta = T(p_o/p)^{R/C_p}$; p_o is a reference pressure where $\Theta = T$.

$$d\eta = C_{po}\frac{d\Theta}{\Theta} - \frac{\mu\, dS}{T} \tag{A8}$$

For processes where heat transfer, viscous dissipation and salt diffusion are null, $D\Theta/Dt = DS/DT = 0$; then, from (A8), $D\Theta/Dt = 0$; i.e. the process is isentropic. An integral relation obtained from (A7) is

$$T(z) - \Theta = \int_p^o \frac{\alpha' T'}{C_p'}\frac{dp'}{\rho'} = -\int_z^o \frac{\alpha' T' g}{C_p'} dz' \tag{A9}$$

The hydrostatic pressure relation is used to obtain the second expression on the right of the equal sign. In (A9), $\Theta(z) = \Theta(0) = T(0)$. For $T = 10^oC$, $S = 35$ psu and $p = 0$, one finds (Gill, 1982, p.603) that $\alpha Tg / C_p \cong 0.12K / 1000m$. Equation (A-9) allows one to initialize potential temperature in the model given *in situ* properties. An algorithm to do this is provided by Bryden (1973).

Static Stability

To conveniently provide further background information and also inquire into an aspect of the Boussinesq approximation, we review the following equations for two-dimensional isentropic flow (see also Mellor and Ezer, 1995, for evaluation of a non-Boussinesq version of POM).

$$\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{w}}{\partial z} = 0 \tag{A10}$$

$$\tilde{\rho}\left(\frac{\partial \tilde{u}}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \tilde{u}\right) = -\frac{\partial \tilde{p}}{\partial x} \tag{A11}$$

$$\tilde{\rho}\left(\frac{\partial \tilde{w}}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \tilde{w}\right) = -\frac{\partial \tilde{p}}{\partial z} - \tilde{\rho} g \tag{A12}$$

where we have made the Boussinesq approximation in (A10) but have not done so in (A11) and (A12). Equation (A10) is justified by examination of the full equation $\nabla\cdot\mathbf{u} + \tilde{\rho}^{-1} D\tilde{\rho} / Dt = 0$. The first term scales like u_o/L whereas the second scales as $(u_o/L)\delta\tilde{\rho} / \tilde{\rho}$ Since $\delta\tilde{\rho} / \tilde{\rho} \lesssim .05$ in the ocean, the second term can be neglected.

Let mean quantities be denoted by upper case letters and fluctuating quantities by lower case letters; the exception to this is density where ρ and ρ' are the mean and fluctuating values. For this analysis the mean velocity will be zero. Therefore we have $(\tilde{u}, \tilde{w}) = (u, w)$, $\tilde{p} = P + p$, $\tilde{\rho} = \rho + \rho'$, $\partial P / \partial z = -\rho g$ and $\rho = \rho(z)$ so that, for small perturbations,

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \tag{A13}$$

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial p}{\partial x} \tag{A14}$$

$$\rho \frac{\partial w}{\partial t} = - \frac{\partial p}{\partial z} - \rho' g \tag{A15}$$

Now for isentropic flows the equations of state yields $D\tilde{\rho} / Dt = c^{-2} D\tilde{p} / Dt$ where $c^2 \equiv (\partial \tilde{p} / \partial \tilde{\rho})_{\Theta,S}$ is the speed of sound squared. The corresponding density perturbation equation is

$$\frac{\partial \rho'}{\partial t} + w \frac{\partial \rho}{\partial z} = \frac{1}{c^2} \left(w \frac{\partial p}{\partial z} + \frac{\partial p}{\partial t} \right)$$

or

$$\frac{\partial \rho'}{\partial t} - w \frac{\rho N^2}{g} = \frac{1}{c^2} \frac{\partial p}{\partial t} \tag{A16}$$

where

$$\rho \frac{N^2}{g} \equiv - \frac{\partial \rho}{\partial z} + \frac{1}{c^2} \frac{\partial p}{\partial z} = - \frac{\partial \rho}{\partial z} - \frac{\rho g}{c^2} \tag{A17}$$

N^2 is the Brunt-Vassala frequency squared or the static stability. If one eliminates u, p and ρ' from (A13) to (A16), the resulting equation for w is

$$\frac{\partial^2}{\partial t^2} \left[\frac{\partial^2 w}{\partial z^2} + \frac{\partial^2 w}{\partial x^2} + \frac{N^2}{g} \frac{\partial w}{\partial z} \right] + N^2 \frac{\partial^2 w}{\partial x^2} = 0 \tag{A18}$$

The last term in the square brackets can be neglected compared with the first. To check this, let $g^{-1} N^2 w_z / w_{zz} \simeq g^{-1} N^2 L_z$ where L_z is the vertical scale height. $g^{-1} N^2 L_z$ has two parts as shown in (A17). If we take $L_z \approx 1000$m, then the first part, $\rho^{-1} \rho_z L_z \approx -.010$ and the second part, $c^{-2} g L_z \approx .005$. Tracing back through the original equations, we find that this approximation is equivalent to setting ρ = constant = ρ_o in (A14) and (A15) and neglecting the right side of (A16).

A solution to (A18) for N^2 = constant is $w \propto \exp[i(lz + kx - \sigma t)]$ where the dispersion relation is $\sigma^2 = N^2 k^2 / (l^2 + k^2)$. If $N^2 < 0$, the flow is unstable; if $N^2 > 0$, the flow is stable. Thus, N^2, given by (A17) is the correct static stability parameter for

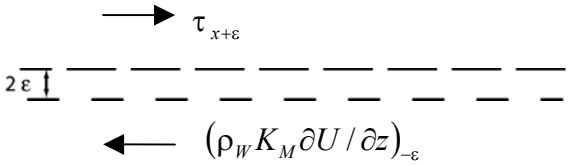
use in the turbulence closure model which are constructed from perturbation equations like (A16) together with other equations and terms.

APPENDIX B: Flux Balances across the Air/Sea Interface

The Princeton Ocean Model (POM) accepts six flux terms as surface boundary conditions; they are vflux, wusurf, wvsurf, wtsurf, wssurf and swrad. This note provides some detail as to how they may calculated. It presumes that there exist data sources that provide wind velocity, air temperature and humidity, precipitation rate and fractional cloud cover. Alternately, direct observation of ocean surface temperatures from satellites can be used obviating the need for air temperature and cloud cover. Other strategies might entail partial or complete use of climatological data. Thus, each application probably differs one from another and each user wishing to create simulations of real water bodies will need to create their own tailor made subroutine to ingest data at the available time intervals and output interpolated POM surface boundary conditions at dti intervals.

Consider an indefinitely thin interface of thickness, 2ε . The interface has no mass since we can let $\varepsilon \to 0$. Nevertheless, important processes can occur within the interface; for example, liquid salt water can enter the interface on the water side ($-\varepsilon$) and emerge as water vapor, transporting zero salt, on the air side ($+\varepsilon$) .

Momentum Balance. A schematic of the momentum balance is



so that, generalizing to two dimensions,

$$\rho_W K_M \left(\partial U/\partial z, \partial V/\partial z\right)_{-\varepsilon} = (\tau_x, \tau_y)_{+\varepsilon} \tag{B1}$$

where a bulk atmospheric relation is

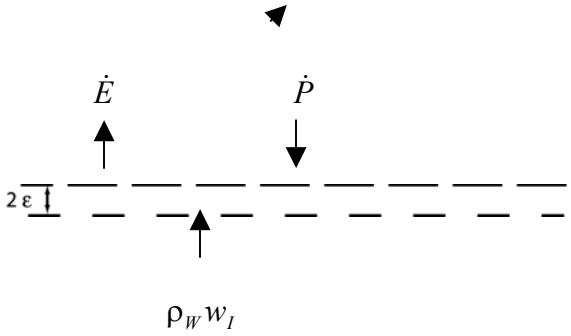
$$(\tau_x, \tau_y)_{+\varepsilon} = \rho_A C_D \left|\mathbf{U}_{10} - \mathbf{U}_{\mathbf{w}}\right| (U_{10} - U_w, V_{10} - V_w) \tag{B2}$$

C_D is the drag coefficient which is derived from the law of the wall in Appendix C along with the other coefficients for heat and water vapor discussed below; ρ_A is the density of air and (U_{10}, V_{10}) are wind velocity component measured at 10 m above the sea surface and (U_w, V_w) are the surface water and air

velocity components; generally the later are small and can be neglected (but consider the Gulf Stream whose surface velocities are about 2 m s^{-1}). The scalar factor, $|\mathbf{U}_{10} - \mathbf{U}_{\mathbf{w}}|$, is the velocity modulus or wind speed

. A surface boundary condition in POM is (wusurf, wvsurf) $\equiv -K_M(\partial U/\partial z, \partial V/\partial z)_{-\varepsilon}$ which is obtained from (B1) after division by $\rho_W = 1025\ \mathrm{kg\,m^{-3}}$, the density of the surface seawater.

Mass Balance A mass balance schematic is



so that the balance is simply

$$\rho_W w_I = \dot{E} - \dot{P} \tag{B3}$$

where

w_I = the vertical advection velocity at the sea surface

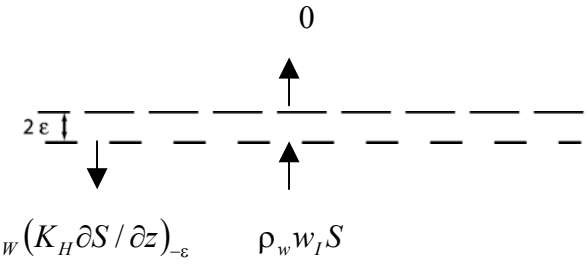
$$\dot{E} = \text{Evaporation flux} = \rho_A C_E |\mathbf{U}_{10}\text{-}\mathbf{U}_{\mathbf{w}}|(q_W\text{-}q_{10}) \tag{B4}$$

$\dot{P}$ = Precipitation flux

q_W is the specific humidity (density of water vapor/density of moist air) taken to be the saturation humidity evaluated at the sea surface temperature, and q_{10} is the air humidity at the 10 m height. The specific humidity may be related to the partial pressure of the water vapor, e, according to $q = (e/p)\left(1-(1-\varepsilon)e/p\right)^{-1}$ where $\varepsilon = 0.6220$ is the ratio of the molecular mass of water to dry air and p is the atmospheric pressure. Thus, $q_w = (e_w/p)\left(1-(1-\varepsilon)e_w/p\right)^{-1}$ where the saturation vapor pressure is given by $\log_{10} e_w = (0.7859 + 0.03477\,t)(1 + 0.00412\,t)^{-1}$; here the units of p and e_w are mb.

POM needs to know vflux = w_I, the vertical velocity component of water exiting the water column surface; it is obtained from (B3) after division by ρ_w.

Salt Balance. A schematic of the salt balance is



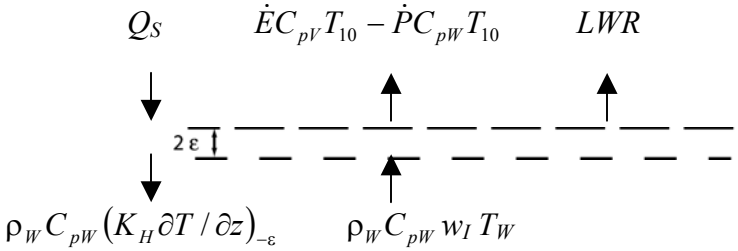
since the salinities of the precipitation, $\dot{P}$, and water vapor flux, $\dot{E}$, are assumed to be nil. Thus

$$-\rho_W (K_H \partial S/\partial z)_{-\varepsilon} + \rho_w w_I S_W = 0 \qquad \text{(B5)}$$

$S_W = S_{-\varepsilon}$ is the salinity at the sea surface.

For surface boundary conditions, POM requires the diffusive flux at the surface, wssurf = $-(K_H \partial S/\partial z)_{-\varepsilon}$ (the sign is that of turbulence fluxes, $\overline{w's'}$) which is readily obtained from (B5) after division by ρ_W.

Heat Balance. The thermal heat balance is more complicated. A schematic is



so that the balance may be written

$$\left(Q_S - (\dot{E}C_{pV} - \dot{P}C_{pW})T_{10} - LWR\right)_{+\varepsilon} - \left(\rho_W C_{pW}(K_H \partial T/\partial z)_{-\varepsilon} - \rho_W C_{pW} w_I T_W\right)_{-\varepsilon} = \dot{E}L \qquad \text{(B6)}$$

where

$$L = \text{Latent heat of evaporation} = (2.501\text{-}0.002T_W/^{o}C)\times 10^6 \quad \text{J kg}^{-1} \qquad \text{(B7)}$$

$$Q_S = \text{Sensible Heat} = \rho_A C_{pA} C_H |\mathbf{U}_{10}\text{-}\mathbf{U}_{\mathbf{w}}|(T_{10}\text{-}T_W) \qquad \text{(B8)}$$

LWR = net Long Wave Radiation

ρ_A, C_{pA}, T_{10} are the air density, specific heat and temperature of the air at 10 m above the water surface and ρ_W, C_{pW}, T_W are the density, specific heat and temperature at the water surface respectively. C_{pV} is the specific heat of water vapor and is about half that of water. Equation (6) may be manipulated so that

$$\rho_W C_{pW}\left(K_H \partial T/\partial z\right) = \dot{E}L - Q_S + LWR + (\dot{E}-\dot{P})C_{pW}(T_A - T_W) \tag{B9}$$

where we have deleted the subscripts, $-\varepsilon$ and $+\varepsilon$. The simplifying difference between (B6) and (B9) is $\dot{E}(C_{pV}-C_{pW})T_A \cong \dot{E}(C_{pV}-C_{pW})T_W$, but the excluded factor $(C_{pV}-C_{pW})T_W$ is small and can be absorbed into the definition of $L(T_W)$. Thus, $(C_{pV}-C_{pW})T_W = (789-3986)T_W \text{ J (kg}^{\circ}\text{C)}^{-1} = -0.003\times10^6 T_W \text{J (kg}^{\circ}\text{C)}^{-1}$ so that we merely need to change the coefficient 0.002 to 0.005 in (B7).

For surface boundary conditions, POM requires the diffusive flux at the surface, wtsurf $\equiv -\left(K\partial T/\partial z\right)_{-\varepsilon}$ readily obtained from (B9) after division by $\rho_W C_{pW}$ where $C_{pW} = 3986 \text{ J kg}^{-1}$.

Long Wave Radiation. Long Wave Radiation is (nearly) black body radiation emitted from the surface of land or sea or radiation emitted from the atmosphere after absorption of solar short wave radiation. In the absence of direct input from an atmospheric model or from climatology, there are many empirical prescriptions in the literature to estimate the long wave radiation at the sea surface. The following are taken from Price et al. (1978), Gill (1982) and Rosati and Miyakoda (1984) who reference numerous sources. Thus,

$$LWR = 0.98\sigma(T_W+273)^4(0.39-0.05e_{10}^{1/2})(1-0.8n) - [(T_w+273)^4 - (T_{10}+273)^4] \tag{B10}$$

where the factor, 0.98, is the emissivity; $\sigma = 5.7\times10^{-8} \text{ W m}^{-2}\text{K}^{-4}$ is the Stefan-Boltzman constant; e_{10} is the vapor pressure (units are mb) and n ($0 \le n \le 1$) is the fractional cloud cover.

Short Wave Radiation. Direct solar, short wave radiation is absent from the balance in (6) or (9) since it is either reflected or passes through our thin control volume to be absorbed in the underlying water column. In POM, this occurs in subroutine proft.

The short wave radiation passing into the water column at the sea surface is

$$SRW = SC(\sin\alpha\,\delta^{\csc\alpha} + 0.91 - \delta^{\csc\alpha})(1-0.71n)(1-\beta) \tag{B11}$$

$SC = 1370\,\mathrm{Wm}^{-2}$ is the solar constant (solar radiation at the top of the atmosphere); $\delta = 0.85$ is the atmospheric transmission coefficient; the sun's altitude, α, is a complicated geometrical expression given by

$$\sin\alpha = \cos\theta\,\cos lat\cos\phi + \sin\theta\,\sin lat$$

The sun's declination, $\theta = 23(2\pi/360)\cos(2\pi(357-t)/365)$; the sun's hour angle, $\phi = 2\pi(t+0.5)$, where t is the Julian date from 0000 local time, 1 January. The albedo, β, is taken from Fig.24 of Jerlov (1968).

In POM, $\mathrm{swrad} = -SWR$ from (10) after division by $\rho_W C_{pW}$. Absorption of the short wave radiation into the water column is built into subroutine proft.

APPENDIX C: Atmospheric Bulk Coefficients

This appendix reviews the derivation of the bulk coefficients. It is believed to be simpler and as comprehensive as other versions which have appeared in the literature. In this appendix and in the enabling code included at the end, (τ_{ox},τ_{oy}) is the surface stress divided by ρ_w, $\dot{E}$ is the evaporation flux divided by ρ_w and Q is the sensible heat flux divided by $\rho_w C_{pW}$

ANALYSIS

The law of the wall may be written for the velocity components

$$\frac{u_*(U-U_W)}{\tau_{ox}} = \frac{1}{\kappa}\left[\ln\frac{z}{z_0} + \Psi(z/L)\right] = \frac{1}{C_U} \qquad \text{(C1a)}$$

$$\frac{u_*(V-V_W)}{\tau_{oy}} = \frac{1}{\kappa}\left[\ln\frac{z}{z_0} + \Psi(z/L)\right] = \frac{1}{C_U} \qquad \text{(C1b)}$$

In the above, u_* is the friction velocity. (U, V) are the wind velocity components and (τ_{ox},τ_{oy}) are the surface stress components. Equations (C1a, b) apply to any z, small with respect to the overall boundary layer height (~1000 – 3000 m) and large compared to the viscous or roughness sub-layer.

For temperature and specific humidity

$$\frac{u_*(T - T_W)}{Q} = \frac{P_{rt}}{\kappa}\left[\ln\frac{z}{z_0} + \Psi(z/L)\right] + F_{YK}\left(\frac{z_0 u_*}{\nu}, P_r\right) = \frac{1}{C_T} \tag{C2}$$

$$\frac{u_*(q - q_W)}{\dot{E}} = \frac{P_{rt}}{\kappa}\left[\ln\frac{z}{z_0} + \Psi(z/L)\right] + F_{YK}\left(\frac{z_0 u_*}{\nu}, S_c\right) = \frac{1}{C_E} \tag{C3}$$

The potential temperature is T and Q is the heat flux. The specific humidity is q and $\dot{E}$ is the moisture flux. All fluxes are negative values relative to turbulence quantities; for example, $\tau_{ox} = -\left(\overline{w'u'}\right)_o$.

The roughness parameter is given by

$$z_0 = \max\left(0.14\frac{\nu}{u_*}, \alpha_{CH}\frac{u_*^2}{g}\right) \tag{C4}$$

The first part is Charnock's relation for the roughness effect of waves whereas the second part applies to smooth surfaces and $\nu = 15\text{x}10^{-6}$ m^2 s^{-1} is the kinematic viscosity of air. The factor, F_{YK}, has been determined by Yaglom and Kader (1974) in laboratory experiments to be

$$F_{YK} = 3.14\left(\frac{u_* z_0}{\nu}\right)^{1/2}(P_r^{2/3} - 0.2) + 2.11 \tag{C5}$$

The molecular Prandtl number (kinematic viscosity/heat diffusivity) for air is $Pr = 0.72$ whereas the Schmidt number for water vapor (kinematic viscosity/water vapor diffusivity) is $S_c = 0.60$, but we approximate $S_c \cong P_r$ in equation (A-3) and, therefore $C_E \cong C_T$

The stability function, Ψ (z/L), is taken to be the same for momentum and temperature. This is a simplifying approximation that is expected to introduce negligible error; introduction of a *turbulence* Prandtl number, $P_{rt} = 0.9$, will compensate somewhat for the simplification. Ψ is derived from the Monin-Obukhov similarity function, ϕ (z/L), according to

$$\Psi = \int_0 [\phi(z/L) - 1]\frac{dz}{z}$$

For stable flow, $\phi \cong 1 + 5z/L$ and

$$\Psi = 5\frac{z}{L};\ L > 0 \tag{C6a}$$

For unstable flow, $\phi \cong (1 - 10z/L)^{-1/3}$ and a complicated expression is obtained such that

$$\Psi = -\frac{3}{2}\ln\frac{x^2 + x + 1}{3} + \sqrt{3}\left\{\tan^{-1}\frac{2x+1}{\sqrt{3}} - \tan^{-1}\sqrt{3}\right\}, x \equiv \left(1 - 10\frac{z}{L}\right)^{1/3};\ L < 0 \tag{C6b}$$

In the above, the Monin-Obukhov length is obtained according to

$$L \equiv \frac{u_*^3}{\kappa g\left(\beta_T Q + \beta_q \dot{E}\right)_o} \tag{C7}$$

where $\beta_T = 1/T$ (degrees Kelvin) and $\beta_q = 0.608$.

The above equations form the basis of a subroutine, reproduced below, to calculate the fluxes from the properties at the 10 m height (or, with a slight change in the program, at any height). The functions (A-6a, b) are represented by a look-up table in the code.

DISCUSSION

At this stage we specialize to z = 10 m and U, V, T and q are to be evaluated at the 10 m height. Data is usually presented in the form of "bulk" coefficients such that

$$(\tau_{ox,}\tau_{ox}) = C_D \mid \mathbf{U}_{10} - \mathbf{U}_W \mid (U_{10} - U_W, V_{10} - V_W) \tag{C8a, b}$$

$$Q = C_H \mid \mathbf{U}_{10} - \mathbf{U}_W \mid (T_{10} - T_W) \tag{C9a}$$

$$\dot{E} = C_H \mid \mathbf{U}_{10} - \mathbf{U}_W \mid (q_{10} - q_W) \tag{C9b}$$

It can then be shown that

$$C_D = C_U^2 \tag{C10}$$

$$C_H = C_E = C_U C_T \tag{C11}$$

The portion of the above formulation pertaining to the static stability depends on the *Monin-Obukhov* relations determined from measurements over land. This aspect is generally considered to be reliable. Thus we discuss the neutral drag and heat flux coefficients, C_{DN} and C_{HN} and let $\Psi = 0$. Oftentimes, C_{DN} and C_{HN} are taken as simple constants, independent of wind speed. In particular, we will see that data and our formulation yield nearly constant values of C_{HN} and one could easily justify using a constant value independent of wind speed. However, one might also avoid understanding *why* C_{HN} is nearly constant (it is so for air but not for other fluids) and *why* it differs from C_{DN}.

In Figure 1 we compare the present formulation with the neutral, data based, drag coefficients (dashed line) of Garratt (1977). It is seen that $\alpha_{CH} = 0.0144$ very nearly reproduces the straight line $C_{DN}(U)$ by Garratt except for low wind speeds. (However, Large and Pond 1982 present C_{DN} results that are somewhat lower than those of Garratt whereas Smith et al. 1992 show higher values.) Presumably these results apply to fully developed waves. More recent considerations (e.g., Donelan et al. 1995) indicate that rising or falling seas result in increased C_{DN}. Therefore, we have included a curve for $\alpha_{CH} = 0.020$, but, clearly, uncertainty remains.

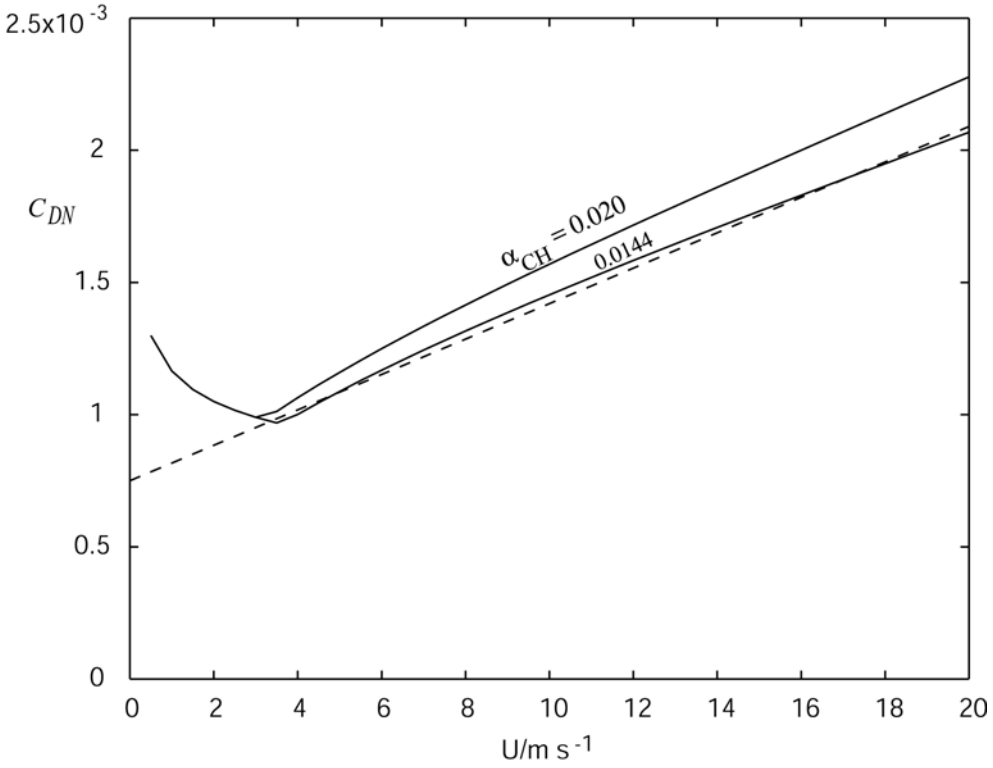


Figure 1. C_{DN} as a function of wind speed. The dashed curve is the neutral drag coefficient as recommended by Garratt (1977). The solid lines are from equations (C1a, b), (C4) and (C10).

We note that, except for the second term on the right side of (C4), operative only at low wind speeds when the wave roughness is minimal, the momentum transfer is virtually independent of molecular viscosity. At the air-sea interface, momentum is transferred via pressure drag or form drag. On the other hand, there are no pressure terms in the transport equations for temperature or humidity. Therefore, at the air-sea interface, molecular diffusivities for heat and humidity must be involved; for air, the two diffusivities are nearly the same and no distinction is made here.

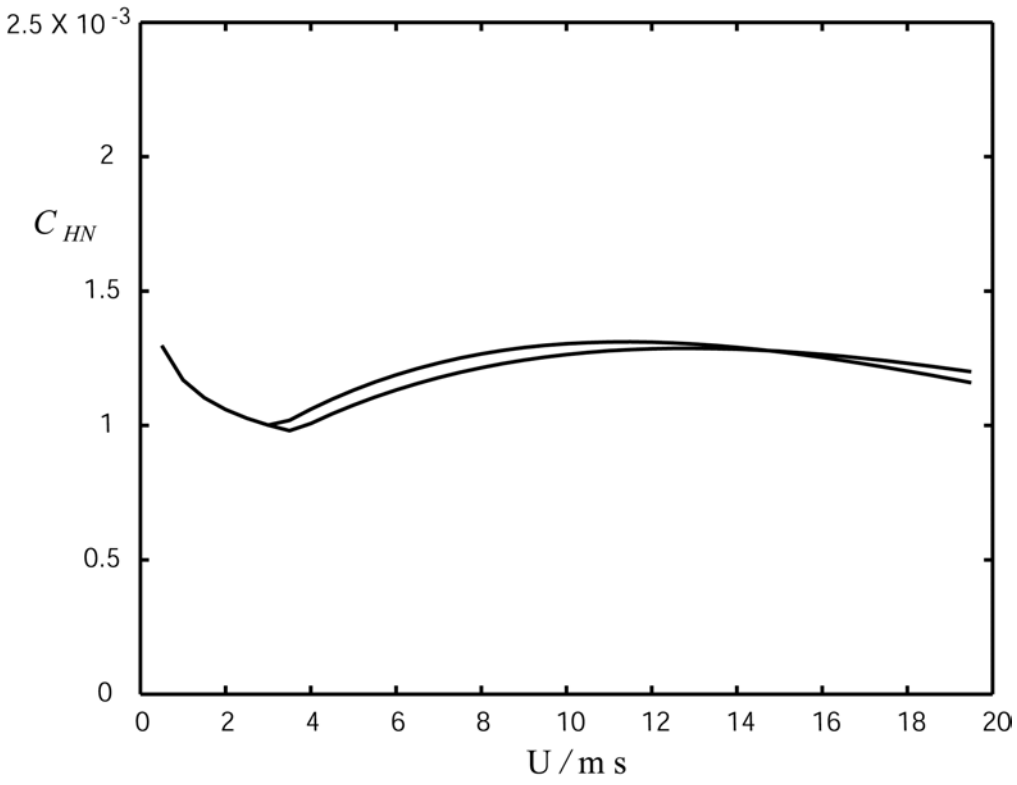


Figure 2. C_{HN} as a function of wind speed. The two curves for differentα_{CH} are virtually indistinquishable.

The formulas used here, equations (C2), (C4) and (C5) are plotted in Figure 2 as solid curves. In fact, $C_{HN} \cong C_{EN}$ is often taken (e.g., Large and Pond 1982) to be a constant $\cong 0.0012$.

A subroutine to calculate ABL surface fluxes. This example for $\alpha_{CH} = 0.0144$

```
      PROGRAM MAIN
C  This MAIN program is to demonstrate the workings of SUBROUTINE FLUX; only the latter
C  would be included in a user created subroutine to supply surface forcing data.
      DO N=1,3
      write(6,'(/,"       DU       DV       DT       DQ     TAUX
     &   TAUY      SHF      SVF       CD       CH     ")')
      DT=-20+10.*FLOAT(N)
      do i=1,26,5
      DU=FLOAT(i-1)
      DV=0.
      DQ=0.
      CALL FLUX(CU,CT,DU,DV,DT,DQ,TAUX,TAUY,SHF,SVF)
      write(6,'(1x,8F10.4)') DU,DV,DT,DQ,TAUX,TAUY,SHF,SVF
      CD=CU*CU
      CH=CU*CT
      write(6,'(1x,8(10x),2F10.4)') CD,CH
      enddo
      ENDDO
      stop
      end

      SUBROUTINE FLUX(CU,CT,DU,DV,DT,DQ,TAUX,TAUY,SHF,SVF)
C
C     Subroutine to calculate ABL surface fluxes.
C
C     CU, CT = inverse law of the wall variables.
C       (Routine should benefit from a reasonable first guess)
C     DU, DV = velocity components relative to surface
C     TAUX,TAUY = momentum fluxes/air density
C     DT, DQ = temperature and specific humidity relative to surface
C     SHF, SVF = Sensible Heat/(air density*specific heat)
C          and Vapor flux/air density.
C     ITER = Number of iterations.
C
      REAL kappa,LINV
      DATA gee/9.807/
      DATA kappa/0.41/,BETAT/.00357/,BETAQ/0.608/
      DATA VISC/15.E-6/
      DATA PRT/0.9/
      ALPHA=0.0144
      ITER=4
C  Begin interation
      CU=.03
      CT=.03
      DO I=1,ITER
      UST=ABS(CU)*SQRT(DU**2+DV**2+.0001)
      SHF=CT*UST*DT
      SVF=CT*UST*DQ
      LINV=kappa*gee*(BETAT*SHF+BETAQ*SVF)/UST**3
      ZETA=10.*LINV
      CALL ZETAPSI(ZETA,PSI)
      Z0=MAX(ALPHA *UST**2/gee,0.14*VISC/UST)
      YKF=3.14*sqrt(ust*z0/VISC)*.60+2.11
      CU=kappa/(log(10/Z0)+PSI)
      CU=MAX(CU,0.)
```

```
      CT=kappa/(log(10/Z0)+kappa*YKF/PRT+PSI)/PRT
      CT=MAX(CT,0.)
C     write(6,'(1x,8F10.4)')
C    1    DU,DV,DT,DQ,ZETA,PSI,CU,CT
      ENDDO
C  END iteration
      UST=CU*SQRT(DU**2+DY**2)
      TAUX=CU*UST*DU
      TAUY=CU*UST*DV
      SHF=CT*UST*DT
      SVF=CT*UST*DQ
      Return
     end
c
      SUBROUTINE ZETAPSI( ZETA,PSI)
      DIMENSION PS(41)
      DATA PS/
     1 -2.0095,  -1.9915,  -1.9732,  -1.9544,  -1.9352,  -1.9155,
     2 -1.8953,  -1.8747,  -1.8534,  -1.8316,  -1.8092,  -1.7862,
     3 -1.7624,  -1.7380,  -1.7127,  -1.6867,  -1.6597,  -1.6318,
     4 -1.6029,  -1.5728,  -1.5416,  -1.5091,  -1.4751,  -1.4395,
     5 -1.4023,  -1.3631,  -1.3218,  -1.2781,  -1.2316,  -1.1821,
     6 -1.1291,  -1.0718,  -1.0097,  -0.9416,  -0.8663,  -0.7818,
     7 -0.6853,  -0.5723,  -0.4352,  -0.2583,   0.0000/
      I=1+(ZETA+4.)/.1
      IF(I.LT.1) PSI=-2.0095
      IF(I.GE.1.and.I.LT.41)
     1      PSI=PS(I)+(PS(I+1)-PS(I))*(ZETA+4.-.1*(I-1))/.1
      IF(I.GE.41) PSI=5.*ZETA
      RETURN
      END
```

Program output

```
 Unstable
     DU         DV         DT         DQ       TAUX      TAUY       SHF        SVF        CD         CH
  0.0000     0.0000   -10.0000     0.0000     0.0000    0.0000    -0.0837    0.0000
                                                                                        0.0046     0.0042
  5.0000     0.0000   -10.0000     0.0000     0.0388    0.0000    -0.0747    0.0000
                                                                                        0.0014     0.0014
 10.0000     0.0000   -10.0000     0.0000     0.1697    0.0000    -0.1433    0.0000
                                                                                        0.0017     0.0014
 15.0000     0.0000   -10.0000     0.0000     0.4305    0.0000    -0.2021    0.0000
                                                                                        0.0019     0.0013
 20.0000     0.0000   -10.0000     0.0000     0.8585    0.0000    -0.2437    0.0000
                                                                                        0.0021     0.0012
 25.0000     0.0000   -10.0000     0.0000     1.5036    0.0000    -0.2692    0.0000
                                                                                        0.0024     0.0011

 Neutral
     DU         DV         DT         DQ       TAUX      TAUY       SHF        SVF        CD         CH
  0.0000     0.0000     0.0000     0.0000     0.0000    0.0000     0.0000    0.0000
                                                                                        0.0027     0.0026
  5.0000     0.0000     0.0000     0.0000     0.0293    0.0000     0.0000    0.0000
                                                                                        0.0011     0.0011
 10.0000     0.0000     0.0000     0.0000     0.1481    0.0000     0.0000    0.0000
                                                                                        0.0015     0.0013
 15.0000     0.0000     0.0000     0.0000     0.4012    0.0000     0.0000    0.0000
                                                                                        0.0018     0.0013
 20.0000     0.0000     0.0000     0.0000     0.8297    0.0000     0.0000    0.0000
                                                                                        0.0021     0.0012
 25.0000     0.0000     0.0000     0.0000     1.4765    0.0000     0.0000    0.0000
                                                                                        0.0024     0.0011

 Stable
     DU         DV         DT         DQ       TAUX      TAUY       SHF        SVF        CD         CH
```

0.0000	0.0000	10.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5.0000	0.0000	10.0000	0.0000	0.0031	0.0000	0.0066	0.0000	0.0001	0.0001
10.0000	0.0000	10.0000	0.0000	0.0952	0.0000	0.0898	0.0000	0.0009	0.0009
15.0000	0.0000	10.0000	0.0000	0.3420	0.0000	0.1754	0.0000	0.0015	0.0012
20.0000	0.0000	10.0000	0.0000	0.7715	0.0000	0.2331	0.0000	0.0019	0.0012
25.0000	0.0000	10.0000	0.0000	1.4218	0.0000	0.2658	0.0000	0.0023	0.0011

APPENDIX D: Derivation of the Sigma Equations

We treat only *x*, *z* dependencies; extension to include the *y* dependency is straight forward.

Let

$$\phi(x,z,t) = \phi^*(x^*,\sigma,t^*) \tag{D1}$$

and

$$x = x^*,\ t = t^*,\ \sigma = \frac{z-\eta(t)}{D}\ ;\ D = H(x)+\eta(x,t) \tag{D2a, b, c, d}$$

Note that when $z = \eta$, $\sigma = 0$ and when $z = -H$, $\sigma = -1$. The chain rule for ϕ is

$$\frac{\partial\phi}{\partial x} = \frac{\partial\phi^*}{\partial x^*} + \frac{\partial\phi^*}{\partial\sigma}\frac{\partial\sigma}{\partial x} = \frac{\partial\phi^*}{\partial x^*} - \frac{\partial\phi^*}{\partial\sigma}\frac{1}{D}(\eta_x + \sigma D_x) \tag{D3a}$$

$$\frac{\partial\phi}{\partial t} = \frac{\partial\phi^*}{\partial t^*} + \frac{\partial\phi^*}{\partial\sigma}\frac{\partial\sigma}{\partial t} = \frac{\partial\phi^*}{\partial t^*} - \frac{\partial\phi^*}{\partial\sigma}\frac{1}{D}(\eta_t + \sigma D_t) \tag{D3b}$$

$$\frac{\partial\phi}{\partial z} = \frac{\partial\phi^*}{\partial\sigma}\frac{1}{D} \tag{D3c}$$

Now the continuity equation is

$$\frac{\partial U}{\partial x} + \frac{\partial W}{\partial z} = 0 \tag{D4}$$

Using (D3a) and (D3c), after which the asterisks are deleted, yields

$$\frac{\partial U}{\partial x} - \frac{\partial U}{\partial\sigma}\frac{1}{D}(\eta_x + \sigma D_x) + \frac{1}{D}\frac{\partial W}{\partial\sigma} = 0 \tag{D5}$$

Now substitute

$$W = \omega + U(\sigma D_x + \eta_x) + \sigma D_t + \eta_t \tag{D6}$$

into (D5) and multiply all terms by D to obtain

$$\frac{\partial U}{\partial x} - \frac{\partial U}{\partial\sigma}(\eta_x + \sigma D_x) + \frac{\partial\omega}{\partial\sigma} + \frac{\partial U}{\partial\sigma}(\eta_x + \sigma D_x) + UD_x + D_t = 0$$

so that

$$\frac{\partial UD}{\partial x} + \frac{\partial\omega}{\partial\sigma} + \frac{\partial\eta}{\partial t} = 0 \tag{D7}$$

since $D_t = \partial\eta/\partial t$. Note that, if in (D6), we set $\omega = 0$ when $\sigma = 0$, $W = U\eta_x + \eta_t$ and if $\omega = 0$ when $\sigma = -1$, $W = -UH_x$ thus satisfying the no-flow top and bottom boundary conditions.

The energy equation may be written

$$\frac{\partial T}{\partial t} + U\frac{\partial T}{\partial x} + W\frac{\partial T}{\partial z} = -\frac{\partial \overline{wt}}{\partial z} \tag{D8}$$

Using (D3a, b, c)

$$\frac{\partial T}{\partial t} - \frac{\partial T}{\partial \sigma}\frac{1}{D}(\eta_t + \sigma D_t) + U\frac{\partial T}{\partial x} - U\frac{\partial T}{\partial \sigma}\frac{1}{D}(\eta_x + \sigma D_x) + W\frac{\partial T}{\partial z} = -\frac{\partial \overline{wt}}{\partial z}$$

Now, multiply through by D and substitute for W from (D6). After some algebra, we obtain

$$D\frac{\partial T}{\partial t} + UD\frac{\partial T}{\partial x} + \omega\frac{\partial T}{\partial \sigma} = -\frac{\partial \overline{wt}}{\partial \sigma}$$

Finally, multiply (D7) by T and add to the above equation to obtain

$$\frac{\partial TD}{\partial t} + \frac{\partial UDT}{\partial x} + \frac{\partial \omega T}{\partial \sigma} = -\frac{\partial \overline{wt}}{\partial \sigma} \tag{D9}$$

Prognostic equations for *U* and *V* may be obtained in the same manner. A difference is that these equations contain a pressure gradient term which, in Cartesian coordinates, is

$$\frac{\partial p}{\partial x} = \frac{\partial}{\partial x}\int_z^\eta \rho' g dz' = \rho(\eta) g\frac{\partial \eta}{\partial x} + \int_z^\eta \frac{\partial \rho'}{\partial x} g dz'$$

After transforming according to (D3a), we have

$$\frac{\partial p}{\partial x} = \rho(0) g\frac{\partial \eta}{\partial x} + g\int_\sigma^0 \left[D\frac{\partial \rho'}{\partial x} - \left(\sigma\frac{\partial D}{\partial x} + \frac{\partial \eta}{\partial x}\right)\frac{\partial \rho'}{\partial \sigma'}\right] d\sigma \tag{D10}$$

The third term in the square brackets can be integrated so that

$$\frac{\partial p}{\partial x} = \rho(\sigma) g\frac{\partial \eta}{\partial x} + g\int_\sigma^0 \left(D\frac{\partial \rho'}{\partial x} - \sigma'\frac{\partial D}{\partial x}\frac{\partial \rho'}{\partial \sigma'}\right) d\sigma \tag{D11}$$

In the past, $\rho(\sigma)$ has been approximated with the constant, ρ_o, which is a valid approximation in most applications. However, John Hunter, in his application of flow under an ice shelf where $\partial\eta/\partial x$ is large, amends (D10) by including that term under the integral sign and approximating $\rho(0)$ by ρ_o. Whereas, (D11) may appear more elegant, (D10) requires fewer code changes.

References

Andre, J. C., G. DeMoor, G. Therry, and R. DuVachat, Modeling the 24-hour evolution of the mean and turbulence structures of the planetary boundary layer, *J. Atmos. Sci., 35*, 1861-1863, 1978.

Asselin, R., Frequency filters for time integrations, *Mon. Weather Rev., 100*, 487-490, 1972.

Baringer, M. O., and J. F. Price, Mixing and spreading of the Mediterranean outflow, *J. Phys. Oceanogr*., submitted, 1996.

Blumberg, A. F.,and G. L. Mellor, A coastal ocean numerical model, in Mathematical *Modelling of Estuarine Physics, Proc. Int. Symp., Hamburg, Aug. 1978*, edited by J. Sunderman and K.-P. Holtz, pp.203-214, Springer-Verlag, Berlin, 1980.

Blumberg, A.F., and G.L. Mellor, Diagnostic and prognostic numerical circulation studies of the South Atlantic Bight, *J. Geophys. Res., 88,* 4579-4592, 1983.

Blumberg, A.F., and G.L. Mellor, A description of a three-dimensional coastal ocean circulation model, in *Three-Dimensional Coastal Ocean Models*, Vol. 4, edited by N.Heaps, pp. 208, American Geophysical Union, Washington, D.C., 1987.

Bryden, H. L., New polynomials for thermal expansion, adiabatic temperature gradient, and potential temperature of sea water, *Deep-Sea Res., 20*, 401-408, 1973.

Bunker, A.F., 1976: Computations of surface energy flux and annual air-sea interaction cycles of the North Atlantic Ocean. *Mon. Weather Rev.* **104**, 1122-1140.

Donelan, M. A., F. W. Dobson, S. D. Smith, and R. J. Anderson, 1995: Reply to Jones and Toba. *J. Phys. Oceanogr.*, **25,** 1908-1909.

Ezer, T., H. Arango and A. F. Shchepetkin, Developments in terrain-following ocean models: intercomparison of numerical aspects, *Ocean Modelling*, **4**, 249-267.

Ezer, T. and G. L. Mellor, 2004: A generalized coordinate ocean model and a comparison of the bottom bounday layer dynamics in terrain-following and in z-level grids. *Ocean Modelling*, **6,** 379-403.

Mellor, G. and A. Blumberg, 2004: Wave Breaking and Ocean Surface Thermal Response. *J. Phys. Oceanogr.*, **34,** 693-698.

Galperin, B., L. H. Kantha, S. Hassid, and A. Rosati, A quasi-equilibrium turbulent energy model for geophysical flows, *J. Atmos. Sci., 45*, 55-62, 1988.

Garratt, J. R., 1977: Review of drag coefficients over oceans and continents. *Mon. Weatherly Rev.*, **105,** 915-929.

Gill, A.E., *Atmosphere-Ocean Dynamics*, 662 pp., Academic Press, New York, 1982.

Jerlov, N. G., *Marine Optics*, 14, 231 pp., Elsvier Sci. Pub. Co., Amsterdam, 1976.

Jungclaus, H., and G. L. Mellor, A three-dimensional model study of the Mediterranean out flow, *J. Mar. Res*, **24,** 41-66, 2000.

Klein, P., A simulation of the effects of air-sea transfer variability on the structure of the marine upper layers, *J. Phys. Oceanogr*., 10, 1824-1841, 1980.

Knudsen, M., Hydrographical Tables. G.E.C. Gad, Copenhagen, pp., Williams and Norgate, London, 19010.

Large, W.G. and S. Pond, 1981: Open ocean momentum flux measurements in moderate to strong winds. *J. Phys. Oceanogr*. **11**, 324-323.

Large, W.G. and S. Pond, 1982: Sensible and latent heat flux measurements over the ocean. *J. Phys. Oceanogr*. **12**, 464-482.

Madala, R. V., and S. A. Piacsek, A semi-implicit numerical model for baroclinic oceans, *J. Comput. Phys*., 23, 167-178, 1977.

Martin, P.J., Simulation of the mixed layer at OWS November and Papa with several models, *J. Geophys. Res*., 90, 903-916, 1985.
Mellor, G.L., and T. Yamada, Development of a turbulence closure model for geophysical fluid problems, *Rev. Geophys. Space Phys*., 20, 851-875, 1982.
Mellor, G. L., Retrospect on oceanic boundary layer modeling an second moment closure, Hawaiian Winter Workshop on *"Parameterization of Small-Scale Processes*", January 1989, University of Hawaii, Honolulu, Hawaii, 1989.
Mellor, G.L., Analytic prediction of the properties of stratified planetary surface layers., *J. Atmos. Sci*., 30, 1061-1069, 1973.
Mellor, G.L., and A.F. Blumberg, Modeling vertical and horizontal diffusivities with the sigma coordinate system, *Mon. Wea. Rev*, 113, 1380-1383, 1985.
Mellor, G.L., and T. Yamada, A hierarchy of turbulence closure models for planetary boundary layers, *J. Atmos. Sci*., 31, 1791-1806, 1974.
Mellor, G. L., L. H. Kantha, and H. J. Herring, On Gulf Stream frontal eddies. A numerical experiment, *Ocean Modelling*, 68, 7-11, 1986.
Mellor, G.L., An equation of state for numerical models of oceans and estuaries. *J. Atmos. Oceanic Tech*. 8, 609-611, 1991.
Mellor, G. L., T. Ezer and L. Y. Oey, The pressure gradient conundrum of sigma coordinate ocean models, *J. Atmos. Oceanic. Technol.*, 11, 1126-1134, 1994.
Mellor, G. L. and T. Ezer, Sea level variations induced by heating and cooling: An evaluation of the Boussinesq approximation in ocean model, *J. Geophys. Res*., 100(C10), 20,565-20,577, 1995.
Mellor, G. L., and X. H. Wang, Pressure compensation and the bottom boundary layer, *J. Phys. Oceanogr*., in press, 1996.
Mellor, G. L., L.-Y. Oey and T. Ezer, Sigma coordinate gradient errors and the seamount problem. *J. Atmos. Oceanic. Technol*., 12, 1122-1131, 1998.
Mellor, G. L., One-dimensional, ocean surface modeling, a problem and a solution. *J. Phys. Oceanogr.*, **31**, 790-809, 2001.
Oey, L.-Y., G.L. Mellor, and R.I. Hires, A three-dimensional simulation of the Hudson-Raritan estuary. Part I: Description of the model and model simulations, *J. Phys. Oceanogr*., 15, 1676-1692, 1985a.
Oey, L.-Y., G.L. Mellor, and R.I. Hires, A three-dimensional simulation of the Hudson-Raritan estuary. Part II: Comparison with observation, *J. Phys. Oceanogr*., 15, 1693-1709, 1985b.
Oey, L.-Y., G.L. Mellor, and R.I. Hires, A three-dimensional simulation of the Hudson-Raritan estuary. Part III: Salt flux analyses, *J. Phys. Oceanogr*., 15, 1711-1720, 1985c.
Paulson, C. A., and J. Simpson, Irradiance measurements in the upper ocean. *J. Phys. Oceanogr*., 7, 952-956, 1977.
Phillips, N. A., A coordinate system having some special advantages for numerical forecasting, *J. Meteorol*., 14, 184-185, 1957.
Price, J. F., C. N. K. Mooers, and J. C. V. Leer, 1978: Obseravtions and simulation of storm-induced mixed-layer deepening. *J. Phys. Oceanogr.*, **8,** 582-599.
Richtmyer, R. D., and K. W. Morton, *Difference Methods for Initial-Value Problems*, 2nd Ed., 405pp., Interscience, New York, 1967.
Rosati, A. and K. Miyakoda, 1988: A general circulation model for upper ocean simulations. *J. Phys. Oceanogr.*, **18,** 1601-1626
Simons, T. J., Verification of numerical models of Lake Ontario. Part I, circulation in spring and early summer, *J. Phys. Oceanogr*., 4, 507-523, 1974.

Smith, S. D., R. J. Anderson, W. A. Oost, C. Kraan, N. Maat, J. DeCosmo, K. B. Katsaros, K. L. Davidson, K. Bumke, L. Hasse, and H. M. Chadwick, 1992: Sea surface wind stress and drag coefficients: the HEXOS results. *Boundary Layer Meteorol.*, **60,** 109-142.

UNESCO. Tenth report of the joint panel on oceanographic tables and standards. UNESCO Tech. Pap.in Marine Science No. 36. UNESCO, paris, 25pp., 1981

Yaglom, A.M. and B.A. Kader, 1974: Heat and Mass Transfer between a rough wall and turbulent fluid flow at high Reynolds and Peclet numbers. *J.Fluid Mech*., **62**, 601-623.

Zavatarelli, M., and G. L. Mellor, A numerical study of the Mediterranean Sea Circulation*, J. Phys. Oceanogr*., 25, 1384-1414, 1995.