USERS GUIDE for a modesplit σ-coordinate numerical ocean model

Version 3.0
September 1, 2002

Jarle Berntsen
Department of Mathematics
University of Bergen
Johs. Bruns gt. 12
N-5008 Bergen, Norway

Abstract. The user guide presents the governing equations for the ocean model and the σ-coordinate transformation. The spatial discretizations, the FORTRAN variables and the numerical routines for propagating the solutions in time are described. To implement the model for a new model area or model problem, the user has to define the model area, the driving forces, boundary conditions and his/her choice of model outputs. The user guide tries to assist a new user through these steps. Instabilities are frequently occurring when running ocean models. Variables and parameters that may or must be adjusted to avoid these are discussed. The model implemented for the fjord system around Osterøy close to Bergen, Norway, is freely available and may be used as a starting point for new applications.

Contents

1 Introduction

This report is a documentation of a σ-coordinate numerical ocean model developed at the Institute of Marine Research and the University of Bergen. The work on the model started in 1995. A first documentation was presented in 1996 [6]. The first version of the model applies implicit methods to solve the depth integrated shallow-water equations. This facilitates the use of the same time step for both the 2-D steps and the 3-D steps, but accuracy may be lost in the 2-D steps due to Courant numbers larger than unity. A time-split version was therefore implemented and described in 1998 [4]. None of the documents above give much details that may assist new users to implement the code for new problems. The present user guide attempts to give more of the information typically needed when setting up the model for a new model area.

The model has been used in some studies, see [8, 5, 4, 7, 15, 16, 14, 2]. In the development of the code especially the ability of the model to maintain the properties of different water masses has been in focus and in [7] the model capabilities in this respect are documented using observations from the SKAGEX experiment in 1990 [13]. Model results from the present code are also compared to model results from the Blumberg and Mellor model [9].

In version 1.1 of the code the statement ETA = ETAP in Main.f90 has been corrected to ETAP = ETA after Yngve Heggelund found this serious bug. In version 1.2 the effects of wind and bottom stress are included in the 2-D part of modesplit.f90. This had negligible effects on the results for the Osterøy case, but when simulating the storm surge described in [25] it proved to be important to include these effects also in the 2-D steps.

In version 2.0 the Coriolis term averaging described in Espelid et al. [17] is introduced to avoid instabilities that may or will occur when applying 1/4-weights for these terms. The superbee-limiter method is also used for advection of momentum. One may select the POM-form on the viscosity terms by setting a logical. One may choose between the UNESCO-formulation as given in Gill [20] or the simplified equation of state found in Wang [42]. A number of routines have been made more efficient on shared memory parallel computers by introducing OPEN MP-directives and rewriting loops. Some routines are made more efficient by introducing arrays that enable looping over wet points only. The work on parallelization and improving the efficiency has been done by Helge Avlesen, Parallab, University of Bergen.

In version 3.0, the time stepping methods are changed. In the 2-D steps, the forward-backward method is replaced with the implicit Θ-method. For $\Theta = 0.5$, the method is 2nd order accurate. For $\Theta = 1.0$, the method is 1th order and fully implicit. In the 2-D steps, the leapfrog method is used as a predictor of values at the new time step. For Θ less than 0.5, this method is unstable. In the 3-D steps, one may still run the model with an explicit technique (except for the vertical mixing). There is also the option of using the Θ-method in the 3-D step. In this case, the

Euler forward method is used as a predictor. The motivation for introducing more implicitness into the model is to avoid noise and growth of numerical errors and thereby allow simulations with less viscosity and more physical free waves. The cost is more computer time. If Θ is larger than 0 in both the 2D steps and the 3D steps, the computing time will almost be doubled since the method is based on computing first a predictor and then a corrector. In version 3.0 there is also an option for using the 4th order advection scheme presented by Shchepetkin and McWilliams (2002) [35]. This method is more accurate than the default TVD-scheme with a superbee limiter. However, over- and under-shooting near fronts may occur. The method is not monotonic.

2 The σ-coordinate model

2.1 The Basic Variables and Equations

The symbols used in the description of the model are given in Appendix A. The model assumes that the weight of the fluid identically balances the pressure (hydrostatic assumption), and that density differences are neglected unless the differences are multiplied by gravity (Boussinesq approximation). The following equations are used to describe the variables as functions of the Cartesian coordinates x, y, z.

The continuity equation is

$$\nabla \cdot \vec{U} + \frac{\partial W}{\partial z} = 0, \tag{1}$$

and the Reynolds momentum equations are

$$\frac{\partial U}{\partial t} + \vec{U} \cdot \nabla U + W\frac{\partial U}{\partial z} - fV = -\frac{1}{\rho_0}\frac{\partial P}{\partial x} + \frac{\partial}{\partial z}(K_M\frac{\partial U}{\partial z}) + F_x, \tag{2}$$

$$\frac{\partial V}{\partial t} + \vec{U} \cdot \nabla V + W\frac{\partial V}{\partial z} + fU = -\frac{1}{\rho_0}\frac{\partial P}{\partial y} + \frac{\partial}{\partial z}(K_M\frac{\partial V}{\partial z}) + F_y, \tag{3}$$

$$\rho g = -\frac{\partial P}{\partial z}. \tag{4}$$

The pressure at depth z may be obtained by integrating equation (4) vertically

$$P = P_{atm} + g\rho_0\eta + g\int_z^0 \rho(\acute{z})d\acute{z}. \tag{5}$$

The conservation equations for temperature and salinity are

$$\frac{\partial T}{\partial t} + \vec{U} \cdot \nabla T + W\frac{\partial T}{\partial z} = \frac{\partial}{\partial z}(K_H\frac{\partial T}{\partial z}) + F_T, \tag{6}$$

$$\frac{\partial S}{\partial t} + \vec{U} \cdot \nabla S + W \frac{\partial S}{\partial z} = \frac{\partial}{\partial z}(K_H \frac{\partial S}{\partial z}) + F_S. \tag{7}$$

The density is computed according to an equation of state of the form

$$\rho = \rho(T, S) \tag{8}$$

taken from [42] if the logical UNESCO is false and from the UNESCO formulation given in Gill [20] otherwise.

Motions induced by small scale processes (sub-grid scale) are parameterized by horizontal and vertical eddy viscosity/diffusivity terms. The horizontal terms F_x, F_y, F_T and F_S may be written

$$F_{x,y} = \frac{\partial}{\partial x}(A_M \frac{\partial(U, V)}{\partial x}) + \frac{\partial}{\partial y}(A_M \frac{\partial(U, V)}{\partial y}), \tag{9}$$

$$F_{T,S} = \frac{\partial}{\partial x}(A_H \frac{\partial(T, S)}{\partial x}) + \frac{\partial}{\partial y}(A_H \frac{\partial(T, S)}{\partial y}). \tag{10}$$

If the logical VISCPOM is true, then the formulations found in Mellor [27] and Mellor et al. [30], see the equations for the σ-coordinate system, are applied. Otherwise, the simplified formulations above are applied.

The horizontal diffusivities, A_M and A_H, may be computed according to Smagorinsky [38]

$$(A_M, A_H) = (C_M, C_H)\Delta x \; \Delta y \; [(\frac{\partial U}{\partial x})^2 + \frac{1}{2} \; (\frac{\partial V}{\partial x} + \frac{\partial U}{\partial y})^2 + (\frac{\partial V}{\partial y})^2]^{\frac{1}{2}}, \tag{11}$$

or chosen to be constant in space and time.

2.1.1 Equations for K_M and K_H

The system of equations given in the previous section is not closed without equations for the vertical viscosity, K_M, and the vertical diffusivity, K_H. There are are a long range of turbulence closure schemes that are being used in ocean models, see for instance Davies and Xing (1995) [12] and Davies *et al.* (1995) [11]. There are a number of factors that may govern the choice of turbulence closure in the vertical: the stratification, tidal forcing, the vertical resolution etc.

The present numerical ocean model may be linked to any of the options found in the above literature or elsewhere.

In cases where a limited number of physical phenomena are occurring simpler models may play a role, but our experience is that when we have model areas where both stratification and tidal forcing are important, we need at least the complexity of the Mellor and Yamada [31] 2 1/2 level model. The governing equations with the modifications due to Galperin et al. [18] for turbulent kinetic energy, $q^2/2$, and turbulent macro scale, l, are given below,

$$\frac{\partial q^2}{\partial t} + \vec{U}\cdot\nabla q^2 + W\frac{\partial q^2}{\partial z} = \frac{\partial}{\partial z}(K_q\frac{\partial q^2}{\partial z}) +$$

$$2K_M\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right] + \frac{2g}{\rho_0}K_H\frac{\partial \rho}{\partial z} - \frac{2q^3}{B_1 l} \tag{12}$$

$$\frac{\partial q^2 l}{\partial t} + \vec{U}\cdot\nabla q^2 l + W\frac{\partial q^2 l}{\partial z} = \frac{\partial}{\partial z}(K_q\frac{\partial q^2 l}{\partial z}) +$$

$$lE_1K_M\left[\left(\frac{\partial U}{\partial z}\right)^2 + \left(\frac{\partial V}{\partial z}\right)^2\right] + \frac{lE_1 g}{\rho_0}K_H\frac{\partial \rho}{\partial z} - \frac{q^3}{B_1}\tilde{W} \tag{13}$$

where

$$\tilde{W} = 1 + E_2\left(\frac{l}{\kappa L}\right)^2 \tag{14}$$

and where

$$L^{-1} = (\eta - z)^{-1} + (H + z)^{-1}. \tag{15}$$

$\kappa = 0.4$ is the von Karman constant. Defining

$$G_H = \frac{l^2}{q^2}\frac{g}{\rho_0}\frac{\partial \rho}{\partial z} \tag{16}$$

the stability functions become

$$S_H[1 - (3A_2B_2 + 18A_1A_2)G_H] = A_2[1 - 6A_1/B_1] \tag{17}$$

and

$$S_M[1 - 9A_1A_2G_H] - S_H[18A_1^2 + 9A_1A_2)G_H] = A_1[1 - 3C_1 - 6A_1/B_1] \tag{18}$$

K_M and K_q are then computed according to

$$K_M = lqS_M \tag{19}$$
$$K_H = lqS_H \tag{20}$$
$$K_q = 0.20lq \tag{21}$$

The empirical values in the expressions above are

$$(A_1, A_2, B_1, B_2, C_1, E_1, E_2) = (0.92, 0.74, 16.6, 10.1, 0.08, 1.8, 1.33) \tag{22}$$

2.1.2 Boundary conditions

At the free surface, $z = \eta(x, y)$, we have

$$\rho_0 K_M \left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}\right) = (\tau_{0x}, \tau_{0y}), \tag{23}$$

$$\rho_0 K_H \left(\frac{\partial T}{\partial z}, \frac{\partial S}{\partial z}\right) = (\dot{T}_0, \dot{S}_0), \tag{24}$$

$$q^2 = B_1^{2/3} u_{\tau s} \tag{25}$$

$$l = 0, \tag{26}$$

where $u_{\tau s} = (\vec{\tau}_0^{\,2})^{1/2}$.

There are no volume fluxes through the side walls. On the side walls free slip conditions for the flow is applied. On the side walls and bottom of the basin there are no advective or diffusive heat and salt fluxes. The vertical velocities at the free surface and at the bottom are given by

$$W_0 = U\frac{\partial \eta}{\partial x} + V\frac{\partial \eta}{\partial y} + \frac{\partial \eta}{\partial t}, \tag{27}$$

$$W_b = -U_b\frac{\partial H}{\partial x} - V_b\frac{\partial H}{\partial y}. \tag{28}$$

The effect of the bottom drag on horizontal velocities is given by

$$\rho_0 K_M \left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}\right) = (\tau_{bx}, \tau_{by}). \tag{29}$$

$$\tag{30}$$

The bottom stress is specified by

$$\vec{\tau}_b = \rho_0 C_D |\vec{U}_b| \vec{U}_b \tag{31}$$

where the drag coefficient C_D is given by

$$C_D = \max[0.0025, \frac{\kappa^2}{(ln(z_b/z_0))^2}] \tag{32}$$

and z_b is the distance of the nearest grid point to the bottom. The von Karman constant $\kappa = 0.4$. In lack of further information we use $z_0 = 0.01m$ for the bottom roughness parameter, see Weatherly and Martin [43]. For q^2 and l the following conditions are applied at the bottom

$$q^2 = B_1^{2/3} u_{\tau b}, \tag{33}$$

$$l = 0, \tag{34}$$

where $u_{\tau b} = (\vec{\tau}_b^{\,2})^{1/2}$.

2.2 The σ-coordinate system

The basic equations have been transformed into a bottom following sigma coordinate system [34]. The independent variables (x, y, z, t) are transformed to (x^*, y^*, σ, t^*), where

$$x^* = x \quad y^* = y \quad \sigma = \frac{z-\eta}{H+\eta} \quad t^* = t. \tag{35}$$

σ ranges from $\sigma = 0$ at $z = \eta$ to $\sigma = -1$ at $z = -H(x, y)$. Introducing the total depth, $D \equiv H + \eta$, the basic equations may now be written as (after deletion of the asterisks)

$$\frac{\partial UD}{\partial x} + \frac{\partial VD}{\partial y} + \frac{\partial \omega}{\partial \sigma} + \frac{\partial \eta}{\partial t} = 0 \tag{36}$$

where the relationship between W and the new vertical velocity ω is given by

$$W = \omega + U\left(\sigma\frac{\partial D}{\partial x} + \frac{\partial \eta}{\partial x}\right) + V\left(\sigma\frac{\partial D}{\partial y} + \frac{\partial \eta}{\partial y}\right) + (1+\sigma)\frac{\partial \eta}{\partial t} \quad . \tag{37}$$

The momentum equations on flux form become

$$\frac{\partial UD}{\partial t} + \frac{\partial U^2 D}{\partial x} + \frac{\partial UVD}{\partial y} + \frac{\partial U\omega}{\partial \sigma} - fVD + \frac{D}{\rho_0}\frac{\partial P_{atm}}{\partial x} + gD\frac{\partial \eta}{\partial x} = \\ \frac{\partial}{\partial \sigma}(\frac{K_M}{D}\frac{\partial U}{\partial \sigma}) - \frac{gD^2}{\rho_0}\int_\sigma^0 \left(\frac{\partial \rho}{\partial x} - \frac{\sigma}{D}\frac{\partial D}{\partial x}\frac{\partial \rho}{\partial \sigma}\right) d\sigma + DF_x, \tag{38}$$

$$\frac{\partial VD}{\partial t} + \frac{\partial UVD}{\partial x} + \frac{\partial V^2 D}{\partial y} + \frac{\partial V\omega}{\partial \sigma} + fUD + \frac{D}{\rho_0}\frac{\partial P_{atm}}{\partial y} + gD\frac{\partial \eta}{\partial y} = \\ \frac{\partial}{\partial \sigma}(\frac{K_M}{D}\frac{\partial V}{\partial \sigma}) - \frac{gD^2}{\rho_0}\int_\sigma^0 \left(\frac{\partial \rho}{\partial y} - \frac{\sigma}{D}\frac{\partial D}{\partial y}\frac{\partial \rho}{\partial \sigma}\right) d\sigma + DF_y. \tag{39}$$

The new conservation equations take the form

$$\frac{\partial TD}{\partial t} + \frac{\partial TUD}{\partial x} + \frac{\partial TVD}{\partial y} + \frac{\partial T\omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}(\frac{K_H}{D}\frac{\partial T}{\partial \sigma}) + DF_T, \tag{40}$$

$$\frac{\partial SD}{\partial t} + \frac{\partial SUD}{\partial x} + \frac{\partial SVD}{\partial y} + \frac{\partial S\omega}{\partial \sigma} = \frac{\partial}{\partial \sigma}(\frac{K_H}{D}\frac{\partial S}{\partial \sigma}) + DF_S, \tag{41}$$

and the horizontal viscosity and diffusion terms are now defined according to

$$DF_{x,y} = \frac{\partial}{\partial x}(A_M\frac{\partial(UD, VD)}{\partial x}) + \frac{\partial}{\partial y}(A_M\frac{\partial(UD, VD)}{\partial y}), \tag{42}$$

If the logical VISCPOM is true, then the formulations found in Mellor [27] and Mellor et al. [30]

$$DF_x = \frac{\partial}{\partial x}(2HA_M\frac{\partial U}{\partial x}) + \frac{\partial}{\partial y}(HA_M(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x})), \tag{43}$$

$$DF_y = \frac{\partial}{\partial y}(2HA_M\frac{\partial V}{\partial y}) + \frac{\partial}{\partial x}(HA_M(\frac{\partial V}{\partial x} + \frac{\partial U}{\partial y})), \tag{44}$$

are applied. Otherwise, the simplified formulations above are applied.

$$DF_{T,S} = \frac{\partial}{\partial x}(DA_H\frac{\partial(T,S)}{\partial x}) + \frac{\partial}{\partial y}(DA_H\frac{\partial(T,S)}{\partial y}). \tag{45}$$

It should be noted that several terms originating from the σ- coordinate transformation are neglected in equations (42), (43), (44) and (45). These simplified formulations for horizontal viscosity/diffusivity terms in σ-coordinate models are suggested by Mellor and Blumberg [28]. In [28] a description of the complete terms is also given. It should be noted that with the present model we try to run with A_H equal to zero to avoid isopycnal diffusion. Using a monotonic and stable scheme for advecting the scalar fields S and T this has proved to be feasible without introducing instabilities.

2.2.1 Vertical boundary conditions

The new boundary conditions for the vertical velocity, ω, in equation (36) become

$$\omega(0) = \omega(-1) = 0. \tag{46}$$

The new conditions at the surface ($\sigma = 0$) becomes

$$\frac{K_M}{D}\left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma}\right) = \frac{1}{\rho_0}(\tau_{0x}, \tau_{0y}), \tag{47}$$

$$\frac{K_H}{D}\left(\frac{\partial T}{\partial \sigma}, \frac{\partial S}{\partial \sigma}\right) = (\dot{T}_0, \dot{S}_0), \tag{48}$$

and at the bottom ($\sigma = -1$) the boundary conditions become

$$\frac{K_M}{D}\left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma}\right) = \frac{1}{\rho_0}(\tau_{bx}, \tau_{by}), \tag{49}$$

$$\frac{K_H}{D}\left(\frac{\partial T}{\partial \sigma}, \frac{\partial S}{\partial \sigma}\right) = 0. \tag{50}$$

3 Mode splitting

To be able to represent gravity waves and the effects of these correctly, the time step must be chosen such that the Courant number become less than unity. To avoid this restriction when propagating the 3-D fields, mode splitting very similar to the splitting described by Berntsen et al. [3] and applied by Slagstad [36] and later described in [1] is applied.

The 3-D velocity field is split into its baroclinic part and its depth integrated part according to:

$$(U(x,y,\sigma),V(x,y,\sigma)) \quad = \quad (U_A(x,y)+U_B(x,y,\sigma),V_A(x,y)+V_B(x,y,\sigma)) \tag{51}$$

where

$$(U_A(x,y),V_A(x,y)) \quad = (\textstyle\int_{-1}^{0} U(x,y,\sigma)d\sigma, \int_{-1}^{0} U(x,y),\sigma d\sigma).$$

Depth integrating the moment equations and the continuity equation gives, neglecting for a moment the atmospheric pressure terms and the vertical viscosity terms, gives:

$$\frac{\partial U_A D}{\partial x}+\frac{\partial V_A D}{\partial y}+\frac{\partial \eta}{\partial t}=0, \tag{52}$$

$$\begin{aligned}&\frac{\partial U_A D}{\partial t}-fV_AD+gD\frac{\partial \eta}{\partial x}=\\ &\quad\frac{\partial}{\partial x}(A_{M2D}\frac{\partial U_A D}{\partial x})+\frac{\partial}{\partial y}(A_{M2D}\frac{\partial U_A D}{\partial y})+\frac{1}{\rho_0}(\tau_{0x}-\tau_{bx})+A_x\end{aligned} \tag{53}$$

and

$$\begin{aligned}&\frac{\partial V_A D}{\partial t}+fU_AD+gD\frac{\partial \eta}{\partial y}=\\ &\quad\frac{\partial}{\partial y}(A_{M2D}\frac{\partial V_A D}{\partial y})+\frac{\partial}{\partial x}(A_{M2D}\frac{\partial V_A D}{\partial x})+\frac{1}{\rho_0}(\tau_{0y}-\tau_{by})+A_y,\end{aligned} \tag{54}$$

where

$$(A_x,A_y) \quad = \quad (\int_{-1}^{0}\Delta U d\sigma, \int_{-1}^{0}\Delta V d\sigma), \tag{55}$$

$$\Delta U \quad = \quad -\frac{gD^2}{\rho_0}\int_{\sigma}^{0}\left(\frac{\partial \rho}{\partial x}-\frac{\sigma}{D}\frac{\partial D}{\partial x}\frac{\partial \rho}{\partial \sigma}\right)-\frac{\partial U^2 D}{\partial x}+\frac{\partial UVD}{\partial y}+\frac{\partial U\omega}{\partial \sigma}, \tag{56}$$

$$\Delta V \quad = \quad -\frac{gD^2}{\rho_0}\int_{\sigma}^{0}\left(\frac{\partial \rho}{\partial y}-\frac{\sigma}{D}\frac{\partial D}{\partial y}\frac{\partial \rho}{\partial \sigma}\right)-\frac{\partial UVD}{\partial x}+\frac{\partial V^2 D}{\partial y}+\frac{\partial V\omega}{\partial \sigma}. \tag{57}$$

The non-linear and the internal pressure terms are represented through A_x and A_y. The vertical integration is exact except for the horizontal viscosity terms. These terms take mainly care of small scale oscillations, and the 2-D A_{M2D} field is computed according to (11) where (U, V) is replaced by (U_A, V_A) or chosen to be constant in space and time.

The equations for the baroclinic fields U_B and V_B become after subtracting (53) from (38) and (54) from (39) :

$$\frac{\partial U_B D}{\partial t} - fV_B D + \frac{1}{\rho_0}(\tau_{0x} - \tau_{bx}) + A_x = \frac{\partial}{\partial x}(A_M \frac{\partial U_B D}{\partial x}) + \frac{\partial}{\partial y}(A_M \frac{\partial U_B D}{\partial y}) + \frac{\partial}{\partial \sigma}(\frac{K_M}{D}\frac{\partial U_B}{\partial \sigma}) + \Delta U \tag{58}$$

and

$$\frac{\partial V_B D}{\partial t} + fU_B D + \frac{1}{\rho_0}(\tau_{0y} - \tau_{by}) + A_y = \frac{\partial}{\partial y}(A_M \frac{\partial V_B D}{\partial y}) + \frac{\partial}{\partial x}(A_M \frac{\partial V_B D}{\partial x}) + \frac{\partial}{\partial \sigma}(\frac{K_M}{D}\frac{\partial V_B}{\partial \sigma}) + \Delta V. \tag{59}$$

4 The numerical σ-coordinate model

The governing equations form a set of simultaneous partial differential equations which cannot be solved using known analytic methods. Therefore the equations have been discretized using finite difference methods. The routines called to approximate the solution of the governing equations in each time step of the model are described in this section. The horizontal finite difference scheme is staggered, and the Arakawa C-grid [33] has been used, see figures 1, 2 and 3.

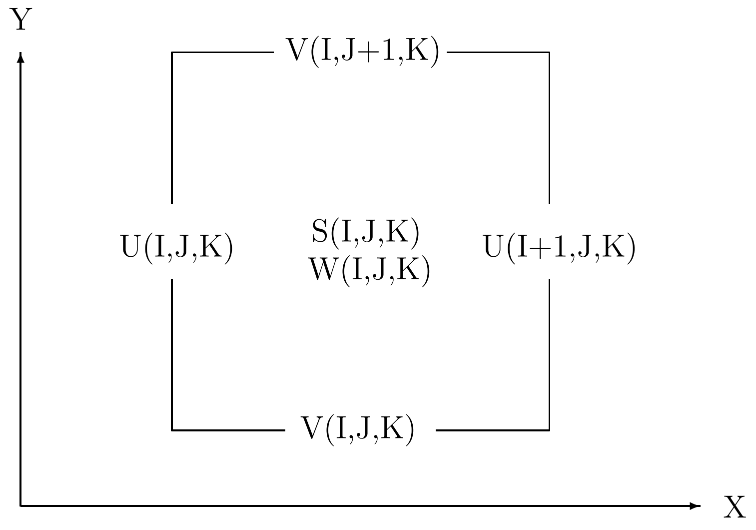

Figure 1. Horizontal view of the location of 3-D variables in the staggered grid. T, RHO and other scalar fields are defined in S-points. KM, KH, Q2 and Q2L are defined in W-points.

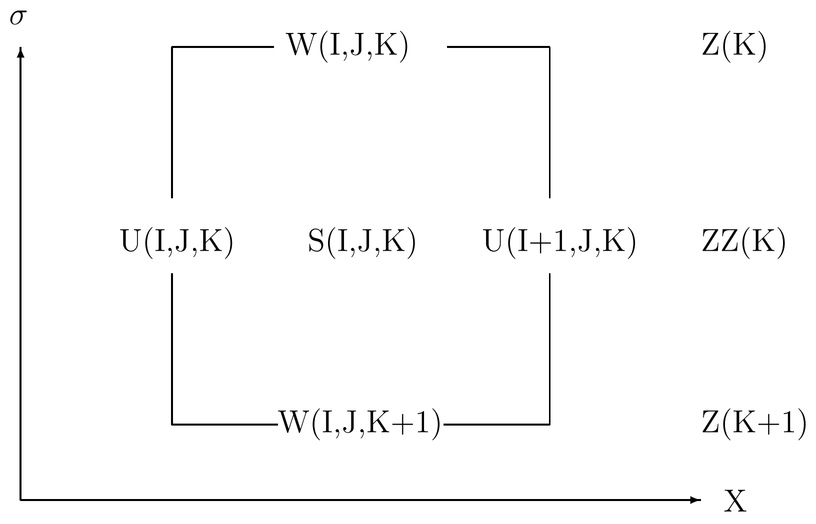

Figure 2. Vertical view of the location of 3-D variables in the staggered grid. T, RHO and other scalar fields are defined in S-points. KM, KH, Q2 and Q2L are defined in W-points.

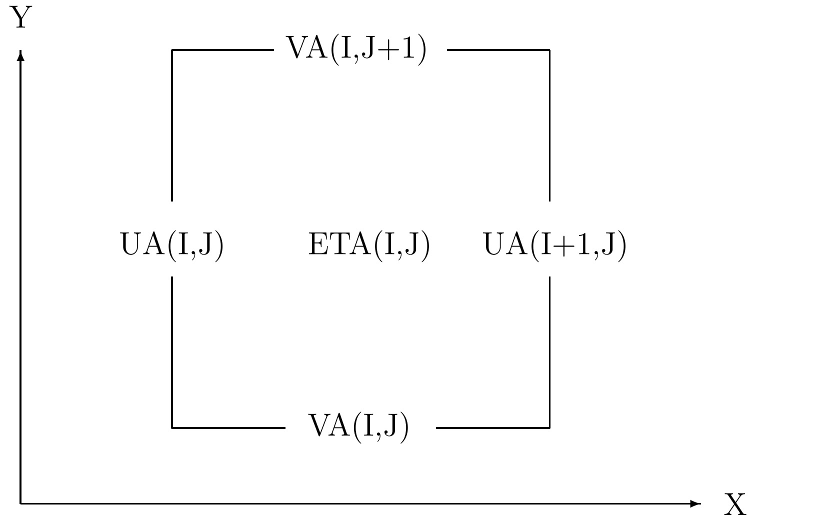

Figure 3. Horizontal view of the location of 2-D variables in the staggered grid.

The model is written in FORTRAN 90 and the discrete versions of the state variables and parameters are gathered in a module, STATE, that may be addressed by all subroutines. Equations (36) - (41) are stepped forward in time using the same time step for all equations. The method of fractional steps is applied. That is a sequence of subroutines is called to perform specific subtasks and update the corresponding variables in MODULE STATE in each time step. After all subroutines are called the effects of all terms in the governing equations are included.

A description of the variables in MODULE STATE is given in the appendix.

4.1 The time stepping algorithm

Below follows a coarse algorithm of the actions taken during one 3-D time step in the code. The involved subroutines are indicated in parentheses.

For each time step until finished do

> Compute possible atmospheric, river runoff and/or tidal forcing
> Compute ρ^n from S^n and T^n. (DENS or DENSUNESCO)
> Compute K_M and K_H from U^n, V^n, and ρ^n. (MY2HALV, UPSTREAMQ and BOUND)
> Save values at time step n of the prognostic variables.
> Compute ΔU and ΔV from U^n, V^n, W^n, and ρ^n. (INTERNAL and SUPERBEEUV)
> Compute in a predictor step η^*, U^*, V^*, W^*, S^* and V^*. (MODESPLIT, ADVECT or SUPERBEEF and BOUND)
> If $\Theta_{3D} > 0$ Then
>> Compute ρ^* from S^* and T^*. (DENS or DENSUNESCO)
>> Compute ΔU and ΔV from U^*, V^*, W^*, and ρ^*. (INTERNAL and SUPERBEEUV)
>> Compute in a corrector step η^{**}, U^{**}, V^{**}, W^{**}, S^{**} and V^{**}. (MODESPLIT, ADVECT or SUPERBEEF and BOUND)
>> Estimate final values for η at time step $n+1$ from
>> $\eta^{n+1} = \eta^n + \Theta_{3D}(\eta^{**} - \eta^n) + (1 - \Theta_{3D})(\eta^* - \eta^n)$
>> and correspondingly for U^{n+1}, V^{n+1}, W^{n+1}, S^{n+1}, and T^{n+1}.
>
> Else If $\Theta_{3D} = 0$ Then
>> Use the predictor values as estimates at time step $n+1$.
>
> End if
> Apply vertical diffusion to S^{n+1}, and T^{n+1} (VERTDIFF and BOUND).

End main do-loop in time

In the routine MODESPLIT N2D 2D steps are taken to propagate the depth integrated velocities and η one 3D time step.

The order in which some of the above operations are taken, is not unique. However, some operations must be performed before others, and the non-expert should be careful with changing the order of the operations. It is particularly important that the fields UADV, VADV, W, ETA and ETAP used for advecting scalar fields satisfy the equation of continuity. That is, ETAP must be set equal to ETA before the call to MODESPLIT and MODESPLIT must be called before SUPERBEEF or ADVECT.

It is normally not necessary to modify the subroutines DENS, DENSUNESCO, MY2HALV, UPSTREAMQ, INTERNAL, SUPERBEEUV, MODESPLIT, SMAGOR2D, HORVISCUV2D, STEPU2D, STEPV2D, SMAGOR, HORVISCUV, STEPU3D, STEPV3D, VERTVISCUVB, UPDATEDD, SUPERBEEF/ADVECT and VERTDIFF and the order in which these are called from one application to another.

4.2 SUBROUTINES DENS AND DENSUNESCO

If UNESCO is false, the subroutine DENS is called from the main program in the beginning of the main do-loop in time to update the density field, RHO, from S and T before calling MY2HALV and INTERNAL which both are depending on the density field. DENS applies the equation of state given in [42]. If UNESCO is true, DENSUNESCO is called and the UNESCO formulation for $p = 0$ given in Gill [20] is applied. The unit is kgm^{-3}. To reduce the cancellation errors associated with differences in the density field computed both in MY2HALV and INTERNAL, the average density, RHOAVER, is subtracted from the density before exiting from SUBROUTINE DENS or DENSUNESCO.

In version 3.0, a test on S < 0 is introduced in DENSUNESCO since S**(3/2) occur in the full equation of state. This operation gives *nan* if S < 0. This term in the equation of state is neglected if S < 0 in DENSUNESCO. This unphysical value of S may occur due to undershooting in the advection.

4.3 SUBROUTINE MY2HALV

The subroutine MY2HALV is called from the main program to update the fields q^2 and q^2l taking into account all terms of equations (12) and (13) except for the advective terms which are treated by UPSTREAMQ. The approximation to q^2 and q^2l are stepped forward in time using an explicit method for the production terms and an fully implicit method for the loss terms. The $\frac{\partial \rho}{\partial z}$ terms may change sign, and depending on the sign explicit or implicit methods are applied. This has proved to be necessary to avoid negative values of q^2 and q^2l .

From q^2 and q^2l K_M, K_H and K_q are computed following equations (19), (20) and (21).

For stratified cases the loss terms will typically be dominating and the vertical viscosity and diffusivity will be determined not by the equations for q^2 and q^2l, but by the minimum allowed values KHMIN and KMMIN defined in the computer code. Model results have proved to be sensitive to these values. We try to keep these as small as possible to maintain the density structure in the ocean as well as possible. Typically values are KHMIN $= 1 \times 10^{-7} m^2 s^{-1}$ and KMMIN $= 1 \times 10^{-5} m^2 s^{-1}$. Larger values of KHMIN, in particular, may cause unphysical down mixing of surface water in stratified areas.

KHMIN and KMMIN are given in setupfile.dat.

4.4 SUBROUTINE UPSTREAMQ

The subroutine UPSTREAMQ is called from the main program after the call to MY2HALV to advect the fields q^2 and $q^2 l$ with the upstream method. This is an economic, but diffusive method. The time scales for production or loss of q^2 and $q^2 l$, which is computed in MY2HALV, are, however, typically smaller than the advective time scales over a grid cell. Therefore, we consider it to be mostly a waste of computer time to apply more accurate advection schemes for the advection part of the problem. For coarser grids advection of q^2 and $q^2 l$ may be neglected because of the time scale considerations.

Define ADVTURB equal to .FALSE. in setupfile.dat to avoid advection of turbulence.

4.5 SUBROUTINE INTERNAL

The contributions to ΔU and ΔV given in equations (55) to (59) from the internal pressure terms represented by the vertical integrals in equations (56) and (57) may be computed with a selection of methods. The internal pressure force is a very important force which in areas with steep bottom topography, as we have in Norwegian waters, is difficult to approximate in σ-coordinate models. Large and erroneous currents may be demonstrated. However, there is also evidence that these errors may die out due to advective adjustment of the density field in prognostic experiments. See for instance [40, 32, 21, 39, 29, 30, 37, 22, 26, 10] for further details and discussions. In recent studies for the North Sea and the Skagerrak using 20km and 4km horizontal grid resolution Eliassen and Berntsen [16] apply measurements across the Norwegian trench to evaluate different proposed internal pressure methods. The major conclusion is that the the model results are relatively robust to the choice of internal pressure method and that the the 2nd order method described by Blumberg and Mellor [9] is a reasonable choice. This method is relatively inexpensive compared to the alternative methods and we therefore apply this method in real applications until it is more clearly demonstrated that other methods produce better results.

The horizontal differential operators are estimated with 2nd order central differences and averages in the horizontal are also of 2nd order. In the vertical the integrals are estimated with the trapezoidal rule which is 2nd order except for the integrals from the surface to the first velocity point which are estimated using the values of the integrand at the velocity point. This gives a 1th order method for this interval. However, this distance is typically very small and this is not expected to degrade the quality of the internal pressure estimates.

WARNING:
The first executable statement in internal.f90 is

REFDENS = 1.0 / RHO0

This is correct if RHO is computed in kgm^{-3}. Some prefer to compute RHO in gcm^{-3}. The statement above must then be modified to: REFDENS = 1000.0 / RHO0.

4.6 SUBROUTINE SUPERBEEUV

The subroutine SUPERBEEUV is called from the main program after the call to INTERNAL. SUPERBEEUV estimates contributions to ΔU and ΔV due to the terms in equations (56) and (57) describing advection of momentum and add these to the contributions estimated by INTERNAL. SUPERBEEUV applies an combination of a 1th orher upwind method and a 2nd order Lax-Wendroff method to compute the advective fluxes. The method is implemented following the description in Yang and Przekwas [44].

If RIVLOG is .TRUE. , fluxes of the momentum from rivers are also included.

4.7 SUBROUTINE MODESPLIT

After INTERNAL and SUPERBEEUV have estimated ΔU and ΔV, MODESPLIT is called to update the U, V , W and ETA fields including the effects of all terms in the momentum equations and the equation of continuity, except the atmospheric pressure term.

In MODESPLIT the following steps are taken:

- The contributions in ΔU and ΔV are vertically integrated to produce terms A_x and A_y according to equation (55).
- The U and V fields are split into 2-D external or barotropic and 3-D internal or baroclinic parts according to equation (51).
- Then for each call to MODESPLIT N2D steps are used to propagate the depth integrated velocities U_A and V_A and the water level with a Θ-method using

DTE = DT/N2D as time step. In each 2D time step the following actions are taken:

Estimate the A_{M2D} field through a call to SMAGOR2D, if SMAG is true.

Include the effects of horizontal viscosity on the U_A and V_A fields. (Calls to HORVISCUV2DPOM or HORVISCUV2D)

In the first substep, a forward method is used as a predictor method.

That is: $\eta^* = \eta^n +$ DTE $\times \Delta\eta^n$,

where $\Delta\eta^n$ contain the spatially discretized terms of the η-equation.

In the first substep, U_A^* and V_A^* are computed correspondingly with the forward method.

In each of the following 2D sub-steps, the leapfrog method is used as predictor for η^*, U_A^*, and V_A^*.

That is: $\eta^* = \eta^{n-1} + 2 \times DTE \times \Delta\eta^n$.

The final values at time step $n+1$ are computed with a θ-method according to:

$\eta^{n+1} = \eta^n + DTE \times (\Theta_{2D}\Delta\eta^* + (1 - \Theta_{2D})\Delta\eta^n)$.

Θ_{2D} must be in the range $0.5 \leq \Theta_{2D} \leq 1$ to ensure stability.

- After the 2D steps, the velocity fields, that have been used to change the water level from ETAP to ETA over the N2D 2D time steps, are saved in the UADV and VADV arrays. It is essential that these non-divergent fields are used in the advection of scalar fields and momentum to maintain the mass balance.
- Estimate the A_M and A_H fields through a call to SMAGOR, if SMAG is true.
- Estimate the horizontal viscosity fluxes in equations (58) and (59) and store these through a call to HORVISCUVPOM if VISCPOM is true, and a call to HORVISCUV otherwise.
- Update the U_B, V_B, U and V fields taking into account the effects of the Coriolis terms, the horizontal viscosity terms, the A_x, A_y, ΔU and ΔV terms of equations (58) and (59).
- Call VERTVISCUVB to include the effects of vertical viscosity on the momentum.
- Include the effects of surface and bottom stress on the U_B, V_B, U and V fields.
- Compute the sigma coordinate vertical velocities W from equation (36) using fluxes produced by UADV and VADV and the water levels ETA and ETAP.

If we are running experiments with river runoff, RIVLOG = .TRUE. , the effects of the runoff on the U, V, W and ETA fields are also included in modesplit.f90.

4.7.1 SUBROUTINE SMAGOR2D

SMAGOR2D computes, if SMAG is .TRUE., the 2D viscosity field from a depth integrated variant of equation (11):

$$A_{M2D} = C_M \Delta x\ \Delta y\ [(\frac{\partial U_A}{\partial x})^2 + \frac{1}{2}\ (\frac{\partial V_A}{\partial x} + \frac{\partial U_A}{\partial y})^2 + (\frac{\partial V_A}{\partial y})^2]^{\frac{1}{2}}. \tag{60}$$

Explicit methods in time are used to include the viscosity effects. Too large values of A_{M2D} may cause instabilities and/or oscillations. Therefore, an upper bound for A_{M2D} is selected to be:

$$A_{M2D} \leq \frac{1}{16}\frac{\Delta x^2}{DTE}$$

NOTE: Recent experiences with applying the model with small values of Δx in areas with strong variability suggest that A_{M2D} should be set larger than the values produced with the above formulation and values of C_M of the order 1. One may try to set SMAG equal to .FALSE. in setupfile.dat, and choose A_{M2D} equal to some fraction, for instance 0.5, times the upper bound $\frac{1}{16}\frac{\Delta x^2}{DTE}$ to avoid 2DT and/or $2\Delta x$ noise in the solutions.

4.7.2 SUBROUTINES HORVISCUV2D and HORVISCUV2DPOM

HORVISCUV2D computes the contributions from the horizontal viscosity terms of equations (53) and (54) and stores these in the arrays C2U and C2V using the formulation (42). HORVISCUV2DPOM is called if VISCPOM is true and then the formulations (43) and (44) are applied.

4.7.3 SUBROUTINE SMAGOR

SMAGOR computes, if SMAG is .TRUE., the 3D viscosity field A_M and the diffusivity field A_H from equation (11).

Reasonable values for the viscosity coefficient C_M are in the range 0.1 to 1.0. Recent experiences suggest that for smaller values of Δx larger values of C_M is necessary to avoid noise in the solution. See also the note at the end of this section. To avoid unphysical isopycnal mixing of the salinity, temperature and density fields the diffusivity coefficient C_H is by default selected to be zero. The experience so far is that the model runs stable also for strongly stratified cases with $A_H = 0$. With this choice we avoid much of the isopycnal unphysical diffusion normally associated with sigma coordinate models. There is still some isopycnal diffusion in the model due to the inherent diffusion in the advection scheme, but also this is reduced by applying a gradient preserving advection scheme, see the comments on SUPERBEEF. If the choice $C_H = 0.$ causes instabilities, we may cautiously increase towards the value selected for C_M. To run the model over a long period with large values of C_H will produce an unphysical smoothing of the density field. The effects of this may be seen in many model outputs.

Explicit methods in time are used to include the viscosity effects. Too large values of A_M or A_H may cause instabilities and/or oscillations. Therefore, an upper bound for A_M and A_H is selected to be:

$$(A_M, A_H) \leq \frac{1}{16}\frac{\Delta x^2}{DT}$$

NOTE: Recent experiences with applying the model with small values of Δx in areas with strong variability suggest that A_M should be set larger than the values produced with the above formulation and values of C_M of the order 1. One may try to set SMAG equal to .FALSE. in setupfile.dat, and choose A_M equal to some fraction, for instance 0.5, times the upper bound $\frac{1}{16}\frac{\Delta x^2}{DT}$ to avoid 2DT and/or $2\Delta x$ noise in the solutions.

4.7.4 SUBROUTINES HORVISCUV and HORVISCUVPOM

HORVISCUV computes the contributions from the horizontal viscosity terms of equations (58) and (59) and stores these in the arrays C3U and C3V using the formulation (42) if VISCPOM is false. HORVISCUVPOM is called if VISCPOM is true and then the formulations (43) and (44) are applied.

4.7.5 SUBROUTINE STEPU3D and SUBROUTINE STEPV3D

STEPU3D and STEPV3D propagates the U_B and V_B fields taking into account the Coriolis terms, the horizontal viscosity terms and contributions from ΔU, ΔV, A_x and A_y of equations (58) and (59). The time stepping is performed with an explicit method. The order in which the U_B-equation and V_B-equation is treated is alternated from one 3D time step to the next. The Coriolis term averaging is done with the method given in Espelid et al. [17].

4.7.6 SUBROUTINE VERTVISCUVB

VERTVISCUVB propagates the U_B and V_B fields taking into account the effects of the vertical viscosity terms

$$\frac{\partial U_B D}{\partial t} = \frac{\partial}{\partial \sigma}(\frac{K_M}{D}\frac{\partial U_B}{\partial \sigma})$$

and

$$\frac{\partial V_B D}{\partial t} = \frac{\partial}{\partial \sigma}(\frac{K_M}{D}\frac{\partial V_B}{\partial \sigma}),$$

using the boundary conditions

$$\frac{K_M}{D}\left(\frac{\partial U_B}{\partial \sigma}, \frac{\partial V_B}{\partial \sigma}\right) = \frac{1}{\rho_0}(\tau_{0x}, \tau_{0y}),$$

at the surface and

$$\frac{K_M}{D}\left(\frac{\partial U_B}{\partial \sigma}, \frac{\partial V_B}{\partial \sigma}\right) = \frac{1}{\rho_0}(\tau_{bx}, \tau_{by}),$$

at the bottom. The solutions are stepped forward in time using a fully implicit method in order to avoid time step limitations associated with thin sigma layers.

4.8 SUBROUTINE WREAL

WREAL computes the Z-coordinate vertical velocities W from equation (37). The model variable WR is a diagnostic variable in the model and it is not strictly necessary to call WREAL each time step to compute WR. However, to diagnose the solution and to understand where and when convergence/divergence occur it is interesting to study the WR field.

4.9 SUBROUTINE UPDATEDD

SUBROUTINE UPDATEDD updates the Dynamic Depth fields D, DU, DV, DDZ, DUDZ and DVDZ after ETA is updated in each time step. By computing the dynamic depths one place in the code one saves operations at the cost of using more storage.

4.10 SUBROUTINES SUPERBEEF and ADVECT

If the logical FOURTH is false, the subroutine SUPERBEEF is called from the main program after the velocities at the new time step are updated. It is essential that the fields UADV,VADV,W,ETA and ETAP satisfy the equation of continuity before the call to SUPERBEEF since these fields are used to propagate the scalar fields. The conservation equations (41) and (40) for S and T are used to propagate the solution one step forward in time using SUPERBEEF taking the advective terms and the horizontal diffusivity terms into account. We want the numerical advection technique to be 2nd order accurate in areas with small gradients, gradient preserving near fronts and monotonic. Among the many recent advection schemes claiming to satisfy these conditions we have chosen to use a superbee limiter scheme due to Roe and Sweby [41]. This scheme performed favorably in a comparison due to Yang and Przekwas [44]. The method is implemented following the description in Yang and Przekwas [44].

If the logical FOURTH is true, an advection scheme described in [35] is used. This scheme is fourth order where the solution is smooth, and second order otherwise. The local order is thus higher. However, small oscillations creating over- and/or under-shooting near fronts may occur. That is: the scheme is not TVD.

If RIVLOG is .TRUE., advective fluxes of riverine inputs into the ocean are also included.

4.11 SUBROUTINE VERTDIFF

VERTDIFF propagates the solution of a scalar field F taking into account the effects of the vertical diffusivity terms

$$\frac{\partial FD}{\partial t} = \frac{\partial}{\partial \sigma}\left(\frac{K_H}{D}\frac{\partial F}{\partial \sigma}\right)$$

and using the boundary conditions

$$\frac{K_H}{D}\frac{\partial F}{\partial \sigma} = \frac{1}{\rho_0} FSURF,$$

at the surface and

$$\frac{K_H}{D}\frac{\partial F}{\partial \sigma} = 0$$

at the bottom. The solutions are stepped forward in time using a fully implicit method in order to avoid time step limitations associated with thin sigma layers.

5 Getting started using BOM

5.1 Requirements and how to obtain the code

The minimum requirements for using BOM are as follows:

- a computer with a big brain.
- a text editor (Emacs, vi, pico, notepad etc.)
- a Fortran 90 compiler.
- a make utility (not absolutely necessary, but very handy!)

Since you are reading this manual there is a chance you already have the source code for BOM, but if you don't - please go to the web page:

`http://www.mi.uib.no/~jarleb/BOM.html`

You may then download the source code by clicking on "Download sourcecode". You will then receive a file bom_3.0.tar.gz, which first have to be gunzipped with

```
gunzip bom_3.0.tar.gz
```

which produces bom.tar. Place bom.tar in the directory where you want the source code to be placed and give the unix command

```
tar xvf bom_3.0.tar
```

All necessary FORTRAN codes + a makefile + 2 input files (setupfile.dat and bottom1km.dat) will be placed on this catalog.

From the same homepage the latest version of this user manual is placed (Click on BOMguide to get the manual).

5.2 Building BOM

In the BOM directory you have a file called "Makefile". This file contain instructions for the "make" utility for how to compile and link BOM on different platforms, as well as perform various common tasks. When we build BOM we normally just use the command "make", sometimes followed by an argument (in UNIX known as a target) here's a list over what the targets of the BOM makefile does:

make clean	remove all object files, module files and the executable
make mod	compile the modules
make bom	build BOM's executable

So when you build BOM for the first time, you use the two commands

make mod
make bom

If everything went well you run bom by typing "bom" at the prompt. So normally after the first build, the only necessary command to rebuild BOM after doing minor changes to the code is a simple "make" at the prompt. When you change something in one of the module files (state.f90, mod_bound.f90 or mod_main.f90), the procedure for rebuilding bom is as follows:

make clean
make mod
make bom

Remember that each time a modification, for instance a change in a parameter statement, in one of the modules is done, we have to go through the above procedure. Variables must not be assigned values in the modules. Parameters are given values in PARAMETER statements in the modules.

6 Setting BOM up for a practical application

This section contain a description of the structure of BOM, and briefly discuss how and where to modify BOM's subroutines for a realistic simulation.

The major tasks involved when setting up the model for a new model area is described below:

1)Select a rectangular model area that may be covered with IM $\times$ JM horizontal grid cells. Find the depths at the cell centers of all cells and store these to a data file. The depths of land or iceland cells may be set to zero or to some negative value. The routine dypgrin.f90 will then produce the masking arrays FSM, DUM and DVM in order to facilitate the book-keeping of the closed boundaries between ocean and land. Define the variables IM, JM and KB in the module state.f90.

2) Define the boundary arrays and possibly river specific variables in the module mod_bound.f90.

3) Define all problem specific variables in mod_main.f90. Especially this last module will often have to be modified during the model development and testing.

4) Open all data files to be used in files.f90.

5) Define the variables to be read from the setup file: setupfile.dat by the routine setup.f90 called from Main.f90.

6) Modify, if necessary, dypgrin.f90 and depth.f90 to the specific application.

7) Modify fieldinit.f90 such that all dynamic variables are initialized according to your specifications.

8) If FRS-zones are used, the routine ncalfa.f90 may have to be adjusted. The parts of the boundary where relaxation should be avoided must be masked away in relax.f90. Make sure that relaxation really takes place at all open boundaries by checking relax.f90 properly.

9) Specify in bound.f90 the boundary conditions that you want to apply at all open boundaries.

10) Modify output.f90.

11) Redefine the forcing subroutines rivers.f90, wind.f90, tidefrs.f90 and tideinit.f90 according to your needs.

Descriptions of the subroutines that a user should get acquainted with and often modify when working with the model are given below. These are the routines that are outside the numerical engine of the model and that deliver inputs and extract outputs from this numerical core.

Check that the model runs stably without rivers, wind and tides, before you start to add forcing, one at the time!

6.1 MODULE STATE

This module defines all major variables of the code. Normally only the parameter statement

PARAMETER (IM=33, JM=55, KB=32)

defining the size of the model domain should have to redefined from one model application to another. See the appendix for a description of all variables in MODULE STATE.

6.2 MODULE MOD_BOUND

This module defines all the boundary arrays normally used. If you are using FRS-zones, the parameter statement

PARAMETER (LB=7)

may have to bee modified.

If you have river forcing, the parameter

PARAMETER (NUMC=7)

defining the number of coastal cells connected to rivers, may have to be modified.

If tidal forcing is included, the number of tidal components defined by

PARAMETER (KCON=4)

may have to be modified.

See the appendix for a description of all variables in MODULE MOD_BOUND.

6.3 MODULE MOD_MAIN

This module uses the variables defined in MODULE STATE and MODULE MOD_BOUND and declares the additional variables to be used by PROGRAM MAIN and utility routines. This module will be very application specific. See the appendix for a description of some of the variables normally defined in MODULE MOD_MAIN.

6.4 PROGRAM MAIN - the main driver routine

Program MAIN contains the main steps involved when running the ocean model. First there is a sequence of calls to subroutines FILES, SETUP, DYPGRIN, FIELDINIT, NCALFA and possibly TIDINIT to open files and to initialize the model variables. Then there is the main do loop in time where the solutions of U, V, W, ETA, S, T, RHO, Q2 and Q2L are updated at each new time step taking all terms of the governing equations into account. The routine OUTPUT is also called to test on possible output.

6.5 SUBROUTINE FILES

All files to be used for input or output to/from the model are opened in files.f90. By having the OPEN statements in one place one reduces the danger of using file numbers already used.

This subroutine must normally be adapted to the given application.

6.6 SUBROUTINE SETUP

SETUP reads major model variables and parameters from the file setupfile.dat. The variables to be read are ISTART, THETA2D, THETA3D, ADVTURB, SMAG, RIVLOG, TIDE, WIND, ATMPRESS, FOURTH, DEPMIN, NFILTER,DSBOT, NDIVIS, N2D, NUMSTEP, DX, CM, CM2D, CH, KMMIN and KHMIN. See the appendices for a description of these variables.

The values of ISTART are used to initiate NDATE. TIME is initiated to 0. DT is computed from NDIVIS and DY is set equal to DX in setup.f90.

If SMAG is .FALSE., then AM is given the value CM, AM2D the value CM2D and AH the value CH in setup.f90. If too large values are chosen, a warning is written to the user and the program is stopped.

THETA2D must satisfy $0.5 \leq$ THETA2D ≤ 1.0 . THETA3D must satisfy $0.0 \leq$ THETA3D ≤ 1.0 .

6.7 SUBROUTINE DYPGRIN

DYPGRIN reads from file the depths of all model cells and places these in the depth array H. H may be Shapiro filtered NFILTER times to avoid $2\Delta x$ -noise in the model fields. On the other hand excessive filtering destroys the topography. The maximum depth, DEPMAX, in the model region is computed. This depth may be used to check whether the selected time step is small enough to satisfy the CFL-criterion. Ocean cells with depths less than DEPMIN are defined as land cells. The Coriolis parameter array COR is defined. The masking arrays FSM, DUM and DVM and the model static depths H, HU and HV are computed. The bottom drag coefficients CBC are computed after DEPTH is called to set up the σ-layer thicknesses. Based on HU and HV the square roots of these arrays are also computed and stored in HUSQR and HVSQR. These roots are applied in the Coriolis term averaging and are introduced to avoid instabilities, see [17].

The subroutine DYPGRIN may have to be modified from one application to another.

6.7.1 SUBROUTINE DEPTH

DEPTH computes the vertical distribution of KB σ-interfaces Z, and the 1-D arrays DZ, DZR, DZZ, ZZ derived from Z. Very often a distribution suggested by Lynch *et al.* [23] is used. Their formula distribute the layers symmetricly about the midpoint in the vertical and such that we get gradually a finer resolution towards the surface and the bottom. The thickness of the bottom layer, DSBOT, is a parameter that should be selected as a function of KB. DSBOT is defined in the setup.f90. For KB=21, we have often applied DSBOT=0.01.

For other applications other distributions focusing for instance more on the upper layer may facilitate a better representation of the phenomena to be studied. The subroutine DEPTH must be modified accordingly. In the Osterøy implementation we have chosen KB=32, and chosen the 5 top surface layers such that their thicknesses will be 1m at 200m depth in order to try to resolve the fresh water layer at the surface of the fjord. After that we have gradually increased the layer thicknesses except for at the bottom where we again need some thin layers because of the bottom friction.

6.7.2 SUBROUTINE CONDITION

CONDITION sets up the logical arrays used to manually do gather-scatter optimization; counts the number of sea points, and generates array indices to access sea points. now looping over wet points can be done without if-statements inside the loop. This routine is written by Helge Avlesen, Department of Informatics, UiB.

6.8 SUBROUTINE FIELDINIT

After the call to DYPGRIN from Main.f90, the subroutine FIELDINIT is called to initialize all dynamic fields of the model. That is: ETA, ETAP, U, V, W, WR, S, T, RHO, KM, KH, KQ, Q2, Q2L, WUSURF, WVSURF, PATM, WUBOT, WVBOT, SSURF, TSURF. Volume averages of S, T and RHO are computed to be used to reduce cancellation errors when computing discrete differences.

Boundary arrays for ETA, U, V, S, T, Q2 and Q2L are also defined.

6.9 SUBROUTINE NCALFA

The model may be run with the users choice of conditions at the open boundaries. Instabilities often occur at or close to these boundaries. In fjord studies we have for instance found it difficult to get fresh water fronts associated with river runoff smoothly through the open boundary at the mouth of the fjord. In such studies we have applied relaxation of the variables through FRS-zones, see Martinsen and Engedahl [24], which at least allows the production of reasonable model results in the model domain interior to the FRS-zone, see [5, 7]. Each prognostic variable, ϕ, in the zone is after each model time step updated according to

$$\phi \quad = \quad (1-\alpha)\phi_M + \alpha\phi_F,$$

where ϕ_M contains the unrelaxed values computed by the model and ϕ_F is a specified forced solution in the zone. The relaxation parameter α varies from 1 at the model boundary to 0 at the end of the zone facing the interior model domain. The quality of this flow relaxation scheme depends strongly upon the quality of the specified forced solutions, ϕ_F, chosen for the prognostic variables.

The SUBROUTINE NCALFA computes the relaxation parameters ALPHA(LB) and ALPHAE(LB), where LB is the width of the FRS-zone, to be used for 3-D variables and 2-D variables respectively.

It may be shown, see [24], that to use this boundary conditions is equivalent to adding a Newtonian friction term of the form

$$-K(\phi_M - \phi_F),$$

to the right hand side of the equations with

$$K \quad = \quad \frac{\alpha}{\Delta t(1-\alpha)},$$

where Δt is the time step used when propagating the variable ϕ. Thus the strength of Newtonian forcing is dependent on the time step, and we are thus attempting to solve a different mathematical problem when we vary the time step. We have also seen in practice that by keeping the α arrays fixed and varying the time step, different solutions may be produced. This problem is not explicitly discussed in [24].

In NCALFA we therefore first compute the α arrays with one of the formulas from [24]. The *tanh* formulation is normally used. We assume that this produces a reasonable Newtonian friction throughout the FRS-zone for the time step Δt_0. The values of K throughout the zone are computed from the above formula. Then we recompute the α arrays, one for the 3-D variables that are propagated with time steps DT and one for the 2-D variables that are propagated with time steps DT/N2D, according to

$$\alpha \quad = \quad \frac{\Delta t K}{(1+\Delta t K)}.$$

The results are placed in the model variables ALPHA and ALPHAE respectively. By doing this the Newtonian friction is unaltered as the time steps are modified and we may at least hope to demonstrate convergence in time as the time step is reduced.

NCALFA may be unchanged from one application to another, but it may be advisable to test the sensitivity of the model results both to the representation of the α arrays and to the choice of Δt_0. As a rule of thumb choose Δt_0 close to the value DX / $\sqrt{2 \times \text{GRAV} \times \text{DEPMAX}}$ where DEPMAX is the maximum depth over the model region. If the user experiences problems at or close to the boundaries, the above options may be attempted. If the quality of the forcing field ϕ_F is poor, this may also reduce the quality of the model outputs and possibly cause instabilities. See [5] for a suggestion on how to produce ϕ_F in idealized fjord studies.

6.10 SUBROUTINE BOUND

Subroutine BOUND is called to update all prognostic variables at the open boundaries. If FRS-zones are used, the method described in the section for subroutine NCALFA is used to update the solution at the open boundary. In the FRS-zones the 2-D fields ETA, UA and VA are updated for each 2-D time step using the relaxation parameters ALPHAE. The 3-D fields U, V, S, T, Q2 and Q2L are updated for each 3-D time step using the relaxation parameters ALPHA. See the description of the routine NCALFA for further details.

The FRS-zone arrays ETAFRS etc. must be modified from one application to another. The same may apply to the arrays ALPHA and ALPHAE defined in NCALFA.

BOUND calls the subroutine RELAX if FRS-zones are used.

BOUND may have to be redefined for new model areas.

6.10.1 SUBROUTINE RELAX

RELAX updates prognostic fields in the FRS-zones of the model with the method described in the section for subroutine NCALFA. RELAX is written such that the relaxation in general is applied in all LB-cell wide FRS-zones against the west, south, east and north boundaries of the model domain. We want the relaxation to take place only where the model is connected to the open ocean through an open boundary. To avoid relaxation in parts of the model domain that are inside the general FRS-zone and not close to an open boundary, one may comment away those statements in the general relax.f90 that are not appropriate for the given model domain. This is the technique applied in the application for Osterøy.

RELAX will have to be redefined for new model areas.

6.11 Subroutines for tidal forcing

If the user want to include tidal forcing, the logical variable TIDE must be defined as .TRUE. in setupfile.dat. The number of tidal components, KCON, must be set in mod_bound.f90. The frequency array FREQ of dimension KCON and the amplitude and phase arrays ETATAMP, ETATPHA, UATAMP, UATPHA, VATAMP, VATPHA all of dimension (IM,JM,KCON) must be declared in mod_bound.f90.

To set up the tidal forcing in order to estimate the tidal effects at specific times, requires precise information about the phase lags and there are factors related to the lunar orbit that may have to be considered. For more idealized studies more simplistic forcing may suffice and the tidal routines applied for the Osterøy case are described below.

The routines TIDEINIT and TIDEFRS will have to be modified in new applications.

6.11.1 SUBROUTINE TIDEINIT

In TIDEINIT the frequencies of the tidal components and all amplitudes and phases of the water level and the depth averaged velocities must be given. The values are to be used only over the FRS-zones of the model.

6.11.2 SUBROUTINE TIDEFRS

TIDEFRS updates the field ETATFRS, UATFRS, VATFRS, UTFRS and VTFRS to be used to force the ETA, UA, VA, U and V fields in the FRS-zones, as functions of the time, the frequencies, the amplitudes and phases for each 3-D time step.

At present we have no data on the amplitudes and phases of the tidal velocities at the open boundaries.

The response of the model is very dependent of the forcing and improved forcing is necessary to get agreement with observations from this fjord system.

6.12 River runoff

If the user wants to include runoff from rivers, the river variables described in the appendix must be declared in mod_bound.f90, and the number of river outlets NUMC must be set in mod_bound.f90. River runoff may be turned off or on by changing the value of the logical variable RIVLOG to be read from setupfile.dat.

For all NUMC coastal cells having entering rivers the indices of the cells, codes indicating the direction, the temperature of the river water and the fluxes must be defined in the subroutine rivers.f90. The fluxes and the temperature may be time dependent and SUBROUTINE RIVERS is therefore called every 3-D time step if RIVLOG is .TRUE.. Outflow/inflow from/to semi-enclosed basins like the Baltic may be treated by utilizing the river runoff capabilities of the model. Therefore, also the salinity may be specified non equal to zero.

IF RIVLOG is .TRUE., the effects of river runoff on the momentum are included in laxwuv.f90, the effects on the water level and the depth integrated velocities UA and VA are included in modesplit.f90. The effects on the temperature and the salinity fields are included in superbeef.f90.

The routine rivers.f90 must be redefined from one application to another.

6.13 Atmospheric forcing

To capture the effects of wind and atmospheric pressure correctly in time and space requires input of real atmospheric forcing. For more idealized studies it may suffice to force with idealized winds and/or atmospheric pressure. WIND must be defined as .TRUE. in setupfile.dat and WSPEEDU and WSPEEDV must be declared in mod_bound.f90 if effects of wind forcing are to be included. ATMPRESS must be defined as .TRUE. if effects of atmospheric pressure gradients shall be included.

6.13.1 SUBROUTINE WINDFORCING

In windforcing.f90, the wind speeds 10m above sea level, WSPEEDU and WSPEEDV, in X-direction and Y-direction respectively, must be given. From these the wind stresses, WUSURF and WVSURF, are computed. They are here given as quadratic functions of the wind speeds for wind speeds less than 11 ms^{-1}. For larger speeds they increase cubicly. WUSURF and WVSURF are used in the routine vertviscuvb.f90 as a boundary condition in the vertical mixing.

The atmospheric pressure PATM may also be given in WINDFORCING.

The wind speed part of windforcing.f90 and possibly the atmospheric pressure must be redefined in new applications.

6.13.2 SUBROUTINE ATMOSP

If ATMPRESS is defined as .TRUE., ATMOSP is called to define the atmospheric pressure, PATM, and to compute the effects of the gradients in this pressure on the velocities. An explicit method is applied to propagate the velocities in time.

If there are gradients in the atmospheric pressure along the open boundaries, this may also affect the water levels accordingly. Water levels that balance these pressure gradients are therefore estimated in ETABFRS and later on added to the corresponding contributions from tides, ETATFRS.

For areas that are much smaller than the extension of typical high/low pressures in the atmosphere, the effects of gradients in PATM may often be neglected by choosing ATMPRESS = .FALSE.

7 Utilities

In addition to the subroutines that must be called to define the forcing and to solve the equations, there are other routines that may be called to diagnose the solution, to drift particles and to perform other tasks with basis in the core model fields. Some such routines are listed below.

7.1 SUBROUTINE POSTID4

POSTID4 may be used to update the model time clock NDATE each hour.

7.2 SUBROUTINE DIAGNOSE

There is always the possibility that instabilities in the model occur. When growing modes grow for too long, the model either stops (core dumped) or starts to produce variables with value NaN or Inf. In either case it may be difficult to locate the place in time and space where the instability first occurred and to understand what created it. It may therefore be helpful to have a routine for diagnosing some of the major fields for abnormal values and in the case that some values seem far from the expected values, warning messages and information may be written out. It may also be advisable to stop the computations if abnormal values are encountered.

This subroutine may often have to be modified. In the development phase of the code it may be advisable to call DIAGNOSE after calls to each of the routines updating the U, V, S and T fields for instance. After the development reaches a stage where all values have proved to be credible, one may save some computer time by calling DIAGNOSE only after each 10th time step for instance.

7.3 SUBROUTINE OUTPUT

In this routine the user may gather all statements concerning print out of model variables or measures derived from these like transports, energies, maximum/minimum values of selected fields etc.

This routine is application specific.

7.4 SUBROUTINE PZLEVEL

PZLEVEL interpolates first the U and V points horizontally to S-points. Then the S, T, U, V and WR fields are interpolated to a z-level specified by the input variable ZLEVEL and stored in 2-D work arrays. The interpolation in the vertical is linear. If the z-level for a given horizontal location is deeper than the depth of the ocean, then the fields are assigned the value SPV to indicate that this is not an in ocean location.

Other users may want to modify this routine to their needs.

7.5 SUBROUTINE NRG3D

NRG3D may be called to compute the total kinetic energy in the model domain from the U and V fields, the total volume and mass and the total potential energy from the ETA field. Variants of NRG3D that computes these measures for sub-domains may also be useful.

7.6 Plotting routines

The subroutines outbottomfem.f90, add1.f90, outsections.f90, sectiout.f90, outetafem.f90 and addeta.f90 are used to produce and store data for plotting with the visualization software TECPLOT. Users with different plotting packages may have to redefine these output routines for visualization purposes.

8 Time step limitations

When instabilities occur in a model experiment, the reason may be that either the 3-D time step, DT, or the 2-D time step, DTE = DT/N2D, is too long. Below we have tried to summarize some of the time step restrictions that any user of the code should have in mind:

1. The CFL-criterion for the shallow water equations.
The Courant number for the 2-D steps is

$$C_o = \mathrm{DTE} \times \frac{\sqrt{2 \times \mathrm{GRAV} \times \mathrm{DEPMAX}}}{\mathrm{DX}}$$

where DEPMAX is the maximum depth of an ocean cell. If C_o is larger than unity, instabilities will occur. This is a criterion that is easy to check, and should be included in Main.f90 after the call to DYPGRIN where DEPMAX is computed. The computations should be stopped if this criterion is violated.

2. The CFL-criterion for internal waves.
Instabilities may also occur because the 3-D time step is too large to resolve the propagation of internal waves. The Courant number for these waves is

$$C_{oi} = \mathrm{DT} \times \frac{\sqrt{2 \times g' \times \mathrm{DEPMAX}}}{\mathrm{DX}}$$

where g' is the reduced gravity. The problem is that the reduced gravity is developing dynamically and may be hard to predict. For a simple 2-layer system

$g' = g|\rho_1 - \rho_2|/\rho_2$ where ρ_1 is the density of the upper layer and ρ_2 the density of the lower level. Since ρ_1 may become very small in some of our fjord studies, C_{oi} may become larger than unity for large values of DT.

If the user believes that instabilities occur because this criterion is violated, then this hypothesis may easily be tested by reducing DT.

As a rule of thumb it is normal to apply 3-D time steps that are 30 times larger than the 2-D time steps. That is N2D equal to 30.

3. CFL-criterions for 3-D advection.
The advection of momentum and scalar fields taking place in SUPERBEEUV, MODESPLIT and SUPERBEEF is performed with explicit methods and the following limitations on the time step apply

$$\begin{aligned}
\mathrm{DT} &\leq max_{\forall I,J,K} \frac{\mathrm{DX}}{|\mathrm{U(I,J,K}|} \quad , \\
\mathrm{DT} &\leq max_{\forall I,J,K} \frac{\mathrm{DY}}{|\mathrm{V(I,J,K}|} \quad , \\
\mathrm{DT} &\leq max_{\forall I,J,K} \frac{\mathrm{DZ(K)} \times \mathrm{D(I,J)}}{|\mathrm{W(I,J,K}|} \quad .
\end{aligned}$$

For larger values of DX and DY the two first criterions are normally satisfied. For experiments with thin layers, small values of DZ, and up/down welling, larger values of W, the third criterion may be violated.

Tests on whether these criterions are satisfied may be built into diagnose.f90 if the user believes that they may be violated.

4. Time step limitations due to horizontal viscosity and diffusivity.
Horizontal viscosity and diffusivity are treated with explicit methods in time. To avoid oscillations and instabilities the following limitations on the time steps apply

$$\begin{aligned}
\mathrm{DT} &\leq max_{\forall I,J,K} \frac{1}{16} \frac{\mathrm{DX}^2}{\mathrm{AM(I,J,K)}} \quad , \\
\mathrm{DT} &\leq max_{\forall I,J,K} \frac{1}{16} \frac{\mathrm{DX}^2}{\mathrm{AH(I,J,K)}} \quad , \\
\mathrm{DTE} &\leq max_{\forall I,J} \frac{1}{16} \frac{\mathrm{DX}^2}{\mathrm{AM2D(I,J)}} \quad .
\end{aligned}$$

With the present versions of smagor.f90 and smagor2d.f90, upper limits on AM, AH and AM2D are given as functions of DT and DTE such that these inequalities are always satisfied.

If a user want to run the model with other values of AM, AH or AM2D than those prescribed by the present versions of smagor.f90 and smagor2d.f90, the criterions above must be checked.

5. Time step limitations due to vertical viscosity and diffusivity.
There are no time step limitations due to vertical diffusion and viscosity because the effects of these terms are treated with implicit methods in time.

NOTE: Even if all the above criterions are satisfied, there may still be noise in the numerical results due to unresolved physics. It is therefore advisable to dump values of for instance ETA and U and V at the surface, at a point where large variability may be expected, each time step. By studying such time series one may see whether there are 2DT noise in the solution or not. If such noise is evident, one may either reduce the time step or enhance the viscosities.

8.1 What if instabilities still occur?

If all the time step criterions are checked, and instabilities still occur it may often help to enhance the viscosity by increasing carefully the values of CM and CM2D specified in setupfile.dat. The same applies to CH, but too large values of these parameters may destroy the quality of the solutions.

For smaller values of Δx it may not be enough to increase CM and CM2D to avoid noise. To choose SMAG = .FALSE., and to set A_M and A_{M2D} as a fraction of their upper limits may then be an useful option. See the comments on smagor.f90 and smagor2d.f90.

If it is $2\Delta x$ noise originating from a rough bottom matrix that creates the instabilities, one may try to increase the value of NFILTER in setupfile.dat.

If the instabilities appear close to an open boundary, it may help to experiment with the formulation for computing ALPHA and ALPHAE in ncalfa.f90 and/or try to modify the parameter DTE0 in this routine. It may also help to try to improve the quality of one or more of the external forcing fields that are used in the FRS-zones. A possibility is also to choose Dirichlet and/or Neumann conditions for some of the variables at open boundaries as in the Osterøy case.

9 Numerical simulation of the fjord system around Osterøy

To make the code available to other users it is set up for the fjord system around Osterøy close to Bergen, Norway. The model area contains of 33 × 55 cells in the horizontal and 32 layers vertically focused towards the surface layer. In the horizontal the resolution is 1000m. The model may be forced by idealized wind, tide and idealized river fluxes which are much larger than in the normal situation. (7 rivers, each with a flux of 200 $m^3 s^{-1}$ enters the fjord system).

The model area is given in Figure 4.

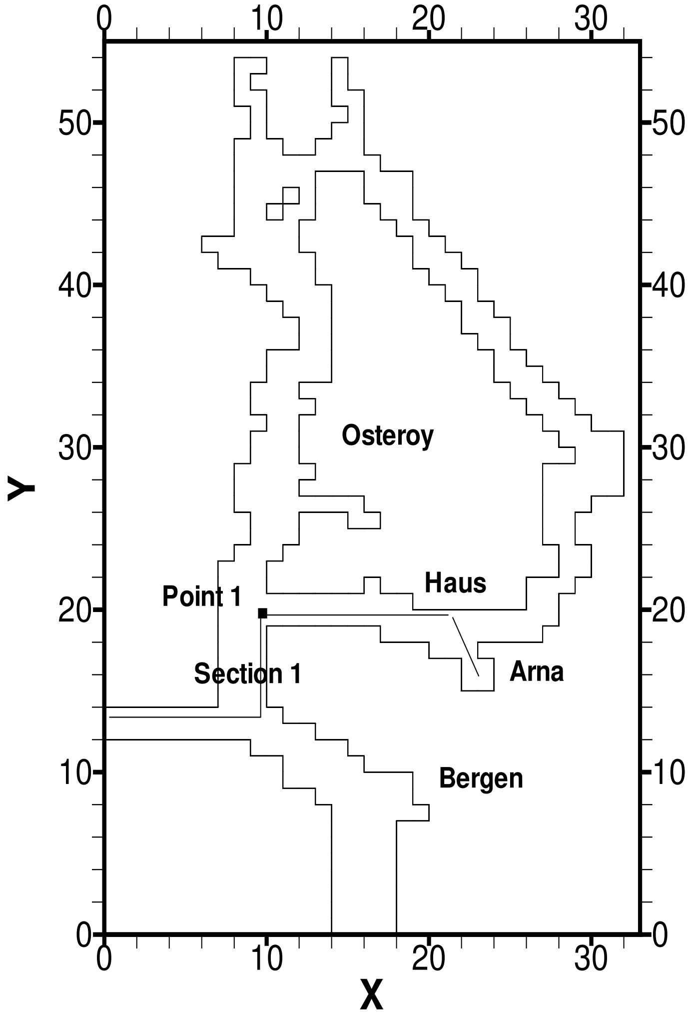

Figure 4. The Osterøy model area. The section from the open boundary to the west to Arna is indicated. The point where the time series of water elevation, and surface velocities is taken, is also indicated (Point 1).

The flow in the fjord is forced with 4 components of the tide. Estimates of the amplitudes of the water levels of these 4 components are used to define the forcing water levels at the open boundaries. The external velocities in the FRS-zones are assumed to be zero, and FRS-sone relaxation is used for all prognostic variables in the tidally forced case with LB = 7. The results given below are produced with the following choices of model variables:

THETA2D	0.5
THETA3D	0.5
SMAG	.TRUE.
CM	2.5
CM2D	2.5
CH	0.0
UNESCO	.TRUE.
VISCPOM	.TRUE.
ADVTURB	.FALSE.
RIVLOG	.FALSE.
TIDE	.TRUE.
WIND	.FALSE.
ATMPRESS	.FALSE.
FOURTH	.FALSE.
DEPMIN	3.0m
NFILTER	4
NDIVIS	30
N2D	32
KMMIN	$1 \times 10^{-5} m^2 s^{-1}$
KHMIN	$1 \times 10^{-7} m^2 s^{-1}$

In Figure 5 the volume averaged kinetic energies and the volume averaged potential energies in Jm^{-3} are given as function of time for a 4 days experiment. In the beginning water is flowing out of the fjord system and the asymmetry between the potential energies on inflowing tide and outflowing tide is very noticeable.

In Figure 6 the water level in m and the two components of the surface velocities in ms^{-1} at Point 1 are given as a function of time.

This is not considered to be a realistic computation of the tidal flow in this fjord system. More realistic values of the water levels and the velocities at the mouths of the fjord will be necessary to drive the system and measurements are also needed to evaluate the quality of the models response. Ideally the open boundary should also be moved to the edge of the continental shelf, see Garrett and Greenberg [19], but this would be very time consuming. Here we have only tried to demonstrate the models response to the boundary conditions for the tide prescribed.

We have also performed an experiment where TIDE = .FALSE. , RIVLOG =.TRUE. . To avoid 2DT noise it was necessary to enhance CM and CM2D to 5. in this study. All other values in the table above were unchanged. In this experiment with river forcing a Neumann condition is applied to all prognostic variables at the open boundaries in order to let the lighter surface water get through the open boundaries.

A vertical section of the density field along section 1 indicated in Figure 4 after 240 hours simulation is given in Figure 7. Contours and some values of the density field at 2 meter depth throughout the fjords are given in Figure 8. From figures 7 and 8 we notice how the fresh surface layer of the fjord has developed over the 10 days with substantial fresh water runoff.

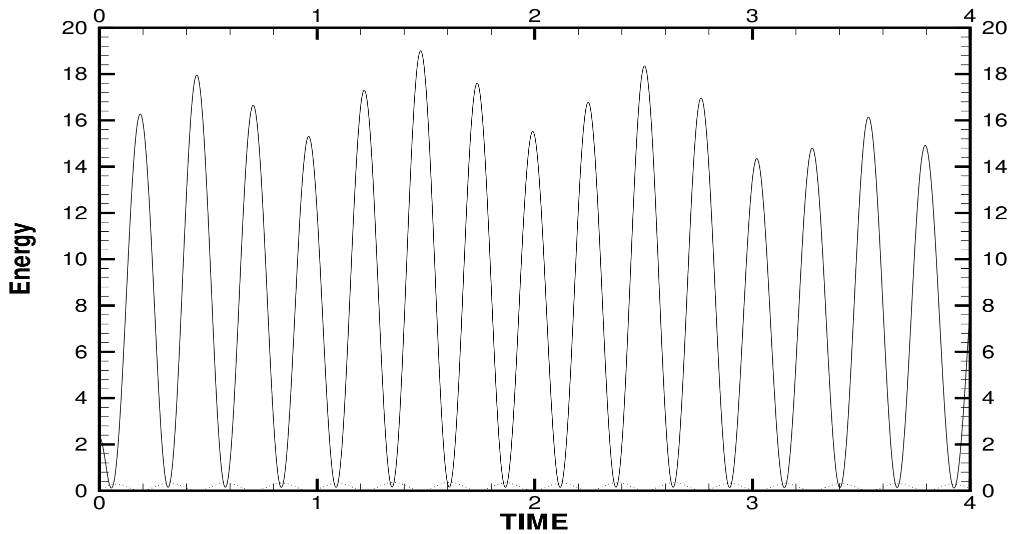

Figure 5. Volume averaged kinetic energy (dotted line) and volume averaged potential energy (solid line) as a function of time in days. Both in Jm^{-3}.

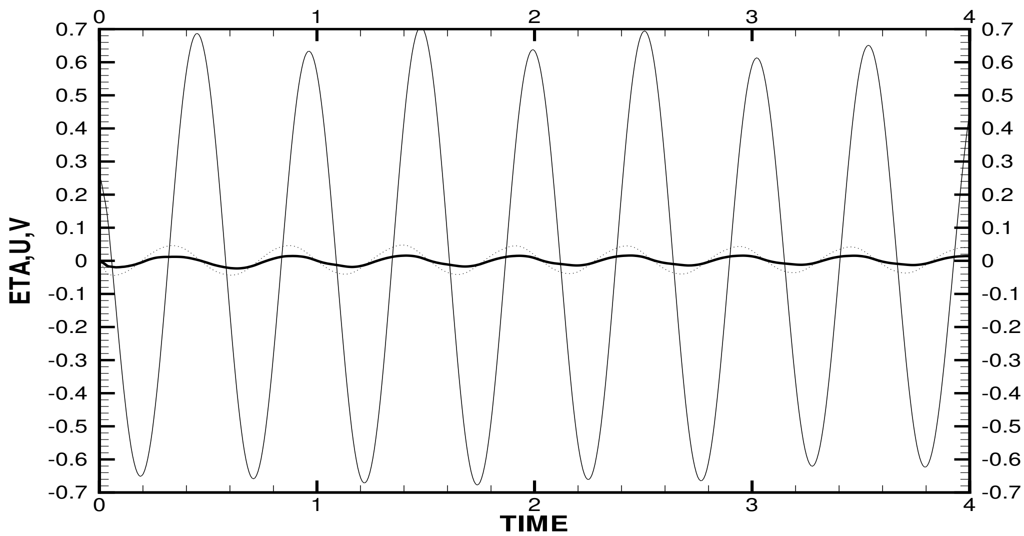

Figure 6. ETA(10,20) (solid line) in m, U(10,20,1) (solid thick line) and V(10,20,1) (dotted line) in ms^{-1} as a function of time in the tidally forced experiment.

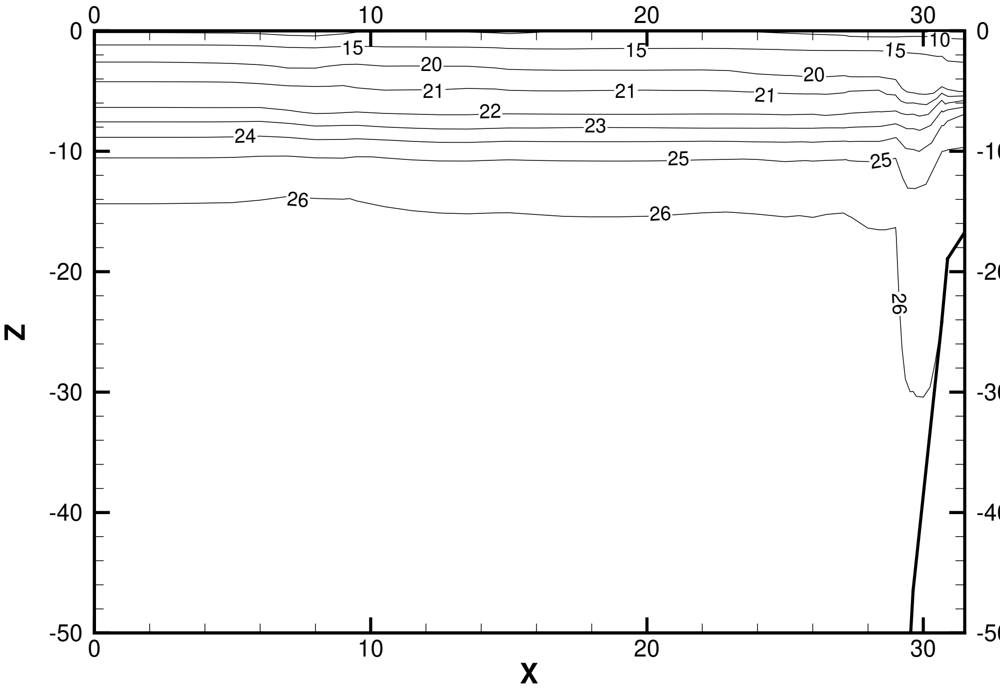

Figure 7. Vertical profile of the density field (-1000kgm^{-3})along section 1 after 240 hours of simulation with TIDE = .FALSE. and RIVLOG =.TRUE.. The unit along the horizontal axis is km from the open boundary and the unit along the vertical axis is depth.

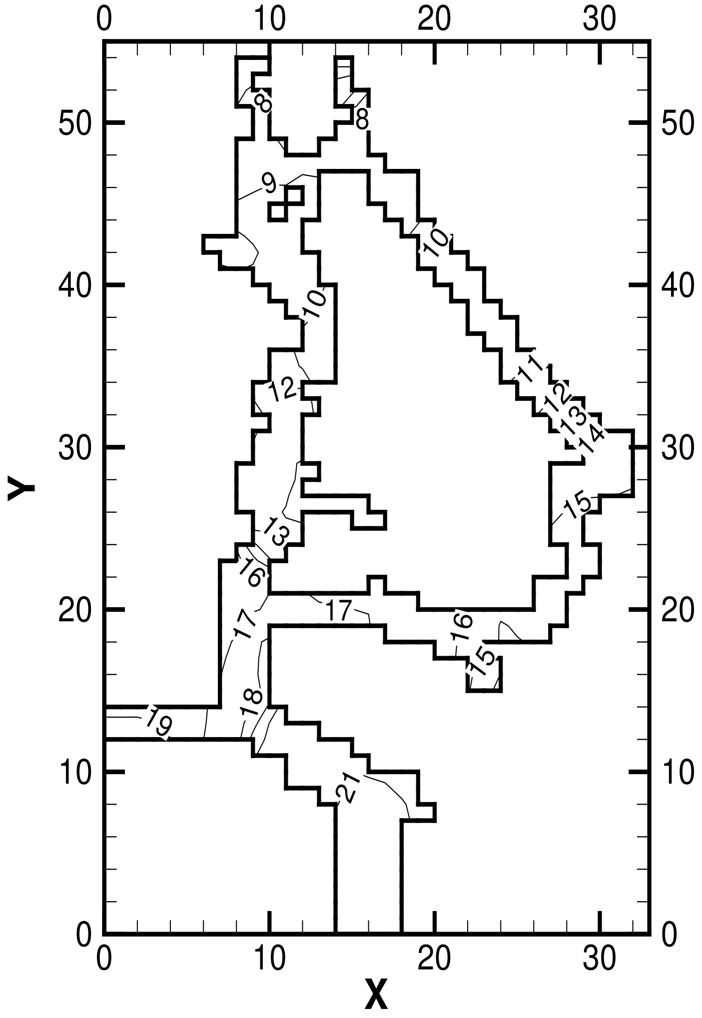

Figure 8. Densities ($-1000kgm^{-3}$) at 2 meter depth after 240 hours. The contour levels are 0(1)25 kgm^{-3}.

After unpacking the bom.tar file and recompiling it, it is advisable to run the model 30 time steps as given in setupfile.dat and compare the output with the output produced by a PC running Linux and using the Portland Group compiler pgf90 and given in the file: outbom010201 Also compare with energies in file energy.plt and point values of ETA, U and V in files point.plt, point2.plt, point3.plt. There may be minor differences on other machines due to different computer arithmetic.

If the outputs from this 1 hour experiment is consistent with those given in outbom010201 and the *.plt files, then it may be a good idea to first run the model 4 days to check consistency with figures 5 and 6. After that set TIDE = .FALSE., RIVLOG = .TRUE., CM = CM2D = 5. and introduce Neumann boundary conditions in routine bound.f90 and run the model for 10 days to check consistency with figures 7 and 8.

After that you may start to play around with the forcing and/or parameter values and/or boundary conditions as you like.

The algoritmic changes from verson 1.2 to 2.0, Coriolis term averaging, UNESCO formulation for density, and the POM form on the viscosity terms, only had very minor impact on the results for the Osteroy case which could be expected. The change in time stepping methods from version 2.0 to 3.0, had only minor impact on the results. This is as expected since the solution is strongly forced and also strongly influenced be the special fjord topography. To set FOURTH = .TRUE., changes the stratification clearly. The fresh water surface layer get fresher. However, undershooting (S less than 0) will occur after some time in the strongly fresh water driven case.

10 Acknowledgements

Input of codes, data and advice from the numerical oceanography groups at the Institute of Marine Research, Bergen, Norway, and the Norwegian Meteorological Institute, Oslo, Norway is greatly appreciated. Co-work and discussions with colleagues at the University of Bergen has also influenced the directions the work with this ocean model has taken. Special thanks to Helge Avlesen and Lars Asplin for useful comments. Inge Eliassen has contributed to this work both with remarks on the code and by making it available on www. The Research Council of Norway has supported the development of this code both through a strategic university program and through grants for computer time (Programme for supercomputing). Trygve Sagberg is thanked for providing the bottom matrix for the Osterøy implementation. This version of the user manual is written during a sabbatical stay at Thayer School of Engineering, Dartmouth College and the hospitality of the staff at Thayer School is greatly appreciated. Yngve Heggelund is thanked for finding the ETA = ETAP bug in Main.f90. Helge Avlesen has introduced a number of OPEN MP directives and rewritten several DO-loops to make the code more efficient on shared memory parallel computers.

A List of symbols

$\vec{U} = (U, V)$	Horizontal velocities in $x-$ and y-direction respectively
W	Vertical velocity in the z-coordinate system
ω	Vertical velocity in the σ-coordinate system
η	Surface elevation
H	Bottom static depth
D	Bottom dynamic depth $(H + \eta)$
P	Pressure
P_{atm}	Atmospheric pressure
T	Temperature
S	Salinity
ρ	In situ density
K_M	Vertical eddy viscosity
A_M	Horizontal eddy viscosity
C_M	Dimensionless horizontal eddy viscosity coefficient
K_H	Vertical eddy diffusivity
A_H	Horizontal eddy diffusivity
C_H	Dimensionless horizontal eddy diffusivity coefficient
$q^2/2$	Turbulent kinetic energy
l	Turbulent macro scale
ρ_0	Reference density
g	Gravity
f	The Coriolis parameter
$\vec{\tau}_0 = (\tau_{0x}, \tau_{0y})$	Surface wind stress
$\vec{\tau}_b = (\tau_{bx}, \tau_{by})$	Bottom stress
$\dot{T}_0$	The surface heat flux
$\dot{S}_0$	The net precipitation/evaporation at the surface
$\vec{U}_b = (U_b, V_b)$	Horizontal velocities at the bottom
W_0	Vertical velocity at the surface (z-coordinate)
W_b	Vertical velocity at the bottom (z-coordinate)
C_D	Bottom drag coefficient
κ	The von Karman constant
z_0	Bottom roughness parameter

B FORTRAN 90 variables defined in MODULE STATE

The main variables of the code are defined in MODULE STATE. In the following table a description of the variables is given.

IM	Number of grid cells in x-direction
JM	Number of grid cells in y-direction
KB	Number of grid cell interfaces vertically
NUMPT	Maximum number of ocean cells to be used when looping over wet-points only
NUMUP	Number of ocean UA-points
NUMVP	Number of ocean VA-points
NUMCP	Number of ocean ETA-points
NUMCP2	Number of ocean ETA-points to be used in MY2HALV
INDU	Integer array of dimension (2,NUMPT) for indices of ocean UA-points
INDV	Integer array of dimension (2,NUMPT) for indices of ocean VA-points
INDC	Integer array of dimension (2,NUMPT) for indices of ocean ETA-points
INDC2	Integer array of dimension (2,NUMPT) for indices of ocean ETA-points used in MY2HALV
NDIVIS	Number of 3-D time steps per hour
N2D	Number of 2-D time steps per 3-D time step
THETA3D	The degree of implicitness in the 3D steps
THETA2D	The degree of implicitness in the 2D steps
DX	The grid spacing in x-direction (m)
DY	The grid spacing in y-direction (m)
DT	The model time step (s)
GRAV	Gravity (ms^{-2})
RHO0	Reference Density
PMEAN	Mean atmospheric pressure
CM	The horizontal viscosity parameter
CM2D	The horizontal 2D viscosity parameter
CH	The horizontal diffusivity parameter
CD	Minimum value of the bottom drag coefficient
Z0	Bottom roughness parameter
ROAIR	Density of air (kg m^{-3})
SPV	Special Value. Used when masking out land cells.
RHOAVER	Average density over the ocean cells. (kg m^{-3})
SAVER	Average salinity over the ocean cells. (p.s.u)
TAVER	Average temperature over the ocean cells. (degree Celcius)
KMMIN	Minimum allowed value on vertical viscosity (m^2s^{-1})
KHMIN	Minimum allowed value on vertical diffusivity (m^2s^{-1})
ADVTURB	Logical indicating whether advection of Q2 and Q2L is on or off
SMAG	Logical indicating whether or not the Smagorinsky formulation shall be used to compute viscosities and diffusivities
VISCPOM	Logical indicating whether the POM form or a simple form of the viscosity terms are to be used
UNESCO	Logical indicating whether the UNESCO or the Wang equation of state is to be used
FOURTH	Logical indicating whether the TVD scheme of Roe and Sweby or the 4th order method of Shchepetkin and McWilliams is to be used

1-D arrays of dimension KB

Z	The σ-coordinates of cell interfaces
ZZ	The σ-coordinates at cell centers (ZZ(K) = (Z(K)+Z(K+1))/2)
DZ	Thickness in σ-coordinates of cells (DZ(K)= Z(K)-Z(K+1))
DZZ	Distance in σ-coordinates between cell centers (DZZ(K) = ZZ(K) - ZZ(K+1))
DZR	1/DZ

2-D arrays of dimension (IM,JM)

COR	The Coriolis parameter (s^{-1})
FSM	Mask array for cell-centered variables FSM = 0 in land points FSM = 1 in ocean points
DUM	Mask array for variables defined in U-points DUM = 0 in land points DUM = 1 in ocean points
DVM	Mask array for variables defined in V-points DVM = 0 in land points DVM = 1 in ocean points
ETA	The water level (m)
ETAP	The water level at the previous time step (m)
H	Static depth in ETA points (m)
HU	Static depth in U points (m)
HV	Static depth in V points (m)
D	Dynamic depth in ETA points (D = H + ETA) (m)
DU	Dynamic depth in U points (m)
DV	Dynamic depth in V points (m)
WUSURF	Momentum flux in x-direction at the surface (m^2s^{-2})
WVSURF	Momentum flux in y-direction at the surface (m^2s^{-2})
WUBOT	Momentum flux in x-direction at the bottom (m^2s^{-2})
WVBOT	Momentum flux in y-direction at the bottom (m^2s^{-2})
SSURF	Salinity flux at the surface (p.s.u. ms^{-1})
TSURF	Heat flux at the surface (degree Celcius ms^{-1})
PATM	Atmospheric pressure
CBC	Bottom drag coefficients

AM2D	The 2-D horizontal viscosities (m^2s^{-1})
UA	The vertically averaged velocities in x-direction (ms^{-1})
VA	The vertically averaged velocities in y-direction (ms^{-1})
HUSQR	Weights for Coriolis averaging $\sqrt{}$(HU)
HVSQR	Weights for Coriolis averaging $\sqrt{}$(HV)
DELETAE	Changes in ETA over an explicit 3D step
DELETAI	Changes in ETA over an implicit 3D step

3-D arrays of dimension (IM,JM,KB)

U	Horizontal velocity in x-direction (ms^{-1})
V	Horizontal velocity in y-direction (ms^{-1})
W	σ-coordinate vertical velocity (ms^{-1})
WR	z-coordinate vertical velocity (ms^{-1})
UADV	Horizontal velocity in x-direction used for advection (ms^{-1})
VADV	Horizontal velocity in y-direction used for advection (ms^{-1}) The fields UADV, VADV, W, ETA and ETAP satisfy the equation of continuity
S	Salinity (p.s.u.)
T	Temperature (Degree Celcius)
RHO	Density (kgm^{-3})
AM	Horizontal viscosity (m^2s^{-1})
AH	Horizontal diffusivity (m^2s^{-1})
KM	Vertical viscosity (m^2s^{-1})
KH	Vertical diffusivity (m^2s^{-1})
KQ	Vertical viscosity used for mixing of Q2 and Q2L (m^2s^{-1})
Q2	q^2, turbulent kinetic energy (m^2s^{-2})
Q2L	q^2l, turbulent kinetic energy times length scale (m^3s^{-2})
DDZ	Dynamic thickness of a cell in a S or T point (m)
DUDZ	Dynamic thickness of a cell in a U point (m)
DVDZ	Dynamic thickness of a cell in a V point (m)
DELTAU	Contributions from the internal pressure and advection terms to the U-momentum equation (m^2s^{-2})
DELTAV	Contributions from the internal pressure and advection terms to the V-momentum equation (m^2s^{-2})

UPREV	U values from time step n.
VPREV	V values from time step n.
SPREV	S values from time step n.
TPREV	T values from time step n.
DELUE	Changes to U in the explicit predictor 3D step.
DELVE	Changes to V in the explicit predictor 3D step.
DELSE	Changes to S in the explicit predictor 3D step.
DELTE	Changes to T in the explicit predictor 3D step.
DELUI	Changes to U in the corrector 3D step.
DELVI	Changes to V in the corrector 3D step.
DELSI	Changes to S in the corrector 3D step.
DELTI	Changes to T in the corrector 3D step.
WE	σ-coordinate vertical velocity from the predictor step.
WI	σ-coordinate vertical velocity from the corrector step.

The variables ADVTURB, SMAG, VISCPOM, UNESCO, FOURTH, THETA3D, THETA2D, NDIVIS, N2D, DX, CM, CM2D, CH, KMMIN and KHMIN will be read from setupfile.dat.

C FORTRAN 90 variables defined in MODULE MOD_BOUND

The boundary variables of the code are defined in MODULE MOD_BOUND. In the following table a description of the variables associated with FRS-zones is given.

LB	Number of grid cells in the FRS-zone

1-D arrays of dimension LB

ALPHA	The relaxation parameters α to be used for 3-D variables
ALPHAE	The relaxation parameters α to be used for 2-D variables

2-D arrays of dimension (IM,JM)

UAFRS	The forced FRS-zone values for UA
UATFRS	The forced FRS-zone values due to tides for UA
VAFRS	The forced FRS-zone values for VA
VATFRS	The forced FRS-zone values due to tides for VA
ETAFRS	The forced FRS-zone values for ETA
ETATFRS	The forced FRS-zone values due to tides for ETA
ETABFRS	The atmospheric forced FRS-zone values for ETA

3-D arrays of dimension (IM,JM,KB)

UFRS	The forced FRS-zone values for U
UTFRS	The forced FRS-zone values due to tides for U
VFRS	The forced FRS-zone values for V
VTFRS	The forced FRS-zone values due to tides for V
SFRS	The forced FRS-zone values for S
TFRS	The forced FRS-zone values for T
Q2FRS	The forced FRS-zone values for Q2
Q2LFRS	The forced FRS-zone values for Q2L

If we are running the model with river runoff, the following variables are used:

RIVLOG	Logical indicating whether river forcing is .TRUE. or .FALSE.
NUMC	Number of coastal ocean cells with rivers entering the cell

1-D INTEGER arrays of dimension NUMC

IRIVER	Index in x-direction of the coastal cell
JRIVER	Index in y-direction of the coastal cell
KRIVER1	Index in σ-direction of the top cell with entering river runoff. KRIVER1 is normally 1, that is the river water enters the top coastal cell. KRIVER1 may be selected as larger than 1, when simulating the effects of submerged river runoff.
KRIVER2	Index in σ-direction of the deepest cell with entering river water.
CODERIVER	Code that indicates from which side of the cell the river water enters. CODERIVER = 1 if river runoff from west. CODERIVER = 2 if river runoff from south. CODERIVER = 3 if river runoff from east. CODERIVER = 4 if river runoff from north.

1-D REAL arrays of dimension NUMC

SRIVER	The salinity of the entering river water. SRIVER is for rivers 0. The exchange of water between the model domain and semi-enclosed basins like the Baltic, may be modelled by using the models river capabilities. For such cases SRIVER not equal to 0 may be used.
TRIVER	The temperature of the entering river water.
FRIVER	The flux in $m^3 s^{-1}$ of the entering river water.
DZRIVER	The fraction in the vertical of the coastal cell covered be the depth of the river.

If we are running the model with tidal forcing, the following variables are used:

TIDE	Logical indicating whether tidal forcing is .TRUE. or .FALSE.
KCON	Number of tidal constituents.

1-D REAL array of dimension KCON:

FREQ	The frequencies of the constituents.

3-D REAL arrays of dimension (IM,JM,KCON):

ETATAMP	The amplitudes of the water level associated with the constituents.
ETATPHA	The corresponding phase.
UATAMP	The amplitudes of the depth averaged U-velocities.
UATPHA	The corresponding phase.
VATAMP	The amplitudes of the depth averaged V-velocities.
VATPHA	The corresponding phase.

If we are running the model with atmospheric forcing, the following variables are used:

WIND	Logical indicating whether the effects of wind stress is on or off
ATMPRESS	Logical indicating whether the effects of gradients in atmospheric pressure is on or off

2-D REAL arrays of dimension (IM,JM):

WSPEEDU	Wind speeds in X-direction 10m above the sea level
WSPEEDV	Wind speeds in Y-direction 10m above the sea level

Values for the variables RIVLOG, TIDE, WIND and ATMPRESS are read from setupfiles.dat.

D FORTRAN 90 variables defined in MODULE MOD_MAIN

Other variables used by PROGRAM MAIN and some of the utility subroutines are defined in mod_main.f90. The variables to be declared may vary from one application to the next. Below follows a description of some of the variables normally declared in MODULE MOD_MAIN.

Integers

NUMSTEP	The total number of 3-D time steps in the main do-loop
ISTEP	Counter in the main do-loop
NFILTER	The number of times the depth matrix shall be filtered
IYEAR	Year
IMONTH	Month
IDAY	Day
NDATE	Array of dimension 5 that measures year, month, day, hour and hours since start
ISTART	Array of dimension 4 that gives year, month, day, hour at the start time

Reals

DEPMIN	The minimum allowed value of H in an ocean cell
DEPMAX	The maximum value of H over the model domain
DSBOT	The thickness of the bottom sigma layer
COURANT	The Courant number computed from GRAV, DEPMAX, DX, NDIVIS and N2D
TIME	Count the time since start in days
EKIN	Total kinetic energy over the model domain
VOLUME	Total volume of the ocean
MASS	Total mass of the ocean
EETA	Potential energy associated with the ETA field
ZLEVEL	Depth at a given Z-level

The variables ISTART, NFILTER, DEPMIN, DSBOT and NUMSTEP are read from setupfile.dat.

References

[1] B. Ådlandsvik and H. Engedahl. Documentation of a three-dimensional baroclinic sea-model. Technical Report 16/1991/HSM, Institute of Marine Research, Bergen, NORWAY, 1991.

[2] L. Asplin, A.G.V. Salvanes, and J.B. Kristoffersen. Non-local wind-driven fjord-coast advection and its potential effect on pelagic organisms and fish recruitment, 1998. Accepted by Fisheries Oceanography.

[3] H. Berntsen, Z. Kowalik, S. Sælid, and K. Sørli. Efficient numerical simulation of ocean hydrodynamics by a splitting procedure. *Modeling, Identifaction and Control*, 2:181–199, 1981.

[4] J. Berntsen. A sensitivity study of a baroclinic model for the North Sea with focus on the Utsira-Orkneys transect. Technical Report Fisken og Havet Nr. 16, Institute of Marine Research, 1998.

[5] J. Berntsen, D.L. Aksnes, and A. Foldvik. Fresh water driven primary production in a fjord. Technical Report 109, Dept. of Applied Mathematics, University of Bergen, 1997.

[6] J. Berntsen, M.D. Skogen, and T.O. Espelid. Description of a σ-coordinate ocean model. Technical Report Fisken og Havet Nr. 12, Institute of Marine Research, 1996.

[7] J. Berntsen and E. Svendsen. Using the Skagex dataset for evaluation of ocean model skills. *Journal of Marine Systems*, 18:313–331, 1999.

[8] J. Berntsen, E. Svendsen, and M. Ostrowski. Validation and sensitivity study of a sigma-coordinate ocean model using the Skagex dataset, 1996. ICES CM 1996/C:5.

[9] A.F. Blumberg and G.L. Mellor. A description of a three-dimensional coastal ocean circulation model. In N.S. Heaps, editor, *Three-Dimensional Coastal Ocean Models, Vol.4*, volume 4 of *Coastal and Estuarine Series*, page 208. American Geophysical Union, 1987.

[10] P.C. Chu and C. Fan. Sixth-order difference scheme for sigma coordinate ocean models. *J.Phys.Oceanogr.*, 27:2064–2071, 1997.

[11] A.M. Davies, P.J. Luyten, and E. Deleersnijder. Turbulence energy models in shallow sea oceanography. In D.R. Lynch and A.M. Davies, editors, *Quantitative Skill Assessment for Coastal Ocean Models*. American Geophysical Union, 1995.

[12] A.M. Davies and J. Xing. An intercomparison and validation of a range of turbulence closure schemes used in three dimensional tidal models. In D.R. Lynch and A.M. Davies, editors, *Quantitative Skill Assessment for Coastal Ocean Models*. American Geophysical Union, 1995.

[13] B.I. Dybern, D.S. Danielssen, L. Hernroth, and E. Svendsen. The Skagerrak Experiment-Skagex report 1988-1994. Technical Report 635, TemaNord 1994:635, 1994.

[14] T. Eldevik and K.B. Dysthe. Short frontal waves: Can frontal instabilities generate small scale spiral eddies? In A. Zatsepin and A. Ostrovskii, editors, *Collected papers of the ISOFRP*. IOC, UNESCO, 1999.

[15] I. Eliassen and J. Berntsen. Effects of Enhanced Resolution in the North Sea and the Skagerrak, 1999. Manuscript, Department of Mathematics, University of Bergen.

[16] I. Eliassen and J. Berntsen. Evaluation of methods for estimating the internal pressure in sigma-coordinate ocean models using measurements from the Norwegian trench and the Skagerrak, 1999. Manuscript, Department of Mathematics, University of Bergen.

[17] T.O. Espelid, Berntsen J., and K. Barthel. Conservation of energy for schemes applied to the propagation of shallow-water inertia-gravity waves in regions with varying depth. *International Journal for Numerical Methods in Engineering*, 49:1521–1545, 2000.

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

[19] C. Garrett and D. Greenberg. Predicting changes in tidal regime: The open boundary problem. *J. Phys. Oceanogr.*, 7:171–181, 1977.

[20] A.E. Gill. *Atmosphere-Ocean Dynamics*. Academic Press, 1982. ISBN-0-12-283520-4.

[21] R.L. Haney. On the pressure gradient force over steep topography in sigma coordinate ocean models. *J. Phys. Oceangr.*, 21:610–619, 1991.

[22] N. Kliem and J.D. Pietrzak. On the pressure gradient error in sigma coordinate ocean models: A comparison with a laboratory experiment. *Journal of Geophysical Research*, 104:29781–29799, 1999.

[23] D.R. Lynch, J.T.C. Ip, C.E. Naimie, and F.E. Werner. Convergence studies of tidally-rectified circulation on Georges Bank. In D.R. Lynch and A.M. Davies, editors, *Quantitative Skill Assessment for Coastal Ocean Models*. American Geophysical Union, 1995.

[24] E.A. Martinsen and H. Engedahl. Implementation and testing of a lateral boundary scheme as an open boundary condition in a barotropic ocean model. *Coastal Engineering*, 11:603–627, 1987.

[25] E.A. Martinsen, B. Gjevik, and L.P. Røed. A numerical model for long barotropic waves and storm surges along the western coast of Norway. *J.Phys.Oceanogr.*, 9:1126–1138, 1979.

[26] J.D. McCalpin. A comparison of second-order and fourth-order pressure gradient algorithms in a σ-co-ordinate ocean model. *Int.J.Numer.Methods in Fluids*, 18:361–383, 1994.

[27] G.L. Mellor. Users guide for a three-dimensional, primitive equation, numerical ocean model. Technical report, Princeton University, 1996.

[28] G.L. Mellor and A.F. Blumberg. Modelling vertical and horizontal diffusivities in the sigma coordinate system. *Mon. Weather Rev.*, 113:1379–1383, 1985.

[29] G.L. Mellor, T. Ezer, and L.-Y. Oey. The pressure gradient conundrum of sigma coordinate ocean models. *J. Atmos. Oceanic Technol.*, 11:1126–1134, 1994.

[30] G.L. Mellor, L.-Y. Oey, and T. Ezer. Sigma coordinate pressure gradient errors and the seamount problem. *J. Atmos. Oceanic Technol.*, 15:1122–1131, 1998.

[31] G.L. Mellor and T. Yamada. Development of a turbulence closure model for geophysical fluid problems. *Rev. Geophys. Space Phys.*, 20:851–875, 1982.

[32] F. Mesinger. On the convergence and error problems of the calculation of the pressure gradient force in sigma co-ordinate models. *Geophys. Astrophys. Fluid Dynamics*, 19:105–117, 1982.

[33] F. Mesinger and A. Arakawa. Numerical methods used in atmospheric models, Volume I, 1976. WMO/ICSU Joint Organizing Committee, Garp Publication Series No. 17.

[34] N.A. Phillips. A coordinate system having some special advantages for numerical forecasting. *J. of Meteorology*, 14:184–185, 1957.

[35] A.F. Shchepetkin and J.C. McWilliams. A method for computing horizontal pressure-gradient force in an oceanic model with a non-aligned vertical coordinate, 2002. To appear in JGR.

[36] D. Slagstad. A 4-dimensional physical model of the Barents Sea. Technical Report STF48 F87013, SINTEF, 1987.

[37] L.H. Slørdal. *A Numerical Study of Dynamic Processes in a Stratified Ocean of Variable Depth. Part III: The Pressure Gradient Force in Sigma-Coordinate Ocean Models.* PhD thesis, Department of Geophysics, University of Oslo, P.O.Box 1022 Blindern, N-0315 Oslo, Norway, 1995.

[38] J. Smagorinsky. General circulation experiments with the primitive equations, I. The basic experiment. *Mon. Weather Rev.*, 91:99–164, 1963.

[39] G.S. Stelling and J.A.T.M. Van Kester. On the approximation of horizontal gradients in sigma coordinates for bathymetry with steep bottom slopes. *Int.J.Numer.Meth.Fluids*, 18:915–935, 1994.

[40] H. Sundquist. On truncation errors in sigma system models. *Atmosphere*, 13:81–95, 1975.

[41] P.K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 21:995–1011, 1984.

[42] D.-P. Wang. Mutual intrusion of a gravity current and density front formation. *J. Phys. Oceanogr.*, 14:1191–1199, 1984.

[43] G. Weatherly and P.J. Martin. On the structure and dynamics of the ocean bottom boundary. *J. Phys. Oceanogr.*, 8:557–570, 1978.

[44] H.Q. Yang and A.J. Przekwas. A comparative study of advanced shock-capturing schemes applied to burgers equation. *J.Comp.Phys.*, 102:139–159, 1992.