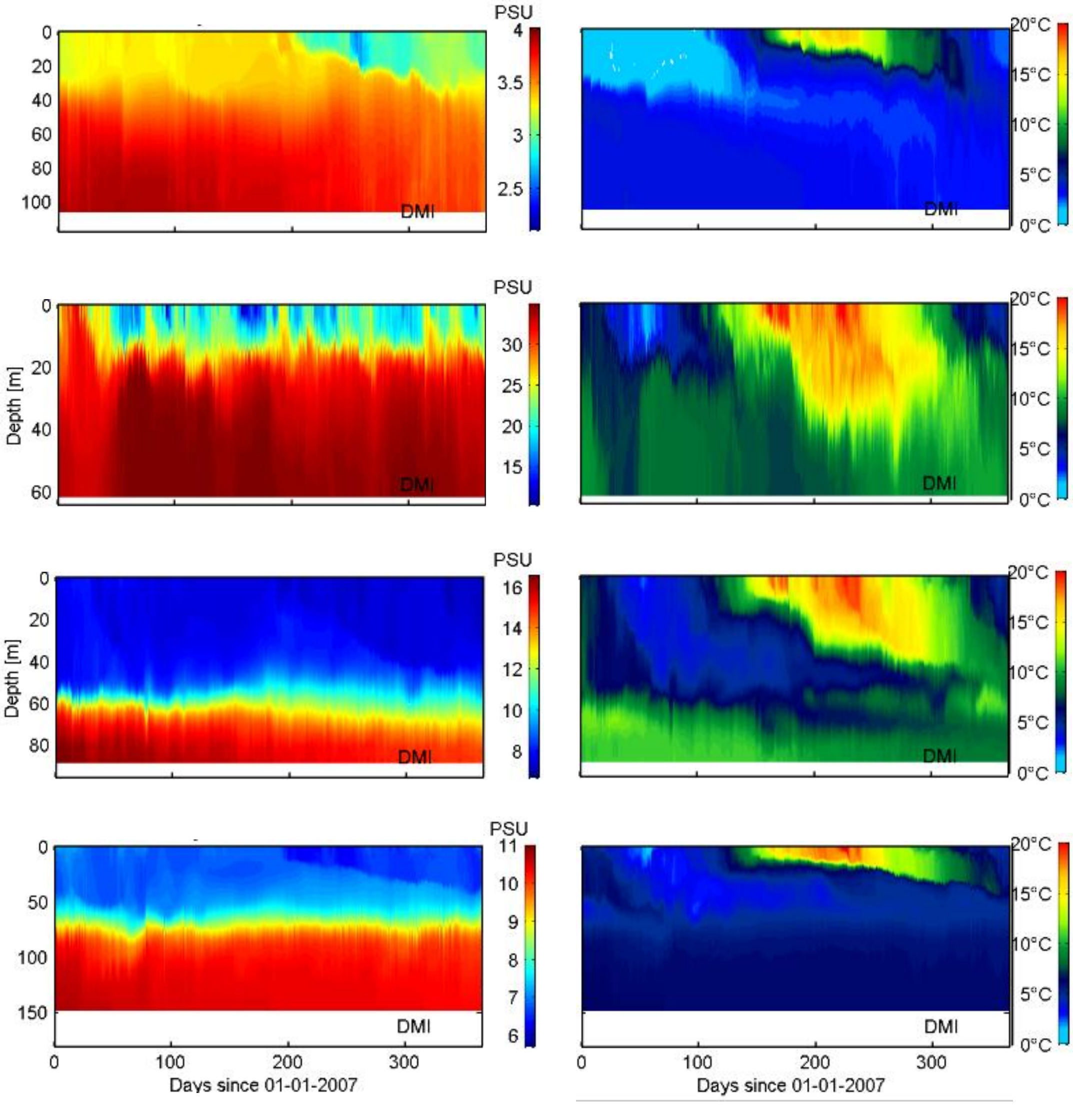
Danish Meteorological Institute

Ministry for Climate and Energy

Scientific Report 12-03

Mixing in HBM

Per Berg



Colophone

Serial title:
Scientific Report 12-03

Title:
Mixing in HBM

Subtitle:

Authors:
Per Berg

Other Contributers:

Responsible Institution:
Danish Meteorological Institute

Language:
English

Keywords:
mixing, structure functions, k-ω turbulence model, third order moments, convection, buoyancy production, vertical diffusivity, surface roughness length scale, critical gradient Richardson number, Neumann boundary conditions, HBM model

Url:
www.dmi.dk/dmi/sr12-03.pdf

ISSN:

ISBN:
978-87-7478-610-8

Version:
1.0

Website:
www.dmi.dk

Copyright:
Danish Meteorological Institute

Contents

Preface

The mixing scheme as described here is implemented into the HIROMB-BOOS ocean circulation model, HBM (Berg and Poulsen, 2012), and has proved its worth for some years in operational applications such as the DMI storm surge forecast model and in the MyOcean Baltic model.

The present document is the proper choice if you need a reference to the mixing scheme in HBM. It is appreciated if you quote the present document as well as the general implementation document for the HBM model by Berg and Poulsen (2012).

Front Page

The figures on the front page shows modelled profiles of salinity and temperature at four stations during 2007:
From top to bottom BMP13 in Botnian Bay, Læsø Øst in Kattegat, BMPK2 in Bornholm Basin and BMPI2 in Gotland Deep; salinity to the left and temperature to the right; vertical axis is depth and horizontal axis is time.

These results are obtained by the MyOcean Baltic V2 set-up using the latest release of the HBM code. Plotting of the results was made by Priidik Lagemaar and Germo Väli (Marine Systems Institute, Tallinn University of Technology) for the "Scientific Calibration Report (ScCR) for V2, WP 6 - Baltic MFC" of the MyOcean project, reference: MYO-WP6-ScCR-V2.

1. Introduction

A good turbulence model has extensive universality, and is not too complex to develop or use. Universality implies that a single set of empirical constants or functions, inserted into the equations, provides close simulation of a large variety of types of flow. Complexity is measured by the number of differential equations which the model contains, and the number of empirical constants and functions which are required to complete them; increase in the first complicates the task of using the model, increase in the second that of developing it.

LAUNDER AND SPALDING, 1973.

In ocean circulation models of the type considered here one solves equations for the mean variables. Mixing, which enters through non-resolved, higher-order moments of the fluctuating components of the variables, can be expressed in terms of vertical diffusivities and vertical gradients of the resolved mean variables. The reliability of a mixing scheme - which is the model of providing the vertical diffusivities - is predicated upon the physical content of its ingredients, and it is these ingredients which we have attempted to improve from previous models.

Our new mixing scheme consists of a two-equation turbulence closure model and algebraic structure functions. The two-equation turbulence closure model consists of transport equations for the turbulent kinetic energy, k, and for the inverse turbulent time scale or turbulent frequency, ω. The structure functions are algebraic functions not only of the gradient Richardson number as is most often the case, but also of the gradients of temperature and salinity and of the turbulent time scale. These choices are motivated by the need for properties such as accuracy, robustness, physical soundness, general applicability, and flexibility to add new features, still maintaining an acceptable level of the computational costs. Thus, from our own experience and testing as well as from literature, e.g. (Umlauf *et al.*, 2003), the k-ω turbulence model has convincingly proved superior to other two-equation models, i.e. better than or at least as good at the same computational cost, and the structure functions of (Canuto *et al.*, 2002) follow a formalism sufficiently general to describe the needed range of processes and to allow for extensions (plug-in features), yet not being too cumbersome to maintain.

Using well-known procedures, it then becomes a matter of algebra, rather than physics or empirical reasoning, to derive expressions for the vertical diffusivities in terms of the prognostic variables of the two-equation turbulence model and gradients of the resolved mean variables of the circulation model.

2. Summary

Our scheme is based on the - in aerodynamics - classical k-ω turbulence model (Wilcox, 1988) extended for buoyancy affected geophysical flows by Umlauf *et al.* (2003) but we apply a new set of coefficients developed by the present author to obtain consistency during transition between the different regimes of turbulence being predicted by the model. Through parameterizations we take both breaking surface waves (Craig and Banner, 1994) and internal waves (Axell, 2002) into account in the turbulence model. The atmospheric forcing (i.e. wind stress and cooling surface heat flux) of the turbulence model is provided through a new set of surface flux boundary conditions for k and for ω . We have introduced a new way of parameterizing extended buoyancy production during convection into the turbulence model, by inclusion of third order moments using concepts from (Abdella and McFarlane, 1997; D'Alessio *et al.*, 1998).

Different algebraic structure functions are applied for the vertical diffusivities of momentum, heat and salt. We have reconstructed the structure functions of (Canuto *et al.*, 2002) into new, computationally sound expressions in turbulent time scale, temperature gradient, salinity gradient, resolved velocity shear and unresolved but parameterized shear from breaking internal waves.

To account for penetration of short wave radiation into the subsurface model layers a parameterization of the insolation properly suited for the Baltic Sea area was implemented (Meier, 2001).

Also, during storm cases, to avoid the surface flow from increasing to unrealistically high current speeds, our implementation of the wind stress uses the effective wind velocity relative to the current velocity. A realistic description of the surface roughness length scale as seen from the ocean, z_0, is one of the major unsolved problems is oceanography, and our choice of implementing z_0 as being proportional to the turbulent surface velocity scale comes from considerable numerical experimentation in attempt to obtain good predictions in storm cases, both with respect to high water and to low water situations. During calm situations a constant z_0 of 10 cm is sufficient. It is, however, of vital importance that there is consistency between the z_0 applied in the turbulence model and the z_0 applied in the momentum equations, otherwise it may lead to spurious results in the forecasts.

An overview of our complete mixing scheme with some details is given in the following sections. First, in section 3, we describe the structure functions and the vertical diffusivities. Obtaining the vertical diffusivities is what is needed for obtaining closure to the system; the remaining sections of this document thus deal with determining the terms from which we can construct the vertical diffusivities. In section 4, we state the k-ω model equations. Then, in section 5, we describe the procedure for determining the coefficients of two-equation turbulence models consistently with the chosen stability functions and the physics being predicted. Parameterization of breaking internal waves and surface waves is described in section 6 and 7, respectively. Our new formulation of the set of surface boundary conditions for the k-ω turbulence model is given in section 8. Section 9 describes how we parameterize convection from third order moments. Thermal diffusivity and insolation is briefly described in section 10 and 11, respectively, before we end with a short description of how we have implemented the mixing scheme.

Most of the sometimes rather lengthy derivations and equations are left out of this document, though. It must be emphasised that our new mixing scheme has not yet been published and the present report documents original work by the author. More detailed documentation and validation exists as internal and personal notes at DMI, extracts of which constitute the present document.

3. Structure Functions And Vertical Diffusivities

To bring closure to the system of equations in our ocean circulation model we need to determine the three vertical diffusivities, namely K_M for momentum, K_H for heat, and K_S for salinity. These can be expressed as

$$K_i = 2\frac{k^2}{\epsilon}S_i \text{ with } i = M, H \text{ or } S$$

where k is the turbulent kinetic energy and ϵ is the dissipation rate of turbulent kinetic energy. The dimensionless functions S_i are called structure (or sometimes, stability) functions. For constructing suitable structure functions there is a variety of approaches described in the literature as well as implemented in ocean circulation models worldwide, ranging from simple constants to more involved expressions, including a variety of different features. A comparative analysis of some candidate formulations is given in (Burchard and Bolding, 2001). We have, however, chosen the more recent structure functions of Canuto *et al.* (2002) which, despite relatively high complexity, after some algebra with pen and paper turned out to become usable for operational oceanographic applications. In brief, these structure functions are expressed in terms of gradients of the prognostic variables of the ocean circulation model and of the turbulent time scale. We have also introduced 'shear' of unresolved scales (i.e. from parameterized internal wave breaking) into the formulae.

Below we give a brief description of the structure functions, but before that we need to define some basic quantities. The gradients of the mean salinity and temperature enter through R_S and R_H which are given by

$$R_S = g\beta_S \frac{\partial S}{\partial z}$$

$$R_H = g\alpha_H \frac{\partial \theta}{\partial z}$$

in which S is salinity, θ is temperaure, z is the vertical coordinate with positive direction upwards, g is gravity, β_S is the local haline contraction coefficient and α_H is the local thermal expansion coefficient. The density gradient enters through the buoyancy frequency squared

$$N^2 = -\frac{g}{\rho}\frac{\partial \rho}{\partial z} = R_H - R_S$$

The total shear

$$\Sigma_T^2 = \Sigma^2 + \sigma^2$$

is contributed by the resolved, large scale shear of the mean flow

$$\Sigma^2 = \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2$$

where (u, v) is the mean current, and by unresolved small scales which we for convenience have added up in the term σ. Such unresolved processes include e.g. internal wave breaking as discussed in section 6. The gradient Richardson number is defined as

$$R_i = \frac{N^2}{\Sigma_T^2}$$

The first thing to do is then defining a critical value of the gradient Richardson number $R_{i,crit}$ above which turbulent mixing ceases to exist. In the regime $R_i > R_{i,crit}$ we set K_i to a small background value. A method to obtain $R_{i,crit}$ as a function of the above-mentioned gradients was presented by Canuto *et al.* (2002). We have adapted their idea and from that we have constructed a computationally sound procedure. Instead of going into algebraic details which involves treating removable singularities when R_S and/or R_H approaches zero, we jump directly to showing our Fortran 90 source code implementation in Appendix 2.

With $R_{i,crit}$ in place, we can derive expressions for the structure functions S_i which are valid in the regime $R_i < R_{i,crit}$. Again we lean towards the work of Canuto *et al.* (2002) and write each structure function as a fraction with a common denominator

$$S_i = \frac{N_i}{D}$$

but we choose to proceed with modified forms of the expressions, outfactorizing common terms, and raising singularities in N^2, R_S and R_H. First, we need to define dimensionless functions representing resolved heat gradient, salt gradient and shear

$$f_H = -\pi_2 \pi_3 \tau^2 R_H$$

$$f_S = \pi_3^2 \tau^2 R_S$$

$$f_M = 0.16 \tau^2 \Sigma^2$$

where τ is the dynamical turbulent time scale which will be defined in the next section, and π_2 and π_3 are constants (see Appendix 1). These are computationally simpler than the corresponding functions applied in (Canuto *et al.*, 2002). Having done so, we find that

$$D = D(f_H, f_S, f_M)$$

$$N_M = N_M(f_H, f_S)$$

$$N_H = N_H(f_H, f_S, f_M)$$

$$N_S = N_S(f_H, f_S, f_M)$$

Our Fortran 90 source code implementation of these is shown in Appendix 3. The constants appearing in the expressions for $R_{i,crit}$ and for f_i, D and N_i are given in Appendix 1.

4. The k-ω Model Equations

The transport equations for the turbulent kinetic energy k and for the turbulent frequency ω read

$$\frac{\partial k}{\partial t} = \frac{\partial}{\partial z}\left(\frac{K_M}{\sigma_k}\frac{\partial k}{\partial z}\right) + P + G - \epsilon$$

$$\frac{\partial \omega}{\partial t} = \frac{\partial}{\partial z}\left(\frac{K_M}{\sigma_\omega}\frac{\partial \omega}{\partial z}\right) + \frac{\omega}{k}\left(c_{1\omega}P + c_{2\omega}G - c_{3\omega}\epsilon\right)$$

where P and G represents the production of turbulent kinetic energy by shear and buoyancy, respectively. The relationships between dissipation rate of turbulent kinetic energy ϵ, turbulent time scale τ, turbulent length scale l, and our prognostic turbulence variables k and ω are

$$\epsilon = c_{\mu 0}^4 k \omega$$

$$\tau = 2k/\epsilon$$

$$l = c_{\mu 0}^3 k^{3/2}/\epsilon$$

The coefficients of turbulence $c_{\mu 0}$, $c_{1\omega}$, $c_{2\omega}$, $c_{3\omega}$, σ_k and σ_ω which occur above will be determined in section 5.

The shear production has contributions from the resolved shear production of the mean flow and from non-resolved processes like e.g. internal wave breaking, i.e.

$$P = P_S + P_{IW}$$

While the first term is expressed in terms of the resolved shear of the mean flow as

$$P_S = K_M \Sigma^2$$

the P_{IW} term must be parameterized, see section 6. One might also consider adding a parameterization of shear production due to Langmuir circulation as done e.g. by Axell (2002); this has also been implemented into the HBM code but it has not been used for any real production run yet.

The buoyancy production has contributions from gradients of the mean salinity and temperature fields, and from higher order moments

$$G = K_S R_S - K_H R_H + G_{TOM}$$

The last term G_{TOM} is short-hand notation for our new way of parameterizing extended buoyancy production during convection by inclusion of third order moments (TOM) as will be described later in section 9.

5. Determine Coefficients In Two-Equation Turbulence Models

We will determine the constants used in the mixing scheme to be consistent with the chosen structure functions and the physics being predicted. The procedure is new and might prove generally applicable to two-equation turbulence closure models, but it is here demonstrated only for the k-ω turbulence model. In the seven steps outlined below we will bring forward our argumentation for the choice of closure coefficients.

Step 1: Having chosen an appropriate set of structure functions we need to determine the structure constant $c_{\mu 0}$. This is done by considering the structure function for momentum during quasi-equilibrium when production equals dissipation, in the absence of buoyancy. After some algebraic manipulations we find to four decimals precision that

$$c_{\mu 0} = 0.5234$$

Our value differs slightly from values found in the literature, where most studies depart from the classical value $c_{\mu 0} = 0.09^{1/4} \approx 0.5477$, see e.g. (Launder and Spalding, 1974; Rodi, 1987; Wilcox, 1988). An example is in (Umlauf *et al.*, 2003) where $c_{\mu 0}$ = 0.55, the rounded classical value, is used. The value $c_{\mu 0}$ = 0.5562 is used by Burchard and Petersen (1999), by Axell and Liungman (2001) and by Axell (2002) without further argumentation. Other authors derive - like we do - a value of $c_{\mu 0}$ consistently from the applied structure functions, e.g. $c_{\mu 0} = 0.077^{1/4} \approx 0.5268$ was derived by Burchard and Bolding (2001) and is in very close agreement with our value due to the close relationship between the applied structure functions (Canuto *et al.*, 2001).

Step 2: While much literature on turbulence modelling, including (Burchard and Bolding, 2001; Meier, 2001; Axell, 2002), uses the value 0.4 for von Karman's constant, κ, experiments and theory suggest a range of values gathered around a slightly greater value, see e.g. (Long *et al.*, 1993; Orszag and Patera, 1981). We here choose the representative value

$$\kappa = 0.41$$

Step 3: From grid-stirring experiments it is known that turbulence decays with distance according to a power law, see e.g. a summary in (Umlauf *et al.*, 2003). That is, the geometric length scale, l, which is the turbulent length scale in the limit of neutral shear flow, has a linear variation

$$l = L(z_0 - z) \text{ with } 0.06 < L < 0.33 < \kappa$$

while the turbulent kinetic energy behaves like

$$k = K(z_0 - z)^\alpha \text{ with } -3 < \alpha < -2$$

Besides the roughness length scale z_0 there are three unknown constants in these expressions, namely K, L and α. We can consider z_0 and K as arbitrary constants at this point but we do need explicitly to assign useful values to L and α. It is, however, important to note that L is not equal to von Karman's constant, κ, as authors commonly assumed in the literature (see e.g. Craig and Banner, 1994; Axell and Liungman 2001; Meier 2001); using L equal to κ would simply yield a too large length scale. We choose nice, rounded values to represent the above quoted ranges, and select

$$L = 0.25$$

$$\alpha = -2.5$$

While the value of α is a rather crucial choice since it affects the value of other coefficients as we shall see in Step 5 below, the value of L can be chosen more freely since L in our model will only enter explicitly through the expression for the surface boundary condition for ω where it appears as a factor to other constants of empirical origin (see the description of our boundary conditions later in section 8) and deviations in L can be absorbed therein.

Step 4: The diffusivity parameters σ_k and σ_ω are found from numerical experimentations. We choose the values following Wilcox (1988) who claims

$$\sigma_k = \sigma_\omega = 2$$

to be a saddle point in closure-coefficient space. We have not found justification for modifying these values of σ_k and σ_ω. Actually, one of the advantages of the k-ω model is indeed that it does not exhibit extreme sensitivity to variations of the values of these diffusivity parameters like the k-ϵ model does, see e.g. (Umlauf *et al.*, 2003).

Step 5: Having determined α, σ_k and σ_ω in steps 3 and 4 above, we now consider the wave affected surface layer where dissipation equals diffusion and we derive

$$c_{2\omega} = \frac{\sigma_k}{\sigma_\omega} \frac{(3 - 4/\alpha)^2 - 1}{24} = 0.84$$

Umlauf *et al.* (2003) also derived the relation between $c_{2\omega}$, α, σ_k and σ_ω but they introduced a sign-bug on α.

Our derived value for $c_{2\omega}$ differs slightly from the values found in the literature; the value $c_{2\omega}$ = (3/40)/(9/100) $\approx$ 0.8333 was originally proposed by Wilcox (1988), and rounded to $c_{2\omega}$ = 0.83 in (Umlauf *et al.*, 2003). Our value can, however, be further justified: In the case of decaying, isotropic turbulence we find that the asymptotic solution for k in our model is

$$k \sim t^{-1/c_{2\omega}} = t^{-1.19}$$

Experiments (Wilcox, 1988) suggest the asymptotic behaviour $k \sim t^{-1.2}$. We believe that our k-ω model thus is sufficiently consistent with those experiments. Furthermore, we have hereby also indirectly justified the chosen value of α, since a different choice of α would result in a different value of $c_{2\omega}$ and thereby a different (and possibly wrong) asymptotic solution.

Step 6: With κ, $c_{\mu 0}$, σ_ω and $c_{2\omega}$ in place, we consider the stationary case in the logarithmic boundary layer to derive

$$c_{1\omega} = c_{2\omega} - \frac{1}{\sigma_\omega} \left(\frac{\kappa}{c_{\mu 0}} \right)^2 = 0.53$$

This value differs somewhat from values found in the literature due to discrepancies in the other four constants, mainly through $c_{\mu 0}$ squared, but it is consistent with the rest of our model. Originally, Wilcox (1988) used $c_{1\omega}$ = 5/9 ≈ 0.5556, while Umlauf *et al.* (2003) rounded that to $c_{1\omega}$ = 0.56.

Step 7: The remaining coefficient to be determined is the buoyancy coefficient $c_{3\omega}$. While all the former coefficients determined during steps 1-6 are constants, this last coefficient need be dynamically adjusted.

During unstable stratification (convection) we use the value

$$c_{3\omega} = 0$$

Since application of the k-ω model for buoyancy affected geophysical flows is of a relatively recent date there is no literature available that discusses the role of $c_{3\omega}$ during unstable stratification, not even, strangely enough, in the original paper (Umlauf *et al.*, 2003). Our chosen value, $c_{3\omega}$ = 0, results from numerical experimentation by the present author. This value amounts to turning off the buoyancy sink for turbulent frequency during unstable stratification. In this respect, it should be mentioned that a straight-forward application of the differential relationship

$$\frac{\mathrm{d}\omega}{\omega} = \frac{\mathrm{d}\epsilon}{\epsilon} - \frac{\mathrm{d}k}{k}$$

in combination with the classical value of the controversial buoyancy coefficient for the k-ϵ model during unstable stratification, $c_{3\epsilon}$ = 1, see e.g. (Rodi, 1987), would confirm our result that $c_{3\omega}$ = 0, i.e. a value that would neglect effects of the unstable stratification in the scale determining equation. It is likely that we in the future will find a more useful value somewhere in the range from 0 to 1, but for now we find the value $c_{3\omega}$ = 0 reasonably well justified.

To determine $c_{3\omega}$ during neutral and stable stratification we consider the k-ω model equations during full equilibrium flow

$$P + G = \epsilon$$

$$c_{1\omega} P + c_{2\omega} G = c_{3\omega} \epsilon$$

which by elimination of ϵ from the first of these and of P and G by use of the definition of the flux Richardson number

$$R_F = -\frac{G}{P}$$

can be written as

$$c_{3\omega} = c_{2\omega} - \frac{c_{2\omega} - c_{1\omega}}{R_{F,ST}}$$

where $R_{F,ST}$ is the stationary flux Richardson number. Clearly, $c_{3\omega}$ must be less than -0.4, else our model will predict unphysical conditions with $R_{F,ST}$ > 0.25. To maintain physically sound predictions we first choose representative values for $R_{F,ST}$ according to the actual regime of turbulence, and then, use the above formula to obtain $c_{3\omega}$. In the regime of patchy turbulence, where the shear production due to breaking internal waves P_{IW} exceeds the shear production due to the resolved mean flow P_S, we set

$$R_{F,ST} = 0.09 \text{ for } P_S < P_{IW}$$

while the value is doubled for non-patchy turbulence

$$R_{F,ST} = 0.18 \text{ for } P_S \geq P_{IW}$$

These values for $R_{F,ST}$ are in close agreement with those used by Axell (2002). We are now able to determine $c_{3\omega}$ during neutral and stable stratification as

$$c_{3\omega} = -2.60 \text{ for } P_S < P_{IW}$$

$$c_{3\omega} = -0.88 \text{ for } P_S \geq P_{IW}$$

This is a new approach to obtaining useful values for $c_{3\omega}$.

It should be noted, that Umlauf *et al.* (2003) occasionally obtain $c_{3\omega}$ values which imply the unphysical condition $R_{F,ST} > 0.25$, in their investigations of two-equation turbulence models and structure functions.

6. Parameterization of Breaking Internal Waves

Different approaches have been suggested for mixing below the mixed-layer in ocean circulation models, ranging from using simple constant background diffusivities to more complex parameterizations of internal waves, see e.g. (Canuto *et al.*, 2002) for a summary. Here, we apply the work of Axell (2002) and let the shear production due to breaking internal waves during stable stratification have the following vertical distribution

$$P_{IW}(z) = \frac{F_0 N(z)}{\rho_0 N_{ave}} \text{ with } F_0 = 0.9 \text{ mW/m}^2$$

where N_{ave} is the depth averaged buoyancy frequency. The justification of this can be found from the numerical experiments performed by Axell (2002) who applies this production term to simulate deep water mixing in the Baltic Sea.

7. Parameterization of Breaking Surface Waves

The wind drag is described in the usual way through

$$u_F = \sqrt{|\boldsymbol{\tau_w}| / \rho}$$

$$\boldsymbol{\tau_w} = C_D \rho_{air} |\boldsymbol{W} - \boldsymbol{u}| (\boldsymbol{W} - \boldsymbol{u})$$

where u_F is the surface friction velocity and bold face letters indicate vector quantities, wind velocity $\boldsymbol{W}$, current velocity $\boldsymbol{u}$, and surface wind stress $\boldsymbol{\tau_w}$. The wind-strength dependent drag coefficient C_D will not be described further here.

There is little known about the surface roughness length scale z_0 *as seen from the ocean*. Some models/authors apply a constant value of typically 10 cm or an approach à la Charnock

$$z_0 \propto H_S = b_S \frac{u_F^2}{g}$$

relating z_0 to the significant wave height H_S, i.e. relating the surface roughness length scale *as seen from the atmosphere* to the friction velocity squared through a free parameter b_S. However, we find the quadratic dependency too strong for high wind speeds and too mild for low wind speeds (remember, we must use the same z_0 in the momentum equations). Numerical experimentations suggest a linear relation to u_F in absence of surface cooling and we are thus lead to choose

$$z_0 = \frac{1}{2}\sqrt{\rho}(q^3)^{1/3}$$

$$q^3 = u_F^3 + 0.54 w_H^3$$

The combined velocity scale q is thus expressed in terms of u_F which is generated mechanically by the wind, and of w_H which is thermally produced by surface cooling. The convective velocity scale w_H can physically be thought of as a sinking velocity of a parcel of fluid undergoing unstable surface forcing through a negative surface buoyancy flux B, i.e.

$$w_H = (-BH)^{1/3}$$

where H is the mixed-layer depth, see later in section 9. The weight factor 0.54 is from (Fischer *et al.*, 1979). In the absence of surface buoyancy (i.e. when B = 0) we see that z_0 varies approximately linearly with the wind speed in our approach, which is different from what we see in most other models where z_0 is chosen simply as a constant, or sometimes has the quadratic Charnock-like variation.

8. Boundary Conditions

Some authors have suggested

$$\frac{K_M}{\sigma_k}\frac{\partial k}{\partial z} = m_F u_F^3 - B\kappa z_0 \text{ at } z = 0$$

where the first term on the right hand side models the injection of turbulent kinetic energy due to surface wave breaking, and m_F=100 is an empirical constant suggested by Craig and Banner (1994) and later commonly used by modellers, see e.g. (Burchard, 2001; Meier, 2001; Umlauf *et al.,* 2003). In the last term on the right hand side B is the negative surface buoyancy flux and κz_0 is taken as a measure of the surface roughness (Meier, 2001; Axell and Liungman, 2001).

Indeed, we agree with above mentioned authors that Neumann boundary conditions are superior to Dirichlet boundary conditions for this purpose. However, we do not find it justified that κ should play any role here since we are not likely dealing with a logarithmic boundary layer, see also the discussion in (Umlauf *et al.,* 2003). We find it much more plausible to apply the combined velocity scale q which was defined in the previous section. Thus, our new surface boundary condition for k simply reads

$$\frac{K_M}{\sigma_k}\frac{\partial k}{\partial z} = m_F q^3 \text{ at } z = 0$$

We then derive the surface boundary conditions for ω analytically. From the power law of decay of k from the surface and the linear increase in length scale l, as we also used in step 3 of section 5, we differentiate with respect to z and apply the above-shown boundary condition for k at z=0. After some algebra our new surface boundary condition for ω then becomes

$$\frac{K_M}{\sigma_\omega}\frac{\partial \omega}{\partial z} = \frac{1 - \frac{1}{2}\alpha}{\sigma_\omega L z_0}\left(\frac{\sigma_k}{-\alpha L c_{\mu 0}} m_F q^3\right)^{2/3} \quad \text{at } z = 0$$

The boundary conditions at the bottom are obtained by assuming, between the sea bottom and the lower-most grid point, a logarithmic boundary layer and a local balance between production and dissipation of turbulent kinetic energy, in the absence of buoyancy, and are commonly expressed in terms of k and ϵ as

$$k = \left(\frac{u_{Fb}}{c_{\mu 0}}\right)^2$$

$$\epsilon = \frac{u_{Fb}^3}{\kappa z_{0b}}$$

where z_{0b} is the bottom roughness length scale which we assign the typical value of 10 cm, and u_{Fb} is the friction velocity obtained from the bottom-most velocity components as

$$u_F = \sqrt{r\left(u^2+v^2\right)}$$

In the above, r is a model setup specific bottom roughness parameter with a typical value of around 0.002. This formulation is unfortunately sensitive to the vertical grid resolution and there is no consistency between the applied turbulence model and the bottom friction implemented in the momentum equations, which in some situations lead to some spurious model effects. A more consistent formualtion, in line with the above-shown approach for the surface boundary conditions, is under considerations/development.

9. Extended Buoyancy Production During Convection

The only term we still need to find is G_{TOM} which enters the buoyancy production, see section 4. We apply the usual notation with a prime denoting fluctuations, and angled brackets mean averaging over the grid cell and the time step of the circulation model. The buoyancy production may then be expressed as a buoyancy flux which again is described in terms of a heat flux and a salinity flux

$$G = \langle b'w'\rangle = g\left(\alpha_H\langle\theta'w'\rangle - \beta_S\langle S'w'\rangle\right)$$

The two flux terms are treated separately. After some manipulations of expressions derived in (D'Alessio *et al.*, 1998), we arrive at

$$\langle\theta'w'\rangle = -K_H\frac{\partial\theta}{\partial z} + \gamma\frac{\langle\theta'w'\rangle_0}{H}\tau\left[0.4w_H + \frac{\alpha_H g}{2}\frac{c_2}{c_3}\theta_H\tau\right]$$

for the heat flux. We see that on the right hand side, besides the first term which is the usual down-gradient term resulting from treating second order moments, we obtain non-local terms as a result of including third order moments. In these new terms, H is the mixed-layer depth and

$$w_H = \left(\langle b'w'\rangle_0 H\right)^{1/3}$$

is the convective velocity scale established during surface cooling with surface buoyancy flux given by

$$\langle b'w'\rangle_0 = g\alpha_H\langle\theta'w'\rangle_0 = -g\alpha_H\frac{Q}{\rho C_p} = -B$$

where Q represents the net non-solar heat flux (that is, the sum of net long wave radiation, sensible, and latent heat fluxes) received at the sea surface and is negative for cooling. In the above, the kinematic surface heat flux is given by

$$\langle\theta'w'\rangle_0 = -\frac{Q}{\rho C_p}$$

The convective temperature scale θ_H is expressed in terms of the kinematic surface heat flux and the convective velocity scale w_H

$$\theta_H = \frac{\langle\theta'w'\rangle_0}{w_H}$$

Entrainment parameter, temperature variance dissipation constant and time scale constant are given as:

$$\gamma = 1.2\,,\, c_2 = 7.8\,,\, c_3 = 1.56$$

Now we have everything we need to implement the increased buoyancy production due to convection, except that we must know the mixed-layer depth H. This is the penalty we have to pay for the non-local terms, and it can be argued that this is a severe drawback of the model, since estimating H is not always that straight-forward and different authors prefer different methods. Here we determine H as the depth were the buoyancy frequency becomes zero (for B<0 of course). This choice has been justified sufficiently well through numerical experiments.

Possibly, at a later stage we may extend our model with yet another feature by introducing non-local terms to the salinity flux in a similar way by including a net precipitation surface flux. The expressions are ready:

$$\langle S'w'\rangle = -K_S\frac{\partial S}{\partial z} + 0.4\gamma w_S\frac{\langle S'w'\rangle_0}{H}\tau$$

where again the first term is the usual local down-gradient term, and in the last term

$$w_S = (\langle b'w'\rangle_0 H)^{1/3}$$

$$\langle b'w'\rangle_0 = -g\beta_S\langle S'w'\rangle_0 = g\beta_S S_0(E-P)$$

$$\langle S'w'\rangle_0 = -S_0(E-P)$$

It is left as an exercise for the future when/if applications should require such non-local terms; presently they do not, but *if* we should find it beneficial the description is developed, ready to plug in. Including the salinity flux may also be useful in cases of brine rejection under freezing sea water; the model will then be as described above with $E-P$ replaced by the brine rejection rate.

10. Thermal Diffusivity

To complete our mixing scheme we will include heat conduction through the water column as a diffusion process. The thermal diffusivity can be easily calculated from the specific heat capacity and thermal conductivity of sea water. Values for thermal conductivity of sea water is in the range from 0.561 W/m/K at 272 K to 0.673 W/m/K at 353 K, so without much argumentation we add the thermal diffusivity coefficient

$$D_{therm} = \frac{0.58\text{W/m/K}}{\rho C_p}$$

to the vertical diffusivity for temperature K_H. Here C_p is specific heat of sea water. Please note, the thermal diffusivity is very small in magnitude ($\sim 10^{-7}$ m^2/s) and thus just amounts to a small background diffusion so that the precise value is not important for our applications.

11. Penetrating Insolation

The model equation for conservation of temperature is

$$\frac{\partial\theta}{\partial t} + A(\theta) = D(\theta) + \frac{1}{\rho C_p}\frac{\partial I}{\partial z}$$

where A is the advection operator, D is the diffusion operator, and I is the intensity of the insolation down through the water column. The last term on the right hand side of this equation represents the non-turbulent source flux due to solar radiation. Recall, the turbulent heat flux has been taken into account through the boundary conditions for k and for ω as well as through the TOM parameterization. We parameterize I following (Meier, 2001) as

$$I(z) = Q_{SW}\left[R_{SW}\mathrm{e}^{z/\zeta_1} + (1-R_{SW})\mathrm{e}^{z/\zeta_2}\right]$$

where Q_{SW} is the short wave energy flux at the sea surface. The above formulation is suitable for the Baltic Sea with R_{SW}=0.64 and extinction lengths ζ_1=1.78 m and ζ_2=3.26 m.

It has been indicated through testing that the above-mentioned parameterization of I might actually not be very well suited for the North Sea - Baltic Sea region, so another formulation is being considered but until then we stick to Meier's (2001) formulation.

12. Discretization And Solution Method

Most of the model equations presented here are algebraic expressions and it is a more or less trivial task to implement these into the model code. We do, however, need to solve the transport equations for the turbulent kinetic energy and for the turbulent frequency. The equations and related quantities are discretized on the staggered grid with e.g. k and ω located at the grid points and with the diffusivities K_M, K_H, and K_S located at the top faces of the grid cells.

We use a semi-implicit scheme, where the diffusion, negative production and dissipation terms are treated fully implicitly while positive production terms are treated explicitly. The resulting system of equations can be solved with a standard tri-diagonal solver.

13. Concluding Remarks

The mixing scheme described here is implemented in the HBM ocean circulation model. The latest tagged release of HBM code runs operationally, using different setups, at DMI as the storm surge model and at the four MyOcean[1] Baltic Model Forecasting Centre production units BSH, DMI, SMHI, and FMI (Finnish Meteorological Institute) as the MyOcean Baltic Sea Version 2. Improving the storm surge forecasts as well as other operational model activities and project hindcasts (e.g. climate modelling) is a continuous mission of the HBM development.

As indicated in respective chapters it is reasonable to expect improvements and to include more phenomena. It is rather straight-forward to plug in extensions and improvement into the presented framework. One of the on-going tasks is to improve the bottom friction description, another is to test other formulations of the insolation. In a more long term perspctive it will be interesting to include wave effects from a wave model coupled to the circulation model.

[1] MyOcean is the main European project dedicated to the implementation of the GMES Marine Service for ocean monitoring and forecasting, `http://www.myocean.eu/`

References

Abdella, K. and N. McFarlane, 1997: A new second-order turbulence closure scheme for the planetary boundary layer. J. Atmos. Sci. **54**, 1850-1867.

Axell, L. B., 2002: Wind-driven internal waves and Langmuir circulations in a numerical model of the southern Baltic Sea. *J. Geophys. Res.* **107**, C11, 3204.

Axell, L. B. and O. Liungman, 2001: A one-equation turbulence model for geophysical applications: Comparison with data and the k-ϵ model. *Enviromental Fluid Mechanics* **1**, 71-106.

Berg, P. and J. W. Poulsen, 2012: Implementation details for HBM. DMI Technical Report No. 12-11, ISSN: 1399-1388, Copenhagen.

Burchard, H., 2001: Simulating the wave-enhanced layer under breaking surface waves with two-equation turbulence models. *J. Phys. Ocean.* **31**, 3133-3145.

Burchard, H. and K. Bolding, 2001: Comparative analysis of four second-moment turbulence closure models for the oceanic mixed layer. *J. Phys. Ocean.* **31**, 1943-1968.

Burchard, H. and O. Petersen, 1999: Models of turbulence in the marine environment - a comparative study of two-equation turbulence models. *J. Mar. Syst.* **21**, 29-53.

Canuto, V.M., A. Howard, Y. Cheng, and M. S. Dubovikov, 2001: Ocean Turbulence. Part I: One-point closure model – momentum and heat vertical diffusivities *J. Phys. Ocean.* **31**, 1413-1426.

Canuto, V.M., A. Howard, Y. Cheng, and M. S. Dubovikov, 2002: Ocean Turbulence. Part II: Vertical Diffusivities of Momentum, Heat, Salt, Mass, and Passive Scalars. *J. Phys. Ocean.* **32**, 240-264.

Craig, P. D. and M. L. Banner, 1994: Modelling Wave-Enhanced Turbulence in the Ocean Surface Layer. *J. Phys. Ocean.* **24**, 2546-2559.

D'Alessio, S. J. D., Abdella, K. and N. McFarlane, 1998: A new second-order turbulence closure scheme for modelling the ocean mixed layer. *J. Phys. Ocean.* **28**, 1624-1641.

Fischer, H. B., E. S. List, C.Y. Koh and J. Imberger, 1979: Mixing in Inland and Coastal Waters, *Academic Press*, Section 6.3.2, pp. 177-180.

Launder, B. E. and D. B. Spalding, 1974: The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering* **3**, 269-289.

Long, C. E., P. L. Wiberg, and A. R. M. Nowell, 1993: Evaluation of von Karman's constant from integral flow parameters. *J. Hydraulic Engineering* **119**, 1182-1190.

Meier, H. E. M., 2001: On the parameterization of mixing in three-dimensional Baltic Sea models. *J. Geophys. Res.* **106**, C12, 30997-31016.

Orszag, S. A. and A. T. Patera, 1981: Calculation of von Karman's constant for turbulent flow. *Phys. Rew. Lett.* **47**, 832-835.

Rodi, W., 1987: Examples of calculation methods for flow and mixing in stratified fluids. *J. Geophys. Res.* **92**, C5, 5305-5328.

Umlauf, L., H. Burchard, and K. Hutter, 2003: Extending the k-ω turbulence model towards oceanic applications. *Ocean Modelling* **5**, 195-218.

Wilcox, D. C., 1988: Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal* **26**, 1299-1310.

Previous reports

Previous reports from the Danish Meteorological Institute can be found on:
http://www.dmi.dk/dmi/dmi-publikationer.htm

Appendix 1: Constants In The Structure Functions

```
!  Constants for Part II structure functions (Canuto et al, 2002):
real(8), parameter :: pi1 = 0.08372_8,   pi4 = pi1,                &
                      pi2 = 1.0_8/3.0_8, pi3 = 0.72_8,             &
                      pi5 = pi3,                                   &
                      p1  = 0.832_8,     p2  = 0.545_8,            &
                      p3  = 0.2093_8,    p4  = 0.0323_8,           &
                      p5  = 0.0538_8,    p6  = 0.0698_8,           &
                      p7  = 1.6666_8,    p8  = p1,                 &
                      p9  = 0.2511_8,    p10 = p5,                 &
                      p11 = 0.1163_8,                              &
                      a1  = 0.3022_8,    a2  = 0.2986_8,           &
                      a3  = 0.064780_8,  a4  =-3.99459_8,          &
                      a5  =-1.8493_8,                              &
                      b1  =-0.0625_8,    b2  =-0.1163_8,           &
                      b3  = 0.5702_8,    b4  =-0.9689_8,           &
                      b5  =-2.0930_8,    b6  =-0.0538_8,           &
                      b7  =-0.13488_8,                             &
                      d1  = 0.03201_8,   d2  = 0.0318_8,           &
                      d3  = 0.00686_8,   d4  =-0.0289_8,           &
                      d5  =-0.04272_8,   d6  =-0.019780_8,         &
                      d7  =-0.0028750_8, d8  =-0.41319_8,          &
                      d9  =-0.1912_8,    d10 = 1.1773_8,           &
                      d11 = 1.1612_8,    d12 = 0.2523_8,           &
                      d13 = 1.1857_8,    d14 =-10.7721_8,          &
                      d15 =-4.9871_8
```

Appendix 2: Critical Value Of The Gradient Richardson Number

```
  subroutine CheckRgCrit( rT, rS, Rg, CRg )
    !---------------------------------------------------------------------------
    ! check if critical Rg is exceeded.
    !---------------------------------------------------------------------------
```

```
!- directives ----------------------------------------------------------------
implicit none

!- arguments -----------------------------------------------------------------
real(8), intent(in)  :: rT, rS, Rg
logical, intent(out) :: CRg

!- local vars ----------------------------------------------------------------
real(8)              :: f, f1, f2, rT2, rTS, rS2
real(8), parameter   :: rsmall = 0.00000001_8   ! rather small value
real(8), parameter   :: A0 = -6.25_8*pi2*pi3, B0 = 6.25_8*pi3*pi3
real(8), parameter   :: AB = A0*B0

!- check for obvious Rg value ----------------------------------------------
if (Rg < one) then
  CRg = .false.

else

  !- treat small rT:
  if (abs(rT) < rsmall) then
    f1 =  two*a3*B0 + five*pi1*b3*b6 - six*d3*B0
    f2 = (six*d7*B0 - five*pi1*b4*b6)*B0

  !- treat larger rT:
  elseif (abs(rS) < rsmall) then
    f1 =  two*a1*A0 - five*pi4*b3*b2 - six*d1*A0
    f2 = (six*d4*A0 + five*pi4*b2*b5)*A0

  !- super-critical density ratio:
  elseif (rS/rT > Rrho_crit) then
    f1 = ten      ! dummy values to make f larger than Rg
    f2 = one

  !- general case:
  else
    rT2 = rT**2
    rTS = rT*rS
    rS2 = rS**2
    f1 = (two*a1*A0-five*pi4*b3*b2-six*d1*A0)*A0*rT2                       &
       + (two*a2*AB-five*pi4*b3*b1*B0+five*pi1*b3*b7*A0-six*d2*AB)*rTS     &
       + (two*a3*B0+five*pi1*b3*b6-six*d3*B0)*B0*rS2
    f2 = (six*d4*A0+five*pi4*b2*b5)*A0*A0*rT2*rT                           &
       + (six*d5*AB+five*pi4*(b1*b5+b2*b4)*B0-five*pi1*b5*b7*A0)*A0*rTS*rT &
       + (six*d6*AB-five*pi1*(b6*b5+b7*b4)*A0+five*pi4*b1*b4*B0)*B0*rTS*rS &
       + (six*d7*B0-five*pi1*b4*b6)*B0*B0*rS2*rS

  endif

  !- compare Rg to f:
  if (f2 >= zero .and. f2 < tinyv) then
    f2 = tinyv
  elseif (f2 < zero .and. f2 > -tinyv) then
    f2 = -tinyv
  endif
  f = ccrit*f1/f2
  if (f > one) then
    CRg = (Rg >= f)
  else
```

```
       CRg = .false.
     endif

   endif

 end subroutine CheckRgCrit
```

Appendix 3: Computation Of The Structure Functions

```
subroutine PartIIDiff( Km, Kh, Ks, tke, diss, rT, rS, Sh2, kb,                &
                       eddydif_max, eddyvis_max, Piw)
   !---------------------------------------------------------------------------
   ! Calculates the stability functions
   ! and the resulting eddy viscosity and eddy diffusivities.
   !
   ! Inputs:        tke:       Turbulent kinetic energy.
   !                diss:      Dissipation.
   !                rT:        Temperature gradient, g*alphaT*dT/dz
   !                rS:        Salinity gradient, g*alphaS*dS/dz
   !                Sh2:       Mean shear.
   !                kb:        Number of active cells.
   !                Piw:       Shear production due to internal waves.
   !
   ! Outputs:       Km:        Momentum diffusivity.
   !                Kh:        Heat diffusivity.
   !                Ks:        Salt diffusivity.
   !---------------------------------------------------------------------------

   !- directives --------------------------------------------------------------
   implicit none

   !- arguments ---------------------------------------------------------------
   integer(4), intent(in)    :: kb
   real(8),    intent(inout) :: Km(1:), Kh(1:), Ks(1:)
   real(8),    intent(in)    :: tke(1:), diss(1:), rT(1:), rS(1:), Sh2(1:)
   real(8),    intent(in)    :: Piw(2:)
   real(8),    intent(in)    :: eddydif_max, eddyvis_max

   !- locals vars -------------------------------------------------------------
   real(8)    :: t2, ss, D, Nm, Nh, Ns, fc
   real(8)    :: fm, fs, fh, Rg_t, Sh2_t, N2
   integer(4) :: k
   logical    :: CheckRg
   real(8)    :: PiwKm(2:kb)
   real(8), parameter :: y0 = 0.16_8, D0 = 24.0_8
   real(8), parameter :: nn1 = 8.0_8, nn2 = 12.0_8, nn3 = 75.0_8
   real(8), parameter :: ff1 = 4.0_8, ff2 = 60.0_8, ff3 = 15.0_8

   !- skip one-layers ---------------------------------------------------------
   if (kb < 2) return

   !- some initial settings ---------------------------------------------------
   PiwKm(2:kb) = Piw(2:kb)/Km(2:kb)

   !- find vertical diffusivities ---------------------------------------------
```

```
  !  k = 1 (dummy values, not used anywhere) .................................

  !  Do the rest of the water column k=2,kb ..................................
  do k=2,kb
    !- dynamical turbulent time scale squared, t2
    t2 = (tke(k-1)/diss(k-1) + tke(k)/diss(k))**2

    !- total shear and Rg:
    N2    = rT(k) - rS(k)
    Sh2_t = Sh2(k) + PiwKm(k)
    Rg_t  = min( N2/max(Sh2_t,tinyv), Rg_max )

    !- no mixing if critical Rg is exceeded:
    call CheckRgCrit(rT(k), rS(k), Rg_t, CheckRg)
    if (CheckRg) then
      Km(k) = laminarvis
      Kh(k) = eddydif_min
      Ks(k) = eddydif_min

    !- mixing if critical Rg is not exceeded:
    else
      fh = -pi2*pi3*t2*rT(k)
      fs =  pi3*pi3*t2*rS(k)
      fm =       y0*t2*Sh2_t

      D  = D0                                                                &
         + ((d1*fh + d2*fs +  d8)*fh + (d3*fs +  d9)*fs + d13)*fm            &
         + ((d4*fh + d5*fs + d10)*fh + (d6*fs + d11)*fs + d14)*fh            &
         + ((d7*fs + d12)*fs + d15)*fs

      if (D >= zero .and. D < tinyv) then
        D = tinyv
      elseif (D < zero .and. D > -tinyv) then
        D = -tinyv
      endif

      !- momentum:
      Nm = nn1*(nn2 + (a1*fh + a2*fs + a4)*fh + (a3*fs + a5)*fs)/nn3

      !- heat:
      fc = ff1*(ff2 + b3*fm + b4*fs +b5*fh)/ff3
      Nh = pi4*(one + b1*fs + b2*fh)*fc

      !- salinity:
      Ns = pi1*(one + b6*fs + b7*fh)*fc

      !- eddy diffusivity normalised w/ structure function=1, ss:
      ss = tke(k-1)**2/diss(k-1) + tke(k)**2/diss(k)

      !- vertical diffusivities:
      Km(k) = min( max(ss*Nm/D, laminarvis),  eddyvis_max )
      Kh(k) = min( max(ss*Nh/D, eddydif_min), eddydif_max )
      Ks(k) = min( max(ss*Ns/D, eddydif_min), eddydif_max )
    endif

  enddo

end subroutine PartIIDiff
```