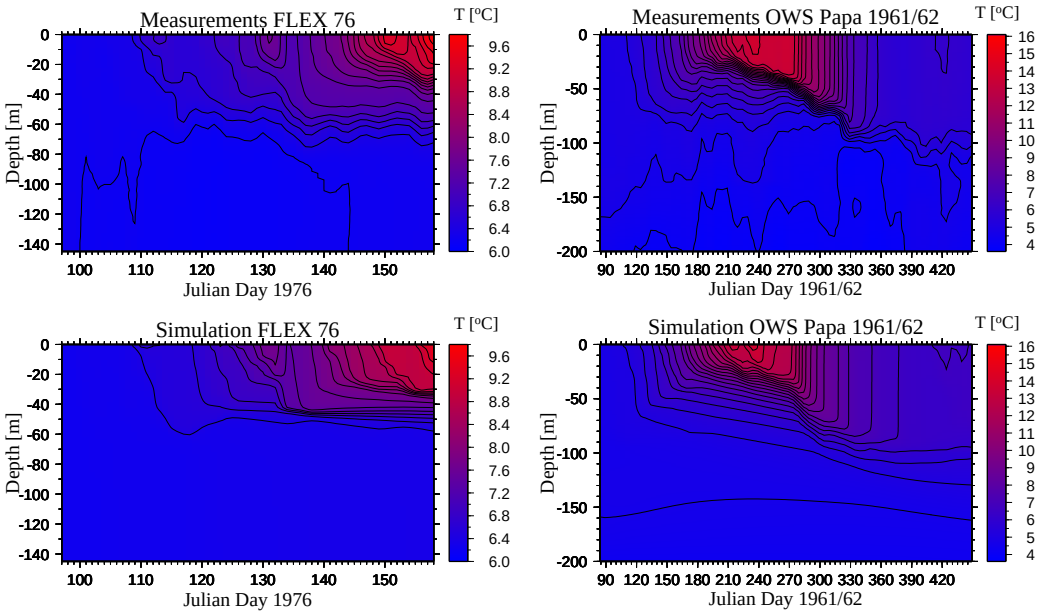
GOTM,
a **G**eneral **O**cean **T**urbulence **M**odel.
Theory, implementation and test cases.

Hans Burchard, Karsten Bolding and Manuel R. Villarreal



Contents

Chapter 1

Introduction

1.1 The idea behind GOTM

General **O**cean **T**urbulence **M**odel (**GOTM**) is a very ambitious name for a one-dimensional water column model which simply allows for different combinations of momentum and tracer equations and a choice between some standard turbulence parameterizations. A turbulence modelling expert might not find any new or sophisticated turbulence closure scheme in **GOTM**. Whoever is missing any feature in **GOTM**, is invited to contribute to **GOTM** and add her or his personal preferences or needs. **GOTM** is far from being complete and it will never be completed. At some certain stage during the development of **GOTM** in winter 1998/99, we decided to freeze innovations and to consolidate, comment and test what was already included. The result of these efforts consists of the present report, the **GOTM** source code, and the data and input files for some test cases. All this is put together on the following site of the World Wide Web:

`http://www.ifm.uni-hamburg.de/~wwwto/ResearchTopics/GOTM/gotm_main.html`

The idea of **GOTM** is that scientists with interest in water column processes download **GOTM** to their computer in order to

- learn about the numerical treatment of vertical mixing processes;
- compare the performance of different given turbulence closure schemes;
- add their personally preferred turbulence closure scheme to **GOTM** and compare its performance with standard closure schemes;
- code turbulence closure schemes from literature or develop own schemes and compare their performance with schemes already included in **GOTM**;
- transform their own field data to the **GOTM** format and simulate numerically the water column processes under investigation;
- add new processes like e.g. biological modules to **GOTM** and investigate their dynamics in idealized or realistic test cases;
- copy pieces of code from **GOTM** and integrate them into their personal one-, two-, or three-dimensional hydrodynamic codes;

- etc.

The structure of **GOTM** is modular, which should support the extension to further processes or turbulence closure schemes. Interested colleagues should report to us their experiences with or extensions to **GOTM** such that they may be considered and integrated into further releases of **GOTM**. All contributors will then be considered as co-developers and co-authors of **GOTM** and thus be acknowledged for it. The idea is that such **GOTM** is steadily developing and hopefully more and more deserving its name.

1.2 The motivation for GOTM

There are many different water column model existing world wide. It might be concluded from this that there is no need for a further model like **GOTM**. But the experience of the authors of this report is different. Although the pre-versions of **GOTM** (see section 1.3) have never been promoted as public domain codes, various requests for the code reached the authors of this report. Four typical cases will be mentioned here. Andy Visser (Danish Fisheries Institute, Charlottenlund, Denmark) used an old PASCAL version of the k-ε model for calculating a realistic eddy viscosity profile in the mixed layer which he needed as input for a random-walk model for marine bugs (*Visser* [1997]). Olof Liungman (Göteborg University, Earth Sciences Centre, Sweden) modified and extended one of the first FORTRAN versions of the k-ε model for calculating internal wave mixing in fjord basins (*Liungman* [1998]). Chris Sherwood (Commonwealth Scientific and Industrial Research Organisation, Hobart, Australia) took a pre-version of **GOTM** as an example how to code the k-ε two-equation turbulence model. That was then used for simulating sea ice - current - turbulence - sediment interactions in the Kara Sea (*Sherwood* [1998]). And, questioning the **O** in **GOTM**, Gerrit Goudsmit (Swiss Federal Institute for Environmental Science and Technology, Switzerland) included his internal seiche model for simulating dissipation rate measurements in horizontally averaged Swiss lakes (*Goudsmit et al.* [1999]).
These four completely different applications showed us the need for a public domain water column model with a high degree of generality and a good state of documentation. This release of **GOTM** is understood as a first step towards this aim.

1.3 Short history of GOTM

Many people have been giving important impulses to the development of **GOTM**. The first lines of the first PASCAL code have been programmed in late 1992 by the first author of this report as part of a Ph.D. project at the Institut für Meereskunde of the Universität Hamburg, Germany, under the supervision of Jürgen Sündermann and Helmut Baumert. This was the the basis for a one-dimensional water column model for simulating the so-called FLEX'76 data (see *Burchard and Baumert* [1995] and also section 6.2.1 in this report), a two-dimensional channel model (*Burchard and Baumert* [1998]) and a three-dimensional shallow water model (*Burchard* [1995]). Further substantial advice for these modelling activities came also from Eckard Kleine (Bundesamt für Seeschifffahrt und Hydrographie, Hamburg, Germany) who together with Helmut Baumert had a long time experience in turbulence modelling (see e.g. *Baumert et al.* [1989a, 1989b] and *Baumert and Radach* [1992]).
A strong motivation for extending the model code (which at that stage only included the k-ε model) came in 1995 from George Mellor (Princeton University, New Jersey) who provoked with

the statement that he never saw a k-ε model (in contrast to the Mellor-Yamada model) reproducing the Monin-Obukhov stability theory. In an effort together with Ole Petersen (International Centre of Computational Hydrodynamics, Hørsholm, Denmark) the equivalence of the k-ε and the Mellor-Yamada models under certain circumstances and thus the successful reproduction of the Monin-Obukhov stability theory could be demonstrated (*Burchard et al.* [1998], *Burchard and Petersen* [1999]). Stability problems occurred when the water column model (now already with a switch for choosing either the k-ε or the Mellor-Yamada model) was transferred to FORTRAN in late 1996. It was Boris Kagan (Shirshov Institute of Oceanology, St. Petersburg, Russia) who insisted that oscillations in the turbulent macro length scale under very small stable stratification are numerical artifacts and therefore unacceptable. The solution to that problem was finally to use double instead of single precision for the calculations.

In a cooperation with Nadia Pinardi and Sergio Castellari (Istituto per lo Studio delle Metodologie Geofisiche Ambientali, Bologna, Italy) during 1997 and 1998, the experiences of that FORTRAN code were incorporated into a version of the Modular Ocean Model (MOM, Geophysical Fluid Dynamics Laboratory, Princeton, New Jersey) for calculating the general circulation and convective events in the Mediterranean Sea (*Castellari et al.* [1999]). During the same years, the turbulence model was under strong criticism from Walter Eifler and Adolf Stips (Joint Research Centre, Ispra, Italy) when it was inter-calibrated with in-situ measurements of dissipation rate in two different estuaries. During this effort both, the model and the measurement strategy were significantly improved. However, discrepancies remained and the discussion is still going on (*Burchard et al.* [1999], and sections 6.3.1 and 6.3.2 in this report).

The idea of **GOTM**, as it is presented here, was born in spring 1998, when Karsten Bolding (the second author of this report) joined the development. It was a major effort to collect the many different codes (each set up for only one test case), written in different programming languages, stored on different computers and merge them together into one product. In summer 1998, Manuel Ruiz Villarreal (the third author of this report) joined us on all levels of **GOTM** such as internal wave parameterizations, algebraic length scales, documentation, test cases, graphics, etc.

1.4 The structure of this report

This report is designed as the scientific documentation of **GOTM**. It should help the user of **GOTM** to understand the physics, mathematics and numerics behind it. The chapter **Equations** (2) is basically structured into **Mean flow equations** (2.1) and **Turbulence equations** (2.2) which are only coupled through the eddy viscosity and the eddy diffusivity (from 2.2 to 2.1) and the shear and the buoyancy frequencies (from 2.1 to 2.2). It has been tried to document all equations included in **GOTM** in detail. In contrast to that, the chapter **Numerical Treatment** (3), briefly mentions the numerical treatment of the equations as far as standard methods are used. Some non-standard procedures in discretizing especially near bottom gradients are discussed in detail and compared to analytical solutions. Chapter **Implementation** (4) discusses, how the numerical discretization of the model is implemented as a FORTRAN 77 code. In the **User's Guide** (chapter 5), support for those who want to use or change **GOTM** is given. Finally, in **Test Cases** (6), idealized and realistic examples of simulations for various sites are described and discussed, and some results are shown as figures.

1.5 Acknowledgments

GOTM and this report could only be made with the support of a number of colleagues who accompanied its development with interest. Pierre-Philippe Mathieu (Joint Research Centre, Ispra, Italy), who is definitely our first user, made a number of suggestions how to structure the code in order to allow for future extensions towards the inclusion of biological models. Chris Sherwood (Commonwealth Scientific and Industrial Research Organisation, Hobart, Australia), Olof Liungman (Göteborg University, Earth Sciences Centre, Sweden) and Georg Umgiesser (Istituto per lo Studio della Dinamica delle Grandi Masse, Venezia, Italy) gave us some advice and feedback about installing and running **GOTM**. This report had been sent out for two rounds of reviews. It were Walter Eifler, Pierre-Philippe Mathieu (both Joint Research Centre, Ispra, Italy), Ole Petersen (Danish Hydraulic Institute, Hørsholm, Denmark), and Thomas Pohlmann (Institut für Meereskunde, Universität Hamburg, Germany), who gave helpful and substantial comments and suggestions. Dagmar Hainbucher (Institut für Meereskunde, Universität Hamburg, Germany) gave an introduction into the miracles of GMT (Generic Mapping Tools, see *Wessel and Smith* [1998]). The work of Hans Burchard had been partially funded by the European Communities' Program in Marine Science and Technology (MAST) under the contract numbers MAS3-CT96-0053 ('PhaSE') and MAS3-CT96-0051 ('MTP II-MATER') and by a grant from the German Research Foundation, the work of Karsten Bolding by the MAST contracts MAS3-CT97-0159 ('PROVESS') and MAS3-CT97-0148 ('KEYCOP'). The work of Manuel Ruiz Villarreal has been performed during a stay at the Joint Research Centre in Ispra, Italy, which was sponsored by a grant from the Xunta de Galicia, Spain. We are very grateful to these sponsors for their generous support.

Chapter 2

Equations

2.1 Mean flow equations

2.1.1 Three-dimensional form

GOTM is based on the hydrodynamic equations of motion by applying the Boussinesq approximation (density only considered variable when multiplied with gravitational acceleration), the hydrostatic approximation (pressure is considered hydrostatic), and the eddy viscosity approximation (vertical turbulent transport is approximated by a diffusive transport, see also section 2.2). The equations will all be given in Cartesian coordinates with x directed east, y directed north, and z upwards with a reference value of $z = 0$ located at the mean sea level. The water column ranges from the bottom at $z = -H(x, y)$ to the surface at $z = \zeta(t, x, y)$. These coordinates for bottom and surface are located at the edge of the validity of the following model equations, which is excluding the viscous zones near the boundaries (see figure 2.1).

Therefore, roughness lengths at bottom and surface, z_0^b and z_0^s have to be defined. In **GOTM**, this is done as follows (see e.g. *Kagan* [1995]):

$$z_0^b = 0.1\frac{\nu}{u_*^b} + 0.03h_0^b. \tag{2.1}$$

with the molecular viscosity ν, the friction velocity u_*^b (see equation (2.20)), and the height of the bottom roughness elements, h_0^b. At the surface, the simple relation

$$z_0^s = 0.03h_0^s \tag{2.2}$$

with the height of the surface roughness elements, h_0^s. So far, h_0^s is nothing than a tuning parameter.

2.1.1.1 Hydrodynamic equations

In three-dimensional coordinates, the momentum equations are of the following form:

$$\partial_t u + \partial_x(u^2) + \partial_y(uv) + \partial_z(uw) - \partial_z\Big((\nu_t + \nu)\partial_z u\Big) = -\frac{1}{\rho_0}\partial_x p + fv, \tag{2.3}$$

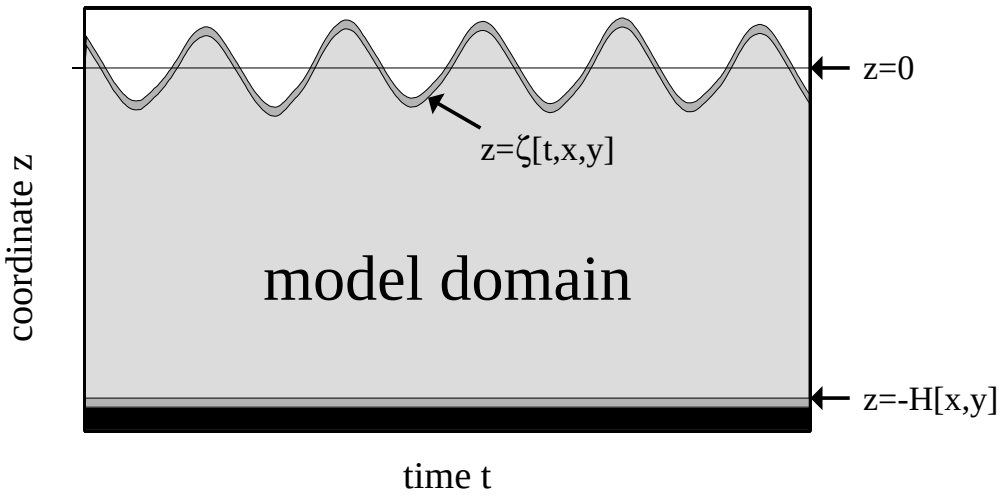


Figure 2.1: Sketch showing the definition of the model domain, which is only the light grey area in the (t, z)-space. The dark grey areas are the viscous sublayers, which are not described by the model equations and the effect of which has to be parameterized. The thicknesses of these sublayers are z_0^b for the bottom layer and z_0^s for the surface layer.

$$\partial_t v + \partial_x(vu) + \partial_y(v^2) + \partial_z(vw) - \partial_z\left((\nu_t + \nu)\partial_z v\right) = -\frac{1}{\rho_0}\partial_y p - fu. \tag{2.4}$$

Here, u, v and w are the velocity components with respect to the x, y and z direction, respectively. ν_t is the vertical eddy viscosity, $\nu = 1.3 \cdot 10^{-6}\mathrm{m^2 s^{-1}}$ the kinematic viscosity, f the Coriolis parameter, ρ_0 a constant reference density and p the pressure. Horizontal diffusion parameterizations will not be specified here in more detail, because in the one-dimensional form of the equations (see section 2.1.2.1) only their vertical mean effect will be considered anyway.
With the hydrostatic assumption

$$\partial_z p + g\rho = 0, \tag{2.5}$$

where $g = 9.81\mathrm{m\,s^{-2}}$ is the gravitational acceleration, and ρ the density, the pressure gradients can now be expressed as

$$-\frac{1}{\rho_0}\partial_x p = -g\frac{\rho(\zeta)}{\rho_0}\partial_x \zeta + \int_z^\zeta \partial_x b\, dz' \tag{2.6}$$

and

$$-\frac{1}{\rho_0}\partial_y p = -g\frac{\rho(\zeta)}{\rho_0}\partial_y \zeta + \int_z^\zeta \partial_y b\, dz' \tag{2.7}$$

where

$$b = -g\frac{\rho - \rho_0}{\rho_0} \tag{2.8}$$

is the buoyancy. The first term on the right hand side in (2.6) and (2.7) is the external (due to surface slopes) and the second the internal (due to density gradients) pressure gradient.
For the mathematical closure of the system of equations (2.3) - (2.8), the calculation of (i) the surface elevation ζ, (ii) the density ρ, (iii) the eddy viscosity ν_t, and (iv) lateral and vertical boundary conditions for u and v are missing.

(i) The calculation of the surface elevation from a differential equation (derived from mass conservation and incompressibility of water) will not be discussed here, because it is considered as an input parameter for one-dimensional models.

(ii) The density will be calculated from an equation of state which takes all active tracers into consideration (see section 2.1.1.2.4).

(iii) The calculation of the eddy viscosity (and diffusivity for the tracer equations) is the main topic of this report and is described in section 2.2.

(iv) Lateral boundary conditions will not be needed for the one-dimensional form of the equations. The vertical (so-called dynamic) boundary conditions parameterizing bottom friction and surface momentum flux due to wind stress are discussed just below.

The surface momentum flux is given by the surface stress which can be calculated from the components of the wind vector, the wave state and other parameters (see e.g. *Kondo* [1975]):

$$(\nu_t + \nu)\partial_z u = \frac{\tau_x^s}{\rho_0}, \qquad z = \zeta, \tag{2.9}$$

$$(\nu_t + \nu)\partial_z v = \frac{\tau_y^s}{\rho_0}, \qquad z = \zeta. \tag{2.10}$$

At the bottom, no-slip boundary conditions are assumed:

$$u = 0, \qquad z = -H, \tag{2.11}$$

$$v = 0, \qquad z = -H. \tag{2.12}$$

The Dirichlet boundary conditions (2.11) and (2.12) might appear to be non-standard for high resolution modelling of turbulent flows. The well-known law of quadratic bottom friction will be introduced later on the numerical level (see section 3.3), only in order to account for the staggering of the numerical grid. While the physical velocity at the bottom is zero, the lowest numerical value will be considered as the (in general non-zero) mean of the bottom layer. This means that we physically apply Dirichlet conditions but numerically the quadratic bottom friction as Neumann type condition. It will be shown that for the limit of vanishing thickness of vertical layers both formulations are equivalent, if the drag coefficient is properly computed.

2.1.1.2 Tracer equations

A tracer may be any property of the water (like temperature, density) or dissolved (salt, nutrients) or non-dissolved (sediment, phytoplankton) substance. There are attempts even to treat sea grass canopies as tracers (see *Verduin and Backhaus* [1998]). In principle, also turbulent quantities are tracers, but however, because of their specific importance for the whole dynamics of the flow, they are treated separately (see section 2.2).

The general form of the three-dimensional equation for a tracer A, which we adopt for our work is:

$$\partial_t A + \partial_x(Au) + \partial_y(Av) + \partial_z\left(A(w - w_s) - (\nu_t'(A) + \nu'(A))\partial_z A\right) = R(A). \tag{2.13}$$

The molecular diffusivity of A is denoted by $\nu'(A)$, the turbulent diffusivity by $\nu_t'(A)$. The term $R(A)$ on the right hand side of (2.13) is the sum of all inner sources and sinks for the tracer A. A vertical movement of the tracer relative to the vertical velocity W due to negative or positive buoyancy or active swimming[1] is denoted by w_s, which is chosen positive for a downward movement.

2.1.1.2.1 Temperature The molecular diffusivity for temperature T used here is chosen as $\nu' = 1.4 \cdot 10^{-7}\mathrm{m^2 s^{-1}}$.
The incoming short-wave solar radiation is here generally treated as an inner source of heat which leads to an increase of temperature in the upper water column. This absorption of radiation is modelled as:

$$R(T) = \frac{\partial_z I}{C_p \rho_0} \tag{2.14}$$

with the average specific heat capacity of sea water, $C_p = 3980\ \mathrm{J\,kg^{-1}K^{-1}}$ and the energy flux I. According to *Paulson and Simpson* [1977] the radiation I in the upper water column may be parameterized by

$$I(z) = I_0\left(ae^{-\eta_1 z} + (1-a)e^{-\eta_2 z}\right). \tag{2.15}$$

Here, I_0 is the albedo corrected radiation normal to the sea surface. The weighting parameter a and the attenuation lengths for the longer and the shorter fraction of the short-wave radiation, η_1 and η_2, respectively, depend on the turbidity of the water. According to *Jerlov* [1968], 6 different classes of water can be defined which are all included in **GOTM**. Time variations of turbidity due to biomass dynamics may be considered by reading in measured values from a file.
In contrast to the short-wave radiation, latent and sensible heat flux, Q_E and Q_H, and net long-wave radiation Q_B are treated as a surface boundary condition:

$$(\nu_t' + \nu')\partial_z T = \frac{-Q_E - Q_H - Q_B}{C_p \rho_0} \qquad z = \zeta. \tag{2.16}$$

Discussing the parameterizations of the processes is beyond the scope of this report, the interested reader might study e.g. *Kondo* [1975] or *Rosati and Miyakoda* [1988]. However, for the modelling of the data at the Ocean Weather Stations Papa and November in the Northern Pacific (see section 6.2.3.1), some more details about surface flux parameterizations and solar radiation calculation are given.
It should be mentioned here, that the surface fluxes we use are not calculated interactively, i.e. they do consider a measured instead of a modelled sea surface temperature. There are two reasons for that, a practical and a scientific one. The practical argument is that the fluxes can such be precalculated and read in from files with the advantage that the whole program code is less complex. The scientific argument is that with interactively calculated fluxes, a bad mixed layer parameterization

[1] Since the final model is one-dimensional anyway, active swimming in horizontal directions is assumed negligible compared to flow velocity.

can be hidden. If, for example, a mixed layer parameterization generally calculates too shallow mixed layers, and thus too high sea surface temperatures, this deficiency would be corrected by stronger heat losses through latent and sensitive heat flux and a stronger long wave back radiation. Although this feedback is important in reality, this would contradict the scope of **GOTM** which is demonstrating the performances of various mixed layer models.

2.1.1.2.2 Salinity The molecular diffusivity for salinity S used here is $\nu'' = 1.1 \cdot 10^{-9}\mathrm{m}^2\mathrm{s}^{-1}$. Inner sources or sinks for salinity are not considered.
In the current version of **GOTM**, fresh water (and consequently salinity) flux through the surface is not considered for the salinity equation:

$$(\nu_t' + \nu'')\partial_z S = 0 \qquad z = \zeta. \tag{2.17}$$

However, a freshwater flux could be easily added to the salinity equation, although it is rarely measured with sufficient accuracy.
A simple way of substituting surface salinity fluxes would be to relax the sea surface salinity to measured values. Also this feature is not yet implemented into **GOTM**. However, it should be mentioned that the resulting net surface salinity flux not only depends on the relaxation time, but also on the thickness of the surface layer. Therefore, that method should be applied with care.

2.1.1.2.3 Sediment In the present version of **GOTM**, only uniform, non-cohesive sediments as silt and sand are considered. The inclusion of cohesive sediments into this context will be computationally easy if only the equations for this are known (see e.g. *van Leussen* [1994]). The molecular diffusivity is set to zero in this equation.
The sediment is calculated here as a non-dimensional concentration and is supposed to be conservative. Therefore, inner sources and sinks are not considered.
According to *Zanke* [1977] the fall velocity w_s of sphere shaped sediment particles is a function of grain diameter d, molecular viscosity ν and specific weight $\rho_C = 2650\ \mathrm{kg\,m}^{-3}$, only:

$$w_s = 10\frac{\nu}{d}\left(\left(1 + \frac{0.01 g' d^3}{\nu^2}\right)^{1/2} - 1\right). \tag{2.18}$$

with the reduced gravity

$$g' = g\frac{\rho_C - \rho_0}{\rho_0}. \tag{2.19}$$

At the surface a no-flux condition for sediment is used.
The boundary condition for sediment at the bottom plays a central role in sediment modelling. Here, we follow *Smith and McLean* [1977] who prescribe the bottom sediment concentration as a function of the bottom friction velocity

$$u_*^b = (\nu_t M)^{1/2}\Big|_{z=-H} \tag{2.20}$$

with the shear frequency

$$M = \sqrt{(\partial_z u)^2 + (\partial_z v)^2} \tag{2.21}$$

and a critical friction velocity

$$u_*^c = \begin{cases} \dfrac{4w_s}{d_*} & \text{for } 1 < d_* \leq 10, \\ 0.4w_s & \text{for } d_* > 10. \end{cases} \tag{2.22}$$

In (2.22),

$$d_* = d\left(\frac{g'}{\nu^2}\right)^{1/3} \tag{2.23}$$

denotes the so-called sedimentologic grain diameter.
The bottom sediment concentration is then calculated as

$$C_0 = \gamma_1 \left[\left(\frac{u_*^b}{u_*^c}\right)^2 - 1\right]^+ \tag{2.24}$$

where the exponent $^+$ denotes that all negative values will be converted to zero. Here, $\gamma_1 = 1.56 \cdot 10^{-3}$ is chosen.
For the numerical treatment, the same problem as already for the bottom boundary condition for momentum arises. The boundary value C_0 is located directly at the bottom ($z = -H$) while the lowest discrete value for C, C_b, is located at the centre of the lowest layer. In order to achieve this, an idealized sediment profile for the lowest layer has to be evaluated there (see section 3.4). Therefore, a bottom roughness length z_c, which is relevant for this profile, is needed. According to *Smith and McLean* [1977] this quantity can be expressed as:

$$z_c = a_0 \frac{(u_*^b)^2}{g'} \left[1 - \left(\frac{u_*^c}{u_*^b}\right)^2\right]^+ \tag{2.25}$$

with $a_0 = 26.3$.
At the surface, a no-flux boundary condition for sediment is prescribed:

$$\nu_t' \partial_z C = 0 \qquad z = \zeta. \tag{2.26}$$

2.1.1.2.4 Equation of state The equation of state which is calculated in **GOTM** has the following form:

$$\rho = (1 - C)\rho_{T,S,p} + C\rho_C, \tag{2.27}$$

where $\rho_{T,S,p}$ is the well-known UNESCO formula for sea water (see *Gill* [1982]).
If $T = 10°$C, S=30 psu and $C = 0$ is taken as a basic state, then the density increase equivalent to an decrease of T by 1°C is an increase of S by 0.199 psu and an increase of C by $9.54 \cdot 10^{-5}$.

2.1.1.2.5 Buoyancy For some basic theoretical studies (see section 6.1.2), it is useful to have one single transport equation for the buoyancy b (which is in fact a normalized density) only. Such an equation could be achieved for example by combining the tracer equations for temperature T and salinity S into one, if the expansion coeffients $\partial_T b$ and $\partial_S b$ are assumed to be constant (see *Burchard* [1995], p. 75). Another (weak) condition would be that for T and S the same molecular diffusivity would have to be chosen. In the present version of **GOTM**, this molecular diffusivity is set to zero. A surface buoyancy flux may be considered as a boundary condition (not implemented yet in **GOTM**).

2.1.2 One-dimensional form

As **GOTM** is substantially a one-dimensional model, which has been designed to study water column models for vertical turbulent exchange, all equations from section 2.1.1 have to be transformed into one dimension along the z-axis. This is simply done by parameterizing the horizontal derivatives ∂_x and ∂_y of the needed variables as far as possible and assuming horizontal homogeinity (i.e. neglecting horizontal derivatives) for the other variables. In some cases, the derivatives may also be measured in-situ or estimated from three-dimensional models. In this section, it is shown how this can be done.
At sites where the variation of water depth plays a substantial role like in shallow tidal estuaries, it needs to be measured and directly inserted into the model. Such measurements may be available from nearby tidal gauges, local soundings or the pressure sensor of CTD-probes etc.
Such a one-dimensional model is of course limited to fairly simple flow geometries. However, the better the measurements of forcing parameters and horizontal gradients are, the more complex situations can be simulated. The examples discussed in section 6.3 show the potential of one-dimensional models even in estuaries with complex bathymetry.

2.1.2.1 Hydrodynamic equations

The terms in the hydrodynamic equations (2.3) and (2.4) which contain horizontal derivatives are the horizontal advection and the (external and internal) pressure gradient, see (2.6) and (2.7). The external part of the pressure gradient is the main driving force for momentum in estuarine tidal flow as well as in many shelf seas.
GOTM allows for three methods for considering surface slopes. First, the surface slope may directly be inserted into the equations. This would be useful for idealized studies or in the case when the slopes are available from a two-dimensional (horizontal) or a three-dimensional model (see e.g. *Davies* [1990], *Simpson et al.* [1996]) or may be estimated by physical arguments (see e.g. *Luyten et al.* [1996a]).
The second method is based on velocity measurements. Knowing the time series of velocities u and v at a fixed position (x_m, y_m, z_m) from a current meter, ADCP (Acoustic Doppler Current Profiler), or ADV (Acoustic Doppler Velocimeter) measurements, implies the knowledge of time derivatives $\partial_t u$ and $\partial_t v$ at this position.
If the simplified momentum equations,

$$\partial_t u - \partial_z(\nu_t \partial_z u) = -g\partial_x \zeta + fv, \tag{2.28}$$

$$\partial_t v - \partial_z(\nu_t \partial_z v) = -g\partial_y \zeta - fu. \tag{2.29}$$

(without the advective terms) are considered, then the first terms on the left hand sides are known from measurements for $z = z_m$, while the vertical diffusion and Coriolis rotation terms can be calculated inside the 1D model. In principle, if advection of momentum does not play a major role, such a point measurement of velocity can be thus considered as an indirect measurement of the surface slope.
The solution of the momentum equations will be carried out in three steps by means of an operator splitting (see *Burchard* [1999]). The first step is the calculation of rotation, the second step calculates vertical turbulent mixing and the third step considers the external pressure gradient calculation. In a numerical twin experiment, *Burchard* [1999] shows that such a one-dimensional model, if driven with currents $u(x_m, y_m, z_m)$ and $v(x_m, y_m, z_m)$ calculated by a three-dimensional estuarine model, is able to recalculate the surface slopes from the three-dimensional model with

some accuracy, if the same parameters (depth, rotation, bottom roughness etc.) and the same turbulence model are chosen. When simulating in-situ measurements, such an agreement is of course (due to model and measurement errors) difficult to achieve. However, this method guarantees that the modelled velocities at (x_m, y_m, z_m) are identical with the measurements.
A third method for calculating surface slopes is to use time series of vertical mean current velocity instead of velocities at a certain height. In analogy to the previously discussed method, it can be guaranteed that measured and modelled vertical mean velocities are always identical. This feature might be useful for idealized, channel flow or lake studies, or if vertical mean velocities can be extracted from vertically integrated models.
Internal pressure gradients are not yet considered in **GOTM** but it is principally possible to directly insert estimates or measurements (e.g. from two different CTD stations along an estuarine flow) of $\partial_x b$ and $\partial_y b$ as a forcing.
Burchard [1999] tried to estimate horizontal momentum advection from modelled horizontal velocities, water depth and surface and bottom slopes. Unfortunately, this method proved only promising in simple two-dimensional flows, but failed for more complex situations. Therefore, momentum advection is neglected in **GOTM**.
However, assuming that rotation and vertical turbulent mixing are modelled with high accuracy, and current measurements for $u(x_m, y_m, z_m)$ and $v(x_m, y_m, z_m)$ are of good quality, then the vertical mean effects of advection and internal pressure gradient will be included as well (see *Burchard* [1999]).

2.1.2.2 Tracer equations

In the tracer equations, only the advective terms contain horizontal derivatives. There are some applications imaginable where neglect of tracer advection is justified, e.g. in the ocean mixed layer far away from the coast (see sections 6.2.1, 6.2.2 and 6.2.3.1). In estuarine flows, such an assumption generally leads to significant inaccuracies. Therefore, **GOTM** allows for a relaxation of tracers to prescribed tracer profiles. This is carried out by means of a source term on the right hand side of the considered equation for the tracer A:

$$R(A) = -\frac{1}{\tau_R(A)}(A - A_m) \tag{2.30}$$

where $\tau_R(A)$ is the relaxation time scale, and A_m is a prescribed tracer profile. The shorter the relaxation time scale the more direct the forcing of the tracer profile towards the prescribed profile is.
This approach is not without problems. Only if the other relevant processes (e.g. surface fluxes, inner sources and sinks) are modelled with high accuracy, it is sure that the relaxation mainly parameterizes the advection. Otherwise, the effect of bad physics or input data would be corrected in an unclear way.
Another problem is that the relaxation implies a time lag of the modelled data A with respect to the measurements A_m. Moreover, this time lag seems to be scale selective in a sense that the tracer equation responds faster to the high than to the low frequency of the measurements.
Therefore, the choice of the relaxation time scale is a matter of tuning.
However, relaxation of measured profiles into the model is still much better than a direct insertion ($\tau_R(A) \to 0$), which would strongly limit the prognostic evolution of the model.
A much more efficient but very complex way of advection parameterization would be data assimilation. It might be interesting for the future to add such a feature to **GOTM**.

2.2 Turbulence models

A standard approach for turbulence parameterization is the so-called eddy viscosity principle, which has been adopted here. It relates the turbulent fluxes to the gradient of the transported property by means of an eddy viscosity (for momentum) or eddy diffusivity (for tracers).
For calculating eddy viscosity/diffusivity, we apply for all turbulence models presented here the relation of *Kolmogorov* [1942] and *Prandtl* [1945], which relates these turbulent exchange coefficients to the product of a velocity scale and a length scale:

$$\nu_t = c_\mu \sqrt{k} L \tag{2.31}$$

and

$$\nu'_t = c'_\mu \sqrt{k} L, \tag{2.32}$$

where k is the turbulent kinetic energy (TKE) in $\mathrm{J\,kg^{-1}}$=$\mathrm{m^2 s^{-2}}$, L the turbulent macro length scale in m, and c_μ and c'_μ are the dimensionless so-called stability functions.
For the relations (2.31) and (2.32), different forms are found in the literature. In the well known publications of *Mellor and Yamada* [1974, 1982], the TKE is substituted by the turbulence intensity $q = \sqrt{2k}$. Therefore, the eddy viscosities and stability functions in (2.31) and (2.32) correspond to K_M, K_H, S_M, and S_H used in *Mellor and Yamada* [1974, 1982] in the following way:

$$K_M = \nu_t \quad K_H = \nu'_t \quad S_M = c_\mu / 2^{1/2} \quad S_H = c'_\mu / 2^{1/2}. \tag{2.33}$$

In the k-ε literature (see *Rodi* [1980, 1987]), the length scale is substituted by the normalized dissipation rate which has here the unit $\mathrm{W\,kg^{-1}} = \mathrm{m^2 s^{-3}}$:

$$\varepsilon = (c^0_\mu)^3 \frac{k^{3/2}}{L} \tag{2.34}$$

with the constant $c^0_\mu = 0.5562$. This has the consequence that the stability functions in the literature of *Rodi* [1980, 1987] are by the factor of $(c^0_\mu)^3$ smaller than those given here in this report. The dissipation rate ε is also an interesting physical quantity in the field of turbulence, since it can be directly measured (see e.g. *Simpson et al.* [1996], *Prandke and Stips* [1996]).
In principle, k, L and the stability functions can be calculated independently from each other. Therefore, we designed a section for each of them. It should however be considered that standard turbulence models including each a set of k, L, c_μ and c'_μ are calibrated and even only slight deviations from that might have the consequence that the calibration is not valid any more.
For the TKE calculation see section 2.2.1, for the length scale calculation see section 2.2.2, and for the stability functions see section 2.2.3.
Other important parameters in turbulence modelling are the shear production

$$P = \nu_t M^2 \tag{2.35}$$

and the buoyancy production

$$B = -\nu'_t N^2 \tag{2.36}$$

with the shear frequency M from (2.21) and the Brunt-Väisälä frequency N defined by

$$N^2 = \partial_z b. \tag{2.37}$$

2.2.1 Turbulent kinetic energy

Most of the models found in literature are based on the same transport equation for turbulent kinetic energy. They differ mainly in the treatment of the eddy diffusivity for the TKE. The full transport equation is introduced in section 2.2.1.1, its algebraic idealization in section 2.2.1.2.

2.2.1.1 Transport equation

A three-dimensional form of the TKE equation will not be given here, because even in many three-dimensional models, turbulence models are used as local water column models where horizontal advection and diffusion as well as vertical advection are neglected. Such terms would only be of importance near sharp fronts which cannot be resolved in most of the present models. The reason for this is that the time scales of turbulence are generally much shorter than those of the mean flow which drives advection and horizontal diffusion.
In the one-dimensional form, the TKE equation reads then as:

$$\partial_t k - \partial_z(\nu_k \partial_z k) = P + B - \varepsilon, \tag{2.38}$$

Under only a few modelling assumptions, this equation can be derived from the Navier-Stokes equations (for a full derivation see e.g. *Rodi* [1980]).
The eddy diffusivity relevant for the TKE is calculated as $\nu_k = \nu_t$ for the k-ε and as $\nu_k = 0.283\sqrt{k}L$ for the Mellor-Yamada-model.
Two theoretically equivalent sets of boundary conditions for k, Dirichlet and flux conditions can be derived from the logarithmic law of the wall (see section 3.2). The Dirichlet conditions are easily derived from the assumption $P = \varepsilon$ in the boundary layers. By using the definition of the bottom friction velocity u_*^b from (2.20) and combining it with the relation of Prandtl and Kolmogorov (2.31) and (2.32) and the definition of ε (2.34), this boundary condition can be written as:

$$k = \left(\frac{u_*^b}{c_\mu^0}\right)^2, \qquad z = -H. \tag{2.39}$$

The analogue surface boundary condition reads as

$$k = \left(\frac{u_*^s}{c_\mu^0}\right)^2, \qquad z = \zeta \tag{2.40}$$

with the surface friction velocity $u_*^s = ((\tau_x^s/\rho_0)^2 + (\tau_y^s/\rho_0)^2)^{1/4}$.
Assuming that there is a boundary layer each at the surface and at the bottom with $P = \varepsilon$ and a constant in space value of k, leads to the no-flux conditions

$$\nu_k \partial_z k = 0, \qquad z = -H \tag{2.41}$$

and

$$\nu_k \partial_z k = 0, \qquad z = -H. \tag{2.42}$$

Generally, the Dirichlet conditions are used with the Mellor-Yamada, and the flux conditions with the k-ε model. An advantage of the flux boundary condition for k is the easy treatment of turbulence near the stress-free surface (see section 6.1.1). The Dirichlet conditions would lead to an eddy viscosity profile which is not decreasing towards the surface. The choice of the boundary

conditions is even more important for the length scale equation (see the discussion in sections 2.2.2.1.1 and 3.5). The current version of **GOTM** allows for both sets of boundary conditions.
It should be noted here that the effect of breaking surface waves on near surface mixing is not yet included in **GOTM**. Following the approach from *Craig and Banner* [1994], a flux of turbulent kinetic energy proportional to the surface friction velocity cubed could be considered for this purpose. They obtained a good approximation to measured near surface dissipation rates with a one-equation Mellor-Yamada model, where the near surface macro length scale increases linearly with depth. Our experiments with the two-equation Mellor-Yamada model and with the k-ε model gave such different results, that we were reluctant to include the *Craig and Banner* [1994] approach at this stage of **GOTM**.

2.2.1.2 Algebraic form

From the local equilibrium assumption,

$$P + B = \varepsilon, \tag{2.43}$$

i.e., that production and dissipation of turbulent kinetic energy are in balance, an algebraic form for calculating TKE can easily obtained by using (2.34), (2.31), and (2.32). Together with a lower limit which is necessary in order to avoid unphysical negative values for the turbulent kinetic energy, k is of the following form:

$$k = \max\left\{\frac{L^2}{(c_\mu^0)^3}(c_\mu M^2 - c'_\mu N^2), k_{\min}\right\}. \tag{2.44}$$

2.2.2 Turbulent length scale

The turbulent length scale has always been subject to empirical considerations. One of its properties which is agreed about among turbulence modellers, is the linear decrease towards a fixed boundary:

$$L = \kappa(\tilde{z} + z_0) \tag{2.45}$$

with the von Kármán constant $\kappa = 0.4$, the distance from the boundary, $\tilde{z}$, and the roughness length of the boundary, z_0. All length scale parameterizations described here will follow that law near fixed boundaries, most of them as well near the surface. It should be already noted here, that the k-ε model is the only one which automatically (by only fixing boundary conditions) fulfills this law.

2.2.2.1 Transport equations

In oceanography, mainly two different transport equations for a length scale related variable are in use, the ε-equation in the k-ε and the kL-equation in the Mellor-Yamada model. The structural similarity of these two equations has been shown by *Petersen* [1996], *Burchard et al.* [1998] and *Burchard and Petersen* [1999].
Since the late seventies, a controverse discussion about the higher physical relevance of each of the two length scale equation has been carried out. The arguments of the main protagonists of this discussion are cited here:

Mellor and Yamada [1982]:

> While one cannot assert great confidence in [the kL equation], we prefer it rather than the differential equation for dissipation ...
> ... it seems fundamentally wrong to us to use an equation which describes the small scale turbulence to determine the turbulent macro-scale. Operationally, however, after some terms are modelled, the dissipation transport equation is a special case of a more general length scale equation ...

Rodi [1987]:

> The arguments for the relative merits of the ε and the kL equations are rather academic because both equations are fairly empirical and, with the constants suitable adjusted, perform in a similar manner. One difference is that the kL equation requires an additional near-wall term ... while the ε equation does not.

It is beyond the scope of this report to be more in favour of either or the other argument. However, on a visit to the Joint Research Centre in April 1998, George Mellor stated that the k-ε model, as it is implemented in **GOTM**, is a Mellor-Yamada-model as well. He said this because we adopt here most of the derivations motivated in *Mellor and Yamada* [1974, 1982], and (besides using other stability functions) only substitute the length scale equation suggested by them by an equation for the dissipation rate ε.

2.2.2.1.1 The ε equation Similarly to the TKE-equation (2.38), an exact transport equation for the dissipation rate ε of TKE can be derived from the Navier-Stokes equations. However, a mathematical closure of this equation is only achieved after a number of simplifying model assumptions with the consequence that three new empirical parameters have to be introduced:

$$\partial_t \varepsilon - \partial_z(\nu_\varepsilon \partial_z \varepsilon) = \frac{\varepsilon}{k}\left(c_{\varepsilon 1} P + c_{\varepsilon 3} B - c_{\varepsilon 2}\varepsilon\right). \tag{2.46}$$

By means of equation (2.34), the macro length scale L can now be calculated. The eddy diffusivity for ε in equation (2.46) is modelled as:

$$\nu_\varepsilon = \frac{\nu_t}{\sigma_\varepsilon} \tag{2.47}$$

with $\sigma_\varepsilon = 1.08$ (see equation (3.18) and *Burchard et al.* [1998]).
The values $c_{\varepsilon 1} = 1.44$ and $c_{\varepsilon 2} = 1.92$ are adopted from *Rodi* [1980]. It should be noted that the buoyancy production related parameter $c_{\varepsilon 3}$ in the ε-equation used here deviates from the standard literature (see e.g. *Rodi* [1987]) where this value is non-negative. The demand for a negative $c_{\varepsilon 3}$ has recently been shown by *Burchard and Baumert* [1995] and *Burchard et al.* [1998], but the actual value is sensitive to the stability function chosen. *Burchard et al.* [1998] calibrated $c_{\varepsilon 3}$ by means of a numerical experiment similar to the Kato-Phillips experiment (see section 6.1.2). When using the *Galperin et al.* [1988] stability functions, $c_{\varepsilon 3} = -0.4$ gave the best agreement between theoretically derived and simulated entrainment velocity.
A negative value for $c_{\varepsilon 3}$ provides a source for dissipation in the case of stable stratification. This has a similar effect than the lower limit for ε, equation (2.72), which is included into the model in order to parameterize the effect of transformation of turbulent eddies into internal waves. For unstable stratification, the classical value $c_{\varepsilon 3} = 1.0$ is used here.
By combining (2.34) and (2.45), a Dirichlet boundary condition for ε can be derived,

$$\varepsilon = \left(c_\mu^0\right)^3 \frac{k^{3/2}}{\kappa(\tilde{z} + z_0)}. \tag{2.48}$$

By differentiating (2.48) with respect to $\tilde{z}$ and considering (2.41) and (2.42), an equivalent flux boundary condition for ε can be achieved[2]:

$$\frac{\nu_t}{\sigma_\varepsilon}\partial_{\tilde{z}}\varepsilon = -\left(c_\mu^0\right)^3 \frac{\nu_t}{\sigma_\varepsilon}\frac{k^{3/2}}{\kappa(\tilde{z}+z_0)^2} \tag{2.49}$$

where $\tilde{z}$ is again the distance from bottom or surface, respectively. The reason for considering a flux condition for ε are the numerical problems caused by high gradients near the boundaries (see *Stelling* [1995], *Burchard and Petersen* [1999]). The different numerical performance of ε boundary conditions (2.48) and (2.49) in boundary layer flow will be discussed in section 3.5.

2.2.2.1.2 The kL equation Even more than the ε-equation (2.46), the kL-equation which has been designed by *Mellor and Yamada* [1982] is a result of many empirical considerations. After adapting the notations to those used here in this report, the kL-equation reads as:

$$\partial_t(kL) - \partial_z\left(\nu_L\partial_z(kL)\right) = L\left(c_{L1}P + c_{L3}B - \left(1 + E_2\left(\frac{L}{L_z}\right)^2\right)\varepsilon\right). \tag{2.50}$$

The empirical parameters in this equations are $c_{L1} = c_{L3} = 0.9$ and $E_2 = 1.33$. In (2.50), a wall proximity function with a barotropic length scale L_z has to be prescribed in order to guarantee the log-law near boundaries, see sections 3.2.1 and 3.2.2 and *Mellor and Yamada* [1982]. Two possible profiles which are both included in **GOTM** have been tested by *Burchard et al.* [1998], a parabola shaped one:

$$L_z = \kappa\frac{(d_b + z_0^b)(d_s + z_0^s)}{(d_b + z_0^b) + (d_s + z_0^s)}, \tag{2.51}$$

and a triangle shaped one:

$$L_z = \kappa\min(d_b + z_0^b, d_s + z_0^b). \tag{2.52}$$

Here, d_s is the distance from the surface and d_b the distance from the bottom.
As boundary values for L, the condition (2.45) is used:

$$L = \kappa(d_b + z_0^b), \qquad z = -H. \tag{2.53}$$

$$L = \kappa(d_s + z_0^s), \qquad z = \zeta. \tag{2.54}$$

2.2.2.2 Algebraic forms

Instead of solving a transport equation to obtain the turbulent mixing length, it is possible to calculate it from a simple geometric or a more complex diagnostic expression.

[2]We are grateful to Eckard Kleine (Bundesamt für Seeschifffahrt und Hydrographie, Hamburg, Germany) for suggesting us to use flux boundary conditions for the dissipation rate ε.

2.2.2.2.1 Simple geometric forms Geometric expressions are designed in a way that they fulfill the logarithmic law of the wall (2.45) near boundaries. Two simple expressions that satisfy this requirement in the presence of surface and bottom boundaries are the parabolic

$$L = \kappa \frac{(d_b + z_0^b)(d_s + z_0^s)}{(d_b + z_0^b) + (d_s + z_0^s)} \tag{2.55}$$

and the triangle profile

$$L = \kappa \min(d_b + z_0^b, d_s + z_0^s), \tag{2.56}$$

where d_b and d_s are the distances to bottom and surface, respectively, and z_0^b and z_0^s are the roughness lengths of bottom and surface boundaries, respectively (see (2.1) and (2.2)). Other expressions take into account that the position of the maximum in mixing length may not lay in the middle of the water column, as observed to occur in tidal flows. This anisotropy in the profile is obtained by changing the functional form of surface or bottom mixing length, as suggested by *Robert and Ouellet* [1987] (note that this function is a distorted parabola)

$$L = \kappa(d_b + z_0^b) \left(1 - \frac{d_b}{D}\right)^{1/2} \tag{2.57}$$

or *Xing and Davies* [1995]

$$L = \kappa(d_b + z_0^b) e^{-\beta d_b / D}, \tag{2.58}$$

where β is a tuning parameter. For (2.57) and (2.58), the logarithmic law of the wall (2.45) is not any more fulfilled for the surface. This is in order to take into account free surface effects like breaking waves. Other authors suggest that the turbulent mixing length should depend on local characteristics of the turbulent flow like TKE. The most widespread of these methods is that of *Blackadar* [1962]

$$L_a = \left(\frac{1}{\kappa(d_b + z_0^b)} + \frac{1}{L_0}\right)^{-1} \tag{2.59}$$

with L_0 given by the expression:

$$L_0 = \gamma_0 \frac{\int_{-H}^{\zeta} k^{1/2} z \, dz}{\int_{-H}^{\zeta} k^{1/2} \, dz}, \tag{2.60}$$

where γ_0 is a constant in the range from 0.1 to 0.4. In the presence of two boundaries, the Blackadar mixing length can be written in a form that automatically gives an approximation to a linear profile of L near walls:

$$L = \left(\frac{1}{\kappa(d_s + z_0^s)} + \frac{1}{L_a}\right)^{-1}. \tag{2.61}$$

with L_a from (2.59).
For stratified flow, the mixing length is also affected by stratification. A way of including stratification into these simple geometric expressions for the mixing length is multiplying it by a stratification correction factor,

$$L \quad \longrightarrow \quad LF_s. \tag{2.62}$$

Some proposed forms for this factor are based on measurements of the atmospheric boundary layer, *Geernaert* [1990]:

$$F_s = \begin{cases} 1 - \alpha R_i, & \text{for } R_i > 0 \\ (1 - \beta R_i)^{1/4}, & \text{else} \end{cases} \tag{2.63}$$

with $R_i = N^2/M^2$, the gradient Richardson number and the parameters α (which can range from 5 to 10) and $\beta = 15$.
Nihoul and Djenidi [1987] have suggested

$$F_s = 1 - R_f \tag{2.64}$$

with the flux Richardson number $R_f = -B/P$. Stability functions (introduced in section (2.2.3)) have a similar effect as this correction factor: they increase mixing for unstable stratification and decrease it for stable stratification. This makes it difficult to calibrate a model that uses a stratification-depending functional form of both. Moreover, a calibration of the model with a given mixing length corrected by a stratification factor may no longer be valid if stability functions are changed.

2.2.2.2.2 Complex forms A more complex diagnostic mixing length is the one proposed by *Therry and Lacarrère* [1983] and adapted for **GOTM** from *Blanke and Delecleuse* [1993]. Two characteristic lengths, one for dissipation, L_ε, and another for mixing, L_k, are defined as combinations of two master length scales L_u and L_d obtained in each point as the upward or downward distance that a fluid particle placed at that point must travel to convert TKE in potential energy:

$$\begin{aligned} \int_z^{z+L_u(z)} (b(z) - b(\xi))\, d\xi &= k(z) \\ \int_{z-L_d(z)}^{z} (b(\xi) - b(z))\, d\xi &= k(z). \end{aligned} \tag{2.65}$$

L_k and L_ε are then calculated as

$$\begin{aligned} L_k(z) &= \min(L_d(z), L_u(z)) \\ L_\varepsilon(z) &= (L_d(z) L_u(z))^{1/2}. \end{aligned} \tag{2.66}$$

The expressions for L_k and L_ε are designed such that they take into account the fact that a strongly stratified region affects diffusion as a wall does, while dissipative phenomena, even near a wall, are affected by the maximum vertical scale of convective motions. Deviating from the usual structure of **GOTM**, in this model, the dissipation rate (2.34) and the eddy viscosity (2.31) and diffusivity (2.32) are calculated with different length scales:

$$L = \begin{cases} \frac{c_\mu}{\tilde{c}_\mu} L_k & \text{for insertion into (2.31),} \\ \frac{c'_\mu}{\tilde{c}_\mu} L_k & \text{for insertion into (2.32),} \\ \frac{c_\varepsilon}{(c^0_\mu)^3} L_\varepsilon & \text{for insertion into (2.34),} \end{cases} \tag{2.67}$$

where $c_\varepsilon = 0.7$ and $\tilde{c}_\mu = 0.1$ are constants (see *Gaspar et al.* [1990]).
In the algebraic mixing length parameterization used in the ISPRAMIX ocean circulation model (see *Eifler and Schrimpf* [1992] and *Demirov et al.* [1998]), three different regions are distinguished:

the top and the bottom mixed layers and a stably stratified interior layer. After calculating the height h_m of each mixed layer (obtained as the first point from the boundary with $k < k_{min} = 10^{-5}$ W kg^{-1}), the macro length scale L in both mixed layers is obtained from a *Blackadar* [1962] type formula:

$$L = \frac{\kappa\tilde{z}}{1 + \frac{\kappa\tilde{z}}{c_2 \cdot h_m}}(1 - R_f)^e \tag{2.68}$$

where $\tilde{z}$ is the distance from the interface (surface or bottom). Equation (2.68) predicts an approximation to a linear behavior of L near boundaries (see eq. (2.45)) and a value proportional to the thickness of the mixed layer far from the interface, $L = c_2 h_m$, where $c_2 = 0.065$ is estimated from experimental data as discussed in *Eifler and Schrimpf* [1992]. The factor $(1 - R_f)$, with the flux Richardson number $R_f = -B/P$, accounts for the effect of stratification on the length scale. The parameter e is here a tuning parameter (pers. comm. Walter Eifler, JRC, Ispra, Italy) which is usually set to $e = 1$. However, during the simulations of the Kato-Phillips experiment, we found that a higher value was necessary in order to predict the expected mixed layer depth evolution (see section 6.1.2).
Below the top mixed layer and above the bottom friction layer, a suggestion of *Zilitinkevich and Mironov* [1992] is applied to calculate the length scale in the interface L_i between the thermocline and the mixed layer. This formula parameterizes the penetration of the mixed layer into the stably stratified thermocline by turbulence diffusion:

$$L_i = c_i \frac{k^{1/2}}{N}. \tag{2.69}$$

The constant c_i is determined for every specific case (and for both boundary layers) by making use of the requirement that the length scale in the mixed layer L_m and near the thermocline must match at $z = h_m$. Therefore, we have a matching condition to determine c_i for each boundary layer:

$$c_i = \left(\frac{L_m N}{k^{1/2}}\right)_{z=h_m}. \tag{2.70}$$

In the core region of the thermocline, a constant value of L typical for stratified conditions is imposed, where (2.70) would yield too small values. In **GOTM**, we use $L = 0.01$ m. Care has been taken that the model behaves well in every possible situation. Thus, if the two boundary layers overlap, we suppose there is no stably stratified region, only two boundary layers, with L calculated from (2.68). The same is supposed if only one mixed layer is detected. This way, we guarantee that the length scale satisfies the linear profile near both boundaries.

2.2.2.3 Length scale limitation

In order to include the limiting effect of stable stratification, *Galperin et al.* [1988] found it necessary to introduce an upper limit for the macro length scale L in stably stratified flows,

$$L^2 \leq \frac{0.56k}{N^2} \qquad \text{for } N^2 > 0. \tag{2.71}$$

Applied to the ε-equation, this corresponds to a lower limit for the dissipation rate ε, which can be calculated by means of using (2.45) as

$$\varepsilon^2 \geq 0.045 k^2 N^2 \qquad \text{for } N^2 > 0. \tag{2.72}$$

In section 6.1.2, the effect of these limitations for a simple mixing experiment is shown.

2.2.3 Stability functions

In the literature many different sets of stability functions c_μ and c'_μ are found (see equations 2.31 and 2.32) . They have the function to increase mixing for instable and decrease mixing for stable stratification.

2.2.3.1 Simple stability functions

The simplest set of stability function considers the stability function for momentum, c_μ, as a constant. The stability function for tracers, c'_μ, is related to c_μ via the Prandtl number P_r:

$$c_\mu = c^0_\mu = \text{const (see (2.34))} \tag{2.73}$$

$$c'_\mu = \frac{c_\mu}{P_r}. \tag{2.74}$$

For the very simple case, the Prandtl number is a constant of the order of unity. A gradient Richardson number depending approach had been suggested by *Munk and Anderson* [1948]:

$$P_r = \begin{cases} P^0_r \frac{(1+3.33R_i)^{3/2}}{(1+10R_i)^{1/2}}, & \text{for } R_i \geq 0, \\ P^0_r, & \text{for } R_i < 0 \end{cases} \tag{2.75}$$

with the constant Prandtl number for neutral stratification, P^0_r.
In the ISPRAMIX ocean model (see section 2.2.2.2.1), another approach is used for considering stability effects on vertical mixing. The stability functions in this model are of the form:

$$c_\mu = \text{const} = 0.5, \tag{2.76}$$

$$c'_\mu = c_\mu f(R_f) = c_\mu \frac{1}{P^0_r}(1 - R_f)^{1/2}. \tag{2.77}$$

The neutral Prandtl number used there is $P^0_r = 0.7143$. The function $f(R_f)$ is assumed to lay between the values 0.18 (corresponding to a super-critically stratified situation) and 2.0 (preventing it from growing too much under unstable conditions).
A formulation for $(1 - R_f)$ can be derived from the definition of the flux Richardson number

$$R_f = \frac{c'_\mu}{c_\mu} R_i \tag{2.78}$$

and (2.77) (see *Beckers* [1995]):

$$(1 - R_f) = \left[(\tilde{R}_i^2 + 1)^{1/2} - \tilde{R}_i\right]^2 \tag{2.79}$$

with

$$\tilde{R}_i = \frac{0.5}{Pr} R_i \tag{2.80}$$

where R_i is the gradient Richardson number.

2.2.3.2 *Mellor and Yamada* [1974]

Mellor and Yamada [1974] derived stability functions from prognostic equations for Reynolds stresses and heat fluxes under the assumption of a local equilibrium of stresses and heat fluxes. They obtained the following system of equations for c_μ and c'_μ which is for simplicity here given with coefficients readily calculated from the empirical parameters used:

$$\begin{array}{rcrcrcl} 2.0424\alpha_M & c_\mu & + & (1+15.2958\alpha_N) & c'_\mu & = & 1.0465 \\ (1+2.5392\alpha_M+3.0636\alpha_N) & c_\mu & + & 8.1142\alpha_N & c'_\mu & = & 0.9888. \end{array} \tag{2.81}$$

with the non-dimensional shear

$$\alpha_M = \frac{L^2}{k} M^2. \tag{2.82}$$

and the non-dimensional buoyancy parameter

$$\alpha_N = \frac{L^2}{k} N^2. \tag{2.83}$$

With the constraints

$$\alpha_N \geq -0.064 \tag{2.84}$$

and

$$\alpha_M \leq 1.65 + 25\alpha_N, \tag{2.85}$$

the solutions for c_μ and c'_μ are unique and positive (see figure 2.2). If one of the constraints (2.84) or (2.85) is violated, α_M or α_N will be set to the threshold value and as such inserted into (2.81). This might lead to numerical instabilities for unstable stratification.

2.2.3.3 *Galperin et al.* [1988]

For the local equilibrium $P + B = \varepsilon$, the following relation between non-dimensional parameters can be directly obtained:

$$c_\mu \alpha_M - c'_\mu \alpha_N = \left(c^0_\mu\right)^3. \tag{2.86}$$

If this is inserted into the exact form of (2.81), then the so-called quasi-equilibrium stability functions are obtained, which only depend on the non-dimensional buoyancy parameter α_N:

$$c_\mu = \frac{c^0_\mu + 2.182\alpha_N}{1 + 20.40\alpha_N + 53.12\alpha_N^2} \tag{2.87}$$

and

$$c'_\mu = \frac{0.6985}{1 + 17.34\alpha_N}. \tag{2.88}$$

Also here, we display the equations with the empirical parameters already inserted. These stability functions are commonly referred to as the *Galperin et al.* [1988] the stability functions.

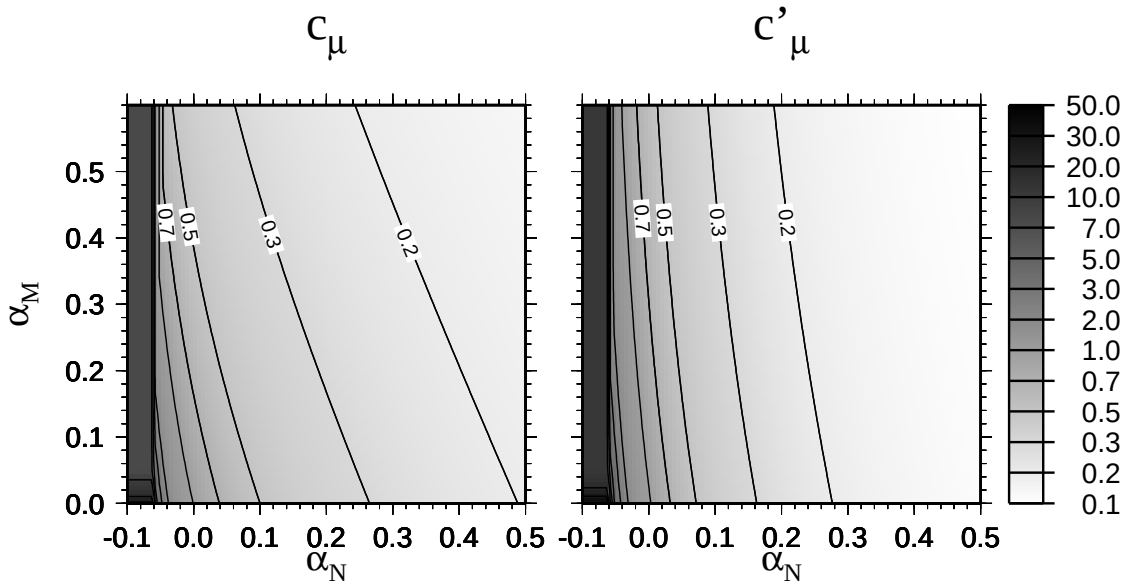


Figure 2.2: Stability functions c_μ (left) and c'_μ (right) according to *Mellor and Yamada* [1974].

The constraint for α_N given by *Galperin et al.* [1988] is:

$$\alpha_N > \alpha_{\min} = -0.0466. \tag{2.89}$$

In the case of convection, much smaller values of α_N may occur. In order to guarantee a smooth transition into this convective regime, α_N is here limited by

$$\alpha_N = \max\{\tilde{\alpha}_N, \tilde{\alpha}_N - (\tilde{\alpha}_N - \alpha_c)^2/(\tilde{\alpha}_N + \alpha_{\min} - 2\alpha_c)\} \tag{2.90}$$

with $\tilde{\alpha}_N = L^2N^2/k^2$ (see equation (2.83)) and $\alpha_c > \alpha_{\min}$ (see *Burchard and Petersen* [1999]). For free convection simulations, it was sufficient to use $\alpha_c = -0.02$. An upper limit for α_N is given by $\alpha_{\max} = 0.56$ (see *Galperin et al.* [1988]). The stability functions c_μ and c'_μ are shown in figure 2.3 for different values of α_c.

2.2.4 Shear instability and IW parameterization

A major weakness of local turbulence models is that they do not take into account the increasing effect of mixing that shear instability and internal wave activity induce in the presence of stable stratification. Some simple methods to model this effect in the framework of a second-moment closure model have been introduced in **GOTM**.

2.2.4.1 Limitation of turbulent magnitudes

The first obvious way of taking into account stable stratification effects is to impose a limitation on turbulent magnitudes by prescribing an upper limit of the turbulent length scale (or equivalently, a lower limit of the dissipation rate ε). The conditions (2.71), suggested by *Galperin et al.* [1988]

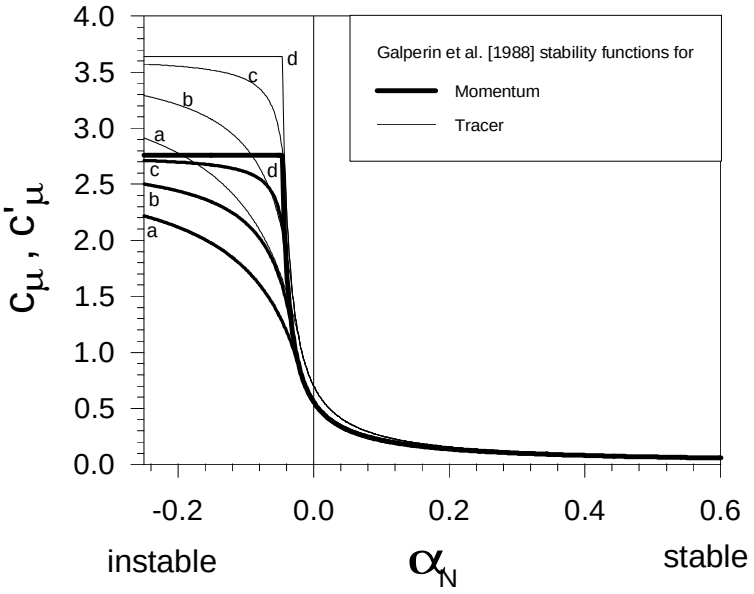


Figure 2.3: Quasi-equilibrium stability functions c_μ and c'_μ according to *Galperin et al.* [1988], but smoothed for unstable stratification. Curves for different critical stratifications are shown (see also eq. (2.90)). $\alpha_c = -0.02$ (a), $\alpha_c = -0.03$ (b), $\alpha_c = -0.04$ (c), unsmoothed (d). This figure has been taken from *Burchard and Petersen* [1999].

and *Luyten et. al.* [1996a], are optionally used in **GOTM**. *Luyten et al.* [1996b] found, that this limitation only has a noticeable effect on results, if a lower limit for the turbulent kinetic energy is set. The oceanic observations by Gregg [1987] suggest that it is reasonable to assume that turbulent kinetic energy tends to a constant limiting value under stably stratified conditions. The actual value of this constant is obtained by comparing measurements and model predictions, and can act as a tuning parameter of the model as illustrated by *Burchard et al.* [1998].

2.2.4.2 *Mellor* [1989]

Mellor [1989] suggests to introduce internal wave mixing as an extra term in the turbulence equations, instead of imposing a limiting condition. The basic idea of the model is that internal waves induce an additional shear that is added to the mean flow shear. As internal wave energy is known to depend on N^2, the internal wave shear is modeled as a term αN_2, which is added to the shear terms (2.35) in both turbulent transport equations. Instead of (2.35), for stable stratification

$$P = \nu_t(M^2 + \alpha N^2) \tag{2.91}$$

is used.
After estimating the difference between observed shear profiles and modeled ones, a value of $\alpha = 0.7$ is suggested.

2.2.4.3 *Kantha and Clayson* [1994]

The mixed-layer model presented in *Kantha and Clayson* [1994] is a second- moment closure model that differs from the standard Mellor-Yamada model in the fact that it incorporates a parameterization of diffusivities below the mixed layer. The stability functions they use are also slightly different and predict extinction of turbulence by stable stratification for a critical flux Richardson number R_i^c of 0.213. These new functions provide a modest increase of mixing under stably stratified conditions. As suggested by *Large et al.* [1994], two different contributions to diffusivities are considered, shear instability-induced (SI) and internal wave-induced (IW) diffusivities.
The shear instability induced viscosities and diffusivities are modeled as a strongly decreasing function of the gradient Richardson number R_i,

$$(\nu_t)^{SI} = (\nu_t')^{SI} = \begin{cases} 0, & \text{for } R_i > 0.7, \\ 5 \cdot 10^{-3} \left(1 - \left(\frac{R_i}{0.7}\right)^2\right)^3, & \text{for } 0 < R_i < 0.7, \\ 5 \cdot 10^{-3}, & \text{for } R_i < 0. \end{cases} \tag{2.92}$$

In contrast to that, it is assumed that internal wave induced viscosities and diffusivities are constant:

$$\begin{aligned} (\nu_t)^{IW} &= 10^{-4}, \\ (\nu_t')^{IW} &= 5 \cdot 10^{-5}. \end{aligned} \tag{2.93}$$

The unit of all diffusivities in (2.92) and (2.93) is $m^2 s^{-1}$. Thus, when the turbulent kinetic energy is small (extinction of turbulence, diagnosed by k smaller than a certain value, usually $10^{-6} m^2 s^{-2}$), instead of using the diffusivities ν_t and ν_t' given by the model, the sum of shear induced and internal wave induced viscosities and diffusivities $(\nu_t)^{SI} + (\nu_t)^{IW}$ is imposed. This model was applied by *Kantha and Clayson* [1994] to different situations and appeared to promote increased mixing in stably stratified situations.

2.3 Outlook

The vertical mixing in all transport equations for mean flow quantities as presented in section 2.1 is based on the eddy viscosity concept. This calculates the turbulent flux of a quantity X as the negative product of an eddy viscosity (for momentum) or eddy diffusivity (for tracers) and the vertical gradient of the quantity to be transported:

$$\langle w'x' \rangle = -\nu_t \partial_z X, \tag{2.94}$$

where $\langle w'x' \rangle$ is the turbulent flux.
The whole turbulence modelling in section 2.2 is furthermore restricted to the relation of Kolmogorov and Prandtl for calculating the eddy viscosity/diffusivity, which is strictly the product of a stability function, the square root of the turbulent kinetic energy and the macro length scale:

$$\nu_t = c_\mu \sqrt{k} L. \tag{2.95}$$

With this limited approach, eddy viscosity concept (2.94) and relation of Kolmogorov and Prandtl (2.95), an immense variety of mixing problems has been simulated in the past, ranging from hydraulic engineering problems (see, e.g., *Rodi* [1980]) to the general circulation of the global ocean (*Rosati and Miyakoda* [1988]).

However, for many applications, simpler or more complex turbulence parameterizations which do not fit into the concept (2.94) and (2.95) are advantageous. This is especially the case when convection due to instable stratification is a relevant process. With the concept presented here, convection is simply parameterized by using high values of eddy viscosity/diffusivity and consequently a strong mixing. In contrast to that, so-called counter-gradient fluxes occur in nature, where the vertical transport is more of an advective type. Such processes in the convective boundary layer are also referred to as non-local processes, because they are (inside a hydrostatic concept) directly linked to the surface fluxes and boundary layer height.
A very popular because simple and economic method is the convective adjustment, where mainly tracers are (within one time step) iteratively mixed wherever a density gradient is instable. This method is still widely applied for general circulation models.
More complex is the application of a non-local flux simply added to the down-gradient flux. This concept has been developed in meteorology (*Deardorff* [1972]) and has become more popular in oceanography in recent years (*Large et al.* [1994], *D'Alessio et al.* [1998]).
A more complex method allowing for counter-gradient fluxes is the application of transport equations for turbulent fluxes (see *Launder et al.* [1975]). However, the disadvantages are an increased computational effort, which would limit the applicability of such parameterizations inside three-dimensional models, and a number of additional empirical parameters. Therefore, two-equation models are generally supposed to be an optimal compromise between computational efficiency and accuracy.
However, direct parameterizations of the eddy viscosity/diffusivity are often applied. These might be simple forms like constant, piecewise linear, or higher polynomial forms as they are often used for tidal flow calculations (see, e.g., *Davies* [1990]). In order to account for stratification, such expressions are often multiplied with an exponential function of the Richardson number (see, e.g., *Perrels and Karelse* [1982]). Also the rather complex KPP model (K-profile parameterization, see *Large et al.* [1994]) is such a direct eddy viscosity/diffusivity parameterization.
All these before-mentioned extensions can easily be added to the current version of **GOTM**. Some of them are already included in test versions.
The role of surface waves is completely neglected in this first version of **GOTM**. Parameterizations for additional near surface transport (Stokes drift), injection of turbulent kinetic energy due to wave breaking (see *Craig and Banner* [1994]), and increased bottom friction due to surface waves in shallow water can be integrated into the framework of **GOTM** later.

Chapter 3

Numerical Treatment

3.1 General

For the discretization the depth is divided into M not necessarily equidistant intervals h_i. The indexing is upwards with $i = 1$ for the bottom layer and $i = M$ for the surface layer. **GOTM** includes an algorithm which allows for zooming of layers towards the bottom and/or the surface (pers. comm. Antoine Garapon, see *Burchard* [1996]):

$$h_i = D\frac{\tanh\left((d_l + d_u)\frac{i}{M} - d_l\right) + \tanh(d_l)}{\tanh(d_l) + \tanh(d_u)} - 1 \tag{3.1}$$

with $d_l, d_u > 0$ and the water depth D. With the aid of (3.1), the vertical discretization can be refined at the surface (large d_u) or at the bottom (large d_l). With $d_u = d_l \to 0$, an equidistant discretization is achieved.
The discrete values for the mean flow quantities u, v, T, and S represent interval means and are therefore located at the centers of the intervals, and the turbulent quantities k, L, ε, ν_t, ν_t', N, P, B, c_μ, and c_μ' are positioned at the interfaces of the intervals. The indexing is such, that the interface above an interval has the same index than the interval itself. This means that mean flow quantities range from $i = 1, .., M$ while turbulent quantities range from $i = 0, .., M$ (see figure 3.1). The staggering of the grid allows for a (in the case of equidistant intervals) second-order approximation of the vertical fluxes of momentum and tracers without averaging. However, for the vertical fluxes of k, L, and ε, averaging of the eddy diffusivities is necessary. This is only problematic for the fluxes near the surface and the bottom, where viscosities at the boundaries have to be considered for the averaging. These can be derived from the boundary values for k and L by applying (2.31) in connection with c_μ^0.
The time stepping is equidistant, based on two time levels and not limited by Courant numbers, because of the absence of advection and a fully implicit treatment of diffusion. In the following, the discretization of a simple diffusion equation

$$\partial_t X - \partial_z(\nu\partial_z X) = 0 \tag{3.2}$$

will be shown with the flux boundary conditions

$$\nu\partial_z X = F_s \qquad z = \zeta, \tag{3.3}$$

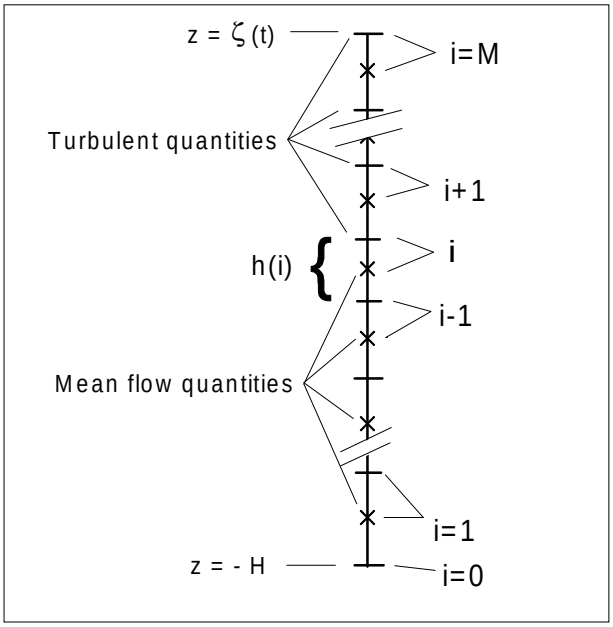


Figure 3.1: Spatial organization and indexing of the numerical grid.

and

$$\nu \partial_z X = F_s \qquad z = \zeta. \tag{3.4}$$

The fully implicit discretization is for a mean flow quantity such as momentum or tracers of the following form:

$$\frac{X_M^{n+1} - X_M^n}{\Delta t} - \frac{F_s - \nu_{M-1}^n \frac{X_M^{n+1} - X_{M-1}^{n+1}}{0.5(h_M^{n+1} + h_{M-1}^{n+1})}}{h_M^{n+1}} = 0, \tag{3.5}$$

$$\frac{X_i^{n+1} - X_i^n}{\Delta t} - \frac{\nu_i^n \frac{X_{i+1}^{n+1} - X_i^{n+1}}{0.5(h_{i+1}^{n+1} + h_i^{n+1})} - \nu_{i-1}^n \frac{X_i^{n+1} - X_{i-1}^{n+1}}{0.5(h_i^{n+1} + h_{i-1}^{n+1})}}{h_i^{n+1}} = 0 \qquad 1 < i < M, \tag{3.6}$$

$$\frac{X_1^{n+1} - X_1^n}{\Delta t} - \frac{\nu_1^n \frac{X_2^{n+1} - X_1^{n+1}}{0.5(h_2^{n+1} + h_1^{n+1})} - F_b}{h_1^{n+1}} = 0, \tag{3.7}$$

This leads for each transport equation to a system of linear equations with the following tri-diagonal matrix:

$$\begin{aligned} &-X_{M-1}^{n+1} \frac{\Delta t \nu_{M-1}^n}{0.5 h_M^{n+1} (h_M^{n+1} + h_{M-1}^{n+1})} \\ &+X_M^{n+1} \left(1 + \frac{\Delta t \nu_{M-1}^n}{0.5 h_M^{n+1} (h_M^{n+1} + h_{M-1}^{n+1})} \right) = X_M^n + \frac{\Delta t}{h_M^{n+1}} F_s \end{aligned} \tag{3.8}$$

$$
\begin{aligned}
&-X_{i-1}^{n+1}\frac{\Delta t\nu_{i-1}^{n}}{0.5h_i^{n+1}(h_i^{n+1}+h_{i-1}^{n+1})}\\
&+X_i^{n+1}\left(1+\frac{\Delta t\nu_{i-1}^{n}}{0.5h_i^{n+1}(h_i^{n+1}+h_{i-1}^{n+1})}+\frac{\Delta t\nu_i^{n}}{0.5h_i^{n+1}(h_{i+1}^{n+1}+h_i^{n+1})}\right)\\
&-X_{i+1}^{n+1}\frac{\Delta t\nu_i^{n}}{0.5h_i^{n+1}(h_{i+1}^{n+1}+h_i^{n+1})}=X_i^{n} \qquad 1<i<M,
\end{aligned}
\tag{3.9}
$$

$$
\begin{aligned}
&+X_1^{n+1}\left(1+\frac{\Delta t\nu_1^{n}}{0.5h_1^{n+1}(h_2^{n+1}+h_1^{n+1})}\right)\\
&-X_2^{n+1}\frac{\Delta t\nu_1^{n}}{0.5h_1^{n+1}(h_2^{n+1}+h_1^{n+1})}=X_1^{n}-\frac{\Delta t}{h_1^{n+1}}F_b,
\end{aligned}
\tag{3.10}
$$

which is solved by means of the simplified Gaussian elimination. It should be mentioned here, that this fully implicit treatment of diffusion leads to a first-order-in-time accuracy, only. This does not impose problems inside a one-dimensional model, since time steps may be chosen sufficiently small. However, in further **GOTM** releases, a second-order-in-time discretization of the diffusion term (Crank-Nicholson scheme) should be included.
The positivity of k, ε and kL is guaranteed by applying the quasi-implicit treatment proposed by *Patankar* [1980] to the sink terms on the right hand sides of these equations. This will be explained with the aid of an idealized equation (neglecting diffusion terms):

$$\partial_t X = P - Q, \tag{3.11}$$

where X is a positive quantity, and P and Q positive source and sink terms, respectively. Defining n as the old and $n+1$ as the new time level, the quasi-implicit discretization of (3.11) is of the following form:

$$\frac{X_i^{n+1}-X_i^{n}}{\Delta t}=P_i^{n}-Q_i^{n}\frac{X_i^{n+1}}{X_i^{n}}. \tag{3.12}$$

The solution for X_i^{n+1} is then

$$X_i^{n+1}=\frac{X_i^{n}+\Delta tP_i^{n}}{1+\Delta t\frac{Q_i^{n}}{X_i^{n}}}, \tag{3.13}$$

which shows the positivity of this numerical scheme[1].
Due to the fact that some non-linear terms still contain explicit values from the previous time step, the time step is limited. Large time steps tend to create oscillations of turbulent quantities especially if algebraic turbulence models are used.

3.2 The log-law

All quantities which contain large gradients near boundaries require careful numerical treatment. In the models introduced in the previous sections, this is the case for momentum and sediment near the bottom, and dissipation rate near bottom and surface.
For these quantities, the log-law is a good approximation near boundaries. Therefore, the numerical treatment has to be carried out such that at least the log-law is well reproduced even for a relatively

[1] Such a scheme should of course only be applied to equations for quantities which are positive by definition. For the positivity of the k-ε model, see *Mohammadi and Pironneau* [1994].

coarse vertical resolution. This is partially carried out by explicitly assuming log-law behaviour near boundaries (for momentum near the bottom, see section 3.3, and for sediment concentration near the bottom, see section 3.4) or by choosing the best among various consistent boundary condition discretizations (for the dissipation rate, see section 3.5).
For the sections 3.2.1 and 3.2.2, z is exceptionally denoting the distance from the bottom.

3.2.1 Constant stress

As a test for the quality of the numerical treatment chosen here, an analytical solution for the constant stress steady-state bottom boundary layer (under consideration of the k-ε model) is introduced now. Further simplifications are due to the lack of rotational, pressure and viscous effects. Under these conditions, the momentum equation (2.3), the relation of Kolmogorov and Prandtl (2.31), the k-equation (2.38) and the ε-equation form (together with (2.34)) a closed system of equations:

$$\nu_t \partial_z u = u_*^2 = \text{const}, \tag{3.14}$$

$$\nu_t = c_\mu^0 \sqrt{k} L = \left(c_\mu^0\right)^4 \frac{k^2}{\varepsilon}, \tag{3.15}$$

$$P = \varepsilon, \tag{3.16}$$

and

$$\partial_z \left(\frac{\nu_t}{\sigma_\varepsilon} \partial_z \varepsilon \right) = (c_{\varepsilon 1} P - c_{\varepsilon 2} \varepsilon) \frac{\varepsilon}{k}. \tag{3.17}$$

with the boundary conditions for ε, (2.48) and, alternatively, (2.49) at $z = 0$, and $\varepsilon = u_*^3/(\kappa(H + z_0))$ at $z + H$ where H is the height of the boundary layer. We further prescribe $u = 0$ for $z = 0$ at the bottom.
With

$$\sigma_\varepsilon = \frac{\kappa^2}{(c_{\varepsilon 2} - c_{\varepsilon 1}) \left(c_\mu^0\right)^2} \approx 1.08 \tag{3.18}$$

(see section 2.2), the system (3.14) - (3.17) has the following solution for u, ν_t, k, and ε:

$$\frac{u(z)}{u_*} = \frac{1}{\kappa} \ln \left(\frac{z + z_0}{z_0} \right), \tag{3.19}$$

$$\nu_t(z) = \kappa u_* (z + z_0), \tag{3.20}$$

$$k = \left(\frac{u_*^b}{c_\mu^0} \right)^2, \tag{3.21}$$

and

$$\varepsilon(z) = \frac{u_*^3}{\kappa(z + z_0)}. \tag{3.22}$$

Equation (3.22) is equivalent to a linear macro length scale $L = \kappa(z + z_0)$.
Equation (3.19) is the well-known logarithmic velocity profile for boundary layer flow and will be used for (i) calculating the momentum in the middle of the bottom layer (see section 3.3) and (ii) as a test for different boundary conditions for ε (see section 3.5).

3.2.2 Linear stress

A more realistic situation is given, when a constant pressure gradient over the whole water column (resulting in a shear stress decreasing linearly towards the surface) and a non-linear length scale are chosen:

$$\nu_t \partial_z u = (u_*^b)^2 \left(1 - \frac{z}{D}\right), \tag{3.23}$$

$$L = \kappa(z + z_0)\left(1 - \frac{z}{D}\right)^{1/2}, \tag{3.24}$$

$$P = \varepsilon, \tag{3.25}$$

$$\nu_t = c_\mu^0 \sqrt{k} L. \tag{3.26}$$

With $u(0) = 0$, the analytical solution of the system of equations (3.23) - (3.26) for momentum is again the log-law from equation (3.19).
The solution for the TKE is now a (with distance from the bed) linearly decreasing profile,

$$k = \left(\frac{u_*^b}{c_\mu^0}\right)^2 \left(1 - \frac{z}{D}\right), \tag{3.27}$$

such that for the eddy viscosity the parabolic profile

$$\nu_t = \kappa u_*^b (z + z_0)\left(1 - \frac{z}{D}\right) \tag{3.28}$$

is resulting.

3.3 Hydrodynamic equations

It has already been mentioned that the Dirichlet boundary conditions for momentum, (2.11) and (2.12), will not be directly inserted into the numerical solution, because the discrete bottom values for u and v (denoted by u_b and v_b, respectively) are positioned half a grid box above the bottom (see figure 3.1). Instead, a flux condition using bottom stresses is derived from the log-law discussed in section 3.2.
For the determination of the normalized bottom stresses

$$\frac{\tau_b^x}{\rho_0} = u_*^x u_*, \tag{3.29}$$

$$\frac{\tau_b^y}{\rho_0} = u_*^y u_* \tag{3.30}$$

with the friction velocities $u_* = \sqrt{\tau_b/\rho_0}$ with $\tau_b = \sqrt{(\tau_b^x)^2 + (\tau_b^y)^2}$, assumptions about the structure of velocity inside the bottom layer have to be made. We use here the logarithmic profile (3.19) calculated in sections 3.2.1 and 3.2.2. Therefore, estimates for the velocities in the centre of the bottom layer can be achieved by:

$$u_b = \frac{u_*^x}{\kappa} \ln\left(\frac{0.5 h_1 + z_0^b}{z_0^b}\right), \tag{3.31}$$

$$v_b = \frac{u_*^y}{\kappa} \ln\left(\frac{0.5h_1 + z_0^b}{z_0^b}\right). \tag{3.32}$$

Another possibility would be to specify the bottom velocities u_b and v_b such that they are equal to the layer-averaged log-law velocities. The calculation of this is however slightly more time consuming and does not lead to a higher accuracy.
(3.31) and (3.32) together with the definitions of u_*, u_*^x and u_*^y (see (3.29) and (3.30) give

$$\tau_b^x = r u_b \sqrt{u_b^2 + v_b^2}, \tag{3.33}$$

$$\tau_b^y = r v_b \sqrt{u_b^2 + v_b^2}, \tag{3.34}$$

with the bottom drag coefficient

$$r = \left(\frac{\kappa}{\ln\left(\frac{0.5h_1 + z_0^b}{z_0^b}\right)}\right)^2, \tag{3.35}$$

which is for $h_1 \to 0$ is equivalent to the Dirichlet bottom boundary conditions for momentum, (2.11) and (2.12), $u = v = 0$ (see section 3.5 and figure 3.2). A closer inspection of (3.33) and (3.34) shows that they are non-linear implicit functions for u_*, u_*^x and u_*^y (this is because of the definition of the roughness length z_0^b, see equation (2.1)). Therefore, their calculation requires one or more iteration steps.

3.4 Sediment equation

With the viscosity from the linear stress log-law, (3.28), the tracer equation for sediment with constant fall velocity w_s, constant Prandtl number $\sigma_t = \nu_t/\nu_t'$, and a for sediment relevant roughness length z_c has the following steady state solution:

$$\frac{C}{C_0} = \exp\left[-\sigma_t \frac{w_s}{\kappa u_*^b} \ln\left(\frac{z + z_c}{z_c} \frac{D - z_c}{D - (z + z_c)}\right)\right]. \tag{3.36}$$

This so-called Rouse profile will be used in order to calculate the the bottom discrete value for sediment, C_b. This is achieved by replacing z by $0.5h_1$ in (3.36).
For the vertical advection included in the tracer equation in order to simulate the sinking of sediments, three different numerical methods are optional in **GOTM**:

(i) first order upstream scheme,

(ii) third order (polynomial) scheme, QUICKEST (see *Leonard et al.* [1995]),

(iii) a non-linear combination of (i) and (ii), the so-called flux-corrected transport scheme (see *Zalesak* [1979]).

Method (i) is monotone, but diffusive, such that sharp gradients (lutoclines) are not resolved. In contrast to that, method (ii) is dispersive, i.e. monotonicity is violated and unphysical oscillations occur. The method (iii) is quite complex, but combines the advantages of both other methods, i.e. it is strictly monotone and preserves sharp gradients. It should be noted, that in the present version of **GOTM** the QUICKEST scheme is consistent for equidistant vertical discretization, only.

3.5 Turbulence equations

In three-dimensional ocean models, a good vertical resolution can often not be achieved. It is therefore desirable that also the turbulence closure schemes are of sufficient accuracy already for coarse vertical resolutions. We found that the Mellor-Yamada model fulfills this requirement. In order to obtain this property also for k-ε models, the boundary conditions have to be formulated and discretized with care. This is necessary in order to cope with the asymptotical behaviour of the dissipation rate near vertical boundaries. In the following, the straight forward discretizations of the boundary conditions (2.48) and (2.49) will be discussed and their numerical performances compared to an analytical test case. The viscosities ν_k, ν_L, and ν_ε have to be averaged from ν_t values (placed at the same position as other turbulent quantities) in order to guarantee second order accuracy for the discretization of diffusion terms. In the standard discretization used here, this is also applied to the surface and the bottom interval. The spatial accuracy of the standard discretization of ε boundary conditions (2.48) and (2.49) will be tested by discretizing the constant stress steady state boundary layer equations (3.14) - (3.17).
The results for u are shown in figure 3.2 on a logarithmic scale. With the flux condition (2.49), the logarithmic profile of u is reproduced for all vertical resolutions, even if the boundary layer is resolved with three intervals only. With the Dirichlet condition (2.48), an extremely fine vertical resolution is needed near the bottom in order to reproduce the analytical solution accurately. Only the bottom values for u are by definition (see section 3.3) identical with the analytical solutions. This fine resolution could only be achieved with a non-equidistant zooming resulting in a height of the near bottom interval of only $0.003 z_0$. Such fine resolutions are not affordable in three-dimensional models. *Stelling* [1995] discussed the low accuracy of the standard discretization of the Dirichlet condition. He suggests to compute the first eddy viscosity for ε by $\nu_\varepsilon = u_*^4/(\sigma_\varepsilon \varepsilon)$, a relation which can be derived from $P = \varepsilon$. A test computation carried out by us for the analytical test case discussed above shows the same high accuracy than achieved by the standard discretization of the flux boundary condition (2.49). However, the *Stelling* [1995] approach does not work for stress free surface flow, because it assumes a no-flux condition for ε there. In contrast to the flux condition (2.49) this predicts unrealistically high near surface values for the eddy viscosity. Therefore we decided only to use the flux condition (2.49) for our k-ε model computations.
It has been shown by *Burchard and Petersen* [1999] that with this boundary condition, the k-ε and the Mellor-Yamada models are of similar accuracy also in the mixed layer (see figure 6.17).

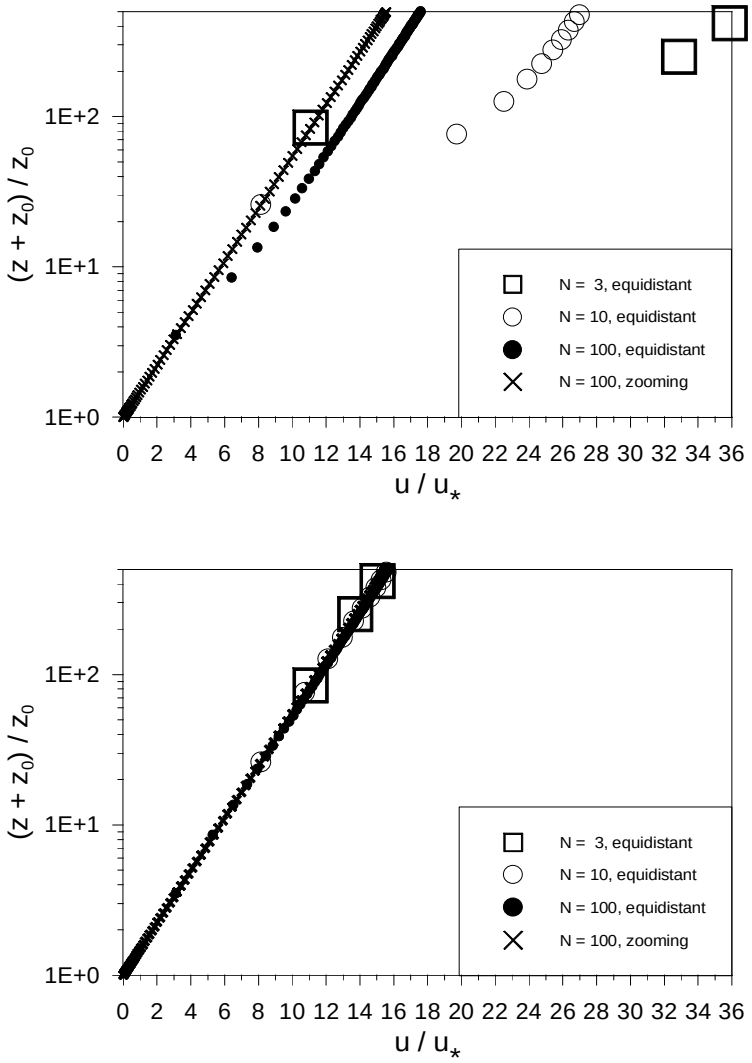


Figure 3.2: Non-dimensional profiles of velocity for non-stratified boundary layer flow calculated with a k-ε model with different vertical resolution. Upper panel: Dirichlet boundary condition for ε, (2.48); Lower panel: Flux boundary condition for ε, (2.49). This figure has been taken from *Burchard and Petersen* [1999].

Chapter 4

Implementation

In this chapter, a description of different aspects of the implementation of **GOTM** is given. Since the amount of computations done in **GOTM** is modest compared to a full three-dimensional fluid dynamical code, it has not been the aim to produce an extremely fast and efficient code. With todays PC's even quite long integrations can be done in a matter of the most of a few minutes[1]. It was also out of our scope to construct a 'point and click' application which conformed to the Windows 95 or some other Graphical User Interface standard. Rather it has been the aim to produce something in between a relatively efficient, easily extendable and also relatively easy to use numerical tool. Concerning the 'point and click' approach there are pros and cons. The major advantage of it is the apparent ease of use. But since one of the aims of using **GOTM** is to investigate different turbulence closure schemes, we believe that a set-up using scripts in order to make a number of simulations is more efficient. However, the implementation of a Graphical User Interface (GUI) for **GOTM** is desirable for the near future. Moreover, a graphical data processing tool would increase the applicability of **GOTM**. Both such tools should be of sufficient generality to be able to run on a number of different platforms such as UNIX or Windows based systems. Such a work could possibly be conducted in the framework of a master thesis for computer scientists.
Another topic of concern is the transformation of the different algorithms developed in **GOTM** to a 3D-model. We do not think that there is a simple solution to this problem, because in reality many factors have to be considered such as the coding style in the 3D-model (common blocks), the numerical grid layout, indexing conventions etc. However, one of the early FORTRAN realizations of the **GOTM** turbulence part (the k-ε model) has recently been successfully implemented into the Hamburg Shelf and Ocean Model (HAMSOM) by *Herklotz* [1999]. This was done by creating an environment inside HAMSOM, which is completely consistent with the **GOTM** notations. This guarantees, that innovations made inside **GOTM** can easily be transferred to HAMSOM now. So far, no previsions have been considered for **GOTM**, which could simplify a transfer of parts of **GOTM** to three-dimensional models.
During the development and also during the daily use of **GOTM**, we have used a few tools which are not strictly necessary but might prove helpful to others as well. We will give a short presentation of each of the tools and information on how to obtain them. It should be noted that **GOTM** has been developed under UNIX, and therefore most of these tools may not be available in that form on other platforms such as Windows.

[1] One year of integrating the Ocean Weather Station Papa or November with high resolution in time and space, i.e. with a time step of 120 s and a number of 250 vertical layers took about 50 minutes on a Pentium 200 MHz PC.

When more than one person works on a set of source code files, it is a necessity to have some kind of version control system in order to keep track on, who changed what at what time in order to merge different versions. In the UNIX world there are a few standard programs and tools to achieve this task. We have chosen to use RCS (Revision Control System), which has worked fine, and which is installed on many commercial UNIX workstations. This was a helpful tool, since during the major part of the program development time we were situated together at the Joint Research Centre in Ispra, Italy. In the future we might change to another tool, CVS (Concurrent Version Control) which builds on RCS but has some advantages when cooperating via the Internet.
As will be described below, we have embedded the documentation in the source code using the PROTEX software. PROTEX relies on Perl to be installed on the computer. PROTEX generates LaTeX which by using another perl-based package, LaTeX2HTML, can be converted into HTML-code to be displayed using the WWW (World Wide Web).
Finally we have relied on Make to compile the relevant source files during the development phase. We have used GNU's Make programme but have not used any of the more advanced features such that any Make should be able to digest the supplied `Makefile`.
More informations about these features can be found on the **GOTM** web site (for the address, see section 1.1 of this report).
There are not yet any experiences how to run **GOTM** under other platforms than UNIX (such as Windows). However, for experienced numerical modellers, this should not be a problem. We will include support for this issue into the **GOTM** web site as soon as we receive feedback from users under other platforms.

4.1 Programming language

Due to the wide acceptance of FORTRAN 77 in the scientific community, we have chosen to implement **GOTM** in this language. We are fully aware that some features could be implemented more elegantly and efficiently in other languages like e.g. C, C++ or FORTRAN 90 (the first versions of **GOTM** were actually implemented in PASCAL). But since we hope that different scientists will contribute to the code with new turbulence schemes and new features, we chose the language that is most widespread. The source code is not strictly FORTRAN 77 compliant, since we have used a few non-conformant features, which are however widely accepted. One of such features is the length of the names of subroutines and functions - where the standard dictates a maximum of 6 characters. Here, we have decided to use descriptive names instead. Another example is the use of `include` statements in order to include `const.i` in various files.
One of the problems when writing a general tool in FORTRAN 77, like **GOTM**, is the lack of dynamic memory allocation. The implication is that at compile time one has to decide how big arrays are to be used in the future. And in addition, checks at a lot of different places in the code have to be included in order to assure that the compile time limits are sufficient. In reality, the compile time dimensions are of course never enough, so at some time re-compilation is necessary, as we realized several times during the development phase.
One other thing to mention is that we in all subroutines and functions have used the `implicit none` option, which means that the implicit naming convention of variables in FORTRAN 77 is avoided and all variables have to be explicitly declared. This makes debugging much easier with almost no overhead, and it is good practice to declare variables anyway.
Common blocks are not used in **GOTM**. Instead, for every subroutine and function, a list of parameters has to be passed.

4.2 Documentation

The documentation of **GOTM** consists of three parts, the first is this report which contains physical, numerical and practical documentation and discusses the test cases. The second is the **GOTM** web site which is intended to be the entrance gate and market square of **GOTM**. Through there, **GOTM** including this report and the test cases can be downloaded. In addition, the web page will be the basis for keeping contact to the users and there, informations about **GOTM** will be given on a regular basis.
The third means of documentation is the source code documentation. One of the most difficult tasks in software development is to keep the documentation consistent with the source code. This is especially true in a project where more people work simultaneously on the code. We have chosen to embed at least some documentation into the source code itself. The advantage of this procedure is that it is not necessary to maintain two sets of files - source code and documentation. The disadvantage is the limitations in the layout of the documentation. Then we discovered PROTEX which has been developed at the NASA Data Assimilation Office[2]. PROTEX, which is a Perl script, defines a number of tags which can be used in order to describe the various parts of the source code. The tags include e.g. subroutine name, description, author etc. PROTEX scans the source code for tags and reformats into LaTeX from where either a hardcopy can be made using e.g. `dvips`, or ready to Internet publication documentation using LaTeX2HTML can be generated. For those users who do not have access to the necessary tools to create the embedded source code documentation, we have on the **GOTM** web site placed a set of postscript files and HTML documents for download.

4.3 Program structure

The structure of the program is to a large extent given by the problem at hand. The source code is split into a number of FORTRAN files with names that describe the task they perform. The present version of **GOTM** consists of 49 FORTRAN files and 1 include file. Based on user chosen input, the governing equations are solved and output is produced. This is reflected in the source code where we have an initialization part and a calculation part (actually the time loop) which also performs the output. The overall program execution control is contained in the `gotm.f` source file. The first part of the file consists of the initialization procedure, then comes the actual integration in time, and finally the program is properly exited.

4.3.1 Initialization

Initialization of variables is done by using namelists. Instead of having one big namelist, we have chosen to have one for each of the different components of the program (these can be - and in fact are - contained in the same physical file). In the beginning of `gotm.f`, variables are declared. We have chosen not to use common blocks at all in **GOTM** but rather pass variables as parameters in subroutine and function calls. After this, namelists are specified and read from `namelist.inp`. We have hard-coded the name of the namelist file into `gotm.f`. This is not very elegant, but the reason is that not all FORTRAN 77 compilers know `getarg` (not FORTRAN 77 compliant) and in case they do, the syntax might slightly differ between different compilers. There are two simple

[2]For further informations, look at the Perl home page, `http://www.Perl.org`. PROTEX may be downloaded via ftp from `dao.gsfc.nasa.gov/pub/tools/protex/protex1.4.tar.gz`. There, further informations how to use PROTEX are given.

solutions. To implement an appropriate call to `getarg` in order to get the namelist filename from the command line - or to use a symbolic link to point to different namelist files (UNIX solution only). After the namelist is read and basic program controlling variables have been initialized, a number of different variables is initialized in the subroutine `Initialization`, the output system is setup in `OutPutSetUp` - see below for further information. The initial forcing is read in `Forcing`, and we are ready for the actual integration after initial fields have been written in `DoOutPut`.

4.3.2 Time loop

The time loop is implemented as a simple `goto` loop, in which the program iterates until the stop criterium is met (the model time exceeds the specified stop time). The loop starts by updating the time and continues by calling a number of subroutines. There are also a few `if`-statements for conditional execution, and printing to the terminal and creating output.
In the following, the layout of the time loop is given. For better readability, some print statements, diagnostic routines and the parameter lists have been removed:

```
900       step=step+1
          time0=time0+dt
          date=startdate+time0/86400

          call Forcing
          call NewGrid
          call Coriolis
          call uEquation
          call vEquation
          call PressureGradient
          call Friction
          if (TRelaxTau.ge.0) call Temperature
          if (SRelaxTau.ge.0) call Salinity
          if (SediCalc)       call Sediment
          call StabilityFunctions
          call TKE
          call LengthScale
          call KolPran
          if(Iwmodel.eq.2) call InternalWave
          if(mod(nint(step),ISAVE).eq.0) call DoOutPut

      if (date.lt.stopdate) goto 900
```

In `Forcing`, a number of informations is read in from files or is generated analytically. This includes also a varying water depth. In `NewGrid`, the layer depths are adjusted to the new water depth. This is so far only done by stretching or compressing the vertical increments proportional to the water depth, as practised in σ-coordinate models. In `Coriolis`, the velocity vector is simply rotated by the angle $f \cdot \Delta t$. In `uEquation` and `vEquation`, vertical diffusion (including bottom and surface boundary conditions) for the momentum vector components are computed. If the surface slope is directly given as pressure gradient forcing, it is calculated here as well. If pressure forcing is given indirectly as a prescribed velocity at a specified height above the bed or as vertical mean velocities, it is calculated in `PressureGradient`. The bottom friction coefficients are processed

in `Friction`. The tracer equations `Temperature`, `Salinity` and `Sediment` are executed only, if their execution is prescribed in `namelist.inp`. The turbulence part is contained in the subroutines `StabilityFunctions`, `TKE`, `LengthScale` and `KolPran`. In `StabilityFunctions`, the stability of the water column and as a function of that, the production terms for the turbulent kinetic energy and the stability functions are calculated. `TKE` and `LengthScale` are only wrapper routines, which call various methods for calculating the turbulent kinetic energy (algebraic or transport equation) and the macro length scale (algebraic or transport equations for dissipation rate or length scale), respectively. In `KolPran`, the eddy viscosity and diffusivity are calculated by using the relation of Kolmogorov and Prandtl. Finally, `InternalWave` executes the internal wave and shear instability parameterization of *Kantha and Clayson* [1994] if selected in the `namelist.inp` file.

4.3.2.1 Getting the forcing data

All reading of external data in the time loop is contained in one single subroutine `Forcing`. With the amount of possible different input parameters, and thereby the numbers of parameters passed to `Forcing`, which we have in this present version of **GOTM**, this is feasible. In the case that the model is significantly extended (through a biological module for example), it might be necessary to implement a system such that each subcomponent of the model takes care of its own in- and output. This could for example be carried out through a number of reserved unit numbers for each model component. In the chapter 5 (Users Guide), a more detailed description of the various input data formats is given.
The forcing data do not have to be consistent with the spatial and temporal discretization of the model. They might even be irregular in time and space. Forcing data may be of three different types:

- one-dimensional in time, such as meteorological forcing,
- one-dimensional in space, such as for initial conditions for temperature and salinity,
- two-dimensional in time and space, such as records of temperature, salinity and dissipation rate, when they are read in for relaxation (temperature and salinity, only) or validation. Temperature and salinity will also be used as initial conditions.

The only requirement is, that the simulation period has to be within the time span defined by the forcing data. The forcing data will then be interpolated in time and space to the numerical grid. Near the surface and the bed, constant data extrapolation is applied if grid points lay outside the interval defined by the data.

4.3.2.2 Writing the output

Since all users have different preferences for post-processing programmes, we have implemented an almost generic output system. The idea is further described in the chapter 5 (User's Guide). The default output format is an easy to read ASCII format, where data profiles are written in a tabular form with additional information concerning the time and the different variables. The name of the output file is given in the namelist file.

Chapter 5

User's Guide

5.1 General

In this chapter, it is explained how to access, compile and run **GOTM**. We tried to thoroughly discuss even details, such that also unexperienced users can work with **GOTM**. It might be that some information given here, is redundant for those who have already read the chapter 4 (Implementation). Since it is intended to further develop **GOTM**, the user instructions given in this report might soon run out of date. Therefore, we have set up a web site for **GOTM**, which will be regularly updated and always contains the most actual informations on **GOTM**. The whole model has to be obtained through this web site anyway. The address is given in section 1.1 (Introduction).
The best way of learning how to work with **GOTM** is to experiment with some of the test cases included into the **GOTM** package as well. These are shortly described in section 5.3 below. An extensive physical discussion can be found in chapter 6. After this, the user will be able to construct her or his own scenarios. After some study of the actual code implementation, also source code changes for improving physical or numerical features should be possible for the slightly experienced user.

5.2 Accessing and compiling GOTM

GOTM has to be downloaded through the World Wide Web (WWW). The address is given in section 1.1 (Introduction). The user's web browser should be able to download binary files and copy them to specified directories.
First of all, the source code of **GOTM** has to be copied into an empty directory which should be named `gotm`. The downloaded source code file is named `gotm.tar.gz` and is a compressed archive. First, the file has to be uncompressed, and then all included files have to be extracted. Under UNIX systems, this is simply done with the two command lines:

```
gunzip gotm.tar.gz
tar -xvf gotm.tar
```

Our experience with other systems than UNIX is limited, and therefore, we can not yet give support to such systems (like Windows). However, as soon as we receive some feedback from users under such systems, we will include the respective support into the **GOTM** web site.

After having extracted the currently 49 FORTRAN files (ending with a `.f`), one include file (`const.i`), and the `Makefile` will be included in the directory `gotm/gotm1.0`. **GOTM** is now ready for compilation.
Under most UNIX systems, the command

```
make
```

in the directory `gotm/gotm1.0` should create an executable file simply named `gotm`. We have experienced that this does not work on certain SUN work stations where `ranlib` has to be run explicitly. Therefore we designed another `Makefile` for such systems which has be downloaded from the web site in order to replace the `Makefile` which is included in `gotm.tar.gz`. With the two commands

```
make clean
make
```

the executable should now also be created for SUN work stations.

5.3 Running GOTM's test cases

In order to run **GOTM**, at least one more file is needed. This is the namelist file, which for all applications is called `namelist.inp`. An example for a namelist file is given in the Appendix 7. It contains all specifications for the different scenarios. Re-compilation is not needed when the user wants to switch from one to another application. Most of the applications will also require some more input files to be read in, such as forcing files containing the meteorological forcing or initial data for temperature or salinity.
Sets of namelist and forcing files are included into the **GOTM** package as the following test cases:

- Kato-Phillips experiment (`KatoPhillips.tar.gz`, 7.5 KB), a constant surface stress applied to a linearly stratified water column initially at rest, such that a mixed layer is developing (see section 6.1.2).
- Fladenground Experiment 1976 (`Flex.tar.gz`, 0.26 MB), two months of surface forcing and temperature profile data from the northern North Sea in spring and early summer 1976 (see section 6.2.1).
- Irish Sea 1993 (`IrishSea.tar.gz`, 0.15 MB), 24 hours of surface forcing, tidal forcing and dissipation rate measurements (see section 6.2.2).
- Oosterschelde 1996 (`Ooster.tar.gz`, 0.15 MB), 3 days of forcing and temperature, salinity and dissipation rate profile measurements in a well-mixed tidal estuary (see section 6.3.1).
- Knebel Vig 1997 (`KnebelVig.tar.gz`, 0.21 MB), 2.5 days of forcing and temperature, salinity and dissipation rate profile measurements in a strongly stratified embayment (see section 6.3.2).
- Ocean Weather Station Papa (`OWS_Papa.tar.gz`, 2.53 MB), long term observations of meteorological forcing (August 1959 - July 1981) and temperature profiles (1960 - 1968) in the northern Pacific at 50° N (see section 6.2.3.1).

- Ocean Weather Station November (`OWS_November.tar.gz`, 1.95 MB), long term observations of meteorological forcing (December 1959 - April 1974) and temperature profiles (1960 - 1970) in the northern Pacific at 30° N (see section 6.2.3.2).

The test cases can be accessed in the same way than the source code. After copying e.g. the Fladenground Experiment '76 test case (`Flex.tar.gz`) into the directory `gotm`, the following two command lines have to be given:

```
gunzip Flex.tar.gz
tar -xvf Flex.tar
```

This creates a directory named `Flex`, containing the `namelist.inp`, a `README` file with some basic informations, temperature and salinity profiles and some forcing files.
In order to run this test case, the executable file `gotm` has to be copied into the test case directory (here `Flex`), or, more elegantly, a symbolic link has to be set:

```
ln -s ../gotm1.0/gotm gotm
```

The test case is then executed by simply giving the command

```
gotm
```

Now, **GOTM** is running and producing output on the screen and to files in the test case directory.

5.4 Model output

The following output should then appear on the screen:

```
 Using ASCII  file format
 Storing data in file:
 Flex.out

 Initializing S-profile
 Searching for  96 21600


 Initializing T-profile
 Searching for  96 21600

  Model Day      Modelled SST    Measured SST    rms error
 96.25000000     6.22000         6.21294         0.00706
 96.50000000     6.24993         6.21288         0.01850
 96.75000000     6.25684         6.21288         0.03723
 97.00000000     6.23151         6.21288         0.03512
 97.25000000     6.21748         6.21294         0.03098
 97.50000000     6.24636         6.21294         0.02863
 97.75000000     6.25686         6.21294         0.03252
 98.00000000     6.22784         6.21294         0.03190
```

```
98.25000000     6.22355         6.21294         0.03014
98.50000000     6.25417         6.21294         0.02957
98.75000000     6.24413         6.21288         0.03040
...             ...             ...             ...
```

This output indicates that the output of model results will be written to `Flex.out` in the ASCII format, currently the only output option (the NetCDF and GrADS formats will be provided with the next **GOTM** release). It is furthermore indicated that initial profiles for salinity and temperature have been found (`96` meaning the observation day and `21600` the second of that day). And then, a time series of calculated data is given in 6-hourly intervals, each line containing model time in days, simulated sea surface temperature (temperature in uppermost grid point), measured sea surface temperature (as contained in the `Meteo.dat` file as last column, but interpolated to the time step), and the root mean square error between these two temperature time series.
The `Flex.out` file is created inside the `Flex` directory during run time. It contains a variety of model parameters as time series of vertical profiles. In the following, some lines extracted from the head of the `Flex.out` file are given:

```
50 -145.
Time:   96.25  Set#  1
       z          u           v           T           S           C       buoy
  -1.450 0.000E+000 0.000E+000 0.6220E+01 0.3506E+02 0.0000E+00 0.0000E+00
  -4.350 0.000E+000 0.000E+000 0.6220E+01 0.3506E+02 0.0000E+00 0.0000E+00
  -7.250 0.000E+000 0.000E+000 0.6220E+01 0.3506E+02 0.0000E+00 0.0000E+00
  ...    ...         ...         ...         ...         ...         ...
-137.750 0.000E+000 0.000E+000 0.6220E+01 0.3510E+02 0.0000E+00 0.0000E+00
-140.650 0.000E+000 0.000E+000 0.6220E+01 0.3510E+02 0.0000E+00 0.0000E+00
-143.550 0.000E+000 0.000E+000 0.6220E+01 0.3510E+02 0.0000E+00 0.0000E+00
       z         num         nuh           k         eps           L          NN
  -2.900 0.000E+000 0.000E+000 0.3000E-05 0.5000E-09 0.1000E-01 0.0000E+00
  -5.800 0.000E+000 0.000E+000 0.3000E-05 0.5000E-09 0.1000E-01 0.0000E+00
  -8.700 0.000E+000 0.000E+000 0.3000E-05 0.5000E-09 0.1000E-01 0.0000E+00
  ...    ...         ...         ...         ...         ...         ...
-136.300 0.000E+000 0.000E+000 0.3000E-05 0.5000E-09 0.1000E-01 0.0000E+00
-139.200 0.000E+000 0.000E+000 0.3000E-05 0.5000E-09 0.1000E-01 0.0000E+00
-142.100 0.000E+000 0.000E+000 0.3000E-05 0.5000E-09 0.1000E-01 0.0000E+00
       z          Td          Sd        epsd
  -1.450 0.622E+001 0.351E+002 0.0000E+00
  -4.350 0.622E+001 0.351E+002 0.0000E+00
  -7.250 0.622E+001 0.351E+002 0.0000E+00
  ...    ...         ...         ...
-137.750 0.622E+001 0.351E+002 0.0000E+00
-140.650 0.622E+001 0.351E+002 0.0000E+00
-143.550 0.622E+001 0.351E+002 0.0000E+00
Time:   96.2916667  Set#  2
       z          u           v           T           S           C       buoy
  -1.450 0.432E+000 -.444E+000 0.6187E+01 0.3506E+02 0.0000E+00 0.0000E+00
  -4.350 0.242E+000 -.189E+000 0.6202E+01 0.3506E+02 0.0000E+00 0.0000E+00
  -7.250 0.132E+000 0.149E-001 0.6216E+01 0.3506E+02 0.0000E+00 0.0000E+00
```

```
...    ...      ...      ...      ...      ...      ...
```

Dots ... indicate that a number of lines has been omitted. In the first line, the number of vertical layers and the bottom coordinate in m are given. The next line is the header as it will occur for every record. It gives model time in days and the number of the record. Then, profiles of mean flow quantities are given: u-velocity (`u`), v-velocity (`v`, each in $\mathrm{m\,s^{-1}}$), potential temperature (`T`, in °C), salinity (`S`, in psu), sediment concentration (`C`, in kg sediment per kg sediment-laden water), and buoyancy (`buoy`, in $\mathrm{m\,s^{-2}}$, only given if the buoyancy transport equation is explicitly calculated). After the mean flow quantities, a number of profiles for turbulent quantities is given: eddy viscosity (`num`), eddy diffusivity (`nuh`, each in $\mathrm{m^2 s^{-1}}$), turbulent kinetic energy (`k`, in $\mathrm{J\,kg^{-1}}$), dissipation rate (`eps`, in $\mathrm{W\,kg^{-1}}$), macro length scale (`L`, in m), and the square of the Brunt-Väisälä frequency (`NN`, in $\mathrm{s^{-2}}$, not calculated for the initial conditions). Afterwards, as a third group of profiles, data read in from external files, interpolated to the model grid and time step are given: temperature (`Td`, in °C), salinity (`Sd`, in psu), and dissipation rate (`epsd`, in $\mathrm{W\,kg^{-1}}$). Then follows the second record, and so on.
Two other output files are created: `mld.out` and `heat.out`, the first giving mixed layer diagnostics, the latter calculating the heat budget of the water column. For `mld.out`, the first three lines look like this:

```
96.2558    2.9000  145.0000  145.0000  145.0000  145.0000    6.2116    6.2366
96.2616    2.9000  145.0000  145.0000  145.0000  145.0000    6.2034    6.2366
96.2674    5.8000  145.0000  145.0000  145.0000  145.0000    6.1955    6.2366
...        ...     ...       ...       ...       ...         ...       ...
```

The columns indicate model time in days, mixed layer depth (MLD) from the minimum TKE criterium (starting from zero even for a well-mixed water column like here), MLD from a minimum temperature difference criterium for simulated and for measured temperatures, the MLD from a minimum stability criterium for simulated and for measured temperatures, and simulated and measured sea surface temperature. The critical values can be changed by editing the source code file `mldcalc.f`. After that, the model would have to be recompiled.
The `heat.out` file is of the following form:

```
96.2557870   0.5287130E+02  -0.9710086E+05   19.5449217 -210.8531861
96.2615741   0.1187956E+03  -0.1905936E+06   25.4011133 -209.4901722
96.2673611   0.1847199E+03  -0.2804767E+06   31.2573050 -208.1271583
...           ...          ...              ...         ...
```

Here, the columns show the model time in days, the heat content of the water column relative to the initial heat content in $\mathrm{J\,m^{-2}}$, as calculated from the measured temperature profiles, the accumulated heat gain of the water column in $\mathrm{J\,m^{-2}}$, as calculated from surface heat fluxes, the incoming short wave radiation in $\mathrm{W\,m^{-2}}$, and the surface heat flux (latent, sensible and long-wave back radiation) in $\mathrm{W\,m^{-2}}$.
These three output files can be used by the user for further processing and creating graphics. This is what we have done for the test cases in chapter 6.

5.5 Creating a new scenario

For changing a given test case (or scenario), or creating a new one, the user has to achieve a deeper understanding of **GOTM's** structure. Such changes can still be undertaken without looking into

the source code or recompiling. Thus, no understanding of the programming language FORTRAN is required at this stage. It is only necessary to study the namelist file `namelist.inp` and the format of the forcing files. The easiest way of creating a new scenario would be to copy an existing test case, which is similar to the new one, into a new directory.

5.5.1 What is the time in GOTM ?

Before we describe how to prepare the input files, we will give a short explanation on how `time` is treated in **GOTM**.
GOTM can both be used for theoretical simulations as the Kato-Phillips experiment and also to simulate real observations with real time information. We do not use a real calendar time in **GOTM**, instead the user specifies in the namelist file (see below) a simulation start time and a simulation stop time. The unit for these two times are days as real numbers. In the Kato-Phillips experiment, for example, the input is

```
starttime = 0.0
```

and

```
stoptime = 1.2501
```

This will result in a few seconds more than a 30 hour simulation.
For the Fladenground Experiment 1976 example, the following is used:

```
starttime = 96.25
```

and

```
stoptime = 158.0
```

Here, we have chosen the day number in the year 1976 as time reference. All other times in input files have to be adjusted to this time convention and all time stamps on output will reflect this convention.

5.5.2 The namelist file

The namelist file `namelist.inp` for the Fladenground Experiment 1976 scenario is given in the appendix. This namelist file contains a number a namelists:

- `param`: General settings for the simulation such as time, water depth, turbulence closure, number of layers, zooming etc.
- `consts`: Some general constants.
- `keps`: Parameters for the k-ε model.
- `my`: Parameters for the Mellor-Yamada model.
- `stabfunc`: Parameters for the stability functions.
- `iw`: Specifications about the internal wave and shear-instability parameterizations.

- `output`: Name of output file and output time step.
- `sobs`: File name and function (relaxation or only validation) of the salinity profile file.
- `tobs`: File name and function (relaxation or only validation) of the temperature profile file.
- `eobs`: File name of the dissipation rate measurements.
- `bobs`: Parameters for a linear initial buoyancy profile. Only used, if buoyancy transport equation explicitly calculated.
- `meteo`: Meteorological forcing.
- `press`: Pressure gradient forcing.
- `zetaspec`: Surface elevation.
- `extinct`: Light extinction coefficients.
- `sedi`: From here, the sediment transport equation is steered.

For each of these namelists, a documentation of all parameters is given in the `namelist.inp` file.

5.5.3 Specifying the forcing

The forcing is specified in the respective namelists. For idealized cases, all forcing is analytically calculated such as in the Kato-Phillips simulation. For realistic simulations, a number of data files has to be read in, each following a certain format. In the following, these analytical expressions and data formats are explained.
Some effort has been made in order to make data handling easy. So, data from files do not need to be regular in time and space, profiles do not all have to have the same length, and they do not have to match the model discretization in time or space.

5.5.3.1 Meteorological forcing

For the meteorological forcing, surface stresses in $\mathrm{N\,m^{-2}}$, incoming short wave radiation in $\mathrm{W\,m^{-2}}$, outgoing radiation in $\mathrm{W\,m^{-2}}$ and sea surface temperature have to be given. This can be either done by specifying constant values or reading in data from a file. For the Fladenground Experiment 1976 simulation, the first three records of the `Meteo.dat` file are of the following form:

```
  96  21600
0.3953950E+00 -0.6470100E+00  0.1368873E+02 -0.2122162E+03  0.6212939E+01
  96  25200
0.6952790E+00 -0.5391750E+00  0.5585331E+02 -0.2024025E+03  0.6212939E+01
  96  28800
0.8113300E+00 -0.6182540E+00  0.1519728E+03 -0.1914594E+03  0.6212939E+01
  96  32400
0.5586880E+00 -0.6254430E+00  0.2032298E+03 -0.1583254E+03  0.6212939E+01
...           ...             ...           ...             ...
```

Each record consists of two lines, the first of which gives time in days and seconds of that day, the second line giving the forcing data as listed above. For all other test cases with meteorological forcing read in from a file, the format has to be the same than here.

5.5.3.2 Pressure gradient forcing

The pressure gradient might be analytically given as constant values or as a combination of two harmonic components:

$$p^u(t) = A_M^u \sin\left(\frac{2\pi(t-\tau_M^u)}{T_M}\right) + A_S^u \sin\left(\frac{2\pi(t-\tau_S^u)}{T_S}\right) \tag{5.1}$$

$$p^v(t) = A_M^v \sin\left(\frac{2\pi(t-\tau_M^v)}{T_M}\right) + A_S^v \sin\left(\frac{2\pi(t-\tau_S^v)}{T_S}\right) \tag{5.2}$$

Here, p may be the surface slope, velocity at a specified height above the bed (for this case the height has to be specified), or mean current velocity component. A is the amplitude, τ the phase and T the period of p. A superscript u or v denotes, whether the u- or the v direction is considered, a subscript M or S denotes which of the two harmonics is used (default harmonics are semi-diurnal M_2 and S_2 oscillations).
If the pressure gradient information is read in from a file, the format of the pressure gradient file has to be as follows (again, the first three records from `Pressure.dat` for the Fladenground Experiment 1976 test case are given):

```
  96  21600    0.000
-3.2010771E-06 -1.6297247E-06
  96  22500    0.000
-2.9533401E-06 -2.1274509E-06
  96  23400    0.000
-2.6595433E-06 -2.5899826E-06
...            ...
```

The first line of each record gives day and seconds of that day, and the height above the bed (this is only used if current speed measurements are given; here in this example, surface slopes are read in, and the third value is therefore set to zero and ignored). The second line of each record contains values for p^u and p^v (see above).

5.5.3.3 Surface elevation

Also for the surface elevation, a harmonic tide may be specified:

$$\zeta(t) = A_M \sin\left(\frac{2\pi(t-\tau_M)}{T_M}\right) + A_S \sin\left(\frac{2\pi(t-\tau_S)}{T_S}\right), \tag{5.3}$$

the notations are consistent with those for the pressure gradient.
The data format for the surface elevation file is similar to that for the pressure gradient. Here, the first three records from `Zeta.dat` for the Knebel Vig 1997 simulation are given, since for the Fladenground Experiment 1976 simulation surface elevation variations are ignored and no `Zeta.dat` file is read in:

```
11      0
 0.40297E-02
11    600
 0.15610E-01
```

```
11   1200
 0.23821E-01
 ...
```

Again, observation day and second of day are given, and, in the second line of each record, the surface elevation.

5.5.3.4 Salinity and temperature, and dissipation rate profiles

Profiles of salinity and temperature and dissipation rate can be read in from `SObs.dat`, `TObs.dat` and `EObs.dat`, respectively, for various purposes. First of all, salinity and temperature profiles can be used as initial conditions. Then, only one profile is needed. It has to contain exactly the time at which the model is started. Profile data will then be spatially interpolated to the grid levels, above the uppermost and below the lowest data given, constant extrapolation is carried out.
If more than one salinity or temperature profile is given, the whole model simulation period has to lay within the dates given by the profiles. The model then searches for the suitable two profiles for salinity and as well for temperature, and interpolates them to the model grid in time and space such that they can be used as initial conditions. In this case, this procedure is executed at every time step, providing measured temperature and salinity at every grid point. These values can be used for model validations, they are also printed out in the profile output file. They can also be used for relaxing modelled temperature and salinity to measurements, when advection of these quantities has to be parameterized. It is explained in the `namelist.inp` file how to do this.
The dissipation rate measurements can only be used for model validation.
In the following, parts of the first lines of the temperature profiles from `TObs.dat` for the Fladenground Experiment 1976 are shown:

```
84 43200 56 1
-138.75  6.06000042
-136.25  6.06000042
-133.75  6.06000042
...      ...
  -6.25  6.11250019
  -3.75  6.11999989
  -1.25  6.12874985
84 64800 56 1
-138.75  6.22333336
-136.25  6.22333336
-133.75  6.22249985
...      ...
```

Each profile consists of a header line and the profile data. In the header line, four integer numbers have to be specified, observation day, second of that day, length of record (this may vary from profile to profile, see e.g. Ocean Weather Station Papa simulation), and the direction of the profile (`1` for bottom up and `2` for surface down). The following profile data have to be monotone. It can be seen for the data file shown here, that it starts already on day 84, while the meteorological forcing only starts at day 96. The first days will simply be skipped, but might be used with some extended meteorological forcing files later.

5.5.3.5 Light extinction

Light extinction in **GOTM** is calculated by means of the double exponential formula (2.15) as presented in section (2.1.1.2.1) which depends on the empirical parameters a (weighting parameter, dimensionless), η_1 and η_2 (attenuation lengths, in m). **GOTM** has the option to either specify these parameters according to the six *Jerlov* [1968] classes or to read these data from a file. Such a file (named `Extinction.dat`) for the Fladenground Experiment 1976 is shown below:

```
 96  21600
0.6200000      0.6000000      22.47189
 96  43200
0.6200000      0.6000000      22.35083
 96  64800
0.6200000      0.6000000      22.22977
...            ...            ...
```

Again, it consists of two lines per record, the first one giving observation day and second of that day, the second one giving the three empirical values as listed above.

5.6 How to contribute to GOTM

This first **GOTM** release is a basic version where various features are still missing (see also section 2.3). We are going to further extend the model. Informations on updates will be found on the **GOTM** web page, for the address, see section 1.1. Everybody else is asked to contribute to **GOTM** as well. Such contributions might be the collection of data for new test cases or new features for the source code itself. The advantage of such a feedback to us is obvious: The extensions might be tested and verified by a number of other scientists. And the contributions would be fully acknowledged by means of mentioning the contributor's name in the source code and on the web page, or even with co-authorship for substantial contributions.
Those who want to contribute, should contact us, we have set up a **GOTM** email address:

`gotm@bolding-burchard.com`

We could give advice how to set up new scenarios or how to integrate new features into **GOTM**. It is desirable that such contributions are based on publications or submitted manuscripts (where the contributor does not necessarily have to be one of the authors). The contributions would then be published with the next **GOTM** release.

Chapter 6

Test Cases

The test cases and their simulations which are discussed in this chapter are all calculated either with **GOTM** or earlier versions of it. For the test cases, where earlier versions are used, slight deviations might occur in the results if recalculated with the present version of **GOTM**. The figures for those cases (channel flow (section 6.1.1) and the Irish Sea simulation (section 6.2.2)) have been copied from *Burchard et al.* [1998].

6.1 Idealized studies

6.1.1 Channel flow

As basic comparative tests for the performance of the models, two steady state barotropic channel flow situations, including comparisons with data, have been chosen: A surface stress forced flow which induces a constant stress over the vertical (Couette flow) and a pressure gradient-driven flow (e.g., tidal or river flow). Empirical data and numerical model results are displayed in figure 6.1 in non-dimensional form. For a more extensive discussion see *Burchard et al.* [1998].

6.1.1.1 Forcing by surface stress

If surface stress is the only forcing for steady-state open channel flow, then a constant stress over the whole water column and a solution symmetric to mid-depth results for all models. It can be shown furthermore that $P = \varepsilon$ and $k = (u_*^b/c_\mu)^2$ hold everywhere, if molecular viscosity is neglected. The solution for the turbulent kinetic energy k implies that for a constant stability function c_μ, the TKE will also be constant. In contrast to that, the empirical data in figure 6.1 strongly increase towards bottom and surface. This deficiency of the model is even given for the stability functions of *Mellor and Yamada* [1974] which depend on shear. This is because local equilibrium of turbulence ($P = \varepsilon$) directly implies $\alpha_M = (c_\mu^0)^3/c_\mu$, which means that α_M and c_μ (which is in this case with $\alpha_N = 0$ a function of α_M only, see equation (2.81)) are here not depending on the flow structure and are therefore constants. A solution to this problem could here be the introduction of wall proximity functions (see *Rodi* [1980]). Except at mid-depth where the measurements might not be reliable, the k-ε model result for eddy viscosity shows good agreement with the measurements. For the Mellor-Yamada model, a triangle-shaped length scale L_z had to

be prescribed rather than the usual parabolic-shaped one in order to achieve sufficient agreement between measurements and model simulations.

6.1.1.2 Forcing by pressure gradient

Here the only forcing is a surface slope which leads to a shear stress linearly decreasing from bottom to surface. As figure 6.1 shows, the models sufficiently reproduce the measured turbulent profiles with the limitations mentioned in the previous section 6.1.1.1: the increase of TKE towards the bottom is underestimated by all models. And for the Mellor-Yamada model, only a triangle-shaped L_z sufficiently reproduces the viscosity. It should be mentioned here, that a straight forward discretization of the k-ε model with Dirichlet boundary conditions and an averaging of eddy viscosity for turbulent quantities would lead to an eddy viscosity profile with an unrealistic maximum near the surface (see also *Burchard et al.* [1998]). Therefore, the flux boundary conditions (equation (2.41) for k and equation (2.49) for ε) have been used here.

6.1.2 Kato-Phillips experiment

In this experiment, a mixed layer induced by a constant surface stress penetrates into a stably stratified fluid with density increasing linearly down from the surface. The water depth is assumed to be infinite. *Price* [1979] suggested a solution for the evolution of the mixed-layer depth D_m based on a constant Richardson number

$$D_m(t) = 1.05 u_*^s N_0^{-1/2} t^{1/2}, \tag{6.1}$$

where u_*^s is the surface friction velocity and N_0 the constant initial Brunt-Väisälä frequency. Following several authors (see e.g. *Deleersnijder and Luyten* [1994], *Burchard et al.* [1998]) we transform this laboratory experiment to ocean dimensions with $u_*^s = 10^{-2}\ \mathrm{m\,s^{-1}}$ and $N_0 = 10^{-2}\ \mathrm{s^{-1}}$.

This simple numerical experiment allows here for a qualitative and quantitative comparison of various turbulence models as introduced in section 2.2. Turbulence measurements are unfortunately not available for this experiment. Figures 6.2 - 6.13 show profiles of mixing coefficients ν_t and ν_t' and turbulent kinetic energy k after 30 hours, and the development of the mixed-layer depth, which here is defined as the deepest point in the water column with $k > 10^{-5}\ \mathrm{W\,kg^{-1}}$. The discretization is carried out with a vertical resolution of $\Delta z = 0.25$ m and a time step of $\Delta t = 20$ s. The results will be discussed in the following.

Figures 6.2 - 6.11 show results for the k-ε and the Mellor-Yamada two-equation models, with different stability functions (see section 2.2.3) and with or without the length scale limitations (2.71) and (2.72). For the *Galperin et al.* [1988] stability functions and with a limitation of the length scale, the k-ε and the Mellor-Yamada model show nearly identical profiles of mixing coefficients and turbulent kinetic energy (see figures 6.2 and 6.3). Characteristic for the TKE-profiles are the maxima of k at about 18 m depth. However, for the Mellor-Yamada model, the length of the surface roughness elements had to be increased from $h_0^s = 0.05$ m to $h_0^s = 0.3$ m in order to achieve a numerically stable solution. It should be noted here, that the numerical stability of a scheme strongly depends on the actual discretization. For instance, the order of calculation of equations plays an important role. Here, as little as necessary was changed from simulation to simulation in order to allow for a good comparison of the various turbulence models.

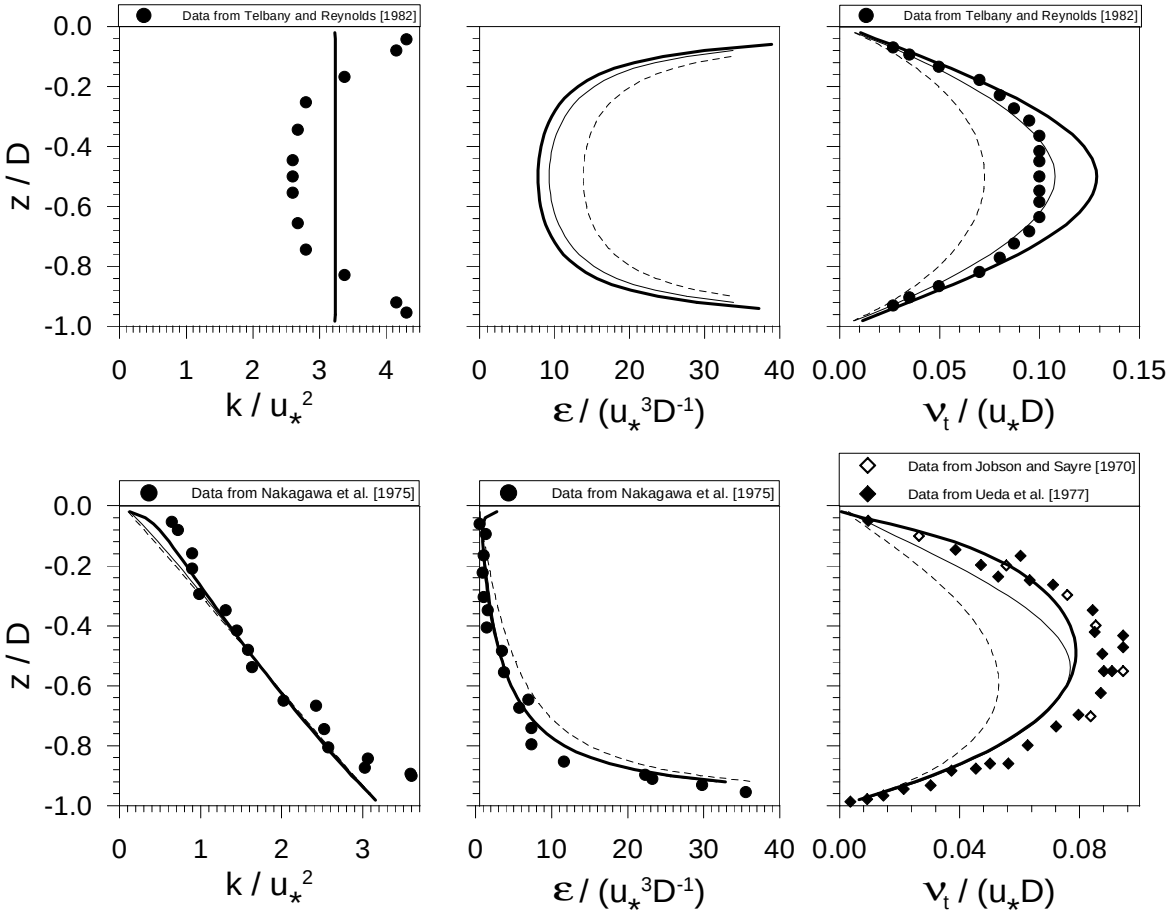


Figure 6.1: Simulations of barotropic open channel flow with comparison against data. (top) Stress-driven Couette flow. Data from *Telbany and Reynolds* [1982] (turbulent kinetic energy and eddy viscosity). (bottom) Pressure gradient-driven flow. Data from *Nakagawa et al.* [1975] (turbulent kinetic energy and dissipation rate), *Jobson and Sayre* [1970], and *Ueda et al.* [1977] (both eddy viscosity). Thick line, k-ε model; thin line, Mellor-Yamada model, triangle-shaped L_z, see equation (2.52); dashed line, Mellor-Yamada model, parabola-shaped L_z, see equation (2.51). This figure has been taken from *Burchard et al.* [1998].

Figures 6.4 and 6.5 demonstrate the importance of the length scale limitation for the two-equation models. The mixed-layer depth resulting from the k-ε model is slightly higher now, and is in good agreement with the empirical curve (this is, because the k-ε model used here had been calibrated to this curve, see *Burchard et al.* [1998]). It can be stated however, that the length scale limitation does not have a strong effect for the k-ε model result, and therefore seems to be intrinsically fulfilled by the model. This is no true for the Mellor-Yamada model. Without limitation, the Mellor-Yamada model mixes too strongly, which results into a mixed-layer depth more than 10 m higher than with limitation. It seems that if a length scale limitation is prescribed, that it applies over a large portion of the water column. This problem of the Mellor-Yamada model might be due to the fact that the built-in length scale L_z (see equations 2.51 and 2.52) linearly increases down from surface (due to the infinite water depth) and has a relatively high value in the mixing layer. The model results for both, the k-ε and the Mellor-Yamada model become numerically instable if the stability functions from *Mellor and Yamada* [1974] are used. It seems that the solution for the mixing coefficients oscillates in time and space between two solutions. These numerical problems have been discussed in more detail by *Deleersnijder and Luyten* [1994]. Nevertheless, the results for the turbulent kinetic energy and the mixed-layer depth look reasonable. This might be the reason why this kind of stability function has been in use for such a long time. These stability functions are still implemented in the well-known Modular Ocean Model (MOM), which is available from the Geophysical Fluid Dynamics Laboratory (GFDL) in Princeton, New Jersey[1]. In MOM, a vertical filtering of α_S is carried out in order to damp the oscillations. Test runs with that kind of filter with **GOTM** however still resulted in oscillating profiles for the eddy viscosity and diffusivity.
Figures 6.8 - 6.11 show model results for the two-equation models combined with simple stability functions, i.e. a constant c_μ and a constant or Richardson number depending c'_μ. These have been already used together with the k-ε model by *Rodi* [1980], and both give reasonable results (see figures 6.8 and 6.10). For the Mellor-Yamada model again, mixing is too strong, although here the length scale limitation (2.71) is applied.
The TKE maximum at about 18 m depth, which has been calculated by applying the stability functions of *Galperin et al.* [1988] and similarly also by those of *Mellor and Yamada* [1974], vanishes with these simpler stability functions (see figures 6.8 - 6.11). Unfortunately, TKE measurements for this experimental set-up, which could give preference to either set of stability functions, are not available.
Figures 6.12 and 6.13 finally show model results for one-equation models, where the length scale is directly calculated by an algebraic formula from mean flow properties (see section 2.2.2.2). The mixing length parameterization of *Eifler and Schrimpf* [1992] slightly overestimates the empirical mixed layer depth, although the parameter e (see equation (2.68)) had been tuned up to $e = 3$. A dependence of the numerical solution on the minimum length scale $L_{\min}$ was found, the present simulation was carried out with $L_{\min} = 0.002$ m. The algebraic algorithm from *Bougeault and André* [1986]) is numerically noisy and shows strong oscillations for the mixed-layer depth. Also a reduction of the tome step to $\Delta t = 5$ s did not produce a smoother numerical solution. However, the model predicts the overall mixed layer depth evolution well. The cause for these instabilities might be that the length scales L_k and L_ε needed for the calculation of the length scale is here available only in discrete steps due to the vertical discretization of the model. In order to avoid these unwanted oscillations in three-dimensional models, horizontal filtering of parameters (such as shear) which determine the turbulent properties might improve the model performance. However, it could also be discretization details within **GOTM** which cause the noise produced by the model.

[1] The Modular Ocean Model can be downloaded from the following web site: `http://www.gfdl.gov/~kd/MOMwebpages/MOMWWW.html`.

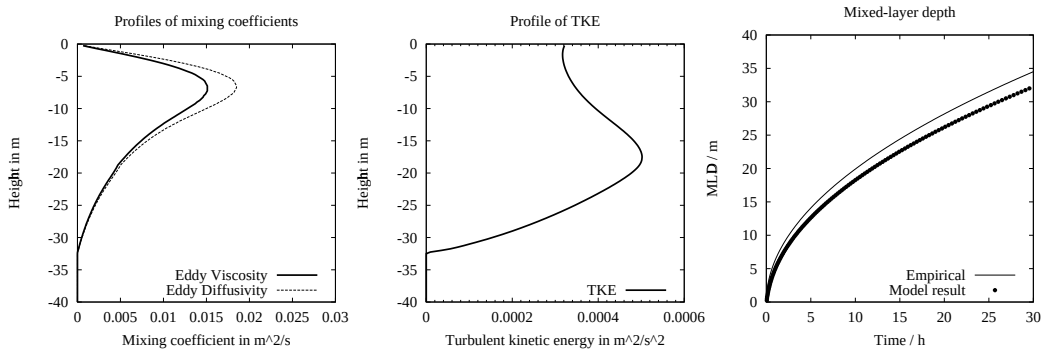


Figure 6.2: Profiles of mixing coefficients and turbulent kinetic energy after 30 hours, and development of the mixed-layer depth (deepest point with $k > 10^{-5}$ W kg^{-1}) for the simulation of the Kato-Phillips experiment. Model results for the k-ε model with stability functions and length limitation according to *Galperin et al.* [1988].

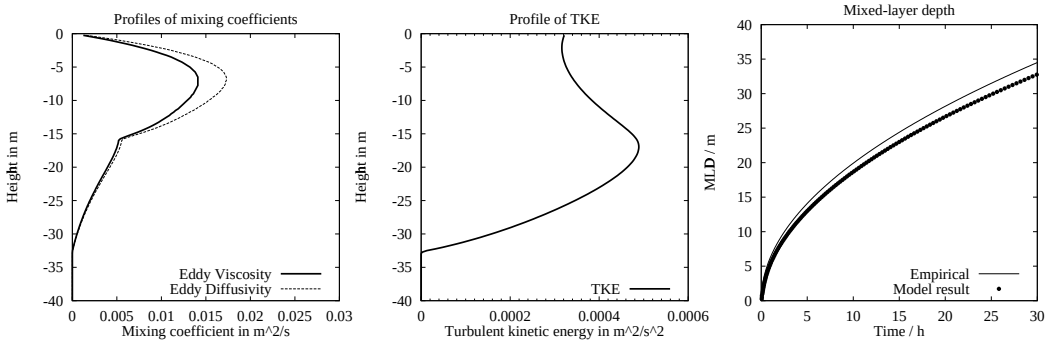


Figure 6.3: Same as figure 6.2, but for the Mellor-Yamada model with stability functions and length limitation according to *Galperin et al.* [1988].

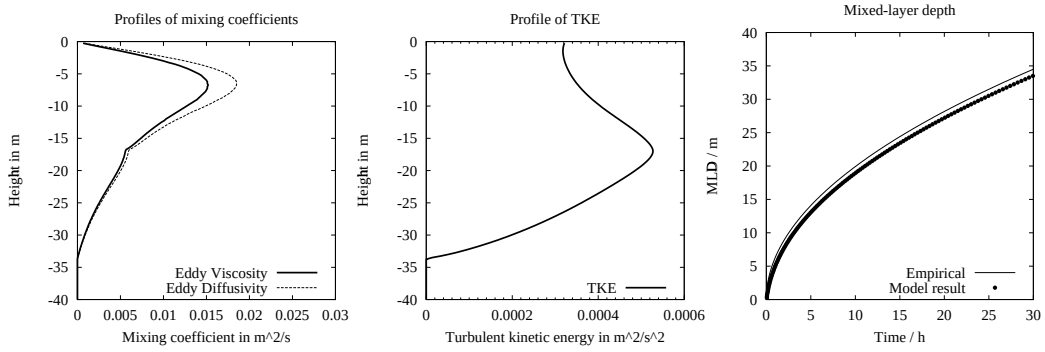


Figure 6.4: Same as figure 6.2, but for the k-ε model with stability functions from *Galperin et al.* [1988] and without length limitation.

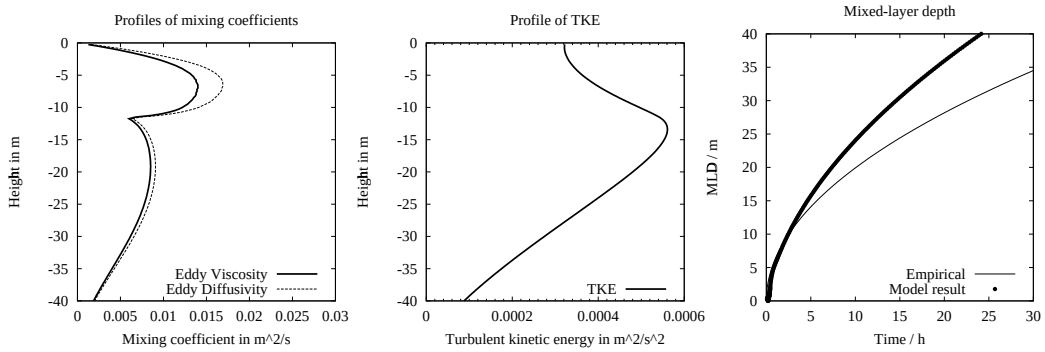


Figure 6.5: Same as figure 6.2, but for the Mellor-Yamada model with stability functions from *Galperin et al.* [1988] and without length limitation.

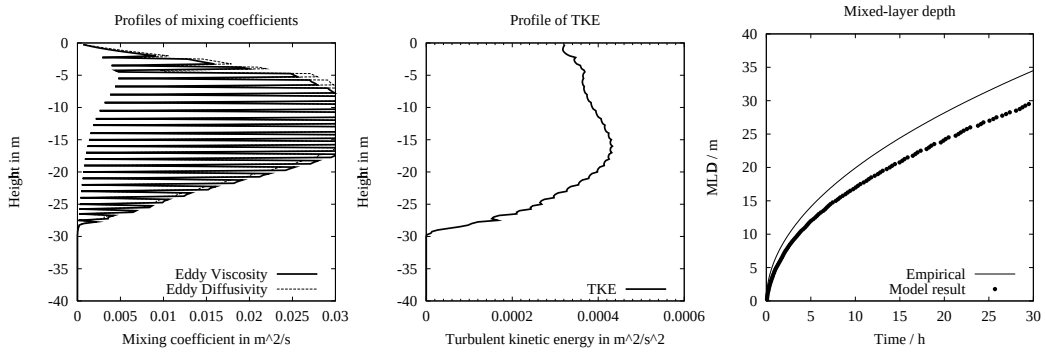


Figure 6.6: Same as figure 6.2, but for the k-ε model with stability functions from *Mellor and Yamada* [1974] and length limitation according to *Galperin et al.* [1988].

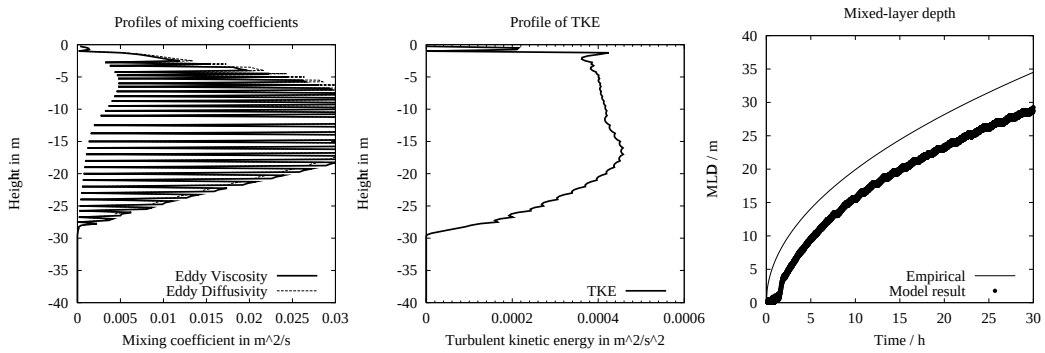


Figure 6.7: Same as figure 6.2, but for the Mellor-Yamada model with stability functions from *Mellor and Yamada* [1974] and length limitation according to *Galperin et al.* [1988].

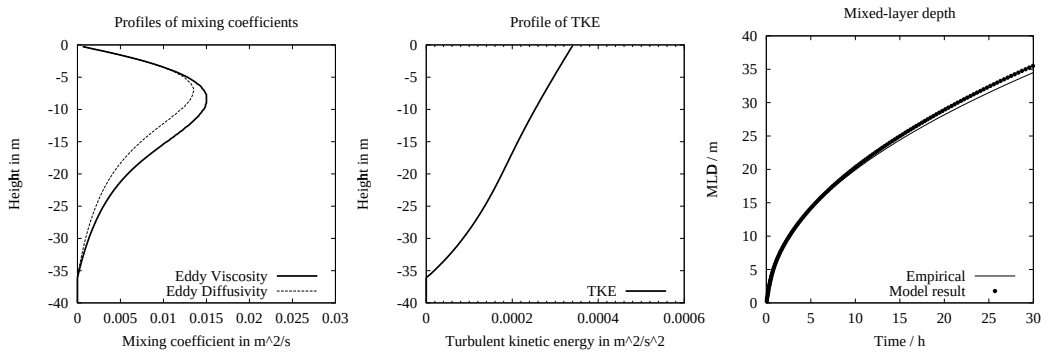


Figure 6.8: Same as figure 6.2, but for the k-ε model with stability functions from *Munk and Anderson* [1948] and length limitation according to *Galperin et al.* [1988].

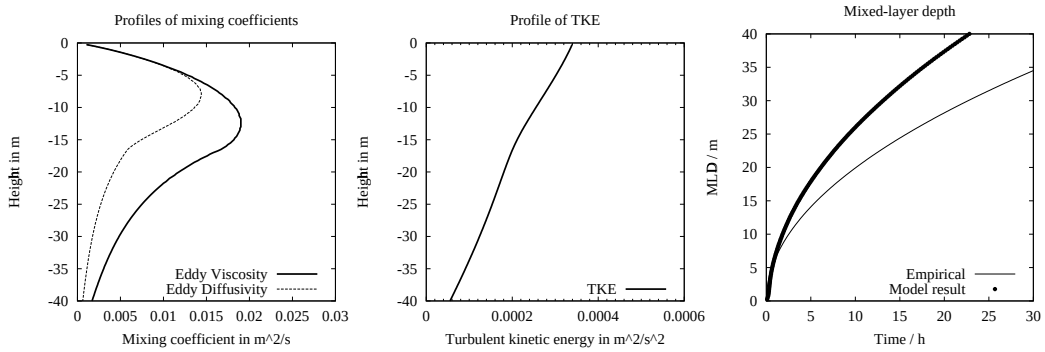


Figure 6.9: Same as figure 6.2, but for the Mellor-Yamada model with stability functions from *Munk and Anderson* [1948] and length limitation according to *Galperin et al.* [1988].

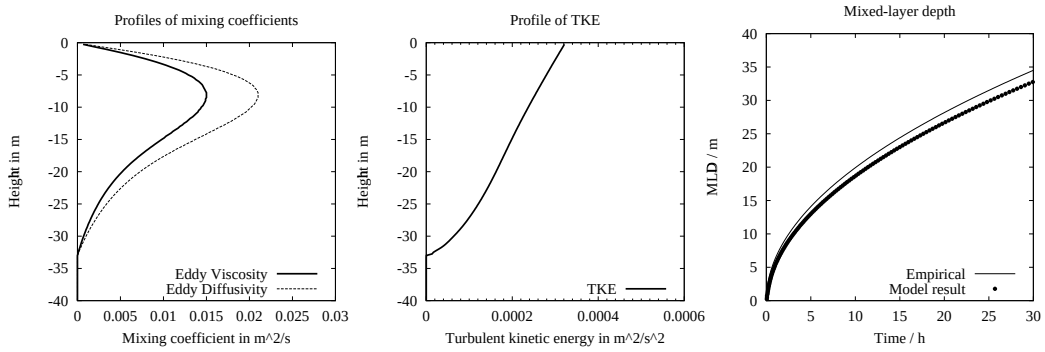


Figure 6.10: Same as figure 6.2, but for the k-ε model with constant stability functions and length limitation according to *Galperin et al.* [1988].

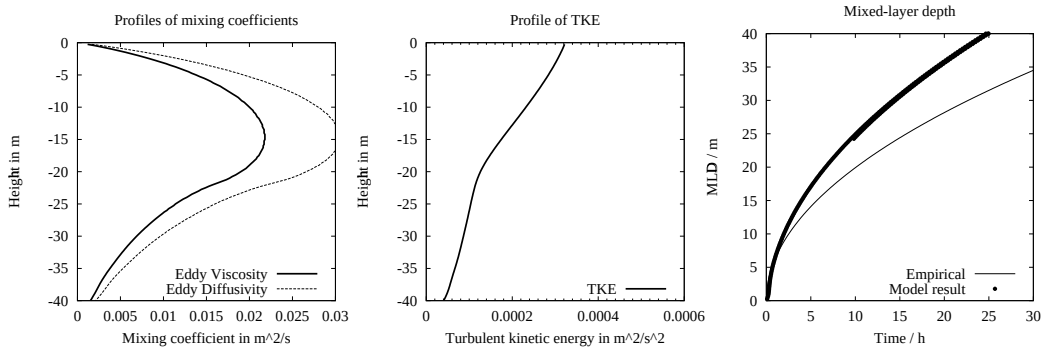


Figure 6.11: Same as figure 6.2, but for the Mellor-Yamada model with constant stability functions and length limitation according to *Galperin et al.* [1988].

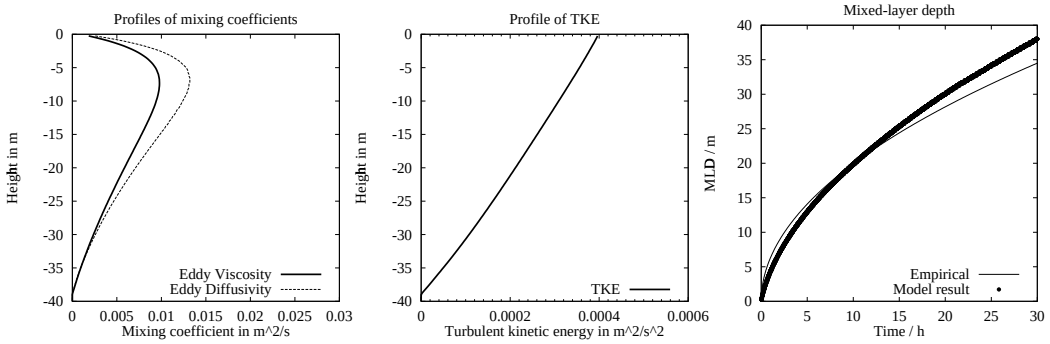


Figure 6.12: Same as figure 6.2, but for the one-equation model (only TKE equation calculated prognostically) with algebraic length scale calculation according to *Eifler and Schrimpf* [1992] without length limitation.

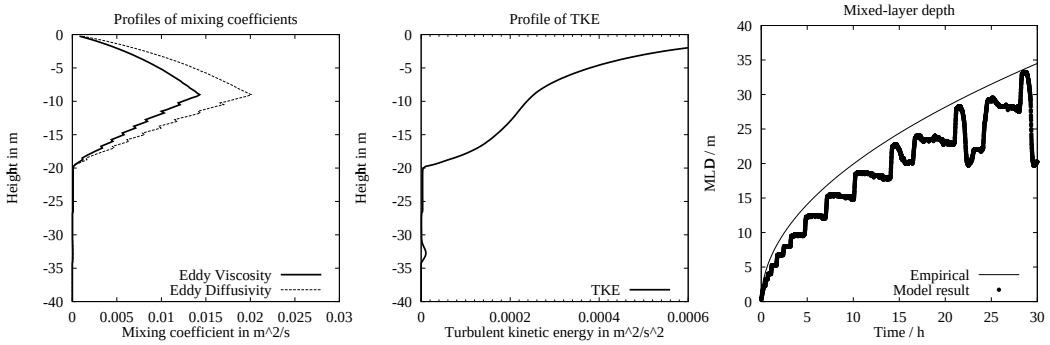


Figure 6.13: Same as figure 6.2, but for the one-equation model (only TKE equation calculated prognostically) with algebraic length scale calculation according to *Bougeault and Andre* [1986] without length limitation.

6.2 Open sea studies

6.2.1 Fladenground experiment

A data set which has been throughout the last 20 years used as a calibration for mixing parameterizations has been collected during the measurements of the Fladenground Experiment 1976 (FLEX'76) campaign. These measurements of meteorological forcing and temperature profiles were carried out in spring 1976 in the northern North Sea at a water depth of about 145 m and a geographical position at $58^o55'$N and $0^o32'$E. Turbulent quantities have not been measured. For further details concerning the measurements, see *Soetje and Huber* [1980] and *Brockmann et al.* [1984]. This FLEX'76 data set has been used by several authors in order to test different mixing schemes (see e.g. *Friedrich* [1983], *Frey* [1991], *Burchard and Baumert* [1995], *Pohlmann* [1997], *Burchard and Petersen* [1999]). Some of the results shown here are taken from *Burchard and Petersen* [1999].

Surface friction velocity and surface buoyancy flux (including solar radiation, which is penetrated into the water column by means of attenuation lengths fitted to turbidity measurements) are shown in figure 6.14. An important feature of this surface forcing is the storm on Julian day 133 which is following a relatively calm period. The consequences of this storm can be seen in the temperature profile measurements as well as in the model results for temperature (see figure 6.16). The near surface stratification is eroded during this storm and homogenised down to 50 m (measurements) and 40 m (model results) with the consequence that the sea surface temperature jumps down by approximately 0.5 K (see figure 6.15). The underestimation of the mixed-layer depth by the model might be due the neglect of mixing due to internal wave or shear instability which is typical for differential turbulence closures as discussed by *Martin* [1985]. Another reason might be a too small surface stress as calculated from meteorological measurements.

Figure 6.17 shows profiles of temperature for the k-ε and the Mellor-Yamada model with different vertical resolutions before and after the storm. It is demonstrated that the models are discretized such that even for a coarse resolution of about 10 m ($M = 14$ layers), the effect of the storm is well reproduced, although the mixed layer before the storm is not resolved. As already discussed in section 3.5, only a careful numerical treatment of the k-ε model near the surface allows for this good performance (see also *Burchard and Petersen* [1999]).

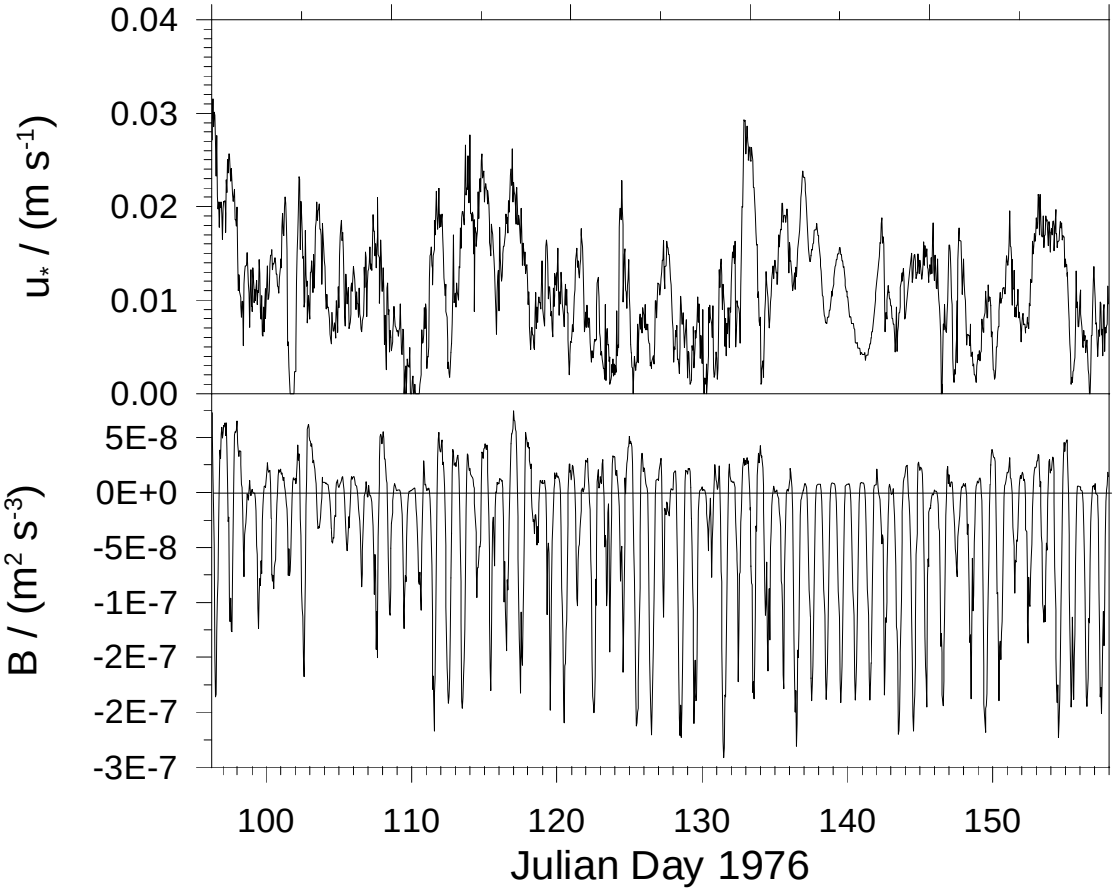


Figure 6.14: Time series of surface forcing for the FLEX'76 simulation. Upper panel: Friction velocity u_*; Lower panel: Buoyancy flux B. It should be noted that the solar radiation which is here included in B, is absorbed in the upper layers of the model by using an exponential law. This figure has been taken from *Burchard and Petersen* [1999].

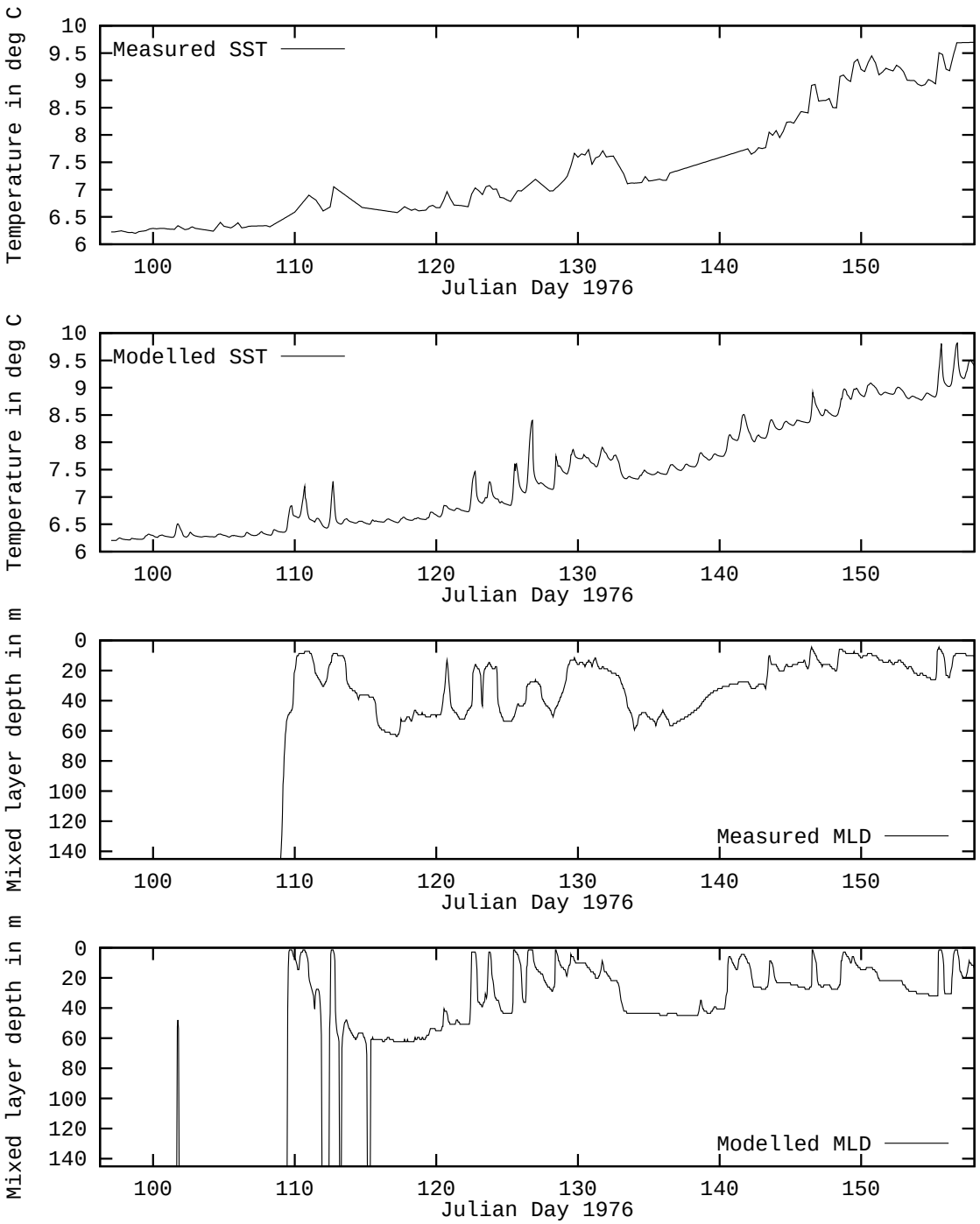


Figure 6.15: Simulation of the temperature measurements during FLEX'76. The upper two panels show measured and modelled sea surface temperature, and the lower two show measured and modelled mixed layer depth (calculated according to the temperature difference criterium using $\Delta T = 0.25$ K).

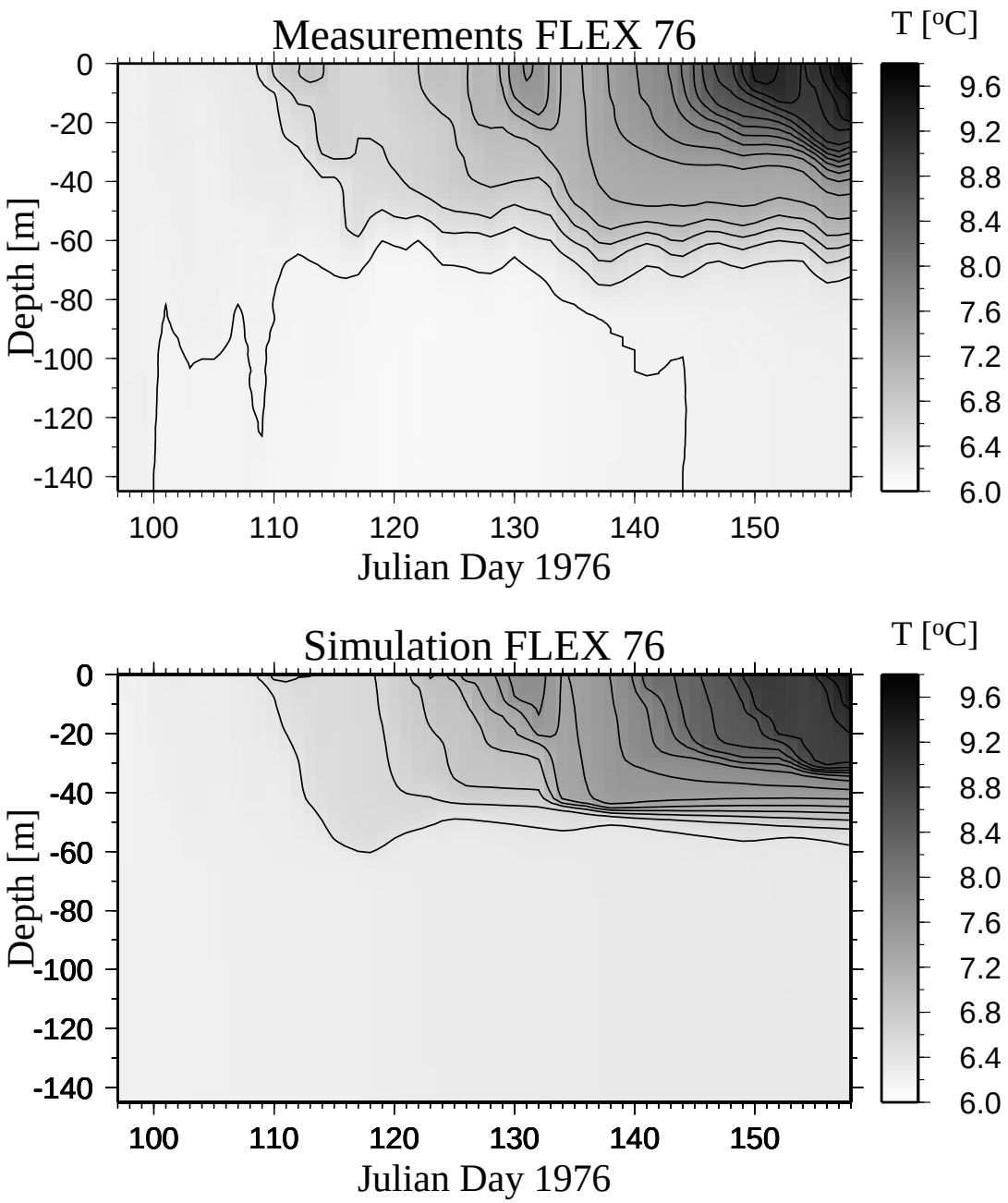


Figure 6.16: Isopleth diagram of temperature distribution during FLEX'76. Upper panel: CTD measurements; Lower panel: k-ε model results. The Mellor-Yamada model results were similar to these.

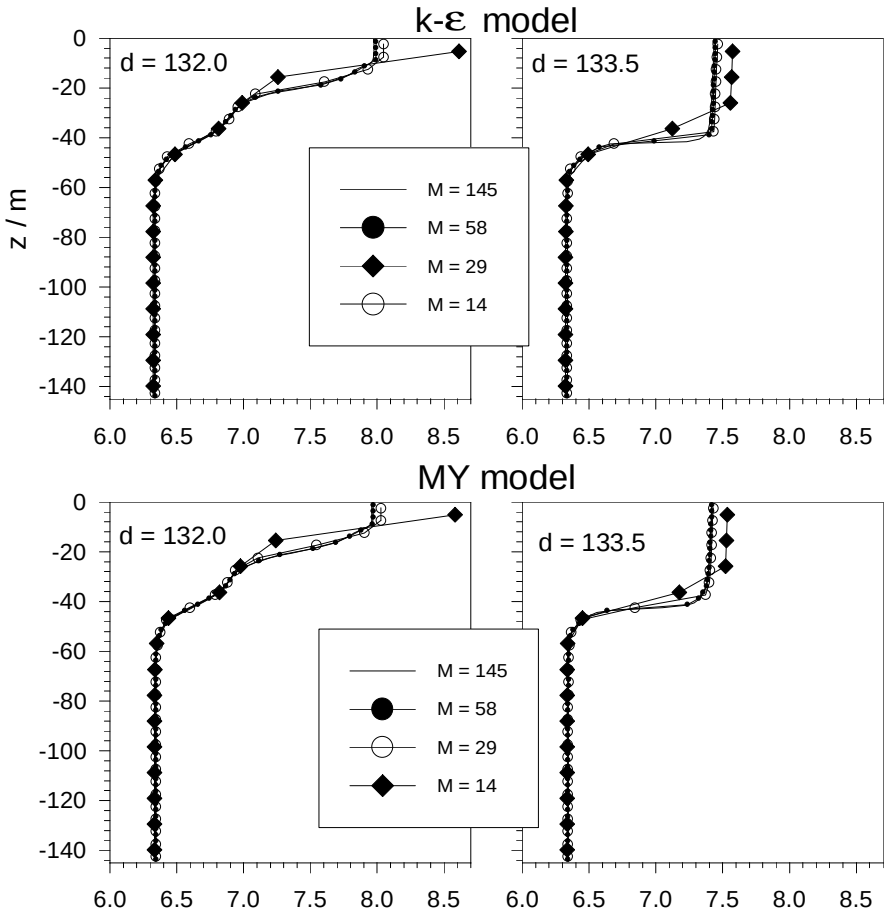


Figure 6.17: Profiles of modelled temperature for different vertical resolutions. Results are shown for strong near surface stratification before a storm (d=132) and for a well-mixed situation after a subsequent storm (d=133.5). The upper panels show k-ε model results, the lower panels Mellor-Yamada model results. This figure has been taken from *Burchard and Petersen* [1999].

6.2.2 Irish Sea

In the last years, various in-situ turbulence measurements have been carried out in shelf and coastal seas. Here, we are going to simulate the Irish Sea measurements of *Simpson et al.* [1996], which include observation of the dissipation of turbulent kinetic energy. The measurements were made over a 24 hour period in July 1993 at a site with water depth of 90 m and position 53°49'N, 5°27'W. The site has rectilinear tidal currents, and the tidal current amplitude at the time of the measurements was 0.45 m s^{-1}. During the observational period the water column was thermally stratified with near surface temperature of 14°C and temperatures of 10°C below the thermocline, which was situated 75 m above the sea-bed. In addition to hourly conductivity-temperature-depth profiles and moored current meter measurements, turbulent dissipation was measured using a FLY free-fall probe. Six profiles were made per hour, and these were averaged together to give hourly means. Measurements made in the top 10 m of the water column have been excluded as the probe may still be accelerating, and the signal may be contaminated by the ships wake.
The forcing data for the model is provided by a time series of current measurements taken 12 m above the bed, from which a sea surface slope is calculated so that the model exactly reproduces the observed flow (see section 2.1.2.1). Cross-surface heat exchange is calculated from dew point temperature, wind speed, and direct solar radiation observed at a nearby meteorological station. Cross-surface momentum exchange is calculated from the observed wind speed and direction (following *Simpson et al.* [1996]).
Figure 6.18 shows isopleth diagrams of measured and modeled dissipation rate. The model results are shown for the stability functions by *Galperin et al.* [1988] and the simple internal wave model by *Luyten et al.* [1996b] and reproduce the observed M_4 period oscillations in the bottom layer (caused by the flooding and ebbing of the predominant semi-diurnal M_2 tide). The time lag between dissipation rate and bottom friction, which is increasing with distance from the bottom, can be clearly seen in the measurements (figure 6.18). This is well reproduced by the simulations with both of the models. However, the variability of the dissipation rate in the stratified core region of the flow cannot be resolved by the simple internal wave model included.
The similarity between the k-ε and the Mellor-Yamada model, which is demonstrated in figure 6.18, has been extensively discussed by *Burchard et al.* [1998], where further details of this simulation are given. Earlier model simulations of the Irish Sea measurements have been carried out by *Simpson et al.* [1996] and *Luyten et al.* [1996b]. In those papers, measurements at a well-mixed site in the Irish Sea have been successfully simulated as well.

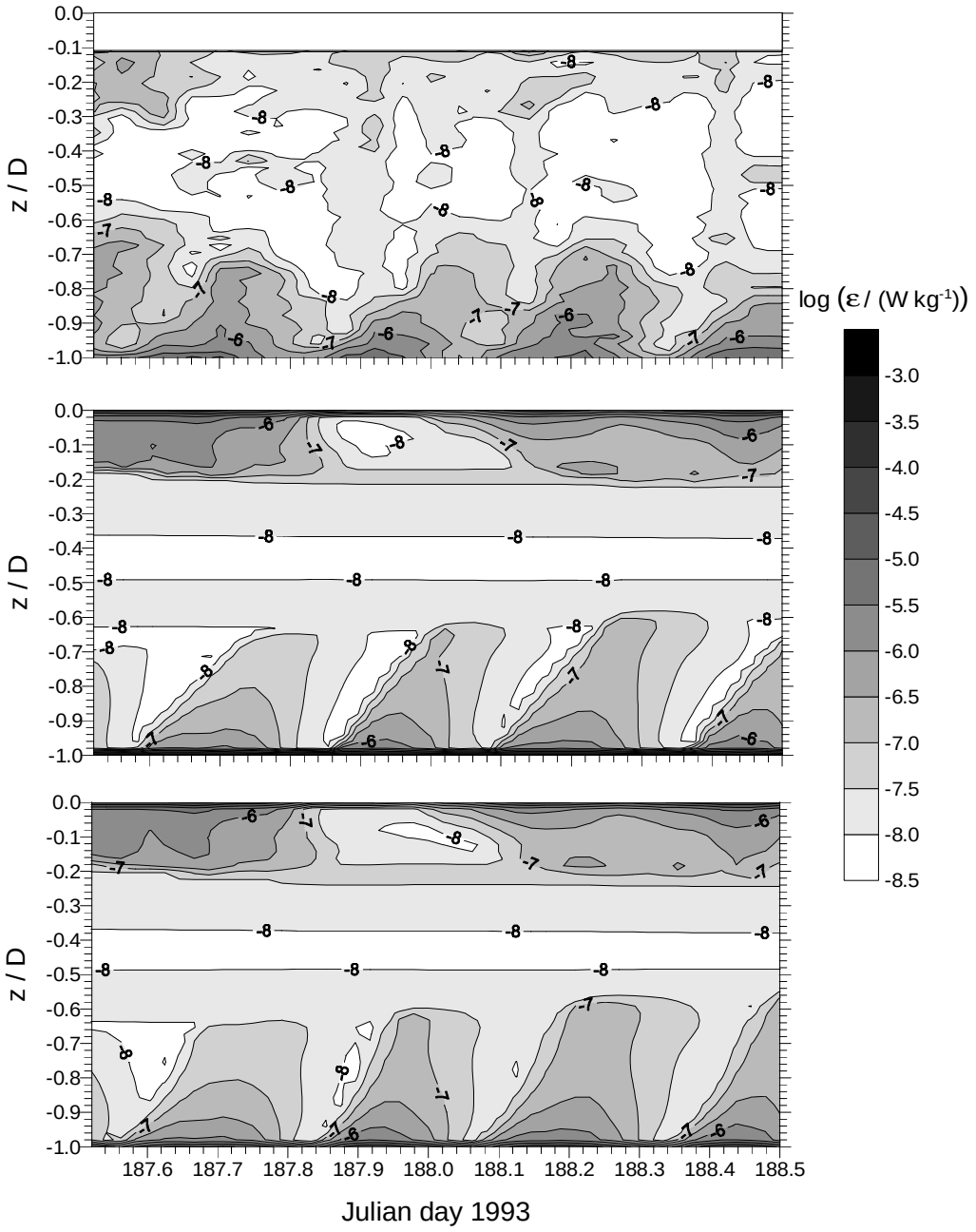


Figure 6.18: Irish Sea simulation: Isopleth diagrams showing the development of dissipation rate during 24 hours in summer 1993. (top) Measurements, (middle) k-ε model result, (bottom) Mellor-Yamada model result. For both model simulations, the *Galperin et al.* [1988] stability functions and the simple internal wave model by *Luyten et al.* [1996b] have been used. This figure has been taken from *Burchard et al.* [1998].

6.2.3 Ocean Weather Stations Papa and November

For the Northern Pacific, long term observations of meteorological parameters and temperature profiles are available. The stations Papa at 145°W, 50°N and November at 140°W, 30°N have the advantage that they are situated in regions where the horizontal advection of heat and salt is assumed to be small. Various authors used these data for validating turbulence closure schemes (*Denman* [1973], *Martin* [1985, 1986], *Large et al.* [1994], *Kantha and Clayson* [1994], *D'Alessio et al.* [1998]). As any other realistic oceanic test case, also other factors than the choice of the mixed-layer model play an important role for the agreement between the model results and the measurements. First of all measurements such as fractional cloud cover are never exact. Furthermore, horizontal advection of heat and salt, which is neglected in one-dimensional water column models, can strongly influence the measured profiles of temperature and salinity. And, maybe the most important, the bulk formulae for the parameterization of cross surface fluxes of momentum, heat and fresh water are strictly empirical.

For the stations Papa and November, meteorological measurements of wind speed and direction, sea surface temperature, air temperature, wet bulb temperature, dew point temperature, cloud cover and air pressure are available. From those we calculate the surface fluxes needed as boundary conditions for the momentum and temperature equation and as inner source for the latter. The wind stress, and the sensible and latent heat flux are calculated according to the bulk formulae of *Kondo* [1975]. The long-wave back radiation is calculated with the aid of the formula of *Wyrtki* [1965]. For the incoming short-wave radiation we basically follow *Rosati and Miyakoda* [1988] who calculate in their circulation model for the global ocean the solar radiation under clear skies according to *List* [1958] and the cloud cover correction according to *Reed* [1977].

With an atmospheric transmission coefficient of $\tau = 0.7$ (as in *Rosati and Miyakoda* [1988]) we achieve for the station November site for the whole year 1961 an average total heat loss of 8 $\mathrm{W\dot{m}^{-2}}$. This is a bit more than *Martin* [1985] calculates with the formulae he used (2 $\mathrm{W\dot{m}^{-2}}$ loss). For the November station, the five day averages of heat content in the upper 250 m of the water column (calculated using the measured profiles in the upper 250 m, see figure 6.19) indicate a heat loss of that order (it should be noted that the measured heat content jumps down by about 10^9 $\mathrm{J\,m^2}$ within the first four days of 1961). However, the overall agreement between accumulated surface heat flux and heat content calculated from the measured temperature profiles is rather poor. This might be due to gaps in the measurements, advective events or deficiencies in the surface heat flux parameterizations.

For the station Papa, the use of an atmospheric transmission coefficient of $\tau = 0.7$ results in an average heat loss of 30 $\mathrm{W\dot{m}^{-2}}$. This contradicts to the general assumption of a heat access for the station Papa which is balanced by winter advection of cold water in the seasonal thermocline (see *Large et al.* [1994]). Here, we do not apply any atmospheric correction other than the cloud correction. With this, the heat flux calculated by *Large et al.* [1994] for the ocean year 1961 (from March 1961 to March 1962) is well reproduced. From late October 1961 on, heat content and surface heat flux start to be inconsistent. In nature, this is caused by advection of cold water below the seasonal thermocline (see *Large et al.* [1994] and figure 6.21 of this report).

For the station Papa, *Large et al.* [1994] initialize their annual cycle in March because then the heat content of the water column reaches a minimum. As an initial profile for the temperature T they used a 5-day-average around March 15. Instead of that, we initialize our model for the station Papa at March 25 with the instantaneous T-profile. Due to the similar surface heat fluxes, our model performance can directly be compared with that of the model of *Large et al.* [1994], who used the so-called K-profile parameterization (KPP-model) for the eddy viscosity and diffusivity calculation.

For both stations, the water depth in the model was set to 250 m, deep enough for avoiding that surface mixing reaches the bottom. The initial conditions for temperature were taken from the measurements, initial salinity profiles are interpolated from monthly mean climatological profiles (*Levitus* [1992]). The surface salinity flux was set to zero, a condition which is acceptable for annual model runs. Initial velocities were set to zero, turbulent quantities to their minimum values. Following *Martin* [1985], attenuation coefficients from *Jerlov* [1968] are used, type II (turbid water) for station Papa, and type I (clear water) for station November. *Large et al.* [1994] show in a sensitivity study how strongly different water type specifications influence the sea surface temperature. For the simulation of both sites, a k-ε model equipped with the *Galperin et al.* [1988] stability functions is used. The dissipation rate was constrained by a lower limit given by (2.72). As internal wave parameterization, four different approaches have been used:

- No internal wave parameterization, $k_{\min}$ was set to the very low value of 10^{-10} J kg^{-1}.
- The internal wave parameterization of *Mellor* [1989], see section 2.2.4.2.
- The internal wave parameterization of *Kantha and Clayson* [1994], see section 2.2.4.2.
- The limitation of the turbulent kinetic energy according to *Luyten et al.* [1996b] of $k > k_{\min} = 3 \cdot 10^{-6}$ J kg^{-1}.

The vertical discretization is carried out with a constant layer depth of $\Delta z = 1$ m, the time step is $\Delta t = 120$ s, although even a time step of $\Delta t = 1$ h gave numerically stable results with an r.m.s. deviation of less than 0.3°C relative to the SST achieved with $\Delta t = 120$ s.
The mixed layer depth is here defined as the distance from the surface where the temperature is first by 0.1°C colder than at the surface (see *Martin* [1985]). We use this rather simple definition in order to be able to calculate a mixed layer from the measured profiles which is comparable to the modelled mixed layer depth.

6.2.3.1 Ocean Weather Station Papa

An inspection of figure 6.20 shows that only the internal wave parameterization of *Kantha and Clayson* [1994] produces a calculated sea surface temperature which is comparable with the measurements. The other three methods overestimate the maximum sea surface temperature by about 2°C. However, after the temperature peak around day 230, the SST calculated by the *Kantha and Clayson* [1994] model sinks too rapidly, whereas the measurements suggest that the SST should stay around that value until day 270. Figure 6.21 shows an explanation for that unwanted temperature decrease: In the *Kantha and Clayson* [1994] model simulations, the near surface heat is permanently diffused downwards through the thermocline whereas the thermocline as indicated by the measurements retains a strong temperature gradient. For the station Papa simulations, *Kantha and Clayson* [1994] do not show an isopleth diagram like in figure 6.21. Therefore, it can only be assumed that they have had the same problems with the *Mellor and Yamada* [1982] two-equation model than we had with the k-ε model, which usually perform similar. A figure similar to figure 6.21 is shown by *Large et al.* [1994]. There, a strong temperature gradient is retained in the simulations using the KPP-model. It should be noted that *Large et al.* [1994] use the same internal wave parameterization than *Kantha and Clayson* [1994], which seems to perform better with the KPP-model.
The overestimation of the sea surface temperature (equivalent to an underestimation of the mixed-layer depth, see figure 6.20) by the model simulation without any internal wave parameterization

is in agreement with the result of *Martin* [1985] who found that differential models in contrast to integral models tend to underestimate the mixed layer depth. Following *Large et al.* [1994], an increase of more than 1°C in maximum SST could also be caused by a change from Jerlov water type II to IA. After day 320, when the surface heat flux is much higher than the measured warming of the water column, the modelled SST is for all four model simulations decreasing more slowly than the measured. This discrepancy is due to the winter advection in the seasonal thermocline and its subsequent mixing into the mixed layer as discussed by *Large et al.* [1994].

6.2.3.2 Ocean Weather Station November

Although the agreement between accumulated surface heat flux and measured content of the water column is poor for the station November, the overall agreement between measured and simulated mixed-layer depth is fairly good during summer. This might be due to well resolved surface momentum fluxes. This good agreement in measured and simulated mixed-layer depth is not reflected by the sea surface temperatures. An inspection of the measured temperature development (see figure 6.23) reveals a strong internal wave activity which might be responsible for this discrepancy. It seems, that generally the station Papa data are more suited for one-dimensional simulations than these November data.

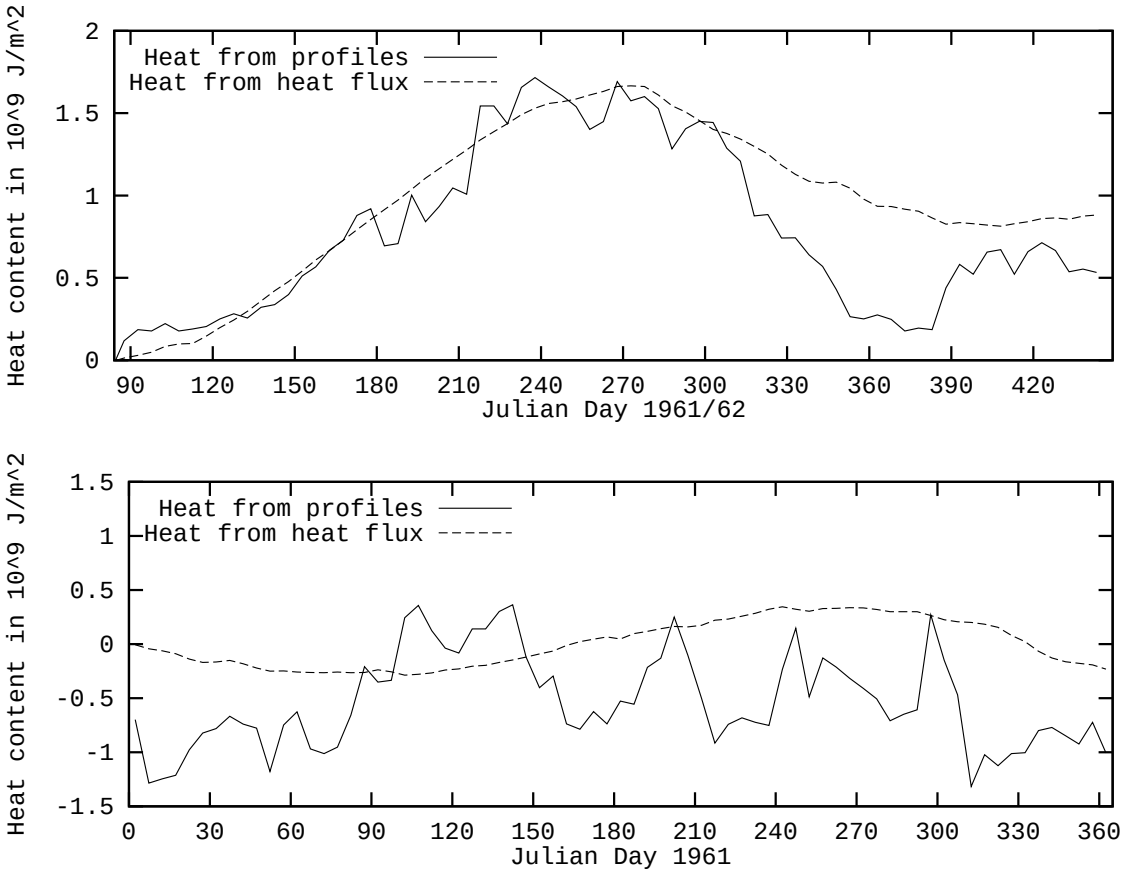


Figure 6.19: Net heat fluxes for the stations Papa and November for the year 1961 calculated by measured temperature profiles and by accumulation of surface heat fluxes calculated by using bulk formulae and other parameterizations (see text). Upper panel: station Papa, lower panel: station November. The data shown are 5-day averages.

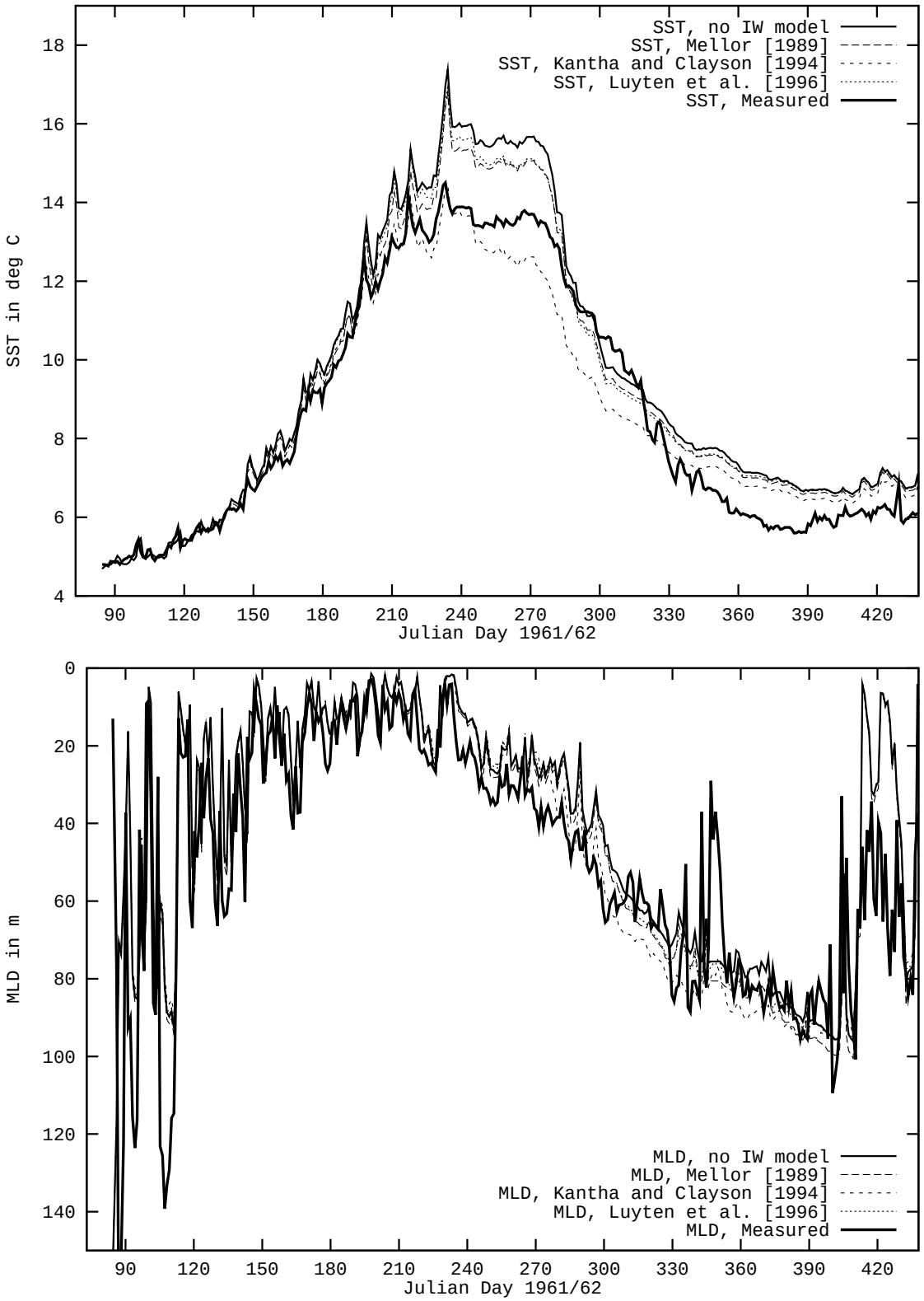


Figure 6.20: Simulation of the OWS Papa data for the year 1961/62. The upper panel shows measured and modelled sea surface temperature, and the lower shows measured and modelled mixed layer depth.

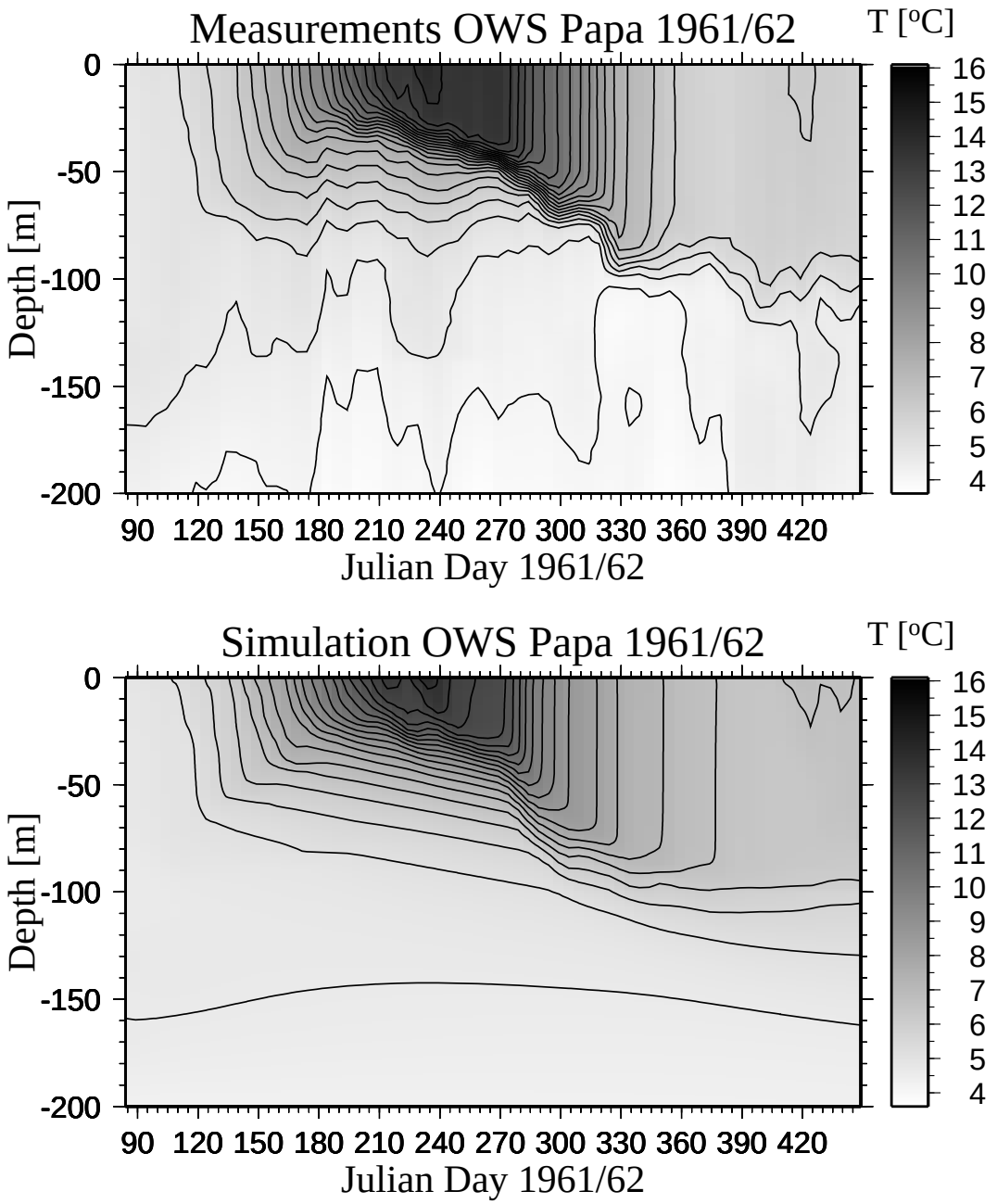


Figure 6.21: Isopleth diagram of temperature distribution during 1961/62 at the Ocean Weather Station Papa. Upper panel: CTD measurements; Lower panel: k-ε model results with *Kantha and Clayson* [1994] internal wave parameterization.

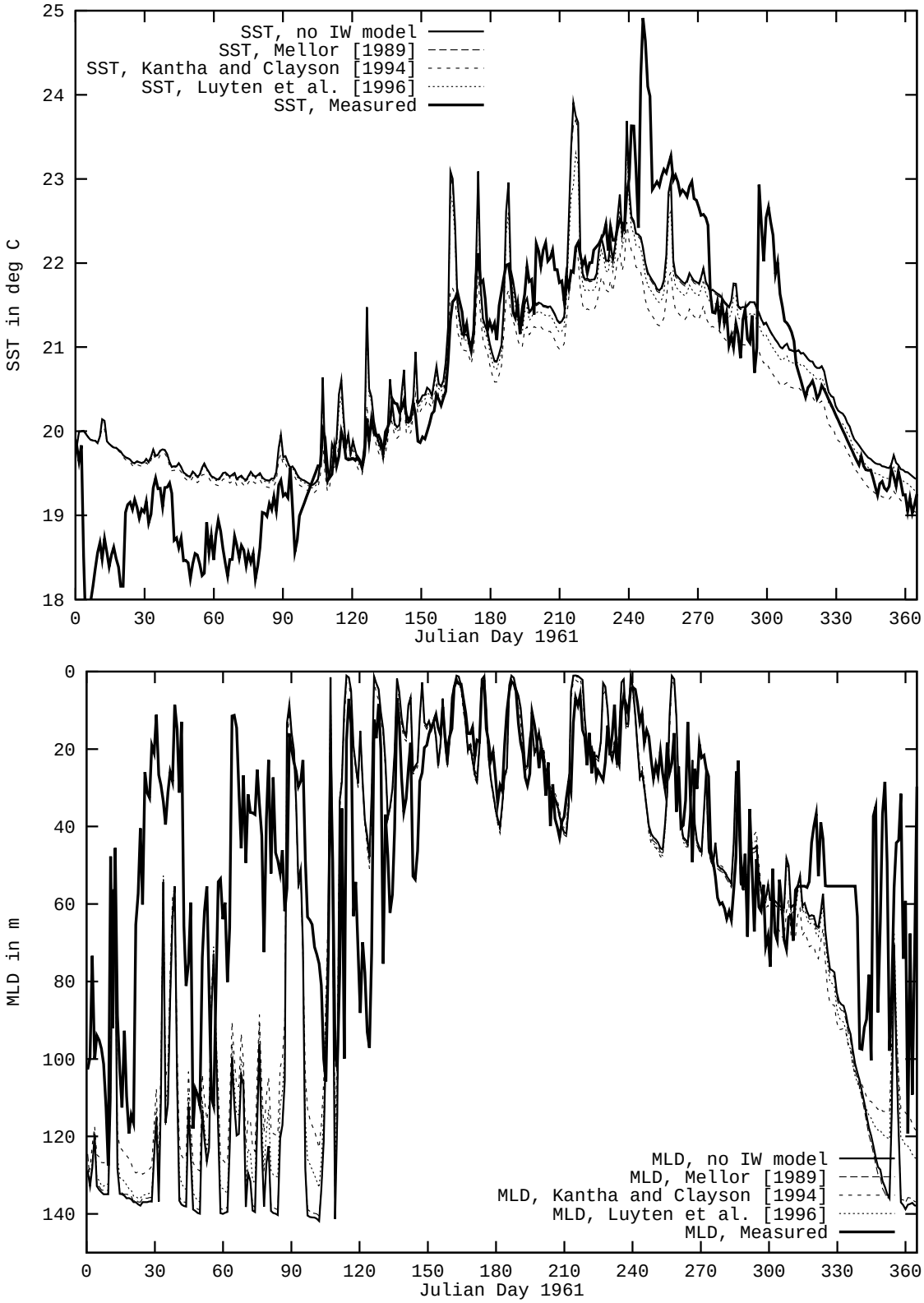


Figure 6.22: Simulation of the OWS November data for the year 1961. The upper panel shows measured and modelled sea surface temperature, and the lower shows measured and modelled mixed layer depth.

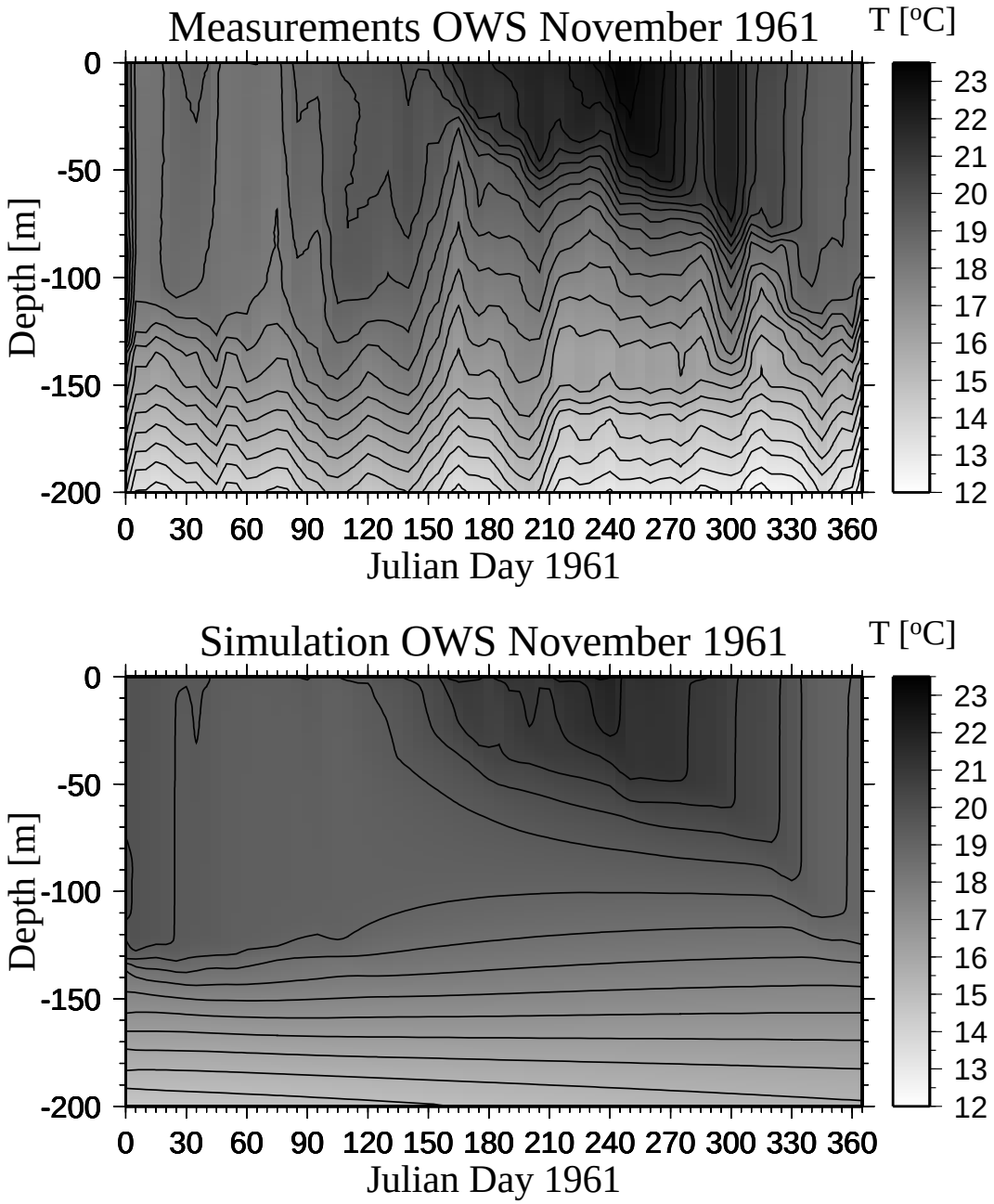


Figure 6.23: Isopleth diagram of temperature distribution during 1961 at the Ocean Weather Station November. Upper panel: CTD measurements; Lower panel: k-ε model results with *Kantha and Clayson* [1994] internal wave parameterization.

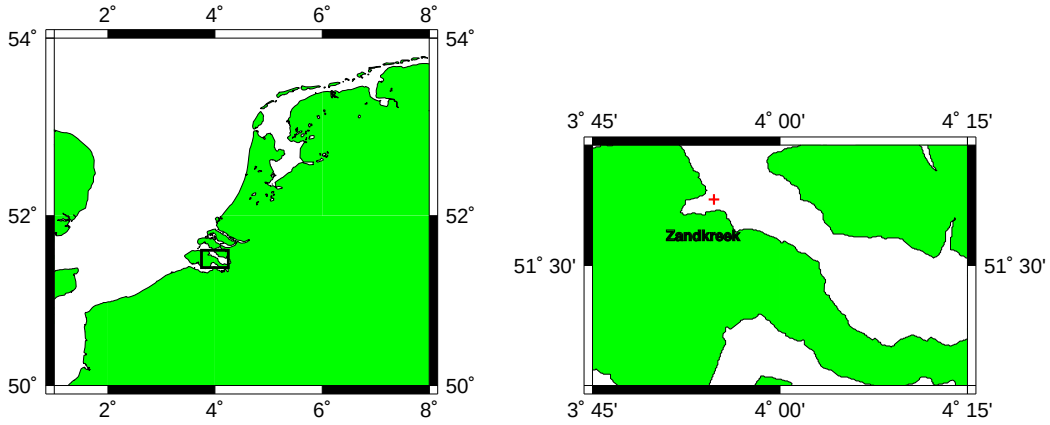


Figure 6.24: Contours of The Netherlands (left) with a blow-up of the Zandkreek region (right).

6.3 Estuarine studies

In this section, the attempt to directly simulate in-situ measurements of dissipation rate in two different estuaries with the aid of **GOTM** is discussed (see *Burchard et al.* [1999]). Here, only the k-ε model with the *Galperin et al.* [1988] stability functions and the length limitation (2.72) is used. The two estuaries investigated in this study were the weakly stratified Eastern Scheldt in the southern part of The Netherlands, and the strongly stratified Knebel Vig at the east coast of Jutland in Denmark. These measurements were carried out with free falling and rising MST (micro-structure-turbulence) shear probes also equipped with a conductivity and a fast temperature sensor.

6.3.1 Eastern Scheldt

The Eastern Scheldt is a tidal inlet in Zeeland, The Netherlands, with a length of about 40 km. It is separated from the North Sea by a storm surge barrier (built 1984-1987). Compared to other estuaries, the freshwater discharge from rivers into the Eastern Scheldt is very small. Therefore, and due to the high flow velocities, vertical density stratification and an estuarine circulation typical for tidal estuaries with high freshwater discharge cannot develop. The Eastern Scheldt consists of a complex bathymetry with tidal channels up to a water depth of over 40 m and extensive tidal flats. The M_2 tide is predominant with a tidal range of about 4 m.
The measurements which are simulated here have been carried out in June 1996 at 51° 33' N, 03° 54' E. The site was located at a mean water depth of 15 m at the side of a curved tidal channel near the mouth of the Zandkreek, a narrow side arm of the Eastern Scheldt (see figure 6.24).
The forcing of the model is provided by current meter, sea surface elevation and wind measurements. The current meter measurements were taken 6.8 m above the bottom and low pass filtered

with a cut-off period of 1800 s. The tidal elevation was taken from the tidal gauge at Stavenisse, 8 km from the measurement site, but with the same distance from the storm surge barrier. The wind speed was directly measured at the site, the wind direction was provided from a land station at Wilhelminadorp, 3 km south from the site. The wind during the second part of the measurements was weak with wind speeds of about 2 ms^{-1}. At the site, dissipation rate measurements were carried out with a profiler in the free falling mode only, with the consequence that the upper 3 m of the water column are not covered by measurements. The sampling frequency was relatively low with only 3 profiles per hour. Therefore, the resulting dissipation rate has a high noise level. However, the tidal cycle with a maximum of about 10^{-6} $W\,kg^{-1}$ at mid-depth is clearly visible. For the site under investigation here, maximum concentrations of fine sediments of about 25 mg/l at mid-depth and a smooth and slippery bed have been observed.
In figure 6.25, measurements and model results for the dissipation rate are shown for a period of almost three days. A strong overestimation of dissipation rate results, if no sediments are considered. The best agreement between measured and modelled dissipation rate is achieved by considering fine sediments with diameters between 16 μ and 32 μ (see section 2.1.1.2.3). However, the vertical structure of the measurements is not reproduced by any of the simulations, in which the dissipation rate is generally significantly decreasing with distance from the bed. The measurements show a background dissipation rate of above 10^{-8} $W\,kg^{-1}$, whereas the minimum in the simulations is determined by the cut-off value prescribed in the model, which was here 10^{-11} $W\,kg^{-1}$. It seems that the quality of the measurements for the tidal forcing as well as for the dissipation rate are not sufficient for evaluating the predictability of the numerical model. For further results and a discussion of the potential reasons for this discrepancy see *Burchard et al.* [1999].

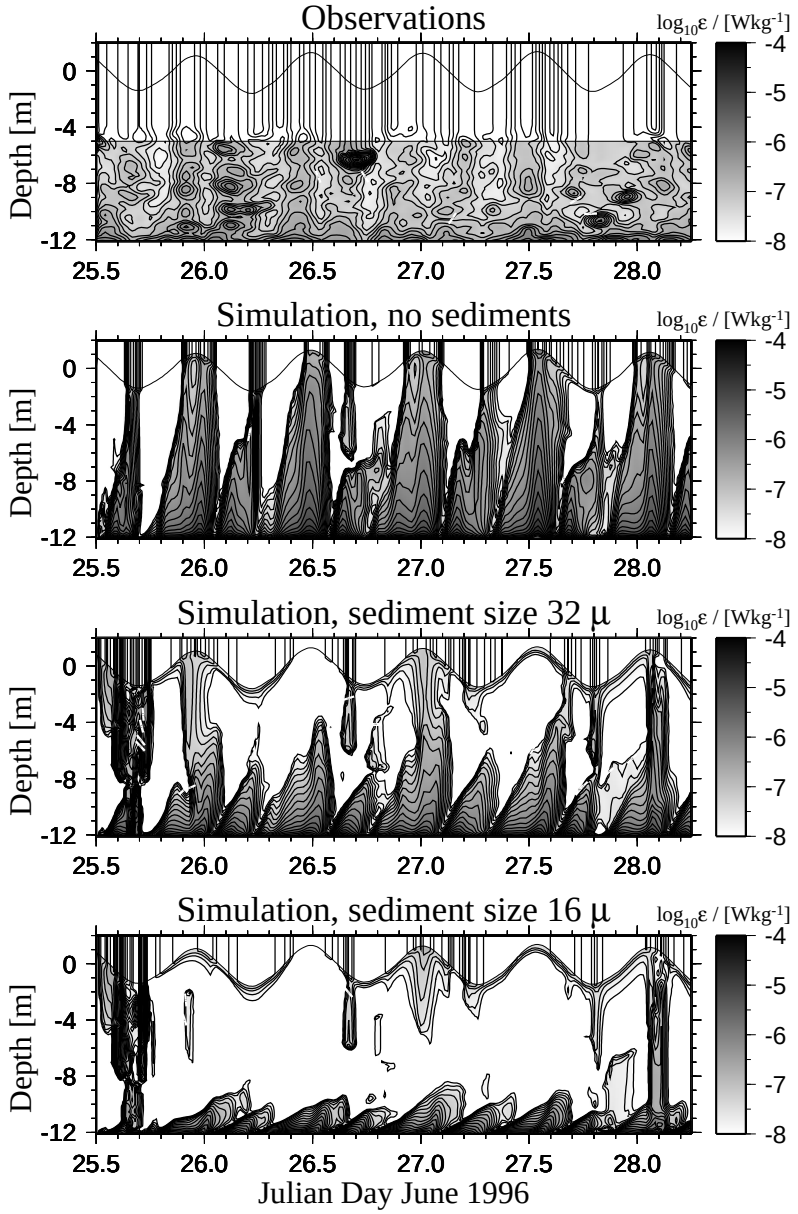


Figure 6.25: Measured and modelled dissipation rate in W kg^{-1} in the Eastern Scheldt. The model results are obtained with clear water, sediment of 32 μ diameter and sediment of 16 μ diameter.

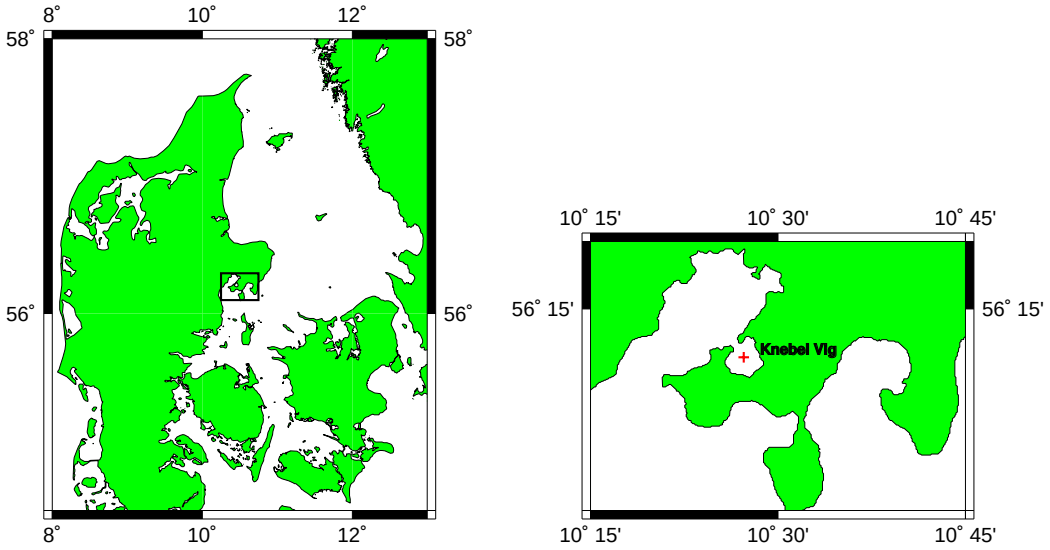


Figure 6.26: Contours of Denmark (left) with a blow-up of the Knebel Vig region (right).

6.3.2 Knebel Vig

Knebel Vig is a small embayment on the eastern coast of Jutland in Denmark at 56° 13'N, 10° 27'E. Its shape can roughly be described as quadratic with a side length of 2.5 km (see figure 6.26). The maximum depth is 16 m near the small inlet towards north-west with a width of about 500 m. Due to the situation in the Kattegat, which connects the Baltic to the North Sea, a strong halocline with a salinity difference of 5 to 10 psu is existing most of the time. The dissipation rate measurements have been carried out from September 7-20, 1997 at different positions in Knebel Vig during a cooling period with strong westerly winds. Each half hour, 7-10 profiles have been measured within 10 min with both of the profilers.

A few days before the campaign began, there was a dramatic change from easterly (outflow of low saline water) to westerly (inflow of high saline water) winds. This was accompanied by a strong decrease of air temperature. Due to the change in wind direction the flow in the Kattegat changed direction from outflow (low salinity) to inflow (high salinity) with the consequence of a strong two-layer flow in Kattegat.

The consequences for Knebel Vig can be seen in figure 6.27. The halocline is rising up to a maximum of 3.5 m below surface on day 11.7. A change in wind direction between day 12.2 and 12.5 together with a wind speed maximum of more than 10 $\mathrm{m\,s^{-1}}$ obviously causes an increase of near surface inflow into Knebel Vig with the consequence of a shear maximum in the entrance channel at day 12.6 (see *Burchard et al.* [1999]). It is probably this strong shear in the entrance channel rather than a local process at the central station in Knebel Vig which causes the erosion of the halocline at day 12.6 which can be clearly seen in figure 6.27.

The simulation results shown here cover a period of 2.5 days when the measurements were taken at a central position in Knebel Vig. The model results for temperature and salinity are not shown here, because due to the relaxation of the measured profiles with a relaxation time of $\tau_T = \tau_S = 30$

min they are very similar to the measurements, but with less variability.
The similarities between measurements and model results are the generally higher dissipation rate above the halocline although the model seems to overestimate the measured values by about half an order of magnitude (see figure 6.28). Some details of the temporal evolution of the measured dissipation rate are well reproduced by the model results for the upper layer, such as the maximum at day 12.6 and the minimum at day 12.8. In the model results for the dissipation rate, the position of the halocline can be seen much clearer than in the measurements. However, both, measurements and model results show local dissipation rate maxima at the halocline. This is because the halocline acts similar to a fixed boundary like the bottom or the surface. In the bottom layer, near bottom vertical gradients of dissipation rate are much lower in the measurements than in the model results. This may be due to the same problems already discussed for the Eastern Scheldt simulation. The simulation results without any internal wave parameterization (not shown here) give dissipation rates which directly below the halocline tend to the minimum values prescribed by the model (we generally use $\varepsilon_{\min} = 10^{-11}\ \mathrm{W\,kg^{-1}}$) (see figure 6.28). Between the *Luyten et al.* [1996] and the *Mellor* [1989] parameterization, only small differences can be seen if an identical minimum value for the turbulent kinetic energy (here $k_{\min} = 3 \cdot 10^{-6}\ \mathrm{J\,kg^{-1}}$) has been chosen. This value appears therefore as a calibration parameter and may vary for different simulations. The *Kantha and Clayson* [1994] approach for effects of stratification on turbulence predicts a much smoother transition from the upper to the lower layer. However, the differences between dissipation measurements and simulation results are so different for all internal wave models used here that it cannot be evaluated here which of the parameterizations is the most reliable. For further details see *Burchard et al.* [1999].

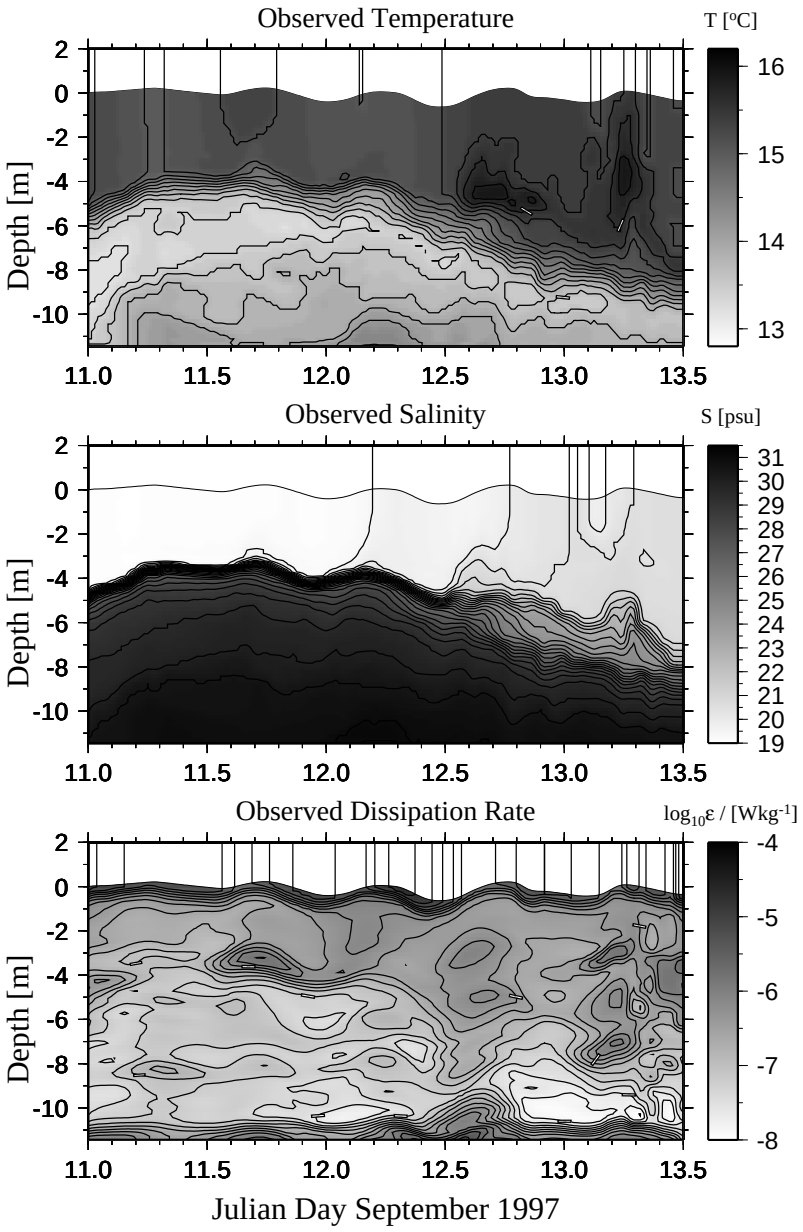


Figure 6.27: Measurements of temperature, salinity and dissipation rate at the central station in Knebel Vig.

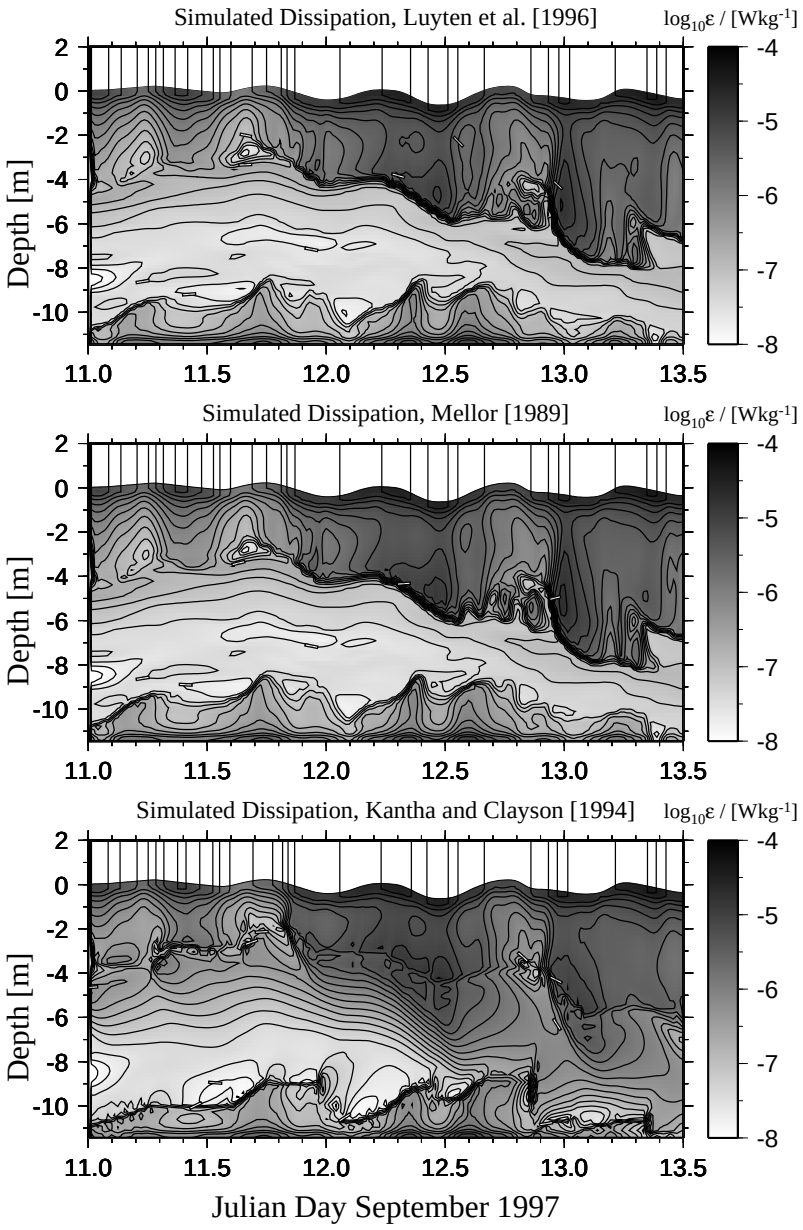


Figure 6.28: Simulated dissipation rate in $\mathrm{W\,kg^{-1}}$ for different internal wave parameterizations in Knebel Vig.

Bibliography

[1] D'Alessio, S.J.D., K. Abdella, and N.A. McFarlane, A new second-order turbulence closure scheme for modeling the ocanic mixed layer, *J. Phys. Oceanogr., 28*, 1624-1641, 1998.

[2] Baumert, H. and G. Radach, Hysteresis of turbulent kinetic energy in nonrotational tidal flows: A model study, *J. Geophys. Res., 97*, 3669-3677, 1992.

[3] Baumert, H., G. Bruckner, E. Kleine, R. Kluge, R., and W. Müller, Abschlußbericht zur G4-Verteidigung, Akademie der Wissenschaften der DDR, Karl-Weierstraß-Institut für Mathematik, Berlin, 1989a.

[4] Baumert, H., G. Bruckner, E. Kleine, R. Kluge, R., and W. Müller, Numerical simulation of estuarine hydrodynamics, *Syst. Anal. Model. Simul., 6*, 503-506, 1989b.

[5] Beckers, J.-M., La mediterranée occidentale: de la modélisation mathématique à la simulation numérique, Ph.D. thesis, Université de Liège, Collection des publications de la Faculté des Sciences Appliquées No. 136, Liège, Belgium, 1995.

[6] Blackadar, A.K., The vertical distribution of wind and turbulent exchange in neutral atmosphere, *J. Geophys. Res., 67*, 3095-3102, 1962.

[7] Blanke, B., and P. Delecluse, Variability of the tropical Atlantic Ocean simulated by a general circulation model with two different mixed-layer physics, *J. Phys. Oceanogr., 23*, 1363-1388, 1993.

[8] Bougeault, P. and J.C. André, On the stability of the third order turbulence closure for the modeling of the stratocumulus-topped boundary layer, *J. Atmos. Sci., 43*, 1574-1581, 1986.

[9] Brockmann, U.H., K. Eberlein, K. Huber, H.-J. Neubert, G. Radach and K. Schulze, JONSDAP'76: FLEX/INOUT Atlas, Vol. 1, Copenhagen: Conseil International pour l'Exploration de la Mer, 1984.

[10] Burchard, H.: Turbulenzmodellierung mit Anwendungen auf thermische Deckschichten im Meer und Strömungen in Wattengebieten, Ph.D. thesis, report GKSS 95/E/30, GKSS Research Centre Geesthacht, Geesthacht, 1995.

[11] Burchard, H.: The 3D hydrostatic equations in a generalized vertical co-ordinate system. Theory and test cases , *Rep. ICCH R22*, 57 pp., ICCH, Hørsholm, Denmark, 1996.

[12] Burchard, H.: Recalculation of surface slopes as forcing for numerical water column models of tidal flow, *Appl. Math. Mod.*, accepted for publication, 1999.

[13] Burchard, H., and H. Baumert, On the performance of a mixed-layer model based on the k-ε turbulence closure, *J. Geophys. Res., 100*, 8523-8540, 1995.

[14] Burchard, H., O. Petersen, and T.P. Rippeth, Comparing the performance of the Mellor-Yamada and the k-ε two-equation turbulence models, *J. Geophys. Res., 103*, 10,543-10,554, 1998.

[15] Burchard, H., and O. Petersen, Models of turbulence in the marine environment – A comparative study of two-equation turbulence models, *J. Marine Systems*, in print, 1999.

[16] Burchard, H., A. Stips, W. Eifler, K. Bolding, and M.R. Villarreal, Numerical simulation of dissipation measurements in non-stratified and strongly stratified estuaries, Proceedings of the 9^{th} Conference of Physics of Estuaries and Coastal Seas, Matsuyama, Japan, September 24-26 1998, accepted for publication, 1999.

[17] Castellari, S., H. Burchard, W. Eifler, and N. Pinardi, Modelling of deep and shallow convection in the Mediterranean Sea, in preparation, 1999.

[18] Craig, P.D., and M.L. Banner, Modeling wave-enhanced turbulence in the ocean surface layer, *J. Phys. Oceanogr., 24,* 2546-2559, 1994.

[19] Davies, A.M., On the importance of time varying eddy viscosity in generating higher tidal harmonics, *J. Geophys. Res., 95*, 20287-20312, 1990.

[20] Deardorff, J.W., Theoretical expressions for the counter-gradient vertical heat flux, *J. Geophys. Res., 77*, 5900-5904, 1972.

[21] Deleersnijder, E. and P. Luyten, On the practical advantages of the quasi-equilibrium version of the Mellor and Yamada level 2.5 turbulence closure applied to marine modelling, *Appl. Math. Modelling, 18*, 281-287, 1994.

[22] Demirov, E., W. Eifler, M. Ouberdous, and N. Hibma, ISPRAMIX - a three-dimensional free surface model for coastal ocean simulations and satellite data assimilation on parallel computers, Technical report EUR 18129EN, European Commission, 76pp., 1998.

[23] Denman, K.L., A time-dependent model of the upper ocean, *J. Phys. Oceanogr., 3*, 173-184, 1973.

[24] Eifler, W. and W. Schrimpf, ISPRAMIX, a hydrodynamic program for computing regional sea circulation patterns and transfer processes, CEC Report EUR 14856 EN, 1992.

[25] Frey, H., A three-dimensional, baroclinic shelf sea circulation model - 1. The turbulence closure scheme and the one-dimensional test model, *Cont. Shelf Res., 11*, 365-395, 1991.

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

[27] Gaspar, P., Y. Gregoris and J. Lefevre, A simple eddy kinetic energy model for simulations of the oceanic vertical mixing: Tests at station Papa and long-term upper ocean study site, *J. Geophys. Res., 95*, 16179-16193, 1990.

[28] Geernaert, G. L., Bulk parameterizations for the wind stress and heat fluxes, In: *Surface waves and fluxes, Volume I, Current theory*, edited by G.L. Geernaert and W.J. Plant, Kluwer Academic Publishers, Dordrecht, 91-172, 1990.

[29] Gill, A.E.Atmosphere–ocean dynamics, Academic Press, 662 pp., 1982.

[30] Goudsmit, G.-H., H. Burchard, F. Peeters, P. Reichert, and A. Wüest, Application of k-ε turbulence models to lakes - the role of internal seiches, submitted to *Water Resources Research*, 1999.

[31] Gregg, M.C., Diapycnical mixing in the thermocline: A review, *J. Geophys. Res., 92*, 5249-5286, 1987.

[32] Herklotz, K., Implementierung eines k-ε-Modells in ein hydrodynamisches Modell, Diploma thesis, Institut für Meereskunde, Universität Hamburg, Germany, 113 pp., 1999.

[33] Jerlov, N.G., Optical oceanography, Elsevier, 194 pp., 1968.

[34] Jobson, H.E., and W.W. Sayre, Vertical mass transfer in open channel flow, *J. Hydraul. Div. Am. Soc. Civ. Eng., 96*(HY3), 703-724, 1970.

[35] Kagan, B.A., Ocean-Atmosphere interaction and climate modelling, Cambridge University Press, Cambridge, 1995.

[36] Kantha, L.H., and C.A. Clayson, An improved mixed layer model for geophysical applications, *J. Geophys. Res., 99,* 25235-25266, 1994.

[37] Kolmogorov, A.N., Equations of turbulent motion on an incompressible fluid, *Izv. Akad. Nauk. SSSR, Seria fizicheska Vi.*, 56-58, 1942. (English translation: Imperial College, *Mech. Eng. Dept. Rept. ON/6*, 1968).

[38] Kondo, J., Air-sea bulk transfer coefficients in diabatic conditions. *Bound. Layer Meteor., 9*, 91-112, 1975.

[39] Large, W.G., J.C. McWilliams, and S.C. Doney, Oceanic vertical mixing : a review and a model with nonlocal boundary layer parameterization, *Reviews of Geophysics, 32*, 363-403, 1994.

[40] Launder, B.E., G.J. Reece, and W. Rodi, Progress in the development of a Reynolds-stress turbulence closure, *J. Fluid Mech., 68*, 537-566, 1975.

[41] Leonard, B.P., M.K. MacVean, and A.P. Lock, The flux integral method for multidimensional convection and diffusion, *Appl. Math. Modelling, 19*, 333-342, 1995.

[42] van Leussen, W., Estuarine macroflocs and their role in fine-grained sediment transport, Ph.D. thesis, Universiteit Utrecht, 1994.

[43] Levitus, S., World Ocean Atlas, an atlas of objectively analysed fields of major ocean parameters, 1994. (Data downloaded from `http://ingrid.ldgo.columbia.edu/SOURCES/.LEVITUS94/`).

[44] List, R.J., The Smithsonian Meteorological Tables, Smithsonian Institution, Washington, DC, 527 pp., 1958.

[45] Liungman, O.,Tidally forced internal wave mixing in a k-ε model framework applied to fjord basins, submitted 1998.

[46] Luyten, P.J., E. Deleersnijder, J. Ozer, and K.G. Ruddick, Presentation of a family of turbulence closure models for stratified shallow water flows and preliminary application to the Rhine outflow region, *Cont. Shelf Res., 16*, 101-130, 1996a.

[47] Luyten, P.J., J.H. Simpson, and T.P. Rippeth, Comparison of turbulence models for homogeneous and stratified flows with turbulence measurements in the Irish Sea, paper presented at MAST Workshop on Turbulence Modelling, Bergen, Norway, Aug. 8-10, 1996, 1996b.

[48] Martin, P.J., Simulation of the mixed layer at OWS November and Papa with several models, *J. Geophys. Res. 90*, 903-916, 1985.

[49] Martin, P.J., Testing and comparison of several mixed-layer models, Naval Oceanographic Research and Development Agency (NORDA) Report 143, 27 pp., Naval Research Laboratory, Stennis Space Center, Mississippi, 1986.

[50] Mellor, G.L., Retrospect on oceanic boundary layer modelling and second moment closure, in *Parameterization of Small-Scale Processes; Proceedings of the Aha Hulikoa Hawaiian Winter Workshop,* edited by P. Müller and D. Henderson, pp. 251-271, Univ. of Hawaii at Manoa, Honolulu, 1989.

[51] Mellor, G.L., and T. Yamada, A hierarchy of turbulence closure models for planetary boundary layers, *J. Atmos. Sci., 31*, 1791-1806, 1974.

[52] Mellor, G.L., and T. Yamada, Development of a turbulence closure model for geophysical fluid problems, *Rev. Geophys., 20*, 851-875, 1982.

[53] Mohammadi, B., and O. Pironneau, Analysis of the k-ε turbulence model, Paris e.a., Masson, 1994.

[54] Munk, W.H., and E.R. Anderson, Notes on the theory of the thermocline,, *J. Marine Res., 3*, 276-295, 1948.

[55] Nakagawa, H., I. Nezu, and H. Ueda, Turbulence in open channel flow over smooth and rough beds, *Proc. Jpn. Soc. Civ. Eng., 241*, 155-168, 1975.

[56] Nihoul, J.C., and S. Djenidi, Perspectives in three-dimensional modelling of the marine system, In:*Three-dimensional models of marine and estuarine dynamics*, edited by J.C. Nihoul and B.M. Jamart, Elsevier Oceanography Series No. 45 Amsterdam, 1-33, 1987.

[57] Patankar, S.V., Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York, 1980.

[58] Paulson, C.A., and J.J. Simpson, Irradiance measurements in the upper ocean, *J. Phys. Oceanogr., 7*, 952-956, 1977.

[59] Perrels, P.A.J., and M. Karelse, A two-dimensional, laterally averaged model for salt intrusion in estuaries, Waterloopkundig Laboratorium, Delft Hydraulics Laboratory, Publication No. 262, 1982.

[60] Petersen, O., Numerical models of wind induced mixing in stably stratified flows, *submitted to Cont. Shelf Res.*, 1996.

[61] Pohlmann, T., Estimating the influence of advection during FLEX'76 by means of a three-dimensional shelf sea circulation model, *Deutsch. Hydrogr. Z., 49*, 215-226, 1997.

[62] Prandke, H., and A. Stips, Investigation of microstructure and turbulence in marine and limnic waters using the MST profiler , CEC-JRC, Space Applications Institute, Technical Note No.I.96.87, 90 pp., 1996.

[63] Prandtl, L., Über ein neues Formelsystem für die ausgebildete Turbulenz, *Nachr. Akad. Wiss., Göttingen, Math.-Phys. Klasse*, 6, 1945.

[64] Price, J.F., On the scaling of stress-driven entrainment experiments, *J. Fluid Mech., 90*, 509-529, 1979.

[65] Reed, R.K., On estimating insolation over the ocean, *J. Phys. Oceanogr., 7*, 482-485, 1977.

[66] Robert, J.L. and Y. Ouellet, A three-dimensional finite element model for the study of steady and non-steady natural flows, In: *Three-dimensional models of marine and estuarine dynamics*, edited by J.C. Nihoul and B.M. Jamart, Elsevier Oceanography Series No. 45 Amsterdam, 359-372, 1987.

[67] Rodi, W., Turbulence models and their application in hydraulics, report, Int. Assoc. for Hydraul. Res., Delft, Netherlands, 1980.

[68] Rodi, W., Examples of calculation methods for flow and mixing in stratified flows, *J. Geophys. Res., 92*, 5305-5328, 1987.

[69] Rosati, A., and K. Miyakoda, A general circulation model for upper ocean simulation, *J. Phys. Ocean., 18*, 1601-1626, 1988.

[70] Sherwood, C.R., Numerical model of frazil-ice and suspended sediment concentrations, and formation of sediment-laden ice in the Kara Sea, *J. Geophys. Res.,* submitted, 1998.

[71] Simpson, J.H., W.R. Crawford, T.P. Rippeth, A.R. Campbell, and J.V.S. Cheok, The vertical structure of turbulent dissipation in shelf seas, *J. Phys. Oceanogr., 26,* 1579-1590, 1996.

[72] Smith, J.D., and S.R. McLean, Spatially averaged flow over a wavy surface, *J. Geophys. Res., 82*, 1735-1746, 1977.

[73] Soetje, K.C., and K. Huber, A compilation of data on the thermal stratification at the central station in the northern North Sea during FLEX'76, *"Meteor"-Forsch.-Ergebnisse, Reihe A 22,* 69-77, 1980.

[74] Stelling, G.S., Compact differencing for stratified free surface flow, *Advances in Hydro-Science and -Engineering, 2*, March 22-26, 1995, Bejing, China, 378-386, 1995.

[75] Telbany, M.M.M.E., and A.J. Reynolds, The structure of turbulent plane Couette flow, *J. Fluids Eng., 104*, 367-372, 1982.

[76] Therry, G. and P. Lacarrère Improving the eddy kinetic energy model for planetary boundary layer description, *Bound. Layer Meteor., 17*, 63-88, 1983.

[77] Ueda, H., R. Möller, S. Komori, and T. Mizushina, Eddy diffusivity near the free surface of open channel flow, *Int. J. Heat Mass Transfer, 20*, 1127-1136, 1977.

[78] Verduin, J.J., and J.O. Backhaus, Dynamics of plant-flow interactions for the sea-grass *Amphibolis Antarctica*: Field observations and model simulations, *Estuarine Coastal and Shelf Science*, accepted, 1998.

[79] Visser, A.W., Using random walk models to simulate the vertical distribution of particles in a turbulent water column, *Mar. Ecol. Prog. Ser., 158*, 275-281, 1997.

[80] Wessel, P, and W.H.F. Smith, New, improved version of Generic Mapping Tools released, *EOS Trans. Amer. Geophys. U., 79*, 579 pp., 1998.

[81] Wyrtki, K., The average annual heat balance of the North Pacific Ocean and its relation to ocean circulation, *J. Geophys. Res., 70*, 4547-4559, 1965.

[82] Xing, J. and A.M. Davies, Application of three dimensional turbulence energy models to the determination of tidal mixing and currents in a shallow sea, *Prog. Oceanog., 35*, 153-205, 1995.

[83] Zalesak, S.T., Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comp. Phys., 31*, 335-362, 1979.

[84] Zanke, U., Berechnung der Sinkgeschwindigkeiten von Sedimenten, *Mitteilungen des Franzius-Institutes, 46*, 231-245, 1977.

[85] Zilitinkevich, S.S., and D.V. Mironov, Theoretical model of the thermocline in a freshwater basin, *J. Phys. Oceanogr., 22*, 988-996, 1992.

Chapter 7

Appendix - The `namelist.inp` file

The `namelist.inp` file for the FLEX-experiment has been included verbatim in this appendix.

```
!-----------------------------------------------------------------------
!         JRC/SAI/ME Generel Ocean Turbulence Model (GOTM)              !
!-----------------------------------------------------------------------
!
! ! namelist.inp: Model configuration file
!
! Nmx        : Number of vertical layers
! dt         : Time step [s]
! startdate  : Startdate [days]
! stopdate   : Stopdate [days]
! depth0     : Mean water depth [m]
! h0b        : Length of bottom  roughness elements. Note: z0b=0.03*h0b+... [m]
! h0s        : Length of surface roughness elements. Note: z0s=0.03*h0s     [m]
! MaxItz0b   : Maximum number of iterations for z0b as function of u_taub
! Phi        : Latitude [deg], positive for northern hemisphere
! UNPress    : Pressure effect in UNESCO equation of state
!     : .true.,  Pressue effect included
!     : .false., Pressue effect excluded
! BuoyMethod : Method for computing buoyancy
!     : 0, Buoyancy from dynamic equation for buoyancy
!     : 1, Buoyancy from UNESCO equation of state
! TKEMeth    : Method for computing TKE (Turbulent Kinetic Energy)
!     : 1, Algebraic equation
!     : 2, Dynamic equation for k-epsilon model
!     : 3, Dynamic equation for Mellor-Yamada model
! LengthMethod : Method for computing turbulent length scale L
!     : 1, Parabolic shape
!     : 2, Triangle shape
!     : 3, Xing and Davies [1995]
!     : 4, Robert and Ouellet [1987]
!     : 5, Blackadar (two boundaries) [1962]
```

```
!     : 6, Bougeault and Andre [1986]
!     : 7, Eifler and Schrimpf (ISPRAMIX) [1992]
!     : 8, Dynamic dissipation rate equation
!     : 9, Dynamic Mellor-Yamada kL equation
! lengthlim : Limitation of L for stable stratification, Galperin et al. [1988]
!     : .true.,  Length limitation on
!     : .false., Length limitation off
! k_min : Minimum value of TKE [J/kg]
! epsmin: Minimum value of Dissipation rate [W/kg]
! L_min : Minimum value of turbulent length scale [m]
! Stab: Method for computing stability functions (SF)
!     : 0, constant SF (constant Prandtl number)
!     : 1, not used
!     : 2, quasi-equilibrium, Galperin et al. [1988]
!     : 3, quasi-equilibrium, Kantha and Clayson [1994]
!     : 4, Eifler and Schrimpf (ISPRAMIX) [1992]
!     : 5, Munk and Anderson [1948]
! cmucst   : Value of SF cmue1 if Stab=0, for LengthMeth=6 set cmucst=0.1
! Prandtl0 : Turbulent Prandtl number, only relevant if Stab=(0,1,4)
! galp     : Coefficient for length limitation, should be 0.53
! avmolu   : Molecular (or background) viscosity for momentum    [m^2/s]
! avmolT   : Molecular (or background) diffusity for temperature [m^2/s]
! avmolS   : Molecular (or background) diffusity for salinity    [m^2/s]
! ddu      : 0: no zooming towards surface, >0: zooming towards surface
! ddl      : 0: no zooming towards bottom,  >0: zooming towards bottom
!
 &param
      Nmx        = 50
      dt         = 100.
      startdate  = 96.25
      stopdate   = 158.0
      depth0     = 145.
      h0b        = 0.05
      h0s        = 0.5
      MaxItz0b   = 1
      Phi        = 58.916666
      UNPress    = .false.
      BuoyMethod = 1
      TKEMeth    = 3
      LengthMeth = 9
      lengthlim  = .true.
      k_min      = 3e-6
      epsmin     = 5e-10
      L_min      = 0.01
      Stab       = 2
      Prandtl0   = 0.714
      cmucst     = 0.1
```

```
      galp       = 0.53
      avmolu     = 1.3e-6
      avmolT     = 1.4e-7
      avmolS     = 1.1e-9
      ddu        = 0.00
      ddl        = 0.00
 &end
! pi    : Circle number 3.141592654
! rho_0 : Mean density of water [kg/m^3]
! cp    : Specific heat capacity of sea water [J/kg/K]
! g     : Gravitational acceleration [m/s^2]
! kappa : von Karman constant
! cm0   : Stability function for momentum for unstratified flow,
!         Galperin et al. [1988], important for relation between k, L and eps.
 &consts
      pi         = 3.141592654
      rho_0      = 1027.
      cp         = 3985.
      g          = 9.81
      kappa      = 0.4
      cm0        = 0.5562
 &end
!
! ce1     : Empirical coefficient in dissipation equation
! ce2     : Empirical coefficient in dissipation equation
! ce3minus: Empirical coefficient in dissipation equation (stable strat.)
! ce3plus : Empirical coefficient in dissipation equation (unstable strat.)
! sig_k   : Schmidt number for TKE eddy diffusivity
! fluxcond: Switch for type of boundary condtions for k and eps
!         : .true.,  flux boundary conditions
!         : .false., Dirichlet boundary conditions
 &keps
      ce1        = 1.44
      ce2        = 1.92
      ce3minus   = -0.4
      ce3plus    = 1.0
      sig_k      = 1.
      fluxcond   = .true.
 &end
!
! sl : Parameter for calculating eddy diffusivities of k and kL (sl=cl/sqrt(2))
! e1 : Coefficient in Mellor-Yamada kL equation
! e2 : Coefficient in Mellor-Yamada kL equation
! MYLength: Prescribed barotropic lengthscale in Mellor-Yamada kL equation
!     : 1, Parabolic shape
!     : 2, Triangle shape
!     : 3, Linear from surface, infinit depth
```

```
!
 &my
      sl          = 0.2
      e1          = 1.8
      e2          = 1.33
      MYLength    = 3
 &end
!
! a1,a2,b1,b2,c1 : Coefficients in Galperin quasi-equilibrium SF for Stab=2
! qesmooth: .true. smoothing of SF for unstable stratification for Stab=2
! qeghmax : Maximum value of stratification parameter gh for Stab=2
! qeghmin : Minimum value of stratification parameter gh for Stab=2
! qeghcrit: Critical value of gh to start smoothing for Stab=2
!
 &stabfunc
      a1          = 0.92
      a2          = 0.74
      b1          = 16.6
      b2          = 10.1
      c1          = 0.08
      qesmooth    = .true.
      qeghmax     = 0.0233
      qeghmin     = -0.28
      qeghcrit    = 0.02
 &end
!
! IWModel : Method for modelling internal wave effects on mixing
!     : 0, No IW Model (except from possible limitations of TKE and epsilon)
!     : 1, Mellor ([989]
!     : 2, Kantha and Clayson [1994]
! alfa    : Coefficient for Mellor [1989] IW model
! klimiw  : Critical value of TKE in Kantha and Clayson [1994] IW model
! iter_ini: Number of initial iterations before applying Kantha-Clayson model
!
 &iw
      IWModel     = 0
      alfa        = 0.7
      klimiw      = 1e-6
      iter_ini    = 10
 &end
!
! format: format of the output file
!     : 1, ASCII
!     : 2, netcdf
! OutFN: output file name
! ISAVE: number of time step between output instants
!
```

```
 &output
      format     = 1
      OutFN      = 'Flex.out'
      ISAVE      = 36
 &end
!
! ObsType : Type of diagnostic data
!          : 0, not coded yet
!         : 1, not coded yet
!         : 2, data read from file
! ObsFile : Input file of observations
! RelaxTau: Relaxation Time in seconds
!         : RelaxTau<0     --> equation not calculated
!         : RelaxTau>1.e10 --> no relaxation
!
 &sobs
      SObsType   = 2
      SObsFile   = 'SObs.dat'
      SRelaxTau  = 172800.
 &end
 &tobs
      TObsType   = 2
      TObsFile   = 'TObs.dat'
      TRelaxTau  = 1.e15
 &end
! EObsConst: Value for epsilon if no data available, [W/kg]
 &eobs
      EObsType   = 0
      EObsConst  = 1.e-12
      EObsFile   = 'EObs.dat'
 &end
!
! Only used if dynamic equation for buoyancy calculated, BuoyMethod=0
! BObsSurf: Prescribed buoyancy at the surface [m/s^2]
! BObsNN  : Prescribed constant Brunt-Vaisalla frequency [1/s]
!
 &bobs
      BObsSurf   = 0.0
      BObsNN     = 1.e-4
 &end
!
! MeteoType : Type of meteorological forcing
!    :0, constant values, constants are defined below
!    :1, not coded yet
!    :2, data read from file
! WxConst:   x wind stress [N/m^2]
! WyConst:   y wind stress [N/m^2]
```

```
! QinConst:  Solar radiation normal to the sea surface [W/m^2]
! QoutConst: Outgoing heat fluxes [W/m^2]
! SSTConst : Observed sea surface temperature
! MeteoFile: Input file for meteorological forcing
!
 &meteo
      MeteoType  = 2
      WxConst    = 0.5
      WyConst    = 0.5
      QinConst   = 0.0
      QoutConst  = 0.0
      SSTConst   = 15.0
      MeteoFile  = 'Meteo.dat'
 &end
!
! External pressure forcing
! PressType:
!    0: prescribed constant values
!    1: prescribed  amplitude, phase and period of 2 harmonics
!    2: data read from file
! PressMethod : Method of calculating external pressure in model
!    0: Surface elevation gradients directly given
!    1: Recalculating pressure gradient from current meter measurements
!    2: Recalculating pressure gradient from given vertical mean velocities
! PressFile: file name if PressType=2
! PressConstU: prescribed pressure gradient
! PressConstU: prescribed pressure gradient
! PeriodM    : Period of harmonic 1 (M)
! PeriodS    : Period of harmonic 2 (S)
! AmpMu      : u Amplitude of harmonic M
! AmpMv      : v Amplitude of harmonic M
! AmpSu      : u Amplitude of harmonic S
! AmpSv      : v Amplitude of harmonic S
! PhaseMu    : u Phase of harmonic M
! PhaseMv    : v Phase of harmonic M
! PhaseSu    : u Phase of harmonic S
! PhaseSv    : v Phase of harmonic S
!
 &press
      PressType   = 2
      PressMethod = 0
      PressFile   = 'Pressure.dat'
      PressConstU = 0.0
      PressConstV = 0.0
      PressHeight = 0.0
      PeriodM     = 44714.0
      PeriodS     = 43200.0
```

```
      AmpMu       = 0.0
      PhaseMu     = 0.0
      AmpSu       = 0.0
      PhaseSu     = 0.0
      AmpMv       = 0.0
      PhaseMv     = 0.0
      AmpSv       = 0.0
      PhaseSv     = 0.0
 &end
!
! Sea surface elevation
! ZetaType:
!    0: prescribed constant values
!    1: prescribed  amplitude, phase and period of 2 harmonics
!    2: data read from file
! ZetaFile: file name if ZetaType=2
! ZetaConst: prescribed sea surface eleveation
! PeriodM  : Period of harmonic 1 (M)
! PeriodS  : Period of harmonic 2 (S)
! AmpM     : Amplitude of harmonic M
! AmpS     : Amplitude of harmonic S
! PhaseM   : Phase of harmonic M
! PhaseS   : Phase of harmonic S
 &zetaspec
      ZetaType   = 0
      ZetaFile   = 'Zeta.dat'
      ZetaConst  = 0.00000
      AmpM       = 1.00000
      PeriodM    = 44714.0
      PhaseM     = 0.00000
      AmpS       = 0.50000
      PeriodS    = 43200.0
      PhaseS     = 0.00000
 &end
! Light extinction coefficients for absorbtion of solar radiation
! ExtinctType: Specifying water type or reading data from file
!      :1, Jerlov type I
!      :2, Jerlov type 1 (upper 50 m)
!      :3, Jerlov type IA
!      :4, Jerlov type IB
!      :5, Jerlov type II
!      :6, Jerlov type III
!      :7, data read from file
! ExtinctFile: file name if ExtinctType=7
!
 &extinct
      ExtinctType = 7
```

```
      ExtinctFile = 'Extinction.dat'
 &end
! Sediment equation
! SediCalc:
!      : .true.,  calculation of sediment
!      : .false., no calculation of sediment
! SediDens:
!      : .true.,  density effect of sediment included
!      : .false., density effect of sediment excluded
! rho_sed : Density of sediment [kg/m^3]
! size    : Diameter of sediment particles [m]
! InitConc: Value of initial sediment concentration
! CbObsDt : Relaxation time for bottom value of sediment [s]
! AdvMeth : Numerical scheme for sediment advection
!     :1, first order upstream
!     :2, third order QUICKEST (only for equidistant discetization)
!     :3, flux corrected transport, combination of 1 & 2
 &sedi
      SediCalc  = .false.
      SediDens  = .false.
      rho_sed   = 2650.0
      size      = 62.5e-6
      InitConc  = 0.0
      CbObsDt   = 600.0
      AdvMeth   = 3
 &end
```

List of Figures