1. INTRODUCTION

The TELEMAC-3D code solves such three-dimensional equations as the free surface flow equations (with or without the hydrostatic pressure assumption) and the transport-diffusion equations of intrinsic quantities (temperature, salinity, concentration). Its main results, at each point in the resolution mesh, are the velocity in all three directions and the concentrations of transported quantities. Water depth is the major result as regards the surface mesh. The TELEMAC-3D's prominent applications can be found in free surface flow, in both seas and rivers; the software can take the following processes into account:

- Influence of temperature and/or salinity on density,
- Bottom friction,
- Influence of the Coriolis force,
- Influence of weather elements : air pressure and wind,
- Consideration of the thermal exchanges with the atmosphere,
- Sources and sinks for fluid moment within the flow domain,
- Simple or complex turbulence models (K-Epsilon) taking the effects of the Archimedean force (buoyancy) into account,
- Dry areas in the computational domain : tidal flats,
- Current drift and diffusion of a tracer, with generation or disappearance terms.

The code is applicable to many fields. The main ones are related to the marine environment through the investigations of currents being induced either by tides or density gradients, with or without the influence of such an external force as the wind or the air pressure. It can be applied either to large extent areas (on a sea scale) or to smaller domains (coasts and estuaries) for the impact of sewer effluents, the study of thermal plumes or even sedimentary transport. As regards the continental waters, the study of thermal plumes in rivers, the hydrodynamic behaviour or natural or man-made lakes can be mentioned as well.

1.1. POSITION OF THE TELEMAC-3D CODE WITHIN THE TELEMAC MODELLING SYSTEM

The TELEMAC-3D code is part of a processing chain, namely the TELEMAC system. That package provides with all the required modules for constructing a model and for conducting simulations of hydrodynamic flow, contaminant and sediment transport.

The TELEMAC system comprises the following modules:

- The MATISSE software designed, using the bathymetric and/or topographic data, to generate a mesh consisting of triangular elements,
- The STBTEL software designed to retrieve the file from a mesh generator, possibly interpolate a bathymetry, and generate a geometry file in the SELAFIN format that can be read by the simulation modules as well as the RUBENS software. STBTEL also conducts a number of mesh coherence checks,
- The TELEMAC-2D software designed to perform the hydrodynamic simulation in two horizontal space dimensions. In addition, TELEMAC-2D can simulate the transport of dissolved tracers,
- The TELEMAC-3D software itself, designed to carry out the hydrodynamic simulations of flows in three space dimensions. Besides, TELEMAC-3D can simulate the transport of tracers. The SEDI-3D library contains of the relevant subroutines for the simulation of noncohesive sediment transport. The implementation of the TELEMAC-3D software is the subject matter of this document,
- The SUBIEF-2D software designed to carry out the simulation, in two horizontal space dimensions, of

the transport of suspended sediments and the transport of dissolved substances without any gravity effect. SUBIEF-2D particularly makes it possible to handle the water quality-related issues.
- The SUBIEF-3D software designed to make the three-dimensional simulation of the transport of dissolved substances without any gravity effect. SUBIEF-3D is also provided for handling the water quality-related issues.
- The SISYPHE software designed to carry out the simulation the transport of sediment through bed load traction and suspension.
- The ARTEMIS software designed to simulate the changes in the features of wave agitation either in a coastal water body or a harbour,
- The TOMAWAC software designed to simulate, through a spectral method, the sea state in permanent or transitory conditions,
- The ESTEL-2D software designed to simulate the underground flows in two vertical space dimensions,
- The ESTEL-3D software designed to simulate the underground flows in three dimensions,
- The POSTEL-3D software designed to prepare the 2D cross sections in the 3D result file, for a processing by the RUBENS graphics software,
- The RUBENS software designed to graphically process the results of the various simulation modes,
- The SPARTACUS-2D software designed to simulate the two-dimensional laminar and turbulent flows through the SPH method.

As a complement to the TELEMAC chain, the FUDAA-PREPRO software (as developed from the FUDAA platform by the CETMEF's Recherche, Informatique et Modélisation Department) covers all the preprocessing tasks involved by the achievement of a digital hydraulic study.

1.2. SOFTWARE ENVIRONMENT

All the simulation modules are written in Fortran-90, with no use of the specific language extensions in a given machine. They can be run on all the PCs (or PC "clusters") under Windows and Linus operating systems as well as on the workstations under the Unix operating system. An implementation run on some vector computers (particularly Cray and Fujitsu) is available as well.

The graphics modules (RUBENS, MATISSE) can be used in a workstation operated under Unix and provided with the X_Window and OSF/Motif libraries, as well as on a microcomputer working under Windows.

For the sake of maintenance of the Windows release of TELEMAC, the release 5.7 is the last release which will be compatible with the Compaq Visual Fortran compiler.

That compiler will be replaced by the Intel Fortran Compiler 9.1 from the next release onwards. The V5P8 release of TELEMAC, which is compatible with the Intel Fortran Compiler 9.1, is already available on request.

1.3. USER PROGRAMMING

When he uses a simulation module from the TELEMAC system, the user may have to program specific functions which are not provided in the code's standard release. In particular, that is made through a number of so-called « user » subroutines the sources of which are supplied within the distribution. The procedure to be carried out in that case comprises the steps of:

- Recovering the standard version of the user subroutine(s) as supplied in the distribution and copying it into the current directory.
- Amending the subroutine(s) according to the model to be constructed.

- Concatenating the whole set of subroutines into a single Fortran file which will be compiled during the TELEMAC-3D launching process.

During that programming stage, the user can gain access to the various variables of the software through the Fortran 90 structures.

All the data structures are gathered within Fortran files, which are known as modules. For TELEMAC-3D, the file name is DECLARATION_TELEMAC3D. To gain access to the TELEMAC-3D data, just insert the command USE DECLARATIONS_TELEMAC3D into the beginning of the subroutine. Adding the command USE BIEF may also be necessary in order to reach the structure in the BIEF library.

Nearly all the arrays which are used by TELEMAC-3D are declared in the form of a structure. The access to the water depth array will then be take place in the form H%R, %R meaning it is the “real number-typed field” component in the structure. In case of a integer-typed component, the %R is replaced by a %I. In order to avoid having to handle too many %R and %I, however, a number of aliases are defined, such as, for instance, the NPOIN3D, NELEM3D et NPTFR2D variables. For further details, the user can refer to the programming guide in TELEMAC [5].

2. THEORETICAL ASPECTS

2.1. NOTATIONS

TELEMAC-3D is a three-dimensional computational code describing the 3D velocity field (u, v, w) and the water depth h (and, from the bottom depth, the free surface S) at each time step. Besides, it solves the transport of several tracers which can be grouped intoi two categories, namely the socalled ”active” tracers (primarily temperature and salinity1), which change the water density and act on flow through gravity), and the so-called “passive” tracers which do not affect the flow and are merely transported.

2.2. EQUATIONS

The reader will refer to the J.M. Hervouet’s book [1] for a detailed statement of the theoretical aspects which TELEMAC-3D is based on.

2.2.1. EQUATIONS WITH THE HYDROSTATIC ASSUMPTION

In its basic release, the code solves the three-dimensional hydrodynamic equations with the following assumptions:

- three-dimensional Navier-Stokes equations with a free surface changing in time,
- negligible variation of density in the conservation of mass equation (incompressible fluid),
- pressure-hydrostatic assumption (that assumption results in that the pressure at a given depth is the sum of the air pressure at the fluid surface plus the weight of the overlying water body),
- Boussinesq approximation for the momentum (the density variations are not taken into account in the gravity term).

Due to these assumptions, the three-dimensional equations being solved are:

$$\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}+\frac{\partial W}{\partial z}=0$$

$$\frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+V\frac{\partial U}{\partial y}+W\frac{\partial U}{\partial z}=-g\frac{\partial Z_s}{\partial x}+\nu\Delta(U)+F_x$$

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + W\frac{\partial V}{\partial z} = -g\frac{\partial Z_s}{\partial y} + \nu\Delta(V) + F_y$$

$$p = p_{atm} + \rho_0 g(Z - z) + \rho_0 g\int_z^{Z_s}\frac{\Delta\rho}{\rho_0}\mathrm{d}z$$

$$\frac{\partial T}{\partial t} + U\frac{\partial T}{\partial x} + V\frac{\partial T}{\partial y} + W\frac{\partial T}{\partial z} = \nu\Delta(T) + Q$$

wherein:

h (m) water depth.
Z_S (m) free surface elevation.
U, V, W (m/s) three-dimensional components of velocity.
T (°C, g/l …) passive or active (acting on density) tracer.
p (X) pressure.
g (m/s^2) gravity acceleration.
ν (m^2/s) velocity and tracer diffusion coefficients.
Z_f (m) bottom depth.
ρ_0 (X) reference density.
$\Delta\rho$ (X) variation of density.
t (s) time.
x, y (m) horizontal space components.
z (m) vertical space component.
F_x, F_y (m/s^2) source terms.
Q (tracer unit) tracer source of sink.
h, U, V, W and T are the unknown quantities, also known as computational variables
F_x and F_y are source terms denoting the wind, the Coriolis force and the bottom friction (or any other process being modelled by similar formulas). Several tracers can be taken into account simultaneously. They can be of two different kinds, either active, i.e. influencing the flow by changing the density, or passive, without any effect on density and then on flow.

The TELEMAC-3D basic algorithm can be split up in three computational steps (three fractional steps). The first step consists in finding out the advected velocity components by only solving the advection terms in the momentum equations. The second step computes, from the advected velocities, the new velocity components taking into account the diffusion terms and the source terms in the momentum equations. These two solutions enable to obtain an intermediate velocity field. The third step is provided for computing the water depth from the vertical integration of the continuity equation and the momentum equations only including the pressure-continuity terms (all the other terms have already been taken into account in the earlier two steps). The resulting twodimensional equations (analogous to the Saint-Venant equations without diffusion, advection and source terms) are written as:

$$\frac{\partial h}{\partial t} + \frac{\partial uh}{\partial x} + \frac{\partial vh}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} = -g\frac{\partial Z_s}{\partial x}$$

$$\frac{\partial v}{\partial t} = -g\frac{\partial Z_s}{\partial y}$$

The u and v in lower case denote the two-dimensional variables of the vertically integrated velocity.

These two-dimensional equations are solved by the libraries in the TELEMAC-2D code and enable to obtained the vertically averaged velocity and the water depth. The water depth makes it possible to recompute the elevations of the various mesh points and then those of the free surface. Lastly, the computation of the U and V velocities is simply achieved through a combination of the equations linking the velocities. Finally, the vertical velocity W is computed from the continuity equation.

2.2.2. NON-HYDROSTATIC NAVIER-STOKES EQUATIONS

The following system (with an equation for W which is similar to those for U and V) sis then to be solved:

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} = 0$$

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} + W\frac{\partial U}{\partial z} = -g\frac{\partial Z_s}{\partial x} + \nu\Delta(U) + F_x$$

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + W\frac{\partial V}{\partial z} = -g\frac{\partial Z_s}{\partial y} + \nu\Delta(V) + F_y$$

$$\frac{\partial W}{\partial t} + U\frac{\partial W}{\partial x} + V\frac{\partial W}{\partial y} + W\frac{\partial W}{\partial z} = -g\frac{\partial Z_s}{\partial z} + \nu\Delta(W) + F_z$$

In order to share a common core as much as possible with the solution of the hydrostatic equations, the pressure is split up into a hydrostatic pressure and a “dynamic” pressure term.

$$p = p_{atm} + \rho_0 g(Z - z) + \rho_0 g\int_z^{Z_s}\frac{\Delta\rho}{\rho_0}\mathrm{d}z + p_d$$

The TELEMAC-3D algorithm solves a hydrostatic step which is the same as in the previous paragraph, the only differences lying in the continuity step (“projection” step in which the dynamic pressure gradient changes the velocity field in order to provide the required zero divergence of velocity) and the computation of the free surface.

2.2.3. THE LAW OF STATE

Two laws of state can be used by default through TELEMAC-3D.

In most of the simulations, salinity and temperature make it possible to compute the variations of density. The first law expresses the variation of density from these only two parameters. The second law is more general and enables to construct all the variations of density with the active tracers being taken into account in the computation. The first law is written as:

$$\rho = \rho_{ref}\left[1 - \left(T(T - T_{ref})^2 - 750S\right)10^{-6}\right]$$

With T_{ref} as a reference temperature of 4°C and ρ_{ref} as a reference density at that temperature when the salinity is zero, then ρ_{ref} 999,972kg /m3. That law remains valid for 0°C < T < 40°C and 0g / l < S < 42g / l . The second law is written as:

$$\rho = \rho_{ref}\left[1 - \sum_i \beta_i\left(T_i - T_i^0\right)_i\right]$$

ρ_{ref} , the reference density can be modified by the user together with the volumetric expansion coefficients β_i related to the tracers T_i .

2.2.4. K-EPSILON MODEL

The turbulent viscosity can be given by the user, as determined either from a mixing length model or from a k-ε model the equations of which are:

$$\frac{\partial k}{\partial t}+U\frac{\partial k}{\partial x}+V\frac{\partial k}{\partial y}+W\frac{\partial k}{\partial z}=\frac{\partial}{\partial x}\left(\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial z}\right)+P-G-\varepsilon$$

$$\frac{\partial \varepsilon}{\partial t}+U\frac{\partial \varepsilon}{\partial x}+V\frac{\partial \varepsilon}{\partial y}+W\frac{\partial \varepsilon}{\partial z}=\frac{\partial}{\partial x}\left(\frac{\nu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\nu_t}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial z}\right)$$

$$+C_{1\varepsilon}\frac{\varepsilon}{k}\left[P+\left(1-C_{3\varepsilon}\right)G\right]-C_{2\varepsilon}\frac{\varepsilon^2}{k}$$

wherein:

$k=\frac{1}{2}\overline{u'_i u'_i}$ denotes the turbulent kinetic energy of the fluid,

$\varepsilon=\nu\overline{\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_i}{\partial x_j}}$ is the dissipation of turbulent kinetic energy,

P is a turbulent energy production term,

G is a source term due to the gravitational forces

$$P=\nu_t\left(\frac{\partial\overline{U_i}}{\partial x_j}+\frac{\partial\overline{U_j}}{\partial x_i}\right)\frac{\partial\overline{U_i}}{\partial x_j}\qquad G=-\frac{\nu_t}{Pr_t}\frac{g}{\rho}\frac{\partial\rho}{\partial z}$$

and ν_t verifies the equality : $\nu_t=C_\mu\frac{k^2}{\varepsilon}$

C_μ , Pr_t , $C_{1\varepsilon}$, $C_{2\varepsilon}$, $C_{3\varepsilon}$, σ_k , σ_ε are constants in the K-Epsilon model.

2.2.5. EQUATIONS OF TRACERS

The tracer can be either active (it affects hydrodynamics) or passive in TELEMAC-3D. Temperature, salinity and in some cases a sediment are active tracers. The tracer evolution equation is formulated as:

$$\frac{\partial T}{\partial t}+U\frac{\partial T}{\partial x}+V\frac{\partial T}{\partial y}+W\frac{\partial T}{\partial z}=\frac{\partial}{\partial x}\left(\nu_T\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_T\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_T\frac{\partial T}{\partial z}\right)+Q$$

with :

- T (tracer unit) tracer either passive or affecting the density.
- ν_T (m^2/s) tracer diffusion coefficients.
- t (s) time.
- x , y , z (m) space components.
- Q (tracer(s) unit) tracer source or sink.

2.3. MESH

2.3.1. THE DISCRETIZATION

The TELEMAC-3D mesh structure is made of prisms. In order to prepare that mesh of the 3D flow domain, a two-dimensional mesh comprising triangles which covers the computational domain (the bottom) in a plane is first constructed, as for TELEMAC-2D. The second step consists in duplicating that mesh along the vertical direction in a number of curved surfaces known as “planes”. Between two such planes, the links between the split triangles make up prisms.

The computational variables are defined at each point of the three-dimensional mesh, inclusive of bottom and surface. Thus, they are “three-dimensional variables” except, however, for the water depth and the bottom depth which are obviously defined only once along a vertical. Thus, they are “two-dimensional variables”. Some TELEMAC-3D actions are then shared with TELEMAC-2D and use the same libraries, such as the water depth computation library. Therefore, it is well understood that TELEMAC-3D should manage a couple of mesh structures: the first one is two-dimensional and is the same as that used by TELEMAC-2D, and the second one is three-dimensional. That implies managing two different numberings a detailed account of which is given below.

2.3.2. THE TWO-DIMENSIONAL MESH

The two-dimensional mesh, which is made of triangles, can be prepared using the MATISSE mesh generator included in the TELEMAC chain.

Using a mesh generator that does not belong to the TELEMAC chain involves converting the resulting file, through the STBTEL interface, to the SELAFIN format, which can be read by TELEMAC as well as by the RUBENS post-processor. In addition, STBTEL checks such things as the proper orientation of the local numbering of the mesh elements.

The two-dimensional mesh (included in the GEOMETRY FILE) consists of NELEM2 elements and NPOIN2 apices of elements which are known through their X, Y, Z co-ordinates (the BOTTOM variable). Each element is identified by a code known as IELM2 and includes NDP nodes (3 for a triangle with a linear interpolation). The nodes on an element are identified by a local number ranging from 1 to NDP. The link between that element-wise numbering (local numbering) and the mesh node numbering ranging from 1 to NPOIN2 (global numbering) is made through the connectivity table IKLE2. The global number of the IDP local-number node in the IELEM2 element is IKLE2 (IELEM2,IDP).

2.3.3. THE THREE-DIMENSIONAL MESH

The three-dimensional mesh, which is made of prisms, is automatically constructed by TELEMAC- 3D from the previous mesh. The data in the three-dimensional mesh of finite elements are as follows:

- NPOIN3 : the number of points in the mesh (NPOIN3 = NPOIN2 x NPLAN).
- NELEM3 : the number of elements in the mesh.
- NPLAN : the number of planes in the mesh.
- X, Y, Z : NPOIN3 dimensional arrays. X and Y are obtained by merely duplicating the above-described arrays of the two-dimensional mesh. Dimension Z obviously depends on the mesh construction being selected (keyword MESH TRANSFORMATION).
- IKLE3 : dimensional arrays (NELEM3,6). IKLE3(IELEM3,IDP) provides with the global number of the IDP point in the IELEM3 element. IKLE3 defines a numbering of the 3D elements and a local numbering of the points in each element, it provides for the transition from that local numbering to the global numbering.

From these data, TELEMAC-3D constructs other arrays, such as the edge point global address array.

Figure 1 hereinbelow illustrates a TELEMAC-3D three-dimensional mesh.

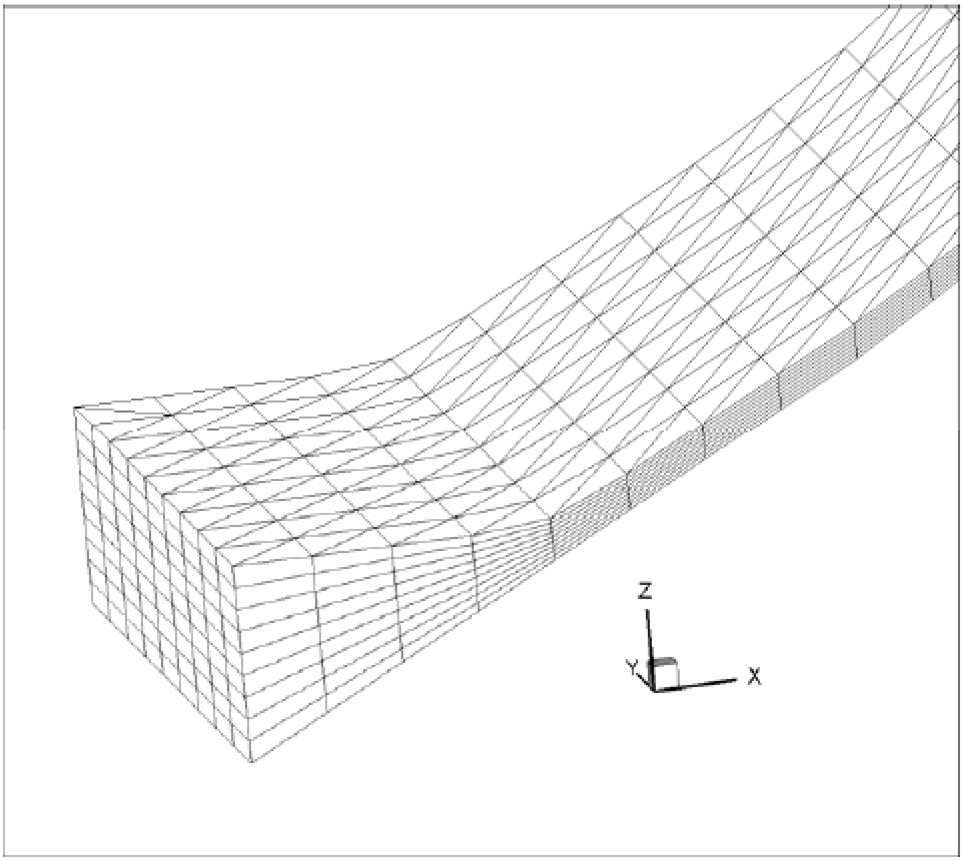


Figure 1 : A three-dimensional mesh

3. THE INPUTS / OUTPUTS

In a computation, the TELEMAC-3D code uses a number of input and output files, some of which are optional. Most of these files are similar or identical to their counterparts in TELEMAC-2D.

The input files are:

- The steering file (mandatory),
- The geometry file (mandatory),
- The boundary conditions file (mandatory),
- The liquid boundaries file,
- The bottom topography file,
- The Fortran file,
- The previous computation file.

The output files are:

- The 3D result file,
- The 2D result file,
- The output listing,
- The Scope file.

The input and/or output files are:

- The reference file,
- The binary files 1 and 2,
- The formatted files 1 and 2.

3.1. THE STEERING FILE

The steering file contains all the data about the selection of computational options (physical, numerical, etc.). It is an ASCII test file which can be generated either through the FUDAA-PREPRO software or directly using a text editor. In a way, it serves as the computation dashboard. It includes a set of keywords to which values are assigned. If a keyword does not occur in that file, then TELEMAC-2D will assign it the default value as defined in the dictionary file. If such a default value is not defined in the dictionary, then the computation will be interrupted with an error message. For instance, the command TIME STEP = 10.0 enables to specify that the computation time step value is 10 seconds.

TELEMAC-3D reads the steering file at the beginning of the computation

Both dictionary file and steering file are read using the DAMOCLES library which is included in the TELEMAC chain. It is therefore necessary, when generating the steering file, to observe the DAMOCLES syntax rules (what is performed automatically if the file is generated using FUDAAPREPRO). These rules are described below and an example is dealt with in Appendix No. 4.

The writing rules are as follows:

- The keywords can be of the Integer, Real, Logical or Character type.
- The sequence order in the steering file is of no importance.
- Several keywords can be on the same line.
- Each line cannot exceed 72 characters. However, one can start a new line as many times as one wishes provided that the keyword name is not astride two lines.
- For the array-types keywords, the character separating successive values is the semicolon. For example :

```
PRESCRIBED FLOWRATES = 10.0;20.0
```

- The symbols”:” and ”=” are used equally well in order to separate a keyword name and its value. They can be either preceded or followed by any number of blanks. The value itself can occur on the following line. For example:

```
TIME STEP = 10.
```

or

```
TIME STEP : 10.
```

Or else

```
TIME STEP =
10
```

- Those characters occurring between two ”/” on one line are taken as comments. Likewise, those characters occurring between a ”/” and a line ending are considered as comments.

For example:

```
TURBULENCE MODEL = 3 / K-Epsilon model
```

- A line beginning with a ”/” in a first column is wholly considered as a comment, even though there is another ¨/¨ on the line. For example:

```
/ The geometry file is ./mesh/geo
```

- Writing the integers: do not exceed the maximum size allowed by the machine (in a machine with a 32 bit architecture, the extreme values range from -2 147 483 647 to +2 147 483 648. Do not insert any blank between the sign (optional for the sign +) and the number. A point after the end of the number is tolerated.
- Writing the reals: the point and the comma are accepted as a decimal separator, as well as the Fortran formats E and D. (1.E-3 0.001 0,001 1.D-3 represent the same value).
- Writing the logical values: the values 1 OUI YES .TRUE. TRUE VRAI on the one hand, and 0 NON NO .FALSE. FALSE FAUX on the other hand, are accepted.
- Writing the character strings: the strings including blanks or reserved symbols (”/”,”:”, ”=”, ”&”) should be inserted between quotes ('). The value of a character keyword may contain up to 144 characters. As in Fortran, the quotes included in a string should be doubled. A string may not begin or end with a blank. For example:

```
TITLE = 'COASTAL ENVIRONMENT STUDY'
```

In addition to the keywords, a number of pseudo instructions or metacommands which are interpreted during the sequential reading of the steering file can also be used:

- The &FIN command specifies the file end (even though the file is not over). Thus, some keywords can be disabled simply by placing them behind that command so that they can easily be reactivated later on. However, the computation keeps running.
- The &ETA command prints the list of keywords and the values which are assigned to them when DAMOCLES meets that command. That display will take place at the head of the output listing (refer to para. 3.2.6).
- The &LIS command prints the list of keywords. That display will take place at the head of the output listing (refer to para. 3.2.6).
- The &IND command prints the detailed list of keywords. That display will take place at the head of the output listing (refer to para. 3.2.6).

• The &STO command causes the program to be halted, whereas the computation does not keep running.

3.2. THE GEOMETRY FILE

It is the same file as that used by TELEMAC-2D. It is a SELAFIN-formatted binary file which can then be read by RUBENS and is generated either by the MATISSE software or by the STBTEL module (from the file(s) originating from of mesh generator). The SELAFIN format structure is described in 0.

That file contains all the data about the two-dimensional mesh (see in para. 2.3.2). It includes the number of points in the mesh (NPOIN2 variable), the number of elements (NELEM2 variable), the number of apices per element (NDP variable), the X and Y arrays containing the co-ordinates of all the points and, lastly, the IKLE array containing the connectivity table.

That file may also contain bathymetric data for each point in the mesh.

NOTE : TELEMAC-3D retrieves the geometry data at the beginning of the 2D result file. Any computational 2D result file can therefore be used as a geometry file when a further simulation on the same mesh is desirable.

That file name is provided using the keyword: GEOMETRY FILE.

3.3. THE BOUNDARY CONDITIONS FILE

It is the same file as that uused by TELEMAC-2D. It is a formatted file which is generated automatically by MATISSE or STBTEL and can be amended using FUDAA-PREPRO or a text editor. Each line in that file is dedicated to a point at the 2D mesh boundary. The edge point numbering is that of the file lines; it first describes the domain outline in the counterclockwise direction from the bottom left-hand point (point the sum of co-ordinates of which is minimum), then the islands in the clockwise direction.

For a thorough description of that file, refer to the specific paragraph 4.3.4.

That file's name is provided using the keyword: BOUNDARY CONDITIONS FILE

3.4. THE LIQUID BOUNDARIES FILE

It is a file enabling the user to specify time-varying boundary conditions values (flow rte, depth, velocity, tracer concentration) at all the liquid boundaries. That filke can be generated under the FUDAA-PREPRO software interface.

For a thorough description of that file, refer to the specific paragraph 4.3.7.

That file's name is provided using the keyword: LIQUID BOUNDARIES FILE.

3.5. THE FORTRAN FILE

The Fortran file may include a number of subroutines (so-called "user" subroutines) available under the TELEMAC-3D tree structure which the user can modify as well as those subroutines which have been specifically developed for the computation.

The user subroutines from the various libraries as used by TELEMAC-3D are listed in 0. Every user subroutine being copied into the user Fortran file is automatically substituted for the samenamed subroutine occurring in the TELEMAC-3D compiled libraries.

Upon the creation and every amendment of the Fortran file, a new executable program is generated (compilation and link) for the simulation.

That file name is provided using the keyword: FORTRAN FILE. An example of a Fortran file is given in 0.

3.6. THE PREVIOUS COMPUTATION FILE

It is a TELEMAC-3D result file which is used for initializing a new computation. In order to activate the optional use of that file, the keyword COMPUTATION CONTINUES should be activated. In order to specify the previous computation file, its name should be stated through the keyword: PREVIOUS COMPUTATION FILE. The initial conditions of the new computation are defined by the last backup time step temps in the previous computation file. The whole set of data from the steering file is read and makes it possible to redefine or amend the variables (time step, turbulence model, addition or deletion of a tracer…).

A computation can also be initialized from a TELEMAC-2D result. In order to activate that option, the 2D CONTINUATION keyword should be validated. The TELEMAC-2D result file should then be associated with the BINARY DATA FILE 1 keyword.

3.7. THE 3D RESULT FILE

It is the file into which TELEMAC-3D stores the information during the computation. It is a SELAFIN-formatted file (refer to 0). It first contains all the data about mesh geometry,then the names of the stored variables. It also contains, for each time step and for each mesh point, the values of the various backed up variables.

Its content varies according to the values of the following keywords:-

NUMBER OF FIRST TIME STEP FOR GRAPHIC PRINTOUTS: provided for determining from which time step onwards storing information is desirable, in order to prevent too bulky files, especially when the computation begins with an uninteresting transient stage related to the definition of unrealistic initial conditions (e.g. invariably zero currents)

GRAPHIC PRINTOUT PERIOD: sets the period (in number of time steps) of printouts in order to prevent a too bulky file. For instance:

```
TIME STEP = 60.0
GRAPHIC PRINTOUT PERIOD = 30
```

The results will be backed iup every 1,800th second, i.e.30th minute

VARIABLES FOR 3D GRAPHIC PRINTOUTS: provided for specifying the list of variables which will be stored in the result file. Each variable is identified by means of a name from the list below.

- U velocity along x axis (m/s) ;
- V velocity along y axis (m/s) ;
- W velocity along z axis (m/s) ;
- TA concentrations for tracers (TA1 for the 1st one, TA2 for the 2nd one …) ;
- NUX viscosity for U and V along x axis (m^2/s) ;
- NUY viscosity for U and V along y axis (m^2/s) ;
- NUZ viscosity for U and V along z axis (m^2/s) ;
- NAX viscosity for tracers along x axis (m^2/s) ;
- NAY viscosity for tracers along y axis (m^2/s) ;
- NAZ viscosity for tracers along z axis (m^2/s) ;
- RI Richardson number in case of a mixing length model;
- K turbulent energy for k-epsilon model (J/kg) ;
- E dissipation of turbulent energy (W/kg) ;
- DP dynamic pressure (multiplied by DT/RHO) ;
- RHO relative density.

That file name is provided using the keyword: 3D RESULT FILE.

3.8. THE 2D RESULT FILE

It is the file into which TELEMAC-3D stores the specifically two-dimensional data during the computation (such as the free surface, the horizontal components of velocity and the verticallyaveraged tracers). It has a SELAFIN format. The free surface and the horizontal components of velocity will then physically correspond to the same data as those being supplied by TELEMAC-2D. The obtained values, however, may be different from an analogue computation being directly made with TELEMAC-2D when the flow is specifically three-dimensional.

Its content varies according to the values of the following keywords: NUMBER OF FIRST TIME STEP FOR GRAPHIC PRINTOUTS: same keyword as that described in para. 3.6.

GRAPHIC PRINTOUT PERIOD: same keyword as that described in para. 3.6.

VARIABLES FOR 2D GRAPHIC PRINTOUTS: provided for specifying the list of variables which will be stored in the result file. Each variable is identified by means of a name from the list below.

- U average velocity along x axis (m/s) ;
- V average velocity along y axis (m/s) ;
- C celerity (m/s) ;
- H water depth (m) ;
- S free surface elevation (m) ;
- B bottom elevation (m) ;
- F Froude number (m) ;
- Q scalar discharge (m^2/s) ;
- I discharge along x (m^2/s) ;
- J discharge along y (m^2/s) ;
- M norm of velocity (m/s) ;
- X wind along x axis (m/s) ;
- Y wind along y axis (m/s) ;
- P air pressure (Pa) ;
- W friction coefficient ;
- RB non erodable bottom elevation (m) ;
- thickness of the fresh deposits (m) ;
- erosion rate (kg/m^3/s) ;
- probability of deposition;
- PRIVE1 work array PRIVE 1 ;
- PRIVE2 work array PRIVE 2 ;
- PRIVE3 work array PRIVE 3 ;
- PRIVE4 work array PRIVE 4 ;
- US friction velocity

That file name is provided by means of the keyword: 2D RESULT FILE.

3.9. THE OUTPUT LISTING

It is a formatted file which can be created par TELEMAC-3D during the computation (program launching with the –s option). It contains the report of a TELEMAC-3D running. Its contents vary according to the values of the following keywords:

NUMBER OF FIRST TIME STEP FOR LISTING PRINTOUTS: provides for determining at which time step it is desired to begin editing the data, in order to prevent too bulky files.

LISTING PRINTOUT PERIOD: sets the period between two time step editings. The value is given in time step number. Por instance, the following sequence:

```
TIME STEP = 30.0
LISTING PRINTOUT PERIOD = 2
```

Editing in the output listing every minute of simulation.

MASS-BALANCE: if it is requested, the user will get information about the mass fmlow (or rather the volumes) and the errors (primarily linked to the precision achieved by the solvers) of that computation in the domain.

INFORMATION ABOUT MASS-BALANCE FOR EACH LISTING PRINTOUT: if they are requested, the user will get, at each time step, information about the flows within the domain.

The file name is directly managed by the TELEMAC-3D launching procedure. Generally, the file has a name which is prepared from the name in the steering file and the number of the process having carried out the computation, followed by the suffix ".sortie". A brief example of an output listing is given in 0.

3.10. THE SCOPE FILE

TELEMAC-3D gives the user an opportunity to retrieve computation variables along a 1D profile during a computation. These profiles will then be backed up in a dedicated file the name of which is defined by the keyword FILE FOR SCOPE. To that purpose, the user should program the SCOPE subroutine.

That subroutine is provided for preparing profiles of computed variables or other variables as created by a user along a segments with (X1,Y1,Z1) and (X2,Y2,Z2) co-ordinates. The user also checks the number of points as distributed along that segment. The data are automatically backed up as per the SCOPE format at all the time steps.

3.11. THE REFERENCE FILE

Upon a validation computation, that file contains the reference file. On completion of the computation, the result of the simulation is compared with the last time step as stored into that file. The result of that comparison is provided in the check listing in the form of a maximum deviation over depth and the velocity components.

That file name is provided by means of the keyword: REFERENCE FILE.

3.12. THE AUXILIARY FILES

Other files can be used by TELEMAC-3D.

One or two binary data files, as specified by the keywords BINARY FILE 1 and BINARY FILE 2. These files can be used for providing data to the program, the user, obviously, having to handle their reading within the Fortran program. The data from these files shall be read in:

- the NBI1 logic unit (value 24) for binary file 1,

- the NBI2 logic unit (value 25) for binary file 2.

One or two formatted data files, as specified by the keywords FORMATTED FILE 1 and FORMATTED FILE 2. These files can be used from providing data to the program, the user, obviously, having to handle their reading within the Fortran program. The data from these files shall be read in:

- the NFO1 logic unit (value 26) for formatted file 1,

- the NFO1 logic unit (value 27) for formatted file 2.

The read or write operations from/into these files should be thoroughly managed by the user (the files are opened and closed by the program). That management can be performed from any point which the user can gain access to. The logic unit numbers NBI1, NBI2, NFO1, NFO2 are stated in the DECLARATIONS_TELEMAC module and the user can access them through a USE DECLARATIONS_TELEMAC command at the beginning of the subroutine. For instance, using a file for providing the initial conditions will result in managing that file within the CONDIN subroutine. Likewise, using a file for inserting boundary conditions will be possible at the BORD3D subroutine. In case of conflicting statements, one can use, for example: DECLARATIONS_TELEMAC, ONY : NBI1.

4. GENERAL SETUP OF THE HYDRODYNAMIC COMPUTATION (NAVIER-STOKES EQUATIONS)

The general setup of the computation is only performed at the steering file level.

The time data are provided by the two keywords TIME STEP (real) and NUMBER OF TIME STEPS. The former set the period of time between two consecutive computational moments (but not necessarily two outputs in the result file) The global duration of the computation is provided through a number of time steps (keyword NUMBER OF TIME STEPS). The global duration is quite obviously equal to the value of the time steps multiplied by the number of time steps.

Both date and hour corresponding to the initial time of the computations can be specified using the keywords ORIGINAL DATE OF TIME (AAAA, MM, JJ format) and ORIGINAL HOUR OF TIME (HH, MM, SS format). Thus, these two data are optional; however, they can be taken into account in programming by means of the MARDAT and MARTIM variables.

The computation title is specified by the keyword TITLE.

Upon the executable file generation, the version of libraries being used is specified by the optional keyword RELEASE.

4.1. MESH DEFINITION

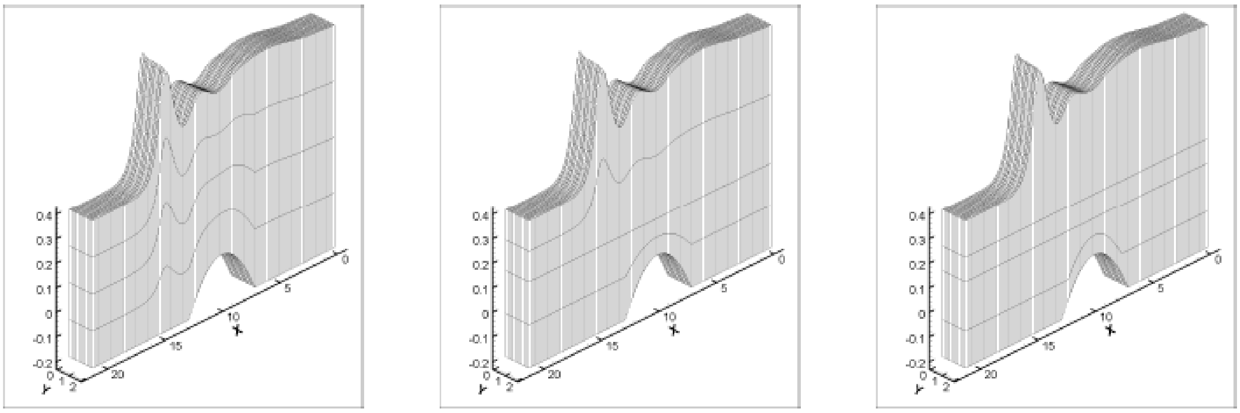
The three-dimensional mesh, consisting of prisms, is automatically constructed by TELEMAC-3D from the three-dimensional mesh. The number of prisms is specified in the data file by means of the keyword NUMBER OF HORIZONTAL LEVELS. That number of levels is equivalent to the number of prisms plus 1. Its minimum value is 2 (1 prism in the vertical direction).

The keyword MESH TRANSFORMATION makes it possible to define the kind of extrapolation in the two-dimensional mesh. The default value is 1 (Fig. 2 (a)) and results in a homogeneous distribution of levels in the vertical direction. The 2 value is provided for defining the distribution of levels (e.g. refinement near surface). The latter choice implies that the user will program his/her distribution in the CONDIN subroutine.

TELEMAC-3D uses a change of variables in order to freeze the mesh on a time step (without such a change, the mesh dimensions z vary in accordance with the free surface evolution). The frequently adopted change of variables is the sigma transform which consists in shifting from the z (x,y,t) co-ordinate to the z* (x,y) co-ordinate. The user should enter the z* co-ordinates in the CONDIN subroutine. The normalized co-ordinates will then range from 0 (the bottom) to 1 (the surface).

For example, let:

```
ZSTAR%R(1)=0.D0
ZSTAR%R(2)=0.3D0
ZSTAR%R(3)=1.D0
```



(a) 1 : sigma 2 : zstar (b) 3 : fixed horizontal level (c) 4 : horizontal levels

Figure 2 : Effect of the MESH TRANSFORMATION keyword .
(example after the TELEMAC-3D case study of the hydraulic jump)

For a better representation of the densimetric stratification areas (thermoclines, halocline and/or outfall), prescribing a maximum number of horizontal “levels” (particularly those where the gradients are highest) is sometimes suitable. To that purpose, the use can select the values 3 or 4 at the keyword MESH TRANSFORMATION.

The value 3 involves that the following keywords be specified:

- NUMBER OF THE INTERMEDIATE REFERENCE LEVEL. That first parameter defines the intermediate level from which the higher levels no longer depend on the bottom depth (at least as long as the defined depth is larger than the bottom depth), but only depend on the free surface evolution (refer to the example in Fig. 2 (b) for which the NUMBER OF THE INTERMEDIATE REFERENCE LEVEL is 3).
- ELEVATION OF INTERMEDIATE REFERENCE LEVEL. That second parameter sets the actual elevation which the user wishes for that level. The elevation as set by the user can be changed automatically by the program either if the bottom depth is larger or if the free surface elevation is smaller (the elevations of points along a vertical line should increase between the bottom and the free surface).

By default, the vertical distribution of levels on either side of the reference level is homogenous. The user can modify that distribution by specifying the z* co-ordinates in the CONDIN subroutine.

Value 4 consists in setting (necessarily by programming in the CONDIN subroutine) the actual elevations of all the levels (Figure 2 ©). As previously, that elevation can vary in accordance with the bottom and the free surface (check for increasing elevations of points between the bottom and the surface).

Value 0 is left available to the user for every other method he would want to program in the CALCOT.f subroutine.

4.2. PRESCRIBING THE INITIAL CONDITIONS

The initial conditions aim at defining the model condition at the beginning of the simulation.

As regards a computational sequence, that conditions is provided by the last time step in the result file of the previous computation (refer to para. 3.6). The indispensable variables (at least the velocity components) when resuming the computation should then have been stored into the file being used upon such resumption.

Otherwise, the default initial condition is defined as follows:

- Steady zero free surface,
- Zero velocities,
- Steady zero active and passive tracers.

If that initial condition is not suitable for a computation, then it should be changed using keywords in the simple cases or through a programming as described in the subsequent paragraphs.

4.2.1. PRESCRIPTION THROUGH KEYWORDS

In all the cases, the kind of initial conditions is set by the keyword INITIAL CONDITIONS. That keyword can have one of the following five values:

- 'ZERO ELEVATION' : Initializes the free surface elevation to 0. The initial water depths are then computed from the bottom depth.
- 'CONSTANT ELEVATION' : Initializes the free surface elevation to the values as provided by the keyword INITIAL ELEVATION. The initial water depths are then computed by getting the difference between the free surface elevation and the bottom depth. In those areas where the bottom depth exceeds the initial elevation, the initial water depth is zero
- 'ZERO DEPTH' : All the water depths are initialized with a zero value (free surface coinciding with bottom). In other words, the whole domain is “dry” at the beginning of the computation.
- 'CONSTANT DEPTH' : Initializes the water depths to the value as provided by the keyword INITIAL ELEVATION.
- 'SPECIAL' : The initial conditions are defined as programmed by the user in the CONDIN subroutine (refer to the next paragraph). That procedure should be used whenever the initial conditions of the model to not correspond to one of above four cases..

4.2.2. PRESCRIBING PARTICULAR INITIAL CONDITIONS (PROGRAMMING THE CONDIN SUBROUTINE)

The CONDIN subroutine should be programmed as soon as the initial conditions having been programmed by default are to be modified (as it often occurs). It is therefore advisable that the user will systematically include it into his/her Fortran file, although it is only optional.

By default, the standard version of the CONDIN subroutine interrupts the computation if the keyword INITIAL CONDITIONS is set to 'PARTICULAR' without any actual amendment of the subroutine.

The CONDIN subroutine successively initializes the two-dimensional variables, then the threedimensional variables:

- the water depth,
- the positions z* of vertical levels,
- the velocities,
- the active and passive tracers.

The user can quite freely fill that subroutine. For instance, he/she can retrieve information in a formatted or binary file, using to that purpose the keywords FORMATTED FILE 1 or 2 or BINARY FILE 1 or 2.

4.2.3. RESUMING THE COMPUTATION

TELEMAC-3D enables to perform a computation by taking as the initial condition the last time step of a computation which was previously made on the same mesh, inclusively with a different number of levels. Thus, such computation data as the time step, some boundary conditions, the turbulence model can me modified, or else a computation can be initiated once a steady state is achieved.

The field to be retrieved, which should have the SELAFIN format, shall then inevitably contain all the data required for TELEMAC-3D, i.e. not only the co-ordinates of the X, Y and Z computational points which it necessarily contains, but also the U, V and W velocities, the tracers.

If some variables do not occur in the resumption file, then they are automatically set to zero values. A usual application consists in using the result of a hydrodynamic computation in order to perform a tracer transport computation. Generally, the resumption file does not include any result for the tracer.

Using a resumption file implies entering two keywords into the steering file.

The keyword COMPUTATION CONTINUED should be set to the YES value.

The keyword PREVIOUS COMPUTATION FILE should provide the name of the file which will provide the initial condition.

WARNING: the two-dimensional mesh on which the useful results have been computed should be strictly identical to the mesh of the case to be handled.

4.3. PRESCRIBING THE BOUNDARY CONDITIONS

The boundary conditions are handled through types of conditions which are related to the computational variables. The combination of these types (from a list of possible choices) describes whether the boundary is liquid or solid and how it should be processed.

In TELEMAC-3D, the water depth H, the horizontal velocities U and V and the tracers are the only variables which necessarily involve defining their type of boundary conditions. That types of boundary conditions applicable to the vertical velocity and the k and Epsilon functions are managed by default by TELEMAC-3D and are therefore not requested from the user. If the computation takes tracers into account, then a single type (common to all the tracers) should also be defined for a given boundary.

Once all the types of the boundary are defined, the user should enter the related values for the computational variables (at least H, U and V).

For example, the user may want to set the sea level and leave the velocity field free (e.g. the tide case). The type of boundary will be: “prescribed depth and free velocity”. The values related to that type are only water depth at every instant at that boundary. The values of velocities (if they are entered) are not taken into account for that boundary.

Thus, for each TELEMAC-3D boundary, the computational variables (at least H, U and V) are necessarily associated with one type and each type may be associated with one value (either used or not)

After such a description of what is a boundary in TELEMAC-3D, we will describe the types, then the related values.

4.3.1. THE BOUNDARIES IN TELEMAC-3D

Water depth is the single two-dimensional variable being computed. Its processing at the boundaries is like that being performed by TELEMAC-2D. The boundary points to be handled are those of the two-dimensional mesh.

For the other variables (velocities and tracers), the boundary conditions should be handled over all the boundaries of the three-dimensional mesh which includes :

- the lateral boundary points (vertical column points linked to the boundaries of the twodimensional mesh), whether it is a liquid or solid boundary,
- the points belonging either to the free surface or the bottom (refer to Fig. 3).

figure03png

By default, TELEMAC-3D automatically handles all the surface and bottom points which do not belong to the side walls. The user, however, can handle them himself/herself if he/she wants so. To do that, he/she should act upon the Fortran sources.

All the remaining points (side edge points) are linked to the two-dimensional mesh boundaries at each horizontal level. Thus, they will be processed somewhat as in TELEMAC-2D. The number of required data, however, increases so much that a full external handling (either through the steering file or the boundary conditions file) would become excessively complex. That is why the range of options offered to the user for checking these boundary conditions is narrower than in TELEMAC- 2D and definitely implies programming the sources of the user-available software. The next following paragraphs describe the way the boundary points are handled.

4.3.2. THE BOUNDARY-RELATED TYPES

The boundary condition type for H, U, V and T of the edge points is read in the boundary conditions file. It can be either modified or directly defined by the user in the LIMTYP subroutine..

The various types of boundary conditions can be combined in order to prescribe the conditions of different physical kinds (liquid inflow or outflow in supercritical conditions, open sea, wall, etc.). Some combinations, however, are not physical (refer to para. 4.3.3. hereinafter).

Some boundary conditions are applicable to such facts as friction at the walls or wall impermeability. However, the wall definition is ambiguous if one only retains a definition of pointwise boundary conditions. The following convention is then observed in order to determine the nature of a segment lying between to different kinds of points: a liquid segment is a segment linking two liquid-types points. Thus, under that convention, the connecting point between the shore and the marine boundary (or between the river and the bank) is preferably of the liquid type.

Any sequential arrangement of the boundary types may exist along an outline (for instance, one may have a liquid boundary with a prescribed depth flowed by a liquid boundary with a prescribed velocity). The only condition to be met is that a boundary should consist of two common points (it is a computational requirement, a number of at least four points being highly advisable from a physical point of view).

4.3.3. DESCRIPTION OF THE VARIOUS TYPES

The type of boundary condition at a given point is provided, in the boundary conditions file, in the form of four integers which are referred to as LIHBOR, LIUBOR, LIVBOR and LITBOR, with values which can range from 0 to 6.

The available options are as follows:

- Depth condition :
  - Prescribed depth liquid boundary: LIHBOR=5
  - Free depth liquid boundary: LIHBOR=4
  - Solid boundary (wall): LIHBOR=2

It is noteworthy that a depth/rate law is considered as a prescribed rate condition. The flow rate value should then explicitly be computed, according to the water depth, by programming the Q3 subroutine.

- Rate or velocity condition:
  - Prescribed rate liquid boundary: LIUBOR/LIVBOR=5
  - Prescribed velocity liquid boundary: LIUBOR/LIVBOR=6
  - Free velocity liquid boundary: LIUBOR/LIVBOR=4
  - Solid boundary with sliding or friction: LIUBOR/LIVBOR=2
  - Solid boundary with one or two zero velocity components: LIUBOR and/or LIVBOR=0
- Tracer condition:
  - Prescribed tracer liquid boundary: LITBOR=5
  - Free tracer liquid boundary: LITBOR=4
  - Solid boundary (wall): LITBOR=2

Note

The type of boundary conditions can be changed at one liquid boundary. The steering file and the listing will then mention an additional liquid boundary.

4.3.4. THE BOUNDARY CONDITIONS FILE

That file is provided as standard by MATISSE or STBTEL, but it can be created or amended by means of FUDAA-PREPRO or a text editor. Each line of that file is dedicated to one point at the two-dimensional mesh boundary. The boundary point numbering is the same as that of the file lines, it first describes the domain outline in the counterclockwise direction, then the islands in the opposite direction.

The convention being observed in TELEMAC implies that the first liquid boundary is that which is defined, browsing the boundary conditions file, by the first two liquid-typed consecutive numbers. In the example below (channel case study), the first liquid boundary is that being defined by the nodes 42-47 (edge numbering) and corresponds to a prescribed depth (codes 5 4 4 at the heads of lines). The second boundary begins at number 76 and end up at number 1 and corresponds to a prescribed rate (codes 4 5 5).

```
4 5 5 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 1 1
2 2 2 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 5 2
2 2 2 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 6 3
…
2 2 2 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 44 41
5 4 4 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 2 42
5 4 4 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 45 43
5 4 4 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 86 44
…
5 4 4 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 3 47
2 2 2 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 87 48
```

```
…
2 2 2 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 73 74
2 2 2 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 74 75
4 5 5 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 4 76
4 5 5 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 88 77
…
4 5 5 0.000 0.000 0.000 0.0 2 0.000 0.000 0.000 89 85
```

For each point, and each line in the boundary conditions file, the following values are entered:

LIHBOR, LIUBOR, LIVBOR, HBOR, UBOR, VBOR, AUBOR, LITBOR, TBOR, ATBOR, BTBOR, N, K

LIHBOR, LIUBOR, LIVBOR and LITBOR are boundary-typed codes for each of the variables.

- HBOR (real) denotes the prescribed depth value when the LIHBOR value would be5.
- UBOR (real) denotes the prescribed U velocity value when the LIUBOR value would be 6.
- VBOR (real) denotes the prescribed V velocity value when the LIVBOR value would be 6.
- AUBOR denotes the coefficient value of the boundary friction law when the LIUBOR or LIVBOR value would be 2. The friction law is then written as:

$$\nu_T \frac{\mathrm{d}U}{\mathrm{d}n} = AUBOR \times U \quad \text{and/or} \quad \nu_T \frac{\mathrm{d}V}{\mathrm{d}n} = AVBOR \times V$$

The AUBOR coefficient is applicable to the segment included between the edge point being considered and the next point (in the counterclockwise direction for the outside outline and in the opposite direction for the islands). By default, the AUBOR value is 0. A friction corresponds to a negative value. With the k-Epsilon value, the value of AUBOR is automatically computed by TELEMAC-3D, the indications in the boundary conditions file being then ignored.

- TBOR (real) denotes the prescribed tracer value when the LITBOR value is 5.
- ATBOR and BTBOR denote the coefficient values of the flux law which is written as:

$$\nu_T \frac{\mathrm{d}T}{\mathrm{d}n} = ATBOR \times T + BTBOR$$

The ATBOR and BTBOR coefficients are applicable to the segment included between the edge point being considered and the next point (in the counterclockwise direction for the outside outline and in the opposite direction for the islands).

- N denotes the edge point global number.
- K denotes the point number in the edge point numbering.

As regards the horizontal velocities, all the points in one water column will have the same type of boundary condition being defined by LIUBOR or LIVBOR. That principle is intrinsic to the TELEMAC-3D formulation. Prescribing a different type of boundary condition in the vertical direction (e.g. for a subterranean stream) may, indeed, induce severe inconsistencies with the hydrostaticity hypothesis and generate, for instance, irrealistic vertical velocities. It is then advisable that the user will keep that principle. However, the boundary condition type in the vertical direction can be altered through direct programming in the LIMTYP subroutine.

The so-called LIHBOR, LIUBOR and LIVBOR integers (which define the boundary type) can assume a value ranging from 0 to 6. The available options are as follows :

- Depth-related condition:
  - Prescribed depth liquid boundary: LIHBOR=5
  - Free depth liquid boundary: LIHBOR=4
  - Solid boundary (wall): LIHBOR=2
- Velocity-related condition:
  - Prescribed velocity liquid boundary: LIUBOR/LIVBOR=6
  - Prescribed rate liquid boundary: LIUBOR/LIVBOR=5
  - Free velocity liquid boundary: LIUBOR/LIVBOR=4
  - Solid boundary with sliding or friction: LIUBOR/LIVBOR=2
  - Solid boundary with one or two zero velocity components: LIUBOR and/or LIVBOR=0

The boundary conditions of physical nature are defined by the relationship among the types of variables. In most cases, the boundary type can be set at the MATISS mesh generator in the TELEMAC chain. The table below summarizes the physical relationship among the boundary types.

LIHBOR	**LIUBOR**	**LIVBOR**	**LITBOR**	
2	2	2	2	Solid wall.
2	0	2	2	Solid wall with zero U.
2	2	0	2	Solid wall with zero V.
2	0	0	2	Solide wall with zero U and V.
5	4	4	4	Prescribed H, free velocities, free T.
5	4	0	4	Prescriped H, free U, zero V, free T.
5	0	4	4	Prescribed H, zero U, free V, free T.
1	1	1	4	Incident wave, free tracer.
4	5	5	5	Free H, prescribed Q, prescribed T.
4	5	0	5	Free H, prescribed Q with zero V, prescribed T.
4	0	5	5	Free H, prescribed Q with zero U, prescribed T.
4	6	6	5	Free H, prescribed velocities, prescribed T.
5	5	5	5	Prescribed H and Q, prescribed T.
5	6	6	5	Prescribed H and velocities, prescribed T.

4.3.5. PROGRAMMING THE BOUNDARY CONDITIONS TYPE

The LIMI3D subroutine should be programmed for handling the boundary condition type, when it is desirable, for the edge points as well as the surface and bottom points.

That subroutine is invoked upon each time step. It is then provided to change the boundary condition type in time, as required.

4.3.6. PRESCRIBING VALUES THROUGH KEYWORDS

In most simple cases, the boundary conditions will be prescribed by means of keywords. However, if the values to be prescribed are time variable, it becomes necessary to resort to program the adequate functions or use the liquid boundaries file.

The appropriate keywords for prescribing the boundary values are as follows:

- PRESCRIBED ELEVATIONS: provided for setting the elevation value of a prescribed height liquid boundary. It is an array which can contain up to 100 reals, and therefore up to 100 boundaries of that kind can be handled. The values yielded by that keyword cancel the depth values read from the boundary conditions file.

Warning: the free surface level is set thereby, whereas the water depth is set in the boundary conditions file.

- PRESCRIBED FLOWRATES: provided for setting the flow ate value of a prescribed flow at a liquid boundary. It is an array which can contain up to 100 reals, and therefore up 100 boundaries of that kind can be handled. A positive value corresponds to a domain inflow rate. The values yielded by that keyword cancel the velocity values read from the boundary conditions file.
- PRESCRIBED VELOCITIES: provided for setting the velocity value of a prescribed velocity liquid boundary. The scalar value is the wall normal intensity of velocity. A positive value corresponds to a domain inflow. It is an array which can contain up to 100 reals, and therefore up 100 boundaries of that kind can be handled. The values yielded by that keyword cancel the values read from the boundary conditions file.

Besides, several simple rules should be observed:

The boundary type as specified in the boundary conditions files should obviously be in accordance with the keywords from the steering file (do not insert the keyword PRESCRIBED FLOWRATES if there are no boundary points the LIUBOR and LIVBOR of which are set to 5). The keyword, however, is ignored when no type matches it.

For each keyword, the number of specified values should be equal to the whole number of liquid boundaries, whatever their types may be. When a boundary is inconsistent with the leyword, then the specified value is ignored (a 0.0 value or, on the contrary, a very high value such as 999.0 may be systematically inserted). In the channel test case, the first boundary (downstream boundary) is of prescribed level type whereas the second one (upstream boundary) is of prescribed flow rate type. The steering file contains a sequence of the following type:

PRESCRIBED ELEVATIONS = 0.5 ; 0.0

PRESCRIBED FLOWRATES = 0.0 ; 50.0

4.3.7. USING THE LIQUID BOUNDARIES FILE

In case of time variable values, which are nevertheless constant along the relevant liquid boundary, the prescription can be made using the liquid boundaries file (as an alternative to programming).

It is a user-edited text file the name of which should be given by the keyword FILE FOR LIQUID BOUNDARIES. That file has the following format:

- The optional line(s) beginning with the sign # (1st character on the line) are comments.
- It should exhibit a line beginning with T for identifying the supplied value(s) within that file. The identification is performed through mnemonic means which are identical to the variable names: Q for the flow rate, SL for the level, U and V for the velocities and T for the tracer. A whole value between brackets is for specifying the rank of the current boundary. That line is necessarily followed by another line indicating the unit of the variables (lines of comments can be inserted, but the line of units should be present). The units are given for information only and TELEMAC-3D does not handle the conversion of units (thuis, the user has to enter the values using the standard unit).
- The values to be prescribed are provided through a sequence of lines the format of which should be consistent with the identification lime. The time value should be increasing and the last time value being supplied should be higher than or equal to the value related to the last time step, otherwise the computation is suddenly interrupted.

Upon the retrieval of that file, TELEMAC-3D performs a linear interpolation in order to compute the value to be prescribed at a particular time step. The value which is actually prescribed by the code is printed on the check listing.

An example of a liquid boundaries file is given below.

# Example of liquid boundaries file

# 2 boundaries are managed

#

T Q(1) SL(2)

s m3/s m

0. 0. 135.0

25. 15. 135.2

100. 20. 136.

500. 20. 136.

In that example, the flow rate is prescribed at the first boundary and the free surface is prescribed at the second boundary.

4.3.8. PRESCRIBING VALUES THROUGH PROGRAMMING

Still in the case of time variable values nevertheless constant along the liquid boundary being processed, the prescription can be made simply by programming particular functions:

- VIT3 function for prescribing a velocity,
- Q3 function for prescribing a flow rate,
- SL3 function for prescribing an elevation.

Functions Q3, VIT3 and SL3 are similarly programmed. In each case, the user knows the time, the boundary rank (e.g. to determine whether the first or the second prescribed flow rate boundary is being processed). By default, the functions prescribe those values that are read from the boundary conditions file or provided by the keywords.

For instance, the body of Q3 function for prescribing a flow rate ramp during 1,000 seconds to reach the value of 400 cu.m/s can take such a form as:

```
IF (AT.LT.1000.D0) THEN
  Q3 = 400.D0 * AT/1000.D0
ELSE
  Q3 = 400.D0
ENDIF
```

Or

```
Q3 = 400.D0 * MIN(1.D0,AT/1000.D0)
```

4.3.9. PRESCRIBING COMPLEX VALUES

If ever the values to be prescribed vary in both space and time, then a programming in the BORD3D routine becomes necessary, since that subroutine enables to prescribed the values in a nodewise way That subroutine describes all the liquid boundaries (loop on NPTFR2). For each boundary point, it dermines the boundary type in order to prescribe the adequate value (velocity, elevation or flow rate). BORD programming for prescribing a flow rate, however, hardly makes any sense, since the flow rate value is generally known for the whole boundary rather than on each boundary segment..

If ever a prescribed flow rate inlet is surrounded by walls with an adherence, hten the corner velocities are cancelled.

Noteworthy, the BORD3D subroutine also makes it possible to prescribe the complex boundary values of the tracers.

4.3.10. PRESCRIBING A PROFILE

4.3.10.1. HORIZONTAL PROFILE

When processing a prescribed flow rate or prescribed velocity boundary, the user has the keyword VELOCITY PROFILES to specify which “horizontal” velocity profile should be prescribed by TELEMAC-3D. The following options are acknowledged;

- 1: The profile is normal and steady along the boundary.
- 2: The values of U and V are read from the boundary conditions file (UBOR and VBOR values). In case of a prescribed flow rate, these values are multiplied by constant in order to get the desirable flow rate.
- 3: The velocity vector is normal at the boundary and its norm is read from the boundary conditions file as the UBOR value. That value is multiplied by a constant in order to get the desirable flow rate or velocity.
- 4: The velocity vector is normal at the boundary and its norm is proportional to the square root of the water depth.

4.3.10.2. VERTICAL PROFILE

Still when processing a prescribed flow rate or prescribed velocity boundary, the user has the keyword VELOCITY VERTICAL PROFILES to specify which “vertical” velocity profile should be prescribed by TELEMAC-3D. The options for that keyword are:

- 0: user programming
- 1: constant (default value for all the liquid boundaries)
- 2: logarithmic

The user programming is performed through the VEL_PROF_Z subroutine.

5. PHYSICAL SETUP OF THE HYDRODYNAMIC COMPUTATION

A number of physical parameters may or should be specified upon a simulation. The general computation setup is only carried out at the steering file.

5.1. HYDROSTATIC ASSUMPTION

Firstly, it should be specified whether one wants to use the hydrostatic assumption or not. That choice is made using the keyword NON-HYDROSTATIC VERSION which, by default, is set to NO. It should be reminded that the hydrostatic pressure assumption consists in simplifying the W vertical velocity assumption, ignoring the diffusion, advection and other terms. Therefore, the pressure at a point is only related to the weight of the overlying water column and to the atmospheric at the surface. Without the hydrostatic assumption (NON-HYDROSTATIC VERSION = YES), TELEMAC-3D solves a W vertical velocity equation which is similar to the U and V equations, with the additional gravity term.

5.2. MODELLING TURBULENCE

The Reynolds numbers (ν R =UL) reached by the currents at sea or in an estuary are excessively high and illustrate basically turbulent flows (L, the scale of eddies, for example, assumes the value of water depth h for a vertically homogeneous flow). For such a kind of flow, the turbulence-induced momentum is by far prevailing (in relation to the molecular diffusion). That diffusion is strictly defined by a tensor which represents different characteristics according to the directions.

The concept of eddy scale, however, is spatially constrained by the horizontal and vertical scales of the modelled domain. At sea, for example, a one kilometre long cape can generates eddies the dimensions of which are horizontally related to that scale. Vertically, however, the eddy size is constrained by the water depth also and even more by possible effects of stratifications. Synthetically, a common practice consists in separating the vertical and horizontal turbulence scales which are not relevant to the same dynamics for the standard applications of TELEMAC-3D. That involves defining horizontal as well as vertical viscosities rather than a single viscosity. On the open sea, for instance, the horizontal and vertical viscosities differ in several orders of magnitude.

Thus, the implementation of TELEMAC-3D requires defining two models of horizontal and vertical turbulence (HORIZONTAL TURBULENCE MODEL, VERTICAL TURBULENCE MODEL).

Turbulence modelling is an awkward task and TELEMAC-3D offers the user several approach options which are different, but also increasingly complex, and are applicable to velocities as well as to active and passive tracers.

5.2.1. CONSTANT VISCOSITY

The simplest turbulence model consists in using a constant viscosity coefficient (option for parameters: 1="CONSTANT VISCOSITY"). In that case, the latter includes the effects of molecular viscosity and dispersion (refer to Theoretical note [1]). The horizontal and vertical turbulent viscosities are then constant throughout the domain. The global (molecular + turbulent) viscosity coefficients) are provided by the user by means of the keywords COEFFICIENT FOR HORIZONTAL DIFFUSION OF VELOCITIES and COEFFICIENT FOR VERTICAL DIFFUSION OF VELOCITIES, as set by default to 10-6.

The value of that coefficient has a definite effect on both size and shape of the recirculations and eddies. A low value will tend to only dissipate the small-sized eddies, a high value will tend to dissipates de large-sized recirculations. The user shall then carefully select that value according to the case being studied. Usually, that value becomes a model calibration data by comparison with measurements. Besides, it is worth mentioning that a value bringing about the dissipation of recirculations of a smaller than two mesh cell extent has nearly no influence on the computation (i.e. there is a threshold beyond which the viscosity or turbulence value has substantially no effect).

TELEMAC-3D makes it possible to get a space- and time-variable coefficient. The VISCOS routine will necessarily be programmed. Within that routine, geometrical information, basic hydrodynamic information (water depth, velocity components) and time are made available to the user.

That option theoretically aims at enabling the user to define himself/herself the turbulent viscosity by programming the VISCOS subroutine.

5.2.2. MIXING LENGTH (VERTICAL MODEL)

The user also has the opportunity to use a vertical mixing length model (VERTICAL TURBULENCE MODEL : 2=“MIXING LENGTH”). The vertical diffusivity of velocities is then automatically computed by TELEMAC-3D by means of the selected mixing length model taking or not taking the effects of density into account. The mixing length model expresses the turbulent viscosity (pr diffusion coefficient) as a function of the mean velocity gradient and the mixing length (Prandtl’s theory):

$$\nu = L_m^2 \sqrt{2 D_{ij} D_{ij}} \; : \quad \text{where} \quad D_{ij} = \frac{1}{2}\left(\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i} \right)$$

Due to that choice, the user should enter the following options for the mixing length model (MIXING LENGTH MODEL):

- 1: PRANDTL. Standard Prandtl’s model. That formulation suits such flows with a strong barotropic component as the tidal flows.
- 3: NEZU ET NAKAGAWA. Nezu and Nakagawa model
- 4: JET. The mixing length is not constructed from the water depth, but from the jet height (only for a heat outfall).
- 5: QUETIN. Better representation of wind drift. In windy weather, a surficial boundary layer is formed and viscosity decreases.
- 6: TSANIS. Better representation of wind drift.

The graph below shows the variations of the mixing length for the various models.

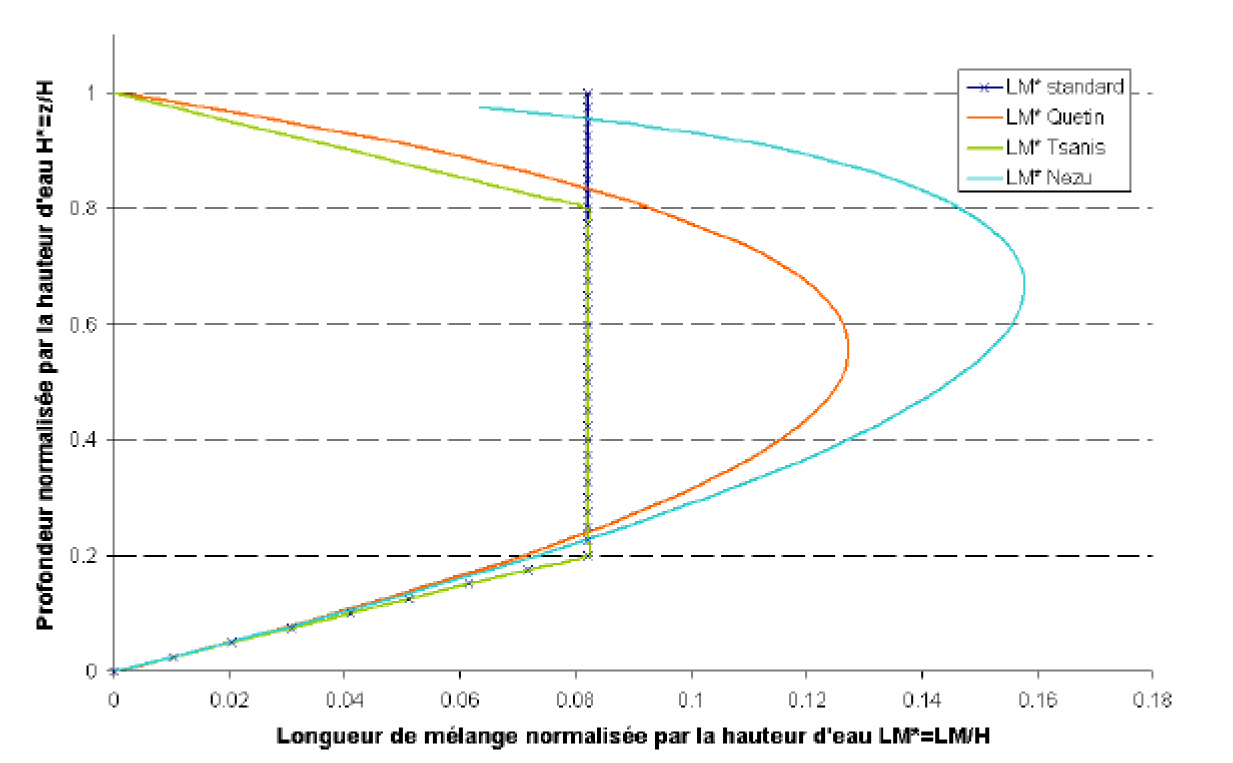


Figure 4 : Mixing lengths versus depth.

In the presence of a vertical density gradient, the environment stability (respectively the instability) hinders (enhances) the vertical exchanges of mass and momentum.

In order to quantify the effects of the gravity terms in the turbulent power balance, the dimensionless Richardson number is commonly used. It is a local number the value of which can obviously be different at each flow point.

In order to take the mixture reduction into a stable stratified flow into account, a damping law is introduced into the turbulence model according to the Richardson number. The user can check himself/herself the damping function through the keyword DAMPING FUNCTION. The available options are:

- 0: NOTHING
- 1: USER-PERFORMED; Law programming in the DRIUTI routine.
- 2: VIOLLET
- 3: MUNK AND ANDERSON

The graph below illustrates the variation of the Munk and Anderson damping function according to the Richardson number for velocity and salinity. In the case of a stable stratification, the pressure fluctuations more readily transmit a momentum flux than a mass flux and the diffusion coefficient becomes higher for the velocities than for the mass.

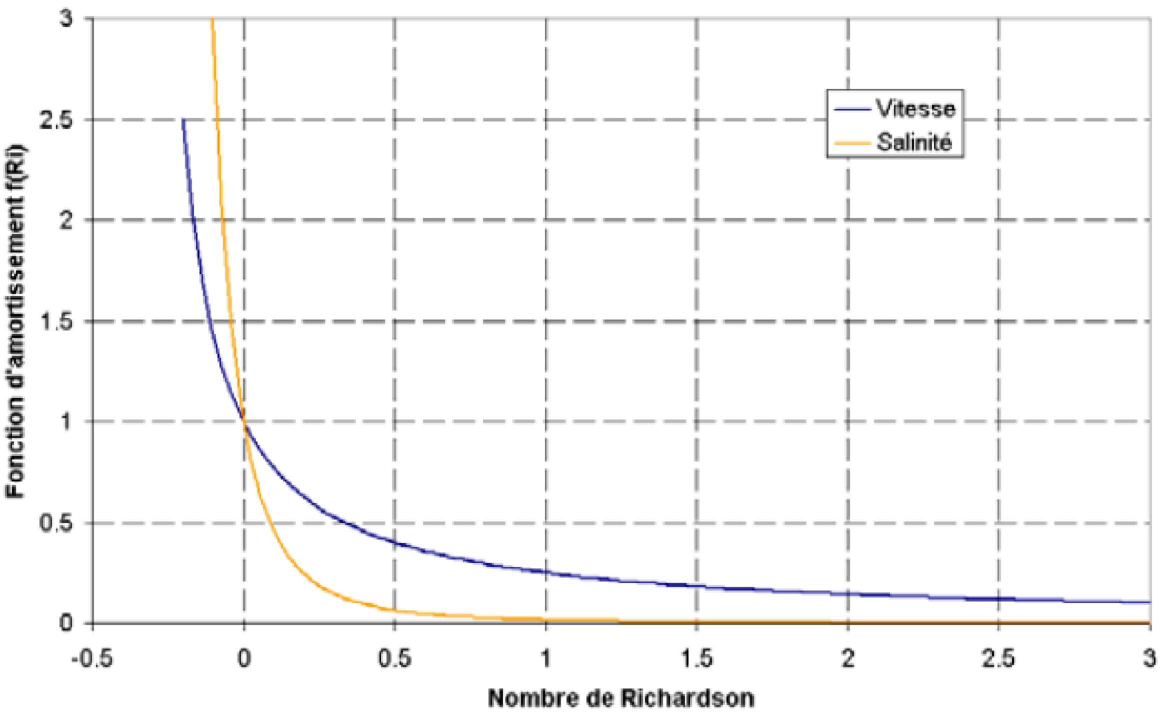


Figure 5 : Munk and Anderson damping function.

5.2.3. SMAGORINSKY (HORIZONTAL MODEL)

That option is activated by setting the HORIZONTAL TURBULENCE MODEL to 4 (Smagorinsky).

The Smagorinsky scheme is recommended, in particular, in the presence of a highly non-linear flow.

5.2.4. K-EPSILON

TELEMAC-3D gives an opportunity to use the so-called k-Epsilon model as proposed by RODI and LAUNDER for solving the turbulence equations. That model is activated by setting the keywords of turbulence models (HORIZONTAL TURBULENCE MODEL and VERTICAL TURBULENCE MODEL) to the value 3.

The k-Epsilon model comprises a couple of equations solving the balance equations for k (turbulent energy) and Epsilon (turbulent dissipation). Applying the k-Epsilon model often requires using a finer two-dimensional mesh than the constant viscosity model and then increases the computation times.

For detailed information about the formulation of the mixing length and k-Epsilon models, the user can refer to the TELEMAC-3D Theoretical Note.

Strictly speaking, except for the constant viscosity model, the diffusion coefficient should be equal to the molecular diffusion of water:

```
COEFFICIENT FOR HORIZONTAL DIFFUSION OF VELOCITIES = 1.D-6
COEFFICIENT FOR VERTICAL DIFFUSION OF VELOCITIES = 1.D-6
```

The basic value of that viscosity may have to be increased to ensure a minimum diffusion, especially upon the initial instants of the computation.

5.3. SETTING UP THE FRICTION

The bottom or sidewall friction reflects the continuity of the constraint at the fluid-liquid interface. Knowing the constraint involves knowing the flow in the vicinity of the bottom. The turbulence models provide a modelling for that flow. The constraint can we written in several forms:

$$\vec{\tau} = -\rho U^{*2} = -\frac{1}{2}\rho C_f \sqrt{U^2+V^2}\vec{U} = \mu\frac{\partial \vec{U}}{\partial n}$$

where U^* denotes the friction, C_f - a dimensionless friction, $\vec{U}$ - the velocity of current recorded far enough from the wall.

The friction condition is then provided:

- Either by a turbulence model which indicates the constraint through a friction velocity formulation,
- Or by the knowledge of the friction coefficient C_f and the related velocity $\vec{U}$ (here, the vertically-averaged velocity). That approach will then use the Chézy, Strickler, Manning … laws.

5.3.1. BOTTOM FRICTION

The friction law being used for the bottom friction modelling is set by the keyword LAW OF BOTTOM FRICTION which can assume the following values:

- 0 : No friction.
- 1 : Haaland law.
- 2 : Chézy law (default value).
- 3 : Strickler law.
- 4 : Manning law.
- 5 : Nikuradze law.

As regards the 1-5 values, the value of the friction coefficient corresponding to the selected law shall be given by means of the keyword FRICTION COEFFICIENT. Obviously, that only holds true if the friction is constant in both space and time. The default value for that parameter is 60.

The computation of turbulent constraint at the bottom depends on the velocity profile avec the bottom (within the boundary layer). The profile depends on the ratio of the wall asperity size to the viscous sub-layer thickness (for further details, refer to the Theoretical Manual). When the asperities are larger than the viscous sub-layer thickness, the latter cannot be established and the friction regime is rough. On the other hand, when there is a viscous sub-layer, the friction regime is smooth.

The computation of the turbulent constraint depends on the keyword TURBULENCE MODEL FOR THE BOTTOM. The available options are:

```
1 : smooth regime
2 : rough (default value)
```

In smooth friction regime conditions, the friction law is not used and the constraint is computed from de Reichard law of velocity profile (a law giving the friction velocity value U_*).

In rough friction regime conditions and for the bottom friction laws 0, 2, 3 and 4, the constraint is computed from the friction velocity U_* and its relation to the C_f coefficient. For law 5, the friction velocity is computed from the velocity profile within the logarithmic layer and from the aspecity size k_S (FRICTION COEFFICIENT).

5.3.2. SIDEWALL FRICTION

The same approach is adopted for both sidewalls and bottom. The friction law used for modelling the sidewall friction is set by the keyword LAW OF FRICTION ON LATERAL BOUNDARIES which can take the following values:

- 0 : No friction (default value).
- 1 : Nikuradse law.

The size of asperities (which is used in the Nikuradse law) is given by the FRICTION COEFFICIENT FOR LATERAL SOLID BOUNDARIES (default value 60!).

The friction is activated using the keyword TURBULENCE MODEL FOR LATERAL SOLID BOUNDARIES . The available options are:

```
 1 : smooth regime
 2 : rough (default value)
```

That option changes the formulation of the velocity profile and, consequently, of the friction velocity.

5.4. PUNCTUAL SOURCE TERMS

TELEMAC-3D offers an opportunity to place momentum sources or sinls in any point of the domain.

The user positions the various sources using the keywords ABSCISSAE OF SOURCES , ORDINATES OF SOURCES. They are arrays of reals giving the coordinates of sources, in meters. Actually, TELEMAC-3D will position a source at the closest mesh point to the point as specified by these keywords. The software will determine itself the number of source according to the number of values being given to each keyword.

At each source, the user should specify the liquid flow rate. That liquid flow rate is given (in cu.m/s) using the keyword WATER DISCHARGE OF SOURCES.

Besides, TELEMAC-3D is provided for taking into account an injection velocity (in m/s) at the sources in the dynamic equations. sources. By default, the injection takes places without any momentum input. The user may prescribe a particular velocity. If the latter is constant throughout the simulation, then its value can be given using the couple of keywords VELOCITIES OF THE SOURCES ALONG X and VELOCITIES OF THE SOURCES ALONG Y. Otherwise, the user should program is the SOURCE subroutine in order to amend USCE (for the velocity at sources along X) et VSCE (for the velocity at sources along Y). The user can get the time and all the parameters of the sources from that subroutine.

5.5. SETTING UP THE WATER-ATMOSPHERE EXCHANGES

5.5.1. THE WIND

TELEMAC-3D makes it possible to carry out a flow simulation taking into account the influence of the wind blowing at the surface of the water body. The logicaL keyword WIND first enables to determine whether that influence is taken into account or not. The wind influence coefficient will then be yielded by the keyword COEFFICIENT OF WIND INFLUENCE. Lastly, the wind velocities along X axis and Y axis will be yielded by the keywords WIND VELOCITY ALONG X and WIND VELOCITY ALONG Y.

The whole formulation of the consideration of wind effects, through the keyword COEFFICIENT OF WIND INFLUENCE (refer to the Principle Note for the definition of that coefficient), on the surface flows is fully stated in the BORD3D subroutine. Two options are available:

- the coefficient is that being set in the steering file,
- the coefficient depends on the wind intensity (an example is set forth in the BORD3D subroutine).

If the wind velocity is space- or time-variable, the user should act at the METEO subroutine .

WARNING: the METEO subroutine is provided to define the wind velocity and direction even though they are space- or time-variable. The BORD3D subroutine describes the law of wind-induced drift of water bodies.

5.5.2. THE TEMPERATURE

TELEMAC-3D makes it possible to take into account the heat exchanges between water and atmosphere through a direct programming in the BORD3D ubroutine. An examplary exchange with a constant temperature atmosphere and a constant salinity sea is given as standard.

5.5.3. THE PRESSURE

The influence of air pressure is talen into account from the moment when the keyword AIR PRESSURE is set to YES (the default value is NO). The value of that pressure is directly set in the METEO subroutine. By default, the latter initializes a pressure of 105 pascals (1 atmosphere) over the whole domain.

5.6. OTHER PHYSICAL PARAMETERS

Upon the modelling of wide areas, the influence of the Coriolis force of inertia has to be talen into account, what is done by activating the logical keyword CORIOLIS (which is set to NO by default). In such a case, the value of the Coriolis coefficient (refer to the Principle Note) is yielded by the keyword CORIOLIS COEFFICENT. The latter should be computed according to latitude λ through the formula:

- 2ω sin (λ) where ω is the Earth's rotational velocity of 7.27 x 10^{-5} rad/s and λ is the average latitude of the model.

TELEMAC-3D additionally offers the opportunity to set the gravity acceleration (keyword GRAVITY ACCELERATION which, by default, is set to 9.81m/s^2.

6. DIGITAL SETUP OF THE COMPUTATION

The digital setup is compartively common to a hydrodynamic computation alone or with a tracer.. Thus, in the following sections of this chapter, the digital parameters as applied to the solution of a tracer equation are integrated into the hydrodynamic parameters.

6.1. GENERAL SETUP

TELEMAC-3D solves the Navier-Stokes equations in several stages, possibly through the three stages of the fractional step method. The first stage consists in finding out the advected velocity components by only solving the convection terms of the momentum equations. The second stage computes, from the advected velocities, the new velocity components by taking into account both diffusion and source terms of the momentum equations. These two solutions enable to get an intermediate velocity field. The third stage makes it possible to compute the water depth from the vertical integration of the continuity equation and momentum equations only including the pressurecontinuity terms.

The user can activate or disactivate, either globally or individually, some of these stages.

6.1.1. ADVECTION STEP

Whether the convection terms will be considered or not will be determined by means of the logical keyword ADVECTION STEP (default value YES). However, even though that keyword is set to YES, then some advection terms can be disactivated by means of the following complete keywords (0 value = “ADVECTION STEP”):

- SCHEME FOR ADVECTION OF VELOCITIES : for advection of velocities,
- SCHEME FOR ADVECTION OF DEPTH: for taking the advection of depth into account,
- SCHEME FOR AVECTION OF K-EPSILON : for the advection of power and turbulent dissipation,
- SCHEME FOR ADVECTION OF TRACERS : pour la convection des traceurs.

6.1.2. DIFFUSION STEP

Whether the diffusion steps are taken into account or not is established for each variable by the following keywords:

- SCHEME FOR DIFFUSION OF VELOCITIES
- SCHEME FOR DIFFUSION OF TRACERS
- SCHEME FOR DIFFUSION OF K-EPSILON

The 0 value at each keyword cancels the diffusion, whereas the 1 value leads to the implicit calculation of diffusion.

6.1.3. PROPAGATION STEP

The keyword OPTION FOR THE HYDROSTATIC STEP (default value 1: typical, value 2 : wave equation) is for modifying the hydrostatic step by solving the diffusion and the continuity equation in one step. As with the 2nd integers in the keyword OPTIONS FOR TELEMAC-2D, option 2 comes down to deleting the velocity coupling of the continuity equation in the linear system.

The velocity and water depth propagation processes are taken into account by the logical keyword

PROPAGATION STEP (default value YES). That step is currently to be conducted.

When the keyword « OPTION FOR THE HYDROSTATIC STEP » is set to 1, the keyword OPTIONS FOR TELEMAC-2D is associated with that propagation step. A couple of integers are associated with the keyword:

- The first integer informs about those terms which are taken into account in the Saint-Venant equation. The option 1 (default value) results in considering the system without the advection and diffusion terms (terms being solved during the previous steps). The option 2 considers again the whole Saint-Venant system (inclusive of diffusion and advection).
- The second integer defines the matricial writing of the linear system to be solved. Option 1(default value) is a standard writing of that system. Option 2 reflects a simplified form of the

system coupling depth and velocities (writing close to the solution of a wave equation). The latter option is inconsistent with the SUPG method for the advection of velocities.

The propagation step can be linearized by activating the keyword LINEARIZED PROPAGATION particularly when one conducts a case study for which an analytical solution in the linearized case is available. It is then necessary to set set the water depth around which the linearization is performed by means of the keyword MEAN DEPTH FOR LINEARIZATION (default value 0).

With the « wave equation » option, the keyword « FREE SURFACE COMPATIBILITY GRADIENT » can be used. A lower than 1 value makes it possible to delete the spurious fluctuations of the free surface, but slightly alters the consistency between the water depth and the velocities in the continuity equation.

6.2. THE ADVECTION SCHEME

The procedure for taking the advection terms into account is individualized for each of the variables liable to be processed. It has previously been explained that the zero option corresponds to a deactivation of the term.

6.2.1. ADVECTION OF THE THREE-DIMENSIONAL VARIABLES

The advection schemes of the three-dimensional variables (i.e. all the variables but the water depth) are (for forther details about these options, the reader shall refer to the Theoretical Note):

- 0 : Desactivation.
- 1 : Method of characteristics. That method involves mutually independent advection and diffusion steps. The method consists in writing that the value of the variable being searched is equal to the value of the same variable in the previous instant by tracing back the path travelled during the time step.
- 2 : Explicit scheme + SUPG (Streamline Upwind Petrov Galerkin). That method uses test functions which are deformed in the direction of the current for the variational method.
- 3 : Deleted
- 4 : Explicit scheme + MURD (Multidimensional Upwind Residual Distribution) N scheme.
- 5 : Explicit scheme + MURD PSI scheme.

The latter two schemes are primarily recommended for the tracers, since they are advantageous in being conservative and monotonic, i.e. they do not generate any numerical oscillation. On the other hand, they are more diffusive than SUPG. In that respect, scheme 5 in an improvement to scheme 4, being less diffusive perpendicularly to the flow but quite obviously somewhat more computation time-consuming. Nevertheless, it is still globally less time-consuming than SUPG.

Schemes 4 and 5, through the automatic calculation of time sub-steps, do not exhibit any restriction in Courant number.

The default value for both velocity and K-Epsilon is 1. That value is advisable, since it is satisfactory in many instances and is by far the most rapid. On the contrary, the default value for the tracers is 5. It is the most reliable scheme, since the “mass” conservation of the active tracers is a frequently essential point in TELEMAC-3D.

According to the schemes being used, the mass conservation can be improved by performing subiterations. That consists in a reactualization, for one given time step, of both advective field and propagative field during several sub-iterations. Upon the first sub-iteration, the field of velocities is yielded by the results achieved during the previous time steps. Through that procedure, the nonlinearities can be taken into account in a better way and the mass conservation can be significantly improved in the cases of schemes 2 and 3. The number of sub-iterations is set by the keyword NUMBER OF SUB-ITERATIONS FOR NON-LINEARITIES the default value of which is 1 (also refer to the Theoretical Note).

6.3. SPECIFIC PARAMETERS IN THE NON-HYDROSTATIC VERSION

The application of the software NON-HYDROSTATIC VERSION requires that complementary keywords be defined.

An equation for vertical velocity is initially solved in a same way as the U and V components. That equation is only written with the hydrostatic pressure which, under that assumption, is cancelled out with the gravity term. The convection scheme in that equation is identical to that being chosen for U and V (keyword SCHEME FOR ADVECTION OF VELOCITIES). The solution of the linear system (for the diffusion step integrating the advection terms or not) is managed by the various following keywords:

- SOLVER FOR VERTICAL VELOCITY refer to para. 6.5.1 below.
- MAXIMUM NUMBER OF ITERATIONS FOR VERTICAL VELOCITY refer to para. 6.5.2 below.
- ACCURACY FOR VERTICAL VELOCITY refer to para 6.5.2 below.
- PRECONDITIONING FOR VERTICAL VELOCITY refer to para. 6.5.3 below.

Afterwards, TELEMAC-3D solves a Poisson equation for the dynamic pressure. The dynamic pressure gradient plays the part of a correction providing the required zero divergence on velocity. The solution of the linear system in that equation is managed by the following keywords:

- SOLVER FOR PPE refer to para. 6.5.1 below.
- MAXIMUM NUMBER OF ITERATIONS FOR PPE refer to para. 6.5.2 below.
- ACCURACY FOR PPE refer to para. 6.5.2 below.
- OPTION OF SOLVER FOR PPE refer to para. 6.5.1 below.
- PRECONDITIONING FOR PPE refer to para. 6.5.3 below.

Once that pressure is computed, the velocities are updated by the dynamic pressure gradient will is going to ensure the zero divergence condition.

That step can be solved through the so-called “consistent projection” method (CONSISTENT PROJECTION = YES (default value = NO) which solves 3 linear equations (one for each velocity component). The solution of these linear systems is managed by the following keywords:

- SOLVER FOR PROJECTION refer to para. 6.5.1 below.

- MAXIMUM NUMBER OF D'ITERATIONS FOR PROJECTION refer to para. 6.5.2 below.
- ACCURACY FOR PROJECTION refer to para. 6.5.2 below.
- OPTION OF SOLVEUR FOR PROJECTION refer to para. 6.5.1 below.
- PRECONDITIONING FOR PROJECTION refer to para. 6.5.3 below.

Updating that “solenoidal” velocity through the default method (CONSISTENT PROJECTI = NO) does not require that a linear system be solved.

6.4. IMPLICITATION

Apart from the terms of the time derivative, the unknowns f (the velocity and water depth components) can be considered in both extreme instants t^n (the equation is then referred to as explicit) or t^{n+1} (the equation is then referred to as implicit). Strictly speaking, and for a 2-order solution in time, an approach consists in considering the terms in the intermediate instant ($f^n + f^{n+1}$) / 2 . Practically, the latter approach is unstable and it becomes necessary to define an implicitation coefficient for which the unknowns are actually discretized in time in the following form:

$\theta f^{n+1} + (1-\theta) f^n$

The implicitation coefficients are theoretically always higher than 0.5 (0.55 or 0.6 will generally yield good results).

The user can use the keyword IMPLICITATION FOR VELOCITIES (default value: 1) which defines the value of the θ_u coefficient for the velocity components. The keyword IMPLICITATION FOR DEPTH (default value: 0.55) is provided for setting the value of “propagation height” multiplying coefficient θ_h . Lastly, in order to make the construction of the various numerical schemes more versatile, a diffusion-specific coefficient θ_u^d is provided (keyword IMPLICITATION FOR DIFFUSION (default value: 1) and can be different from θ_u .

6.5. SOLUTION OF THE LINEAR SYSTEMS

Both discretization and variational formulation of the equations lead to a linear system which now has to be solved. The direct solution methods are not suitable and are overly time extensive as soon as there are many unknowns. Thus, the only way consists in solving the linear systems by means of iterative solvers.

6.5.1. SOLVERS

According to the relevant numerical parameters, various linear systems are liable to be solved. The solver being used for solving one of these systems can be selected by the user through the following keywords:

- SOLVER FOR DIFFUSION OF VELOCITIES (default value: 1),
- SOLVER FOR PROPAGATION (default value: 7),
- SOLVER FOR VERTICAL VELOCITY (default value: 1),
- SOLVER FOR PPE (default value: 1),
- SOLVER FOR PROJECTION (default value: 6),
- SOLVER FOR DIFFUSION OF TRACERS (default value: 1),
- SOLVER FOR DIFFUSION OF K-EPSILON (default value: 1).

Each of these keywords can assume a value ranging from 1 to 7, which values correspond to the following possibilities:

- 1: conjugate gradient method.
- 2: conjugate residual method.
- 3: conjugate gradient method on a normal equation.
- 4: minimum error method.
- 5: square conjugate gradient method.
- 6: CGSTAB (stabilized conjugate gradient) method.
- 7: GMRES (Generalised Minimum RESidual) method.

The GMRES method especially suits the unproperly conditioned systems. The latter method requires that the dimension of the Krylov space be defined. That parameter is set by means of the following keywords:

- OPTION OF SOLVER FOR DIFFUSION OF VELOCITIES,
- OPTION OF SOLVER FOR PROPAGATION,
- OPTION OF SOLVER FOR PPE,
- OPTION OF SOLVER FOR PROJECTION,
- OPTION OF SOLVER FOR DIFFUSION OF TRACERS,
- OPTION OF SOLVER FOR DIFFUSION OF K-EPSILON.

The default values are set to 3. The larger that parameter is, the higher the memory requirements and the number of matrix-vector products per iteration (and consequently the computational time as well) are, but the better the convergence is.

6.5.2. ACCURACIES

The principle of iterative methods consists in get gradually closer to the exact solution during the iterations. The systems to be solved imply a relative accuracy within a range from 10-4 to 10-10 with a restricted number of iterations. Both accuracy and maximum number of iterations should be set for each system.

Accuracy is specified by the following keywords:

- ACCURACY FOR DIFFUSION OF VELOCITIES (default value: 1.E-5),
- ACCURACY FOR PROPAGATION (default value: 1.E-6),
- ACCURACY FOR (default value: 1.E-4),
- ACCURACY FOR PROJECTION (default value: 1.E-6),
- ACCURACY FOR VERTICAL VELOCITY (default value: 1.E-6),
- ACCURACY FOOR DIFFUSION OF TRACERS (default value: 1.E-6),
- ACCURACY FOR DIFFUSION OF K-EPSILON (default value: 1.E-6).

The maximum number of iterations is specified by the following keywords:

- MAXIMUM NUMBER OF ITERATIONS FOR DIFFUSION OF VELOCITIES (default value: 60),
- MAXIMUM NUMBER OF ITERATIONS FOR PROPAGATION (default value: 200),
- MAXIMUM NUMBER ITERATIONS FOR PPE (default value: 100),
- MAXIMUM NUMBER OF ITERATIONS FOR PROJECTION (default value: 100),
- MAXIMUM NUMBER OF ITERATIONS FOR VERTICAL VELOCITY (default value: 100),
- MAXIMUM NUMBER OF ITERATIONS FOR DIFFUSION OF TRACERS (default value: 60),
- MAXIMUM NUMBER OF ITERATIONS FOR DIFFUSION OF K-EPSILON (default value: 60).

The user automatically gets information about the solvers upon each listing printout. The information provided in the listing can be of two types:

- Either the treatment converged before reaching the maximum allowable number of iterations, and then TELEMAC-3D will provide the number of actually performed iterations, as well as the achieved accuracy.
- Or the treatment did not converge early enough. TELEMAC-3D will then provide the message “MAXIMUM NUMBER OF ITERATIONS IS REACHED” and give the actually achieved accuracy. In some cases, and if the maximum number of iterations is already set to a high value (for instance over 100), convergence van then be improved by reducing the time step or, quite often, by improving the mesh.

6.5.3. PRECONDITIONINGS

The iterative methods are sensitive to the “conditioning” of matrices, so that a complementary preconditioning is necessary in order to reduce the number of iterations for getting a prescribed accuracy.

TELEMAC-3D offers several opportunities for preconditioning. The selection is made by means of the following keywords:

- PRECONDITIONING FOR DIFFUSION OF VELOCITIES,
- PRECONDITIONING FOR PROPAGATION,
- PRECONDITIONING FOR PPE,
- PRECONDITIONING FOR PROJECTION,
- PRECONDITIONING FOR VERTICAL VELOCITY,
- PRECONDITIONING FOR DIFFUSION OF TRACERS,
- PRECONDITIONING FOR DIFFUSION OF K-EPSILON.

The available options are:

- 0: no preconditioning,
- 2: diagonal preconditioning,
- 3: diagonal preconditioning with the condensed matrix,
- 7: Crout preconditioning per element,
- 14 : cumulated diagonal preconditioning and Crout preconditioning per element,
- 17 : preconditioning through direct solution along each vertical direction,
- 21 : cumulated diagonal preconditioning with the condensed matrix and Crout preconditioning per element.

The default value is 2 for all the preconditionings. Some preconditionings can be cumulated, namely the diagonal preconditionings with other ones. Since the basic values are prime numbers, a couple of preconditionings are cumulated by allotting to the keyword the value of the product of the two preconditionings which one wants to cumulate (e.g. numbers 14 and 21).

6.6. TIDAL FLATS

TELEMAC-3D offers several treatment options as regards the tidal areas.

First, if the user has ascertained that his/her model has no tidal area throughout the simulation, the processing of such areas can be disactivated by setting the keywords TIDAL FLATS to NO (the default value is YES). That option makes it possible to save computational time (by deleting the tidal flat testing).

The tidal plats can be treated in two different ways:

- In the first case, the equations are treated all over the domain and in a thorough way. The tidal areas are detected and such terms as the free surface gradient (in the absence of water, the free surface gradient becomes the bottom gradient and generates spurious motive terms) are corrected in them.
- In the second case, the tidal areas are withdrawn from the computation. The exposed elements are always part of the mesh, but all their contributions to the computations are cancelled by a so-called “masking” array. Thus, the data structure and the computations remain formally unchanged, to within the masking coefficient. That method, however, raises issues as regards the mass conservation and the exposure and coverage dynamics.

The treatment will be selected by means of the keyword OPTION FOR THE TREATMENT OF TIDAL FLATS which can be set either to 1 or 2, the default value being 1.

The keyword MINIMUM VALUE FOR DEPTH the default value of which is -1000 (zero in the case of the second option for the treatment of tidal flats) enables to set the threshold below which a correction of the water depth is undertaken (for example in order to delete negative values. It should be remembered, however, that such an option will bring about a change in the water mass conservation.

The following three keywords are for setting, after coverage, the value of the variable which has been masked:

- TREATMENT ON TIDAL FLATS FOR VELOCITIES
- TREATMENT ON TIDAL FLATS FOR TRACERS
- TREATMENT ON TIDAL FLATS FOR K-EPSILON

The available options for these keywords are:

- 0: that option corresponds to a setting to zero of the variable on the element,
- 1: that option sets to its prior-to-masking value.

6.7. HYDROSTATIC INCONSISTENCIES

Hydrostatic inconsistencies (linked to the truncature errors in the computation of the buoyancy terms) are liable to occur on the nearly-zero volume prisms. The keyword HYDROSTATIC INCONSISTENCY FILTER is provided for the forces caused by the spurious horizontal pressure gradients and the various diffusion coefficients on those prisms at least one of the lower base nodes of which has a higher elevation than one of the upper base nodes.

6.8. OTHER PARAMETERS

6.8.1. 2D/3D TIME STEP SPLITTING

The time steps of the 2D stages can be split off the 3D stages when solving the equations. That option is activated by the keyword: RATIO OF 2D AND 3D TIME STEPS. In that case, the value of the 3D time step is an integer multiple of the 2D time step value. Thanks to that option, the computational time can be shortened through a less frequent solution of the three-dimensional equations. It is not advisable, however, because it does no longer properly encure the mass conservation.

6.8.2. MASS-LUMPING

Upon the solution of the linearized system, TELEMAC-3D makes it possible to perform a masslumping on the mass matrices. That procedure consists in partly or wholly returning the mass matrix to its diagonal and enables to substantially shorten the computational times. The resulting solutions, however, become smoothed, except for in steady flow conditions in which they are unchanged. The mass-lumping rate is set by means of the keywords MASS-LUMPING FOR DEPTH, MASS-LUMPING FOR VELOCITIES and MASS-LUMPING FOR DIFFUSION. Value 1 means maximum mass-lumping (the mass matrices are diagonal), value 0 (default value) corresponds to the normal treatment without any mass-lumping. For further details, the reader shall refer to the TELEMAC-3D Theoretical Note.

6.8.3. CONVERGENCE AID

Another way to speed up the system convergence when solving the propagation step consists in acting upon the initial solution rather than the matrix proper. To that purpose, the inital value being set for h (actually, the unknown hn+1 is replaced by the incrementationδh = hn+1 − hn) is modified at the beginning of computation, the user can take action at the keyword INITIAL GUESS FOR DEPTH which can assume the following values:

- 0: the initial value of δh = hn+1 − hn is zero.
- 1: the initial value of δh is equal to the value of δh at the previous time step (default value).
- 2: δh = 2hn −δhn−1 where δhn is the value of δh at the previous time step, and δh is the value of δhn−1 two time steps earlier. It is actually an extrapolation.

6.8.4. MATRIX STORAGE

TELEMAC-3D provides a couple of procedures for storing the various matrices he has to handle, namely the conventional EBP (Element By Element) method and the segment-wise storage. The latter procedure is faster (by approx 20%) in most cases. The choice among these two types of storage is made by means of the keyword MATRIX STORAGE which can assume the following values:

- 1: conventional EBE method (default value).
- 3: segment-wise storage

7. TRACER TRANSPORT

The TELEMAC-3D software makes it possible to take into account the transport of passive or active tracers (i.e. the presence of which affects hydrodynamics or not), being either conservative or not.

This chapter discloses the tracer transport features.

7.1. GENERAL SETUP

The number of tracers is defined by the keywords NUMBER OF TRACERS. It that number is set to zero (default value), then the tracers will not be taken into account by TELEMAC-3D.

In addition to the number of tracers, the user should enter the NAMES OF TRACERS. A tracer name should written with 32 characters (16 for the name and 16 for the unit).

NUMBER OF TRACERS : 2

NAMES OF TRACERS : 'TEMPERATURE °C' ; 'SALINITY'

7.1.1. PRESCRIBING THE INITIAL CONDITIONS

If the initial values of tracers are constant all over the domain, just insert, into the steering file, the keyword INITIAL VALUES OF TRACERS to which the desired value in allotted.

In more complex cases, an action shall be taken directly at the CONDIM routine, somewhat as described in the paragraph dealing with the initial hydrodynamic conditions.

When resuming a computation, the initial condition of tracers corresponds to the condition of the last time step which was stored into the resumption file. The tracer management sequence order during the previous computation shall then be well known and the same sequence order shall be followed in the computational suite in order to prevent confusion. It the resumption file includes no information about the tracer, TELEMAC-3D will then use the value as set by the keyword INITIAL VALUES OF TRACERS.

7.1.2. PRESCRIBING THE BOUNDARY CONDITIONS

The tracer boundary conditions are prescribed according to the same principle as the hydrodynamic boundary conditions.

The boundary condition type will be yielded by the value of LITBOR in the boundary conditions file.

In case of an entering liquid boundary with a prescribed tracer (the value of LITBOR is 5), then the tracer value can be yielded in various ways:

- If that value is constant both along the boundary and in time, then it is provided in the steering file by means of the keyword PRESCRIBED TRACERS VALUES. It is an array of reals for managing several boundaries (100 at the very most), the numbering principle being the same as in the case of the hydrodynamic boundary conditions. The values as specified by that keyword cancel the values being read from the boundary conditions file.
- the value is constant in time but varies along the boundary, it will be yielded directly by the TBOR variable in the BOUNDARY CONDITIONS FILE.
- the value is constant along the boundary but varies in time, the user may either use the LIQUID BOUNDARIES FILE or take action at the function TR3. The latter will be programmed somewhat like the functions VIT3, Q3 and SL3.

```
T SL(1) TR(6) TR(7)

s m °C °C

0 0.47 24.7 38.0

1040400 0.57 28.0 36.7
```

In the above example of a liquid boundaries file, the indices IRANK of the tracers are arranged as follows: IRANK=ITRAC+(I-1)*NTRAC where ITRAC denotes the tracer number, I - the boundary index and NTRAC – the total number of tracers. In the discussed example (computation with 5 tracers), TR(6) corresponds to the first tracer being prescribed along the 2nd boundary and TR(7) – to the 2nd tracer along the 2nd boundary.

- If the value varies in both time and space, the user should then directly take action in the BORD3D subroutine, at the part regarding the tracer.

The keyword TREATMENT OF FLUXES AT THE BOUNDARIES enables, during the convection step (with the SUPG, PSI and N schemes); to set a priority among the tracer flux across the boundary and tracer value at that wall. Option 2 (“Priority to fluxes”) will then induce a change in the tracer prescribed value, but will bring about a good assessment of the “mass” of tracer passing across the boundary. On the other hand, Option 1 (“Priority to prescribed values”) sets the tracer value without checking the fluxes.

7.2. PHYSICAL SETUP

7.2.1. ACTIVE TRACERS

The active tracers affect the flow through the hydrostatic pressure gradient term. As a rule, indeed, the pressure is written as:

$$p = p_h + p_d = \rho g(Z_s - z) + \rho_0 g \int_z^{Z_s} \frac{\Delta\rho}{\rho_0} \mathrm{d}z + p_d$$

where p_h and p_d denote the hydrostatic pressure and the dynamic pressure, respectively

Thus, two elements should be defined, namely: ρ_0 and $\frac{\Delta\rho}{\rho_0}$.

The term 0 ρ is defined by the keyword DENSITY FOR STANDARD VALUE. The default value is 1025, which corresponds to sea (ocean) water.

The second term, which operates in the buoyancy source terms, directly depends on the values of the active tracers and is defined by the keyword DENSITY LAW.

The available values for that keyword DENSITY LAW are:

- 0: no interaction with the tracers,
- 1: variation of density according to temperature,
- 2: variation of density according to salinity,
- 3: variation of density according to temperature and salinity,
- 4: variation as a function of the spatial expansion coefficients.

With the 1-3 options, the variations are given by the law as defined in TELEMAC-3D. In such a case, the name of the salinity tracer (expressed in kg/cu.m) shall necessarily begin with SALINI and the name of the temperature tracer (in °C) shall begin with TEMPER. With option 4, the term $\frac{\Delta\rho}{\rho_0}$ is described by a linear function of the Ti tracer of the type:

$$\frac{\Delta\rho}{\rho_0} = -\sum_i \beta_i \left(T_i - T_i^0\right)_i$$

The β_i coefficients (spatial expansion coefficients) are set by the values of the keyword BETA EXPANSION COEFFICIENT FOR TRACERS. They can be either positive (temperature) or negative (salinity, suspended sediment). The values T_i^0 are defined by the values of the keyword STANDARD VALUES FOR TRACERS.

The user shall enter the expansion coefficients, the standard values and the tracer names in the same sequence order in order get the proper matching (for each tracer) among these various parameters.

7.2.2. PUNCTUAL SOURCE TERMS

For each source, the user shall enter the tracer value at the sources by means of the keyword VALUE OF THE TRACERS AT THE SOURCES. Thus, it is an array of reals specifying the concentration of tracers at the source. The writing convention is as follows: source value 1 of tracer 1; source value 1 of tracer 2; … ; source value 1 of tracer n; source value 2 of tracer 1; … ; source value 2 of tracer n; etc.

7.2.3. GENERAL SOURCE TERMS

If one wants to take the tracer generation or disappearance source terms into account, then that has to be inserted into the SOURCE_TRAC subroutine.

7.3. NUMERICAL SETUP

As with hydrodynamics, the advection schemes SCHEME FOR ADVECTION OF TRACERS, (refer to para. 6.1.1) and the diffusion schemes SCHEME FOR DIFFUSION OF TRACERS (refer to para. 6.1.2) can be modified by their default values.

8. DROGUES AND LAGRANGIAN DERIVATIVES

Upon a hydrodynamic simulation, TELEMAC-3D offers an opportunity to follow the paths of a number of particles (drogues) which are released into the fluid from discharge points. The result is provided in the form of a SELAFIN-formatted file which contains the various positions of the drogues as illustrated by a degenerated mesh.

8.1. CONFIGURATION OF SIMULATION

Three pieces of information should be entered into the steering file. First, the user should mention the number of drogues by means of the keywords NUMBER OF DROGUES the default value of which is 0. Second, the user should enter the name of the file into which TELEMAC-3D will store the successive positions of the drogues. To that purpose, he/she has to use the keyword BINARY RESULT FILE (Therefore, the drogues cannot be used during one simulation while storing results into a binary file other than the standard results file). Lastly, the user can configure the printout period within that file by means of the keyword PRINTOUT PERIOD FOR DROGUES (default value = 1. That value is expressed in a number of time steps and is quite independent from of the printout period of the other results in TELEMAC-3D.

After inserting the FLOT3D subroutine into his/her Fortran file, the user should amend it so that, for each drogue, the release time step and the end-of-tracking time step, as well as the co-ordinates of the release point, will appear in it. By default, all the drogues are released at the beginning of the simulation, and the end-of-tracking time corresponds to the end-of-simulation time. If the coordinates of the release point occur outside the domain, the execution is interrupted with an error message. Besides, if, during the simulation, a drogue leaves the domain, the tracking of it is of course interrupted, but the earlier path remains available in the results file.

The example below illustrates the programming of the steering file and the FLOT3D subroutine in the case of two drogues being release at different times.

```
NUMBER OF DROGUES = 2
BINARY RESULTS FILE = './drogues'
PRINTOUT PERIOD FOR DROGUES = 10
DEBFLO(1) = 1 The first drogue is released at the beginning
DEBFLO(2) = 100 of the simulation until the 400th time step.
FINFLO(1) = 400 The second drogue is released at the 100th
FINFLO(2) = NIT time step until the end.
XFLOT(1,1) = 321.
XFLOT(1,2) = 351. The co-ordinates of the two drogues
YFLOT(1,1) = 37. at the release time are specified
YFLOT(1,2) = 95.
```

8.2. EXPLOITATION OF RESULTS

The results are stored into the binary file during the simulation. That file is in the SELAFIN format and can then be read by means of RUBENS.

The drogue positions are stored in the form of a degenerated mesh. Consequently, and in the absence of new developments at the post-processor level, during the preparation of the RUBENS project, a number of error messages are displayed.

The positions are illustrated by means of de “Mesh”-typed graph. Each position is shown in the form of a node. The successive positions of a drogue are identified though the node numbering. The numbering process takes place as follows: at the beginning of the file, all the positions of drogue 1, then all the positions of drogue 2, etc.

If, for example, a simulation is performed over 10 time steps with 3 drogues, then one will get the following numbering:

- Positions 1-10: successive position of drogue 1.
- Positions 11-20: successive positions of drogue 2.
- Positions 21-30: successive positions of drogue 3.

9. PARALLELISM

Upon simulations requiring a high computational power, it can be advisable to run the computations in multi-processor machines, or also in clusters of workstations. TELEMAC-3D is available in a parallel version in order to take advantage of that kind of computational architecture.

The TELEMAC-ED parallel version uses the MPI library which has to be installed beforehand to be implementable. The interface between TELEMAC-3D and that MPI library is achieved through the parallel library which is common to all the TELEMAC system modules. That library is replaced by the paravoid library in the case of a machine running a non-parallel version of the TELEMAC system.

Lots of pieces of information about the implementation of the parallel version can be found in the system’s installation literature

The user shall initially specify the number of processors being used by means of the keyword PARALEL PROCESSORS. That integer type keyword can assume the following values:

- 0 : Implementation of the conventional TELEMAC-3D version,
- 1 : Implementation of the parallel TELEMAC-3D version in a processor,
- 2 … : Implementation of the parallel TELEMAC-3D version using the specified number of processors.

The parallel machines are configured through a simple file (refer to the system installation note).

The method of characteristics is not available in the parallel version (use SUPG or PSI instead).