TELEMAC modelling system

2D hydrodynamics

TELEMAC-2D software

Version 6.0

USER MANUAL

October 2010

This manual has been updated for version 6.0 by Pierre Lang, Ingerop
(pierre.lang@ingerop.com)

The information given in this manual is subject to revision without notice. EDF-R&D disclaims any responsibility for or in relation to the contents hereof.

The TELEMAC system is the property of EDF-R&D.

****

BLUEKENUE is the property of the Canadian Hydraulics Centre, Ottawa, Ontario, Canada

http://www.nrc-cnrc.gc.ca/eng/ibp/chc/software/kenue/blue-kenue.html

DELWAQ is the property of DELTARES, Delft, The Netherlands
http://delftsoftware.wldelft.nl/

EVOLUTIONS OF THE DOCUMENT

DATE	AUTHOR	EVOLUTION
10/2010	Pierre LANG *pierre.lang@ingerop.com*	General update for release 6.0

Typing conventions used in this manual

Keywords are written in *UPPER CASE ITALICS*.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. PRESENTATION OF TELEMAC-2D SOFTWARE

The TELEMAC-2D code solves depth-averaged free surface flow equations as derived first by Barré de Saint-Venant in 1871. The main results at each node of the computational mesh are the depth of water and the depth-averaged velocity components. The main application of TELEMAC-2D is in free-surface maritime or river hydraulics and the program is able to take into account the following phenomena:

- Propagation of long waves, including non-linear effects
- Friction on the bed
- The effect of the Coriolis force
- The effects of meteorological phenomena such as atmospheric pressure and wind
- Turbulence
- Supercritical and subcritical flows
- Influence of horizontal temperature and salinity gradients on density
- Cartesian or spherical coordinates for large domains
- Dry areas in the computational field: tidal flats and flood-plains
- Entrainment and diffusion of a tracer by currents, including creation and decay terms
- Particle tracking and computation of Lagrangian drifts
- Treatment of singularities: weirs, dikes, culverts, etc.
- Inclusion of the drag forces created by vertical structures
- Inclusion of porosity phenomena
- Inclusion of wave-induced currents (by link-ups with the ARTEMIS and TOMAWAC modules).
- Coupling with sediment transport

The software has many fields of application. In the maritime sphere, particular mention may be made of the sizing of port structures, the study of the effects of building submersible dikes or dredging, the impact of waste discharged from a coastal outfall or the study of thermal plumes. In river applications, mention may also be made of studies relating to the impact of construction works (bridges, weirs, groynes), dam breaks, flooding or the transport of decaying or non-decaying tracers. TELEMAC-2D has also been used for a number of special applications, such as the bursting of industrial reservoirs, avalanches falling into a reservoir, etc.

TELEMAC-2D was developed by the National Hydraulics and Environment Laboratory (Laboratoire National d'Hydraulique et Environnement - LNHE) of the Research and Development Directorate of the French Electricity Board (EDF-R&D), in collaboration with other research institutes. Like previous versions of the program, version 6.0 complies with EDF-'s Quality Assurance procedures for scientific and technical programs. This sets out rules for developing and checking product quality at all stages. In particular, a program covered by Quality Assurance procedures is accompanied by a Validation Document that describes the field of use of the software and a set of test cases. This document can be used to determine the performance and limitations of the software and define its field of application. The test cases are also used for developing the software and are checked each time new versions are produced.

1.2. PROGRAMMING BY THE USER

Users may wish to program particular functions of a simulation module that are not provided for in the standard version of the TELEMAC system. This can be done in particular by modifying specific subroutines called user subroutines (see appendix 2). These subroutines offer an implementation that can be modified (provided that the user has a minimum knowledge of Fortran and with the help of the guide for programming in the Telemac system).

The following procedure should be followed:

- Recover the standard version of the subroutine provided with the system, and copy it into the working directory.
- Modify the subroutines according to the model you wish to build.
- Link up the set of subroutines into a single FORTRAN file that will be compiled during the TELEMAC-2D start procedure.

During this programming phase, users must access the various software variables. By using the structures of FORTRAN 90 gathered into a "module" type component, access is possible from any subroutine.

The set of data structures is gathered in FORTRAN files referred to as modules. In the case of TELEMAC-2D, the file is called DECLARATION_TELEMAC2D and is provided with the software. To access TELEMAC-2D data, simply insert the command USE DECLARATIONS_TELEMAC2D at the beginning of the subroutine. It is also necessary to add the command USE BIEF.

Almost all the arrays used by TELEMAC-2D are declared in the form of a structure with pointers. For example, access to the water depth variable is in the form H%R where the %R indicates that a pointer of real type is being used. If the pointer is of integer type, the %R is replaced by a %I. However, to avoid having to handle too many %R and %I, a number of aliases have been defined, such as for example the variables NPOIN, NELEM, NELMAX and NPTFR.

This page is intentionally left blank

2. THEORETICAL ASPECTS

The TELEMAC-2D code solves the following four hydrodynamic equations simultaneously:

$$\frac{\partial h}{\partial t} + u \cdot \vec{\nabla}(h) + hdiv(\vec{u}) = S_h \qquad \text{continuity}$$

$$\frac{\partial u}{\partial t} + \vec{u} \cdot \vec{\nabla}(u) = -g\frac{\partial Z}{\partial x} + S_x + \frac{1}{h}div(h\nu_t\vec{\nabla}u) \qquad \text{momentum along x}$$

$$\frac{\partial v}{\partial t} + \vec{u} \cdot \vec{\nabla}(v) = -g\frac{\partial Z}{\partial y} + S_y + \frac{1}{h}div(h\nu_t\vec{\nabla}v) \qquad \text{momentum along y}$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \vec{\nabla}(T) = S_T + \frac{1}{h}div(h\nu_T\vec{\nabla}T) \qquad \text{tracer conservation}$$

in which:

h	(m)	depth of water
u,v	(m/s)	velocity components
T	(g/l or °C)	passive (non-buoyant) tracer
g	(m/s2)	gravity acceleration
νt, νT	(m2/s)	momentum and tracer diffusion coefficients
Z	(m)	free surface elevation
t	(s)	time
x,y	(m)	horizontal space coordinates
S_h	(m/s)	source or sink of fluid
S_x,S_y	(m/s^2)	source or sink terms in dynamic equations
S_T	(g/l/s)	source or sink of tracer

h, u, v and T are the unknowns.

The equations are given here in Cartesian coordinates. They can also be processed using spherical coordinates.

Sx and Sy (m/s2) are source terms representing the wind, Coriolis force, bottom friction, a source or a sink of momentum within the domain. The different terms of these equations are processed in one or more steps (in the case of advection by the method of characteristics):

- advection of h, u, v and T,

- propagation, diffusion and source terms of the dynamic equations,
- diffusion and source terms of the tracer transport equation.

Any of these steps can be skipped, and in this case different equations are solved. In addition, each of the variables h, u, v and T may be advected separately. In this way it is possible, for example, to solve a tracer advection and diffusion equation using a fixed advecting velocity field.

Turbulent viscosity may be given by the user or determined by a model simulating the transport of turbulent quantities k (turbulent kinetic energy) and Epsilon (turbulent dissipation), for which the equations are the following:

$$\frac{\partial k}{\partial t} + \vec{u}\cdot\vec{\nabla}(k) = \frac{1}{h} div(h \frac{\nu_t}{\sigma_k} \vec{\nabla} k) + P - \varepsilon + P_{kv}$$

$$\frac{\partial \varepsilon}{\partial t} + \vec{u}\cdot\vec{\nabla}(\varepsilon) = \frac{1}{h} div(h \frac{\nu_t}{\sigma_\varepsilon} \vec{\nabla} \varepsilon) + \frac{\varepsilon}{k}(c_{1\varepsilon} P - c_{2\varepsilon}\varepsilon) + P_{\varepsilon v}$$

The right-hand side terms of these equations represent the production and destruction of turbulent quantities (energy and dissipation).

A complete description of the theory is given in the following book : “Hydrodynamics of free surface flows”, by Jean-Michel Hervouet (Wiley, 2007).

3. INPUTS AND OUTPUTS

3.1. PRELIMINARY REMARKS

A set of files is used by TELEMAC-2D as input or output. Some files are optional.

The input files are the following:

- The steering file (**obligatory**), containing the configuration of the computation,
- The geometry file (**obligatory**), containing the mesh,
- The boundary conditions file (**obligatory**), containing the description of the type of each boundary,
- The FORTRAN file, containing the specific programming,
- The bottom topography file, containing the elevation of the bottom. Generally, the bottom information is already available in the geometry file and the bottom topography file is no longer useful.
- The liquid boundary file, containing information about the prescribed values at the open boundaries (elevation, flowrate …).
- The previous computation file, which can give the initial state of the computation. This is an optional file.
- The reference file, which contains the “reference” results and is used in the frame of a validation procedure.
- The friction data file, which contains information about the configuration of the bottom friction when this configuration is complex
- The stage-discharge curves file, which gives information on the open boundaries where the characteristics are prescribed according specific elevation/flowrate laws.
- The sources file, containing information about the sources
- The sections input file, which contains the description of the control sections of the model (sections across which the flowrate is computed).

The output files are the following:

- The results file, containing the graphical results
- The listing printout, which is the “log file” of the computation. If necessary, the user can get additional information in this file by activating the integer keyword *DEBUGGER*. *DEBUGGER* = 1 will show the sequence of calls to subroutines in the main program telemac2d.f. This is useful in case of crash, to locate the guilty subroutine.
- The sections output file, which contains the results of the “control sections” computation

In addition, the user can manage additional files

- 2 binary data files
- 2 formatted date files
- 1 binary results file
- 1 formatted results file

Some of these files are used by TELEMAC-2D for specific applications.

Some others files are also required when coupling TELEMAC-2D with the water quality software DELWAQ. These files are described in appendix 4.

3.1.1. BINARY FILE FORMAT

The binary files managed inside the TELEMAC system can have various formats. The most commonly used format is the Serafin format (also called Selafin), the TELEMAC system internal standard format (described in appendix 3). This Serafin format can be configured in order to store real data as single or double precision. The other possible format is the MED format which is compatible with the Salomé platform developed by EDF and CEA. In the actual version of TELEMAC-2D, this MED format has to be considered only for internal use at EDF.

Depending of the selected format, the binary file can be read by different tools. But in actual version, only single precision Serafin format can be read by the post-processing tools like RUBENS, FUDAA-PREPRO or BLUEKENUE.

The selection of the appropriate format is made using the corresponding key-word. For example, the keyword *GEOMETRY FILE FORMAT* manages the format of the geometry file. Each keyword can take 3 different values (8 characters string):

'SERAFIN ' means the single precision Serafin format and is the default (and recommended) value (do not forget the space at the end), 'SERAFIND' is the double precision Serafin format which can be used for more accurate "computation continued" of more accurate validation, and 'MED ' means the MED hdf5 format (EDF internal use only).

3.2. THE FILES

3.2.1. THE STEERING FILE

This is a text file created by a text editor of by the FUDAA-PREPRO software (but generally, the user starts from an already existing parameter file available in the TELEMAC structure, for example in the test cases directories.

In a way, it represents the control panel of the computation. It contains a number of keywords to which values are assigned. All keywords are defined in a "dictionary" file which is specific to each simulation module. If a keyword is not contained in this file, TELEMAC-2D will assign it the default value defined in the dictionary file of in the appropriate Fortran subroutine (see description in section 3.2.14). If such a default value is not defined in the dictionary file, the computation will stop with an error message. For example, the command *TIME STEP = 10*. enables the user to specify that the computational time step is 10 seconds.

TELEMAC-2D reads the steering file at the beginning of the computation.

The dictionary file and steering file are read by a utility called DAMOCLES, which is included in TELEMAC-2D. Because of this, when the steering file is being created, it is necessary to comply with the rules of syntax used in DAMOCLES. They are briefly described below.

The rules of syntax are the following:

- The keywords may be of Integer, Real, Logical or Character type.
- The order of keywords in the steering file is of no importance.
- Each line is limited to 72 characters. However, it is possible to pass from one line to the next as often as required, provided that the name of the keyword is not split between two lines.
- For keywords of the array type, the separator between two values is the semi-colon. It is not necessary to give a number of values equal to the size of the array. In this case, DAMOCLES returns the number of read values. For example:

TYPE OF ADVECTION = 1;5

(this keyword is declared as an array of 4 values)

- The signs ":" or "=" can be used indiscriminately as separator for the name of a keyword and its value. They may be preceded or followed by any number of spaces. The value itself may appear on the next line. For example:

  TIME STEP =   10.

  or

  TIME STEP: 10.

  or again

  TIME STEP =
  10.

- Characters between two "/" on a line are considered as comments. Similarly, characters between a "/" and the end of line are also considered as comments. For example:

  TURBULENCE MODEL = 3   / Model K-Epsilon

- A line beginning with "/" in the first column is considered to be all comment, even if there is another "/" in the line. For example:

  / The geometry file is ./mesh/geo

- When writing integers, do not exceed the maximum size permitted by the computer (for a computer with 32-bit architecture, the extreme values are -2 147 483 647 to + 2 147 483 648. Do not leave any space between the sign (optional for the +) and number. A full stop is allowed at the end of a number.

- When writing real numbers, the full stop and comma are accepted as decimal points, as are E and D formats of FORTRAN. (1.E-3 0.001 0,001 1.D-3 represent the same value).

- When writing logical values, the following are acceptable: 1 OUI YES .TRUE. TRUE VRAI and 0 NON NO .FALSE. FALSE FAUX.

- Character strings including spaces or reserved symbols ("/",":", "=", "&") must be placed between apostrophes ('). The value of a character keyword can contain up to 144 characters. As in FORTRAN, apostrophes in a string must be doubled. A string cannot begin or end with a space. For example:

  TITLE = 'CASE OF GROYNE'

In addition to keywords, a number of instructions or meta-commands interpreted during sequential reading of the steering file can also be used:

- Command &FIN indicates the end of the file (even if the file is not finished). This means that certain keywords can be deactivated simply by placing them behind this command in order to reactivate them easily later on. However, the computation continues.
- Command &ETA prints the list of keywords and the value that is assigned to them when DAMOCLES encounters the command. This will be displayed at the beginning of the listing printout.
- Command &LIS prints the list of keywords. This will be displayed at the beginning of the listing printout.
- Command &IND prints a detailed list of keywords. This will be displayed at the beginning of the listing printout.
- Command &STO stops the program and the computation is interrupted.

3.2.2. THE GEOMETRY FILE

This is a binary file.

This file contains all the information concerning the mesh, i.e. the number of mesh points (NPOIN variable), the number of elements (NELEM variable), the number of nodes per element (NDP variable), arrays X and Y containing the coordinates of all the nodes and array IKLE containing the connectivity table.

This file can also contain bottom topography information and/or friction coefficient at each mesh point.

TELEMAC-2D stores information on the geometry at the start of the results file. Because of this, the computation results file can be used as a geometry file if a new simulation is to be run on the same mesh.

The name of this file is given with the keyword: *GEOMETRY FILE*.

The format of this file is given by the keyword *GEOMETRY FILE FORMAT.*

3.2.3. THE BOUNDARY CONDITIONS FILE

This is a formatted file generated automatically by MATISSE, FUDAA-PREPRO or STBTEL. It can be modified with a standard text editor. Each line of the file is dedicated to one point on the mesh boundary. The numbering used for points on the boundary is that of the file lines. First of all, it describes the contour of the domain trigonometrically, starting from the bottom left-hand corner (X + Y minimum) and then the islands in a clockwise direction.

See section 4.2.3 for a fuller description of this file.

The file name is given with the keyword: *BOUNDARY CONDITIONS FILE*.

3.2.4. THE FORTRAN USER FILE

Since version 5.0 of the software (the first version to be written in FORTRAN 90), this file has become optional, as TELEMAC-2D uses a dynamic memory allocation process and it is therefore no longer necessary to set the size of the various arrays in the memory.

The FORTRAN file contains all the TELEMAC-2D subroutines modified by the user and those that have been specially developed for the computation.

This file is compiled and linked so as to generate the executable program for the simulation.

The name of this file is given with the keyword: *FORTRAN FILE*.

3.2.5. THE LIQUID BOUNDARIES FILE

This text file enables the user to specify values for time-dependent boundary conditions (tracer flow rate, depth, velocity, and tracers concentration).

See section 4.2.5 for a complete description of this file.

The file name is specified with the keyword: *LIQUID BOUNDARIES FILE*

3.2.6. THE SOURCE FILE

This text file enables the user to specify values for time-dependent conditions for sources (discharge, tracers concentration).

See section 8.4 for a complete description of this file.

The file name is specified with the keyword: *SOURCES FILE*

3.2.7. THE FRICTION DATA FILE

This text file enables the user to configure the bottom friction (used law and associated friction coefficient) in the domain. These information car vary from one zone to another.

The file name is specified with the keyword: *FRICTION DATA FILE* but is used only if the logical keywor*d FRICTION DATA is activated.*

By default, the number of friction domains is limited to 10 but can be modified using the keyword *MAXIMIM NUMBER OF FRICTION DOMAINS.*

See appendix 6 for a complete description of this file.

3.2.8. THE STAGE-DISCHARGE CURVES FILE

This text file enables the user to configure the evolution of the prescribed value on specific open boundaries. This file is used when the prescribed elevation is determined by a elevation/discharge elevation law. The descriptions of the appropriate laws are given through this file.

See section 4.2.6 for a complete description of this file.

The file name is specified with the keyword: *STAGE-DISCHARGE CURVES FILE*

3.2.9. THE SECTIONS INPUT FILE

This text file enables the user to configure the control sections used during the simulation.

See section 5.2 for a complete description of this file.

The file name is specified with the keyword: *SECTIONS INPUT FILE*

3.2.10. THE REFERENCE FILE

When a calculation is being validated, this file contains the reference result. At the end of the calculation, the result of the simulation is compared to the last time step stored in this file. The result of the comparison is given in the control printout in the form of a maximum difference in depth and the two velocity components.

The name of this file is given with the keyword: *REFERENCE FILE* and its format is specified by *REFERENCE FILE FORMAT.*

3.2.11. THE RESULTS FILE

This is the file in which TELEMAC-2D stores information during the computation. It is normally in Serafin (single precision) format. It contains first of all information on the mesh geometry, then the names of the stored variables. It then contains the time for each time step and the values of the different variables for all mesh points.

Its contents are dependent on the value of the following keywords:

NUMBER OF FIRST TIME STEP FOR GRAPHIC PRINTOUTS: this is used to determine at what time step information is first to be stored, so as to avoid having excessively large files, especially when a period of stabilisation precedes a transient simulation.

GRAPHIC PRINTOUT PERIOD: fixes the period for outputs so as to avoid having an excessively large file. In addition, whatever the output period indicated by the user, the last time step is systematically saved.

VARIABLES FOR GRAPHIC PRINTOUTS: this is used to specify the list of variables to be stored in the results file. Each variable is identified by a symbol (capital letter of the alphabet or mnemonic of no more than 8 characters); these are listed in the description of this keyword in the Reference Manual.

OUTPUT OF INITIAL CONDITIONS : this is used to specify whether the initial conditions for the calculation (time step 0) should be written in the results file. The default value for this keyword is YES.

The name of this file is given with the keyword: *RESULTS FILE* and its format is given with *RESULTS FILE FORMAT*

3.2.12. THE LISTING PRINTOUT

This is a formatted file created by TELEMAC-2D during the computation. It contains an account of the running of TELEMAC-2D. Its contents vary in accordance with the values of the following keywords:

NUMBER OF FIRST TIME STEP FOR LISTING PRINTOUTS: this is used to indicate at what time step to begin editing information, so as to avoid having excessively large files, in particular when a stabilisation period precedes a transient simulation.

LISTING PRINTOUT PERIOD : this fixes the period between time step editions. The value is given in numbers of time steps. For example, the following sequence:

TIME STEP = 30.

LISTING PRINTOUT PERIOD = 2

will produce a listing printout every minute of simulation. Moreover, irrespective of the period indicated by the user, the last time step is systematically printed.

LISTING PRINTOUT: this cancels the listing printout if the value is NO (the listing printout then only contains the program heading and normal end indication). However, this is not advisable in any circumstances.

VARIABLES TO BE PRINTED : this is used to specify the list of variables for which all values will be printed at each mesh point. This is a debugging option offered by TELEMAC-2D that should be handled with caution so as to avoid creating an excessively large listing printout.

MASS-BALANCE: if this is required, the user will have information on the mass fluxes (or rather volumes) in the domain at each printed time step.

INFORMATION ABOUT SOLVER: if this is required, at each printed time step the user will have the number of iterations necessary to achieve the accuracy required during solving of the discretized equations, or by default that reached at the end of the maximum number of iterations authorised.

INFORMATION ABOUT K-EPSILON MODEL: if this is required, at each printed time step the user will have the number of iterations necessary to achieve the accuracy required during computation of the diffusion and source terms of the k-Epsilon transport equations, or by default that reached at the end of the maximum number of iterations authorised.

The name of this file is managed directly by the TELEMAC-2D start-up procedure. In general, it has the name of the steering file and number of the process that ran the calculation, associated with the suffix .sortie.

3.2.13. THE ANCILLARY FILES

Other files may be used by TELEMAC-2D.

One or two binary data files, specified by the keywords *BINARY DATA FILE 1* and *BINARY DATA FILE 2* . These files can be used to provide data to the program, with the user of course managing reading within the FORTRAN program (logical units 24 and 25).

One or two formatted data files, specified by the keywords *FORMATTED DATA FILE 1* and *FORMATTED DATA FILE 2*. These files can be used to provide data to the program, with the user of course managing reading within the FORTRAN program (logical units 26 and 27).

A binary results file specified by the keyword *BINARY RESULTS FILE*. This file can be used to store additional results (for example the trajectories followed by floats when these are required). Write operations on the file are managed by the user in the FORTRAN program (logical unit 28).

A formatted results file specified by the keyword *FORMATTED RESULTS FILE*. This file can be used to store additional results (for example results that can be used by a 1D simulation code when two models are linked). Write operations on the file are managed by the user in the FORTRAN program (logical unit 29).

Read and write operations on these files must be managed completely by the user. Management can be done from any point accessible to the user. For example, using a file to provide the initial conditions will mean managing it with the CONDIN subroutine. Similarly, using a file to introduce boundary conditions can be done in the BORD subroutine.

3.2.14. THE DICTIONARY FILE

This file contains all information on the keywords (name in French, name in English, default values, type, documentation on keywords. This file can be consulted by the user but must under no circumstances be modified.

3.3. TOPOGRAPHICAL AND BATHYMETRIC DATA

Topographical and bathymetric data may be supplied to TELEMAC-2D at three levels:

Either directly in the geometry file by a topographical or bathymetric value associated with each mesh node. In this case, the data are processed while the mesh is being built using MATISSE or BLUEKENUE, or when the STBTEL module is run before TELEMAC-2D is started. STBTEL reads the information in one or more bottom topography files (5 at most) and interpolates at each point in the domain.

Or in the form of a cluster of points with elevations that have no relation with the mesh nodes, during the TELEMAC-2D computation. TELEMAC-2D then makes the interpolation directly with the same algorithm as STBTEL. The file name is provided by the keyword *BOTTOM TOPOGRAPHY FILE*. In contrast to STBTEL, TELEMAC-2D only manages one bottom topography file. This may be in SINUSX format or more simply a file consisting of three columns X,Y,Z. The SINUSX format is described in the RUBENS user manual.

Or using the CORFON subroutine (see section 11.1). This is usually used for schematic test cases.

In all cases, TELEMAC-2D offers the possibility of smoothing the bottom topography in order to obtain a more regular geometry. The smoothing algorithm can be iterated several times depending on the degree of smoothing required. The keyword *BOTTOM SMOOTHINGS* then defines the number of iterations carried out in the CORFON subroutine. The default value of this keyword is 0 (see also programming of the CORFON subroutine in section 11.1).

This page is intentionally left blank

4. HYDRODYNAMIC SIMULATION

4.1. PRESCRIPTION OF INITIAL CONDITIONS

The purpose of the initial conditions is to describe the state of the model at the start of the simulation.

In the case of a continued computation, this state is provided by the last time step of the results file for the previous computation. The tables of variables that are essential for continuing the computation must therefore be stored in a file used for this purpose. This case is described in section 4.1.3.

In other cases, the initial state must be defined by the user. This can be done using keywords in simple cases, or by programming in more complex ones.

If the user wants to store the initial state in the results file, the keyword *OUTPUT OF INITIAL CONDITIONS* must be activated (default value)

4.1.1. PRESCRIBING USING KEYWORDS

In all cases, the nature of the initial conditions is fixed with the keyword *INITIAL CONDITIONS*. This may have any of the following five values:

- ‘ZERO ELEVATION’: This initialises the free surface elevation at 0. The initial depths of water are therefore calculated from the bottom elevation.

- ‘CONSTANT ELEVATION’: This initialises the free surface elevation at the value supplied by the keyword INITIAL ELEVATION. The initial depths of water are then calculated by subtracting the bottom elevation from the free surface elevation. In areas where the bottom elevation is higher than the initial elevation, the initial depth of water is nil.

- ‘ZERO DEPTH’: All water depths are initialised with a zero value (free surface same as bottom). In other words, the entire domain is dry at the start of the computation.

- 'CONSTANT DEPTH’: This initialises the water depths at the value supplied by the keyword INITIAL DEPTH.

- ‘PARTICULAR’: The initial conditions are defined in the CONDIN subroutine (see section 4.1.2). This solution must be used whenever the initial conditions of the model do not correspond to one of the four cases above.

4.1.2. PRESCRIBING WITH THE CONDIN SUBROUTINE

The CONDIN subroutine must be programmed whenever the keyword *INITIAL CONDITIONS* has the value ‘PARTICULAR’.

The CONDIN subroutine initialises successively the depth of water, the velocities, the tracer, the k-Epsilon model and the viscosity. That part of the subroutine concerning the initialisation of the water depth is divided into two zones. The first corresponds to the processing of simple initial conditions (defined by the keyword) and the second the processing of particular initial conditions.

By default, the standard version of the CONDIN subroutine stops the computation if the keyword *INITIAL CONDITIONS* is positioned at PARTICULAR without the subroutine actually being modified.

The user is entirely free to fill this subroutine. For example, he can re-read information in a formatted or binary file using the keywords *FORMATTED DATA FILE* or *BINARY DATA FILE* offered by TELEMAC-2D.

When the CONDIN subroutine is being used, it may be interesting to check that the variables are correctly initialised. To do this, it is simply a question of assigning the name of the variables to be checked to the keyword *VARIABLES TO BE PRINTED* , and starting the computation with a zero number of time steps. The user then obtains the value of the variables required at each point of the mesh in the listing printout.

4.1.3. CONTINUING A COMPUTATION

TELEMAC-2D enables the user to carry out a computation taking the last time step of a previous computation on the same mesh as initial state. It is thus possible to modify the computation data, such as, for example, the time step, certain boundary conditions or the turbulence model, or to start the computation once a steady regime has been reached.

In this case, it is essential for the file to contain all the information required by TELEMAC-2D, i.e. the velocities U and V, the water depth and the bottom elevations. However, in certain cases, the software is capable of recomputing certain variables from others provided (for example the depth of water from the free surface and the bottom elevation).

If certain variables are missing from the continuation file, they are then fixed automatically at zero. However, it is possible, in this case, to provide initial values in a standard way (e.g. using a keyword). A frequent application involves using the result of a hydrodynamic computation to compute the transport of a tracer. The continuation file does not normally contain any result for the tracer. However, it is possible to provide the initial value for this by using the keyword INITIAL VALUES OF TRACERS.

In order to use the continuation file, it is necessary to enter two keywords in the steering file:

- The keyword *COMPUTATION CONTINUED* must have the value YES.
- The keyword *PREVIOUS COMPUTATION FILE* must provide the name of the file that will supply the initial state.

N.B.: the mesh for which the results are computed must be exactly the same as the one to be used in continuing the computation.

If necessary, the keyword *PREVIOUS COMPUTATION FILE FORMAT* can be used to select a specific format. For example, in order to increase the accuracy of the initial state, it is possible to use double precision Serafin format. Obviously, this configuration is possible only if the previous computation was correctly configured in terms of results file format.

When continuing a computation, it is necessary to specify the value of the start time of the second computation. By default, the initial time of the second computation is equal to the value of the last time step stored in the previous computation file. This can be modified using the keyword *INITIAL TIME SET TO ZERO* if the user wants to reset the time value (possibly with respect to a basic value set in the preceding calculation. See section 5).

At the beginning of a simulation, the launcher creates a temporary directory where all input files are copied. This is also the case for the previous computation file which can be quite huge. In this situation and to avoid copying too large a file, it is recommended to extract the last time step (the only one used by TELEMAC-2D) using the LastSela utility provided inside the TELEMAC distribution. Note that this utility works only with single precision Serafin files (default configuration).

4.2. PRESCRIBING BOUNDARY CONDITIONS

4.2.1. POSSIBLE CHOICES

Boundary conditions are given for each of the boundary points. They concern the dependent variables of TELEMAC-2D or the values deduced from them: water depth, the two components of velocity (or flowrate) and the tracer. The boundary conditions of functions k and Epsilon in the turbulence model are determined by TELEMAC-2D and are thus not required from the user.

The various types of boundary conditions may be combined to prescribe boundary conditions of any physical type (inflow or outflow of liquid in a supercritical or subcritical regime, open sea, wall, etc.). However, certain combinations are not physical.

Certain boundary conditions apply to segments, such as friction at the walls, no flux condition or incident wave conditions. However, wall definition is ambiguous if boundary conditions are to be defined by points. The following convention is used in such cases to determine the nature of a segment situated between two points of different type. A liquid segment is one between two points of liquid type. In a similar way, when a condition is being prescribed for a segment, the point must be configured at the start of the segment.

The way in which a boundary condition is prescribed depends on the spatial and temporal variations in the condition. Five types of condition may be distinguished:

- The condition is constant at the boundary and constant in time. The simplest solution is then to prescribe the condition by means of a keyword in the steering file.
- The condition is constant at the boundary and variable in time. It will then be prescribed by programming the functions Q,SL and VIT (and TR if a tracer is used) or by the open boundaries file.
- The condition is variable in space and constant in time. It will then be prescribed via the boundary conditions file. In certain cases, the velocity profile can be specified using the keyword *VELOCITY PROFILES* (see section 4.2.8)
- The condition is variable in time and space. Direct programming via the BORD subroutine is then necessary.
- The boundary condition type is variable in time. Direct programming in the PROPIN_TELEMAC2D subroutine is then necessary (see 11.7).

The type of boundary condition, if constant in time, is read from the boundary conditions file. In contrast, the prescribed value (if one exists) may be given at four different levels, namely (in the order in which they are processed during the computation) the boundary conditions file, the steering file, the open boundaries file and the FORTRAN file (programming of functions Q, SL, VIT, TR or BORD). This means that in the case of an open boundary with a prescribed level, the keyword *PRESCRIBED ELEVATIONS*, if it exists, will provide the value of the level at the prescribed depth boundaries even if a depth value is provided in the boundary conditions file.

Boundary types may be connected in any way along a contour (for example, there may be an open boundary with a prescribed depth followed by an open boundary with a prescribed velocity). The only limitation is that a boundary must consist of at least two points (a minimum of four points is strongly advised).

4.2.2. DESCRIPTION OF THE VARIOUS TYPES OF CONDITION

The type of boundary condition at a given point is provided in the boundary conditions file in the form of four integers named LIHBOR, LIUBOR, LIVBOR and LITBOR, which may have any value from 0 to 6.

The possible choices are as follows:

Depth conditions:

- Open boundary with prescribed depth: LIHBOR=5
- Open boundary with free depth: LIHBOR=4
- Open boundary with incident wave: LIHBOR=1
- Closed boundary (wall): LIHBOR=2

Flowrate or velocity condition:

- Open boundary with prescribed flowrate: LIUBOR/LIVBOR=5
- Open boundary with prescribed velocity: LIUBOR/LIVBOR=6
- Open boundary with free velocity: LIUBOR/LIVBOR=4
- Open boundary with slip or friction: LIUBOR/LIVBOR=2
- Closed boundary with one or two nil velocity components: LIUBOR and/or LIVBOR=0
- Open boundary with incident wave: LIUBOR/LIVBOR=1

Tracer conditions:

- Open boundary with prescribed tracer: LITBOR=5
- Open boundary with free tracer: LITBOR=4
- Closed boundary (wall): LITBOR=2

Remarks:

It is possible to change the type of boundary condition within an open boundary. In that case, a new open boundary will be detected in the output control listing.

In the case of an open boundary with an incident wave, the user must fill in the INCIDE subroutine in order to introduce the wave characteristics (see 4.2.5). This boundary condition is such that, if a wave generated inside the domain leaves it perpendicular to the boundary, there will be no reflection. In other cases, reflection phenomena are possible.

The type of boundary condition during the simulation may be modified with the PROPIN_TELEMAC2D subroutine (see 11.7).

4.2.3. THE BOUNDARY CONDITIONS FILE

The file is normally supplied by MATISSE, FUDAA-PREPRO or STBTEL, but may be created and modified using a text editor. Each line of this file is dedicated to one point of the mesh boundary. The numbering of the boundary points is the same as that of the lines of the file. It describes first of all the contour of the domain in a trigonometric direction, and then the islands in the opposite direction.

This file specifies a numbering of the boundaries. This numbering is very important because it is used when prescribing values.

The following values are given for each point (see also the section dedicated to parallel processing for certain specific aspects):

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

LIHBOR, LIUBOR, LIVBOR, and LITBOR are the boundary type codes for each of the variables. They are described in section 4.2.2.

HBOR (real) represents the prescribed depth if LIHBOR = 5.

UBOR (real) represents the prescribed velocity U if LIUBOR = 6.

VBOR (real) represents the prescribed velocity V if LIVBOR = 6.

AUBOR represents the friction coefficient at the boundary if LIUBOR or LIVBOR = 2. The friction law is then written as follows:

$$\frac{dU}{dn} = AUBOR * U \quad and/or \quad \frac{dV}{dn} = AUBOR * v$$

The coefficient AUBOR applies to the segment included between the boundary point considered and the following point (in a trigonometric direction for the outer contour and in the opposite direction for the islands). The default value is AUBOR = 0. Friction corresponds to a negative value. With the k-Epsilon model, the value of AUBOR is computed by TELEMAC-2D and the indications in the boundary conditions file are then ignored.

TBOR (real) represents the prescribed value of the tracer when LITBOR = 5.

ATBOR and BTBOR represent the coefficients of the flow relation, written:

$$\frac{dT}{dn} = ATBOR * T \ + BTBOR$$

The coefficients ATBOR and BTBOR apply to the segment between the boundary point considered and the next point (in a trigonometric direction for the outer contour and in the opposite direction for the islands).

N represents the total number of the boundary points in the case of a non-structured mesh.

K represents initially the point number in the boundary point numbering. But this number can also represent a node colour modified manually by the user (it can be any integer). This number, called BOUNDARY_COLOUR, can be used in parallelism to simplify implementation of specific cases. Without any manual modification, this variable represents the global boundary node number. For example a test like : IF (I.EQ.144) THEN … can be replaced by IF (BOUNDARY_COLOUR%I(I).EQ.144) THEN which is compatible with parallel mode.

4.2.4. PRESCRIBING VALUES USING KEYWORDS

In most simple cases, boundary conditions are prescribed using keywords. However, if the values to be prescribed vary in time, it is necessary to program the appropriate functions or use the open boundaries file (see 4.2.5).

The keywords used for prescribing boundary conditions are the following:

PRESCRIBED ELEVATIONS: This is used to define the elevation of an open boundary with prescribed depth. It is a table that contains up to 100 real numbers for managing up to 100 boundaries of this type. The values provided with this keyword cancel the depth values read from the boundary conditions file.

N.B.: the value given here is the level of the free surface, whereas the value given in the boundary conditions file is the water depth.

PRESCRIBED FLOWRATES: This is used to fix the flowrate value of an open boundary with prescribed flowrate. It is a table that contains up to 100 real numbers for managing up to 100 boundaries of this type. A positive value corresponds to an inflow into the domain. The values provided with this keyword cancel the flowrate values read from the boundary conditions file. In this case, the technique used by TELEMAC-2D to compute the velocity profile is that described in section 4.2.8.

PRESCRIBED VELOCITIES: This is used to fix the velocity value of an open boundary with prescribed velocity. The scalar value provided is the intensity of the velocity perpendicular to the wall. A positive value corresponds to an inflow into the domain. It is a table that contains up to 100 real numbers for managing up to 100 boundaries of this type. The values provided with this keyword cancel the values read from the boundary conditions file.

Some simple rules must also be complied with:

There must of course be agreement between the type of boundary specified in the boundary conditions file and the keywords of the steering file (do not use the keyword *PRESCRIBED FLOWRATES* if there are no boundary points with the LIUBOR and LIVBOR values set at 5).

For each keyword, the number of specified values must be equal to the total number of open boundaries. If a boundary does not correspond to the specified keyword, the value will be ignored (for example, the user can specify 0.0 in all cases). In the examples in the introductory manual, the first boundary (downstream) is with prescribed elevation, and the second one (upstream) is with prescribed flowrate. It is therefore necessary to specify in the steering file:

PRESCRIBED ELEVATIONS = 265.0 ; 0.0
PRESCRIBED FLOWRATES = 0.0 ; 500.0

4.2.5. PRESCRIBING VALUES BY PROGRAMMING FUNCTIONS OR USING THE OPEN BOUNDARIES FILE

Values that vary in time but are constant along the open boundary in question are prescribed by using the open boundaries file or by programming a particular function, which may be:

- Function VIT to prescribe a velocity,
- Function Q to prescribe a flowrate,
- Function SL to prescribe an elevation,
- Function TR to prescribe a tracer concentration (see section 8),
- Subroutine INCIDE to prescribe an incident wave.

Functions Q, VIT and SL are programmed in the same way. In each case, the user has the time, the boundary rank (for determining, for example, whether the first or second boundary with a prescribed flowrate is being processed) and in the case of Q, information on the depth of water at the previous time step. By default the functions prescribe the value read from the boundary conditions file or supplied by keywords.

For example, the body of function Q for prescribing a range of flowrates lasting 1000 seconds and reaching a value of 400 m3/s could take a form similar to:

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

Programming the INCIDE subroutine is slightly more complex. This subroutine is called up at each time step and is used to fix the characteristics of the prescribed incident wave. To complete it, the user must know the direction, height, pulsation and phase of the wave. The subroutine computes the variable COTOND, which represents the free surface elevation.

Using the liquid boundaries file is an alternative to programming the functions mentioned above. This is a text file edited by the user, the name of which is given with the keyword *LIQUID BOUNDARIES FILE*. This file has the following format:

- A line beginning with the sign # is a line of comments.
- It must contain a line beginning with T to identify the value provided in this file. Identification is by a mnemonic identical to the name of the variables: Q for flow rate, SL for water level, U and V for velocities and T for tracer. An integer between brackets specifies the rank of the boundary in question. This line is followed by another indicating the unit of the variables.
- The values to be prescribed are provided by a succession of lines that must have a format consistent with the identification line. The time value must increase, and the last time value provided must be the same as or greater than the corresponding value at the last time step of the simulation. If not, the calculation is suddenly interrupted.

When TELEMAC-2D reads this file, it makes a linear interpolation in order to calculate the value to be prescribed at a particular time step. The value actually prescribed by the code is printed in the control printout.

An example of an open boundaries file is given below:

```
#  Example of open boundaries file
#  2 boundaries managed
#
T    Q(1)     SL(2)
s    m3/s     m
0.   0.       135.0
25.  15.      135.2
100.          20.      136.
500.          20.      136.
```

4.2.6. STAGE-DISCHARGE CURVE

It is possible to manage boundary where the prescribed value of the elevation is a function of the local discharge. This is particularly useful for river application.

First, it is necessary to define which boundary will use this type of condition using the keyword *STAGE-DISCHARGE CURVES* which supply one integer per liquid boundary. This integer can be:

- 0: no stage-discharge curve (default value)
- 1: elevation function of discharge

The keyword *STAGE-DISCHARGE CURVES FILE* supplies the name of the text file containing the curves. One example is presented hereafter:

```
#
#  STAGE-DISCHARGE CURVE BOUNDARY 1
#
Q(1)    Z(1)
m3/s     m
61.     0.
62.     0.1
63.     0.2
#
#  STAGE-DISCHARGE CURVE BOUNDARY 2
#
Z(2)    Q(2)
m      m3/s
10.     1.
20.     2.
30.     3.
40.     4.
50.     5.
```

Order of curves is not important. Order of columns may be swapped like in the example for boundary 2. Lines beginning with # are comments. Lines with units are mandatory but units are not checked so far. The number of points given is free and is not necessarily the same for different curves.

Caution: at initial conditions the discharge at exits may be nil. The initial elevation must correspond to what is in the stage-discharge curve, otherwise a sudden variation will be imposed. To avoid extreme situations the curve may be limited to a range of discharges. In the example above for boundary 1 discharges below 61 m^3/s will all give an elevation of 0. m, discharges above 63 m^3/s will give an elevation of 0.2 m.

4.2.7. PRESCRIBING COMPLEX VALUES

If the values to be prescribed vary in both time and space, it is necessary to program the BORD subroutine as this enables values to be prescribed on a node-by-node basis.

This subroutine describes all the open boundaries (loop on NPTFR). For each boundary point, it determines the type of boundary in order to prescribe the appropriate value (velocity, elevation or flowrate). However, there is little sense in programming BORD to prescribe a flowrate, as this value is usually known for the entire boundary and not for each segment of it.

In the case of a prescribed flowrate boundary located between two solid boundaries with no velocities, the velocities on the angle points are cancelled.

N.B.: The BORD subroutine also enables the tracer limit values to be prescribed (see 8.3).

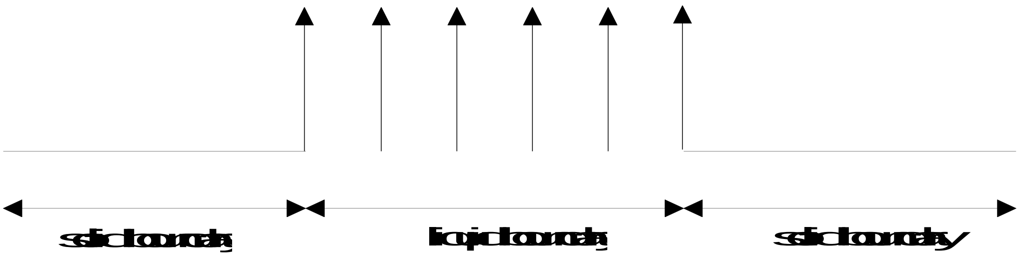
4.2.8. PRESCRIBING VELOCITY PROFILES

In the case of a flowrate or velocity conditions, the user can specify the velocity profile computed by TELEMAC-2D, using the keyword *VELOCITY PROFILES*. The user must supply one value for each open boundary. The following options are available:

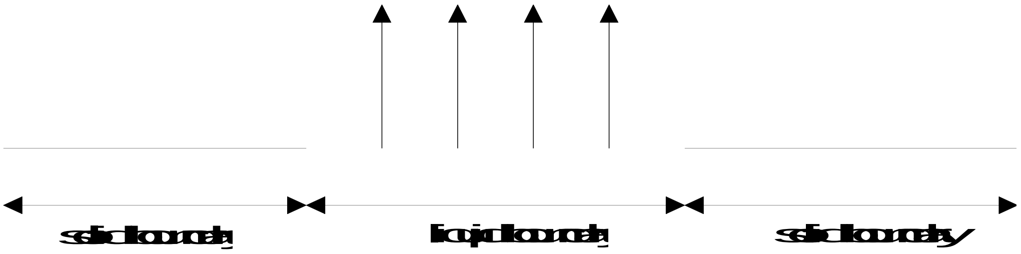
- 1:The velocity vector is normal to the boundary. In the case of a prescribed flowrate, the value of the vector is set to 1 and then multiplied by a constant in order to obtain the desired flowrate (given by the keyword *PRESCRIBED FLOWRATES* or by the function Q). In the case of a prescribed velocity, the value used for the velocity norm is provided by the keyword *PRESCRIBED VELOCITIES* or by the function VIT. In any case, the velocity profile is constant along the boundary.

- 2: The values U and V are read from the boundary conditions file (UBOR and VBOR values). In the case of a prescribed flowrate, these values are multiplied by a constant in order to reach the prescribed flowrate.
- 3: The velocity vector is imposed normal to the boundary. Its value is read from the boundary conditions file (UBOR value). In the case of a prescribed flowrate this value is then multiplied by a constant in order to obtain the appropriate flowrate.
- 4: The velocity vector is normal to the boundary and its norm is proportional to the square root of the water depth. This option is valid only for prescribed flowrate.

In the case of a flow normal to a closed boundary, it is not good to have velocities perpendicular to the solid segments (as shown in the figure hereafter)



because the finite element interpolation will generate a non-zero flow though a solid segment. In this case, it is better to cancel the velocities on the first and last points of the boundary, as shown on the following figure:



4.2.9. THOMPSON BOUNDARY CONDITIONS

In some cases, not all the necessary information concerning the boundary conditions is available. This is usual for coastal domains where only the values of the sea level on several points are known. This kind of model is referred to as an “under-constrained” model.

To solve this problem, the Thompson method uses the characteristics method to calculate the missing values. For example, TELEMAC-2D will compute the velocity at the boundary in the case of a prescribed elevation.

This method can also be used for “over-constrained” models. In this case, the user specifies too much information at the boundary. If the velocity information and the level information are not consistent, too little or too much energy is going into the model. For this, the Thompson technique computes a new value for the velocity and performs small adjustments to cancel the inconsistencies in the information.

For this, the user can use the keyword *OPTION FOR LIQUID BOUNDARIES*, which offers two values (the user must specify 1 value for each open boundary):

- 1: strong setting
- 2: Thompson method

Taking a simplified view, it may be said that, in the case of the first option, the values are “imposed”, in the case of the second option, the values are “suggested”.

Note : the Thompson method is not available when running in parallel mode, it is however expected in release 6.1.

4.2.10. ELEMENTS MASKING

TELEMAC-2D offers the possibility of masking certain elements. This means, for example, that islands can be created in an existing mesh. The boundaries created in this way are processed as solid walls with a slip condition.

This option is activated with the logical keyword *ELEMENTS MASKED BY USER* (default value: NO). In this case, the user must indicate the number of elements masked by programming the MASKOB subroutine. This manages an array of real values MASKEL, the size of which is equal to the number of points and in which each value can be 0.D0 for a masked element and 1.D0 a normal one.

4.2.11. DEFINITION OF TYPES OF BOUNDARY CONDITION WHEN PREPARING THE MESH

When using MATISSE, the boundary condition type is prescribed during the last step of mesh generation.

When using the other mesh generators, it is generally possible to define the type of boundary condition during the mesh generation session, by prescribing a colour code. Each colour code corresponds to a particular type of boundary (wall, open boundary with prescribed velocity, etc.). The table showing colour codes and corresponding types of boundary is given in appendix 5.

5. GENERAL PARAMETER DEFINITION FOR THE COMPUTATION

General parameter definition for the computation is done only in the steering file.

Time information is supplied by the three keywords *TIME STEP* (real), *NUMBER OF TIME STEPS* (integer) and *DURATION*. The first defines the time separating two consecutive instants of the computation (but not necessarily two withdrawals from the results file). The total duration of the computation may be supplied by means of a number of time steps (keyword *NUMBER OF TIME STEPS*) or in the form of a total simulation period expressed in seconds (keyword *DURATION*). In the former case, the total duration is obviously equal to the time step value multiplied by the number of time steps.

If a steering file contains the keywords *DURATION* and *NUMBER OF TIME STEPS*, TELEMAC-2D uses the one that produces the longer simulation. In addition, if the keyword *DURATION* is used and does not correspond to a whole number of time steps, TELEMAC-2D will take the integer immediately higher.

The date and hour corresponding to the initial state of the computation are supplied by the keywords *ORIGINAL DATE OF TIME* (YYYY;MM;DD) and *ORIGINAL HOUR OF TIME* (HH;MM;SS). This is particularly important if the tide generating forces are taken into account (see 6.4)

The title of the computation is specified by the keyword *TITLE*.

5.1. CRITERIA FOR STOPPING A COMPUTATION

Independently of normal time indications (number of time steps and time step value), TELEMAC-2D offers two possibilities for conditionally stopping the computation:

Stopping when reaching a steady state: With this function, it is possible to start a computation, simulate a transient flow and stop the computation when a steady state is reached. The last time step in the results file created in this way can be used as an initial state for other computations (e.g. tracer transport). The test is triggered by indicating YES for the logical keyword *STOP IF A STEADY STATE IS REACHED*. It is then possible to define the permissible area of tolerance using the keyword *STOP CRITERIA*. This keyword is a table of three real numbers, representing the tolerance assigned to the velocity, depth and tracer. The computation is stopped when the absolute increment values of these variables between two time steps at all nodes are below the limits indicated. Assessing the right criterion depends on the case under study. It should be stressed, however, that this function is inoperative in the case of fundamentally non-stationary flows such as Karman eddies behind bridge piers.

Stopping in cases of divergence: This function is used to interrupt a computation if there is divergence. The principle is the same as in the previous case. The option is activated with the keyword *CONTROL OF LIMITS*. The extreme values are indicated with the keyword *LIMIT VALUES*. This is a table of 8 real numbers corresponding successively to:

- The minimum depth value H (by default -1000).
- The maximum depth value H (by default +9000).
- The minimum velocity value U (by default -1000).
- The maximum velocity value U (by default +1000).
- The minimum velocity value V (by default -1000).
- The maximum velocity value V (by default +1000).
- The minimum tracer value (by default -1000).
- The maximum tracer value (by default +1000).

5.2. CONTROL SECTIONS

A control section offers the possibility of obtaining the instantaneous and cumulated flow rates through a specific segment of the domain.

The weak formulation of the no-flux boundary condition through solid boundaries raises a theoretical problem for computed the flow rates. Either they are compatible with the results file, or they are compatible with the weak formulation. To be compatible with the weak formulation, use the key-word *COMPATIBLE COMPUTATION OF FLUXES*. The difference may reach a few percents.

It is also possible to obtain the cumulated flow rates for each control section by activating the logical keyword *PRINTING CUMULATED FLOWRATES*. In that case, to improve the quality of results, the treatment of the control section is done at each time step and not only at each time step concerned by a printing on output listing.

The control sections can be managed using 2 different procedures. The first one uses only keyword and is not valid when running in parallel mode. The second one (new in release 6.0) is based on an external configuration file and is compatible with the parallel mode. It is strongly recommended to use the new procedure. The old procedure will be probably removed in a future release.

5.2.1. CONFIGURATION WITH KEYWORDS ONLY

The section is defined using the keyword *CONTROL SECTIONS*, which is an array of pairs of integers separated by semi-colons, containing the numbers of the beginning and the ending point of the section.

For example, the values: 611;54 ; 651;5210 define 2 control sections. The first one is defined between points 611 and 54, the second one between points 651 and 5210.

The results concerning the flow rates are written by TELEMAC-2D on the output control listing. This information is the value of the instantaneous flow rate and the cumulated positive and negative flow rates (volume going through the section calculated from the beginning of the simulation). The sign is determined with the following rule: going from the beginning to the ending point of the section, the flow is positive when going from right to left.

The user may also use the subroutine FLUXPR (Bief library) to exploit information connected with the control sections.

5.2.2. CONFIGURATION WITH EXTERNAL FILE

The user must supply the name of the sections configuration file using the keyword *SECTIONS INPUT FILE*.

In parallel mode, this file will be modified by the mesh partitioner so that it corresponds locally to every sub-domain.

The file format is the following:

- one comment line (free but must be here)
- two integers: number of sections, steering integer (if negative: node numbers are given, if 0: coordinates are given, if positive: user gives ready segment chains that describes the control sections).
- two lines per section:
- 24 characters for a section name, followed by
- begin and end node number or begin and end coordinates

Example:

```
# Control sections definition
5 –1
Wesxan_outflow
46 70
Wesxan_Middle
639 263
```

```
Wesxan_Inflow
480 414
Wesxan_crazy
142 147
Wesxan_even_worse
144 7864
```

Headers and printouts on control sections may be modified in subroutine fluxpr_telemac2d (telemac2d library).

The printouts will be in the file named by *SECTIONS OUTPUT FILE*.

6. PHYSICAL PARAMETER DEFINITION

A number of physical parameters may or must be specified during a simulation. If the parameter is space-dependent, it is sometimes preferable to define various zones within the mesh and then assign the parameter as a function of the zone number. To do this, it is necessary to activate the logical keyword *DEFINITION OF ZONES* and fill the subroutine DEF_ZONES which assigns a zone number to each point. This zone number may then be used in the various subroutines for specifying a space-dependent physical parameter.

6.1. FRICTION PARAMETER DEFINITION

We describe hereafter the simplest case, when the friction law is the same in all the computation domain, when it is variable in space, refer to Appendix 6 after reading this paragraph. The friction law used to model friction on the bed is defined by the keyword *LAW OF BOTTOM FRICTION*. This may have the following values:

0	: No friction.
1	: Haaland's law.
2	: Chézy’s law.
3	: Strickler’s law.
4	: Manning’s law.
5	: Nikuradse law
6	: Log law of the wall (only for boundary conditions)
7	: Colebrooke-White law

Option 6 can be used only in the friction data file (see Appendix 6)

In the case of options 1 to 5, it is necessary to specify the value of the coefficient corresponding to the law chosen by means of the keyword *FRICTION COEFFICIENT*. This is of course only valid if the friction is constant in time and space.

In the case of option 7, the key-word *MANNING DEFAULT VALUE FOR COLEBROOK-WHITE LAW* must be given.

If the friction coefficient varies in time (and possibly in space as well), it is necessary to use the STRCHE and/or CORSTR subroutines, which supply the friction coefficient at each mesh point.

The following example shows how STRCHE is programmed for a domain in which the friction coefficient is 50 for the left part (X<10000) and 55 for the right part.

```
C       LOOP ON ALL POINTS OF DOMAIN
        DO 1 I = 1,NPOIN
                IF (X(I) .LT. 10000.D0) THEN
                        CHESTR%R (I) = 50.D0
                   ELSE
                        CHESTR%R (I) = 55.D0
                ENDIF
1       CONTINUE
```

When evaluating the friction term, it is possible to specify which depth is used for this computation through the keyword *DEPTH IN FRICTION TERMS*. The two possibilities are: the classical nodal depth (value 1 which is the default), or a depth averaged on the test function area (value 2). The second one is new in release 6.0 and seems to be slightly better on dam break studies.

6.1.1. NON-SUBMERGED VEGETATION

The effect of non-submerged vegetation can be added in the domain with the keyword:

NON-SUBMERGED VEGETATION FRICTION : YES

In this case, 2 additional data must be given:

DIAMETER OF ROUGHNESS ELEMENT

SPACING OF ROUGHNESS ELEMENT

They will be used in the Lindner law giving the drag coefficients of the vegetation.

This option will be preferably used with a definition of friction by domains, see Appendix 6.

6.2. MODELLING TURBULENCE

The modelling of turbulence is a delicate problem. TELEMAC-2D offers the user four options of different complexity.

The first involves using a constant viscosity coefficient. In this case, the coefficient represents the molecular viscosity, turbulent viscosity and dispersion.

The second option involves an Elder model.

The third option involves using a k-Epsilon model. This is a 2D model that solves the transport equations for k (turbulent energy) and Epsilon (turbulent dissipation). The model equations are solved by a fractional step method, with convection of turbulent variables being processed at the same time as the hydrodynamic variables, and the other terms relating to the diffusion and production/dissipation of turbulent values being processed in a single step. Use of the k-Epsilon model also often requires a finer mesh than the constant viscosity model and in this way increases computation time.

The fourth involves a Smagorinski model, generally used for maritime domains with large-scale eddy phenomena.

More detailed information on the formulation of the k-Epsilon model, the Elder model and the Smagorinski model can be found in the literature.

In addition, TELEMAC-2D offers two possibilities for processing the diffusion term. The option is selected by the keyword *OPTION FOR THE DIFFUSION OF VELOCITIES* which can take the value 1 (default) or 2. The first value selects a computation with the form $div\left(\nu\ \overrightarrow{grad}\ (U)\right)$, and the second one with the form $\frac{1}{h}div\left(h\nu\ \overrightarrow{grad}\ (U)\right)$.

This latter option is the only one offering good mass conservation, but difficulties may occur with tidal flats.

6.2.1. CONSTANT VISCOSITY

The first possibility is activated by giving the keyword *TURBULENCE MODEL* the value 1 (default value). Turbulent viscosity is then constant throughout the domain. The overall viscosity coefficient (molecular + turbulent viscosity) is provided with the keyword *VELOCITY DIFFUSIVITY* which has a default value of 10-4 (the minimum value is 10-6, corresponding to the molecular viscosity of water).

The value of this coefficient has a definite effect on the extent and shape of recirculation. A low value will tend to dissipate only small eddies, whereas a high value will tend to dissipate large recirculations. The user must therefore choose this value with care, depending on the case treated (in particular as a function of the size of the recirculation he wishes to dissipate and the mean angular velocity of the recirculation). It should also be noted that a value which results in the dissipation of eddies smaller than two meshes has virtually no effect on the computation.

TELEMAC-2D offers the possibility of having a coefficient that varies in time and space. This is defined in the CORVIS subroutine. This subroutine gives information on the geometry and basic hydrodynamic variables (depth of water, velocity components) and time.

6.2.2. ELDER MODEL

This option is used when the keyword *TURBULENCE MODEL* is set to 2.

The Elder model offers the possibility of specifying different viscosity values along and across the current (Kl and Kt respectively). The formulae used are:

Kl = al U*h and Kt = at U*h

Where:

U* is the friction velocity (m/s) and h the water depth (m)

al and at are the dimensionless dispersion coefficients equal to 6 and 0.6 respectively. Other values can be found in the literature.

The two coefficients can be supplied by the user with the keyword *NON-DIMENSIONAL DISPERSION COEFFICIENTS* (format: Kl,Kt).

6.2.3. K-EPSILON MODEL

If constant viscosity is not sufficient, TELEMAC-2D offers the possibility of using a k-Epsilon model. This is activated by assigning a value of 3 to keyword *TURBULENCE MODEL* .

In this case, the keyword *VELOCITY DIFFUSIVITY* has its real physical value (10-6 for molecular diffusion of water), as this is used as such by the turbulence model.

In the case of a solid boundary, the user may configure the turbulence regime for the walls using the keyword *TURBULENCE MODEL FOR SOLID BOUNDARIES*. If friction at the wall is not to be taken into account, the user must use the value corresponding to a smooth wall (option 1). In contrast, friction will be taken into account by using option 2 (rough wall). In this case, the friction law used for the wall is the same as the bottom friction law (keyword *LAW OF BOTTOM FRICTION*). The friction coefficient is then supplied by the keyword *ROUGHNESS COEFFICIENT OF BOUNDARIES*. This numerical value must of course be in agreement with the law chosen, in appropriate units.

If a k-Epsilon model is used, the information concerning the solution phase must be obtained by activating the keyword *INFORMATION ABOUT K-EPSILON MODEL*.

Parameter definition for the k-Epsilon model is described in chapter 7.

A good level of expertise in turbulence is necessary to use the k-epsilon model, especially to know when it is relevant to resort to it. As a matter of fact the turbulence should be larger than the dispersion terms. We quote here W. Rodi: “It should be emphasized that the model described here does not account for the dispersion terms appearing in the (depth-averaged) momentum equations".

6.2.4. SMAGORINSKI MODEL

The use of this model is activated by assigning a value of 4 to keyword *TURBULENCE MODEL*. Same remark than the k-epsilon model, it does not take into account the dispersion terms.

6.3. PARAMETER DEFINITION FOR WIND AND ATMOSPHERIC PRESSURE

TELEMAC-2D can be used to simulate flow while taking into account the influence of a wind blowing on the water surface. The logical keyword *WIND* is used first of all for determining whether this influence is to be taken into account and if so, the coefficient is then provided with the keyword *COEFFICIENT OF WIND INFLUENCE* (see § below). Lastly, if the are constant in time and space, wind speed in directions X and Y is supplied with the keywords *WIND VELOCITY ALONG X and WIND VELOCITY ALONG Y.*

The coefficient of wind influence hides complex phenomena. In fact, the influence of the wind depends on the smoothness (or, lack of it) of the free surface and the distance over which it acts (called the "fetch"). The coefficient value can be obtained from many different formulas.

This is the formula used by the Institute of Oceanographic Sciences (United Kingdom):

if $\left|\vec{U}_{vent}\right| < 5$ m/s $\quad a_{vent} = 0{,}565\ 10^{-3}$

if $5 < \left|\vec{U}_{vent}\right| < 19{,}22$ m/s $\quad a_{vent} = (-\ 0{,}12 + 0{,}137\ \left|\vec{U}_{vent}\right|\ )\ 10^{-3}$

if $\left|\vec{U}_{vent}\right| > 19{,}22$ m/s $\quad a_{vent} = 2{,}513\ 10^{-3}$

The parameter *COEFFICIENT OF WIND INFLUENCE* asked for by TELEMAC-2D is: $\rho_{air}\ /\ \rho\ \ a_{vent}$ and not a_{vent}. ρ_{air} is approximately 1.023 kg/m^3 and ρ is 1000 kg/m^3. Thus it is necessary to divide the value of a_{vent} by 1000 to obtain the value of the TELEMAC-2D keyword.

If there are tidal flats or dry zones in the domain, the wind may trigger unphysical velocities as it becomes the only driving term in the equations. To avoid this, wind is cancelled below a threshold value of depth, with the key-word *THRESHOLD DEPTH FOR WIND*

If the wind velocity varies in space and time, the user must indicate this in the METEO subroutine (see below).

Atmospheric pressure is taken into account by setting the keyword *AIR PRESSURE* to YES (the default value is NO). The pressure value is indicated directly in the METEO subroutine. By default, a pressure of 105 Pa (1 atmosphere) is initialised throughout the domain.

The METEO subroutine is called up if the wind or atmospheric pressure option is activated. By default, the subroutine is called up at the start of the computation (time value = 0) in order to fix the pressure at 105 Pa throughout the domain and the wind velocity at the values supplied with the corresponding keywords. The user has geometrical information on the mesh, and time information for programming any study situation, in particular winds that vary in time and space (in this case a test must be programmed for time values other than 0).

The following example shows a wind programmed in space and in time. For the left part of the domain (X<1000000) the wind in direction X is fixed at 10 m/s for the first 3600 seconds, and at 5 m/s subsequently. The X and Y wind components in the right part of the domain are nil.

```
C       INITIALISATION WIND Y AND WIND X FOR LT=0
        IF (LT.EQ.0) THEN
                CALL OV ('X=C     ',WINDX, Y, Z, 0.D0, NPOIN)
                CALL OV ('X=C     ',WINDY, Y, Z, 0.D0, NPOIN)
           ELSE
C       INITIALISATION WINDX LEFT PART FOR NON-ZERO TIMES
                DO 1 I=1,NPOIN
                        IF (X(I).LT.1000000.D0) THEN
                                IF (LT.LT.3600.D0) THEN
                                        WINDX (1) = 10.D0
                                   ELSE
                                        WINDY (1) = 5.D0
                                ENDIF
                        ENDIF
1               CONTINUE
        ENDIF
```

6.4. ASTRAL POTENTIAL

When modelling large maritime areas, it is sometimes necessary to take into account the effect of astral forces generating tide inside the area. For this, the user has several keywords at his disposal.

First of all, the logical keyword *TIDE GENERATING FORCE* (default value NO) allows these phenomena to be taken into account.

The keyword *LONGITUDE OF ORIGIN POINT* must be positioned at the right value.

Lastly, the two keywords *ORIGINAL DATE OF TIME* (format YYYY;MM;DD) and *ORIGINAL HOUR OF TIME* (format HH;MM;SS) must be used to give the beginning time of the simulation. This information is necessary for TELEMAC-2D to compute the respective position of the moon and the sun.

6.5. INCLUSION OF WAVE-INDUCED CURRENTS

It is possible to include wave-induced currents by recovering the information calculated by the wave propagation modules (essentially TOMAWAC but also possible with ARTEMIS). In the present state of the system, only a steady state can be taken into account. The procedure is as follows:

Run a wave propagation calculation on the same mesh as the TELEMAC-2D calculation, asking for the moving forces to be stored. In the case of TOMAWAC, these are the variables FX and FY.

Recover the wave results file and specify its name using the keyword *BINARY DATA FILE 1*.

Activate the keyword *WAVE DRIVEN CURRENTS* (default value NO)

Complete the keyword *RECORD NUMBER IN WAVE FILE*. This value corresponds to the iteration number stored in the wave file that must be taken into account by TELEMAC-2D. This is usually the last iteration stored.

If the user wishes to take into account several results from the wave propagation module again (e.g. in order to consider changes in sea level), FORTRAN programming is necessary.

6.6. VERTICAL STRUCTURES

It may be necessary to take into account the presence of an obstacle to flow, such as bridge piers, trees or even buildings, without having to model them in the mesh, especially as, in this case, the force opposing the flow generally varies with the depth of water.

To handle this problem, TELEMAC-2D can include drag forces connected with the presence of vertical structures in the model. This function is activated with the logical keyword *VERTICAL STRUCTURES* .

The drag forces must then be defined in the user subroutine DRAGFO. An example of programming is given in the subroutine itself.

6.7. OTHER PHYSICAL PARAMETERS

When modelling large areas, it is necessary to take into account the inertia effect of the Coriolis force. This is done by activating the logical keyword *CORIOLIS* (which has the default value NO). In this case, the value of the Coriolis coefficient (see Formulation Document) is supplied by the keyword *CORIOLIS COEFFICIENT* . This must be calculated in accordance with the latitude l by the formula:

FCOR = 2 ω sin(λ) ω being the angular velocity of the Earth, equal to 7.27 x 10^{-5} rad/s.

The components of the Coriolis force are thus:

FU = FCOR x V et FV = -FCOR x U

In the case of very large domains such as portions of oceans, it is necessary to carry out a simulation with spherical coordinates, in which case the Coriolis coefficient is adjusted automatically at each point of the domain (see 11.3). In this case, it is necessary to indicate the angle between the geographic north and the Y-axis of the model. This information is supplied by the keyword *NORTH*. This gives the angle (in degrees) between the north and the Y-axis, expressed positively in a trigonometric direction (the default value of the keyword is zero).

TELEMAC-2D also offers the possibility of defining the water density (keyword *WATER DENSITY* ; the default value is 1020 kg/m3, i.e. a value corresponding to a moderately saline sea water) and gravity acceleration (keyword *GRAVITY ACCELERATION* fixed by default at 9.81 m/ s2).

6.8. PARAMETER ESTIMATION

TELEMAC-2D contains a function for automatically determining an unknown physical parameter. In the current version of the software, it is only possible to determine the friction coefficient when using the Strickler or Chézy laws (keyword *LAW OF BOTTOM FRICTION* with value of 2 or 3).

The principle for determining a parameter involves performing a series of calculations and comparing the results provided by TELEMAC-2D with the available measurements. The parameter to be determined is then adjusted in order to obtain identical values.

The algorithm for estimating this parameter is activated with the keyword *PARAMETER ESTIMATION* , which provides the name of the parameter to be determined. The user can specify 'FRICTION' or 'FRICTION, STEADY'. In the second configuration, only the last time step of the simulation is checked. In the actual version of the software, it is strongly recommended to work only in permanent mode.

Measurement data are supplied via the user subroutine MESURES which contains the arguments ITER (iteration number) and TT (absolute time). The latter argument is used in processing real measurements. Each time the MESURES subroutine is called up, it must supply the measured water depth (HD), the two velocity components (UD and VD), as well as the weightings ALPHA1, ALPHA2 and ALPHA3 connected respectively with HD, UD, VD. The weighting is 1 if a measurement is available and 0 if it is not. For example, an ALPHA1 value of 1 for a given point means that a depth measurement is available for that point. Similarly, an ALPHA3 of 1 for a given point means that a velocity measurement V is available for that point. When a measurement is available, it may be advisable to replace the value 1 by a vector proportional to the local mesh size (see VECTOR('MASBAS',…) in the MESURES subroutine).

The comparison data may also be provided by a file in Serafin format, in which case the name is specified with the keyword *REFERENCE FILE*. The data are read automatically in this case.

If the parameter is space-dependent, it is necessary to activate the logical keyword *DEFINITION OF ZONES* and to complete the DEF_ZONES subroutine, which assigns a zone number to each point. In this case, a parameter value will be estimated for each zone. This value will be constant within the zone.

From the numerical point of view, the user must specify a number of parameters:

The cost function used must be indicated with the integer keyword *COST FUNCTION* which may have the value 1 (cost function based on the difference between depths and velocities, which is the default value) or 2 (cost function based on the difference between celerity and velocities). 2 seems to be preferable, even through the effect of this parameter is slight.

The integer keyword *IDENTIFICATION METHOD* is used to specify the technique used for minimising the cost function. It may have the value 1 (gradient, which is the default value), 2 (conjugate gradient) or 3 (Lagrangian interpolation).

As parameter estimation is based on an iterative procedure, it is necessary to specify the required level of accuracy and a maximum number of iterations.

Accuracy is indicated with the keyword *TOLERANCES FOR IDENTIFICATION* . This is an array of four integers corresponding respectively to the absolute accuracy for the depth, velocity u and velocity v, and the relative accuracy of the cost function (default values 1.E-3; 1.E-3; 1.E-3; 1.E-4). The iteration process is stopped when the absolute accuracy levels are reached or when the relative accuracy for the cost function is. The maximum number of iterations is specified with the keyword *MAXIMUM NUMBER OF ITERATIONS FOR IDENTIFICATION* which has the default value 20. As each iteration corresponds to two simulations, the value of this keyword must not be too high.

The results of estimating this parameter are provided in the results file. This is a geometry file in which the FRICTION variable has been added. The file can thus be reused as a geometry file for a new simulation.

7. NUMERICAL PARAMETER DEFINITION

7.1. GENERAL PARAMETER DEFINITION

First, it is necessary to specify the type of equation to be solved. The choice is made by using the *EQUATIONS* keyword, which can take the following values:

- SAINT-VENANT FE (default value)
- SAINT-VENANT FV
- BOUSSINESQ

The first option involves solving the Saint-Venant equations using the finite-element method. This is the "traditional" use of TELEMAC-2D.

The second option involves solving the Saint-Venant equations using the finite-volume method. In this case, the algorithm is explicit and means that the Courant number must be limited to 1. The variable time step option is then automatically used. TELEMAC-2D then adjusts the calculation time step so as to satisfy this Courant number criterion. However, it should be noted that this leads to irregular sampling from the graphic printout file and control listing. Lastly, it should be noted that all the options available when solving the Saint-Venant equations using the finite-element method are not necessarily available here.

The BOUSSINESQ option means that the equations are solved by the Boussinesq method.

In addition, it is necessary to specify the type of discretization to be used: linear triangle (3 nodes triangle), quasi-bubble triangle (4 nodes triangle) or quadratic triangle (6 nodes triangle). The choice is made with the keyword *DISCRETIZATIONS IN SPACE*. This keyword is a table of two integers that are related successively to the velocity and depth. For each of these variables, the value 11 means linear triangle space discretization, the value 12 means quasi-bubble triangle space discretization and value 13 means quadratic element.

In practice, the user can select the 3 following combinations:

- 11 ; 11 (default value) : linear velocity and linear depth (recommended)
- 12 ; 11 : quasi-bubble velocity and linear depth
- 13 ; 11 : quadratic velocity and linear depth.

The first one is the most efficient in terms of memory and CPU time and the third one is recommended for more accurate results (but increases significantly the memory and CPU time). The second one is recommended when observing free surface wiggles (in particular in case of strong bathymetry gradient). But in that situation, the best configuration is to use the wave equation associated with the keyword *FREE SURFACE GRADIENT COMPATIBILITY* = 0.9.

During computation, TELEMAC-2D solves different steps using, if necessary, the fractional step method (the advection equations and propagation-diffusion equations may be solved in two successive stages handled by different numerical schemes). The user can activate or deactivate certain of these steps.

Whether or not the advection terms are taken into account is determined by the logical keyword *ADVECTION* (default value YES). However, even if this keyword is positioned at YES, it is possible to deactivate certain advection terms using the following logical keywords:

- *ADVECTION OF H*: to take into account the advection of depth,
- *ADVECTION OF U AND V*: for the advection of velocity components,
- *ADVECTION OF K AND EPSILON*: for the advection of turbulent energy and turbulent dissipation,
- *ADVECTION OF TRACERS*: for the advection of a tracer.

The default value of these three keywords is YES.

The phenomena of propagation will or will not be taken into account depending on the value of the keyword *PROPAGATION* (default value YES). As propagation and diffusion are processed in the same step, deactivating propagation will automatically entail deactivating diffusion.

However, if the propagation-diffusion step is activated, the user may still decide whether or not to take into account velocity diffusion by setting the logical keyword *DIFFUSION OF VELOCITY* (default value YES).

The propagation step may be linearized by activating the keyword *LINEARIZED PROPAGATION* in particular when running a test case for which an analytical solution is available in the linearized case. It is then necessary to determine the depth of water around which the linearization is to be performed, by using the keyword *MEAN DEPTH FOR LINEARIZATION* (default value 0.).

7.2. THE NUMERICAL SCHEMES

In the present version of TELEMAC-2D, only the depth-velocity option can be used when solving the propagation step. The celerity-velocity option used in versions prior to 3.0 of TELEMAC-2D was felt to be less useful and has been deleted. However, the keyword *PROPAGATION OPTION* is maintained in the dictionary file (but is deactivated) so as to allow for new options in the future.

The main choice concerns the scheme used for solving the advection step. To do this, the user must update the keyword *TYPE OF ADVECTION*. This keyword is a table of four integers that are related successively to the scheme used for advection of the velocity (U and V), depth (H), tracer and turbulent value (k and epsilon). If the model does not include any tracer or turbulence model, the user may simply give the first two values.

In release 6.0, the value concerning depth is ignored by TELEMAC-2D. The optimum numerical scheme is automatically selected by the code (conservative scheme)

Each integer may have a value between 1 and 14, corresponding to the following possibilities:

1: Method of characteristics.

2: Centred semi implicit scheme + SUPG.

3: Upwind explicit finite volume (referenced as 8 in previous releases)

4: N distributive scheme, mass-conservative (new in release 6.0)

5: PSI distributive scheme, mass-conservative (new in release 6.0)

6: PSI scheme on non conservative equation (obsolete)

7: Implicit N scheme on non conservative equation (obsolete)

13: Edge by edge implementation of scheme 3 (will work on tidal flats)

14: Edge by edge implementation of scheme 4 (will work on tidal flats)

Schemes 3 and 4 on one hand, and 13 and 14 on the other hand, are equal in 2 dimensions (they are not in 3D).

When using the finite volume scheme, the keyword *FINITE VOLUME SCHEME* specifies the type of scheme used. The available possibilities are:

- 0: Roe scheme (default value)
- 1: Kinetic order 1
- 2: Kinetic order 2

The stability of a PSI scheme (option 5 and 6) is conditioned by a Courant number lower than 1. When using this scheme, TELEMAC-2D verifies the Courant number for each point at each time step. If the Courant number is greater than 1, the software will automatically execute intermediate time steps in order to satisfy the stability condition. However, if the number of sub-iterations reaches 100, TELEMAC-2D will consider that solving advection is no more possible and the computation is stopped with an appropriate error message printed in the output listing.

Advection of the k-Epsilon model can only be done with the method of characteristics (fourth value of the keyword must be positioned at 1).

The default value is 1;5;1;1, which corresponds to the use of the method of characteristics in all cases, except for the depth for which the appropriate conservative scheme is selected by the code. Note that the value 5 in second position does not mean “Psi distributive scheme’ but is the value used by the previous version of TELEMAC-2D to select the conservative scheme for depth.

The default value is kept for compatibility of old studies, but a recommended value is : 1;5;4 or 4;5;4 when there are no dry zones, and 1;5;14 or 14;5;14 when there are tidal flats. Please note however that scheme 14 is dependent on edge numbering and thus may provide results showing differences between scalar and parallel mode. For dam break studies 14;5 is recommended.

Depending on the scheme used, mass-conservation may be improved by running sub-iterations. This involves updating the advection field for the same time step over several sub-iterations. During the first sub-iteration, the velocity field is given by the results obtained at the previous time step. This technique makes it possible to improve the way in which non-linearity is taken into account and considerably improves mass-conservation in the case of scheme 2. The number of sub-iterations is fixed by the keyword *NUMBER OF SUB-ITERATIONS FOR NON-LINEARITIES* which has a default value of 1.

The SUPG scheme may be configured using specific keywords (see section 7.2.1).

In TELEMAC-2D time discretization is semi-implicit. The various implicitation coefficients are given with the keywords *IMPLICITATION FOR DEPTH*, *IMPLICITATION FOR VELOCITY*, *IMPLICITATION FOR DIFFUSION OF VELOCITY* and in the case of computing tracer transport, *IMPLICITATION COEFFICIENT OF TRACERS*. The defaults values are generally adequate.

When solving the linearized system A X = B, TELEMAC-2D offers the possibility of mass-lumping on the mass matrices (Mh for depth, Mu and Mv for velocity) involved in computing the matrices AM1 for depth, and AM2 and AM3 for velocity. This technique means bringing some or all of the matrix on to the diagonal, and enables computation times to be shortened considerably. However, the solution obtained is very much smoothed. The rate of mass-lumping is fixed with the keywords *MASS-LUMPING ON H*, *MASS-LUMPING ON VELOCITY*. The value 1 indicates maximum mass-lumping (the mass matrices are diagonal) and the value 0 (default value) corresponds to normal processing without mass-lumping.

7.2.1. CONFIGURATION OF SUPG SCHEME

When the SUPG method is being used, the user must fix the type of upwind scheme required with the keyword *SUPG OPTION* which, like the keyword *TYPE OF ADVECTION*, is a table of 4 integers relating, in order, to the velocities, depth, tracer and k-Epsilon model.

The possible values are the following:

- 0: No upwind scheme.
- 1: Upwind scheme with the classic SUPG method, i.e. upwind scheme = 1.
- 2: Upwind scheme with the modified SUPG method, i.e. upwind scheme is equal to the Courant number.

In principle, option 2 is more accurate when the Courant number is less than 1 but must not be used for large Courant numbers. Thus, option 2 can be used only for model for which the Courant number is very low. If the Courant number can not be estimated, it is strongly recommended to use option 1 (which can be considered as more “universal”).

The configuration of the SUPG method concerns option 2 of the keyword *TYPE OF ADVECTION*.

7.3. SOLVING THE LINEAR SYSTEM

7.3.1. TYPE OF PROCESSING

When processing the linear system, it is possible to replace the original equations by a generalised wave equation obtained by eliminating the velocity from the continuity equation using a value obtained from the momentum equation. This technique increases calculation speed but has the disadvantage of smoothing the results.

The keyword used is *TREATMENT OF THE LINEAR SYSTEM*. The keyword may have the value 1 (original equations, which is the default value) or 2 (wave equation). It is important to stress that choosing option 2 automatically selects a number of other options: use of mass lumping on depth and velocities, and use of explicit velocity diffusion.

In most cases, Option 2 is recommended and offers the optimum in terms and CPU time.

When free surface wiggles are observed (for example in areas with strong bathymetry gradient), the best solution is to use the keyword *FREE SURFACE GRADIENT COMPATIBILITY* with a value lower than 1 (0.9 is recommended). The main consequence is a slightly altered compatibility between depth and velocity in the continuity equation but this option avoids the use of quasi-bubble triangle discretization which is more CPU and memory consuming.

7.3.2. SOLVER

During certain steps, the solver used for solving the systems of equations may be selected by means of the following keywords:

SOLVER: for the hydrodynamic propagation step.

SOLVER FOR DIFFUSION OF TRACERS: for the tracers diffusion step.

SOLVER FOR K-EPSILON MODEL: for solving the turbulence model system.

Each of these keywords may have a value of between 1 and 7. These values correspond to the following possibilities, and are all related to the conjugate gradient method:

- 1: Conjugate gradient method.

- 2: Conjugate residual method.
- 3: Conjugate gradient on normal equation method.
- 4: Minimum error method.
- 5: Squared conjugate gradient method.
- 6: BICGSTAB (stabilised biconjugate gradient) method.
- 7: GMRES (Generalised Minimum RESidual) method.

If the GMRES method is used, the dimension of the Krylov space must be specified with the appropriate keyword, i.e.:

SOLVER OPTION: for hydrodynamic propagation

SOLVER OPTION FOR TRACERS DIFFUSION: for tracer diffusion

By default, TELEMAC-2D uses the conjugate gradient on normal equation method (option 3) for solving the propagation step and the conjugate gradient method (option 1) for solving tracer diffusion and the turbulence model.

The GMRES method with a Krylov space dimension equal to 2 or 3 seems to fit most cases, when solving primitive equations, but the optimum value of this parameter generally increases with the mesh size.

The conjugate gradient is generally recommended for symmetric linear systems, thus when solving the wave equation or the diffusion equations.

7.3.3. ACCURACY

The linearized system is solved by an iterative method. It is therefore necessary to determine the accuracy that is to be achieved during the solving process and the maximum number of iterations permissible, to prevent the computation from entering unending loops if the required accuracy is not achieved.

Accuracy is specified with the following keywords:

SOLVER ACCURACY : defines the accuracy required during solution of the propagation step (default value 10-4).

ACCURACY FOR DIFFUSION OF TRACERS: defines the accuracy required during the computation of tracer diffusion (default value 10-6).

ACCURACY OF EPSILON : defines the accuracy required during the computation of diffusion and source terms step of the turbulent dissipation transport equations (default value 10-9).

ACCURACY OF K: defines the accuracy required during the diffusion and source terms step of the turbulent energy transport equation (default value 10-9).

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

MAXIMUM NUMBER OF ITERATIONS FOR K AND EPSILON : defines the maximum permissible number of iterations when solving the diffusion and source terms step of the k-Epsilon transport equations (default value 50).

MAXIMUM NUMBER OF ITERATIONS FOR DIFFUSION OF TRACERS: defines the maximum permissible number of iterations when solving the tracers diffusion step (default value 60).

MAXIMUM NUMBER OF ITERATIONS FOR SOLVER: defines the maximum permissible number of iterations when solving the propagation step (default value 100).

The user may obtain information on the solvers by activating the keywords *INFORMATION ABOUT SOLVER* and, if the k-Epsilon turbulence model is used, *INFORMATION ABOUT K-EPSILON MODEL* .This information is provided in the listing printout and may be of the following two types:

Either the process has converged before reaching the maximum permissible number of iterations, and in this case TELEMAC-2D provides the number of iterations actually run and the accuracy achieved.

Or the process has not converged quickly enough. TELEMAC-2D then displays the message “MAXIMUM NUMBER OF ITERATIONS REACHED” and the accuracy actually achieved. In certain cases, and if the number of iterations is already positioned at a high value (e.g. more than 120), the convergence may still be improved by decreasing the time step or by modifying the mesh.

7.3.4. CONTINUITY CORRECTION

Residual mass errors (of the order of a few percent) may appear when using boundary conditions with imposed depth (case of a river downstream). Indeed the continuity equation is not solved for these points and is replaced by the equation depth = imposed value. Therefore, the resultant discharge is not properly computed and leads to error. The keyword *CONTINUITY CORRECTION* helps in correcting the velocity at these points so that the overall continuity is verified. This correction has enabled the error to be divided by as much as 1000.

This correction is quite time-consuming and should be used only if the outlet discharge needs to be known precisely (in the case of calibration for example).

7.3.5. PRECONDITIONING

When solving a system of equations by a conjugate gradient method, convergence can often be hastened by means of preconditioning.

TELEMAC-2D offers several possibilities for preconditioning. These are selected with the keywords *PRECONDITIONING*, *PRECONDITIONING FOR DIFFUSION;OF TRACERS* and *PRECONDITIONING FOR K-EPSILON MODEL*.

The possibilities may be different depending on the keywords.

The keyword *PRECONDITIONING* concerns the propagation solution step, and can have only the following values:

- 0: No preconditioning.
- 2: Diagonal preconditioning (default value).
- 3: Block diagonal preconditioning
- 5: Diagonal preconditioning with absolute value.
- 7: Crout preconditioning per element.
- 11: Gauss-Seidel EBE preconditioning (not convenient for parallelism).

The keyword *PRECONDITIONING FOR DIFFUSION OF TRACERS* concerns the tracer diffusion solution step, and can have only the following values:

- 0: No preconditioning.
- 2: Diagonal preconditioning (default value).
- 5: Diagonal preconditioning with absolute value.
- 7: Crout preconditioning per element.
- 11: Gauss-Seidel EBE preconditioning.

The keyword *PRECONDITIONING FOR K-EPSILON MODEL* concerns the turbulence model solution step, and can have only the following values:

- 0: No preconditioning.
- 2: Diagonal preconditioning (default value).
- 3: Block diagonal preconditioning
- 5: Diagonal preconditioning with absolute value.
- 7: Crout preconditioning per element.
- 11: Gauss-Seidel EBE preconditioning.

Certain options of preconditioning can be cumulated, namely the diagonal ones with the others. As the base values are integers, two options are cumulated by assigning the keyword the value of the product of the two options to be cumulated.

The block-diagonal preconditioning can be used only when solving the primitive equations (it is not valid with the wave equation)

In addition, when the propagation step is being solved, convergence may possibly be improved by modifying the initial value taken for H at the start of the solving process. The user may then assign the keyword *INITIAL GUESS FOR H* any of the following values:

- 0: Initial value of DH = Hn+1 - Hn nil.
- 1: Initial value of DH equal to the value of DH at the previous time step (default value).
- 2: DH = 2DHn - DHn-1 in which DHn is the value of DH at the previous time step and DHn-1 the value of DH two time steps before. This is in fact an extrapolation.

The same process may be used for the velocity by using the keyword *INITIAL GUESS FOR U*. The possibilities are the same as before, but apply to U (or V) and not to the increase of U (or V).

7.3.6. C-U PRECONDITIONING

When solving the linear system, C-U preconditioning consists of replacing the unknown depth by the celerity. This technique was used automatically in the previous versions of the software and is now configurable as an option with the keyword *C-U PRECONDITIONING*. The default value is YES. This technique is very useful in sea modelling but not in river modelling. It is not activated with the wave equation.

7.4. COURANT NUMBER MANAGEMENT

During a model simulation, the Courant number value (number of grid cells crossed by a water particle during a time step) considerably influences the quality of the results. Irrespective of numerical schemes with a stability condition on the Courant number, experience shows that result quality decreases if the Courant number is above 7 or 8. Yet it is not so easy to estimate the value of the Courant number - especially in sea models with a large tidal range. To help, TELEMAC-2D allows the user to check the Courant number during computation: the software automatically executes intermediate time steps so that the Courant number keeps below a given value.

This function is activated using the keyword *VARIABLE TIME STEPS* (Default value NO) and the maximum Courant number value can be specified using the keyword *DESIRED COURANT NUMBER* (default value: 1).

It should be stressed that when a variable time step is used, sampling from the results file and control listing is no longer regular in time, as it depends directly on the time step value.

7.5. TIDAL FLATS

TELEMAC-2D offers several processing options for tidal flat areas.

First of all, if the user is sure that the model will contain no tidal flats throughout the simulation, these may be deactivated by assigning NO to the keyword *TIDAL FLATS* (the default value is YES). This may mean that computational time can be saved.

Tidal flats can be processed in three different ways and several possibilities are offered concerning the treatment of negative depths.

In the first case, the tidal flats are detected and the free surface gradient is corrected.

In the second case, the tidal flat areas are removed from the computation. Exposed elements still form part of the mesh but any contributions they make to the computations are cancelled by a so-called "masking" table. The data structure and the computations are thus formally the same to within the value of the masking coefficient. However, in this case, mass-conservation may be slightly altered.

In the third case, processing is done in the same way as in the first case, but a porosity term is added to half-dry elements. Consequently, the quantity of water is changed and is no longer equal to the depth integral over the entire domain but to the depth integral multiplied by the porosity. The user can modify the porosity value determined by the processing in the CORPOR subroutine.

The type of processing is chosen with the keyword *OPTION FOR THE TREATMENT OF TIDAL FLATS* which may have a value of 1 or 2, the default value being 1.

The treatment of the negative depths can be specified using the keyword *TREATMENT OF NEGATIVE DEPTHS*. Value 1 (default value) is the previously only option consisting of smoothing the negative depths in a conservative way. The option 2 (new in release 6.0) is a flux limitation that ensures strictly positive depths. This must be preferably coupled with the new advection schemes able to cope with tidal flats. Value 0 means no treatment.

The keyword *THRESHOLD FOR NEGATIVE DEPTHS* (default 0) is used only with the treatment number 1 of negative depths. It specifies the limit of the unchanged value. For example, *THRESHOLD FOR NEGATIVE DEPTHS* = -0.01 means that depths greater than –1 cm will be left unchanged.

In certain cases, it may be advisable to limit the lower water depth value. The most common case involves eliminating negative values of H. To do this the user assigns the value YES to the keyword *H CLIPPING* (default value NO). The keyword *MINIMUM VALUE OF DEPTH* which has a default value of 0, is used to fix the threshold below which clipping is performed. However, it should be borne in mind that this latter option leads to an increase in the mass of water as it eliminates negative water depths.

7.6. OTHER PARAMETERS

7.6.1. MATRIX STORAGE

TELEMAC-2D includes 2 different methods of matrix storage: a classical method and assembled elementary storage method. The second is faster (about 20%) in most cases. The choice between the two storage methods can be made using the keyword *MATRIX STORAGE*, with the following values:

- 1: classical method
- 3: edge-based storage method (default and recommended value)

7.6.2. MATRIX-VECTOR PRODUCT

Two matrix-vector product methods are included in TELEMAC-2D: a classical method for the multiplication of a vector by a non-assembled matrix and a new method of frontal multiplication with an assembled matrix. The keyword *MATRIX-VECTOR PRODUCT* switches between the two methods:

1: multiplication of a vector by a non-assembled matrix (default and recommended value)

2: frontal multiplication with an assembled matrix

8. TRACERS TRANSPORT

8.1. AVAILABLE POSSIBILITIES

With TELEMAC-2D it is possible to take into account the transport of a number of non-buoyant tracers (i.e. one whose presence has no effect on the hydrodynamics), which may or may not be diffused.

The tracer transport computation is activated with the keyword *NUMBER OF TRACERS* (default value 0) which gives the number of tracers taken into account during the simulation. In addition, it is necessary to give the name and the unit of each tracer. This information is given by the keyword *NAMES OF THE TRACERS*. The names are given in 32 characters (16 for the name itself and 16 for the unit). For example, for 2 tracers (the character - means for a blank):
NAMES OF TRACERS = 'SALINITY--------KG/M3-----------',
'NITRATE---------MG/L------------'

The name of the tracers will appear in the result files.

Obviously, it is necessary to add the appropriate specifications in the keyword *VARIABLES FOR GRAPHIC PRINTOUTS*. The name of the variables is a letter T following by the number of tracer. For example 'T1,T3' stand for first and third tracer. In release 6.0 It is possible to use the character * as wildcards (replace any character). T* stand for T1 to T9, and T** stand for T10 to T99.

N.B.: TELEMAC-2D offers the possibility of taking into account density effects when the tracer used is the salinity expressed in kg/m3. In this case it is necessary to position the keyword *DENSITY EFFECTS* at YES (default value NO) and indicate the mean temperature of the water in degrees Celsius using the keyword *MEAN TEMPERATURE* which has a default value of 20. In that case, the first tracer must be the salinity.

8.2. PRESCRIBING INITIAL CONDITIONS

If the initial value of the tracers is constant throughout the domain (for example no tracer), it is simply a question of placing the keyword *INITIAL VALUES OF TRACERS* with the required value in the steering file. The number of supplied values must be equal to the number of declared tracers

In more complex cases, it is necessary to work directly in the CONDIN subroutine, in a similar manner to that described in the section dealing with the initial hydrodynamic conditions.

If a computation is being continued, the initial state of the tracers corresponds to that of the last time step stored in the continuation file (if the continuation file does not contain any information concerning a particular tracer, TELEMAC-2D then uses the value assigned to the keyword *INITIAL VALUES OF TRACERS*).

8.3. PRESCRIBING BOUNDARY CONDITIONS

Boundary conditions are prescribed in the same way as hydrodynamic conditions.

The type of boundary condition will be given by the value of LITBOR in the boundary conditions file (see sections 4.2.2 and 4.2.3).

In the case of an inflowing open boundary with prescribed tracer (LITBOR = 5), the value of the tracer may be given in various ways:

If the value is constant along the boundary and in time, it is provided in the steering file by the keyword *PRESCRIBED TRACERS VALUES* . This is a table of real numbers for managing several boundaries and several tracers (100 at most). The numbering principle is the same as that used for the hydrodynamic boundary conditions. The values specified by the keyword cancel the values read from the boundary conditions file. The order of this table is: first tracer at the first open boundary, first tracer at the second open boundary, ….., second tracer at the first open boundary, second tracer at the second open boundary… .

If the value is constant in time but varies along the boundary, it will be given directly by the variable TBOR from the boundary conditions file.

If the value is constant along the boundary but varies in time, the user must specify this with the function TR or open boundaries file. Programming is done in the same way as for the functions VIT, Q and SL (see 4.2.5).

If the variable is time- and space-dependent, the user must specify this directly in the BORD subroutine, in the part concerning the tracer (see 4.2.7).

8.4. MANAGING TRACER SOURCES

TELEMAC-2D offers the possibility of placing tracer sources at any point of the domain.

The user positions the various sources with the keywords *ABSCISSAE OF SOURCES* and *ORDINATES OF SOURCES*. These are tables of real numbers, giving the source coordinates in metres. In fact, TELEMAC-2D will position a source at the closest mesh point to that specified by these keywords. The program itself will determine the number of sources as a function of the number of values given to each keyword.

At each source, the user must indicate the discharge and the value of the tracer. The discharge is specified in m3/s using the keyword W*ATER DISCHARGE OF SOURCES* and the value of the tracer by the keyword *VALUES OF THE TRACERS AT THE SOURCES*. However, if these two variables are time-dependent, the user can then program the two functions DEBSCE (source discharge) and TRSCE (value of tracer at source). It is also possible to use a specific file to define the time evolution of the sources: the source file (keyword *SOURCES FILE*). This file has exactly the same structure as the one of the liquid boundary file. An example is presented here with 2 sources and 2 tracers. Between 2 given times, the values are obtained by linear interpolation.

```
#
#  TIME-DEPENDENT DISCHARGES AND TRACERS AT SOURCES 1 AND 2
#
#  T IS TIME
#
#  Q(1) IS DISCHARGE AT SOURCE 1
#  Q(2) IS DISCHARGE AT SOURCE 2
#
#  TR(1) IS TRACER 1 AT SOURCE 1
#  TR(2) IS TRACER 2 AT SOURCE 1
#  TR(3) IS TRACER 1 AT SOURCE 2
#  TR(4) IS TRACER 2 AT SOURCE 2
#
#  GENERAL FORMULA FOR TRACERS : TR(I) FOR
#  I=RANK OF TRACER + NUMBER OF TRACERS * (RANK OF SOURCE - 1)
#
T    Q(1)   TR(1)   TR(2)   Q(2)   TR(3)  TR(4)
s    m3/s    °C      °C     m3/s    °C     °C
0.    0.     99.     20.     0.     30.    40.
```

```
2.   1.      50.     20.      2.      30.     20.
4.   2.      25.     80.      4.      30.     20.
```

With TELEMAC-2D it is also possible to take into account a momentum flux from the sources. By default, the value used is that computed at the source point with no added momentum. The user may prescribe a particular velocity. If this is constant throughout the simulation, the value may be given with the keywords *VELOCITIES OF THE SOURCES ALONG X* and *VELOCITIES OF THE SOURCES ALONG Y.* If not, the user must program the two functions VUSCE (for the velocity along X) and VVSCE (for the velocity along Y). In both these functions, the time, source number and depth of water are available to the user.

If source terms are to be taken into account for the creation or decay of the tracer, these must be introduced in the DIFSOU subroutine.

From a theoretical point of view, complete mass conservation can only be ensured if the source is treated as a Dirac function and not as a linear function. The type of treatment is indicated by the user with the keyword *TYPE OF SOURCE*, which may have a value of 1 (linear function, default value) or 2 (Dirac function). It should be noted that in the second case, the solutions are of course less smoothed.

It is possible to manage sources without simulating tracer transport.

8.5. NUMERICAL SPECIFICATIONS

The way of treating advection is specified in the third value of the keyword *TYPE OF ADVECTION*. The possibilities are the same as for velocity.

The user can also use the real keyword *IMPLICITATION COEFFICIENT OF TRACERS* (default value 0.6) in order to configure the implicitation values in the cases of semi-implicit schemes.

When solving the tracer transport equations, the user can choose whether or not to take into account diffusion phenomena, using the logical word *DIFFUSION OF TRACERS* (default value YES).

Furthermore, the tracers diffusion coefficient should be specified using the real keyword *COEFFICIENT FOR DIFFUSION OF TRACERS* (default value 10^{-6}). This parameter is the same for all tracers. This parameter has a very important influence on tracer diffusion in time. As for velocity diffusion, a time- or space-variable tracer diffusion coefficient should be programmed directly in the CORVIS subroutine.

As for velocity diffusion, the user can configure the type of solution he requires for the diffusion term. To do this, he should use the real keyword *OPTION FOR THE DIFFUSION OF TRACERS* with the following values:

- 1: treatment of the term of type: $div\left(\nu\ \overrightarrow{grad}\ (T)\right)$ (default value)
- 2: treatment of the term of type: $\frac{1}{h}div\left(h\nu\ \overrightarrow{grad}\ (T)\right)$ (good tracer mass conservation but critical in the case of tidal flats).

This page is intentionally left blank

9. DROGUES AND LAGRANGIAN DRIFTS

9.1. DROGUE DISPLACEMENTS

During a hydrodynamic simulation, TELEMAC-2D offers the possibility of monitoring the tracks followed by certain particles (drogues) introduced into the fluid from outflow points. The result is produced in the form of a Serafin format file containing the various positions of the drogues in the form of a degenerated mesh.

9.1.1. CONFIGURATION OF THE SIMULATION

In order to use drogues, the user must program the steering file and BORD subroutine, which must be inserted in the FORTRAN file.

Three types of information must be indicated in the steering file. First of all, the user must indicate the number of drogues using the keyword *NUMBER OF DROGUES* which has the default value 0. Secondly, the user must provide the number of the file in which TELEMAC-2D is to store the successive positions of the drogues. To do this, he uses the keyword *BINARY RESULTS FILE* (it is thus impossible, during the same simulation, to use drogues and store the results in a binary file other than the standard results file). Finally, the user can configure the printout period using the keyword *PRINTOUT PERIOD FOR DROGUES* (default value 1).This value is expressed in number of time steps, and is completely independent of the printout period for other results from TELEMAC-2D.

After inserting the FLOT subroutine in the FORTRAN file, the user must modify it in order to indicate the release time step and end-of-monitoring time step for each drogue, together with the coordinates of the release point. By default, the drogues are all released at the start of the simulation and the end-of-monitoring time corresponds to the end of the simulation. If the release point coordinates are outside the domain, the run is interrupted and an error message displayed. In addition, if a drogue leaves the domain during the simulation, it is of course no longer monitored but its previous track remains in the results file for consultation.

The following example shows how the steering file and FLOT subroutine are programmed in the case of two drogues released 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 start
DEBFLO(2) = 100       of the simulation up to the 400th time step.
FINFLO(1) = 400       The second drogue is released at the 100th
FINFLO(2) = NIT       time step up to the end of the simulation.
XFLOT(1,1) = 321.
XFLOT(1,2) = 351.     The coordinates of the two drogues are
YFLOT(1,1) = 37.      specified at the moment of release.
YFLOT(1,2) = 95.
```

9.1.2. EXPLOITATION OF RESULTS

The results are stored in the binary file during the simulation. This file is in Serafin standard, and can thus be read using RUBENS.

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

The positions are displayed using a “mesh” type graph. Each position is shown in the form of a node. The successive positions of the drogue are identified by the node numbering system. Nodes are numbered in the following way: at the beginning of the file, all positions of drogue 1, then all positions of drogue 2, etc.

For example, if a simulation is run for 10 time steps with 3 drogues, the following numbering will be obtained:

- From 1 to 10: successive positions of drogue 1.
- From 11 to 20: successive positions of drogue 2.
- From 21 to 30: successive positions of drogue 3.

9.2. LAGRANGIAN DRIFTS

Computing Lagrangian drifts involves computing the displacement of all mesh points between two given instants. Typically, these two instants may be two consecutive high tides.

To run such a computation, it is necessary to program the steering file and FORTRAN file.

In the steering file, the user must firstly provide the number of drifts required using the keyword *NUMBER OF LAGRANGIAN DRIFTS* (default value 0). This value corresponds to the number of pairs (starting and ending times) for which the Lagrangian drifts are computed. Secondly, the user must include the letters A and G in the list assigned to the keyword *VARIABLES FOR GRAPHIC PRINTOUTS*. These two letters correspond to the drift displacements along X and Y.

As far as the FORTRAN file is concerned, the user must insert the LAGRAN subroutine, in which it is necessary to define the instants at which each computation is to start and end, in the form of time step numbers.

The drift computation results are stored directly in the TELEMAC-2D results file in the form of two scalar variables entitled DRIFT_ALONG_X and DRIFT_ALONG_Y. Given the possible discretization that may occur in printing out graphical results, the rule for introduction in the results file is the following:

If none of the drift computations is completed at the time step considered, the two variables are set to zero.

Otherwise, the two variables contain the results of the most recently completed drift computation.

This means on the one hand that two drifts may not be completed at the same time step, and on the other hand that between two ends of drift computations, a record must be made in the results file (otherwise the result of the first computation is lost).

Lastly, if a drift leaves the domain, the corresponding computation is interrupted and the result reset at zero for this node.

The following example (steering file and FORTRAN file) carries out two drift computations. The first begins at time step 10 and finishes at time step 50. The second begins at time step 15 and finishes at time step 40.

- In the steering file:

```
NUMBER OF LAGRANGIAN DRIFTS   = 2
VARIABLES FOR GRAPHIC PRINTOUTS = 'U,V,H,A,G'
GRAPHIC PRINTOUT PERIOD = 1
```

• In the LAGRAN subroutine of the FORTRAN file:

```
DEBLAG(1) = 10
FINLAG(1) = 50
DEBLAG(2) = 15
FINLAG(2) = 40
```

In this example, the variables DRIFT_ALONG_X and DRIFT_ALONG_Y of the results file will contain the following values:

- From time steps 1 to 39: 0 values (no finished drift computation).
- From time steps 40 to 49: results of the second drift computation.
- Time step 50: results of the first drift computation.

10. WEIRS AND CULVERTS

10.1. WEIRS

Weirs are considered as singularities. The number of weirs is specified by the keyword *NUMBER OF WEIRS* (default value 0). Information about weirs is given in the *FORMATTED DATA FILE 1*. If there are also culverts, information about them should be placed after.

A weir must be prepared in the mesh and consists of two boundary lines which are actually linked by the weir. In principle, these boundaries should be sufficiently far apart, upstream and downstream of the weir. The upstream and downstream boundary points should correspond 1 to 1, and the distance between two points should be the same on both sides. The following file gives an example of two weirs (the comments are part of the file):

```
Nb of culverts Option for tangential velocity
    2             0
---------------------------- singularity 1
Nb of points for 1 side
11
Points side 1
71 72 73 74 75 76 77 78 79 80 41
Points side 2
21 20 19 18 17 16 15 14 13 12 11
level of the dike
1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8 1.8
flowrate coefficients
 .4  .4  .4  .4  .4  .4  .4  .4  .4  .4  .4
---------------------------- singularity 2
Nb of points
11
Points side 1
111 112 113 114 115 116 117 118 119 120 81
Points side 2
61 60 59 58 57 56 55 54 53 52 51
Level of the dike
1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6 1.6
flowrate coefficient
 .4  .4  .4  .4  .4  .4  .4  .4  .4  .4  .4
```

Line 2 indicates the number of weirs and then an option for the treatment of tangential velocities on the weir, with the following meanings:

0: the velocities are nil (recommended option)

1: the velocities will be calculated with the Chézy formula (as a function of the local free surface inclination)

For each weir, it is then necessary to indicate: the number of points for the first side of the weir (line 5 for the first weir) and the list of their numbers in the boundary conditions file (line 7 for the first weir), following the numbering used in that file.

The numbers of their twin points on the second side should be given on line 9 in the reverse order. On line 11, the level of the weir is specified for each couple of points and at line 13 the discharge coefficient noted μ. All this data is repeated for all weirs.

The formulae used to calculate the discharge for each point are the following:

Unsubmerged weir:

$$Q = \mu \sqrt{2g}\ (\textit{upstream level} - \textit{threshold level})^{3/2}$$

Submerged weir:

$$Q = \frac{1}{\frac{2}{3}\sqrt{\frac{1}{3}}}\ \mu \sqrt{2g}\ (\text{downstream level} - \text{threshold level}) . \sqrt{(\text{upstream level} - \text{downstream level})}$$

The weir is not submerged if:

$$\text{Upstreamlevel} < \frac{\text{threshold level} + 2 * \text{upstream level}}{3}$$

Depending on the shape and roughness of the weir, the value of μ is between 0.4 and 0.5. However, the above formulae neglect the velocity in the computation of the upstream head. If this is not the case, the value of μ may be higher.

If the user want to modify the different laws, it is possible to modify the appropriate subroutines (LOIDEN.f and LOINOY.f)

10.2. CULVERTS

As for weirs, the keyword *NUMBER OF CULVERTS* specifies the number of culverts to be treated. Culverts or siphons are described as couples of points between which flow may occur, as a function of the respective water level at these points. Beforehand, each culvert inflow and outflow has to be described as a source point with the usual keywords, for example:

ABSCISSAE OF SOURCES: 100. ; 400.
ORDINATES OF SOURCES: 100. ; 100.
WATER DISCHARGE OF SOURCES: 0. ; 0.

And possibly:

VELOCITIES OF THE SOURCES ALONG X: 0. ; 0.
VELOCITIES OF THE SOURCES ALONG Y: 0. ; 0.

Information about culvert characteristics is stored in the *FORMATTED DATA FILE 1*. If there are also weirs, weir information must be placed before.

The following file gives an example of a culvert:

```
Relaxation   Number of siphons

  0.2            1
I1  I2  d1  d2   CE1  CE2  CS1   CS2   S12  L12  z1   z2  a1   a2
 1   2  0.  90.   0.5   0.5  1.0      1.0     20.  0.2  0.3  0.1  0.   90.
```

The relaxation coefficient is initially used to prescribe the discharge in the culvert on a progressive basis, in order to avoid the formation of an eddy. I1 and I2 are the numbers of each end of the culvert in the source point numbering system (which is given implicitly by the keywords *ABSCISSAE OF SOURCES*, etc.).

The culvert discharge is given by the following formula (with flow going from 1 to 2):

$$Q = S_{12}\left(\frac{2g\,(\text{upstream level} - \text{downstream level})}{CE_2 + CE_1 + L_{12}}\right)$$

S12 is the cross-section of the pipe. CS1 and CS2 are the head loss coefficients of 1 and 2 when they are operating as outlets. CE1 and CE2 are the head loss coefficients of 1 and 2 when they are operating as inlets. L12 is the linear head loss coefficient, generally equal to $\lambda \frac{L}{D}$ where L is the length of the pipe, D its diameter and l the friction coefficient.

z1 and z2 are the levels of the tapers.

a1 and a2 are the angles that the pipe makes with respect to the bottom, in degrees. For a vertical intake, the angle with the bottom will therefore be 90°.

d1 and d2 are the angles with respect to the x axis. They are used when allowing for the current direction at source or sink point (in cases where the keywords *VELOCITIES OF THE SOURCES ALONG X* (and Y) are given).

11. OTHER CONFIGURATIONS

11.1. MODIFICATION OF BOTTOM TOPOGRAPHY (CORFON)

Bottom topography may be introduced at various levels, as was stated in section 3.5.

TELEMAC-2D offers the possibility of modifying the bottom topography at the start of a computation using the CORFON subroutine. This is called up once at the start of the computation and enables the value of variable ZF to be modified at each point of the mesh. To do this, a number of variables such as the point coordinates, the element surface value, connectivity table, etc, are made available to the user.

By default, the CORFON subroutine carries out a number of bottom smoothings equal to FILTER, i.e. equal to the number specified by the keyword *BOTTOM SMOOTHINGS*.

The CORFON subroutine is not called up if a computation is being continued. This avoids having to carry out several bottom smoothings or modify the bottom topography during the computation.

11.2. MODIFYING COORDINATES (CORRXY)

TELEMAC-2D also offers the possibility of modifying the mesh point coordinates at the start of a computation. This means, for example, that it is possible to change the scale (from that of a reduced-scale model to that of the real object), rotate or shift the object.

The modification is made in the CORRXY subroutine (BIEF library), which is called up at the start of the computation. This subroutine is empty by default and gives an example of programming concerning a change of scale and origin, in the form of commented statements.

It is also possible to specify the coordinates of the origin point of the mesh. This is done using the keyword *ORIGIN COORDINATES* which specify 2 integers. These 2 integers will be transmitted to the results file in the Serafin format, for a use by post-processors for superimposition of results with digital maps (coordinates in meshes may be reduced to avoid large real numbers). These 2 integers may also be used in subroutines under the names I_ORIG and J_ORIG. Otherwise they are not yet used.

11.3. SPHERICAL COORDINATES (LATITU)

If a simulation is being performed over a large domain, TELEMAC-2D offers the possibility of running the computation with spherical coordinates.

This option is activated when the keyword *SPHERICAL COORDINATES* is positioned at YES (default value NO). In this case, TELEMAC-2D calls a subroutine named LATITU at the beginning of the computation. This calculates a set of tables depending on the latitude of each point. To do this, it uses the Cartesian coordinates of each point provided in the geometry file, and the latitude of the point of origin of the mesh provided by the user in the steering file with the keyword *LATITUDE OF ORIGIN POINT*.

It is also necessary to indicate the angle between the geographic north and the Y-axis of the model. This information is provided with the keyword NORTH. This gives the angle (in degrees) between the north and Y-axis, expressed positively in the trigonometric direction (the default value of the keyword is 0).

In its present version, TELEMAC-2D assumes that the mesh coordinates are given in accordance with Mercator’s projection.

The LATITU subroutine (BIEF library) may be modified by the user to introduce any other latitude-dependent computation.

11.4. ADDING NEW VARIABLES (NOMVAR-TELEMAC2D AND PRERES_TELEMAC2D)

A standard feature of TELEMAC-2D is the storage of certain computed variables. In certain cases, the user may wish to compute other variables and insert them in the results file (the number of variables is currently limited to four).

TELEMAC-2D has a numbering system in which, for example, the array containing the Froude number has the number 7. The new variables created by the user may have the numbers 23, 24, 25 and 26.

In the same way, each variable is identified by a letter in the keyword *VARIABLES FOR GRAPHIC PRINTOUTS*. The new variables are identified by the letters N, O, R and Z, which correspond respectively to the numbers 23, 24, 25 and 26.

At the end of NOMVAR_TELEMAC2D, it is possible to change the abbreviations (mnemonics) used for the keywords *VARIABLES FOR GRAPHIC PRINTOUTS* and *VARIABLES FOR LISTING PRINTOUTS*. Sequences of 8 letters may be used. Consequently, the variables must be separated by spaces, commas or semi-colons in the keywords, e.g.:

VARIABLES FOR GRAPHIC PRINTOUTS :'U,V,H;B'

In the software data structure, these four variables correspond to the tables PRIVE%ADR(1)%P%R(X), PRIVE%ADR(2)%P%R(X) PRIVE%ADR(3)%P%R(X) et PRIVE%ADR(4)%P%R(X) (in which X is the number of points in the mesh). These may be used in several places in the programming, like all TELEMAC variables. For example, they may be used in the subroutines CORRXY, CORSTR, BORD, etc. If a PRIVE table is used for programming a case, it is essential to check the value of the keyword *NUMBER OF PRIVATE ARRAYS*. This value fixes the number of tables used (0, 1, 2, 3 or more) and then determines the amount of memory space required. The user can also access the tables via the aliases PRIVE1, PRIVE2, PRIVE3 and PRIVE4.

An example of programming using the second PRIVE table is given below. It is being initialised with the value 10.

```
DO I=1,NPOIN
  PRIVE%ADR(2)%P%R(I) = 10.D0
ENDDO
```

New variables are programmed in two stages:

- Firstly, it is necessary to define the name of these new variables by filling in the NOMVAR_TELEMAC2D subroutine. This consists of two equivalent structures, one for English and the other for French. Each structure defines the name of the variables in the results file that is to be generated and then the name of the variables to be read from the previous computation if this is a continuation. This subroutine may also be modified when, for example, a file generated with the English version of TELEMAC-2D is to be continued with the French version. In this case, the TEXTPR table of the French part of the subroutine must contain the English names of the variables.

- Secondly, it is necessary to modify the PRERES_TELEMAC2D subroutine in order to introduce the computation of the new variable(s). The variables LEO, SORLEO, IMP, SORIMP are also used to determine whether the variable is to be printed in the printout file or in the results file at the time step in question.

11.5. ARRAY MODIFICATION OR INITIALIZATION

When programming TELEMAC-2D subroutines, it is sometimes necessary to initialize a table or memory space to a particular value. To do that, the BIEF library furnishes a subroutine called FILPOL that lets the user modify or initialize tables in particular mesh areas.

A call of the type CALL FILPOL (F,C, XSOM, YSOM, NSOM, MESH, XMESH) fills table F with the C value in the convex polygon defined by NSOM nodes (coordinates XSOM, YSOM). The variables MESH and XMESH are needed for the FILPOL subroutine but have no meaning for the user.

11.6. VALIDATING A COMPUTATION (VALIDA)

The structure of the telemac-2d software offers a point of entry for validating a computation, in the form of a subroutine named VALIDA which has to be filled by the user in accordance with each particular case. Validation may be carried out either with respect to a reference file (which is therefore a file of results from the same computation that is taken as reference, the name of which is supplied by the keyword *REFERENCE FILE*), or with respect to an analytical solution that must then be programmed entirely by the user.

When using a reference file, the keyword *REFERENCE FILE FORMAT* specifies the format of this binary file (SERAFIN by default).

The VALIDA subroutine is called at each time step when the keyword VALIDATION has the value YES, enabling a comparison to be made with the analytical solution at each time step. By default, the VALIDA subroutine only makes a comparison with the last time step. The results of this comparison are given in the control listing.

11.7. CHANGING THE TYPE OF A BOUNDARY CONDITION (PROPIN_TELEMAC2D)

During a simulation, the type of boundary condition is generally fixed and, in the case of TELEMAC-2D, is provided by the boundary conditions file. However, in certain cases, it may be necessary to change the type of boundary conditions during the computation (section of a river subject to tidal effects where the current alternates, for instance).

This change in boundary condition type must be made in the PROPIN_TELEMAC2D subroutine.

N.B: modifying PROPIN_TELEMAC2D is a difficult operation and must be done with great care!

11.8. COUPLING

The principle of coupling two (or in theory more) simulation modules involves running the two calculations simultaneously and exchanging the various results at each time step. For example, the following principle is used to link a hydrodynamic module and a sediment transport module:

The two codes perform the calculation at the initial instant with the same information (in particular the mesh and bottom topography).

The hydrodynamic code runs a time step and calculates the depth of water and velocity components. It provides this information to the sediment transport code.

The sediment transport code uses this information to run the solid transport calculation over a time step and thus calculates a change in the bottom.

The new bottom value is then taken into account by the hydrodynamic module at the next time step, and so on.

Only the SISYPHE module can be linked in the current version of the code. The time step used for the two calculations is not necessarily the same and is managed automatically by the coupling algorithms and the key-word *COUPLING PERIOD*.

This function requires two keywords. The keyword *COUPLING WITH* indicates which simulation code is to be linked to TELEMAC-2D. The value of this keyword can only be INTER-SISYPHE at present, which means internal coupling with Sisyphe (previous versions included a coupling with files). An additional necessary key-word is the *SISYPHE PARAMETER FILE*. It will be used by the perl scripts to prepare the Sisyphe computation.

Except the fact that data, e.g. the bottom and velocities, the printout periods, are given by Telemac-2D to Sisyphe, The Sisyphe data file must enable a standalone computation. The FORTRAN FILE of both programmes may be used and will be compiled independently (make sure that the Sisyphe Fortran file contains no main program).

For further information, e.g. to have information on the quasi-steady state procedure, refer to Sisyphe documentation.

The keyword *COUPLING WITH* is also used if the computation has to generate the appropriate files necessary to make a water quality simulation. In that case, it is necessary to specify *COUPLING WITH = DELWAQ*. Please refer to appendix 4 for all information concerning the communication with DELWAQ.

11.9. ASSIGNING A NAME TO A POINT

During certain types of processing, for example a Fourier series analysis (see 11.10), it may be useful to assign a name to a point. This is easy to do by using the two keywords *LIST OF POINTS* and *NAMES OF POINTS*. The former provides a list of node numbers (100 max) in the general numbering system, and the second provides the corresponding names (string of 32 characters max).

For example, in the case of a model of the Channel, point 3489 corresponds to the port of Saint-Malo and point 56229 to the port of Cherbourg. In this case, the names will be assigned as follows

LIST OF POINTS: 3489 ; 56229
NAMES OF POINTS: 'SAINT MALO';'CHERBOURG'

11.10. FOURIER ANALYSIS

TELEMAC-2D allows the user to analyse free surface variations in order to determine the phase and amplitude of one or more waves. This can only be done if the mean level is zero. Amplitudes and phases are supplied for each point and for each period.

This function is activated by the keyword *FOURIER ANALYSIS PERIODS* and provides a list of the analysis periods (e.g. the periods of tide-induced waves that are to be studied). The results are supplied directly at the last time step in the results file with the names AMPLITUDE1, AMPLITUDE2, etc. for the amplitudes and PHASE1, PHASE2, etc. for the phases. The user estimates the minimum duration of the simulation. The keyword *NUMBER OF FIRST TIME STEP FOR GRAPHIC PRINTOUTS* can be used to reduce the size of the results file.

It is also necessary to specify the time range using the keyword *TIME RANGE FOR FOURIER ANALYSIS* associated with 2 real values: the starting time in seconds and the ending time in seconds separated by a semicolon. If this keyword is left with its default values (0;0) the computation wills stop with an appropriate error message.

12. PARALLELISM

TELEMAC-2D is generally run on single-processor computers of the workstation type. When simulations call for high-capacity computers and in the absence of a super-computer, it may be useful to run the computations on multi-processor computers or clusters of workstations. A parallel version of TELEMAC-2D is available for use with this type of computer architecture.

The parallel version of TELEMAC-2D uses the MPI library, which must therefore be installed beforehand. The interface between TELEMAC-2D and the MPI library is the parallel library common to all modules of the TELEMAC system. This is replaced by the Paravoid library in the case of a computer running a non-parallel version of TELEMAC.

A considerable amount of information on the use of the parallel version is given in the system installation documents.

Initially, the user must specify the number of processors used by means of the keyword *PARALLEL PROCESSORS*. The keyword may have the following values:

- 0: Use of the classical version of TELEMAC-2D
- 1: Use of the parallel version of TELEMAC-2D with one processor
- N: Use of the parallel version of TELEMAC-2D with the specified number of processors, here N (it can work also just for testing on a single processor!).

In contrast to the previous version of TELEMAC-2D, domain breakdown and results file combination operations are now automatic and handled completely by the start-up procedure.

Parallel machines are configured by a single file (see system installation document).

This page is intentionally left blank

13. RECOMMENDATIONS

The purpose of this chapter is to provide the user with advice on using the software.

13.1. MESH

Certain precautions need to be taken when constructing the mesh. The following list should help in avoiding some of them, but it is not of course exhaustive.

A liquid boundary should consist of at least 5 points, with 10 being preferable.

In the case of a river mesh, and in particular for simulations of low-flow periods, it is essential to refine the elements in the low-water bed so as to ensure at least 3-4 points for conveying the flow. If this rule is not followed, there will be very serious difficulties in terms of mass conservation and the results will be of poor quality. In this case, it is possible to build the mesh of the low-water bed using the polarized mesh option implemented in MATISSE (based on the defined zone).

In domains with steep gradients in the topography or bathymetry, the slope mesh must be refined if the current is not tangential to it.

It is preferable for triangles to be as nearly equilateral as possible, as this type of element gives the best results. However, in the case of river meshes, it is sometimes interesting to elongate the grid cells in the direction of the current, in order to reduce the number of computation points and hence the simulation time.

13.2. INITIAL CONDITIONS

The technique most commonly used for maritime domains subject to tidal effects is to initialise the unconfined surface with a value corresponding to high tide and the velocities with zero, and then gradually empty the domain. The transitional stage for reaching regular conditions is limited to a half-tide.

In the case of river domains, two techniques are often used. If the domain is relatively small (i.e. the bed level does not vary much between upstream and downstream, the computation can be initialised with constant elevations, by setting the value that will be prescribed downstream of the computation domain as initial elevation. Inflow is then gradually introduced from upstream. This technique cannot be used if the model domain is very large, as the initial elevation generally means that there will be a dry area upstream of the model. In this case, it is relatively easy, in the CONDIN subroutine, to initialise an elevation with a ramp (the value of the elevation is proportional to the X or Y values) and to introduce the nominal inflow progressively. Another possibility is to use the free surface initialization implemented in FUDAA-PREPRO. This function offers the possibility to specify, in a very easy way, a free surface slope defined by a longitudinal profile prescribed as a set of points.

13.3. NUMERICAL PARAMETER DEFINITION

13.3.1. TYPE OF ADVECTION

Taking into account the recent improvement of TELEMAC-2D in this domain, the following configuration can practically be considered as a “quasi universal” configuration (even in parallel mode):

TYPE OF ADVECTION : 1 ; 5

Models with steep bottom topography gradients and tidal flats very often pose serious difficulties (oscillations of the unconfined surface, long computation times, etc.). In the light of experience, the configuration that appears to be best in such cases is as follows:

TREATMENT OF THE LINEAR SYSTEM = 2
FREE SURFACE GRADIENT COMPATIBILITY = 0.9

13.3.2. SOLVER

When using primitive equations (which is no longer recommended), the solver giving the best results in terms of computation time is GMRES (keyword value 7). In this case, it is sometimes useful to configure the dimension of the Krylov space in order to optimise computation time. The larger the dimension, the more time is required to run an iteration, but the faster the system converges. The user is therefore strongly advised to run simulations over a few time steps by varying the keyword *SOLVER OPTION* (and *OPTION FOR THE SOLVER FOR K-EPSILON MODEL*) so as to reach the best compromise between computation time for one iteration and the number of iterations, remembering that the more points there are in the mesh the higher the optimum value. This optimum value generally varies from 2 (small meshes) to 4 or 5 (large meshes). When using this solver, the optimum value for the time step (in terms of computational time) is generally reached when the convergence occurs with 10 to 20 iterations.

When using the wave equation, the recommended solver is the conjugate gradient (value 1). In that case, the optimum value for the time step is generally reached when the convergence occurs with 30 to 50 iterations.

13.4. SPECIAL TYPES OF PROGRAMMING

13.4.1. CHANGING BOTTOM TOPOGRAPHY BETWEEN TWO COMPUTATIONS

The CORFON subroutine is used to change the bottom topography read from the geometry file. Everything is programmed so that this change is made only once. The list of operations is as follows:

Reading of geometry;

Bottom correction with CORFON;

If a computation is being continued, the bottom from the previous computation results file is used, if there is one.

Any change of CORFON for a continued computation will therefore be inoperative if the bottom topography is saved in the results file, even if CORFON is actually called.

The procedure for changing bottom topography between two successive computations is as follows:

Run an initial computation without saving the bottom topography or water depth, but saving the unconfined surface.

Modify CORFON.

Continue the computation. TELEMAC-2D will then use the new bottom topography and as it only finds the unconfined surface in the results of the previous computation, it will recalculate the new depth of water as being the old unconfined surface minus the new bottom topography.

13.5. TIDAL FLATS

The following explanations concern the Finite Element option. In finite volume options (see key-word *EQUATIONS*), mass-conservation is ensured on tidal flats and the depth remains positive. However, e.g. in the case of the Malpasset dam break test-case, these explicit techniques will be much more time-consuming (factor around 10).

The treatment of tidal flats is a very strategic issue in flood and dam-break flood wave computations. Over the years a number of specific procedures have been developed in Telemac-2D to cope with this difficulty. Historically, the basic option (*TREATMENT OF THE TIDAL FLATS* : 2) consisted of removing from the computation the dry elements. This option is now completely operational. Moreover it cannot be used in parallel computations. With this option, the key-word "*MINIMUM VALUE OF DEPTH*" is used to decide whether an element is dry or not. This option is not generally recommended, but proved to be more stable with quasi-steady flows in rivers.

The preferred option is obtained with *TREATMENT OF THE TIDAL FLATS* : 1. In this case, all the finite elements are kept in the computation, which implies a specific treatment of dry points, especially when divisions by the depth occur in the equations. For example the friction terms as they appear in the non-conservative momentum equations would be infinite on dry land, and are limited in the computation. Mass-conservation is guaranteed with this option, but it is never imposed that the depth should remain positive, and slightly negative depths may appear (any correction with the key-word *H CLIPPING* would spoil the mass-conservation).

The option *TREATMENT OF THE TIDAL FLATS* : 3 is basically the same as option 1, but on partially dry elements a porosity coefficient is applied to take into account the fact that in reality the finite element has a size limited to its wet part. This option has been designed mainly for dam break studies, though users report a good behaviour in quasi-steady flows. Unless specific reasons and waiting for more convincing tests, option 1 is recommended rather than 3.

When using option 1 or 3, it is possible to use a specific treatment concerning the negative depths by selecting the appropriate value for the keyword *TREATMENT OF NEGATIVE DEPTHS*. The possibilities are :

- 0 : no treatment. The negative depths are left unchanged
- 1: smoothing of negative depth (default value)
- 2: “Flux control”. This treatment means that some fluxes between points may be limited to avoid negative depths.

When using option 1, it is possible to fix the limit value for the smoothing using the keyword *THRESHOLD FOR NEGATIVE DEPTHS* which default value is 0.

Hereafter are general recommendations when there are tidal flats in your domain:

of course use the key-word *TIDAL FLATS* : YES

avoid tidal flats every time it is possible, e.g. very steep banks can sometimes be replaced by a vertical wall.

refine the mesh on dykes or other features that will be submerged and that have a critical effect on flooding. Preferably use the wave equation.

Here are the main options chosen for a quasi-steady flow (Wesel-Xanten case originally provided by BAW):

/

VELOCITY PROFILES = 4;0
TURBULENCE MODEL = 1 *VELOCITY DIFFUSIVITY* = 2.
TIDAL FLATS : YES
OPTION FOR THE TREATMENT OF TIDAL FLATS : 1
TREATMENT OF NEGATIVE DEPTHS : 2
FREE SURFACE GRADIENT COMPATIBILITY : 0.9
H CLIPPING : NO
TYPE OF ADVECTION : 1;5
SUPG OPTION : 0;0
TREATMENT OF THE LINEAR SYSTEM:2
SOLVER:2 PRECONDITIONING:2
SOLVER ACCURACY = 1.E-5
CONTINUITY CORRECTION : YES

The wave equation (*TREATMENT OF THE LINEAR SYSTEM*:2) proved here to be more stable than primitive equations.

These options are also convenient for the Malpasset dam-break computation, and can thus be taken as a starting point for a new case.

The key-word *OPTION FOR THE DIFFUSION OF VELOCITIES* should normally set to 2, as it is the correct theoretical formula, however the simplified form corresponding to option 1 is preferred, because it avoids the problem of division by 0 on dry zones. So far no clear test-case proved the superiority of option 2.

Appendix 1

Running a TELEMAC-2D computation

A computation is started using the command telemac2d. It is also possible to start the simulation directly in FUDAA-PREPRO. This command activates the execution of a unix script which is common to all the computation modules of the TELEMAC-2D processing chain.

The syntaxes of this command are as follows:

```
telemac2d [-s] [-D] [-b | -n | -d time] [-cl] [-t] [case]
```

Note : some options depends on the operating system used

-s :	When the computation is started in interactive mode, generates check list on the disk (by default, the check list is only displayed on screen).
-cl :	Compile and link the user executable without starting the simulation
-D :	Compilation and execution using a debugger.
-b :	Running in batch mode (immediate start-up).
-n :	Running in deferred batch mode (start-up at 20h00).
-d :	Running in deferred batch mode (start-up at the specified time).
-t :	Do not delete working directory after normal run
case :	Name of steering file.

telemac2d -h | -H (short or long help).

If no name for the steering file is indicated, the procedure uses the name cas. By default, the procedure executes the computation in interactive mode, and displays the check list on screen.

Examples:

telemac2d start computation immediately in interactive mode using the cas steering file.

telemac2d -b test2 start computation immediately in batch mode using the test2 steering file.

telemac2d -d 22:00 modtot start computation at 22:00 the same evening in batch mode using the modtot steering file.

telemac2d -n start computation at 20:00 the same evening in batch mode using the cas steering file.

The following operations are carried out using this script:

- Creation of a temporary directory,
- Copy of the dictionary and steering file in this directory,
- Execution of DAMOCLES software in order to determine the name of the workfiles,
- Creation of the script to start the computation,
- Allocation of files,
- Compilation of the FORTRAN file and link edition (if necessary),
- Start of the computation,
- Restitution of the results files, and destruction of the temporary directory.

Procedure operation differs slightly depending on the options used.

A detailed description of this procedure may be obtained by using the command telemac2d -H.

Appendix 2

List of subroutines that can be usually modified by the user

BORD	Imposition of particular boundary conditions
CONDIN	Imposition of particular initial conditions
CORFON	Modification of bottom elevations
CORPOR	Modification of porosity
CORRXY	Modification of mesh coordinates
CORSTR	Space-dependent friction coefficient
CORVIS	Modification of viscosities
DEBSCE	Time-dependent tracer source flow rates (function)
DECLARATIONS_TELEMAC2D	Declaration of TELEMAC-2D variables (module)
DEF_ZONES	Definition of zones
DRAGFO	Definition of vertical structures
FLOT	Initial position of drogues
FLUXPR	Management of control sections
INCIDE	Definition of incident waves
LAGRAN	Lagrangian drifts
LATITU	Computation of variables depending on latitude
MASKOB	Masking of elements
MESURES	Reading of measurement data
METEO	Atmospheric conditions (wind, pressure)
NOMVAR_TELEMAC2D	Definition of names of additional variables

PRERES_TELEMAC2D	Computation of additional variables
PROPIN_TELEMAC2D	Change of the type of boundary conditions
Q	Imposition of a time-dependent boundary flowrate (function)
SL	Imposition of a time-dependent boundary free surface elevation (function)
STRCHE	Space-dependent friction coefficient
TR	Imposition of a time-dependent boundary tracer value (function)
TRSCE	Imposition of time-dependent tracer values at the sources (function)
VALIDA	Validation of a computation
VIT	Imposition of a time-dependent boundary velocity (function)
VUSCE	Variable velocity along X of a source (function)
VVSCE	Variable velocity along Y of a source (function)

Appendix 3

Description of Serafin file standard

This is a binary file.

The records are listed below:

- 1 record containing the title of the study (72 characters) and a 8 characters string indicating the type of format (SERAFIN or SERAFIND)
- 1 record containing the two integers NBV(1) and NBV(2) (number of linear and quadratic variables, NBV(2) with the value of 0 for Telemac, as quadratic values are not saved so far),
- NBV(1) records containing the names and units of each variable (over 32 characters),
- 1 record containing the integers table IPARAM (10 integers, of which only the 6 are currently being used),
    - if IPARAM (3) ≠ 0: the value corresponds to the x-coordinate of the origin of the mesh,
    - if IPARAM (4) ≠ 0: the value corresponds to the y-coordinate of the origin of the mesh,
    - if IPARAM (7) ≠ 0: the value corresponds to the number of planes on the vertical (3D computation),
    - if IPARAM (8) ≠ 0: the value corresponds to the number of boundary points (in parallel),
    - if IPARAM (9) ≠ 0: the value corresponds to the number of interface points (in parallel),
    - if IPARAM(8) or IPARAM(9)≠0: the array IPOBO below is replaced by the array KNOLG (total initial number of points). All the other numbers are local to the sub-domain, including IKLE.
- if IPARAM (10) = 1: a record containing the computation starting date,
- 1 record containing the integers NELEM,NPOIN,NDP,1 (number of elements, number of points, number of points per element and the value 1),

- 1 record containing table IKLE (integer array of dimension (NDP,NELEM) which is the connectivity table. Attention: in TELEMAC-2D, the dimensions of this array are (NELEM,NDP)),

- 1 record containing table IPOBO (integer array of dimension NPOIN); the value of one element is 0 for an internal point, and gives the numbering of boundary points for the others,

- 1 record containing table X (real array of dimension NPOIN containing the abscissae of the points),

- 1 record containing table Y (real array of dimension NPOIN containing the ordinates of the points),

Next, for each time step, the following are found:

- 1 record containing time T (real),

- NBV(1)+NBV(2) records containing the results tables for each variable at time T.

Appendix 4

Generating output files for DELWAQ

The TELEMAC-2D software is able to generate the appropriate files necessary to run a DELWAQ simulation. This generation is managed only thanks to the following keyword:

BOTTOM SURFACES DELWAQ FILE
DELWAQ PRINTOUT PERIOD
DELWAQ STEERING FILE
DIFFUSIVITY DELWAQ FILE
DIFFUSIVITY FOR DELWAQ
EXCHANGE AREAS DELWAQ FILE
EXCHANGE BETWEEN NODES DELWAQ FILE
NODES DISTANCES DELWAQ FILE
SALINITY DELWAQ FILE
SALINITY FOR DELWAQ
TEMPERATURE DELWAQ FILE
TEMPERATURE FOR DELWAQ
VELOCITY DELWAQ FILE
VELOCITY FOR DELWAQ

More information about these keywords can be found in the TELEMAC-2D reference manual. For more information, please refer to the DELWAQ user documentation.

This page is intentionally left blank

Appendix 5

Correspondence between colour codes and types of boundary conditions

This appendix is intended for users who do not possess the MATISSE software. The following table gives the colour codes prescribed by the mesh generation software and the corresponding boundary conditions included in STBTEL.

TRIGRID	STB(*)	LIHBOR	LIUBOR	LIVBOR	LITBOR	BOUNDARY CONDITION
1	11	2	2	2	2	Solid boundary.
2	4	5	4	0	4	Prescribed H, free U, zero V, free T.
3	5	1	1	1	4	Incident wave, free T.
4	7	5	0	4	4	Prescribed H , zero U, free V, free T.
5	8	4	5	5	5	Free H, prescribed Q, prescribed T.
6	9	4	6	6	5	Free H, prescribed velocities, prescribed T.
7	1	5	4	4	4	Prescribed H, free velocities, free T.
8	12	4	5	0	5	Free H, prescribed Q with zero V, prescribed T.
9	15	4	0	5	5	Free H, prescribed Q with zero U, prescribed T.
10	2	5	5	5	5	Prescribed H and Q, prescribed T.
11	3	5	6	6	5	Prescribed H and velocities, prescribed T.
12	14	2	0	2	2	Solid boundary with zero U.
13	13	2	2	0	2	Solid boundary with zero V.

Note : Boundary conditions on the tracer are managed by SISYPHE as sedimentological boundary conditions in the following way :

(*) SUPERTAB

This page is intentionally left blank

Appendix 6

Defining friction by domains

When a complex definition of the friction has to be used for a computation, the user can use this option, which divides the domain in sub-domains (domains of friction) where different parameters of friction can be defined and easily modified. The procedure is triggered by the key-word “FRICTION DATA=YES” and the data are contained in a file “FRICTION DATA FILE”.

The user has to:

- define the domains of friction in the mesh,
- define the parameters of friction for each domain of friction,
- add the corresponding keywords in the steering file of Telemac-2d in order to use this option.

I – Friction domains

In order to make a computation with variable coefficients of friction, the user has to describe, in the computational domain, the zones where the friction parameters will be the same. For that, a code number, which represents a friction domain, has to be given to each node. The nodes with the same code number will use the same friction parameters.

This allocation is done thanks to the user subroutine friction_user.f. All nodes can be defined ”manually” in this subroutine, or this subroutine can be used in order to read a file where the link between nodes and code numbers is already generated (for example with the software JANET from the BAW).

Warning : the number of the nodes in the whole domain is not known during a parallel computation (especially when a file with the link between nodes and code numbers of domain is read) !

II – Friction parameters

The frictions parameters of each friction domain are defined in a special friction data file. In this file we find, for each code number of friction domain:

a law for the bottom and their parameters,

a law for the boundary conditions and their parameters (only if the option k-epsilon is used),

the parameters of non-submerged vegetation (only if the option is used)

Example of friction data file :

*Zone	Bottom			Boundary	condition		Non submerged	vegetation
*no	TypeBo	Rbo	MdefBo	TypeBo	Rbo	MdefBo	Dp	sp
From 4 to 6	NFRO			LOGW	0.004		0.002	0.12
20	NIKU	0.10		NIKU	0.12		0.006	0.14
27	COWH	0.13	0.02	LOGW	0.005		0.003	0.07
END								

The first column defines the code number of the friction domain. Here, there is 3 lines with the code numbers : 4 to 6, 20, 27.

The columns from 2 to 4 are used in order to define the bottom law: the name of the law used (NFRO, NIKU or COWH for this example, see below for the name of the laws), the roughness parameter used and the Manning’s default value (used only with the COlebrook-WHite law). If the friction parameter (when there is no friction) or the Manning’s default are useless, nothing has to be written in the column,

The columns from 5 to 7 are used in order to describe the boundary conditions laws: name of the law, roughness parameter, Manning’s Default. These columns have to be set only if the turbulence model is k-epsilon (=> 3) with the option rough (=> 2) for turbulence model for solid boundaries, else nothing has to be written in these columns,

The columns 8 and 9 are used for the non-submerged vegetation: diameter of roughness element and spacing of roughness element. These columns have to be set only if the option non-submerged vegetation is used, else nothing has to be written in these columns, The last line of the file must have only the keyword END (or FIN or ENDE),

In order to add a comment in the friction data file, the line must begin a star ”*”.

Link between the laws implemented and their names in the friction data file :

Law	Telemac2d numbering	Name for data file	Parameters used
No Friction	0	NOFR	No parameter
Haaland	1	HAAL	Roughness coefficient
Chézy	2	CHEZ	Roughness coefficient
Strickler	3	STRI	Roughness coefficient
Manning	4	MANN	Roughness coefficient
Nikuradse	5	NIKU	Roughness coefficient
Log law of wall (1)	6	LOGW	Roughness coefficient
Colebrook-White	7	COWH	Roughness coefficient Manning coefficient

(1) : can be used only for boundaries conditions

III – Steering file

In order to use a friction computation by domains, the next keyword have to be added:

For the friction data file :

FRICTION DATA = YES

FRICTION DATA FILE = ‘name of the file where friction data are’

For the non-submerged vegetation (if used) :

NON-SUBMERGED VEGETATION = YES

By default, 10 zones are allocated, this number can be changed thanks to the keyword:

MAXIMUM NUMBER OF FRICTION DOMAINS = 80

If the link between nodes and code numbers of friction domain is built thanks to a file:

FORMATTED DATA FILE 1 or 2 = ‘name of the file’

(check that the right formatted data file is read in the user subroutine).

IV – Advanced options

If some friction domains with identical parameters have to be defined, it is possible to define them only with one line thanks to the keyword : from... to... (it is also possible to use de... a... or von... bis...).

The first code number of the domains and the last code number of the domains have to be set. All domains of friction with a code number between these two values will be allocated with the same parameters, except :

If a friction domain is defined in two different groups, the priority is given to the last group defined.

A single friction domain has ever the priority on a group even if a group with this domain is defined afterwards,

If a single friction domain is defined two times, the priority is given to the last definition.

V – Programming

A new module, FRICTION_DEF, has been created in order to save the data read in the friction file. This module is built on the structure of the BIEF objects. The domain of friction ”i” is used as follows:

```
TYPE(FRICTION_DEF) :: TEST_FRICTION

TEST_FRICTION%ADR(I)%P
```

The components of the structure are:

TEST_FRICTION%ADR(I)%P%GNUM (1)	1st code number of the friction domains
TEST_FRICTION%ADR(I)%P%GNUM (2)	Last code number of the friction domains
TEST_FRICTION%ADR(I)%P%RTYPE(1)	Law used for the bottom
TEST_FRICTION%ADR(I)%P%RTYPE(2)	Law used for the boundaries conditions
TEST_FRICTION%ADR(I)%P%RCOEF(1)	Roughness parameters for the bottom
TEST_FRICTION%ADR(I)%P%RCOEF(2)	Roughness parameters for the boundaries conditions
TEST_FRICTION%ADR(I)%P%NDEF(1)	Default Manning for the bottom
TEST_FRICTION%ADR(I)%P%NDEF(2)	Default Manning for the boundary conditions
TEST_FRICTION%ADR(I)%P%DP	Diameter of the roughness element
TEST_FRICTION%ADR(I)%P%SP	Spacing of the roughness element

TEST_FRICTION%ADR(I)%P%GNUM(1) and TEST_FRICTION%ADR(I)%P%GNUM(2) have the same value if a single friction domain is defined.

TEST_FRICTION%ADR(I)%P%RTYPE(1) is KFROT when there is only one domain.

TEST_FRICTION%ADR(I)%P%RCOEF(1) is CHESTR when there is only one domain.

The link between Telemac2d and the computation of the friction is done with the subroutine friction_choice.f. It is used in order to initialize the variables for the option DAT FRICTION at the beginning of the program and/or in order to call the right friction subroutine for the computation at each iteration.

Initializing:

During the initialization, the parameters of the friction domains are saved thanks to the subroutine friction_read.f and the code number of each nodes are saved thanks to friction_user.f in the array KFROPT%I. With the subroutine friction_init.f, the code numbers for all nodes are checked and the arrays CHESTR%R and NKFROT%I (KFROT for each node) are built. KFROT is used in order to know if all friction parameters are null or not. This information is used during the computation.

Computing:

For the optimization, the computation of the friction coefficient is done in the subroutine friction_calc.f for each node thanks to the loop I = N_START, N_END. When the option FRICTION DATA is not used, N_START and N_END are initialized to 1 and NPOIN in the subroutine friction_unif.f. Else, they take the same value and the loop on the node is done in the subroutine friction_zone.f (the parameters used for each node can be different).

With this choice, the subroutine friction_unif.f is not optimized when the option NON-SUBMERGED VEGETATION is called (friction_lindner.f). This option aims to correct the value of the bottom friction coefficient when there is partial submerged vegetation.

VI – Accuracy

When the option FRICTION DATA is not used, CHESTR can be read in the geometry file. The values stored in this file are in simple precision. However CHESTR is defined in double precision, then, the CHESTR value is not exactly the right value.

With the option FRICTION DATA, CHESTR is set thanks to the friction data file where the value of each domains are stored in double precision.

Then when a comparison is done between the method, the difference comes from this precision. In order to avoid that, it is possible to use the subroutine corstr.f in order to improve the precision of CHESTR when the friction data option is not called (thanks to the test IF ABS(CHESTR%R(I) – ROUGHNESS)< 1.D-7 THEN CHESTR = ROUGHNESS.D0, where ROUGHNESS is the roughness coefficient, but they must all be known)

.

INDEX

Note: this index does not include the appendices