# TELEMAC-2D

# User Manual

**Version v8p3**
**December 6, 2021**

# Contents


1 Introduction . . . . . 7

1.1 Presentation of the TELEMAC-2D software 7

1.2 Position of the TELEMAC-2D code within the telemac modelling system 8

1.3 User programming 9

2 Theoretical aspects . . . . . 10

3 Inputs and outputs . . . . . 12

3.1 Preliminary remarks 12

3.1.1 Binary files format . . . . . 14

3.2 The files 14

3.2.1 The steering file . . . . . 14

3.2.2 The geometry file . . . . . 16

3.2.3 The boundary conditions file . . . . . 17

3.2.4 The FORTRAN user file . . . . . 17

3.2.5 The liquid boundaries file . . . . . 17

3.2.6 The sources file . . . . . 17

3.2.7 The friction data file . . . . . 17

3.2.8 The stage-discharge or elevation-discharge curves file . . . . . 17

3.2.9 The sections input file . . . . . 18

3.2.10 Files dedicated to construction works . . . . . 18

3.2.11 The reference file . . . . . 18

3.2.12 The results file . . . . . 18

3.2.13 The listing printout . . . . . 19

3.2.14 The auxiliary files . . . . . 19

3.2.15 The dictionary file . . . . . 20

3.2.16 Topographic and bathymetric data . . . . . 20

3.2.17 Files dedicated to NESTOR . . . . . 22

4 Hydrodynamic simulation . . . . . 23

4.1 Prescribing initial conditions 23

4.1.1 Prescribing using keywords . . . . . 23

4.1.2 Prescribing particular initial conditions with user subroutines . . . . . 24

4.1.3 Continuing a computation . . . . . 24

**4.2 Prescribing boundary conditions 25**
4.2.1 Possible choices . . . 25
4.2.2 Description of various types of boundary conditions . . . 26
4.2.3 The boundary conditions file . . . 27
4.2.4 Prescribing values through keywords . . . 28
4.2.5 Prescribing values by programming subroutines or using the open boundaries file . . . 29
4.2.6 Stage-discharge curves . . . 30
4.2.7 Prescribing complex values . . . 31
4.2.8 Prescribing velocity profiles . . . 32
4.2.9 Thompson boundary conditions . . . 32
4.2.10 Elements Masking . . . 33
4.2.11 Definition of types of boundary condition when preparing the mesh . . . 34
4.2.12 Tidal harmonic constituents databases . . . 34

**5 General parameter definition for the computation . . . 38**

**5.1 Criteria for stopping a computation 38**

**5.2 Control sections 39**
5.2.1 Configuration with keywords only . . . 39
5.2.2 Configuration with an external file . . . 40

**5.3 Computation of fluxes over lines (FLUXLINE) 40**

**6 Physical parameter definition . . . 42**

**6.1 Friction parameter definition 42**
6.1.1 Vegetation friction . . . 43
6.1.2 Wave friction enhancement . . . 43
6.1.3 Sidewall friction . . . 43
6.1.4 Non-Newtonian fluid . . . 44

**6.2 Modelling of turbulence 45**
6.2.1 Constant viscosity . . . 45
6.2.2 Elder model . . . 46
6.2.3 $k$-$\varepsilon$ model . . . 46
6.2.4 Smagorinski model . . . 46
6.2.5 Mixing length model . . . 47
6.2.6 Spalart-Allmaras model . . . 47

**6.3 Setting up of meteorological phenomena 47**
6.3.1 Wind influence . . . 47
6.3.2 Atmospheric pressure . . . 48
6.3.3 Rain and evaporation . . . 49
6.3.4 Rainfall-runoff Modelling . . . 49

**6.4 Astral potential 50**

**6.5 Wave induced currents 50**

**6.6 Vertical structures 51**

**6.7 Other physical parameters 51**

**6.8 Tsunami generation 52**

6.9 Parameter estimation 52

7 Numerical parameter definition . . . . . 54

7.1 General parameter definition 54

7.2 Numerical schemes 55
7.2.1 Finite elements . . . . . 55
7.2.2 Finite volumes . . . . . 60

7.3 Solving the linear system 62
7.3.1 Solver . . . . . 62
7.3.2 Accuracy . . . . . 63
7.3.3 Continuity correction . . . . . 64
7.3.4 Preconditioning . . . . . 64
7.3.5 C-U preconditioning . . . . . 66

7.4 Courant number management 66

7.5 Tidal flats 66

7.6 Other parameters 68
7.6.1 Matrix storage . . . . . 68
7.6.2 Matrix-vector product . . . . . 68
7.6.3 Finite Element assembly mode in parallel . . . . . 68

7.7 Convergence study 69

8 Managing water sources . . . . . 70

9 Tracer transport . . . . . 73

9.1 General setup 73

9.2 Prescribing initial conditions 74

9.3 Prescribing boundary conditions 74

9.4 Managing tracer sources 75

9.5 Numerical specifications 75

9.6 Law of tracer degradation 75

10 Secondary currents . . . . . 76

11 Water quality . . . . . 77

12 Particle transport and lagrangian drifts . . . . . 78

12.1 Drogues displacements 78
12.1.1 Input Files . . . . . 78
12.1.2 Steering file . . . . . 78
12.1.3 Initial drogues locations . . . . . 79
12.1.4 Output file . . . . . 80

**12.2 Algae modelling** 81
12.2.1 Input files . . . 81
12.2.2 Steering file . . . 81
12.2.3 Initial algae locations . . . 82
12.2.4 Dislodgement . . . 82
12.2.5 Output files . . . 83
**12.3 Oil spill modelling** 83
12.3.1 Input files . . . 83
12.3.2 Steering file . . . 83
12.3.3 Oil spill steering file . . . 84
12.3.4 The OIL_FLOT subroutine . . . 86
12.3.5 Output files . . . 86
**12.4 Lagrangian drifts** 86
12.4.1 Input files . . . 86
12.4.2 Output files . . . 87

**13 Construction works modelling . . . 88**
**13.1 Weirs** 88
**13.2 Culverts** 89
**13.3 Dykes breaches** 90
13.3.1 General overview . . . 90
13.3.2 Breach computation for rectangular shapes . . . 92
13.3.3 Breach computation for trapezoidal shapes . . . 95
13.3.4 Breaches data file . . . 97

**14 Other configurations . . . 100**
**14.1 Modification of bottom topography (USER_CORFON)** 100
**14.2 Modifying coordinates (USER_CORRXY)** 100
**14.3 Spherical coordinates (LATITU)** 101
**14.4 Adding new variables (USER_NOMVAR_TELEMAC2D and USER_PRERES_TELEMAC2D)** 101
**14.5 Array modification or initialization** 102
**14.6 Validating a computation (BIEF_VALIDA)** 102
**14.7 Changing the type of a boundary condition (PROPIN_TELEMAC2D)** 102
**14.8 Coupling** 103
**14.9 Assigning a name to a point** 105
**14.10 Fourier analysis** 105
**14.11 Checking the mesh (CHECKMESH)** 106

**15 Parallelism . . . 107**

16 Recommendations . . . . . 108
16.1 Mesh 108
16.2 Initial conditions 108
16.3 Numerical parameter definition 109
16.3.1 Type of advection . . . . . 109
16.3.2 Solver . . . . . 109
16.4 Special types of programming 109
16.4.1 Changing bottom topography between two computations . . . . . 109
16.5 Tidal flats 110

17 API . . . . . 112

A Launching the computation . . . . . 113

B List of user subroutines . . . . . 115

C The SELAFIN format . . . . . 117

D Generating output files for DELWAQ . . . . . 119

E Defining friction by domains . . . . . 120

Bibliography . . . . . 125

# 1. Introduction

## 1.1 Presentation of the TELEMAC-2D software

The TELEMAC-2D code solves depth-averaged free surface flow equations as derived first by Barré de Saint-Venant in 1871 (also known as Shallow Water Equations). The main results at each node of the computational mesh are the water depth 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, rain or evaporation 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,
- Transport and diffusion of a tracer by currents, including creation and decay terms,
- Particle tracking and computation of Lagrangian drifts,
- Treatment of singularities: weirs, dykes, culverts, etc.,
- Dyke breaching,
- Drag forces created by vertical structures,
- Porosity phenomena,

- Wave-induced currents (by coupling or chaining with the ARTEMIS and TOMAWAC modules),
- Coupling with sediment transport,
- Coupling with water quality tools,
- Coupling with ice/frazil.

The software has many fields of application. In the maritime sphere, particular mention may be given of the sizing of port structures, the study of the effects of building submersible dykes or dredging, the impact of waste discharged from a coastal outfall or the study of thermal plumes. In river applications, mention may also be given of studies relating to the impact of construction works (bridges, weirs, tubes), 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 initially 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), and is now managed by a consortium of other consultants and research institutes, more informations can be found on the website www.opentelemac.org. Like previous releases of the program, release 8.1 complies with EDF-R&D'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 at least each time new releases are produced.

## 1.2 Position of the TELEMAC-2D code within the telemac modelling system

The TELEMAC-2D software is part of the TELEMAC modelling system developed by the LNHE of EDF R&D. TELEMAC is a set of modelling tools allowing to treat every aspects of natural free surface hydraulics: currents, waves, transport of tracers and sedimentology.
The pre-processing and post-processing of simulations can be done either directly within the TELEMAC system or with different software that present an interface of communication with the system. We can particularly mention the following tools:

- The FUDAA-PREPRO software, developed from the FUDAA platform by the CEREMA's Recherche, Informatique et Modélisation Department, covers all the pre-processing tasks involved by the achievement of a numerical hydraulic study, as well as a graphical post-processing tool,
- The Blue Kenue software, developed the Hydraulic Canadian Center, proposes a powerful mesh generation tool and a user-friendly post-processing tool,
- The Janet software, developed by Smile Consult GmbH, which offers among others, a mesh generation tool,
- The ParaView software, developed by Sandia National Laboratories, Los Alamos National Laboratory and Kitware, which enables to visualise 3D results, big data in particular and is open source,

- The SALOME-HYDRO software based on the SALOME platform, developed by EDF, CEA and OPENCASCADE which enables to handle raw data (bathymetry, maps, pictures, LIDAR...) until the mesh generation. The post-processing tool ParaViS available in the SALOME platform is based on the ParaView software and can visualise 1D, 2D or 3D results. A first version of SALOME-HYDRO has been available since Spring 2016,
- The Tecplot 360 software, developed by Tecplot which enables to visualise 2D and 3D results,
- The QGIS software, which is an open source Geographic Information System.

## 1.3 User programming

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. 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 subroutines provided with the system, and copy them into a single file or in a directory that will be the specific `FORTRAN FILE` of the given case, see section 3.2.4 for more details,
- Modify the subroutines according to the model you wish to build,
- Link up the set of subroutines into a single file or in a single directory that will be compiled during the TELEMAC-2D start procedure. The name for the file or directory is given by the `FORTRAN FILE` keyword.

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 may also be 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**.

# 2. Theoretical aspects

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

$$\frac{\partial h}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla}(h) + h\,div(\boldsymbol{u}) = S_h \qquad \text{continuity,}$$

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

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

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

in which:

- $h$ (m) water depth,
- $u, v$ (m/s) velocity components,
- $T$ (e.g.: g/l, °C or no unit) passive (non-buoyant) tracer,
- $g$ (m/s$^2$) gravity acceleration,
- $\nu_t$,$\nu_T$ (m$^2$/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,
- $S_T$ (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.
$S_x$ and $S_y$ (m/s$^2$) 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 $\varepsilon$ (turbulent dissipation), for which the equations are the following:

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

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

The right-hand side terms of these equations represent the production and destruction of turbulent quantities (energy and dissipation).
When non hydrostatic effects are not negligible, Saint-Venant equations can be improved by adding extra terms. Several trials can be found in the literature (Serre, Boussinesq, Korteweg and De Vries). To use Boussinesq assumptions, the following terms are added to the right-hand side of Saint-Venant equations (thus called Boussinesq equations):

$$-\frac{H_0^2}{6} \overrightarrow{grad} \left[ div \left( \frac{\partial \overrightarrow{u}}{\partial t} \right) \right] + \frac{H_0}{2} \overrightarrow{grad} \left[ div \left( H_0 \frac{\partial \overrightarrow{u}}{\partial t} \right) \right].$$

A complete description of the theory is given in the following book “Hydrodynamics of free surface flows”, by Jean-Michel Hervouet [18].

# 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 (**mandatory**), containing the configuration of the computation,
- The geometry file (**mandatory**), containing the mesh,
- The boundary conditions file (**mandatory**), containing the description of the type of each boundary,
- The previous computation file, which can give the initial state of the computation (case of a restart comoputation). This is an optional file,
- The bottom topography file, containing the elevation of the bottom. Generally, the topographic data are already available in the geometry file and the bottom topography file is generally not used,
- The reference file, which contains the "reference" results and is used in the frame of a validation procedure,
- The liquid boundaries file, containing information about the prescribed values at the open boundaries (elevation, flowrate...),
- The FORTRAN file, containing the specific programming,
- 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 contains information on the liquid boundaries where the characteristics are prescribed according to 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 oil spill steering file, which contains all the parameters necessary to the simulation of an oil spill event. See section 12.3 for more details,
- The tidal model file, which contains data used for tide simulation. See section 4.2.12 for more details,
- The ASCII database for tide,
- The binary database 1 and 2 for tide files,
- The weirs data file which contains all needed parameters related to weirs,
- The culverts data file,
- The breaches data file which contains the characteristics of the breaches initiation and growth. See section 13.3,
- The drogues file, which contains the parameters for drogues creation and release. See section 12.1,
- The zones file which contains the description of friction zones, or any other zone.

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` (default value = 0) in the `STEERING FILE`. `DEBUGGER` = 1 will show the sequence of calls to subroutines in the main program **TELEMAC2D**. 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 (input),
- 2 formatted data files (input),
- 1 binary results file (output),
- 1 formatted results file (output).

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 files format

The binary files managed inside the TELEMAC system can have various formats. The most commonly used format is the SERAFIN format (also known as 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 available format is the MED format which is compatible with the SALOME platform jointly developed by EDF and CEA. The full description of the MED format is available on the SALOME website http://www.salome-platform.org.
Depending on the specified format, the binary file can be read by different tools. FUDAA-PREPRO and Blue Kenue can also read double precision.
The selection of the appropriate format is done using the corresponding keyword. 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" or more accurate validation, and 'MED ' means the MED HDF5 format.
The keywords involved are:

- `BINARY ATMOSPHERIC DATA FILE FORMAT`,
- `BINARY DATA FILE 1 FORMAT`,
- `BINARY DATA FILE 2 FORMAT`,
- `BINARY RESULTS FILE FORMAT`,
- `GEOMETRY FILE FORMAT`,
- `PREVIOUS COMPUTATION FILE FORMAT`,
- `REFERENCE FILE FORMAT`,
- `RESULTS FILE FORMAT`,
- `TIDAL MODEL FILE FORMAT`.

## 3.2 The files

### 3.2.1 The steering file

This is a text file created by a text editor or 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 3.2.15). 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. 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. The format of this file is given by the keyword `GEOMETRY FILE FORMAT`. There are 3 possibilities:

- SERAFIN: classical single precision format in TELEMAC,
- SERAFIND: classical double precision format in TELEMAC,
- MED: MED double precision format based on HDF5.

If it is a SERAFIN-formatted binary file, it can be read by Paraview, FUDAA-PREPRO, Tecplot 360, Blue Kenue or RUBENS and can either be generated by Janet, Blue Kenue or MATISSE, or by the STBTEL module (from the file(s) originating from of mesh generator). The SERAFIN format structure is described in Appendix C.
If it is a MED-formatted binary file, it can be read by Paraview only and can be generated by SALOME-HYDRO or by the STBTEL module (from the file(s) originating from of mesh generator).
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 or friction ID 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`.

### 3.2.3 The boundary conditions file

This is a formatted file generated automatically by most of the mesh generators compatible with TELEMAC (Janet, Blue Kenue, MATISSE), but also by FUDAA-PREPRO or STBTEL. It can be modified using FUDAA-PREPRO or 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 release 5.0 of the software (the first release 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 (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 sources file

This text file enables the user to specify values for time-dependent conditions for sources (discharge, tracers' concentration).
See Chapter 8 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 can vary from one zone to another.
The file name is specified with the keyword: `FRICTION DATA FILE` but is used only if the logical keyword `FRICTION DATA` is activated (default = NO).
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 E for a complete description of this file.

### 3.2.8 The stage-discharge or elevation-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 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 Files dedicated to construction works

When using specific treatment of singularity (weirs, culverts, breaches), these files are used to specify the elements necessary for the concerned treatment. The keywords identifying these files are:

- WEIRS DATA FILE,
- CULVERTS DATA FILE,
- BREACHES DATA FILE.

### 3.2.11 The reference file

During the validation step of a calculation, 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, the two velocity components and other variables such as $k$, $\varepsilon$ and tracers.
The name of this file is given by the keyword REFERENCE FILE and its format is specified by the keyword REFERENCE FILE FORMAT (default value: 'SERAFIN '), see subsection 3.1.1.

### 3.2.12 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 content depends 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 stabilization precedes a transient simulation,
- GRAPHIC PRINTOUT PERIOD: fixes the period for outputs so as to avoid having an excessively large file. Default value is 1 (writing at every time step). 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.

The name of this file is given with the keyword RESULTS FILE and its format is given with RESULTS FILE FORMAT.
If the GEOMETRY FILE is given in longitude-latitude, the result files can also be written with the coordinates in longitude-latitude by setting the keyword RESULT FILE IN LONGITUDE-LATITUDE at YES (default value).

### 3.2.13 The listing printout

This is a formatted file created by TELEMAC-2D during the computation. It contains the report of a TELEMAC-2D running. Its contents vary according to 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 printing information, so as to avoid having excessively large files, in particular when a stabilisation period precedes a transient simulation. The default value is 0 (writing of the listing printouts at the beginning of the simulation),
- LISTING PRINTOUT PERIOD: this sets the period between two time step printings. The value is given in numbers of time steps. By default, it is equal to 1, i.e. every time step. For instance, the following sequence:

  ```
  \telkey{TIME STEP = 30}.
  \telkey{LISTING PRINTOUT PERIOD = 2}
  ```

  Prints the output listing every minute of simulation. Moreover, irrespective of the period indicated by the user, the last time step is systematically printed,
- LISTING FOR PRINTOUT PERIOD: this sets the period between two time step printings. The value is given in numbers of time steps. By default, it is equal to 1, i.e. every time step. If this keyword is present in the steering file, its value has priority to LISTING PRINTOUT PERIOD.
- 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 get information on the mass fluxes (or rather volumes) in the domain at each printed time step. This is not done by default,
- 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 authorized,
- 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 $k$-$\varepsilon$ transport equations, or by default that reached at the end of the maximum number of iterations authorized.

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

### 3.2.14 The auxiliary files

Other files may be used by TELEMAC-2D. Using these files will most often require an implementation in FORTRAN. Details on their logical units in FORTRAN are given below.

**Auxiliary 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 using the **GET_FREE_ID** subroutine, the **T2DBI1** logic unit for binary data file 1 and the **T2DBI2** logic unit for binary data file 2.

- 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 using the **GET_FREE_ID** subroutine, the **T2DFO1** logic unit for formatted data file 1 and the **T2DFO2** logic unit for formatted data file 2.

- 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 using the **GET_FREE_ID** subroutine and the **T2DRBI** logic unit.

- 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 using the **GET_FREE_ID** subroutine and the **T2DRFO** logic unit.

Read and write operations on these files must be managed completely by the user. That 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 or the subroutines called by it (e.g.: **USER_CONDIN_H**). Similarly, using a file to introduce boundary conditions can be done in the **USER_BORD** subroutine.
TELEMAC-2D can also use other files when using harmonic constants databases. These files are described in detail in Section 4.2.12.
All the logical units are stored in a structure called **T2D_FILES**. The logical unit of `BINARY DATA FILE 1`, for instance, will be **T2D_FILES(T2DBI1)%LU**.

Note:
In some subroutines, it will be necessary to add

```
USE DECLARATIONS_TELEMAC2D, ONLY: T2D_FILES, T2DBI1
```

for example, to have access to the logical units of files.

### 3.2.15 The dictionary file

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

### 3.2.16 Topographic and bathymetric data

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

1. Either directly in the GEOMETRY FILE by a topographic or bathymetric value associated with each mesh node. In this case, the data are processed while the mesh is being built using SALOME-HYDRO, Janet, Blue Kenue or MATISSE, or when STBTEL 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,

2. 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. Unlike STBTEL, TELEMAC-2D only manages one single bottom topography file. This file consists of three columns X, Y, Z,

3. Or using the **USER_CORFON** subroutine (see section 14.1). This is usually what is done 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 14.1). This smoothing preserves volumes.

**Dykes modelling**

Dykes representation requires special attention from the modeler. To properly handle the flow behavior at the dykes level (including the apparition of overflow phenomena), it is necessary to provide a minimum discretization of the cross sections of these dykes. As shown in the figure below, this discretization should be based on a minimum of 5 points (generally corresponding to 5 constraints lines in the mesh generation tool):

2 points representing the base of the dyke, 2 points representing the ends of the upper level of the dyke and an extra point, slightly above the middle of the dyke.
This latter point allows, when the sides of the dyke are half wet, to avoid the apparition of a parasitic flow over the dyke if the water level at the highest point, calculated by the tidal flat algorithms, is not strictly zero.
Despite the care taken in meshing and the quality of the algorithms developed within TELEMAC-2D, there are sometimes parasitic overflows over some dykes (the presence of water on the crest of the dyke whereas the surrounding free surface is located below that level). This is sometimes due to insufficient spatial discretization around dykes, or because of the influence of inertia phenomena overvalued by the code given that the dykes slopes could be too low compared to reality (the size of the elements generally prevents to respect these slopes). To handle this type of situation, a specific treatment algorithm has been implemented in TELEMAC-2D. This allows to automatically perform a receding procedure when the water level on the crest of the dyke is less than a threshold set by the user and that the slope of the free surface at the

dyke is too high. This threshold, typically of a few millimeters to a few centimeters is set using the keyword THRESHOLD DEPTH FOR RECEDING PROCEDURE (expressed in meters and whose default value is 0 m). It is recommended to use this algorithm with convection schemes that ensures a perfect mass conservation. It is also compatible with a correct treatment of the convection of tracers. If necessary, the user can refer to the subroutine **RECEDING**.
Note that release 7.0 of TELEMAC-2D allows taking into account the phenomena of dyke failure. This function is described in detail in Section 13.3.

### 3.2.17 Files dedicated to NESTOR

When using the module NESTOR to carry out dredge or dump activities, these files are used to specify the action, location, geometry and the last status. The keywords identifying these files are:

- NESTOR ACTION FILE,
- NESTOR POLYGON FILE,
- NESTOR SURFACE REFERENCE FILE,
- NESTOR RESTART FILE.

These files are described in detail in the NESTOR User Manual.

# 4. Hydrodynamic simulation

## 4.1 Prescribing 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 of 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. In simple cases, this can be done using keywords, or by programming in more complex ones. It is also possible to define the initial state using FUDAA-PREPRO.

### 4.1.1 Prescribing using keywords

In all cases, the kind of the initial conditions is set by the keyword `INITIAL CONDITIONS`, except if it has been defined using FUDAA-PREPRO INITIAL CONDITIONS. The keyword `INITIAL CONDITIONS` may have any of the following six values:

- 'ZERO ELEVATION': This initializes the free surface elevation at 0 (default value). The initial water depths are therefore calculated from the bottom elevation,
- 'CONSTANT ELEVATION': This initializes the free surface elevation at the value supplied by the keyword `INITIAL ELEVATION` (default value = 0.). The initial water depths 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 water depth is zero,
- 'ZERO DEPTH': All water depths are initialized 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 initializes the water depths at the value supplied by the keyword `INITIAL DEPTH` (default value = 0.),
- 'TPXO SATELLITE ALTIMETRY': The initial conditions are set using information provided by the OSU harmonic constants database (TPXO for instance) in the case of the use of this database for the imposition of maritime boundary conditions (see subsection 4.2.12),

- 'PARTICULAR' or 'SPECIAL': The initial conditions are defined in the **USER_CONDIN_H** 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 five cases above.

### 4.1.2 Prescribing particular initial conditions with user subroutines

The **USER_CONDIN** subroutine must be programmed whenever the keyword `INITIAL CONDITIONS` has the value 'PARTICULAR' or 'SPECIAL'.
The **CONDIN** subroutine initializes successively the velocities, the water depth, the tracer and the viscosity. The part of the subroutine concerning the initialization of the water depth is divided into two zones. The first corresponds to the processing of simple initial conditions (defined by keywords) and the second regards the processing of particular initial conditions with the **USER_CONDIN_H** subroutine.
By default, the standard version of the **USER_CONDIN_H** subroutine stops the computation if the keyword `INITIAL CONDITIONS` is set at 'PARTICULAR' or 'SPECIAL' without the subroutine being actually 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 **USER_CONDIN_H** 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.
The user may also change initial velocities with the help of the **USER_CONDIN_UV** subroutine in addition to or without any treatment for initial water depth.

### 4.1.3 Continuing a computation

TELEMAC-2D enables the user to resume a computation taking a 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, some boundary conditions or the turbulence model, or to start the computation once a steady state has been reached.
By default, TELEMAC-2D reads the last time step of the previous computation result file. Using the keyword `RECORD NUMBER FOR RESTART` allows specifying the number of the iteration to be read.
Note that FUDAA-PREPRO is using this possibility: when defining the initial conditions, information is actually written in a pseudo continuation file.
In this case, it is essential that the continuation file contains all the information required by TELEMAC-2D, i.e. the velocities $U$ and $V$, the water depth and the bottom elevations. However, in some cases, the software is capable of recomputing some variables from others provided (for example the depth of water from the free surface and the bottom elevation).
If some 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 is to use 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 (default value = NO),

- 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 ('SERAFIND') or MED format ('MED'). 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 time step in the previous computation file used for continuation. This can be modified using the keyword INITIAL TIME SET TO ZERO (default = 0) if the user wants to reset the time value (possibly with respect to a basic value set in the preceding calculation. See Chapter 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 time step used for the continuation (the only one used by TELEMAC-2D).

## 4.2 Prescribing boundary conditions

The maximum number of boundaries is set to 30 by default but it can be changed by the user with the keyword MAXIMUM NUMBER OF BOUNDARIES. This avoids changing the previously hardcoded values (until release 7.0), which required recompiling the whole package.

### 4.2.1 Possible choices

Boundary conditions are given for each of the boundary points. They concern the main 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 $\varepsilon$ in the turbulence model are determined by TELEMAC-2D and are thus not required from the user. Turbulence specialists may want to change the boundary conditions of $k$ and $\varepsilon$ in **KEPSCL** subroutine.
The various types of boundary conditions may be combined to prescribe boundary conditions of any type (inflow or outflow of liquid in a supercritical or subcritical regime, open sea, wall, etc.). However, some combinations are not physical.
Some 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 located 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 subroutines **USER_Q**, **USER_SL** and **USER_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 some 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 **USER_BORD** subroutine is then necessary,
- The boundary condition type is variable in time. Direct programming in the **PROPIN_TELEMAC2D** subroutine is then necessary (see 14.7).

The type of boundary condition, if constant in time, is read from `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 (not frequently used), the steering file, the open boundaries file and the FORTRAN file (programming of subroutines **USER_Q**, **USER_SL**, **USER_VIT**, **TR** or **USER_BORD**). Boundary types may be connected in any way along a contour. However, two liquid boundaries must be separated by at least a solid segment (for example, there cannot be an open boundary with a prescribed depth directly followed by an open boundary with a prescribed velocity). Moreover, another 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 various types of boundary conditions

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,
  - 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,
  - Closed boundary with slip or friction: **LIUBOR/LIVBOR** = 2,
  - Closed boundary with one or two null velocity components: **LIUBOR** and/or **LIVBOR** = 0,
- 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,
- The type of boundary condition during the simulation may be modified with the **PROPIN_TELEMAC2D** subroutine (see 14.7).

### 4.2.3 The boundary conditions file

The file is normally supplied by most of the mesh generators compatible with TELEMAC, or STBTEL but may be created and modified using FUDAA-PREPRO or 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 some specific aspects):
**LIHBOR, LIUBOR, LIVBOR, HBOR, UBOR, VBOR, AUBOR, LITBOR, TBOR, ATBOR, BTBOR, N, K**.

- **LIHBOR, LIUBOR, LIVBOR** and **LITBOR** are the boundary type integers 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:

$$\nu_t \frac{dU}{dn} = AUBOR \times U \qquad \text{and/or} \qquad \nu_t \frac{dV}{dn} = AUBOR \times V$$

The **AUBOR** coefficient applies to the segment included between the boundary point considered and the following point (in the counter clockwise direction for the outside outline and in the clockwise direction for the islands). The default value is **AUBOR** = 0. Friction corresponds to a negative value. With the $k$-$\varepsilon$ 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 flux law which is written as:

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

The **ATBOR** and **BTBOR** coefficients apply to the segment between the boundary point considered and the next point (in the counter clockwise direction for the outside outline and in the clockwise direction for the islands).

- **N** represents the global number of boundary points.

- **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. Be careful not to modify the last column of the boundary conditions file that contains this **BOUNDARY_COLOUR** table, when using tidal harmonic constants databases (cf. [10]).

### 4.2.4 Prescribing values through 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 elevation (free surface). It is an array that can contain up to **MAXFRO** (set to 300 and can be changed by user) real numbers for managing up to **MAXFRO** boundaries of this type. The values defined by that keyword overwrite the depth values read from the `BOUNDARY CONDITIONS FILE`.

N.B.: the value given here is the free surface level, whereas the value given in the `BOUNDARY CONDITIONS FILE` is the water depth.

- `PRESCRIBED FLOWRATES`: This is used to set the flowrate value of an open boundary with prescribed flowrate. It is an array that contains up to **MAXFRO** real numbers for managing up to **MAXFRO** boundaries of this type. A positive value corresponds to an inflow into the domain, whereas a negative value corresponds to an outflow. The values provided with this keyword overwrite 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 set 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 an array that contains up to **MAXFRO** real numbers for managing up to **MAXFRO** boundaries of this type. The values provided with this keyword overwrite 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),

- If a boundary type is defined in the `BOUNDARY CONDITIONS FILE`, the corresponding keyword must be defined in the steering file,

- The keywords `PRESCRIBED ...`, if present, supersede the data read in the `BOUNDARY CONDITIONS FILE`,

- 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 subroutines 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 subroutine, which may be:

- Subroutine **USER_VIT** to prescribe a velocity,
- Subroutine **USER_Q** to prescribe a flowrate,
- Subroutine **USER_SL** to prescribe an elevation,
- Function **TR** to prescribe a tracer concentration (see Chapter 9)

Subroutines **USER_Q**, **USER_VIT** and **USER_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), the global number of the boundary point (useful in case of parallel computing) and in the case of **USER_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 subroutine **USER_Q** for prescribing a flowrate ramp lasting 1,000 seconds and reaching a value of 400 $m^3/s$ could take a form similar to:

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

Using the liquid boundaries file is an alternative to programming the subroutines mentioned above. This is an ASCII 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** (**T** meaning time) 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 **TR** for tracer. An integer between brackets specifies the rank of the boundary in question. In the case of tracers, the identification, uses a 2-index mnemonic **TR**(b,t) with b providing the rank of the boundary and t the number of the tracer. 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 will stop.

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.
```

**Note:**
Up to release 7.0, it is necessary to have the corresponding keywords `PRESCRIBED ...` to trigger the use of the liquid boundaries file.

Since release 8.2, a time reference can be given: If a #REFDATE with a date + hour in YYYY-MM-DD HH:MM:SS in year, month, day, hour, minute, second format is written in the file (after first # line) the date+hour will be added to the times in these ASCII files.

### 4.2.6 Stage-discharge curves

It is possible to manage a boundary where the prescribed value of the elevation is a function of the local discharge (and vice versa). 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 requires one integer per liquid boundary. This integer can be:

- 0: no stage-discharge curve (default value),
- 1: elevation as function of discharge. In that case, the boundary has to be defined as a prescribed elevation boundary,
- 2: discharge as function elevation. In that case, the boundary has to be defined as a prescribed discharge boundary.

The keyword `STAGE-DISCHARGE CURVES FILE` supplies the name of the ASCII file containing the curves. One example is shown 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.
```

The order of curves is not important. The columns order 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.

N.B.: at initial conditions the discharge at exits may be null. 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.

When using value 1 for the keyword `STAGE-DISCHARGE CURVES`, the relation between elevation and discharge may not be exactly the expected values. Indeed, there is a delay (relaxation) to avoid triggering resonances and high oscillations by means of a relaxation coefficient. This coefficient allows to smooth high gradients of the prescribed values. If set to 1., the elevation is instantaneously prescribed corresponding to the stage-discharge curve, but this may lead to instabilities. Setting a value between 0. and 1. is a compromise between the goal of the stage-discharge curve and possible instabilities.

Since release 8.3, the relaxation coefficient can be changed with the keyword `STAGE-DISCHARGE CURVES RELAXATION COEFFICIENT` (default value = 0.02). This keyword substitutes the old hard-coded value of 0.02 at the end of function **STA_DIS_CUR** (this value could have been changed manually).

### 4.2.7 Prescribing complex values

If the values to be prescribed vary in both time and space, it is necessary to program the **USER_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 **USER_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 **USER_BORD** subroutine also enables the tracer limit values to be prescribed (see 9.3).

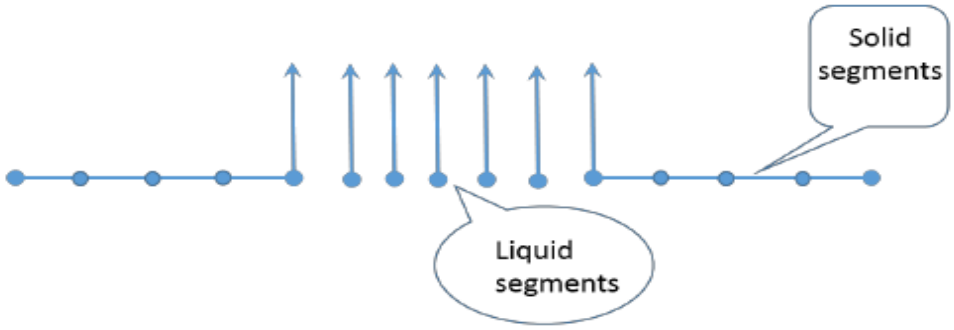


Figure 4.1: Bad prescription of velocity profile.

### 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 (default value for all boundaries). 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 subroutine **USER_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 subroutine **USER_VIT**. In any case, the velocity profile is constant along the boundary. It is the default configuration,
- 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.
- 5: The velocity vector is normal to the boundary and its norm is proportional to the square root of the virtual water depth computed from the lower point of the free surface at the boundary.

In the case of a flow normal to a closed boundary, it is not recommended to have velocities perpendicular to the solid segments (as shown in the Figure 4.1),
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 Figure 4.2.

### 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.

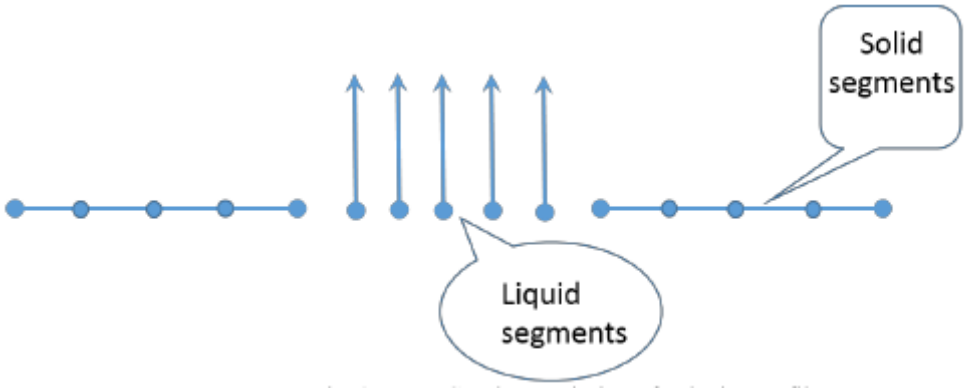


Figure 4.2: Good prescription of velocity profile.

To solve this problem, the Thompson method uses the theory of characteristics 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, the Thompson technique computes new values that will comply with the theory of characteristics.
For this, the user can use the keyword `OPTION FOR LIQUID BOUNDARIES`, which offers two values (the user must specify one value for each open boundary):

- 1: strong setting (default value for all boundaries),
- 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:

1. This keyword should be given for ALL liquid boundaries like given in the following example:
   `OPTION FOR LIQUID BOUNDARIES`= 2;1;2;2
   which means that you are asking TELEMAC-2D to apply Thompson boundary condition to boundaries number 1,3 and 4; unlike boundary number 2, where TELEMAC-2D will use a strong prescription.
2. This option will trigger the computation of characteristics’ trajectories in order to get informations from inside the domain.

### 4.2.10 Elements Masking

TELEMAC-2D offers the possibility of masking some 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 **USER_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.
N.B.: This option is not compatible with perfectly conservative advection schemes.

### 4.2.11 Definition of types of boundary condition when preparing the mesh

When using Blue Kenue, 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 of some meshers and corresponding types of boundary is given in Appendix E.

### 4.2.12 Tidal harmonic constituents databases

**General parameters**

To prescribe the boundary conditions of a coastal boundary subject to tidal evolution, it is generally necessary to have the information characterizing this phenomenon (harmonic constants). One of the most common cases is to use the information provided by large scale models.
4 databases of harmonic constants are interfaced with TELEMAC-2D:

- the JMJ database resulting from the LNHE Atlantic coast TELEMAC model by Jean-Marc JANIN,
- the global TPXO database and its regional and local variants from the OSU (Oregon State University),
- the regional North-East Atlantic atlas (NEA) and the global atlas FES (e.g. FES2004 or FES2012) coming from the works of LEGOS (Laboratoire d'Etudes en Géophysique et Océanographie Spatiales),
- the PREVIMER atlases.

However it is important to note that, in the current release of the code, the latter 2 databases are not completely interfaced with TELEMAC-2D and their use is recommended only for advanced users.
The keyword `OPTION FOR TIDAL BOUNDARY CONDITIONS` activates the use of one of the available database when set to a value different from 0 (the default value 0 means that this function is not activated). Since release 7.1, this keyword is an array of integers separated by semicolons (one per liquid boundary) so that the user can describe whether tidal boundary conditions should be computed or not (e.g. a weir) on a liquid boundary. When this keyword is activated, every tidal boundary is treated using the prescribed algorithms for the boundaries with prescribed water depths or velocities, with the same option for tidal boundary conditions (the values not equal to 0 have to be the same), except the boundaries with prescribed flowrate.
The database used is specified using the keyword `TIDAL DATA BASE` which can take the values:

- 1: JMJ,
- 2: TPXO,
- 3: Miscellaneous (LEGOS-NEA, FES20XX, PREVIMER...).

Depending on the database used, some keywords have to be specified:

- if using the JMJ database, the name of the database (typically bdd_jmj) is given by the keyword ASCII DATABASE FOR TIDE and the corresponding mesh file is specified using the keyword TIDAL MODEL FILE,
- if using TPXO database, the name of the water level database is given by the keyword BINARY DATABASE 1 FOR TIDE (for example h_tpxo7.2) and the name of the velocity database is given by the keyword BINARY DATABASE 2 FOR TIDE (for example u_tpxo7.2). Moreover, it is possible to activate an interpolation algorithm of minor constituents from data read in the database using the logical keyword MINOR CONSTITUENTS INFERENCE, activation not done by default.

The keyword OPTION FOR TIDAL BOUNDARY CONDITIONS allows specifying the type of tide to prescribe. Default value 0 means no prescribed tide or that the tide is not treated by standard algorithms. Value 1 corresponds to prescribing a real tide considering the time calibration given by the keywords ORIGINAL DATE OF TIME (YYYY ; MM ; DD format) and ORIGINAL HOUR OF TIME (HH ; MM ; SS format). Other options are the following, available for every tidal database (JMJ, TPXO-type from OSU, LEGOS-NEA, FES, PREVIMER ...). They are called "schematic tide" for values from 2 to 6:

- 2: exceptional spring tide (French tidal coefficient approximately equal 110),
- 3: mean spring tide (French tidal coefficient approximately equal 95),
- 4: mean tide (French tidal coefficient approximately equal 70),
- 5: mean neap tide (French tidal coefficient approximately equal to 45),
- 6: exceptional neap tide (French tidal coefficient approximately equal to 30),
- 7: real tide (before 2010 methodology, only available with JMJ).

In the case of options 2 to 6 (schematic tides), the boundary conditions are imposed so that the reference tide is approximately respected. In order to shift the phases of the waves of the tidal constituents so that the computation starts close to a High Water, two keywords are available. If using a TPXO-type tidal database from Oregon State University, the keyword GLOBAL NUMBER OF THE POINT TO CALIBRATE HIGH WATER has to be filled with the global number of the point (between 1 and the number of boundary nodes in the 2D mesh) with respect to which the phases are shifted to start with a high water (mandatory, otherwise the computation stops). This point has to be a maritime boundary node. If using one of the other tidal databases (JMJ, NEA/FES, PREVIMER) the keyword LOCAL NUMBER OF THE POINT TO CALIBRATE HIGH WATER should be filled in with the local number between 1 and the number of tidal boundary points of the HARMONIC CONSTANTS FILE; If not filled in (default value = 0), a value is then automatically calculated. However, it is usually necessary to wait for the second or third modelled tide in order to overcome the transitional phase of start-up of the model. It is also necessary to warn the user that the French tidal coefficients shown are approximate.
During a simulation, data contained in the tidal database are interpolated on boundary points. When using the JMJ database, this spatial interpolation can be time consuming if the number of boundary points is important, and is not yet available in case of parallel computing. It is therefore possible to generate a file containing harmonic constituents specific to the model treated.

The principle is at a first step, to perform a calculation on a single time step whose only goal is to extract the necessary information and to generate a file containing for each boundary point of the model, the harmonic decomposition of the tidal signal. Subsequent calculations directly use that specific file rather than directly addressing to the global database. The harmonic constants specific file is specified using the keyword HARMONIC CONSTANTS FILE, this file is an output file in the first calculation, and an input file in subsequent calculations.

If using tidal solutions coming from OSU (e.g. TPXO), to get velocity components, it is necessary to divide the transports terms (water depth times velocity components) by water depth. A minimum value of water depth to get them is taken to avoid divisions by 0. Since release 8.2, it is possible to change this old hard coded value of 0.1 m, both for boundary conditions and initial conditions, with the keywords MINIMUM DEPTH TO COMPUTE TIDAL VELOCITIES BOUNDARY CONDITIONS and MINIMUM DEPTH TO COMPUTE TIDAL VELOCITIES INITIAL CONDITIONS. Both default values are 0.1 m. Moreover, for initial conditions, if water depth is below MINIMUM DEPTH TO COMPUTE TIDAL VELOCITIES INITIAL CONDITIONS, the velocity components are set to 0. These 2 keywords enable to decrease artificially too high velocities in particular at open boundaries with shallows or if the tidal solutions have shallows at the same location. If this effect of the latter keyword is not sufficient, the user can initialise the computation with no velocities with INITIAL VELOCITIES COMPUTED BY TPXO = NO (default = YES, i.e. the tidal velocities are computed by algorithms to use OSU tidal solutions).

**Horizontal spatial calibration**

In order to perform the spatial interpolation of the tidal data, it is imperative to provide to TELEMAC-2D information on the spatial positioning of the mesh model relative to the grid of the tidal database. To do this, the user has two keywords:
The first keyword specifies the geographic system used to establish the coordinates of the 2D mesh of TELEMAC-2D. This keyword GEOGRAPHIC SYSTEM, which has no default value, may take the following values:

- 0: User Defined,
- 1: WGS84 longitude/latitude in real degrees,
- 2: WGS84 UTM North,
- 3: WGS84 UTM South,
- 4: Lambert,
- 5: Mercator projection.

The second keyword is used to specify the area of the geographic system used to establish the coordinates of the 2D mesh of TELEMAC-2D. This keyword ZONE NUMBER IN GEOGRAPHIC SYSTEM which has no default value, may take the following values:

- 1: Lambert 1 North,
- 2: Lambert 2 Center,
- 3: Lambert 3 South,
- 4: Lambert 4 Corsica,

- 22: Lambert 2 extended,
- 93: Lambert 93,
- X: UTM zone value of the WGS84 (X is the number of the zone).

If using the Lambert 93 projection, the user has to copy the file provided in the tide examples of TELEMAC-2D called gr3df97a.txt which is used for the conversion in the Lambert 93 projection. The keyword LAMBERT 93 CONVERSION FILE has to indicate the path and the name of the gr3df97a.txt file.

Since release 8.2, it is possible to use $x$ and $y$ origin coordinates stored in the geometry file to decrease the number of digits of coordinates when modelling tide, e.g. when using UTM or Lambert projections. The two numbers are stored in the **I_ORIGIN** and **J_ORIGIN** variables reachable with the help of **GET_MESH_ORIG** subroutine. Caution: these two numbers are integers, not floats as the preliminary structure available in the SERAFIN format expected this type for these 2 variables. The tidal computations automatically take into account this offset for UTM + Lambert projections when generating the HARMONIC CONSTANTS FILE for JMJ database or when interpolating harmonic constants to compute boundary or initial conditions for solutions coming from OSU (e.g. TPXO). In particular for the operations to locate the nodes correctly (a simple translation) but TELEMAC-2D still continues to compute other steps in a local coordinate system.

**Calibration of the information**

The transfer of information between a large scale model and the boundaries of a more local model generally requires calibration.
To do this, the user has three keywords:

- the keyword COEFFICIENT TO CALIBRATE SEA LEVEL (default real value 0.) allows to calibrate the mean tide level (the harmonic decomposition of information provided by the various databases are used to generate the tidal signal oscillating around mean tide level). The calibration of the mean tide level must obviously be made depending on the altimetric reference used in the model,
- the keyword COEFFICIENT TO CALIBRATE TIDAL RANGE (default real value 1.) allows to specify a calibration coefficient applied on the amplitude of the tidal wave. This coefficient is applied to the amplitude of the overall signal, and not on the amplitude of each of the elementary waves,
- the keyword COEFFICIENT TO CALIBRATE TIDAL VELOCITIES (default real value 999,999.0) allows to specify the coefficient applied on velocities. The default value (999,999.0) means that the square root of the value specified by the keyword COEFFICIENT TO CALIBRATE TIDAL RANGE tidal is used.

For more information, the reader may refer to the methodological guide for tide simulation with version 6.2 [10].

# 5. General parameter definition for the computation

General setup of the computation is only done in the steering file.
Time information is supplied by the three keywords: TIME STEP (real set at 1. by default), NUMBER OF TIME STEPS (integer set at 1 by default) and DURATION (real set at 0. by default). The first one 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 both 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) whose default value is (1900;1;1) i.e. January 1$^{st}$ 1900 and ORIGINAL HOUR OF TIME (HH;MM;SS) whose default value is (0;0;0) i.e. midnight. This is particularly important if the tide generating forces are taken into account (see 6.4) and are generally necessary when using tidal harmonic constituents databases.
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 (default = NO). It is then possible to define the permissible area of tolerance using the keyword STOP CRITERIA (default value = (1.E-4 ; 1.E-4 ; 1.E-4)). This keyword is an array of three real numbers, representing the tolerance assigned to the velocity, water depth and tracer. The computation is stopped when the absolute increment values of these variables be-

tween 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 stop 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 (default = NO). 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 for $H$ (by default -1,000 m),
  - The maximum depth value for $H$ (by default +9,000 m),
  - The minimum velocity value for $U$ (by default -1,000 m/s),
  - The maximum velocity value for $U$ (by default +1,000 m/s),
  - The minimum velocity value for $V$ (by default -1,000 m/s),
  - The maximum velocity value for $V$ (by default +1,000 m/s),
  - The minimum tracer value (by default -1,000),
  - The maximum tracer value (by default +1,000).

## 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 computing 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 keyword COMPATIBLE COMPUTATION OF FLUXES (default = NO). The difference may reach a few percents.
It is also possible to obtain the cumulated flow rates in the listing for each control section by activating the logical keyword PRINTING CUMULATED FLOWRATES (default value = NO). 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 a keyword and is not valid when running in parallel mode. The second one (available since 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 an 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 and steering integer. For the steering integer, the convention is as follow:
  - if negative (< 0): node numbers (global numbers) are given,
  - if equal to 0: coordinates are given. Note that if positive (> 0) and parallel mode, coordinates can also given, BUT the combo positive steering integer (> 0) and serial mode does not work!
- 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 **FLUXPR_TELEMAC2D** subroutine (TELEMAC-2D library).
The printouts will be in the file named by the keyword `SECTIONS OUTPUT FILE`.

## 5.3 Computation of fluxes over lines (FLUXLINE)

It is possible to compute fluxes over lines by using `FLUXLINE` = YES (default = NO). The input file with data on cross-sections has to be given with the keyword `FLUXLINE INPUT FILE`. The format of this ASCII file is:

- the first line contains the number of lines to be read $n$,
- and then $n$ lines of 9 floating numbers describing each fluxline.

# 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 **USER_DEF_ZONES** subroutine 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. Another possibility is to fill the `ZONES FILE`, see Appendix E.
The spatial distribution of some parameters may be specified interactively using FUDAA-PREPRO (friction coefficient and velocity diffusivity coefficient especially).

## 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 E 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 is only for boundary conditions,
In the case of options 1 to 6, 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, an additional coefficient is needed which can be given by the keyword MANNING DEFAULT VALUE FOR COLEBROOK-WHITE LAW (whose default value = 0.02).
If the friction coefficient varies in time (and possibly in space as well), it is necessary to use the **USER_STRCHE** and/or **USER_CORSTR** subroutines, which supply the friction coefficient at each mesh point.
The following example shows how **USER_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.

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

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:

- 1: the classical nodal depth (default value),
- 2: a depth averaged on the test function area.

The second one is available since release 6.0 and seems to be slightly better on dam break studies.

### 6.1.1 Vegetation friction

The effect of additional roughness by vegetation can be added with the keyword VEGETATION FRICTION (default = NO).
This option can only be used with a definition of friction by domains, see Appendix E. The vegetation laws and their parameters must be given in an extra file. Currently 8 vegetation laws are implemented. Please find the details of the vegetation laws there description and application range in [14] and the cited original literature.
In case of unsteady flow conditions which lead to submerged and non-submerged vegetation the vegetation law of Baptist is recommended. In order to take the flexibility of vegetation into account the approach of Jaervelae is recommended. The Lindner approach is not recommended as it can increase the overall computing time by approx. 20 % because of the iteration of the drag coefficient. This is only useful if the shape of the vegetation is cylindrical, regularly arranged, non-submerged and diameter and spacing are well known.

### 6.1.2 Wave friction enhancement

Combined wave and current flows influence the resultant bed shear stress due to additional mass transport [6]. A proper consideration of the wave–current interaction effects on the friction coefficient can be accounted by activating with the keyword WAVE ENHANCED FRICTION FACTOR (default = NO). This feature is only possible by coupling TELEMAC-2D with TOMAWAC.

### 6.1.3 Sidewall friction

By default, TELEMAC-2D ignores friction phenomena on the solid boundaries of the model (sidewall). This consideration may still be enabled using the keyword LAW OF FRICTION ON LATERAL BOUNDARIES (default = 0 i.e. no friction). This keyword offers the same

option than the keyword LAW OF BOTTOM FRICTION described above. The coefficient of friction to take into account is then provided by the keyword FRICTION COEFFICIENT FOR LATERAL SOLID BOUNDARIES (default value = 60!) if it is constant on all solid boundaries. If this value varies spatially, the user can fill the column **AUBOR** in the BOUNDARY CONDITIONS FILE (see Section 4.2.3). **AUBOR** is then considered as a quadratic coefficient whose interpretation depends on the chosen friction law.
The friction is activated using the keyword TURBULENCE REGIME FOR SOLID BOUNDARIES. The available options are:

- 1: smooth regime,
- 2: rough (default value).

That option changes the formulation of the velocity profile and consequently, the friction velocity. See previous section for more information.

### 6.1.4 Non-Newtonian fluid

The non-Newtonian behaviour of the fluid can be modeled with a friction source term in the finite volume framework. This can be activated using the keyword EQUATIONS = 'SAINT-VENANT FV' coupled with NON-NEWTONIAN MODEL:

- 0: Newtonian model (default),
- 1: Bingham,
- 2: Herschley-Bulkley.

If the Bingham model is chosen, three options are available to estimate the exact solution of the model, through the keyword BINGHAM OPTION:

- 1: Papanastasiou's (1987) [24] exponential regularization (default),
- 2: Effective viscosity with cross formulation,
- 3: Rickenmann's (1990) [25] Cubic equation.

If the Herschley-Bulkley model is chosen, it is possible to choose the power law index (real number) of the model with the keyword HERSCHEL-BULKLEY POWER-LAW INDEX.
Several physical parameters of the fluid can be set, like its viscosity, with the keyword NON-NEWTONIAN VISCOSITY (default = 0 Pa.s), its yield stess with the keyword NON-NEWTONIAN YIELD STRESS (default = 0 Pa), and its density with the keyword NON-NEWTONIAN FLUID DENSITY (default = 1000 kg/m$^3$). The non-Newtonian laminar resistance can also be modeled with the $K$ parameter set by NON-NEWTONIAN LAMINAR RESISTANCE PARAMETER K (default = 24).
In addition, a simplified pseudo-biphasic model is available in order to simulate the effect of a spatially varying volumetric sediment concentration on fluid density and rheological parameters. The volumetric sediment concentration should be specified with a tracer with value ranging between 0 (zero sediment concentration, water density) and the considered volumetric sediment concentration that should be positive and lower than 1 (nominal volumetric sediment concentration, nominal non-Newtonian fluid density). The yield stress and dynamic viscosity are computed as a function of the volumetric sedimentation concentration with power laws and their coefficients that need to be defined in subroutine **NONNEWT_FV**. To activate this model, the user needs to specify NON-NEWTONIAN PSEUDO-BIPHASIC MODEL = YES (default =

NO). The tracer values should then be specified as explained in part 9. When using the pseudo-biphasic model, the sediment specific density (grains, typically 2650 kg/m$^3$ or higher) should be specified with the keyword NON-NEWTONIAN FLUID DENSITY.
More information about the theoretical background and the simplified pseudo-biphasic model can be found in the validation documentation available in the folder examples/telemac2d/nn_newt.

## 6.2 Modelling of turbulence

Whether or not the diffusion of velocity is taken into account is determined by the logical keyword DIFFUSION OF VELOCITY (default value = YES).
The modelling of turbulence is a delicate problem. TELEMAC-2D offers the user six options of different complexity by setting the keyword TURBULENCE MODEL:

- 1: a constant viscosity coefficient. In this case, the coefficient represents the molecular viscosity, turbulent viscosity and dispersion,
- 2: an Elder model,
- 3: a $k$-$\varepsilon$ model. This is a 2D model that solves the transport equations for $k$ (turbulent energy) and $\varepsilon$ (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. The use of the $k$-$\varepsilon$ model also often requires a finer mesh than the constant viscosity model and in this way increases computational time,
- 4: a Smagorinski model, generally used for maritime domains with large-scale eddy phenomena,
- 5: a mixing length model,
- 6: a Spalart-Allmaras model. This is a 2D model that solves a transport equation for $\tilde{\nu}$ (a derivated turbulent viscosity).

More detailed information on the formulation of the $k$-$\varepsilon$ model, the Elder model, the Smagorinski model, the mixing length model and the Spalart-Allmaras 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 $\mathrm{div}\left(\vartheta_t \overrightarrow{grad}\,(U)\right)$, and the second one with the form $\frac{1}{h}\mathrm{div}\left(\mathrm{h}\vartheta_\mathrm{t}\overrightarrow{grad}\,(\mathrm{U})\right)$.
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 $10^{-6}$ m$^2$/s (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 or she 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 mesh cells has virtually no effect on the computation.
TELEMAC-2D makes it possibile to have 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 (water depth, 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 ($K_l$ and $K_t$ respectively). The formulae used are:

$$K_l = a_l U^* h \quad \text{and} \quad K_t = a_t U^* h \tag{6.1}$$

where:
$U^*$ is the friction velocity (m/s) and $h$ the water depth (m), $a_l$ and $a_t$ 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 $k$-$\varepsilon$ model

If constant viscosity is not sufficient, TELEMAC-2D offers the possibility of using a $k$-$\varepsilon$ model. This is activated by assigning a value of 3 to the keyword `TURBULENCE MODEL`.
In this case, the keyword `VELOCITY DIFFUSIVITY` has its real physical value ($10^{-6}$ m$^2$/s 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 REGIME 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` (default value is 100). This numerical value must of course be in agreement with the chosen law, in appropriate units.
If a $k$-$\varepsilon$ model is used, the information concerning the solution phase must be obtained by activating the keyword `INFORMATION ABOUT K-EPSILON MODEL` (default = YES).
Parameter definition for the $k$-$\varepsilon$ model is described in Chapter 7.
A good level of expertise in turbulence is necessary to use the $k$-$\varepsilon$ 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 as the $k$-$\varepsilon$ model, it does not take into account the dispersion terms (see [19] for more details).

### 6.2.5 Mixing length model

The use of this model is activated by giving the value 5 to keyword TURBULENCE MODEL. A description of this model, implementation and validation can be found in [9]. It is a combination of the depth-averaged parabolic eddy viscosity model with the Prandl's mixing length theory for the horizontal in order to account both the vertical and horizontal turbulence production. The calibration coefficients $C_l$ and $\alpha_t$ used in the **MIXLENGTH** subroutine may be changed by the means of the MIXING LENGTH MODEL COEFFICIENTS which is an array of 2 values in the same order: $C_l$ ; $\alpha_t$. Default values are $C_l = 0.1066667$ ($C_l = \frac{4\kappa}{15}$) and $\alpha_t$ = 0.0666667 ($\alpha_t = \frac{\kappa}{6}$) with $\kappa$ the Von Karman constant set by default at 0.4 in TELEMAC-2D.
Warning: until release 8.1, the $C_l$ default value was hard-coded in the **MIXLENGTH** subroutine at $\frac{4}{15}$ without the $\kappa$ factor.

### 6.2.6 Spalart-Allmaras model

The use of this model is activated by giving the value 6 to keyword TURBULENCE MODEL. If a Spalart-Allmaras model is used, the information concerning the solution phase can be obtained by activating the keyword INFORMATION ABOUT SPALART-ALLMARAS MODEL (default = YES).
Parameter definition for the Spalart-Allmaras model is described in Chapter 7. Some parameters are common with the $k-\varepsilon$ model's ones.

## 6.3 Setting up of meteorological phenomena

### 6.3.1 Wind influence

TELEMAC-2D can be used to simulate flow while taking into account the influence of a wind blowing on the water surface. The force induced by wind is considered in the same way as the friction effect on the bottom. The following force is consequentely added to the right handside term of the momentum equation:

$$F_x = \frac{1}{h}\frac{\rho_{air}}{\rho_{water}} a_{wind} U_{wind} \sqrt{U_{wind}^2 + V_{wind}^2}, \tag{6.2}$$

$$F_y = \frac{1}{h}\frac{\rho_{air}}{\rho_{water}} a_{wind} V_{wind} \sqrt{U_{wind}^2 + V_{wind}^2}. \tag{6.3}$$

This expression does not consider the wind influence as drag force, like it is the case commonly. In order to retrieve a drag-like expression, the reader should write this force in the following form: $F_x = \frac{1}{2}\rho_{air} C_d U^2 A$ where $C_d$ is the drag coefficient, $A$ the effective area and $U$ is the velocity magnitude of the wind. The logical keyword WIND (default = NO, i.e. no wind) is used first of all for determining whether this influence is to be taken into account. If so, the coefficient is then provided with the keyword COEFFICIENT OF WIND INFLUENCE (see below) or automatically calculated if COEFFICIENT OF WIND INFLUENCE VARYING WITH WIND SPEED = YES which is the default value since release 8.2. Since release 7.0, wind effect is managed using a new keyword, OPTION FOR WIND (default = 0): this keyword can have the following values:

- 0: this means no wind effect (this is equivalent to put the keyword WIND to FALSE),
- 1: wind is constant in time and space, wind speed in directions $x$ and $y$ are supplied with the keywords WIND VELOCITY ALONG X and WIND VELOCITY ALONG Y (default values = 0.), or through the keyword SPEED AND DIRECTION OF WIND (default = 0.;0.) which gives the speed (in m/s) and the direction (in degrees from 0 to 360) of the wind,

- 2: wind variable in time, constant is space, it is given through the formatted file ASCII ATMOSPHERIC DATA FILE. Shortnames WINDS and WINDD (for wind velocity magnitude and direction) or WINDX and WINDY (for $x$ and $y$ wind velocity components) are to be written in the headline of this file (see the section 3.1 of the WAQTEL user manual),

- 3: wind is variable in time and space, this option is not implemented as there are multiple choices of implementation. In this case, the user must program him/herself the **METEO** subroutine. If the ASCII ATMOSPHERIC DATA FILE or the BINARY ATMOSPHERIC DATA FILE does not follow the format expected by the **METEO_TELEMAC** module and implementation is required to handle them outside this module, the keyword FREE FORMAT FOR ATMOSPHERIC DATA FILE has to be set to YES (default = NO). An example of implementation is given in validation test case "wind_txy" (in folder examples/telemac2d/wind_txy).

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):

$$\begin{array}{ll} \text{if } \|\boldsymbol{U}_{wind}\| < 5 \text{ m/s} & a_{wind} = 0.565 \times 10^{-3} \\ \text{if } 5 < \|\boldsymbol{U}_{wind}\| < 19.22 \text{ m/s} & a_{wind} = (-0.12 + 0.137\|\boldsymbol{U}_{wind}\|)10^{-3} \\ \text{if } \|\boldsymbol{U}_{wind}\| > 19.22 \text{ m/s} & a_{wind} = 2.513 \times 10^{-3} \end{array}$$

This formula can be activated by setting the keyword COEFFICIENT OF WIND INFLUENCE VARYING WITH WIND SPEED to YES (default value is YES since release 8.2). If YES, the value of COEFFICIENT OF WIND INFLUENCE is overwritten and the coefficient is automatically computed depending on the wind velocity.
If the previous keyword is set to NO (no automatic computation), the parameter COEFFICIENT OF WIND INFLUENCE asked for by TELEMAC-2D is: $a_{wind}\frac{\rho_{air}}{\rho}$ and not only $a_{wind}$.
$\rho_{air}$ is approximately 1.2 kg/m$^3$ and $\rho$ is 1,000 kg/m$^3$. Thus it is necessary to divide the value of $a_{wind}$ by 1,000 to obtain the value of the TELEMAC-2D keyword. The default value of COEFFICIENT OF WIND INFLUENCE has been set to $1.55 \times 10^{-6}$ since release 8.2.
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, the influence of the wind is cancelled below a threshold value of depth, with the keyword THRESHOLD DEPTH FOR WIND (default value = 1 m). Be careful if the model includes shallow waters, lower than this value.

### 6.3.2 Atmospheric pressure

Atmospheric pressure is taken into account by setting the keyword AIR PRESSURE to YES (the default value is NO). Since release 7.0, the pressure value is set in the **METEO** subroutine by the keyword VALUE OF ATMOSPHERIC PRESSURE (default $10^5$ Pa). By default, the latter initializes a pressure of $10^5$ Pa ($\approx$ 1 atm) over the whole domain.
The **METEO** subroutine is called if the wind or atmospheric pressure or water quality options are activated. By default, the subroutine is called only at the beginning of the computation (time value = 0) in order to set the wanted pressure throughout the domain and the wind speed at the values provided by the corresponding keywords.

The user has geometrical information on the mesh, and as well as time information for programming any case, in particular winds that may 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 ($X < 1,000,000$ m) the wind in direction $x$ is fixed at 10 m/s for the first 3,600 s, and at 5 m/s subsequently. The $x$ and $y$ wind components in the right part of the domain are nil.

```
! 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
! INITIALISATION WINDX LEFT PART FOR NON-ZERO TIMES
   DO I=1,NPOIN
     IF (X(I).LT.1000000.D0) THEN
       IF (LT.LT.3600.D0) THEN
         WINDX (I) = 10.D0
       ELSE
         WINDY (I) = 5.D0
       ENDIF
     ENDIF
   ENDDO
 ENDIF
```

Atmospheric pressure can be given through the formatted file ASCII ATMOSPHERIC DATA FILE. Shortname PATM is to be written in the headline of this file (see the section 3.1 of the WAQTEL user manual)

### 6.3.3 Rain and evaporation

The modelling of the influence of precipitation or evaporation is activated with the logical keyword RAIN OR EVAPORATION (default value = NO). The value of the contribution or the loss of water at the surface is specified using the keyword RAIN OR EVAPORATION IN MM PER DAY which default value is 0. (a negative value reflects an evaporation). The duration of the rainfall or evaporation can be set with the keyword DURATION OF RAIN OR EVAPORATION IN HOURS (units: hours, default is infinite).

Rain and evaporation can also vary in time and space. They can be introduced through the ASCII ATMOSPHERIC DATA FILE or BINARY ATMOSPHERIC DATA FILE. Shortnames RAINI and RAINC (with a final I for rain classically interpolated as other meteo variables or with a final C as a cumulated variable during a certain period of time) are to be written in the headline of this file (see the section 3.1 of the WAQTEL user manual). Also see water quality, wind and rain validation test cases (in folders examples/telemac2d or examples/waqtel).

In case of calculation with consideration of tracers, it is possible to specify the contribution related to the rain with the keyword VALUES OF TRACERS IN THE RAIN (default value is 0.). It is important to note that, in the case of evaporation, no tracer is taken into account in the water loss, which is incorrect if the tracer is the temperature.

### 6.3.4 Rainfall-runoff Modelling

Rainfall-runoff can be modelled with the Curve Number runoff model developed by USA Soil Conservation Service [31]. It takes into account infiltration processes to compute the effective rain induced runoff. Runoff potential is defined by a unique parameter called the Curve Number

(CN) which is function of hydrological soil groups, land use, hydrologic surface condition of native pasture and antecedent moisture conditions. See Applied Hydrology [12] for more details and typical CN values.
The Curve Number runoff model is activated when the keyword RAINFALL-RUNOFF MODEL is set to 1 (default is 0, no runoff model). The keyword TIDAL FLATS must be set to YES (which is default value). The CN values should be spatially defined. Two methods are available: coordinates of polygons with constant CN values given in a formatted data file (FORMATTED DATA FILE 2, default) or CN values stocked directly in the GEOMETRY FILE as an additional variable (see validation case pluie in folder examples/telemac2d).
The antecedent moisture conditions class can be defined with the keyword ANTECEDENT MOISTURE CONDITIONS (1: dry, 2: normal [default], 3: wet). Two options regarding the definition of the initial abstraction ratio are available and can be set with the keyword OPTION FOR INITIAL ABSTRACTION RATIO (1: original method with $\lambda = 0.2$ [default]; 2: revised method with $\lambda = 0.05$ and automatic conversion of CN values). CN values to be given in input must correspond to the original method (initial abstraction ratio $\lambda = 0.2$) and to normal antecedent moisture conditions. See [43] for more details about the initial abstraction ratio.
Other options can be activated manually in the **RUNOFF_SCS_CN** subroutine (in folder sources/telemac2d):

- Correction of CN values to account for steep slopes,
- Rainfall defined as a so-called CDS-type hyetograph (Chicago Design Storm) based on a three-parameter Intensity-Duration-Frequency equation (constant in space),
- Rainfall defined as a block-type hyetograph giving the rainfall depth in mm between two consecutive times provided in a formatted data file (constant in space).

Several examples of use are provided in the validation test case pluie (in folder examples/telemac2d).
Evaporation is not supported.

## 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. If YES, the keyword SPHERICAL COORDINATES has to be activated, it is impossible to account tide generating force in cartesian coordinates.
The keyword LONGITUDE OF ORIGIN POINT must be positioned at the right value (default = 0 degree).
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 Wave induced currents

We describe here the chaining procedure. A more dynamic solution, coupling, is described in section 14.8 and should be preferred.
It is possible to include wave-induced currents by recovering the information calculated by the wave propagation modules (mainly 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 driving 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` (default value = 1). 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 take into account changes in sea level), FORTRAN programming is required.

The user can also have variables stored, not used by TELEMAC-2D but used when coupling TELEMAC-2D with another code. Thus the stored variables belong to the other code and are given back in the `RESULTS FILE`. To do this, the user can set the `NAMES OF CLANDESTINE VARIABLES` keyword and implement what he/she wants to do.

## 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` (default = NO).
The drag forces must then be defined in the **DRAGFO** user subroutine. 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 is set to NO by default). In such case, the value of the Coriolis coefficient (see Formulation Document) is defined by the keyword `CORIOLIS COEFFICIENT` (default value = 0.). This must be calculated according to the latitude $\lambda$ through the formula:
FCOR = 2 $\omega$ sin($\lambda$) where $\omega$ is the angular velocity of the Earth, equal to 7.2921 $\times$ $10^{-5}$ rad/s.
The components of the Coriolis force are thus:
FU = FCOR $\times$ V and FV = -FCOR $\times$ 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 by activating the keyword `SPHERICAL COORDINATES` (see 14.3). Its default value is NO.
TELEMAC-2D also offers the opportunity of defining the water density with the keyword `WATER DENSITY`. Its default value is 1,000 kg/m$^3$, i.e. a value corresponding to a fresh river water.
Gravity acceleration can be changed with the keyword `GRAVITY ACCELERATION` whose default value is set at 9.81 m/s$^2$.

## 6.8 Tsunami generation

TELEMAC-2D can model tsunami generation by computing the free surface displacement according to Okada model (1992), assuming it is similar to that of the seabed. It is treated as an initial condition on water depth, depending on the location given by the keywords `LONGITUDE OF ORIGIN POINT` and `LATITUDE OF ORIGIN POINT`. Tsunami generation can be activated with the keyword `OPTION FOR TSUNAMI GENERATION`:

- 1: no tsunami (default value),
- 2: tsunami generated on the basis of the Okada model (1992).

The main physical characteristics of the tsunami can be set by the keyword `PHYSICAL CHARACTERISTICS OF THE TSUNAMI` which is an array of 10 values (in the order):

- $HH$ focal depth (in m),
- $L$ fault length (in m),
- $W$ fault width (in m),
- $D$ dislocation (in m),
- $TH$ strike direction (in decimal degrees),
- $DL$ dip angle (in decimal degrees),
- $RD$ slip angle (in decimal degrees),
- $Y0$ epicentre latitude (in decimal degrees),
- $X0$ epicentre longitude (in decimal degrees),
- $C0$ size of the ellipse of influence ($L \times W$).

Default values for this keyword are: (100.;210000.;75000.;13.6;81.;41.;110.;0.;0.;3.).

## 6.9 Parameter estimation

TELEMAC-2D contains a function for automatically determining an unknown physical parameter. In the current release 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 current release of the software, it is strongly recommended to work only in permanent mode.
Measurement data are supplied via the **USER_MESURES** user subroutine which contains the arguments **ITER** (iteration number) and **TT** (absolute time). The latter argument is used in processing real measurements. Each time the **USER_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 **USER_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` (default value = NO) and to complete the **USER_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 celerities and velocities). 2 seems to be preferable, even though the effect of this parameter is slight. Anyway, `COST FUNCTION` must be 1 with tidal flats.
The integer keyword `IDENTIFICATION METHOD` is used to specify the technique used for minimizing 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 should 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 re-used 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 (or Shallow Water quations) using the finite-element method. This is the "traditional" use of TELEMAC-2D.
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 Boussinesq equations are solved.
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),
- quadratic triangle (6 nodes triangle).

The choice is done with the keyword DISCRETIZATIONS IN SPACE. This keyword is an array of five integers that are related successively to the velocity, depth, possible tracer(s), $k/\varepsilon$ or $\tilde{\nu}$ variables. 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. By default, the value 11 is set for the four variables. If only setting the first ones, the others are set to the default value 11.
In practice, the user can select the 3 following combinations (example for the first two variables velocities and water depth):

- 11 ; 11 (default value) : linear velocity and linear water depth (recommended),
- 12 ; 11 : quasi-bubble velocity and linear water depth,
- 13 ; 11 : quadratic velocity and linear water 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, yet every option of computations with TELEMAC-2D is not available). The second one is recommended when observing free surface wiggles (in particular in case of strong bathymetry gradients). But in that situation, the recommended configuration is to use the wave equation associated with the keyword FREE SURFACE GRADIENT COMPATIBILITY = 0.9. The user can also decrease that value down to 0. if needed.

Wave equation (TREATMENT OF THE LINEAR SYSTEM = 2) has not been implemented for quadratic elements (13).
Weak characteristics and distributive schemes have not been implemented for quasi-bubble and quadratic elements (12 or 13) either.

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 some 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 set 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 ($k-\varepsilon$ model) or the advection of $\tilde{\nu}$ (Spalart-Allmaras model),
- ADVECTION OF TRACERS: for the advection of tracer(s).

The default value of these four 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 can be linearized by activating the keyword LINEARIZED PROPAGATION (default value = NO), in particular when running a test case for which an analytical solution is available in the linearized case. It is then necessary to set the water depth around which the linearization is to be performed, by using the keyword MEAN DEPTH FOR LINEARIZATION (default value = 0.).

## 7.2 Numerical schemes

### 7.2.1 Finite elements

Finite elements resolution is based on the primitive equations. It is possible to replace the original equations by a generalized wave equation obtained by eliminating the velocity from the continuity equation using a value obtained from the momentum equation. This technique decreases calculation time but has the disadvantage of smoothing the results. The choice between these two options is done using the keyword TREATMENT OF THE LINEAR SYSTEM:

- 1: original equations (= system of coupled equations, which was the old default value until release 8.1),
- 2: wave equation (new default value since release 8.2).

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 of stability and CPU time.
Another choice concerns the scheme used for solving the advection step. To do this, the user has to update the keyword TYPE OF ADVECTION. This keyword is an array of four integers that are related successively to the scheme used for advection of the velocity ($U$ and $V$), depth ($H$), tracer and turbulent values ($k$ and $\varepsilon$, or $\tilde{\nu}$). If the model does not include any tracer or turbulence model, the user may simply give the first two values.
Since release 6.0, the value concerning depth is ignored by TELEMAC-2D. The optimum numerical scheme is automatically selected by the code (conservative scheme).
Alternatively to TYPE OF ADVECTION, the keyword SCHEME FOR ADVECTION OF VELOCITIES can be used.
Each integer may have a value between 1 and 15, corresponding to the following possibilities:

- 1: Method of characteristics, not mass-conservative,
- 2: Centred semi implicit scheme + SUPG (Streamline Upwind Petrov Galerkin),
- 3: Upwind explicit finite volume (referenced as 8 before release 6.0), mass-conservative,
- 4: N distributive scheme, mass-conservative,
- 5: PSI distributive scheme, mass-conservative,
- 13: Edge by edge implementation of scheme 3 (will work on tidal flats), mass-conservative,
- 14: Edge by edge implementation of scheme 4 (will work on tidal flats), mass-conservative,
- 15: ERIA scheme (will work on tidal flats), mass-conservative.

Schemes 3 and 4 on the one hand, and 13 and 14 on the other hand, are equal in 2D (they are not in 3D) and correspond to the so called NERD scheme.
The stability of the N and the PSI scheme (type of advection 4 and 5) is conditioned by a Courant number lower than 1. When using these schemes, TELEMAC-2D checks 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 200, TELEMAC-2D will consider that solving the advection term is no longer possible and the computation is stopped with an appropriate error message printed in the output listing.
The distributive schemes N and PSI have been improved from release 7.0 to deal with time dependent problems. Several options are offered to the user through the keyword SCHEME OPTION FOR ADVECTION OF VELOCITIES, which can be set to:

- 1: explicit scheme (default value),
- 2: first order predictor-corrector scheme,
- 3: second order predictor-corrector scheme,
- 4: locally semi-implicit predictor-corrector scheme (for tidal flats): LIPS.

In addition, the predictor-corrector schemes need an additional parameter which represents the number of iterations for every time step (or sub-time step) to converge to the solution. The keyword NUMBER OF CORRECTIONS OF DISTRIBUTIVE SCHEMES plays this role and it is useful for unsteady cases. For quasi-steady flows, the NUMBER OF CORRECTIONS OF DISTRIBUTIVE SCHEMES does not have a large impact on the solution, so it can be left to its default value. On the other hand, for unsteady flows, it is suggested to set the keyword NUMBER OF CORRECTIONS OF DISTRIBUTIVE SCHEMES to 2 (at least), which is a good compromise between accuracy and computational time. Indeed, increasing the number of corrections the scheme is more accurate but the CPU time rapidly increases. The keyword NUMBER OF CORRECTIONS OF DISTRIBUTIVE SCHEMES can be used with SCHEME FOR ADVECTION OF... = 3, 4, 5 or 15 AND SCHEME OPTION FOR ADVECTION OF... = 2, 3 or 4.
The keyword MAXIMUM NUMBER OF ITERATIONS FOR ADVECTION SCHEMES enables to limit the number of solver iterations for the advection schemes of type NERD or ERIA (SCHEME FOR ADVECTION OF... = 13, 14 or 15). The default value is 50 (old default value = 10 until release 8.1).
The keyword NUMBER OF SUB-STEPS OF DISTRIBUTIVE SCHEMES can only be activated for locally semi-implicit predictor-corrector schemes aka LIPS (SCHEME FOR ADVECTION OF... = 3, 4 or 5 + SCHEME OPTION FOR ADVECTION OF... = 4). The default value is 1. As the keyword mentions, it allows to subdivide the time step given by the user in the steering file, into several sub-steps. Again, it produces an effect on the precision of the scheme and it is convenient to set this keyword in order to have Courant numbers not too large (around 1).

Note:

- If present, the keyword SCHEME OPTION FOR ADVECTION OF VELOCITIES replaces and has priority over the following keywords: OPTION FOR CHARACTERISTICS and SUPG OPTION.
- The same remark are valid for advection of tracer, $k$, $\varepsilon$ and $\tilde{\nu}$. However there are dedicated keywords: SCHEME FOR ADVECTION OF TRACERS and SCHEME FOR ADVECTION OF K-EPSILON (see sections 9.5 and 6.2),
- MATRIX STORAGE = 3 is mandatory with a distributive scheme for advection (= 3, 4, 5, 13, 14 or 15). In addition, SUPG OPTION for water depth = 0. (i.e. no upwinding) is mandatory for a distributive scheme with tidal flats (13, 14 or 15).

The default value for TYPE OF ADVECTION 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 release 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. For dam break studies option 14;5 is recommended.
The keyword TYPE OF ADVECTION will be soon replaced by the following keywords (already active):

- SCHEME FOR ADVECTION OF VELOCITIES (default = 1),

- SCHEME FOR ADVECTION OF TRACERS (default = 1),
- SCHEME FOR ADVECTION OF K-EPSILON (default = 1).

The keyword for advection of water depth is not necessary, since the scheme is automatically selected by TELEMAC-2D, as already said here above.
Depending on the scheme used, accuracy 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. The number of sub-iterations is set by the keyword NUMBER OF SUB-ITERATIONS FOR NON-LINEARITIES, which has a default value of 1. This option is time consuming but it can be helpful for some studies like dam-break studies.
In TELEMAC-2D time discretization is semi-implicit. The various implicitation coefficients are given with the keywords IMPLICITATION FOR DEPTH (corresponding to the **TETAC** FORTRAN variable, default = 0.55), IMPLICITATION FOR VELOCITY (corresponding to the **TETAU** FORTRAN variable, default = 0.55), IMPLICITATION FOR DIFFUSION OF VELOCITY (corresponding to the **TETAD** FORTRAN variable, default = 1.), and in the case of computing tracer transport, IMPLICITATION COEFFICIENT OF TRACERS (corresponding to the **TETAT** FORTRAN variable, default = 0.6). The default values are generally adequate.
The reader's attention is drawn to the fact that in earlier releases of TELEMAC-2D and under certain conditions, the value of some parameters could be arbitrarily set in the code regardless of the specified keywords value.
When solving the linearized system $A.X = B$, TELEMAC-2D offers the possibility of mass-lumping on the mass matrices ($M^h$ for depth, $M^u$ and $M^v$ for velocities) involved in computing the matrices $AM1$ for depth, and $AM2$ and $AM3$ for velocity (see [18] for more details). 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 more smoothed. The mass-lumping rate is set with the keywords MASS-LUMPING ON H, MASS-LUMPING ON VELOCITY and MASS-LUMPING FOR WEAK CHARACTERISTICS. 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.
As the mass-lumping is applied only on time derivatives, it does not change steady state results.

If using OPTION FOR THE TREATMENT OF TIDAL FLATS = 1 (default value) and TREATMENT OF NEGATIVE DEPTHS = 2 (flux control), the keyword MASS-LUMPING ON H must be equal to 1.
The keyword MASS-LUMPING ON TRACERS is read but automatically replaced by the value of MASS-LUMPING ON H to ensure tracer mass conservation.
As told at the beginning of this subsection, if using TREATMENT OF THE LINEAR SYSTEM = 2 (wave equation) automatically changes MASS-LUMPING ON VELOCITY to value 1.

#### Configuration of the SUPG scheme

When using the SUPG method, the user has to determine the type of upwinding desired with the keyword SUPG OPTION which, like the keyword TYPE OF ADVECTION, is an array of 4 integers related to the velocities, the water depth, tracer(s) and $k-\varepsilon$ model (or Spalart-Allmaras model) respectively. Default value = (2;2;2;2).
The possible values are the following:

- 0: No upwind scheme,

- 1: Upwind scheme with the classical SUPG method, i.e. upwind scheme = 1,
- 2: Upwind scheme with the modified SUPG method, i.e. upwinding equal to the Courant number.

In theory, 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 should be used only for models in which the Courant number is very low. If the Courant number cannot 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` but the second number for `SUPG OPTION` applies to the depth (even if the SUPG method cannot be selected for the advection of water depth). If it is 1 or 2, the corresponding extra term that SUPG would bring to the advection of the depth, i.e. the part due to upwind, is added to the continuity equation. This "advection of the depth" appears when the term $div(h\boldsymbol{u})$ is split into $hdiv(\boldsymbol{u}) + \boldsymbol{u}grad(h)$. This SUPG treatment is mathematically not far from adding a diffusion or from smoothing the depth and it has a powerful effect on stability, e.g. in cases with hydraulic jumps. However this numerical trick cannot be used with tracers, since the presence of the extra term in the continuity equation breaks the tracer mass conservation (it has no effect on the water mass conservation).

#### Configuration of the weak characteristics

When choosing the method of characteristics, two forms can be used with the keyword `OPTION FOR CHARACTERISTICS`:

- 1: the strong form (by default),
- 2: the weak form.

If one component of array `TYPE OF ADVECTION` = 1 or `SCHEME FOR ADVECTION OF...` = 1, and also the corresponding keyword `SCHEME OPTION FOR ADVECTION OF...` = 2, `OPTION FOR CHARACTERISTICS` is automatically set to 2.

None of the two choices for `OPTION FOR CHARACTERISTICS` are recommended for the advection of tracers because they are not mass conservative. The weak form will decrease the diffusion. If the keyword `MASS-LUMPING FOR WEAK CHARACTERISTICS` = 1. (default value = 0. i.e. no mass-lumping), monotonicity of the scheme appears. This weak form should be more conservative than the strong form. The `NUMBER OF GAUSS POINTS FOR WEAK CHARACTERISTICS` defines the number of Gauss points used to compute the weak characteristics. Possible choices for TELEMAC-2D are:

- 1 point,
- 3 points (default value),
- 4 points,
- 6 points,
- 7 points,
- 12 points.

The bigger the number is, the more conservative the scheme is, but the higher the computational costs are.

### 7.2.2 Finite volumes

When using the finite volume scheme, the primitive equations written in conservative form are solved. The finite volume schemes use control volumes which are centered around the triangular mesh nodes and are also referred to as the dual mesh cells. The dual mesh is defined from the barycentre of the primal mesh triangles (cf. Figure 7.1).

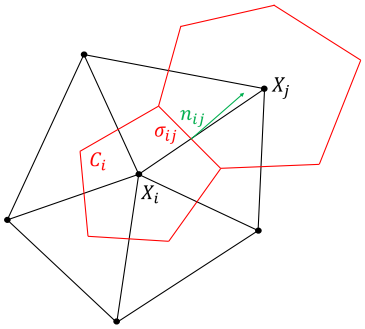


Figure 7.1: Triangular mesh (black) and finite volume dual mesh (red)

#### Numerical schemes for the hyperbolic part

The keyword `FINITE VOLUME SCHEME` specifies the type of scheme used to solve the hyperbolic part of the shallow water system. The available possibilities are:

- 0: Roe scheme. See [30] for more details.
- 1: Kinetic scheme (default value). See [5].
- 3: Zokagoa scheme (implementation not compatible with tidal flats). See [45].
- 4: Tchamen scheme,
- 5: HLLC (Harten Lax Leer-Contact) scheme, which is one the most frequently used scheme in the literature [33].
- 6: WAF (Weighted Average Flux) scheme (parallel not implemented). For more details about this scheme, see [2].

The algorithm of finite volume schemes is explicit which means that the Courant number must be inferior or equal to 1 to ensure stability. It is however recommended to set the keyword `DESIRED COURANT NUMBER` to 0.9. This should be combined with the keyword `VARIABLE TIME-STEP` set to YES (default: NO). TELEMAC-2D then adjusts the calculation time step so as to satisfy this Courant number criterion. The graphic printout is handled with the time step given in the steering file. However, it should be noted that this leads to irregular sampling from the control listing.
The keyword `OPTION OF THE HYDROSTATIC RECONSTRUCTION` enables to choose the option for hydrostatic reconstruction for the Kinetic, HLLC and WAF schemes. The 2 possible choices are:

- 1: Audusse *et al.* [4] (default),
- 2: Chen and Noelle improvement [11], which is useful for rain induced runoff (work only with first order in space schemes).

The keyword FINITE VOLUME SCHEME SPACE ORDER (default = 1) allows the use of second order reconstructions which gives second order accuracy for spatial schemes. The 2 possible choices are:

- 1: First order (default value),
- 2: Second order MUSCL scheme [3] (only compatible with Kinetic and HLLC schemes).

When second order MUSCL scheme is used, the keywords FLUX LIMITOR FOR H PLUS Z, FLUX LIMITOR FOR U AND V and FLUX LIMITOR FOR TRACERS allow the user to switch between several second order TVD flux limitors for the reconstructions of free surface, velocities and tracers. The available choices are:

- 1: Minmod (default value for H+Z),
- 2: Van Albala (default value for U, V and tracers),
- 3: MC (Monotonized Central)
- 4: Generalized minmod

Similarly the keyword FINITE VOLUME SCHEME TIME ORDER (default = 1) gives the possibility to use second order accurate temporal scheme. The 2 possible choices are:

- 1: Euler explicit scheme (default value),
- 2: Newmark scheme.

The keyword NEWMARK TIME INTEGRATION COEFFICIENT (default = 0.5) gives the possibility to change the Newmark scheme integration parameter. When set to 1 the Newmark scheme is equivalent to the Euler explicit time scheme.

#### Numerical schemes for the parabolic part

To solve the parabolic part of the shallow water system i.e. the diffusion of velocities and tracers the keywords FINITE VOLUME SCHEME FOR VELOCITY DIFFUSION as well as FINITE VOLUME SCHEME FOR TRACER DIFFUSION can be used to specify the type of schemes used. For the tracer diffusion scheme choice, one value by tracer must be set. The available possibilities are:

- 1: Explicit P1 finite element scheme (default value),
- 2: TPF (Two Points Flux) finite volume scheme,
- 3: RTPF (Reconstructed Two Points Flux) finite volume scheme.

The first scheme is a finite element P1 explicit scheme with mass lumping which does not require the resolution of a linear system. The scheme gives a second order accuracy for the parabolic part and has specifically been implemented to work alongside finite volume hyperbolic schemes.
The second option corresponds to the finite volume two point flux scheme (TPF) which is not consistent in the case of the TELEMAC-2D dual mesh. It is available for comparison purposes but should be used carefully. Instead of the classic two point flux scheme, the third option is advised. The finite volume reconstructed two point flux scheme (RTPF) consists in reconstructing the fields to ensure consistency of the diffusion flux.

Two options are available for the reconstructions of the RTPF scheme, which can be selected via the keyword OPTION FOR THE RTPF SCHEME RECONSTRUCTIONS. The first option (default value) uses a simple linear reconstruction based on gradients on adjacent triangles of the interfaces which gives a first order accuracy for the parabolic part. The second option uses dual mesh cell gradients.
For each scheme, both Neumann and Dirichlet boundary conditions are implemented. For the Dirichlet boundary condition, it is possible to choose the type of imposition (weak or strong) with the keyword OPTION FOR DIRICHLET CONDITION IN FV DIFFUSION. The available possibilities are:

- 1: weak (default value),
- 2: strong.

The numerical schemes implemented for the diffusion being all explicit, the resolution requires to fullfill a condition on the time step, namely the Fourier condition (similar to the CFL condition). As a consequence, the Fourier number must be set via the keyword DESIRED FOURIER NUMBER to a value inferior or equal to 1 to ensure stability. This should be combined with the keyword VARIABLE TIME-STEP set to YES (default: NO).

## 7.3 Solving the linear system

This section only concerns Finite Elements method.

### 7.3.1 Solver

During some steps, the solver used for solving the systems of equations can be selected through the following keywords:

- SOLVER: for the hydrodynamic propagation step (default value = 3),
- SOLVER FOR DIFFUSION OF TRACERS: for the tracers diffusion step (default value = 1),
- SOLVER FOR K-EPSILON MODEL: for solving the $k-\varepsilon$ model system or Spalart-Allmaras model system (default value = 1).

Each keyword may have a value between 1 and 8. These values correspond to the following possibilities. Options 1 to 6 are all related to the conjugate gradient method:

- 1: Conjugate gradient method (when the matrix of the system to solve is symmetric),
- 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,
- 8: Direct solver (YSMP, solver of the Yale university), does not work in parallel mode.

The GMRES method is well suited for improperly conditioned systems. If the GMRES method is used, the dimension of the Krylov space has to be specified with the appropriate keyword, i.e.:

- SOLVER OPTION: for hydrodynamic propagation,
- SOLVER OPTION FOR TRACERS DIFFUSION: for tracer(s) diffusion,
- OPTION FOR THE SOLVER FOR K-EPSILON MODEL: for the $k-\varepsilon$ model or Spalart-Allmaras model.

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 $k-\varepsilon$ model or the Spalart-Allmaras model. If the wave equation is used (TREATMENT OF THE LINEAR SYSTEM = 2), SOLVER = 1 is recommended.
The GMRES method with a Krylov space dimension equal to 2 or 3 seems to fit most cases in 2D, 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.

### 7.3.2 Accuracy

When the linearized system is solved by an iterative method, it is necessary to give 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 K: defines the accuracy required during the diffusion and source terms step of the turbulent energy transport equation (default value = $10^{-9}$),
- ACCURACY OF EPSILON: defines the accuracy required during the computation of diffusion and source terms step of the turbulent dissipation transport equation (default value = $10^{-9}$),
- ACCURACY OF SPALART-ALLMARAS: defines the accuracy required during the diffusion and source terms step of the Spalart-Allmaras equation (default value = $10^{-9}$).

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

- MAXIMUM NUMBER OF ITERATIONS FOR SOLVER: defines the maximum permissible number of iterations when solving the propagation step (default value = 100),
- 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 K AND EPSILON: defines the maximum permissible number of iterations when solving the diffusion and source terms step of the $k-\varepsilon$ transport equations or the Spalart-Allmaras equation (default value = 50).

The user may obtain information on the solvers by activating the keywords INFORMATION ABOUT SOLVER and, if the $k-\varepsilon$ turbulence model is used, INFORMATION ABOUT K-EPSILON MODEL. The same keyword exists for the Spalart-Allmaras turbulence model (INFORMATION ABOUT SPALART-ALLMARAS MODEL). The default value of the 3 previous keywords is YES. 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 some cases, and if the number of iterations is already set 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.

### 7.3.3 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 1,000. The default value is NO.
When using TREATMENT OF NEGATIVE DEPTH = 2 or 3 with tidal flats, it is mandatory to activate CONTINUITY CORRECTION = YES.

### 7.3.4 Preconditioning

When solving a system of equations by an iterative linear solver, convergence can often be accelerated 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 last one, common to the $k-\varepsilon$ and Spalart-Allmaras models).
The possibilities may be different depending on the keywords.
The keyword PRECONDITIONING concerns the propagation solution step, and can have 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 (does not work in parallel),

- 11: Gauss-Seidel EBE preconditioning (not convenient for parallelism),
- 13: preconditioning supplied by the user.

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

- 0: No preconditioning,
- 2: Diagonal preconditioning (default value),
- 5: Diagonal preconditioning with absolute value,
- 7: Crout preconditioning per element (does not work in parallel),
- 11: Gauss-Seidel EBE preconditioning (not convenient for parallelism),
- 13: preconditioning supplied by the user.

The keyword PRECONDITIONING FOR K-EPSILON MODEL concerns the turbulence model solution step (for both $k-\varepsilon$ and Spalart-Allmaras models), 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 (does not work in parallel),
- 11: Gauss-Seidel EBE preconditioning (not convenient for parallelism),
- 13: preconditioning supplied by the user.

Some options of preconditioning can be cumulated, namely the diagonal ones with the others. As the base values are prime numbers, 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 only be used 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 water depth $H$ 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 $DH = H_{n+1} - H_n$ null,
- 1: Initial value of $DH$ equal to the value of $DH$ at the previous time step (default value),
- 2: $DH = 2DH_n - DH_{n-1}$ in which $DH_n$ is the value of $DH$ at the previous time step and $DH_{n-1}$ 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 $V$).

### 7.3.5 C-U preconditioning

When solving the linear system, $C-U$ preconditioning consists in replacing the unknown depth by the celerity. This technique was used automatically in the old releases 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 (TREATMENT OF THE LINEAR SYSTEM = 2).

## 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-STEP (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

This section mainly concerns Finite Elements schemes, indeed for Finite Volumes (EQUATIONS = 'SAINT-VENANT FV') no specific treatment for tidal flat is required. All the options cited hereafter are then useless.

TELEMAC-2D offers several processing options for tidal flat areas when Finite Elements schemes are used.
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.
To model tidal flats, the main keywords involved are:

- OPTION FOR THE TREATMENT OF TIDAL FLATS: three different ways can be chosen to process tidal flats,
- SCHEME FOR ADVECTION OF VELOCITIES (OF TRACERS or OF K-EPSILON): only a few schemes are suitable for tidal flats,
- TREATMENT OF NEGATIVE DEPTHS: three different options are available.

Tidal flats can be processed in three different ways setting the keyword OPTION FOR THE TREATMENT OF TIDAL FLATS:

- 1: The tidal flats are detected and the free surface gradient is corrected,
- 2: 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,

- 3: 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 whole domain but to the depth integral multiplied by the porosity. The user can modify the porosity value determined by the processing in the **USER_CORPOR** subroutine.

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 in smoothing the negative depths in a conservative way. The option 2 (since release 6.0) is a flux limitation that ensures strictly positive depths. So is option 3 for ERIA advection scheme. This must be preferably coupled with the advection schemes able to cope with tidal flats (+ MASS-LUMPING ON H = 1. + CONTINUITY CORRECTION = YES + SUPG OPTION for water depth = 0, i.e. no SUPG upwinding on depth). This option is however recommended when conservative tracers are modelled using distributive schemes (SCHEME FOR ADVECTION OF TRACERS = 4 or 5): it allows to obtain a perfect mass balance. Value 0 means no treatment.
The numerical advection schemes (keywords TYPE OF ADVECTION or SCHEME FOR ADVECTION OF...) for tidal flats are:

- 13 or 14: NERD scheme,
- 4 or 5 coupled with OPTION FOR ADVECTION OF... = 4: LIPS scheme,
- 15: ERIA scheme.

Note:
NERD schemes (13 and 14) require the keywords TIDAL FLATS = YES (default) + OPTION FOR THE TREATMENT OF TIDAL FLATS = 1 (default) + TREATMENT OF NEGATIVE DEPTHS = 2 and then MASS-LUMPING ON H = 1. + CONTINUITY CORRECTION = YES + SUPG OPTION for water depth = 0.
ERIA scheme (= 15) requires the keywords TIDAL FLATS = YES (default) + OPTION FOR THE TREATMENT OF TIDAL FLATS = 1 (default) + TREATMENT OF NEGATIVE DEPTHS = 3 and then MASS-LUMPING ON H = 1. + CONTINUITY CORRECTION = YES + SUPG OPTION for water depth = 0.
LIPS requires the keywords TIDAL FLATS = YES (default) + OPTION FOR THE TREATMENT OF TIDAL FLATS = 1 (default) and allows the two types of TREATMENT OF NEGATIVE DEPTHS = 2 or 3. It then also needs: MASS-LUMPING ON H = 1. + CONTINUITY CORRECTION = YES + SUPG OPTION for water depth = 0.
Note also that NERD and ERIA cannot be used simultaneously.

The keyword THRESHOLD FOR NEGATIVE DEPTHS (default = 0.) is only used with TREATMENT OF NEGATIVE DEPTHS = 1. 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 some 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 set the threshold below which clipping is performed.

However, it should be born in mind that this latter option may lead 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: an Element by Element (EBE) method and an edge-based method. The second is faster (about 20%) in most cases.
The choice between the two storage methods can be done using the keyword `MATRIX STORAGE`, with the following values:

- 1: classical Element by Element (EBE) method,
- 3: edge-based storage method (default and recommended value).

`MATRIX STORAGE` = 3 is mandatory with a distributive scheme for advection (= 3, 4, 5, 13, 14 or 15). It is also mandatory when using a direct system solver (`SOLVER...` = 8).

### 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 more recent 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.

When using the frontal matrix-vector product, the number of neighbors of the points in the mesh is limited to 10 so far.
Option 2 is not implemented in parallel and is then automatically changed to 1 (with a warning message). Moreover, option 2 is not implemented for quasi-bubble elements.

### 7.6.3 Finite Element assembly mode in parallel

When assembling Finite Elements in parallel, TELEMAC-2D has several options to do it since some works on reproducibility for TELEMAC (see Rafife Nheili's PhD thesis). It can be done directly with double precision values (option 1 = default value) or with integers to avoid truncation errors in parallel, due to different order of additions of more than 2 numbers. The choice can be done with the keyword `FINITE ELEMENT ASSEMBLY` which can get the following values:

- 1: normal (default value),
- 2: with I8 integers,
- 3: compensation (for reproducibility).

Caution: only some parts of TELEMAC-2D have been implemented with `FINITE ELEMENT ASSEMBLY` different from 1 (e.g. propagation step) and the whole code is not fully available in compensated mode.

## 7.7 Convergence study

To assess the accuracy of numerical schemes, the computation of errors on several meshes is needed. TELEMAC-2D offers the possibility to carry out a mesh convergence by activating the keyword CONVERGENCE STUDY (default = NO). The resulting solutions are then compared to an analytical solution on a fine mesh.

The user has to give the number of refinement levels with the keyword REFINEMENT LEVELS (default = 0). Each level multiplies the number of triangular elements by 4 with STBTEL.
This option is only available in sequential mode (PARALLEL PROCESSORS = 0 or 1, or Python option --ncsize = 0 or 1).

# 8. Managing water sources

TELEMAC-2D offers the possibility of placing water sources (with or without tracer discharge) at any point of the domain.
The user has a few options to place the various sources at differents points of the domain:

- with the keywords ABSCISSAE OF SOURCES and ORDINATES OF SOURCES. These are arrays of real numbers, giving the source coordinates in meters. Actually, TELEMAC-2D will position a source at the closest mesh point to that specified by these keywords. In parallel mode, the sources must coincide exactly with one point of the mesh, so this is recommended in all cases,
- with the keyword GLOBAL NUMBERS OF SOURCE NODES. This is an array of integers which contains the global numbers of nodes that correspond to source point locations.

The program itself will determine the number of sources as a function of the number of values given to these keywords.
Another option consists in associating the source(s) to a spatial area(s) of the domain and not to a single point. To use this option the user must create an external file which contains the coordinates of every source region (polygons). In this case the source will be placed at every point contained in the region (the region must contain at least one node). The advantage of this option is that the user can control the surface of the source. However, it is worth to notice that the surface computed by the program will not correspond exactly to the surface of the polygon indicated by the user. Indeed, the surface will be computed as the sum of the surfaces of every point (integral of basis functions). An example of file is given here: it indicates three source regions, defined through 4 pair of coordinates.

```
#
# COORDINATES AT SOURCE REGION 1
#
X(1)   Y(1)
198.7  25.85
198.7  24.06
201.5  24.06
201.5  25.85
#
# COORDINATES AT SOURCE REGION 2
#
X(2)   Y(2)
```

```
248.7  27.85
248.7  22.06
251.5  22.06
251.5  27.85
#
# COORDINATES AT SOURCE REGION 3
#
X(3)  Y(3)
372.0 8.0
372.0 2.0
376.0 2.0
376.0 8.0
```

The program itself will determine the number of points contained in each region. The keyword to read the file is SOURCE REGIONS DATA FILE. The maximum number of regions is limited by the keyword MAXIMUM NUMBER OF SOURCES, which is 20 by default, while the number of coordinates for every polygons is limited by the keyword MAXIMUM NUMBER OF POINTS FOR SOURCES REGIONS, which is 10 by default.
At each source (single point or region), the user must indicate the discharge (and the values of the tracers, if there are tracers). The discharge is specified in $m^3/s$ using the keyword WATER 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 subroutines **USER_DEBSCE** (source discharge) and **USER_TRSCE** (value of tracer at source). It is also possible to use a specific file to define the time evolution of the sources: the 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(1,1) IS TRACER 1 AT SOURCE 1 TR(1,2) IS TRACER 2 AT SOURCE 1 TR(2,1) IS
#  TRACER 1 AT SOURCE 2 TR(2,1) IS TRACER 2 AT SOURCE 2
#
#
T     Q(1)   TR(1,1)    TR(1,2)   Q(2)   TR(2,1)   TR(2,2)
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.
```

By default, when using the keyword WATER DISCHARGE OF SOURCES, the sources are added in the continuity equations without contribution to the momentum equations. In this case we consider that the velocity of sources is equal to the velocity of the flow.
To take into account a momentum flux from the sources with TELEMAC-2D, the user must prescribe a particular velocity, which will be considered in the momentum equations. 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 subroutines **USER_VUSCE** (for the velocity along $x$) and **USER_VVSCE** (for the velocity along $y$). In both subroutines, time, source number and water depth are available to the user.
When the sources are located through the keywords ABSCISSAE OF SOURCES or GLOBAL NUMBERS OF SOURCE NODES, the WATER DISCHARGE OF SOURCES is mandatory to

take into account the velocity of the sources. On the contrary, if the source is located in a region (through ASCII SOURCE DATA FILE), the user can choose to only impose the velocity of the source. The program will automatically compute the water discharge, using the surface of the region.
Although it is possible in a practical point of view, it is not recommended to use source point at the boundaries of the domain. In these cases, the velocity field could not be as expected by the user even though **USER_VUSCE** and **USER_VVSCE** subroutines are used. The imposition of hydrodynamics boundary conditions could modify the prescribed components of the velocity of the source.
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 SOURCES, 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.

# 9. Tracer transport

The TELEMAC-2D software makes it possible to take into account the transport of a number of passive (non-buoyant) tracers (tracers has no effect on the hydrodynamics), being either diffused or not.

The maximum number of tracers is set to 20 by default but it can be changed by the user with the keyword MAXIMUM NUMBER OF TRACERS.

## 9.1 General setup

The tracer transport computation is activated with the keyword NUMBER OF TRACERS (default value = 0, i.e. no tracer) which gives the number of tracers taken into account during the simulation. Additional modules which can be coupled with TELEMAC-2D (e.g. WAQTEL or GAIA, see section 14.8) can add extra tracers to the initial set defined in the TELEMAC-2D steering file.

In addition, it is possible to give the name and the unit of each tracer. This information is given by the keyword NAMES OF TRACERS. The names are given in 32 characters (16 for the name itself and 16 for the unit). For example, for 2 tracers:

```
NUMBER OF TRACERS = 2
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 followed by the number of tracer. For example 'T1,T3' stand for first and third tracer. It is possible to use the character * as wildcards (replace any character). T* stands for T1 to T9, and T** stands for T10 to T99.

N.B.: TELEMAC-2D offers the possibility of taking into account density effects when the 1st tracer used is the salinity expressed in kg/m$^3$. In this case, it is necessary to set 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 and $\rho_{\text{water}} = 999.972.(1 - 7.10^{-6}(T_{\text{mean}} - 4)^2)$.

## 9.2 Prescribing initial conditions

If the initial values of tracers are constant all over the domain, just insert, into the steering file, the keyword INITIAL VALUES OF TRACERS with the required value(s) separated with a semicolon; if more than one. 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 **USER_CONDIN_TRAC** subroutine, in a similar way to that described in the section dealing with the initial hydrodynamic conditions.

If a computation is being continued, the initial condition of the tracers corresponds to that of the last time step stored in the restart file. If the restart file does not contain any information about the tracer, TELEMAC-2D will use the value assigned to the keyword INITIAL VALUES OF TRACERS).

## 9.3 Prescribing boundary conditions

Tracer boundary conditions are prescribed in the same way as hydrodynamic conditions.

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

In case of an inflowing open boundary with one or several prescribed tracer(s) (**LITBOR** = 5), the tracer value can 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 an array of real numbers for managing several boundaries and several tracers (100 at most, this number can be changed by changing the keyword MAXIMUM NUMBER OF TRACERS). The writing convention 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, second tracer at the first open boundary..., first tracer at the second open boundary, second tracer at second open boundary, etc.,

- If the value is constant in time but varies along the boundary, it will be set directly by the **TBOR** variable in 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 with the LIQUID BOUNDARIES FILE. Programming is done in the same way as for the functions **USER_VIT**, **USER_Q** and **USER_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).

The keyword TREATMENT OF FLUXES AT THE BOUNDARIES enables, during the convection step (with the SUPG, PSI and N schemes), to set a priority among the tracer flux across the boundary and tracer value at that wall. Option 2 ("Priority to fluxes") will then induce a change in the tracer prescribed value, so that the flux is correct. On the other hand, option 1 ("Priority to prescribed values", default value) sets the tracer value without checking the fluxes. Contrary to what is offered in TELEMAC-3D, the TELEMAC-2D keyword has only one value, which is then applied to all liquid boundaries.

## 9.4 Managing tracer sources

TELEMAC-2D offers the possibility of placing tracer sources (with or without tracer discharge) at any point of the domain. The management of these sources is identical the one of all other type of sources. See chapter 8 for more details.

## 9.5 Numerical specifications

The way of treating advection od tracers is specified in the third value of the keyword `TYPE OF ADVECTION` or directly with the keyword `SCHEME FOR ADVECTION OF TRACERS` (which has priority to `TYPE OF ADVECTION`, if present in the `STEERING FILE`). 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. If an advection scheme for tracers is a distributive scheme (e.g.: 3, N = 4, PSI = 5, NERD = 13, 14 or ERIA = 15), `IMPLICITATION COEFFICIENT OF TRACERS` is prescribed at 0. (explicit).
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 could be specified using the keyword `COEFFICIENT FOR DIFFUSION OF TRACERS` which is an array since release 8.2 (one value per tracer separated by a semicolon). It used to be a single value for every tracer until release 8.1. The default values are equal to $10^{-6}$ m$^2$/s for every tracer. This parameter has a very important influence on tracer diffusion in time. As for velocity diffusion, a time- or space-variable tracer diffusion coefficient could be programmed directly in the **CORVIS** subroutine.
As for velocity diffusion (see 6.2), 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).

## 9.6 Law of tracer degradation

By default, TELEMAC-2D tracers are considered as mass-conservative. However, it is possible to specify a degradation law by coupling TELEMAC-2D with the WAQTEL water quality module (see 14.8). Please refer to the WAQTEL documentation for more informations.

# 10. Secondary currents

In a curved channel, the flow experiences a radial acceleration and centrifugal forces act in proportion to the mean velocity. In turn, the surface of the water is tilted radially on the outer bank to produce a super-elevation sufficient to create a pressure gradient to balance the average centrifugal force. At shallower depths, the centrifugal force exceeds the pressure force, whence the resultant force drives the fluid outwards.

TELEMAC-2D allows to take into account the effect of these secondary currents. To activate this, key-word `SECONDARY CURRENTS` must be set to TRUE (default = FALSE). User can also manage some coefficients used in the resolved equation (see release note 7.0 for theoretical aspects and for more details). For instance, the production term in the advection-diffusion equation linearly depends on a coefficient $A_s$ which can be calibrated using the key-word `PRODUCTION COEFFICIENT FOR SECONDARY CURRENTS` (default = 7.071). In the same way, the dissipation part can be modified by varying the coefficient $D_s$ using the keyword `DISSIPATION COEFFICIENT FOR SECONDARY CURRENTS` (default = 0.5). Further details can be found in [41].

An example of use of the secondary currents is given in the validation case seccurrents (in the folder examples/telemac2d). This example is described and well documented in the validation manual of TELEMAC-2D.

# 11. Water quality

This chapter has been integrated in the WAQTEL user manual since v8.1.

# 12. Particle transport and lagrangian drifts

## 12.1 Drogues displacements

During a hydrodynamic simulation, TELEMAC-2D offers the possibility of monitoring the tracks followed by some particles (drogues) introduced into the fluid from outflow points. The result is produced in the form of a Tecplot format file or a binary PCL format file containing the various positions of the drogues in time, see paragraph 12.1.4 for more details.
Note that using this feature provides more accurate results than using the particle tracking features of the post-processing tools. Contrary to TELEMAC-2D for which monitoring floats is determined at each time step, the post-processing tools are based on the `RESULTS FILE` that is usually sampled much coarser. Since release 7.0, the management of drogues is modified to be coherent with other particle transport features of TELEMAC-2D (oil spill and algae). Hereafter, we give the implementation details.

### 12.1.1 Input Files

In addition to the mandatory files for a classical TELEMAC-2D model (steering, geometry, boundary conditions), there is the option to use an input file containing a number of polygons that specify an initial particle distribution.

### 12.1.2 Steering file

The following steering file keywords relate to drogues. The same keywords are used for oil spill and algae.

- The maximum number of allowed particles: `MAXIMUM NUMBER OF DROGUES`. This is the number used to allocate various arrays.
- The drogues printout period: `PRINTOUT PERIOD FOR DROGUES`. This sets the time step interval between successive outputs to the drogues output file,
- The name of the output file containing the drogues positions: `ASCII DROGUES FILE` (Tecplot format) or `BINARY DROGUES FILE` (binary PCL format). In the case of the PCL format, the file extension should be ".pcl" (See section 12.1.4),
- `DROGUES FILE FORMAT`: TECPLOT or BKBINPCL. If TECPLOT is chosen, the `ASCII DROGUES FILE` is written. If BKBINPCL is chosen, the `BINARY DROGUES FILE` is written,

- The file which specifies initial drogues placement and the class of each particle: `DROGUES INITIAL POSITIONING DATA FILE`. The $z$ coordinate sets the class of each particle,
- The format of the file that specifies initial drogues displacement: `FORMAT OF THE DROGUES POSITIONING DATA FILE`. Currently, the only option is BKASCI2S (BlueKenue i2s),
- The number of drogues per unit area for each drogue class (one number for each class): `INITIAL DROGUES SAMPLING DENSITY`,
- TELEMAC-2D offers the possibility to introduce a stochastic diffusion coefficient. When setting the keyword `STOCHASTIC DIFFUSION MODEL` = 1 (default = 0), a stochastic model will generate stochastically a diffusion coefficient which is computed using the turbulent viscosity. If no turbulence is activated, this stochastic diffusion is not considered during the particle transport.

### 12.1.3 Initial drogues locations

The initial drogues locations can be specified in three ways. Any combination of these three ways can be used.

1. The geometry file. An extra variable can be included in the geometry file, called DROGUES CLASSES. An extra variable can be added to the `GEOMETRY FILE` using BlueKenue. This variable specifies the drogues class. The number of drogues per unit area for each class is specified by the keyword `INITIAL DROGUES SAMPLING DENSITY`.

   **Detailed BlueKenue steps:**

   - `Load existing geometry file (File, Open...),`
   - `Create new Selafin file (File, New > SELAFIN Object...),`
   - `Drag bathymetry variable from existing geometry to new Selafin object.`
   - `Select the bathymetry variable, then start the Calculator (Tools, Calculator...),`
   - `Select the bathymetry variable as A,`
   - `Type a*0 in the Expression box (this defines a variable everywhere equal to 0),`
   - `Give the Result Name as DROGUES CLASSES,`
   - `Create a Polygon (see item 2) covering the area of the drogues, with the value equal to the class number,`
   - `Select the DROGUES CLASSES variable, then select Tools, Map Object...`
   - `Select the Polygon in Available Objects, then select Yes,`
   - `Drag DROGUES CLASSES into the new Selafin object,`
   - `Save the new Selafin object (File, Save).`

2. A polygon file. The polygon file name is given by the keyword DROGUES INITIAL POSITIONING DATA FILE. Each polygon defines an area and a drogues class. The polygon file can be created using BlueKenue. The $z$ coordinate sets the class of each particle. The number of drogues per unit area for each class is specified by the keyword INITIAL DROGUES SAMPLING DENSITY.

   **Detailed BlueKenue steps:**

   - Select New, Closed Line...
   - Use the mouse to click on the position of each polygon point,
   - Select New, Closed Line... again,
   - Type the name of the polygon in the Name box,
   - Type the class value in the Value box,
   - Click OK,
   - Save the polygon in i2s format (File, Save),

3. The FORTRAN user subroutine **USER_FLOT**. Commented out examples are included in **USER_FLOT**. Within **USER_FLOT**, the subroutine **ADD_PARTICLE** can be called. Each call of **ADD_PARTICLE** will add a drogue with a specified $x$, $y$, tag and class. Each particle should be defined with a unique tag number.

### 12.1.4 Output file

Besides the classic result file, TELEMAC-2D produces a specific output file for drogues. This file is a formatted file created by TELEMAC-2D during the computation and can be of two different formats:

- a file in the Tecplot format, it is given by the keyword ASCII DROGUES FILE. To visualize the drogues positions with Tecplot software, the user must:
  - Use the File>Load Data File(s) command to load the RESULTS FILE,
  - Use the File>Load Data File(s) command to load the Tecplot drogues file.

Warning:
In order to add the Tecplot DROGUES FILE to Telemac result data that was already loaded, select "Add to current data" set in the **Load Data File Warning dialogue** (cf. Figure 12.1). The Load Data File Warning dialogue will appear after you have selected the file and zones and/or variables to load.

- a file in the BlueKenue PCL format, the output file can be read in to BlueKenue. The file extension of the file should be "pcl".

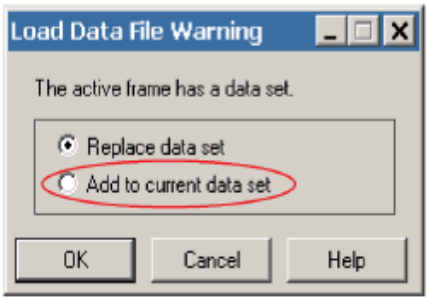


Figure 12.1: Warning dialogue box

**Additional drogues output formats:**

- It is possible to develop additional drogues output formats. This must be done in the subroutine **UTIMP_DROGUES**. A FORTRAN file including the subroutine **UTIMP_DROGUES** with the new format definition needs to be added with the input files. The user can choose whether to write to the `ASCII DROGUES FILE` or the `BINARY DROGUES FILE`.
- It is also possible to convert the `ASCII DROGUES FILE` in Tecplot format to a compatible paraview format. This process can be found in notebook documentation ("$HOMETEL/notebooks/pretel/converter.ipynb")

## 12.2 Algae modelling

Since release 6.3, TELEMAC-2D offers the possibility to simulate algae transport. Theoretical aspects about algae physics and modelling can be found in Joly [20].

### 12.2.1 Input files

Input files for algae modelling are the same as for drogues.

### 12.2.2 Steering file

All of the keywords listed for drogues in section 12.1.2 are also used for algae. In addition, the following keywords are used for algae modelling:

- The option setting the particles as algae: `ALGAE TRANSPORT MODEL = YES` (default = `NO`).
- The number of algae classes: `NUMBER OF ALGAE CLASSES` (default = 0). Keywords in items 3 to 7 have one number per algae class.
- The type of algae particles considered: `ALGAE TYPE` (default 1). The different choices are:
  - 1: Sphere,
  - 2: Iridaeca Flaccida,

  - 3: Pelvetiopsis Limitata,
  - 4: Gigartina Leptorhynchos.

- The physical properties of the algae (diameter, density and thickness):
  - DIAMETER OF ALGAE (default = 0.1 m),
  - DENSITY OF ALGAE (default = 1,050 kg/m$^3$),
  - THICKNESS OF ALGAE (default = 0.01 m),
- ALGAE RELEASE TYPE: 1 for timed release (default) or 2 for dislodgement,
- DURATION BEFORE ALGAE RELEASE: The time in seconds before the algae start moving. This applies when ALGAE RELEASE TYPE = 1,
- WAVE ORBITAL VELOCITY THRESHOLD FOR ALGAE 1, WAVE ORBITAL VELOCITY THRESHOLD FOR ALGAE 2 and RATE OF DEGRADATION FOR ALGAE. These three keywords are used to define algae dislodgement, which applies when ALGAE RELEASE TYPE = 2. This is explained in section 12.2.4.

Warning:

- Even though some of the previous keywords refer to drogues, they are also used for algae,
- To use the algae particle transport module it is necessary to use the $k-\varepsilon$ turbulence model, i.e. the option TURBULENCE MODEL = 3 needs to be set in the STEERING FILE.

### 12.2.3 Initial algae locations

The initial algae particles are specified in the same way as for drogues (see 12.1.3).

### 12.2.4 Dislodgement

If ALGAE RELEASE TYPE = 2, then an algae particle does not start to move until the wave orbital velocity at the bed exceeds a threshold value. This option only works if the TELEMAC-2D run is coupled with TOMAWAC. The TOMAWAC run supplies the wave orbital velocity to the TELEMAC-2D run. TOMAWAC calculates the wave orbital velocity by integrating the following equation, frequency by frequency, through the spectrum of waves set by the user in the TOMAWAC calculation (This approximates to the orbital wave velocity associated with a random wave - see for instance Soulsby (1997) [32]).

$$U_w = \frac{\pi H}{T \sinh(kh)}, \tag{12.1}$$

where $U_w$ is the orbital wave velocity amplitude, $H$ is the wave height, $h$ is the water depth, $T$ is the wave period and $k$ is the wave number.
The threshold value of the wave orbital velocity varies with time, according to the equation:

$$OV_0 = OV_1 + (OV_2 - OV_1)\exp(-A.T_{eff}), \tag{12.2}$$

$OV_1$, $OV_2$ and $A$ are specified in the STEERING FILE with the following keywords:

- $OV_1$ = WAVE ORBITAL VELOCITY THRESHOLD FOR ALGAE 1,
- $OV_2$ = WAVE ORBITAL VELOCITY THRESHOLD FOR ALGAE 2,
- $A$ = RATE OF DEGRADATION FOR ALGAE.

$T_{eff}$ is an "effective time" relating to the cumulative wave forcing that has been experienced by each algae particle, defined by: $T_{eff} = \sum OV.dt$ . This means that $T_{eff}$ is the numerical integral of wave orbital velocity over time.

### 12.2.5 Output files

The algae output file is specified in the same way as for drogues (see 12.1.4).

## 12.3 Oil spill modelling

A feature has been added to TELEMAC-2D that allows the simulation of oil spill problems. These developments are based on the work of Goeury [17].

### 12.3.1 Input files

In addition to the minimum set of input files necessary to run a TELEMAC-2D case, an oil spill computation needs also an OIL SPILL STEERING FILE. Furthermore, to run oil spill model, a FORTRAN file including the routine **OIL_FLOT** needs to be added.

### 12.3.2 Steering file

The following essential information should be specified in the TELEMAC-2D STEERING FILE to run an oil spill propagation model:

- The use of the oil spill model must be declared: OIL SPILL MODEL = YES (default = NO),
- The name of the oil spill steering file which contains the oil characteristics: OIL SPILL STEERING FILE,
- The number of oil particles to be released during the oil spill episode: MAXIMUM NUMBER OF DROGUES,
- The frequency of the drogues printout period: PRINTOUT PERIOD FOR DROGUES,
- The name of the Tecplot file containing the oil displacement: ASCII DROGUES FILE.

Warning:

- Even though some of the previous keywords may refer to drogues, they are also used for algae blooms and oil spills.

  With the oil spill module, it is possible to take into account the transport of soluble oil components in water (whose presence has no effect on the hydrodynamics). These may or may not be diffused within the flow but their characteristics have to be defined in the OIL SPILL STEERING FILE. If these components are allowed to diffuse in the flow, they are then treated with the tracer transport computations of TELEMAC-2D. This implies that the NUMBER OF TRACERS must be set to the number of the oil soluble components.

- If the number of oil components dissolved in water is greater than 1, the result file can contain the sum of dissolved oil concentrations. The user must only add the variable for graphic printout N:

  VARIABLES FOR GRAPHIC PRINTOUTS: '....,N'

  With the variable for graphic printout **N**, be careful not to have private tables or change the table PRIVE1 in the **PRERES_TELEMAC2D** subroutine by the **PRIVEX** array (where X is the number chosen by the user).

### 12.3.3 Oil spill steering file

As seen previously, the OIL SPILL STEERING FILE name is given by the user in the TELEMAC-2D STEERING FILE. This file contains all information on oil calculation based on the composition considered by the user:

- The number of non-soluble components in oil,
- The parameters of these components such as the mass fraction (%) and boiling point of each component (K),
- The number of soluble components in oil,
- The parameters of these components such as the mass fraction (%), boiling point of each component (K), solubility ($kg.m^{-3}$) and the mass transfer coefficient of the dissolution and volatilization phenomena (m/s),
- The oil density,
- The oil viscosity ($m^2.s^{-1}$),
- The volume of the spilled oil ($m^3$),
- The water surface temperature (K),
- The spreading model chosen by the user:

  1. Fay's model,
  2. Migr'Hycar model,

3. Constant area model.

**Warning:**

- The parameters of soluble (or non-soluble) components need to be informed only if the number of these components is not null,
- If the sum of all mass fraction components is not equal to 1, the run is interrupted and an error message is displayed:

  "WARNING::THE SUM OF EACH COMPONENT MASS FRACTION IS NOT EQUAL TO 1."

  "PLEASE, MODIFY THE INPUT STEERING FILE"

An example of the `OIL SPILL STEERING FILE` is given.

```
NUMBER OF UNSOLUBLE COMPONENTS IN OIL
6
UNSOLUBLE COMPONENTS PARAMETERS (FRAC MASS, TEB)
5.1D-02      ,402.32D0
9.2D-02      ,428.37D0
3.16D-01     ,458.37D0
3.5156D-01      ,503.37D0
8.5D-02         ,543.37D0
9.4D-02         ,628.37D0
NUMBER OF SOLUBLE COMPONENTS IN OIL
4
SOLUBLE COMPONENTS PARAMETERS(FRAC MASS, TEB, SOL, KDISS, KVOL)
1.D-02    ,497.05D0,   0.018D0    , 1.25D-05 ,5.0D-05
3.2D-02   ,551.52D0,   0.00176D0  , 5.63D-06 ,1.51D-05
1.D-04    ,674.68D0,   2.0D-04    , 2.D-06   ,4.085D-07
2.D-05    ,728.15D0,   1.33D-06   , 1.33D-06 ,1.20D-07
OIL DENSITY
830.D0
OIL VISCOSITY
4.2D-06
OIL SPILL VOLUME
2.02D-05
WATER TEMPERATURE
292.05D0
SPREADING MODEL(1=FAY'S MODEL,2=MIGR'HYCAR MODEL,3=CONSTANT AREA)
2
```

If in the `OIL SPILL STEERING FILE`, the SPREADING MODEL is set to 3, two lines must be added to the previous example:

```
CONSTANT AREA VALUE CHOSEN BY THE USER FOR EACH OIL PARTICLE
1
/example if the user wants area particle equal to 1 m2
```

### 12.3.4 The OIL_FLOT subroutine

After inserting the **OIL_FLOT** subroutine in the FORTRAN FILE, the user must modify it in order to indicate the release time step, together with the coordinates of the release point. If the release point coordinates are outside the domain, the run is interrupted and an error message is displayed. In addition, if a particle leaves the domain during the simulation, it is of course no longer monitored but its previous track remains in the results file for consultation.
An example of modifications in the **OIL_FLOT** subroutine is given.
The release time step in the first condition statement and the coordinates of the release point must be changed:

```
...
IF(LT.EQ.10000) THEN
  NUM_GLO=0
  NUM_MAX=0
  NUM_LOC=0
  COORD_X=0.D0
  COORD_Y=0.D0
  NUM_MAX=INT(SQRT(REAL(NFLOT_MAX)))
  DO K=1,NUM_MAX
    DO J=1,NUM_MAX
      COORD_X=336000.D0+REAL(J)
      COORD_Y=371000.D0+REAL(K)
      NUM_GLO=NUM_GLO+1
      NFLOT_OIL=0
      CALL ADD_PARTICLE(COORD_X,COORD_Y,0.D0,NUM_GLO,NFLOT_OIL,
&                       1,XFLOT,YFLOT,YFLOT,TAGFLO,
&                       SHPFLO,SHPFLO,ELTFLO,ELTFLO,MESH,1,
&                       0.D0,0.D0,0.D0,0.D0,0,0)
...
    END DO
  ENDDO
END IF
```

### 12.3.5 Output files

There are no additional output files provided than for drogues transport.

> **Warning:**
> If the user wants to develop a new drogues output format for oil spill, he/she must edit **OIL_DERIVE** subroutine and not the **DERIVE** subroutine used for the drogues and algae transport.

## 12.4 Lagrangian drifts

### 12.4.1 Input files

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 written 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.

### 12.4.2 Output files

The result is produced in the form of a SERAFIN format file containing the various positions of the lagrangian drifts in the form of a degenerated mesh.
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 value (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.

Warning:
Lagrangian drifts have not been implemented for parallelism yet.

# 13. Construction works modelling

## 13.1 Weirs

Weirs are considered as linear singularities. Their use is possible in parallel computing (since release 6.2). The number of weirs is specified by the keyword `NUMBER OF WEIRS` (default value 0). Information about weirs is given in the `WEIRS DATA FILE`.
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 weirs      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 dyke
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 dyke
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 meaning:

- 0: the velocities are null (recommended option),
- 1: the velocities will be calculated with the Chézy formula (as a function of the local free surface slope).

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 global numbers (line 7 for the first weir). Note, that before and for release 6.1, the numbering to provide was not the global one but the local numbering of the boundary defined in the BOUNDARY CONDITIONS FILE. However, it is necessary to provide the weirs number in the order of the boundary points.
The numbers of their twin points on side 2 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 m. All these data are repeated for all weirs.
The formulae used to calculate the discharge for each point are the following:

- unsubmerged weir: $Q = \mu\sqrt{2g}\,(upstream - weir)^{\frac{3}{2}}$,
- submerged weir:

$$Q = \left(\frac{2}{3}\sqrt{\frac{1}{3}}\right)^{-1} \mu\sqrt{2g}\,(downstream - weir)\,\sqrt{(upstream - weir)},$$

- the weir is not submerged if:

$$upstream\ level < \frac{weir\ level + 2 \times upstream\ level}{3}.$$

Depending on the shape and roughness of the weir, the value of $\mu$ is between 0.4 and 0.5. However, the above formulae neglect the velocity of the upstream head in the computation. If this is not the case, the value of $\mu$ may be higher.
If the user wants to modify the different laws, it is possible to modify the appropriate subroutines (**LOIDEN** and **LOINOY**).

The keyword TYPE OF WEIRS gives the method to treat weirs. 2 options are available:

- horizontal with same number of nodes upstream/downstream (Historical solution with the **BORD** subroutine, which is the default value),
- general (new solution with sources points).

## 13.2 Culverts

As for weirs, the keyword NUMBER OF CULVERTS (default value = 0) specifies the number of culverts to be treated. Culverts are described as couples of points between which flow may occur, as a function of the respective water level at these points. Since release 6.2 of TELEMAC-2D, it is no longer necessary to describe each culvert inflow and outflow as a source point.
There are two options to treat culverts in TELEMAC-2D. The choice can be done with OPTION FOR CULVERTS (default value = 1). For more information about this choice, the reader is

invited to refer to the TELEMAC-3D theory guide.

Information about culvert characteristics is stored in the CULVERTS DATA FILE.
The following file gives an example of a culvert:

```
Relaxation , Number of culverts
0.2  1
I1   I2   CE1 CE2 CS1 CS2 LRG   HAUT1 CLP LBUS Z1   Z2   CV   C56 CV5 C5
CT   HAUT2 FRIC LENGTH CIRC D1   D2 A1 A2   AA
199  640  0.5  0.5  10   1.0  2.52  2.52   0    0.2   0.3  0.1  0.0  0.0  0.0  0.0  0.0  0.0  2.52
0.0   0.0      1      90.  0.  0.  90.  0
```

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. Relaxation, at time $T$, between result computed at time $T$ and result computed at previous time step. A relaxation coefficient of 0.2 means that 20% of time $T$ result is mixed with 80% of the previous result. I1 and I2 are the numbers of each end of the culvert in the global point numbering system.
The culvert discharge is calculated based on the formulae given in the TELEMAC-3D theory guide and in the release notes of TELEMAC-2D: CE1 and CE2 are the head loss coefficients of 1 and 2 when they are operating as inlets. CS1 and CS2 are the head loss coefficients of 1 and 2 when they are operating as outlets. LRG is the width of the culvert. HAUT1 and HAUT2 are the heights of the construction work (in meters) at the inlet and outlet. The flow direction is also imposed through the keyword CLP:
CLP = 0, flow is allowed in both directions,
CLP = 1, flow is only allowed from section 1 to section 2,
CLP = 2, flow is only allowed from section 2 to section 1,
CLP = 3, no flow allowed.
LBUS is the linear head loss in the culvert, 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 inlet and outlet. CV refers to the loss coefficient due to the presence of a valve and C56 is the constant used to differentiate flow types 5 and 6 in the formulation by Bodhaine. C5 and CV5 represent correction coefficients to C1 and to CV coefficients due to the occurrence of the type 5 flow in the Bodhaine formulation. CT is the loss coefficient due to the presence of trash screens. FRIC is the Manning Strikler coefficient. LENGTH is the length of the culvert, and the culvert's shape can be specified through the parameter CIRC (equal to 1 in case of a circular section, 0 for a rectangular section). A1 and A2 are the angles with respect to the $x$ axis. D1 and D2 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°. They are used to account for the current direction at source or sink point. AA is a parameter which allows the user to choose whether A1 and A2 are automatically computed by TELEMAC-2D or whether the data file values are used to set these angles: AA=1 – automatic angle; AA=0 – user-set angle.

## 13.3 Dykes breaches

### 13.3.1 General overview

TELEMAC-2D allows simulating dykes breaching by suddenly or gradually lowering the altitude of some points. This feature is enabled using the logical keyword BREACH (default = NO). The different kinds of breach developments are completely controlled by user via the BREACHES DATA FILE. Regardless the kind of breach development, to model this phenomenon, it is necessary to know:

- the final length of the breach,
- the duration of the breaching process,
- the final altitude of the breach.

Indeed the breach is represented in TELEMAC-2D by a polygon (defined by users in the BREACHES DATA FILE) within the altitude of mesh nodes lowers with time according to the selected breach law.
Current state-of-the-art on the breaching of earthen dykes due to overtopping flows shows that the dyke breaching expansion is progressive (non-instantaneous) ([21, 26–29, 44], among others), following two main phases (referred to as breach formation and development period, respectively):

- Phase 1 - Deepening and lateral widening: as the overtopping flow depth and velocity over the dyke increase, both breach deepening and widening are promoted with a shift of the breach centerline toward the channel downstream end. The breach sides collapse gradually. The breach expansion during this phase is fast,
- Phase 2 - Lateral widening: the main channel free surface decreases and the flow depth starts stabilizing at its minimum level (approaching the main channel critical flow depth). The breach development becomes slower, the upstream part of the breach stops evolving, and deepening becomes moderate tending to stabilize. The breach widens along the channel flow direction due to side slope failures.

The breach deepening (vertical incision of the breach) is faster than breach widening ([22, 26, 27, 40]). When the breach bottom reaches the foundation of the dyke or a non-erodible layer, no further deepening of the breach is possible and lateral widening is controlling the breach expansion until its stabilisation (i.e. fully formed breach and erosion is stopped).
Selected empirical laws have been implemented in TELEMAC-2D for simulating the time evolution of the breach expansion (widening and deepening). They are described here below.
As the flood period and inundation of the floodplain along with the dyke material characteristics impose certain limits on breach growth, the empirical laws are applied over a given duration. The breach expansion continues until the breach has expanded to its approximate maximum dimensions. Therefore the final (i.e. ultimate, maximum) breach dimensions (width and bottom elevation), the duration of the breaching expansion (or duration of each phase), must be estimated outside of the TELEMAC software by the user. These parameters are indeed mandatory in the BREACHES DATA FILE which contains the description and the characteristics of the breaching process; it will be described here below. Except for the Froehlich model, the breach longitudinal shape is assumed rectangular.
In addition to the breach expansion computation, it is important to define a criterion to start the breaching process. In the current release, 3 types of criteria are available:

1. at a given time. This option can be chosen filling the BREACHES DATA FILE with:

```
# Option for breaching initiation
1
```

2. when the water level above the dyke reaches a given value (in case of overflow). This option can be chosen filling the BREACHES DATA FILE with:

```
# Option for breaching initiation
2
```

3. when the water level at a given point reaches a certain value (in case of safety level). This option can be chosen filling the `BREACHES DATA FILE` with:

```
# Option for breaching initiation
3
```

Since release 7.0, it is possible to take into account a lateral growth of the breach (dyke opening by widening and deepening). Old breaching processes are not affected by this new feature: only the breach deepening was taken into account in the breach computation.
Since release 8.3, several laws for lateral growth have been implemented and they are described in the following section.

### 13.3.2 Breach computation for rectangular shapes

#### Breach widening

For the sake of simplicity, the following empirical laws are given assuming a zero-value for the initial width of the breach.

- Linear widening
  The time evolution breaching process is simulated with the following formula:

$$B(t) = \frac{B_f}{T_f} t \quad \text{for } t \leq T_f \tag{13.1}$$

  where $B_f$ is the final breach width, $T_f$ is the total duration of the breach expansion in hours and $t$ in hours.
  This law can be selected filling the `BREACHES DATA FILE` with:

```
# Option for lateral growth
2
```

- User-defined breach expansion formulations

  The breach widening can be simulated according to a linear-time progression formula. The user has the possibility to specify one or two growth rates: to mimic the real breach widening the user can simulate a breach that initially grows very quickly then slows down towards the end of the development time. The formula is:

$$B(t) = \begin{cases} E_{w1} t & \text{for } t \leq T_1 \\ E_{w1} T_1 + E_{w2}(t - T_1) & \text{for } T_1 \leq t \leq T_f \end{cases} \tag{13.2}$$

  with $t$ time in hours (after the initiation of breaching), $B$ = breach width in meters, $T_1$ = duration of phase 1 in hours, $T_f$ = total duration of the breach expansion (phase 1 and phase 2) in hours, $E_{w1}$ and $E_{w2}$ = breach growth rates (m/h). This law can be selected filling the `BREACHES DATA FILE` with:

```
# Option for lateral growth
3
```

  Information for defining these growth rate parameters could be obtained from literature or physically-based models ([1, 39, 42]). Using available datasets, Resio et al. (2009) concluded that the rate of breach widening is ranging between 9 m/h for erosion-resistant soils (cohesive dykes) and 60 m/h for erodible alluvial material (sand and gravel soils). The widening rate can reach (rarely) 300 m/h for very erodible dykes. USBR [23] recommended a single breach widening rate of 91 m/h for embankment dams - formula taking into account this rate is already implemented for users who wish to apply it.

- USBR formula (1988)
  The breach widening is estimated as a function of time with the following linear progression formula proposed by USBR [23]:

$$B(t) = 91t \quad \text{for } t \leq T_f \tag{13.3}$$

  with $t$ = time in hours (after the initiation of breaching) and $B$ = breach width in meters. This law can be selected filling the `BREACHES DATA FILE` with:

```
# Option for lateral growth
4
```

- Von Thun and Gillette formulas (1990)
  Von Thun and Gillette [38] developed two pairs of equations for breach widening in dykes of low and high erodibility:

  – for erodible dykes (i.e. non-cohesive dykes):

$$B(t) = (4h_w + 61)t \quad \text{for } t \leq T_f \tag{13.4}$$

    This law can be selected filling the `BREACHES DATA FILE` with:

```
# Option for lateral growth
5
```

  – for erosion-resistant dykes (i.e. cohesive dykes):

$$B(t) = 4h_w t \quad \text{for } t \leq T_f \tag{13.5}$$

    This law can be selected filling the `BREACHES DATA FILE` with:

```
# Option for lateral growth
6
```

  with $t$ = time in hours (after initiation of breaching), $B$ = breach width in meters, and $h_w$ = depth of water above the breach invert in meters.

- Verheij formula (2002)
  Verheij [35] provided a simple relationship between the breach width $B$ and time for sand and clay levees, based on field and laboratory data sets:

  – for sand levees (i.e. non-cohesive dykes):

$$B(t) = 37.2t^{0.51} \quad \text{for } t \leq T_f \tag{13.6}$$

    This law can be selected filling the `BREACHES DATA FILE` with:

```
# Option for lateral growth
7
```

  – for clay levees (i.e. cohesive dykes):

$$B(t) = 13.4\sqrt{t} \quad \text{for } t \leq T_f \tag{13.7}$$

    This law can be selected filling the `BREACHES DATA FILE` with:

Table 13.1: Suggested and range of values for coefficients $f_1$ and $f_2$

| Coefficient | Suggested | Range |
|---|---|---|
| $f_1$ | 1.3 | 0.5-5 |
| $f_2$ | 0.04 | 0.01-1 |

```
# Option for lateral growth
8
```

with $t$ = time in hours (after initiation of breaching), and $B$ = breach width in meters.

- Verheij and Van der Knaap (2003) formula
  Verheij and Van der Knaap [36] improved the previous formulation by including the effect of the difference in water levels at both sides of the dyke at the breach location, and the critical flow velocity for the initiation erosion of the dyke material. The empirical equation in its integral form reads as:

$$B(t) = f_1 \frac{g^{0.5} \Delta H^{1.5}}{u_c} \log\left(1 + f_2 \frac{g}{u_c} t\right) \quad \text{for } t \leq T_f \tag{13.8}$$

  With $t$ = time in hours (after the initiation of breaching); $B$ = breach width in meters; $u_c$ = critical flow velocity for the initiation of erosion of dyke material (m/s); $f_1$ and $f_2$ = coefficients; $g$ = gravitational acceleration (m/s$^2$). $\Delta H$ (m) denotes the difference in water level between the upstream and downstream sides of the breach. In TELEMAC-2D, this difference is computed by considering the water head instead of water level, i.e. $\Delta H$ (m) = $H_{up} - H_{down}$ with $H_{up}$ the hydraulic head upstream of the breach (channel side) and $H_{down}$ the hydraulic head downstream of the breach (floodplain side). In addition, in TELEMAC-2D the differential form is implemented instead of the integral one, which reads as:

$$B(t) = \frac{f_1 f_2}{ln10} \frac{(g\Delta H)^{1.5}}{u_c^2} \frac{1}{1 + \frac{f_2 g}{u_c} t} \Delta t \quad \text{for } t \leq T_f \tag{13.9}$$

  The suggested values and ranges have been proposed by Verheij and Van der Knaap [36] for coefficients $f_1$ and $f_2$ (see Table 13.1) ([13, 34]). Equation (13.8) contains the critical flow velocity $u_c$ for the surface erosion of dyke material. Table 13.2 shows characteristic values for critical flow velocity for various soils based on research by Verheij [35].

  This law can be selected filling the `BREACHES DATA FILE` with:

```
# Option for lateral growth
9
```

#### Breach deepening

As mentioned previously, the breach deepening is evolving faster than the breach widening. In the TELEMAC-MASCARET SYSTEM, the breach minimum level $Z_{b,min}$ (elevation of the dyke foundation, main channel bottom or of a rigid layer) is reached in a short period. Therefore, the time-evolution of the breach invert elevation is simulated according to the following linear-time progression law:

$$Z_b(t) = Z_{b0} - \frac{Z_{b0} - Z_{b,min}}{T_d} t \quad \text{for } t \leq T_d \tag{13.10}$$

Table 13.2: Strength characteristics of various soil types [36][37]

| Type of Soil | $u_c$ (m/s) |
|---|---|
| Grass, good | 7 |
| Grass, moderate | 5 |
| Grass, bad | 4 |
| Clay, good (compact; $\tau_{\text{undrained}}$ = 800-100 kPa) | 1.0 |
| Clay with 60% sand (firm; $\tau_{\text{undrained}}$ = 40-80 kPa) | 0.8 |
| Good clay with less structure | 0.7 |
| Good clay, heavily structured | 0.6 |
| Bad clay (loose; $\tau_{\text{undrained}}$ = 20-40 kPa) | 0.4 |
| Sand with 17% silt | 0.23 |
| Sand with 10% silt | 0.20 |
| Sand with 0% silt | 0.16 |

with $t$ = time (after the initiation of breaching), $Z_b$ = elevation of breach invert, $Z_{b0}$ = initial elevation of breach invert, $T_d$ = duration of breach deepening in hours. By default, the duration $T_d$ is taken 10 times smaller than the total duration of breach expansion $T_f$.
Note that the breach deepening computation is common for all the laws presented here above. If the user wishes to model only breach deepening (without widening), the `BREACHES DATA FILE` must be filled as follows:

```
# Option for lateral growth
1
```

### 13.3.3 Breach computation for trapezoidal shapes

Froehlich [16] proposed an empirical approach, composed of three breach evolution variants, developed originally by Fread and Harbaugh [15] to approximate breach expansion (widening and deepening). Each of the three models assumes that a breach begins to form at the top and grows with time into a trapezoidal shape (see Figure 13.1):

- Model A: the breach develops initially following a triangle shape, until the bottom of the breach reaches its lowest elevation. Then lateral expansion begins, and the breach shape becomes trapezoidal. The slope of breach sides is assumed constant,
- Model B: the base width of the trapezoidal shape increases gradually as the breach deepens. The slope of the breach sides is assumed constant,
- Model C: the bottom width of the trapezoidal breach is considered constant. The top width of the breach and the depth increase gradually. The slope of the breach sides is time-varying.

Froehlich [16] used the concept expressed by Brunner [8] who proposed a sine-curve time progression (instead of the common linear time evolution), reflecting slower growth at the start; then acceleration and again slow finish of breach development. In TELEMAC-2D, Models A and B are combined and adapted for two-dimensional simulations as follows:

- the instantaneous top width of the breach is computed as:

$$B(t) = \beta(t) B_f \quad \text{for } t \leq T_f \quad \text{with } \beta(t) = \frac{1}{2}\left\{1 + \sin\left[\pi\left(\frac{t}{T_f} - \frac{1}{2}\right)\right]\right\} \tag{13.11}$$

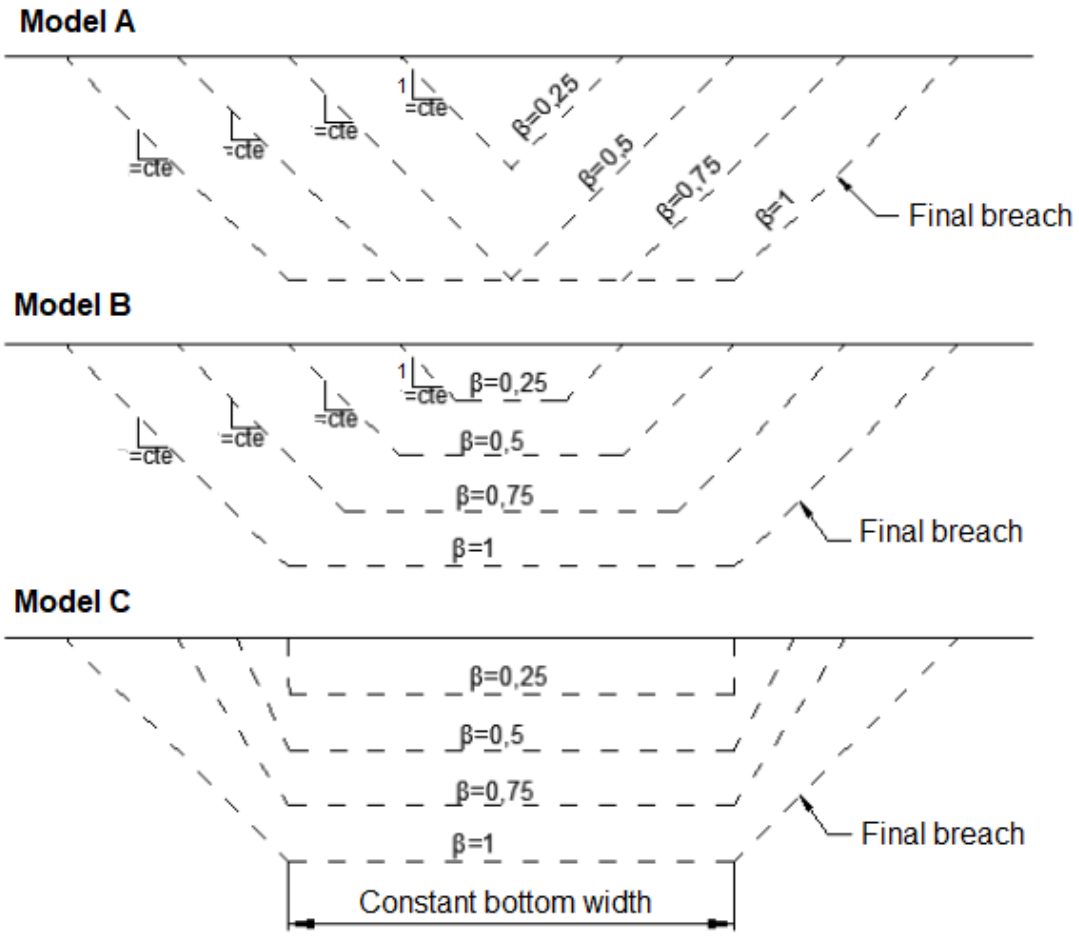


Figure 13.1: Schematic representations of three empirical breach formation models (adapted from Bennani [7])

- the instantaneous elevation of the breach bottom is calculated as follows:

$$Z_b(t) = Z_{b0} - \beta_1(t)(Z_{b0} - Z_{b,min}) \quad \text{for } t \leq T_d \quad \text{with } \beta_1(t) = \frac{1}{2}\left\{1 + \sin\left[\pi\left(\frac{t}{T_d} - \frac{1}{2}\right)\right]\right\} \tag{13.12}$$

with $t$ = time in hours (after the initiation of breaching), $B_f$ = final top width of the breach in meters. Note that $T_f$ and $B_f$ are user-defined.
This law can be selected filling the BREACHES DATA FILE with

```
# Option for lateral growth
10
```

### 13.3.4 Breaches data file

In order to give all informations about the breach process, the user has to complete the BREACHES DATA FILE. When the initial lengths of breaches are known, the keyword INITIAL LENGTHS OF BREACHES must be set to TRUE in the steering file (default value = NO). If the initial lengths are unknwown, TELEMAC-2D will automatically compute the initial length for the breaching process. First, the initial length is estimated as a tenth of the final breach length; then it is computed as the maximum value between the estimated initial length and the distance betweeen the central points of the polyline defining the breach. Indeed in TELEMAC-2D we assume that the breaching process will start at the middle of the polyline. The breaching zone is thus defined by a polyline of several points associated to a bandwidth. The final situation is characterized by a bottom altitude that will be reached by all the points located in the breaching zone.
To avoid errors, the BREACHES DATA FILE has to be written in the following way (the order of parameters cannot be changed):

- Number of breaches,
- Width of the polygon defining the breaches (in m),
- Option for the breaching initiation (from 1 to 3),
- If the option of breaching initiation is at a given time (option 1): breach opening moment in seconds,
- Duration of the breaching process (in s),
- Option for lateral growth (from 1 to 10),
- Final bottom altitude of the breach (in m),
- If option of breaching initiation corresponds to controlling level of breach with global node (option 3): number of global node controlling the breach,
- If option of breaching initiation is not at given time: control level of the breach (in m),
- If the user defined formula with two growth rates is used for lateral growth (option 3): duration for first step, growth rate for step 1 and growth rate for step 2 (in m/hours),
- If the Verheij 2003 formula is used for lateral growth (option 9): critical flow velocity for the initiation of erosion of dyke material (m/s); $f_1$ and $f_2$ empirical coefficients (/),

- If the Froehlich formula is used for lateral growth (option = 10): difference between initial and final elevation of the dyke (m),
- If the initial length is known: value of initial length (m),
- Number of points of the polyline defining the breach,
- Description of the polyline (points with x and y coordinates).

A commented example of BREACHES DATA FILE is provided below. This example is taken from the test case telemac2d/breach.

```
# Number of breaches
3
# Definition of upstream breach
# Width of the polygon defining the breaches
50.0
# Option for the breaching initiation
2
# Duration of the breaching process (0.0 = instantaneous)
300.0
# Option for lateral growth Verheij 2002 (non cohesive)
7
# Final bottom altitude of the breach
5.9
# Control level of the breach
7.2
# Number of points of the polyline
4
# Description of the polyline
2000.0 37.5
2041.0 37.5
2082.0 37.5
2100.0 37.5
# Central breach definition
# Definition of central breach
# Width of the polygon defining the breaches
60.0
# Option for the breaching initiation
3
# Duration of the breaching process (0.0 = instantaneous)
900.0
# Option for lateral growth
10
# Final bottom altitude of the breach
5.5
# Number of global node controlling the breach
9406
# Control level of the breach
6.0
# Difference between initial and final elevation of the dyke
2.049
```

```
# Number of points of the polyline
4
# Description of the polyline
2450.0 37.5
2500.0 37.5
2520.0 37.5
2550.0 37.5
# Downstream breach definition
# Width of Polygon defining the breach
10.0
# Option for the breaching process
1
# Start time of the breaching process
2000.0
# Duration of the breaching process (0.0 = instantaneous)
600.0
# Option of lateral growth
# (1= bottom lowering, 2= opening by widening)
1
# Final bottom altitude of the breach
5.0
# Number of points on the dyke axis where the breach will appear
4
# Description of the polyline
2900.0 37.5
2920.0 37.5
2950.0 37.5
3000.0 37.5
```

It is worth noting that the duration of the breaching process ($T_f$) is effectively used in TELEMAC-2D to compute the length of the breach in case of linear laws and the Froehlich law. For other laws, the breach will continue widening until the final length (defined by the polyline) will be reached.
Despite that, the duration of the breaching process is always considered to compute the duration of breach deepening ($T_d$).

# 14. Other configurations

## 14.1 Modification of bottom topography (USER_CORFON)

Bottom topography may be introduced at various levels, as stated in section 3.2.16.

TELEMAC-2D offers the possibility of modifying the bottom topography at the beginning of a computation using the **USER_CORFON** subroutine. This is called up once at the beginning 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 (which calls the **USER_CORFON** subroutine) carries out a number of bottom smoothings equal to **LISFON**, i.e. equal to the number specified by the keyword `BOTTOM SMOOTHINGS` for which the default value is 0 (no smoothing). The call to bottom smoothings can be done after or before the call to **USER_CORFON** in the **CORFON** subroutine with the keyword `BOTTOM SMOOTHINGS AFTER USER MODIFICATIONS`. By default, potential bottom smoothings are done after potential modifications of the bottom in **USER_CORFON** subroutine (default = YES).

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

## 14.2 Modifying coordinates (USER_CORRXY)

TELEMAC-2D also offers the possibility of modifying the mesh point coordinates at the beginning 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 translate the object.

The modification is done in the **USER_CORRXY** subroutine (BIEF library), which is called up at the beginning of the computation. This subroutine is empty by default and gives an example of programming a change of scale and origin, within 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 (default = (0;0)). 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 do not have a use yet.

## 14.3 Spherical coordinates (LATITU)

If a simulation is 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 set to YES (default value = NO). In this case, TELEMAC-2D calls a subroutine named **LATITU** through the subroutine **INBIEF** 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 origin point of the mesh provided by the user in the steering file with the keyword LATITUDE OF ORIGIN POINT (default value = 48 degrees).
By default, TELEMAC-2D assumes that the mesh coordinates are given in Cartesian coordinates. The user can change this choice by using the keyword SPATIAL PROJECTION TYPE (default is 1 which corresponds to Cartesian coordinates). Indeed, when choosing the value 2, the coordinates are considered in accordance with Mercator's projection. The value 3, the mesh has to be in longitude-latitude (in degrees!). It is important to notice here that, if option SPHERICAL COORDINATES = YES, SPATIAL PROJECTION TYPE has to be 2 or 3.
The **LATITU** subroutine (BIEF library) may be modified by the user to introduce any other latitude-dependent computation.

## 14.4 Adding new variables (USER_NOMVAR_TELEMAC2D and USER_PRERES_TELEMAC2D)

A standard feature of TELEMAC-2D is the storage of some computed variables. In some cases, the user may wish to compute other variables and store 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.
In the **USER_NOMVAR_TELEMAC2D** subroutine, 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 semicolons 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)** and **PRIVE%ADR(4)%P%R(X)** (in which **X** is the number of nodes 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 **USER_CORRXY**, **USER_CORSTR**, **USER_BORD** etc. If a **PRIVE** table is used to program 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 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 **USER_NOMVAR_TELEMAC** 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 restart. 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 **USER_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.

## 14.5 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)** fills table **F** with the **C** value in the convex polygon defined by **NSOM** nodes (coordinates **XSOM, YSOM**). The variable **MESH** is needed for the **FILPOL** subroutine but has no meaning for the user.

## 14.6 Validating a computation (BIEF_VALIDA)

The structure of the TELEMAC-2D software offers an entry point for validating a computation, in the form of a subroutine named **BIEF_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 **BIEF_VALIDA** subroutine is called at each time step when the keyword `VALIDATION` has the value YES, enabling a comparison to be done with the validation solution at each time step. By default, the **BIEF_VALIDA** subroutine only does a comparison with the last time step. The results of this comparison are given in the output listing.

## 14.7 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 some 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 done in the **PROPIN_TELEMAC2D** subroutine.

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

## 14.8 Coupling

The principle of coupling two or several 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 couple the hydrodynamic module and the 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 water depth 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.

Several modules can be coupled in the current version of the code: the sediment transport modules GAIA or SISYPHE, the sea state computational module TOMAWAC, the water quality module WAQTEL (and even DELWAQ) and the ice module KHIONE. The time step used for the two calculations is not necessarily the same and is managed automatically by the coupling algorithms and the keyword `COUPLING PERIOD FOR SISYPHE` and `COUPLING PERIOD FOR TOMAWAC` with default values 1 (coupling at every iteration).
This feature requires two keywords. The keyword `COUPLING WITH` indicates which simulation code is to be coupled with TELEMAC-2D. The values of this keyword can be:

- `COUPLING WITH` = 'GAIA' for coupling with the GAIA module,
- `COUPLING WITH` = 'SISYPHE' for coupling with the SISYPHE module,
- `COUPLING WITH` = 'TOMAWAC' for coupling with the TOMAWAC module,
- `COUPLING WITH` = 'TOMAWAC2' for coupling with the TOMAWAC module and possibly different geometry files (see below),
- `COUPLING WITH` = 'WAQTEL' for coupling with the WAQTEL module,
- `COUPLING WITH` = 'KHIONE' for coupling with the KHIONE module.

If wanting to couple with 2 modules or more, the differents modules are to be written in the `COUPLING WITH` separated with semicolon, for example: `COUPLING WITH` = 'GAIA, TOMAWAC'.
Depending on the module(s) used, the keywords `GAIA STEERING FILE`, `SISYPHE STEERING FILE`, `TOMAWAC STEERING FILE`, `WAQTEL STEERING FILE` and `KHIONE STEERING FILE` indicate the names of the steering files of the coupled modules.

If coupling with the water quality module WAQTEL, the integer keyword `WATER QUALITY PROCESS` must be set to a value different from 1 (default = 1) in the TELEMAC-2D steering file. The possible choices are:

- 0: all available processes,
- 1: nothing (default value),
- 2: $O_2$ module,
- 3: BIOMASS module,
- 5: EUTRO module,
- 7: MICROPOL module,
- 11: THERMIC module,
- 17: degradation law.

Several modules can be combined by giving the multiplication of the process choices, e.g. 55 = 5 $\times$ 11 activates EUTRO and THERMIC modules. Note that AED2 is currently not coupled with TELEMAC-2D contrary to TELEMAC-3D. Please refer to the WAQTEL documentation for additional informations for WAQTEL.

If coupling with the ice module KHIONE, the integer keyword `ICE PROCESSES` must be set to a value different from 1 (default = 1) in the TELEMAC-2D steering file. This keyword provides the ice process number with the number being defined on the basis of a multiplication of primary numbers (2, 3, 5, 7, 11, 13...). For instance, 14 (= 2 $\times$ 7) activates processes 2 and 7. The possible choices are:

- 0: all available processes are included,
- 1: no ice process included (default value),
- 2: thermal budget+frazil growth,
- 3: ice cover impact on hydrodynamics,
- 5: clogging on racks,
- 7: static border ice growth.

The FORTRAN files of the different modules can be used and are compiled independently (check that the FORTRAN files of GAIA, SISYPHE and TOMAWAC do not contain a main program).

The keyword `COUPLING WITH` is also used if the computation has to generate the appropriate files necessary to run a water quality simulation with DELWAQ. In that case, it is necessary to specify `COUPLING WITH` = 'DELWAQ'. Please refer to Appendix D for all informations concerning communications with DELWAQ.

In the case of coupling TELEMAC-2D with GAIA or SISYPHE, the bed roughness can be determined directly by GAIA if the kyeword `COMPUTE BED ROUGHNESS AT SEDIMENT SCALE` is enabled or by SISYPHE if the keyword `BED ROUGHNESS PREDICTION` is enabled in the settings file sediment transport model. This option can be useful for applications accounting for rippled or mega-rippled bed or dunes. If this option is used, the friction law on the bottom used in the hydrodynamic calculation of TELEMAC-2D must necessarily be the law

of Nikuradse (LAW OF BOTTOM FRICTION = 5).

A very particular coupling can be performed using COUPLING WITH = 'TOMAWAC2'. In that case, the geometry files of TELEMAC-2D and TOMAWAC do not need to be the same anymore. It only works if MPI is available and if the two geometry files contain variables indicating nodes and weights to be used to interpolate for each node of the mesh.
For example, if the TOMAWAC geometry file contains the variables:

```
TELTOM01 = 1, 1, 2, 4
TELTOM02 = 2, 2, 3, 5
TELTOM03 = 3, 3, 4, 6
TELTOM01WTS = 0.33, 0., 0.33, 0.5
TELTOM02WTS = 0.33, 1., 0.33, 0.
TELTOM03WTS = 0.34, 0., 0.34, 0.5
```

Then to interpolate the values on the first point, TOMAWAC uses the points 1, 2, 3 of the TELEMAC-2D geometry file with the weights 0.33, 0.33, 0.34, for the second one: 1, 2, 3 with the weights 0., 1., 0., for the third one: 2, 3, 4 with the weights 0.33, 0.33, 0.34 and for the fourth one: 4, 5, 6 with the weights 0.5, 0., 0.5. Obviously for each triplet, the sum of weights has to be equal to 1.
An example can be found in the littoral SISYPHE example with the two geometry files geo_tom_tom2tel_diff.slf and geo_t2d_tel2tom_diff.slf.
In the future, this feature could be extended to the coupling of other modules.

## 14.9 Assigning a name to a point

During some types of processing, for example a Fourier series analysis (see 14.10), 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 3,489 corresponds to the port of Saint-Malo and point 56,229 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'
```

## 14.10 Fourier analysis

TELEMAC-2D allows the user to analyze 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 feature 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 will stop with an error message.

## 14.11 Checking the mesh (CHECKMESH)

The **CHECKMESH** subroutine of the BIEF library is available to look for errors in the mesh, e.g. superimposed points . . . The keyword `CHECKING THE MESH` (default value = NO) should be activated to YES to call this subroutine.
This option works in both sequential and parallel modes.

# 15. 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 (or multi-core) 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 (in folder /sources/utils/parallel).
Informations 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 (in a parallel version same as 1),
- 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.

The parallel version uses a domain decomposition method to split/merge the mesh for each processor. Those extra steps are done using PARTEL/GRETEL and are launched by the Python execution scripts.
Note that the number of processors/cores can be given as an argument for the launching command and not as a hard-coded keyword in the `STEERING FILE` (e.g. telemac2d.py –ncsize=4 cas.txt will run TELEMAC-2D on 4 processors).

# 16. Recommendations

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

## 16.1 Mesh

Some precautions need to be taken when building the mesh. The following list should help, but it is not exhaustive of course.

- 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, the results will be of poor quality. In this case, it is possible to build the mesh of the low-water bed using regular gridding available in most of mesh generators,
- 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 often 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.

## 16.2 Initial conditions

The technique most commonly used for maritime domains subject to tidal effects is to initialize the free surface with a value corresponding to high tide and the velocities with zero, and then gradually empty the domain. Since the tidal solutions coming form Oregon State University (like TPXO global solution), it has been possible to use them to initialize the computation. It may initialize both water depth and horizontal velocity components.
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 initialized 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 **USER_CONDIN_H** subroutine, to initialize an elevation with a tilted plane (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.

## 16.3 Numerical parameter definition

### 16.3.1 Type of advection

Taking into account the recent improvements of TELEMAC-2D in this field, 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 free 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
```

### 16.3.2 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 optimize 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`) 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.

## 16.4 Special types of programming

### 16.4.1 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 only done once. The list of operations is as follows:

- Reading of the geometry,
- Bottom correction with **USER_CORFON** subroutine.

If a computation is being continued, the bottom from the previous computation results file is used, if there is one. Any change of the **USER_CORFON** subroutine for a continued computation will therefore be inoperative if the bottom topography is saved in the results file, even if **USER_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 free surface,
- Modify the **USER_CORFON** subroutine,
- Continue the computation. TELEMAC-2D will then use the new bottom topography and as it only finds the free surface in the results of the previous computation, it will recalculate the new water depth as being the old free surface minus the new bottom topography.

## 16.5 Tidal flats

The following explanations concern the Finite Elements option. In Finite Volumes options (see keyword `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 in removing from the computation the dry elements. This option cannot be used in parallel computations. With this option, the keyword `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 keyword `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 behavior 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”, by segment,
- 3: ”Flux control” ERIA, by triangular element.

The last two treatments mean that some fluxes between points or elements may be limited to avoid negative depths. If using options 2 or 3 with tidal flats, it is mandatory to set the following keywords:

- MASS-LUMPING ON H = 1.,
- CONTINUITY CORRECTION = YES,
- SUPG OPTION for water depth = 0 (no SUPG upwinding on depth).

When using option 1, it is possible to set 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 keyword 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 keyword OPTION FOR THE DIFFUSION OF VELOCITIES should normally be 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.

# 17. API

Information on the TELEMAC-2D API can be found in the telapy user documentation.

# A. Launching the computation

A computation is launched through the **telemac2d.py** command. That command activates the execution of a script which is common to all the computation modules in the TELEMAC system. Depending on the platform, some options may be unavailable.
The syntaxes in that command are as follows:

```
telemac2d.py [cas]    [--options]
```

- cas: name of the steering file,
- --ncsize=NCSIZE: specifies the number of processors forced in parallel mode, default = the number defined in the steering file,
- -c CONFIGNAME or –configname=CONFIGNAME: specifies the configuration name, default is randomly found in the configuration file,
- -f CONFIGFILE, –configfile=CONFIGFILE: specifies the configuration file, default = systel.cfg,
- -s, –sortiefile: specifies whether there is a sortie file, default is no,
- -t or –tmpdirectory: the temporary work directory is not destroyed on completion of computation.

By default, the procedure runs the computation in an interactive mode and displays the control listing on the monitor.
The operations performed by that script are as follows:

- Creation of a temporary directory (**name_cas_YYYY-MM-DD_HHhMMminSSs**),
- Duplication of the dictionary and the input files into that directory,
- Execution of the DAMOCLES software in order to determine the work file names,
- Creation of the computation launching script,
- Allocation of the files,
- Compilation of the FORTRAN file and link editing (as required),

- Launching of the computation,
- Retrieval of the results files and destruction of the temporary directory.

The procedure takes place with slight differences according to the selected options.
The detailed description of that procedure can be obtained through the help command:

```
telemac2d.py --help
telemac2d.py -h
```

# B. List of user subroutines

Even though all subroutines can be modified by the user, some subroutines have been specifically designed to define complex simulation parameters Here is a list of subroutines that can be included in the FORTRAN file and modified by the user:

| | |
|---|---|
| BIEF_VALIDA | Validation of a computation |
| CORVIS | Modification of viscosities |
| DRAGFO | Definition of vertical structures and adding drag force of vertical structures in the momentum equation |
| FLUXPR_TELEMAC2D | Management of control sections |
| LATITU | Computation of variables depending on latitude |
| METEO and METEO_TELEMAC | Atmospheric conditions (wind, pressure, etc.) |
| PROPIN_TELEMAC2D | Change of the type of boundary conditions |
| TR | Imposition of a time-dependent boundary tracer value (function) |
| USER_BORD | Imposition of particular boundary conditions |
| USER_CONDIN | Imposition of particular initial conditions |
| USER_CONDIN_H | Management of the initial conditions for water depth |
| USER_CONDIN_TRAC | Management of the initial conditions for tracer(s) |
| USER_CONDIN_UV | Management of the initial conditions for velocity components |
| USER_CORFON | Modification of bottom elevations |
| USER_CORPOR | Modification of porosity |
| USER_CORRXY | Modification of mesh coordinates |
| USER_CORSTR | Space-dependent friction coefficient |
| USER_DEBSCE | Time-dependent tracer source flow rates (function) |
| USER_DEF_ZONES | Definition of zones |
| USER_FLOT | Management of drogues |
| USER_LAGRAN | Lagrangian drifts |
| USER_MASKOB | Masking of elements |
| USER_MESURES | Reading of measurement data |
| USER_NOMVAR_TELEMAC2D | Definition of names of additional variables |
| USER_PRERES_TELEMAC2D | Computation of additional variables |
| USER_Q | Imposition of a time-dependent boundary flowrate (function) |
| USER_SL | Imposition of a time-dependent boundary free surface elevation (function) |

| | |
|---|---|
| USER_SOURCE_TELEMAC2D | Redefinition of the characteristics of the sources without steering file |
| USER_STRCHE | Space-dependent friction coefficient |
| USER_TRSCE | Imposition of time-dependent tracer values at the sources (function) |
| USER_UTIMP_TELEMAC2D | Additional variable writing |
| USER_VIT | Imposition of a time-dependent boundary velocity (function) |
| USER_VUSCE | Variable velocity along $x$ of a source (function) |
| USER_VVSCE | Variable velocity along $y$ of a source (function) |

# C. The SELAFIN format

Note: for unclear historical reasons this format is also sometimes called SERAFIN
This is a binary file.
This format can be 'SELAFIN ', for single precision storage, or 'SELAFIND' for double precision storage. Double precision storage can be used for cleaner restarts, but may not be understood by all post-processors.
All string in the SELAFIN file must be utf-8 encoded (See for `https://en.wikipedia.org/wiki/UTF-8` for the exact list).
The records are listed below. Records are given in the FORTRAN sense. It means that every record corresponds to a FORTRAN WRITE:
1 record containing the title of the study (80 characters), The last 8 characters must contain the format of the file (SELAFIN or SELAFIND)
1 record containing the two integers NBV(1) and NBV(2) (NBV(1) the number of variables, NBV(2) with the value of 0),
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 4 are currently being used).
If IPARAM (3) is not 0: the value corresponds to the x-coordinate of the origin in the mesh
If IPARAM (4) is not 0: the value corresponds to the y-coordinate of the origin in the mesh
These coordinates in metres may be used by post-processors to retrieve geo-referenced coordinates, while the coordinates of the mesh are relative to keep more digits.
If IPARAM (7) is not 0: the value corresponds to the number of planes on the vertical (in prisms.)
If IPARAM (8) is not 0: the value corresponds to the number of boundary points (in parallel).
If IPARAM (9) is not 0: the value corresponds to the number of interface points (in parallel).
if IPARAM (10) = 1: a record containing the computation starting date in 6 integers: year, month, day, hour, minute, second
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. Beware: in TELEMAC-2D, the dimensions of this array are (NELEM,NDP)),
1 record containing table IPOBO (integer array of dimension NPOIN); the value is 0 for an internal point, and gives the numbering of boundary points for the others. This array is never used (its data can be retrieved by another way). In parallel the table KNOLG is given instead, keeping track of the global numbers of points in the original mesh.

1 record containing table X (real array of dimension NPOIN containing the abscissas 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 arrays for each variable at time T.

# D. Generating output files for DELWAQ

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

```
BOTTOM SURFACES DELWAQ FILE
DELWAQ PRINTOUT PERIOD
DELWAQ STEERING FILE
DIFFUSIVITY DELWAQ FILE
DIFFUSIVITY FOR DELWAQ
EXCHANGE AREAS DELWAQ FILE
EXCHANGES 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
VERTICAL FLUXES DELWAQ FILE
VOLUMES DELWAQ FILE
```

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

# E. Defining friction by domains

When a complex definition of the friction has to be used for a computation, this option can be chosen, 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 keyword `FRICTION DATA` = YES (default = NO) 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 friction ID (code number), which represents a friction domain, has to be given to each node. The nodes with the same friction ID will use the same friction parameters.
This allocation is done thanks to the **FRICTION_USER** user subroutine. 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 friction IDs is already generated (for example with the Janet software from the SmileConsult). This file is called `ZONES FILE` and will be partitioned in case of parallelism.
**II – Friction parameters**
The frictions parameters of each friction domain are defined in the `FRICTION DATA FILE`. In this file we find, for each friction ID of friction domain:

- a friction law for the bottom and their parameters,
- a friction law for the vegetation and the parameters for vegetation (only if the option is used).

Example of friction data file:

| *Zone | Bottom | | | Vegetation | | | |
|---|---|---|---|---|---|---|---|
| *no | TypeBo | Rbo | MdefBo | TypeBo | Par1 | Par2 | Par3 ... - Par15 |
| From 4 to 6 | NFRO | | | NULL | | | |
| 20 | NIKU | 0.10 | | BAPT | 1.0 | 0.04 | 10.0 |
| 27 | COWH | 0.13 | 0.02 | LIND | 1.0 | 5.0 | |
| END | | | | | | | |

- The first column defines the friction ID of the friction domain. Here, there are 3 lines with the friction IDs: 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 column 5 describes the law of the vegetation friction: in the example it is `NULL`, `BAPT` or `LIND`. The columns 6 to max 20 should contain the parameters dedicated to the chosen vegetation law in case of no vegetation (`NULL`) the columns should be empty.

- The last line of the file must have only the word END, (or FIN or ENDE).

In order to add a comment in the `FRICTION DATA FILE`, the line must begin with a star "*".

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

| Law | Number | 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 |
| Colebrook-White | 7 | COWH | Roughness coefficient Manning coefficient |

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

| Law | Number | Name for data file | Parameters used |
|---|---|---|---|
| No Vegetation Friction | 0 | NULL | No parameter |
| Jaervelae | 1 | JAER | Cdx, LAI, Uref, Vogel, Hp |
| Lindner & Pasche | 2 | LIND | D, sp |
| Whittaker et al | 3 | WHIT | Cd0, Ap0, EI, Vogel, sp, Hp |
| Baptist et al | 4 | BAPT | Cd, mD, Hp |
| Huthoff et al | 5 | HUTH | Cd, mD, Hp, sp |
| Van Velzen et al | 6 | VANV | Cd, mD, Hp |
| Luhar & Nepf | 7 | LUHE | Cd, Cv, a, Hp |
| Vastila & Jaervelae | 8 | VAST | Cdf, LAI, Ureff, Vogelf, CDs, SAI, Vogels, Urefs, Hp |

Explanation of the used parameters:

| | |
|---|---|
| Cd: | Vegetation bulk drag coefficient (Baptist) |
| Cdx: | Vegetation drag coefficient specie specific (Jaervelae) |
| Cd0: | Vegetation initial drag coefficient (WHittaker) |
| Cdf: | Foliage drag coefficient |
| Cds: | Stem drag coefficient |
| Lai: | Leaf area index |
| SAI: | Stem area index |
| Uref: | Reference velocity (lowest velocity used to determine the Vogel exponent |
| Ureff: | Foliage reference velocity |
| Urefs: | Stem reference velocity |
| Vogel: | Vogel exponent |
| Vogelf: | Foliage Vogel exponent |
| Vogels: | Stem Vogel exponent |
| Hp: | Plant height |
| D: | Vegetation diameter |
| sp: | Vegetation spacing |
| Ap0: | Initial projected area |
| EI: | Flexural rigidity |
| mD: | m: vegetation density (1/sp**2) D: vegetation diameter 0.5*LAI/Hp (Finnigan 2000) |
| Cv: | Friction coefficient on top of the vegetation layer, for emerged case Cv=0 |
| a: | Frontal area of vegetation per volume (=mD) |

**III – Steering file**

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

For the friction data file:

`FRICTION DATA` = YES.

`FRICTION DATA FILE` = 'name of the file where friction is given'.

For the vegetation friction (if used):

`VEGETATION FRICTION` = YES.

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

MAXIMUM NUMBER OF FRICTION DOMAINS = 80.
Link between nodes and friction IDs of friction domains is achieved with:
ZONES FILE = 'name of the file' or
the friction IDs can be given as variable FRIC_ID in the geometry file.
**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 French de... a... or German von... bis...).
The first friction ID of the domains and the last friction ID of the domains have to be set. All domains of friction with a friction ID 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 twice, 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)** and **TEST_FRICTION%ADR(I)%P%GNUM(2)** have the same value if a single friction domain is defined.
The link between TELEMAC-2D and the computation of the friction is done with the **FRICTION_CHOICE** subroutine. It is used in order to initialize the variables for the option FRICTION DATA 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 **FRICTION_READ** subroutine and the friction ID of each nodes are saved thanks to **FRICTION_USER** in the array **KFROPT%I**. With the subroutine **FRICTION_INIT**, the friction IDs 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 **FRICTION_CALC** subroutine 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**. Else, they take the same value and the loop on the node is done in the **FRICTION_ZONES** subroutine (the parameters used for each node can be different).
The choice of VEGETATION FRICTION is only possible together with FRICTION DATA.
Then one of the routines **FRICTION_BAPTIST**, **FRICTION_HUTHOFF**, **FRICTION_JAERVELAE**,

**FRICTION_LINDNER**, **FRICTION_LUHARNEPF**, **FRICTION_VANVELZEN**, **FRICTION_VASTILA**, **FRICTION_WHITTAKER** calculates the additional roughness due to 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 can be in single precision if `GEOMETRY FILE FORMAT` is not set to SERAFIND. 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 both methods, the difference may come from the difference between single and double precision.

[1] Notice of revised emergency action plan guidelines. Technical report, USA Federal Regulatory Commission, 1988.

[2] R. ATA. Telemac-2d new finite volume schemes for shallow water equations with source terms on 2d unstructured grids . In *Proc. 19th TELEMAC-MASCARET User Conference*, pages 93–98, 2012.

[3] E. Audusse and M.-O. Bristeau. A well-balanced positivity preserving second-order scheme for Shallow Water flows on unstructured meshes. *J. Comput. Phys.*, 206(1):311–333, 2005.

[4] E Audusse, F Bouchut, M-O Bristeau, R Klein, and B Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.

[5] Emmanuel Audusse, Marie-Odile Bristeau, and Benoit Perthame. Kinetic schemes for saint-venant equations with source terms on unstructured grids. 2000.

[6] O'Connor B. and D. Yoo. Mean bed friction combined wave/current flow. *Coastal Engineering*, 12(1):1–21, 1988.

[7] S. Bennani. Analyse des incertitudes de la méthode normée lors de la rupture des barrages en remblai en cas de surverse. Master's thesis, Ecole Polytechnique de Montréal, 2016.

[8] G.L. Brunner. Hec ras - river analysis system user's manual, version 3.1. Technical Report CPD-68, U.S. Army Corps of Engineers, Hydrologic Engineering Center, Davis, CA, 2002.

[9] Dorfmann C. and Zenz G. The depth-averaged mixing length turbulence model for telemac-2d. In *XXIIIrd TELEMAC-MASCARET User Conference*, Paris, France, 2016.

[10] PHAM C.-T., BOURBAN S., DURAND N., and TURNBULL M. Méthodologie pour la simulation de la marée avec la version 6.2 de telemac-2d et telemac-3d. Technical Report H-P74-2012-02534-FR, EDF R&D-LNHE, 2012.

[11] Guoxian Chen and Sebastian Noelle. A new hydrostatic reconstruction scheme based on subcell reconstructions. *SIAM Journal on Numerical Analysis*, 55(2):758–784, 2017.

[12] Ven Te Chow, David R Maidment, and W Larry. Mays. 1988. *Applied Hydrology*, pages 110–117, 1988.

[13] A. Curran, K. M. De Bruijn, and M. Kok. Influence of water level duration on dike breach triggering, focusing on system behaviour hazard analyses in lowland rivers. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 14(1):26–40, 2020. doi: 10.1080/17499518.2018.1542498.

[14] F. Folke, R. Kopmann, and M. Attieh. Comparison of different vegetation models using TELEMAC-2D. In *Proceedings of the 26th TELEMAC-MASCARET User Conference 2019, 15th-17th October 2019, Toulouse–France*, 2019.

[15] D.L. Fread and T.E. Harbaugh. Transient hydraulic simulation of breached earth dams. *Journal of Hydraulic Division*, 99(1):139–154, 1973.

[16] D.C. Froehlich. Embankment dam breach parameters and their uncertainties. *Journal of Hydraulic Engineering*, 134(12):1708–1721, 2008.

[17] Cédric Goeury. *Modélisation du transport des nappes d'hydrocarbures en zone continentale et estuarienne*. PhD thesis, 2012. URL `http://www.theses.fr/2012PEST1131`. Thèse de doctorat dirigée par Hervouet, Jean-Michel Mécanique des fluides Paris Est 2012.

[18] J-M. HERVOUET. *Hydrodynamics of free surface flows. Modelling with the finite element method.* John Wiley & Sons, Ltd, Paris, 2007.

[19] SMAGORINSKY J. General simulation experiments with the primitive equations. *Monthly Weather Review*, 91(3):99–164, March 1963.

[20] A. Joly. *Modelling of the transport of algae in a coastal environment using a stochastic method*. PhD thesis, 2011. URL `http://www.theses.fr/2011PEST1088`. Thèse de doctorat dirigée par Violeau, Damien Mécanique des fluides Paris Est 2011.

[21] M. Morris, M.A.A.M. Hassan, A. Kortenhaus, and M. Visser. Breaching processes: a state of the art review. Technical Report FLOODsite Project Report, HR Wallingford, Wallingford, U. K., 2009.

[22] M.W. Morris, A. Kortenhaus, and P. Visser. Modelling breach initiation and growth. Technical Report FLOODsite Project Report, T06-08-02 (Revision 1_9_P01), HR Wallingford, Wallingford, U. K., 2009.

[23] United States Bureau of Reclamation. Downstream hazard classification guidelines. Technical Report ACER Technical Memorandum No. 11, 56 pp., United States Department of the Interior, 1988.

[24] Tasos C Papanastasiou. Flows of materials with yield. *Journal of Rheology*, 31(5):385–404, 1987.

[25] Dieter Rickenmann. *Bedload transport capacity of slurry flows at steep slopes*. PhD thesis, ETH Zurich, 1990.

[26] I. Rifai, S. Erpicum, P. Archambeau, D. Violeau, M. Pirotton, K. El Kadi Abderrezzak, and B. Dewals. Overtopping induced failure of noncohesive, homogeneous fluvial dikes. *Water Resources Research*, 53:3373–3386, 2017.

[27] I. Rifai, K. El Kadi Abderrezzak, D. Violeau, S. Erpicum, P. Archambeau, , M. . Pirotton, and B. Dewals. Floodplain backwater effect on overtopping induced fluvial dike failure. *Water Resources Research*, 54(11):9060–9073, 2018.

[28] I. Rifai, K. El Kadi Abderrezzak, S. Erpicum, P. Archambeau, D. Violeau, M. Pirotton, and B. Dewals. Flow and detailed 3d morphodynamic data from laboratory experiments of fluvial dike breaching. *Scientific Data. Nature/Scientific Data*, page 6:53, 2019. doi: 10.1038/s41597-019-0057-y.

[29] P. Risher and S. Gibson. Applying mechanistic dam breach models to historic levee breaches. In *FLOODrisk 2016 - 3rd European Conference on Flood Risk Management*, 2016.

[30] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[31] Soil Conservation Service. National engineering handbook, section 4, hydrology, 1972.

[32] Richard Soulsby. *Dynamics of marine sands*. Thomas Telford Publishing, 1997.

[33] Eleuterio F Toro. The hll and hllc riemann solvers. In *Riemann solvers and numerical methods for fluid dynamics*, pages 315–344. Springer, 2009.

[34] M. van Damme. An analytical process-based approach to predicting breach width in levees constructed from dilatant soils. *Natural Hazards*, 101:59–85, 2020. doi: 10.1007/s11069-020-03862-8.

[35] H. Verheij. Time dependent breach development in cohesive material. Technical Report Internal Research Summuary Report, Delft Hydraulics Laboratory, 2002.

[36] H. Verheij and F.C.M. Van der Knaap. Modification breach growth model in his-om. Technical Report WL, Delft Hydraulics Laboratory, 2003.

[37] H. Verheij, P. Visser, and U. Förster. Analysis of chinese dike breaches along the yangtze river and tributaries. Technical Report Desk study, Deltares, 2009.

[38] J.L. von Thun and D.R. Gillette. Guidance on breach parameters. Technical Report Internal Technical Memorandum, 17 pp, United States Bureau of Reclamations, Denver, Colorado, 1990.

[39] T. Wahl. Prediction of embankment dam breach parameters: a literature review and needs assessment. Technical report, U.S., Department of the Interior, Bureau of Reclamation, 1998.

[40] T. Wahl. Evaluation of numerical models for simulating embankment dam erosion and breach processes. Technical Report DSO-2017-02, United States Bureau of Reclamation, 2017.

[41] Dongchen Wang and Pablo Tassi. Secondary flow corrections into the telemac-mascaret modelling system. In *Proceedings of the 21st TELEMAC-MASCARET User Conference 2014, 15th-17th October 2014, Grenoble–France*, pages 225–233, 2014.

[42] M. West, M. Morris, and M.A.A.M. Hassan. A guide to breach prediction. Technical Report Paper discussion, HR Wallingford Ltd, Wallingford, 2018.

[43] Donald E Woodward, Richard H Hawkins, Ruiyun Jiang, Allen T Hjelmfelt, Jr, Joseph A Van Mullem, and Quan D Quan. Runoff curve number method: examination of the initial abstraction ratio. In *World water & environmental resources congress 2003*, pages 1–10, 2003.

[44] L. Wu and H. Li. A simplified physically-based model for coastal dike and barrier breaching by overtopping flow and waves. *Coastal Engineering*, 130:1–13, 2017.

[45] Jean-Marie Zokagoa and Azzeddine Soulaïmani. Modeling of wetting–drying transitions in free surface flows over complex topographies. *Computer methods in applied mechanics and engineering*, 199(33-36):2281–2304, 2010.