Postel3D

User Manual

Otto Mattic

Version 7.2
December 29, 2016

Contents

1. Description

POSTEL-3D allows extracting 2D vertical or horizontal cross sections from the 3D result file. The resulting cross section files are readable with the different post-processing tools of the TELEMAC-MASCARET SYSTEM.
The files generated by POSTEL-3D are in SERAFIN single precision format.
A computation is launched through the command `postel3d.py [cas]` (cas: the name of the steering file).
POSTEL-3D is implemented like all the codes in the TELEMAC treatment chain.
Its execution is structured around the `STEERING FILE`, which is a priori the single file which the user will have to consult and amend. It gathers the names of all the files which define the computation to be carried out.
The input files are:

- the `3D RESULT FILE` as provided by TELEMAC-3D, in the TELEMAC-3D format,
- the ASCII `FORTRAN FILE` containing the amended subroutines. That file is optional.

The output files are:

- the `HORIZONTAL CROSS SECTION FILES`, in the SERAFIN format (can be run under any post-processing tool that can read SERAFIN files),
- the `VERTICAL CROSS SECTION FILES`, in the SERAFIN format (can be run under any post-processing tool that can read SERAFIN files).

2. The horizontal cross sections

The horizontal cross sections are stored into binaries files in the SERAFIN format, on the basis of one file per cross section. The name of each such file consists in a common radical, which is given by the keyword `HORIZONTAL CROSS SECTION FILE`, followed by an extension specifying the cross section number.
A horizontal cross section is not necessarily horizontal, i.e. parallel to the level Z = 0. It may also be in the form of a 2D level of the TELEMAC-3D computation, with a possible vertical offset.
The mesh on which each of these cross sections is based is the 2D mesh which was used for conducting the TELEMAC-3D computation.
The horizontal cross sections are defined by means of the keywords:

- `NUMBER OF HORIZONTAL CROSS SECTIONS` specifies the number of cross sections to be made,
- `REFERENCE LEVEL FOR EACH HORIZONTAL CROSS SECTION` specifies the TELEMAC-3D vertical level from which the cross section shape is defined,
- `ELEVATION FROM REFERENCE LEVEL` specifies the vertical distance from the reference level at which the cross section has to be made.

The keyword `REFERENCE LEVEL` assumes a value ranging from 0 and NPLAN, NPLAN is the number of levels selected for the TELEMAC-3D computation which made it possible to prepare the 3D result file read. If the value is between 1 and NPLAN, the reference level is the relevant mesh level which is liable to move in time; if the value is zero, then the reference level is the level Z = 0. The cross section level is then inferred from the reference level through a mere vertical translation the amount of which is set up by the keyword `ELEVATION FROM REFERENCE LEVEL`.
In doing so, there is a slight difficulty because of the choice the user has to define a cross section is through two parameters: `REFERENCE LEVEL FOR EACH HORIZONTAL CROSS SECTION` and `ELEVATION FROM REFERENCE LEVEL`. That only defines the cross section level, and in such a case, two problematic scenarios can occur:

- either that level locally occurs above the free surface,
- or it occurs locally under the bottom.

We have therefore adopted another approach which consists in performing a linear extrapolation at these points from the closest 2 values, always occurring vertically above that point. For those points located above the surface, that extrapolation is then done from the values computed at the levels NPLAN-1 and NPLAN whereas the points located below the bottom, the extrapolation is computed from the values at levels 1 and 2.

Though the result of that extrapolation generally gives a value which is realistic, since it is not far from the values found in the domain, the user shall be made aware of the fact that such a result has no physical meaning and that it does not occur on the result planes.

In order to indicate the locations of these nodes, an INDICATEUR_DOM variable is provided for the user in the cross section file. When that variable is negative, that means the points are outside the domain. By inserting a coloured (e.g. white) surface of the INDICATEUR_DOM variable with a $]-\infty,0]$ threshold, the user can then mask the out-of-domain areas that have meaningless values.

As regards the VITESSE_U and VITESSE_V variables, we have preferred to preset these variables to 0 at the points located outside the domain. That treatment is more suitable for a vectorial plot.

3. The vertical cross sections

A vertical cross section can be defined in a 2D mesh as a sequence of linked points making up a pecked line consisting of segments. That pecked line is vertically extended from the surface down to the bottom. The minimum point number is 2 (1 segment) and the maximum number is 9 (8 segments).
The vertical cross sections are defined by means of the keywords:

- `NUMBER OF VERTICAL CROSS SECTIONS` specifying the number of cross sections to be made. That number cannot be in excess of 9. If over 9 vertical cross sections are desired, then several software executions should be planned,
- `NUMBER OF NODES FOR VERTICAL CROSS SECTION DISCRETIZATION` setting the number of interpolation points in the horizontal direction of the cross section. The points are evenly spaced. That number should be in excess of 2,
- `ABSCISSAE OF THE VERTICES OF CROSS SECTION X` specifying the abscissa of each point along the pecked line making up the vertical cross section. That number should be higher than or equal to 2 and lower than or equal to 9. X specifies the cross section number ranging from 1 to 9.

  > **Warning:**
  > If X is higher than the `NUMBER OF VERTICAL CROSS SECTIONS`, the information in that keyword will merely not be treated,

- `ORDINATES OF THE VERTICES OF CROSS SECTION X` specifying the ordinate of each point along the pecked line making up the vertical cross section. That number should be higher than or equal to 2 and lower than or equal to 9. X specifies the cross section number ranging from 1 to 9.

  > **Warning:**
  > If X is higher than the `NUMBER OF VERTICAL CROSS SECTIONS`, the information in that keyword will merely not be treated.

The vertical cross sections are stored into binary files in the SERAFIN format, on the basis of one file per cross section and per recorded time step. The name of each such file consists of

a common radical, as given by the keyword `VERTICAL CROSS SECTION FILE`, followed by an extension specifying the number of the cross section, then by an extension specifying the number of the recorded time step. That increased number of files is necessary because the meshes are distorted in time, and so are these cross sections, due to the free surface motions.
The horizontal velocity components are given in a cross section-related co-ordinate system provided for directly drawing the projection of the velocity vector onto the cross section level. The components in the new co-ordinate system are known as:

- VITESSE_UT: tangential component,
- VITESSE_UN: normal component.

The TELEMAC-3D computational domains often include a vertical scale which is much lower than the horizontal scale. The vertical scale can be distorted by a multiplicative factor so that RUBENS or any other post-processing tool, can output a clearer display. That can be done using the keyword `DISTORTION BETWEEN VERTICAL AND HORIZONTAL`.

Warning:
If one uses a distortion factor, the vertical velocities will themselves be multiplied by that factor. That is necessary to properly represent the velocity vector directions.

Lastly, note that the starting point is located to the left of the cross section and the ending point to, the right of it i.e. the length is defined from the bottom to the free surface.

[1] JOLY A., GOEURY C., and HERVOUET J.-M. Adding a particle transport module to telemac-2d with applications to algae blooms and oil spills. Technical Report H-P74-2013-02317-EN, EDF R&D-LNHE, 2013.

[2] AUTHOR. Title. *Journal de Mickey*, 666.

[3] 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.

[4] Sampath Kumar Gurram, Karam S. Karki, and Willi H. Hager. Subcritical junction flow. *Journal of Hydraulic Engineering*, 123(5):447–455, may 1997.

[5] TSANIS I. Simulation of wind-induced water currents. *Journal of hydraulic Engineering*, 115(8):1113–1134, 1989.

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

[7] HERVOUET J.-M. *Méthodes itératives pour la solution des systèmes matriciels.* Rapport EDF HE43/93.049/A, 1996.

[8] HERVOUET J.-M. *Hydrodynamics of Free Surface Flows. Modelling with the finite element method.* Wiley, 2007.

[9] HERVOUET J.-M. Guide to programming in the telemac system version 6.0. Technical Report H-P74-2009-00801-EN, EDF R&D-LNHE, 2009.

[10] JANIN J.-M., HERVOUET J.-M., and MOULIN C. *A positive conservative scheme for scalar advection using the M.U.R.D technique in 3D free-surface flow problems*. XIth International Conference on Computional methods in water resources, 1996.

[11] GAUTHIER M. and QUETIN B. Modèles mathématiques de calcul des écoulements induits par le vent. In *17e congrès de l'AIRH*, Baden-Baden, August 1977.

[12] METCALF M. and REID J. *Fortran 90 explained*. Oxford Science Publications, 1990.