Conference Paper, Published Version

Joly, Antoine; Pham, Chi-Tuan; Andreewsky, Marc; Saviot, Sylvain; Fillot, Lauriane

Using the DRAGFO subroutine to model Tidal Energy Converters in Telemac-2D

Zur Verfügung gestellt in Kooperation mit/Provided in Cooperation with:
TELEMAC-MASCARET Core Group

Verfügbar unter/Available at: https://hdl.handle.net/20.500.11970/104330

Vorgeschlagene Zitierweise/Suggested citation:
Joly, Antoine; Pham, Chi-Tuan; Andreewsky, Marc; Saviot, Sylvain; Fillot, Lauriane (2015): Using the DRAGFO subroutine to model Tidal Energy Converters in Telemac-2D. In: Moulinec, Charles; Emerson, David (Hg.): Proceedings of the XXII TELEMAC-MASCARET Technical User Conference October 15-16, 2031. Warrington: STFC Daresbury Laboratory. S. 182-189.

Using the DRAGFO subroutine to model Tidal Energy Converters in Telemac-2D

Antoine Joly
EDF R&D LNHE / LHSV
6 quai Watier, 78401 Chatou, France
Email: antoine.joly@edf.fr

Chi-Tuân Pham, Marc Andreewsky, Sylvain Saviot & Lauriane Fillot
EDF R&D LNHE
6 quai Watier, 78401 Chatou, France

***Abstract*—In the past few years the research on Tidal Energy Converters (TECs), or Tidal Turbines, has increased (see for example the work done at the EMEC[1]). As such a demand arose to assess the impact of potential large industrial TEC farms. This demand has also been growing in the Telemac User community, see for example Haverson *et al.* [2] or de Paula Kirinus *et al.* [1], but no clear methodology has been defined within the Telemac-Mascaret system.**

To model the hydrodynamic effects of TEC over large areas, two approaches are possible. One can either use the actuator disk method, where the influence of a TEC is modelled through a head drop [6], or it is possible to model the drag force induced by a TEC opposing the flow (this can be thought of as being similar to increasing the bed friction). This second approach has been successfully applied by EDF in Telemac-2D through the PerAWaT project[2] and its recommended implementation in Telemac-2D will be described in this article. Furthermore its implementation will be illustrated by reproducing the experiment conducted by the University of Manchester within the PerAWaT project [7].

I. Introduction

The technology used to extract tidal energy through turbines has made significant progress in recent years. As such, Tidal Energy Converters (TECs) are ready for test sites, and there is a growing need to model their hydrodynamics effects. Telemac-2D has been successfully used to model the hydrodynamic effects of such test sites scenarios, see for example [1], [2], [4]. However, no clear methodology has been defined to use Telemac-2D to model the influence of TECs on the hydrodynamics of a real site. This article is designated as guide to anyone wanting to model TECs using a tested method [4].

II. Modelling approach

To model the hydrodynamic effects of TEC over large areas, two approaches are possible. One can either use the actuator disk method, where the influence of a TEC is modelled through a head drop [6], or it is possible to model the drag force induced by a TEC opposing the flow (this can be thought of as being similar to increasing the ground friction). This second approach has been successfully applied in Telemac-2D [4] and its implementation in Telemac-2D will be described in this article.

As a reminder the drag force of a Tidal Energy Converter is defined through the following equation:

$$F_D = -\frac{1}{2}\pi R^2 \rho C_D U_r \left|U_r\right| \tag{1}$$

Where F_D is the drag force along the central axis of the TEC, ρ is the fluid density, C_D is a drag coefficient (usually given by the designer of the TEC, and checked by the manufacturer if possible), R is the radius of the TEC and U_r is a reference velocity along the central axis of the TEC ($U_r \left|U_r\right|$ is used instead of U_r^2 so that the direction of the flow is kept).

The reference velocity U_r is particularly important when modelling TECs with Telemac-2D, because when a TEC constructor calibrates the drag coefficient this velocity is usually taken at a position where the flow is not disturbed by the presence of the TEC (for example, upstream). Therefore using the velocity at the position of the TEC will usually generate the wrong drag force.

Furthermore the mechanical power extracted by a Tidal Energy Converter is given through the following formula:

$$P = \frac{1}{2}\pi R^2 C_P \rho U_r^2 \left|U_r\right| \tag{2}$$

Where P is the extracted power and C_P is power coefficient (usually given by the designer of the TEC, and also checked by the manufacturer if possible).

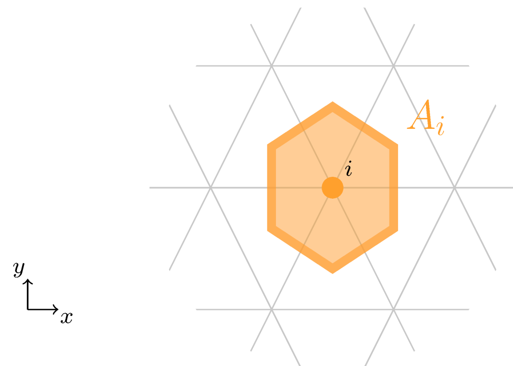
A. Applying the drag force as a source term

To apply (1) to the flow modelled with Telemac-2D, it needs to be slightly modified. This drag force will be applied as a stress spread out over an area representing the TEC. This will ensure a sufficient number of mesh nodes will be impacted by the turbine. The area and the mesh size at the location of the TEC will have to be chosen by the user. For more information one can refer to Joly *et al.* [5]. The stress τ can be found using (1), i.e.:

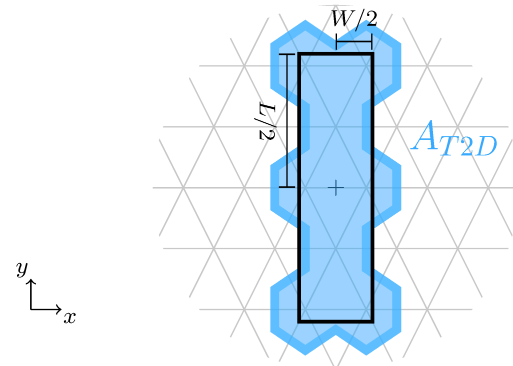
$$\begin{aligned}\tau_D =& \frac{F_D}{A}\\ =& -\frac{1}{2}\frac{\pi R^2}{A}\rho C_D U_r \left|U_r\right|\end{aligned} \tag{3}$$

Where A is the area over which the drag force will be applied.

[1] http://www.emec.org.uk/
[2] http://www.eti.co.uk/project/perawat/



(a) Area of a node i (A_i)



(b) Area occupied by a TEC in Telemac-2D (A_{T2D})

Fig. 1: Different areas used to define a TEC in a telemac-2D simulation.

This stress can be treated as a source term in the Shallow Water Equation, in the same way that friction stresses are taken into account [3]:

$$\begin{aligned} Q_{TEC} =& \frac{\tau_D}{\rho h} \\ =& -\frac{1}{2}\frac{\pi R^2}{Ah} C_D U_r\,|U_r| \end{aligned} \tag{4}$$

Where h is the water depth at the position of the TEC. Finally in Telemac-2D there is a possibility for this term to be implicited, which means that it will be multiplied by the local velocity later on. Therefore, to apply the drag force of a TEC to the flow within Telemac-2D it will be necessary to divide this source term by the local velocity, i.e.:

$$\begin{aligned} \texttt{FDRAG} =& \frac{Q_{TEC}}{U_{loc}} \\ =& -\frac{1}{2}\frac{\pi R^2}{Ah} C_D \frac{U_r\,|U_r|}{U_{loc}} \end{aligned} \tag{5}$$

Where `FDRAG` is the source term used in Telemac-2D to model the impact of a drag force on the flow (it will later be decomposed into a horizontal and vertical components `FUDRAG` and `FVDRAG`). Variable U_{loc} is the local velocity of a node upon which the drag force is applied.

B. Taking into account the geometrical parameters in a mesh

In Telemac-2D, a TEC is defined by an area of length L and a width W. However, since the drag force is applied to nodes in a 2D mesh, the effective area (known as A_{T2D}) is the sum of the area of the nodes inside the area defined by L and W. See figure 1 for more details on the calculations of the areas.

In Telemac-2D, a TEC is also defined by the orientation of its central axis, which will be referred to as the angle θ, see figure 2 for more details. Because of this, the drag force F_D will need to be decomposed into x and y components to be applied in Telemac-2D. The components of the drag force F_D are therefore given by:

$$F_{D,x} = -\frac{1}{2}\pi R^2 C_D U_r\,|U_r|\cos(\theta) \tag{6a}$$

$$F_{D,y} = \frac{1}{2}\pi R^2 C_D U_r\,|U_r|\sin(\theta) \tag{6b}$$

As mentioned previously, the drag coefficient is usually defined for a reference velocity which is typically taken at a point where the flow is undisturbed by the TEC. This point will be assumed to be upstream, and the user will need to define its distance from the centre of the TEC (noted as D_D).

The reference velocity U_r will be calculated from the velocity vector $\boldsymbol{U_{ref}}$ taken at a distance D_D upstream of the flow going through the centre of the TEC. It is therefore necessary to calculate an angle α, which is the angle of the direction of the flow at the centre of the TEC. It will therefore be the direction of the fluid velocity vector at the centre of the TEC $\boldsymbol{U_{TEC}}$. See figure 2 for more details.

The vector $\boldsymbol{U_{ref}}$ then needs to be projected along the central axis of the TEC (given by angle θ) to calculate the reference velocity U_r. The following equations will give the steps necessary to calculate U_r:

$$\alpha = -\,\texttt{ATAN2}\,(U_{TEC,y}, U_{TEC,x}) \tag{7a}$$

$$\begin{aligned} U_r =& U_{ref,x}(D_D, \pi+\alpha)\cos(\theta) \\ & - U_{ref,y}(D_D, \pi+\alpha)\sin(\theta) \end{aligned} \tag{7b}$$

Where `ATAN2` is a function available in most Fortran compiler which gives the angle in the appropriate quadrant, $U_{TEC,x}$ and $U_{TEC,y}$ are the horizontal and vertical components of vector $\boldsymbol{U_{TEC}}$ and $U_{ref,x}(D_D, \pi+\alpha)$ and $U_{ref,y}(D_D, \pi+\alpha)$

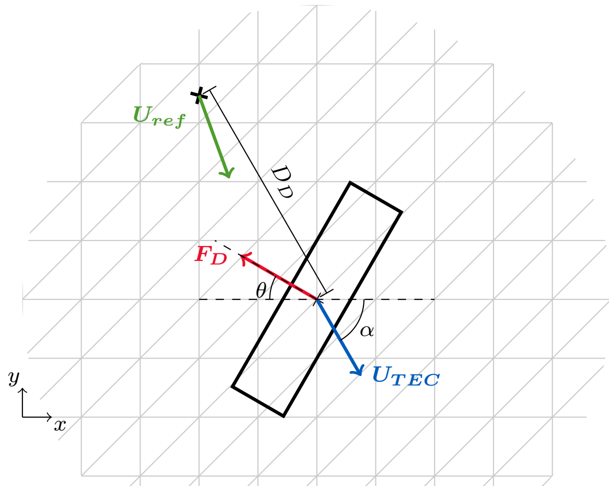


Fig. 2: Detailed geometrical parameters for a TEC in a 2D triangular mesh. Note that θ is positive in the clockwise direction and α is positive in the counter-clockwise direction.

are the horizontal and vertical components of vector $\boldsymbol{U_{ref}}$ taken at a distance D_D and an angle $\pi + \alpha$ of the TEC. In a triangular mesh it is therefore recommended to store the number of the elements that are at a distance D_D of a TEC so that U_r can be calculated fast.

C. Calculating the extracted power of the TEC

Since the extracted power by each TEC is calculated using the reference velocity U_r, equation (2) does not need to be applied to each node in a mesh. Therefore, once U_r is known for each TEC, see (7), equation (2) can be applied directly to find the extracted mechanical power of each TEC.

III. Summary of the system of equations that needs to be solved and of the parameters that need to be defined in Telemac-2D

For a TEC i and a node j in the mesh, equations (2), (5), (6) and (7) need to be solved and applied to a Telemac-2D simulation:

$$\alpha(i) = -\,\mathrm{ATAN2}\left[U_{TEC,y}(i), U_{TEC,x}(i)\right] \tag{8a}$$

$$U_r(i) = U_{ref,x}(i, D_D, \pi+\alpha)\cos[\theta(i)] - U_{ref,y}(i, D_D, \pi+\alpha)\sin[\theta(i)] \tag{8b}$$

$$A_{T2D}(i) = \sum_{j=\text{Nodes in TEC i}} A_j \tag{8c}$$

$$\mathrm{FUDRAG}(i,j) = -\frac{1}{2}\frac{\pi R(i)^2}{A_{T2D}(i)h_j} C_D(i)\frac{U_r(i)\,|U_r(i)|}{U_j}\cos[\theta(i)] \tag{8d}$$

$$\mathrm{FVDRAG}(i,j) = \frac{1}{2}\frac{\pi R(i)^2}{A_{T2D}(i)h_j} C_D(i)\frac{U_r(i)\,|U_r(i)|}{V_j}\sin[\theta(i)] \tag{8e}$$

$$P(i) = \frac{1}{2}\pi R(i)^2 C_P(i)\rho U_r(i)^2\,|U_r(i)| \tag{8f}$$

Where U_j and V_j are the horizontal and vertical components of the fluid velocity of a node j upon which the drag force will be applied.

To solve the system of equations (8) several variables will need to be defined by the user:

- R: Radius of the TEC
- C_D: Drag coefficient of the TEC
- θ: Orientation of the central axis of the TEC.
- L: Length of the 2D area covered by the TEC
- W: Width of the 2D area covered by the TEC
- D_D: Distance from the TEC for which the reference velocity will be taken
- C_P: Power coefficient of the TEC

Variables $\boldsymbol{U_{TEC}}$, $\boldsymbol{U_{ref}}$, $\boldsymbol{U_j}$ and h_j will need to be interpolated from the triangular mesh.

IV. Taking into account the influence of a TEC in Telemac-2D

A. Configuration in the steering file

Through the use of keywords in the steering file several parameters can be made easily available:

```
/------------------------------------------------------/
/                    OPTIONS FOR TECS
/------------------------------------------------------/
/ USING THE DRAGFO SUBROUTINE
VERTICAL STRUCTURES     : YES
FORTRAN FILE            : './t2d_dragfo_TEC.f'
/ FILE WITH POSITIONS AND NUMBER OF THE TEC
FORMATTED DATA FILE 2   : './TEC_positions.xyz'
/ RESULT FILE TO OUTPUT THE EXTRACTED POWER
FORMATTED RESULTS FILE : './TEC_power.txt'
WATER DENSITY           : 1025.
```

Note that the keyword for the water density is only necessary for coastal simulations as the default value is 1000 kg/m^{-3}. Furthermore, the `DRAGFO` subroutine will be modified in the user Fortran code (given by the keyword `FORTRAN FILE`) to take into account TEC through the system of equations (8).

To easily define the number, the positions and physical parameters of the TECs that one wants to model it is recommended to define an external formatted file. This file will be defined in the steering file using the keyword `FORMATTED DATA FILE 2`. It should have the following format:

```
N
X1 Y1 L1 W1 THETA1 R1 DD1 CD1 CP1
X2 Y2 L2 W2 THETA2 R2 DD2 CD2 CP2
X3 Y3 L3 W3 THETA3 R3 DD3 CD3 CP3
...
XN YN LN WN THETAN RN DDN CDN CPN
```

Where `N` is the number of TEC, `X1-XN` are the x-coordinates and `Y1-YN` are the y-coordinates of the centre of the area defining each TEC. `L1-LN`, `W1-WN`, etc. are the TEC parameters defined in section III.

Finally the extracted power output per TEC will be printed in time in a file given by the keyword `FORMATTED RESULTS FILE`.

B. Using the subroutine DRAGFO *to model the drag force*

To apply the system of equations 8 in Telemac-2D it is possible to use the subroutine DRAGFO. This subroutine is only called in Telemac-2D if the keyword VERTICAL STRUCTURES is set to YES in the steering file. This subroutine will then need to be modified in the user FORTRAN FILE. The recommended approach to model several individual turbines in a parallel implementation of Telemac-2D will now be described. It should be noted that the line numbering, when it appears, is that of a user Fortran file containing only the modifications recommended to the user subroutine DRAGFO.

1) New variables specific to TEC simulations: It is recommended to store the simulation parameters that define the TECs in a module as it is a clean way to store values that will be accessible at all time steps. Therefore at the beginning of the Fortran user file the following lines of code will need to be added.

```
@@ -1,6 +1,36 @@
+!---------------------------------------------------------
+      MODULE TEC_PARAM
+!---------------------------------------------------------
+       ! PARAMETERS READ IN THE FORMATTED DATA FILE
+       INTEGER:: NTEC
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: XTEC
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: YTEC
+       ! PARAMETERS DEFINNING THE TEC
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: THETA
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: HDL
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: HDW
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: RTEC
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: DD
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: CDTEC
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: CPTEC
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: VCUT
+       ! PARAMETERS USED TO DEFINE THE NODES WITHIN EACH TEC
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: AREA
+       INTEGER,DIMENSION(:),ALLOCATABLE:: NNODES
+       INTEGER,DIMENSION(:,:),ALLOCATABLE:: INODES
+       ! ELEMENT AND PROCESSOR OF TEC CENTRE
+       INTEGER,DIMENSION(:),ALLOCATABLE:: TECELEM
+       INTEGER,DIMENSION(:),ALLOCATABLE:: TECPID
+       ! PARAMETERS USED TO FIND THE FAR VELOCITY
+       INTEGER,DIMENSION(:,:),ALLOCATABLE:: DDELEM
+       INTEGER,DIMENSION(:,:),ALLOCATABLE:: DDPID
+       ! EXTRACTED POWER
+       DOUBLE PRECISION,DIMENSION(:),ALLOCATABLE:: PTEC
+       END MODULE
+!---------------------------------------------------------
 !---------------------------------------------------------
                        SUBROUTINE DRAGFO
 !---------------------------------------------------------
 !
     &(FUDRAG,FVDRAG)
 !
```

To understand the differences shown in the line of code:

- The lines starting with "@@" give indications of the line numbers before the change. The numbers after the "-" give the line number and number of lines of the hunk of text in the original file, the number after the "+" give the line number and number of lines of the hunk of text in the modified file
- The green lines starting with a "+" show additions to the code
- The red lines starting with a "-" show deleted lines of code
- The black lines and starting with a space show unchanged code

Furthermore the memory for these variables will need to be allocated, and then they will need to be defined.

In the subroutine DRAGFO the following line is added to allow the subroutine to access the variables stored in the module TEC_PARAM:

```
@@ -43,6 +73,7 @@
 !
        USE BIEF
        USE DECLARATIONS_TELEMAC2D
+       USE TEC_PARAM
 !
        IMPLICIT NONE
        INTEGER LNG,LU
```

Finally these new variables will be used in the code:

```
@@ -64,6 +95,27 @@
       EXTERNAL         P_DSUM
 !
 !------------------------------------------------------------
+! ALLOCATE THE EXTRA VARIABLES FOR TEC
+!------------------------------------------------------------
+      DOUBLE PRECISION PI, DTR, RTD, TWOPI
+      DOUBLE PRECISION HDLCOSTHETA, HDLSINTHETA
+      DOUBLE PRECISION HDWCOSTHETA, HDWSINTHETA
+      INTEGER IPOIN,IANGLE,INODE
+      DOUBLE PRECISION ALPHA,XDD,YDD
+      DOUBLE PRECISION UTECX,UTECY,UREFX,UREFY,UR
+      INTEGER N1,N2,N3
+      DOUBLE PRECISION DET1,DET2,DET3,SURDET
+      DOUBLE PRECISION HALFCDSCOSTHETA,HALFCDSSINTHETA
+      DOUBLE PRECISION HI,UI,VI
+!------------------------------------------------------------
+! ALLOCATE THE EXTRA VARIABLES TO WRITE THE POWER EXTRACTED
+!------------------------------------------------------------
+      INTEGER POWRES
+!
+      INTEGER          P_IMAX
+      EXTERNAL         P_IMAX
+!
+!------------------------------------------------------------
 !
 !     COMPUTES THE MASSE INTEGRALS
 !
```

2) Defining TEC simulation parameters: Firstly, a few constants will be defined:

```
@@ -74,3 +126,13 @@
       CALL CPSTVC(VN,FVDRAG)
       CALL OS('X=C     ',FUDRAG,FUDRAG,FUDRAG,0.D0)
       CALL OS('X=C     ',FVDRAG,FVDRAG,FVDRAG,0.D0)
+
+!------------------------------------------------------------
+! TEC EXAMPLE
+!------------------------------------------------------------
+!------------------------------------------------------------
+! DEFINE THE CONSTANTS USED TO CALCULATE ANGLES OF TEC
+!------------------------------------------------------------
+      PI = 4.D0*ATAN(1.D0)
+      DTR = PI/180.D0
+      RTD = 180.D0/PI
```

Furthermore several variables need to be calculated and stored during the first time step. To do so, the variables defined in the module TEC_PARAM need to have their memory allocated:

```
@@ -77,0 +139,39 @@
+!------------------------------------------------------------
+! ON THE FIRST TIME STEP STORE ALL THE TEC PARAMETERS IN
+! THE MODULE
+!------------------------------------------------------------
+! FILE 27 IS DEFINED IN THE STEERING FILE OF TELEMAC-2D
+!  WITH THE KEYWORD:
+! FORMATTED DATA FILE 2
+      IF(LT.EQ.1) THEN
+        ! READ THE NUMBER OF TEC
+        READ(27,*) NTEC
+        WRITE(LU,*) '=========================='
```

```
+         WRITE(LU,*) 'NTEC=',NTEC
+         ! ALLOCATING THE TEC PARAMETERS
+         WRITE(LU,*) 'ALLOCATING TEC PARAMETERS'
+         !
+         ALLOCATE(XTEC(NTEC))
+         ALLOCATE(YTEC(NTEC))
+         !
+         ALLOCATE(THETA(NTEC))
+         ALLOCATE(HDL(NTEC))
+         ALLOCATE(HDW(NTEC))
+         ALLOCATE(RTEC(NTEC))
+         ALLOCATE(DD(NTEC))
+         ALLOCATE(CDTEC(NTEC))
+         ALLOCATE(CPTEC(NTEC))
+         ALLOCATE(VCUT(NTEC))
+         !
+         ALLOCATE(AREA(NTEC))
+         ALLOCATE(NNODES(NTEC))
+         ALLOCATE(INODES(NTEC,1000))
+         !
+         ALLOCATE(TECELEM(NTEC))
+         ALLOCATE(TECPID(NTEC))
+         !
+         ALLOCATE(DDELEM(NTEC,-180:180))
+         ALLOCATE(DDPID(NTEC,-180:180))
+         !
+         ALLOCATE(PTEC(NTEC))
```

The TEC parameters will need to be defined. The positions will be read from the file defined in FORMATTED DATA FILE 2, the other parameters need to be defined by the user, these will be shown as <...>.

```
@@ -77,0 +177,22 @@
+         ! SET THE VALUES OF THE TEC PARAMETERS
+         WRITE(LU,*) '========================='
+         WRITE(LU,*) 'TEC PARAMETERS DEFINED IN FORTRAN FILE'
+         DO I = 1,NTEC
+           ! READING COORDINATES OF THE TURBINES CENTRES
+           READ(27,*) XTEC(I),YTEC(I),
+     &                HDL(I),HDW(I),THETA(I),
+     &                RTEC(I),DD(I),
+     &                CDTEC(I),CPTEC(I),
+           WRITE(LU,*) ' '
+           WRITE(LU,*) 'Turbine index : ',I
+           WRITE(LU,*) 'Position x,y of the turbine : ',
+     &                 XTEC(I),YTEC(I)
+           WRITE(LU,*) ' '
+           !
+           THETA(I) = THETA(I)*PI/180.D0
+           HDL(I)   = HDL(I)/2.D0+0.05D0
+           HDW(I)   = HDW(I)/2.D0+0.05D0
+           VCUT(I)  = 0.D0
+           !
+         END DO
+         WRITE(LU,*) '========================='
```

Then the specific simulation parameters will be calculated. Firstly, the area occupied by the TEC is calculated, A_{T2D} = AREA. While this done, the list of all nodes affected by each TEC will also be stored (in INODES) to increase computation speed later on.

```
@@ -77,0 +199,60 @@
+         ! STORING THE AREA OF THE TEC
+         DO I = 1,NTEC
+           ! WIDTH AND LENGTH ALONG X AND Y COORDINATES
+           HDLCOSTHETA = HDL(I)*COS(THETA(I))
+           HDLSINTHETA = HDL(I)*SIN(THETA(I))
+           HDWCOSTHETA = HDW(I)*COS(THETA(I))
+           HDWSINTHETA = HDW(I)*SIN(THETA(I))
+           !
+           NSOM=4
+           XSOM(1) = XTEC(I) + HDWCOSTHETA + HDLSINTHETA
+           YSOM(1) = YTEC(I) + HDWSINTHETA - HDLCOSTHETA
+           XSOM(2) = XTEC(I) + HDWCOSTHETA - HDLSINTHETA
+           YSOM(2) = YTEC(I) + HDWSINTHETA + HDLCOSTHETA
+           XSOM(3) = XTEC(I) - HDWCOSTHETA - HDLSINTHETA
+           YSOM(3) = YTEC(I) - HDWSINTHETA + HDLCOSTHETA
+           XSOM(4) = XTEC(I) - HDWCOSTHETA + HDLSINTHETA
+           YSOM(4) = YTEC(I) - HDWSINTHETA - HDLCOSTHETA
+           !
+           AREA(I)=0.D0
+           NNODES(I)=0
+           DO IPOIN=1,BIEF_NBPTS(11,MESH)
+             IF(INPOLY(X(IPOIN),Y(IPOIN),
+     &          XSOM,YSOM,NSOM))THEN
+               AREA(I) = AREA(I) + T1%R(IPOIN)
+               NNODES(I) = NNODES(I) + 1
+               IF(NNODES(I).GT.1000)THEN
+                 WRITE(LU,*)'DRAGFO: TOO MANY NODES IN TEC'
+                 WRITE(LU,*)'        MODIFY ALLOC OF INODES'
+                 CALL PLANTE(1)
+                 STOP
+               ENDIF
+               INODES(I,NNODES(I)) = IPOIN
+             ENDIF
+           ENDDO
+           ! QUASI-BUBBLE POINTS
+           IF(FU%ELM.EQ.12) THEN
+             DO IELEM = 1 , NELEM
+               I4=IKLE%I(IELEM+3*NELMAX)
+               X4=(X(IKLE%I(IELEM         ))+
+     &             X(IKLE%I(IELEM+  NELMAX))+
+     &             X(IKLE%I(IELEM+2*NELMAX)))/3.D0
+               Y4=(Y(IKLE%I(IELEM         ))+
+     &             Y(IKLE%I(IELEM+  NELMAX))+
+     &             Y(IKLE%I(IELEM+2*NELMAX)))/3.D0
+               IF(INPOLY(X4,Y4,XSOM,YSOM,NSOM)) THEN
+                 AREA(I) = AREA(I) + T1%R(I4)
+                 NNODES(I) = NNODES(I) + 1
+                 IF(NNODES(I).GT.1000)THEN
+                   WRITE(LU,*)'DRAGFO:TOO MANY NODES IN TEC'
+                   WRITE(LU,*)'       MODIFY ALLOC OF INODES'
+                   CALL PLANTE(1)
+                   STOP
+                 ENDIF
+                 INODES(I,NNODES(I)) = -IELEM
+               ENDIF
+             ENDDO
+           ENDIF
+           ! IN PARALLEL THE AREA MAY BE SPLIT
+           ! INTO SEVERAL SUB-DOMAINS
+           IF(NCSIZE.GT.0) AREA(I)=P_DSUM(AREA(I))
```

The second step stores all the processor and element number inside which the centre of the TEC is located as well as all the number and processors for all elements located at a distance D_D, for all angle between -180 and 180 degrees.

```
@@ -77,0 +260,74 @@
+!
+         ! LOOK FOR THE ELEM NUM FOR ALL ANGLES OF ALPHA
+         DO IANGLE=-180,180
+           XDD = XTEC(I) - DD(I)*COS(DBLE(IANGLE)*DTR)
+           YDD = YTEC(I) + DD(I)*SIN(DBLE(IANGLE)*DTR)
+           ! LOOK IN LOCAL MESH IF ELEMENT IS IN MESH
+           TECELEM(I)=0
+           TECPID(I)=0
+           DDELEM(I,IANGLE)=0
+           DDPID(I,IANGLE)=0
+           DO IELEM=1,NELEM
+             ! GET THE VERTICES
+             N1=IKLE%I(IELEM)
+             N2=IKLE%I(NELEM+IELEM)
+             N3=IKLE%I(2*NELEM+IELEM)
+             ! FIND THE ELEM AND PROC NUM OF THE TEC
+             ! FOR ALL ANGLES
+             DET1=(X(N3)-X(N2))*(YTEC(I)-Y(N2))
+     &            -(XTEC(I)-X(N2))*(Y(N3)-Y(N2))
+             DET2=(X(N1)-X(N3))*(YTEC(I)-Y(N3))
+     &            -(XTEC(I)-X(N3))*(Y(N1)-Y(N3))
+             DET3=(X(N2)-X(N1))*(YTEC(I)-Y(N1))
+     &            -(XTEC(I)-X(N1))*(Y(N2)-Y(N1))
+             IF((DET1.GE.0.D0).AND.
+     &          (DET2.GE.0.D0).AND.
+     &          (DET3.GE.0.D0)) THEN
+               TECELEM(I)=IELEM
+               IF(NCSIZE.GT.0) TECPID(I)=IPID
+             END IF
+             ! FIND THE ELEM AND PROC NUM AT DD
+             DET1=(X(N3)-X(N2))*(YDD-Y(N2))-
+     &            (XDD-X(N2))*(Y(N3)-Y(N2))
+             DET2=(X(N1)-X(N3))*(YDD-Y(N3))
```

```
+     &              -(XDD-X(N3))*(Y(N1)-Y(N3))
+             DET3=(X(N2)-X(N1))*(YDD-Y(N1))
+     &              -(XDD-X(N1))*(Y(N2)-Y(N1))
+             IF((DET1.GE.0.D0).AND.
+     &           (DET2.GE.0.D0).AND.
+     &           (DET3.GE.0.D0)) THEN
+               DDELEM(I,IANGLE)=IELEM
+               IF(NCSIZE.GT.0) DDPID(I,IANGLE)=IPID
+             END IF
+           END DO
+           !
+           IF(NCSIZE.GT.0)THEN
+             TECPID(I)=P_IMAX(TECPID(I))
+             TECELEM(I)=P_IMAX(TECELEM(I))
+             IF(IPID.NE.TECPID(I))THEN
+               TECELEM(I)=-TECELEM(I)
+             END IF
+             !
+             DDPID(I,IANGLE)=P_IMAX(DDPID(I,IANGLE))
+             DDELEM(I,IANGLE)=P_IMAX(DDELEM(I,IANGLE))
+             IF(IPID.NE.DDPID(I,IANGLE))THEN
+               DDELEM(I,IANGLE)=-DDELEM(I,IANGLE)
+             END IF
+           ENDIF
+           !
+           IF(TECELEM(I).EQ.0)THEN
+             WRITE(LU,*) 'DRAGFO:POSITION FOR TEC',I
+             WRITE(LU,*) '   IS OUTSIDE OF THE DOMAIN'
+             CALL PLANTE(1)
+             STOP
+           END IF
+           !
+           IF(DDELEM(I,IANGLE).EQ.0)THEN
+             WRITE(LU,*)'DRAGFO:POSITION FOR TEC',I
+             WRITE(LU,*)'IS TOO CLOSE TO EDGE OF DOMAIN'
+             CALL PLANTE(1)
+             STOP
+           END IF
+         END DO
+       END DO
+     END IF
```

Note that an error message will appear if the TEC is placed too close to the edge of the domain.

3) Applying the drag force: To apply the drag force a loop will be done on all TECs and on all nodes present in a TEC. The first step is to use the velocity at the centre of the TEC to find the angle of the flow α.

```
@@ -77,0 +333,34 @@
+!
+!-----------------------------------------------------------
+! APPLY THE DRAG FORCE OF THE TEC
+!-----------------------------------------------------------
+     DO I = 1,NTEC
+       UTECX=0.D0
+       UTECY=0.D0
+       PTEC(I) = 0.D0
+       IF(IPID.EQ.TECPID(I))THEN
+         N1=IKLE%I(TECELEM(I))
+         N2=IKLE%I(NELEM+TECELEM(I))
+         N3=IKLE%I(2*NELEM+TECELEM(I))
+         !
+         SURDET=1.D0/((X(N2)-X(N1))*(Y(N3)-Y(N1))-
+     &          (X(N3)-X(N1))*(Y(N2)-Y(N1)))
+         !
+         DET1=(X(N3)-X(N2))*(YTEC(I)-Y(N2))
+     &              -(XTEC(I)-X(N2))*(Y(N3)-Y(N2))
+         DET2=(X(N1)-X(N3))*(YTEC(I)-Y(N3))
+     &              -(XTEC(I)-X(N3))*(Y(N1)-Y(N3))
+         DET3=(X(N2)-X(N1))*(YTEC(I)-Y(N1))
+     &              -(XTEC(I)-X(N1))*(Y(N2)-Y(N1))
+         !
+         UTECX=U%R(N1)*DET1*SURDET+
+     &          U%R(N2)*DET2*SURDET+
+     &          U%R(N3)*DET3*SURDET
+         UTECY=V%R(N1)*DET1*SURDET+
+     &          V%R(N2)*DET2*SURDET+
+     &          V%R(N3)*DET3*SURDET
+
+       END IF
+       IF(NCSIZE.GT.0)UTECX=P_DSUM(UTECX)
+       IF(NCSIZE.GT.0)UTECY=P_DSUM(UTECY)
+       ALPHA=-ATAN2(UTECY,UTECX)*RTD
```

It should be noted that function $ATAN2$ might not be present in all compilers, in which case it will need to be computed by the user.

The far velocity $\boldsymbol{U_{ref}}$ will be then computed and projected along the orientation of the TEC to give U_r.

```
@@ -77,0 +367,32 @@
+       ! FIND THE FAR VELOCITY
+       IANGLE=INT(ALPHA+0.5D0)
+       XDD = XTEC(I) - DD(I)*COS(DBLE(IANGLE)*DTR)
+       YDD = YTEC(I) + DD(I)*SIN(DBLE(IANGLE)*DTR)
+       UREFX=0.D0
+       UREFY=0.D0
+       IF(IPID.EQ.DDPID(I,IANGLE))THEN
+         N1=IKLE%I(DDELEM(I,IANGLE))
+         N2=IKLE%I(NELEM+DDELEM(I,IANGLE))
+         N3=IKLE%I(2*NELEM+DDELEM(I,IANGLE))
+         !
+         SURDET=1.D0/((X(N2)-X(N1))*(Y(N3)-Y(N1))-
+     &          (X(N3)-X(N1))*(Y(N2)-Y(N1)))
+         !
+         DET1=(X(N3)-X(N2))*(YDD-Y(N2))
+     &          -(XDD-X(N2))*(Y(N3)-Y(N2))
+         DET2=(X(N1)-X(N3))*(YDD-Y(N3))
+     &          -(XDD-X(N3))*(Y(N1)-Y(N3))
+         DET3=(X(N2)-X(N1))*(YDD-Y(N1))
+     &          -(XDD-X(N1))*(Y(N2)-Y(N1))
+         !
+         UREFX=U%R(N1)*DET1*SURDET+
+     &          U%R(N2)*DET2*SURDET+
+     &          U%R(N3)*DET3*SURDET
+         UREFY=V%R(N1)*DET1*SURDET+
+     &          V%R(N2)*DET2*SURDET+
+     &          V%R(N3)*DET3*SURDET
+
+       END IF
+       IF(NCSIZE.GT.0) UREFX=P_DSUM(UREFX)
+       IF(NCSIZE.GT.0) UREFY=P_DSUM(UREFY)
+       UR=UREFX*COS(THETA(I))-UREFY*SIN(THETA(I))
```

The local nodal values will be read, i.e. U_j, V_y and h_j.

```
@@ -77,0 +399,36 @@
+       ! DO A LOOP OVER ALL THE AFFECTED NODES
+       DO INODE=1,NNODES(I)
+         IPOIN = INODES(I,INODE)
+         IF(IPOIN.LT.0)THEN !IT IS A QUASI-BUBBLE ELEMENT
+           ! FIND IELEM
+           IELEM=-IPOIN
+           N1=IKLE%I(IELEM         )
+           N2=IKLE%I(IELEM+  NELMAX)
+           N3=IKLE%I(IELEM+2*NELMAX)
+           !
+           I4=IKLE%I(IELEM+3*NELMAX)
+           X4=(X(N1)+X(N2)+X(N3))/3.D0
+           Y4=(Y(N1)+Y(N2)+Y(N3))/3.D0
+           !
+           SURDET=1.D0/((X(N2)-X(N1))*(Y(N3)-Y(N1))-
+     &          (X(N3)-X(N1))*(Y(N2)-Y(N1)))
+           !
+           DET1=(X(N3)-X(N2))*(Y4-Y(N2))
+     &          -(X4-X(N2))*(Y(N3)-Y(N2))
+           DET2=(X(N1)-X(N3))*(Y4-Y(N3))
+     &          -(X4-X(N3))*(Y(N1)-Y(N3))
+           DET3=(X(N2)-X(N1))*(Y4-Y(N1))
+     &          -(X4-X(N1))*(Y(N2)-Y(N1))
+           !
+           HI=H%R(N1)*DET1*SURDET+
+     &        H%R(N2)*DET2*SURDET+
+     &         H%R(N3)*DET3*SURDET
+           UI=U%R(I4)
+           VI=V%R(I4)
+           !
+           IPOIN=I4 ! REDEF IPOIN TO BE USED BY FUDRAG
+         ELSE
+           HI=H%R(IPOIN)
+           UI=U%R(IPOIN)
+           VI=V%R(IPOIN)
+         END IF
```

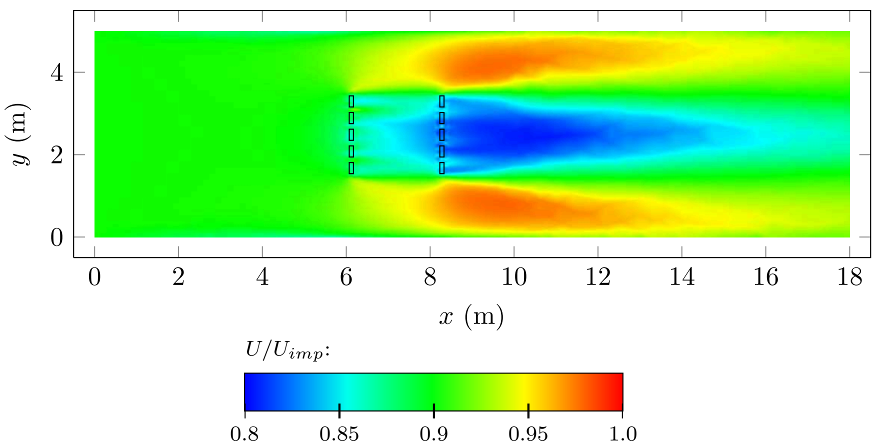


Fig. 3: Velocity contour showing the wake of 2 rows of TECs in a flume. The black rectangles show the positions of the TECs.

The two components of the source term modelling the drag force in Telemac-2D, `FUDRAG` and `FVDRAG`, can now be calculated:

```
@@ -77,0 +435,16 @@
+          !
+          IF((ABS(UR).GT.VCUT(I)).AND.(HI.GT.1.D-4))THEN
+            HALFCDSCOSTHETA=0.5D0*PI*RTEC(I)**2
+     &                      *CDTEC(I)*COS(THETA(I))
+            HALFCDSSINTHETA=0.5D0*PI*RTEC(I)**2
+     &                      *CDTEC(I)*SIN(THETA(I))
+            IF(ABS(UI).GT.1.D-4)THEN
+              FUDRAG%R(IPOIN)=FUDRAG%R(IPOIN)
+     &        -HALFCDSCOSTHETA*UR*ABS(UR)/(AREA(I)*HI*UI)
+            END IF
+            IF(ABS(VI).GT.1.D-4)THEN
+              FVDRAG%R(IPOIN)=FVDRAG%R(IPOIN)
+     &        +HALFCDSSINTHETA*UR*ABS(UR)/(AREA(I)*HI*VI)
+            END IF
+          END IF
+        END DO
```

C. Using the subroutine DRAGFO to calculate the extracted power

The subroutine `DRAGFO` can also be used to calculate the extracted power.

```
@@ -77,0 +451,5 @@
+!
+        PTEC(I) = CPTEC(I)*PI*RTEC(I)**2*0.5D0
+     &            *ROEAU*UR**2*ABS(UR)
+!
+      END DO
```

It can then be written in the file given by `FORMATTED RESULTS FILE`:

```
@@ -77, +456,41 @@
+!
+!-----------------------------------------------------------
+! WRITING THE POWER OUTPUT
+!-----------------------------------------------------------
+! WRITING HEADER OF FILE
+      IF((IPID.EQ.0).AND.(LT.EQ.1))THEN
+        ! FILE ID
+        POWRES = T2D_FILES(T2DRFO)%LU
+        ! HEADER
+        WRITE(POWRES,'(A)') "# TEC power result file:"
+        WRITE(POWRES,'(A)') "# Power extracted at time T"
+        WRITE(POWRES,'(A)') "# for each TEC modelled"
+        ! VARIABLE NAMES
+        WRITE(POWRES,'(A)',ADVANCE='NO') "T, "
+        DO I = 1,NTEC-1
+          WRITE(POWRES,'(A,I0.4,A)',ADVANCE='NO')"P_",I,", "
+        END DO
+        WRITE(POWRES,'(A,I0.4)') "P_",NTEC
+        ! INITIAL RESULT
+        WRITE(POWRES,'(F0.6,X)',ADVANCE='NO') 0.D0
+        DO I = 1,NTEC-1
+          WRITE(POWRES,'(F0.6,X)',ADVANCE='NO') 0.D0
+        END DO
+        WRITE(POWRES,'(F0.6)') 0.D0
+      END IF
+! WRITING RESULTS IN TIME
+      IF((IPID.EQ.0).AND.(MOD(LT,LEOPRD).EQ.0))THEN
+        ! FILE ID
+        POWRES = T2D_FILES(T2DRFO)%LU
+        ! CALCULATED RESULTS
+        WRITE(POWRES,'(F0.6,X)',ADVANCE='NO') LT*DT
+        DO I = 1,NTEC-1
+          WRITE(POWRES,'(F0.6,X)',ADVANCE='NO') PTEC(I)
+        END DO
+        WRITE(POWRES,'(F0.6)') PTEC(NTEC)
+      END IF
 !
 !-----------------------------------------------------------
 !
       RETURN
       END
```

When using the code presented earlier, do not forget to delete the default example present in `DRAGFO`.

V. Example of simulations

Now that the entire user code has been given, a few illustrations of the output will be given. The simulations chosen will reproduce one of the experiment conducted during the PerAWaT project [7], for which two rows of turbines were placed in a canal. Applying a drag force in the flow will create a wake behind the TECs, see figure 3 and 4. However, the velocity close to the TEC is too high, and it is thought to be accurate at a distance greater than 10-15 diameters downstream.

VI. Concluding remarks

The code that should be used to model a group of TECs in a Telemac-2D simulation has been presented here. Nonetheless, several choices have been left to the user:

- The size of mesh elements at the position of the TEC.
- The area over which the drag force will be applied.

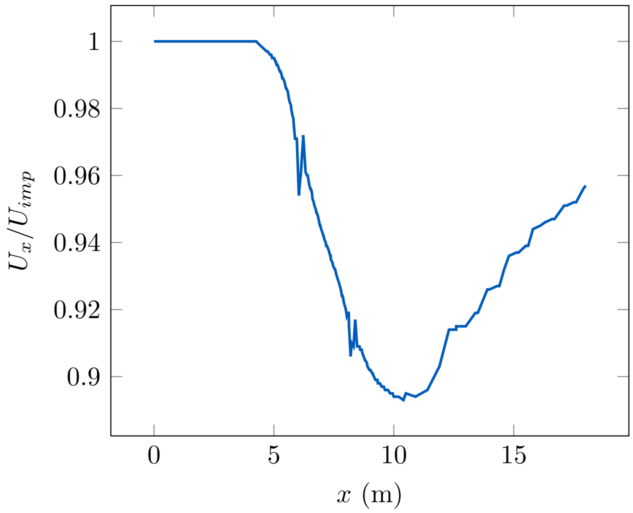


Fig. 4: Velocity profile along $y = 2.483$ m.

- The distance upstream of the TEC for which the reference velocity will be taken.

Other variables will depend on the machine modelled:

- The radius of the device.
- The orientation of the device in the flow.
- The drag coefficient.
- The power coefficient.

It should also be stated that it is not possible to get accurate velocity profiles close to the device in Telemac-2D simulations, as this would require modelling in detail each components of the TEC. The velocity modelled with this approach is thought to be valid 10-15 diameters downstream of the device.

In addition, it is recommended to use quasi-bubble elements in Telemac-2D simulations, as the presence of the TECs tend to create instabilities. It is also recommended to use an unstructured mesh around the TECs. Furthermore, the turbulence model can have an impact on the result and it should be chosen with care.

Finally, when dealing with large farms it is possible to apply the drag force of several TEC over an area covering the size of the farm (which could reduce the number of nodes in the mesh). The only modification will be to multiply surface (πR^2) used in the drag force by the number of TECs in the farm.

Acknowledgment

The developments presented were done and validated through the PerAWaT project, financed by the ETI.

References

[1] E. de Paula Kirinus, C. Eadi Stringari, W. Correa Marques, and H. Barreto Matzenauer, "Harvesting the Currents Power on the Southern Brazilian Shelf," in *Proc. 20th Telemac User Conference*, Karlsruhe (Germany), 2013.

[2] D. Haverson, J. Bacon, and H. Smith, "Modelling the far field impacts of a tidal energy development at Ramsey Sound," in *Proc. 21st Telemac User Conference*, Grenoble (France), 2014.

[3] J.-M. Hervouet, *Hydrodynamics of free surface flows*. Chichester: Wyley, 2007.

[4] A. Joly, M. Andreewsky, V. Martin, S. Saviot, and C.-T. Pham, "Tidal coastal basin numerical modelling. Large scale effects of tidal energy extraction," PerAWaT WG3 WP3: EDF, Tech. Rep., 2013.

[5] A. Joly, M. Andreewsky, S. Saviot, and C.-T. Pham, "Tidal farm modelling: The Alderney Race, the Pentland Firth and the Paimpol-Bréhat sites modelled in Telemac software," PerAWaT WG3 WP3: EDF, Tech. Rep., 2013.

[6] S. Serhadlıoğlu, T. A. A. Adcock, G. T. Houlsby, S. Draper, and A. G. L. Borthwick, "Tidal stream energy resource assessment of the Anglesey Skerries," *International Journal of Marine Energy*, vol. 3-4, pp. e98–e111, 2013.

[7] T. Stallard and R. Collings, "Calibration report for Scale Model Experiments," PerAWaT WG3 WP4: University of Manchester, Tech. Rep., 2011.