Man, Yi; Nistor, Ioan; Rennie, Colin D.; Cousineau, Julien

Implementing Effective Performance Turbine Model into TELEMAC-3D

Implementing Effective Performance Turbine Model into TELEMAC-3D

Yi Man, Ioan Nistor, Colin D. Rennie
Department of Civil Engineering
University of Ottawa
Ottawa, Canada
aman070@uottawa.ca

Julien Cousineau
Ocean, Coastal and River Engineering
National Research Council Canada
Ottawa, Canada
Julien.Cousineau@nrc.ca

Abstract—River Energy Converters (REC) typically need to be installed in an array formation or farm layout to maximize their cost efficiency. Optimizing the arrangement of such formations plays a vital role in achieving the array's maximum effectiveness. In order to analyse the performance of the turbines, a high-definition numerical simulation using a fine computational grid is the most practical approach. The Laboratoire de Mécanique des Fluide Numérique (LMFN) of Laval University, Quebec City, Canada, developed, in collaboration with the National Resources Canada (NRCan), a model approach called the Effective Performance Turbine Model (EPTM) that accurately predicts the wake recovery of turbines in an array formation [1]. The EPTM was originally developed using a computational fluid dynamics (CFD) model and validated with laboratory experiments performed at the National Research Council [2]. However, simulations for fluvial, coastal and ocean domains, spanning several hundred kilometres, are often out of reach for CFD models due to high computational costs. The authors have undertaken the task to implement and validate the EPTM in TELEMAC-3D in the goal of properly simulating power extraction and wake recovery in array turbine formation. The paper presents the implementation of the EPTM in TELEMAC-3D with an example.

I. Introduction

A well-planned river energy converter layout will achieve the best price-quality ratio for economic return. High-fidelity CFD numerical modelling is the chosen approach due to its high precision and flexibility in modifying and adding parameters. The Laboratoire de Mécanique des Fluide Numérique (LMFN) of Laval University developed a fully resolved CFD reference simulation with about 60 million cells in a 20-meter simulated flume. This simulation displayed and described the wake characteristics and power extraction of a crossflow turbine in both 2D and 3D [2]. Their work selected Unsteady Reynolds-Averaged Navier-Stokes (URANS) method for flow resolution and used about 60 million cells to simulate the turbine and turbulence region. The results were highly accurate, with only about 2-5% difference from the results obtained by [3] in 2D and 3D for both crossflow turbine and axial-flow turbine simulations. Natural Resources Canada (NRCan) subsequently tasked LMFN with the development of an efficient simplified CFD tool to reduce the impractical computational expenses, which took a couple of months with hundreds of cores to resolve, for large river energy converter projects. Effective Performance Turbine Model (EPTM) was, as such, proposed. EPTM has been used to simulate axial flow turbine and cross flow turbine in 2D and 3D [1]. Instead of using URANS, EPTM employed steady-state RANS equation as the solver. Combined with a coarser mesh, the computation speed was significantly improved while maintaining similar results. The region of turbine with its simulated drag forces was simplified to an actuating region. In the case of a cross flow turbine (CFT), an actuating annulus or cylindrical region is preferred, as proven by [4] and [5]. The present study aims to implement the EPTM 3D cross flow turbine (EPTM-3DCFT) from Laval University into TELEMAC-3D. The TELEMAC-3D model can be further used for regional scale models spanning hundreds of kilometres across the domain of interest. Therefore, lowering the computational cost while maintaining reasonable accuracy is the focus and the aim of this project.

As will be shown, the downstream wake topology produced by the TELEMAC-3D simulation showed very good agreement to the EPTM result, but noticeably longer wake and near wake ($x/D > 5$) velocity discrepancies at about five turbine diameters downstream are found.

II. Methodology

A. EPTM

The EPTM is a simplified numerical model which replaced the turbine rotor by an actuating region to simulate the wake zone induced by the interactions between the flow and the turbine objects. The actuating region takes the time averaged drag coefficients from the fully resolved reference model, therefore eliminating the need to calculate the interactions between the fluid and the blades. The turbine flow zone is reproduced by mimicking the mean flowrate and the time-averaged forces passing through the turbine. Given similar parameters, the actuating region will determine the averaged power extracted by the system [6] using a drag force equation. Three turbine models are simplified using the EPTM model: the axial flow turbine, EPTM-AFT, and cross flow turbine in 2D and 3D, EPTM-2DCFT and EPTM-3DCFT. In the scope of this simulation using TELEMAC, only the data from EPTM-3DCFT is used.

B. Drag Force Equation

The drag force which the turbine experiences was calculated using the following formula:

$$F = \frac{1}{2}\rho C_d^* A V^2 \qquad (1)$$

Where ρ is the density of the fluid (kg/m^3), C_d^* is the effective drag coefficient from EPTM simulation, A is the area of the turbine's center plane (m^2), and V is the total averaged velocity from every node inside of the turbine (m/s).

The effective drag coefficient, C_d^*, is the drag coefficient normalized with effective velocity, V^* (m/s). The effective velocity, V^*, or otherwise defined as the "axial disk-averaged velocity," is the time- and spatial-averaged velocity on the center mesh plane of the turbine object. The turbine centerplane is illustrated in Fig. 1.

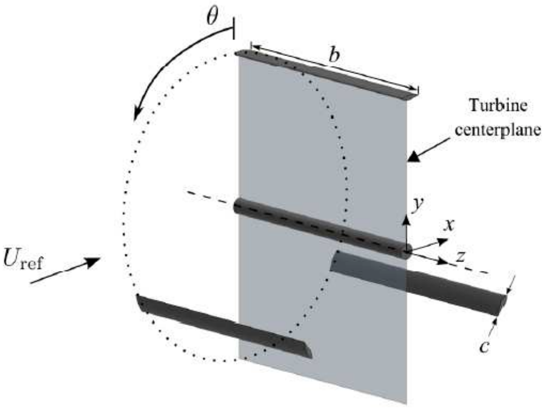


Figure 1 Turbine centerplane illustration [1]

In Fig. 1, U_{ref} is the inflow velocity, θ indicates the rotation of the turbine in radians, b is the turbine span, c is the blade chord length of the turbine, while x, y, z indicates the axes.

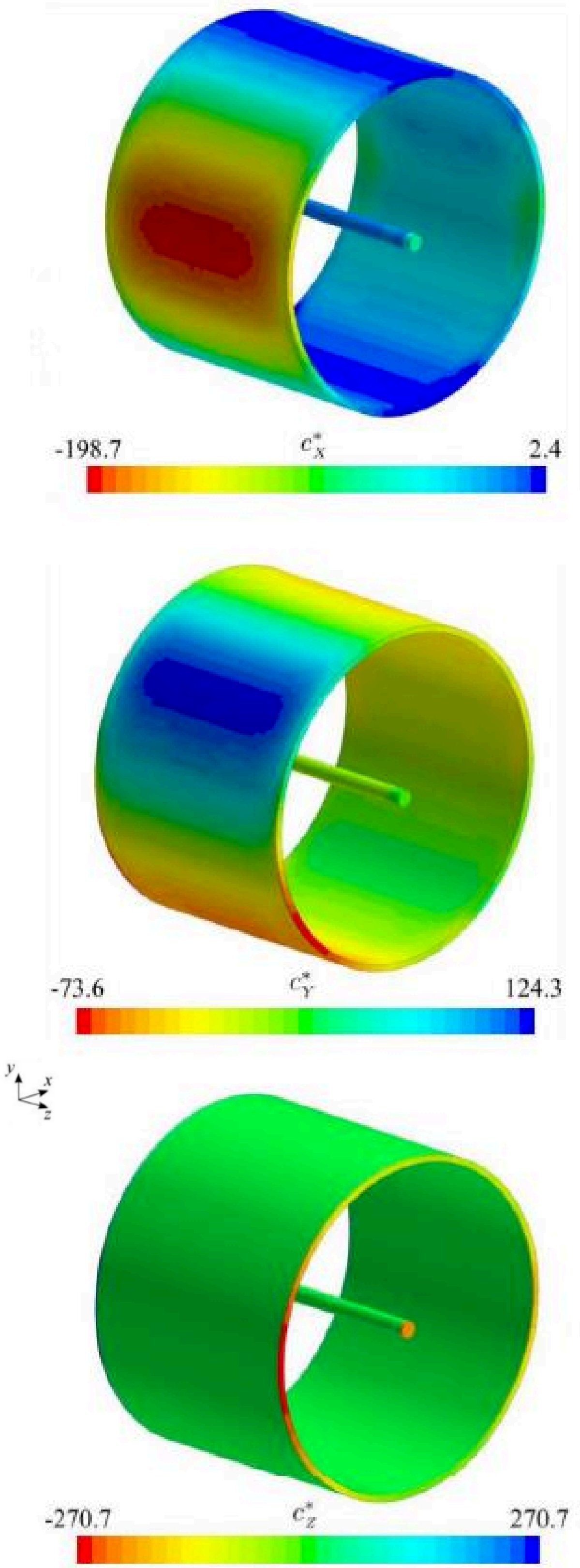


Figure 2 Effective force distribution for each component from EPTM-3DCFT [2]

Figure 2 represents the effective drag coefficient mapped on the turbine from the EPTM-3DCFT result. The notations c_X^*, c_Y^*, c_Z^* are the same effective drag coefficient discussed above (C_d^*). In the x-direction, the region that faces the incoming flow has large negative effective drag coefficients indicated in red, while the top and bottom regions have very low magnitude of effective drag coefficients. The turbine experienced large positive effective drag coefficients indicated in blue in the y-direction, and a significant negative value was observed on the span of the turbine blade on the lower half of the actuating region. Moreover, in the z-direction, the tip of the blade and shaft exhibited large C_d^* values, similar in magnitude but on the opposing side of the blade and shaft. It is noticeable that the forces vary with the changes of the positions of the blades. The model calculates the mean forces of a region in a narrow time interval. If the changes happen too rapidly, then there is a chance that the time interval will not capture the changes in the forces and disrupt the stability of the model [2].

III. Implementation in TELEMAC-3D

A. Drag Force as Source Term

`S1U%R()`, `S1V%R()`, and `S1W%R()` are defined as the implicit source terms on velocity in U, V, and W directions in [4] and [7]. These terms are used by TELEMAC-3D to update the velocity of each node at each time step. Since the source term is set to be drag force divided by volume, and the volume of the turbine object averaged in Laval University's study, the source term is $S1U = Drag\ force/Vol_{TEC}$. The complete formula is as follows:

$$S1U(I3) = (\tfrac{1}{2}\rho C_{dx}^* A U^2)/Vol_{TEC} \qquad (2)$$

$S1U(I3)$ is the source term in the x direction (kg/s^2m^2), and $I3$ is referring to each individual node of the TELEMAC mesh. ρ is the density of the fluid (kg/m^3), C_{dx}^* is the effective drag coefficient of each node in x-direction obtained from the EPTM model. A is area of the turbine centerplane shown in Fig. 1. U is the fluid velocity in the x direction (m/s), and Vol_{TEC} is the sum of discretised mesh node volume within the defined turbine region (m^3).

B. Turbine Volume and Mean Velocity

EPTM-3DCFT takes the average of velocities from the center plane of the turbine as the averaged velocity the turbine experiences. Following the same logic, the velocity terms used for the drag force equation and the source term equation must also be the mean velocity of the turbine. Since the turbine is comprised of different mesh points in the TELEMAC simulation domain, by averaging the velocities from each mesh point inside the turbine object, turbine mean velocity can be calculated. Looping through all the mesh points and using `inpoly` function to check if the mesh point is inside the turbine, all turbine mesh points are collected. The volume of turbine, V_{TEC}, is the sum of the volumes of all the mesh nodes inside the turbine. The turbine mean velocity can be obtained by dividing the sum of all the velocities in three directions from each mesh point to the number of points.

C. Computational Domain

The TELEMAC simulation domain is set up with the same geometry as that of the EPTM-3DCFT test case. The channel is 30 meters long, 1.68 meters wide, and 2.24 meters in depth. With turbine diameter being 1 meter and a transverse span of 0.75 meters, the blockage ratio is calculated to be 20%, which is one of the EPTM simulation trials that this project is aiming to duplicate. The spacing between each TELEMAC-3D mesh node, Δx and Δy, is set to 0.021 meters throughout the domain, yielding 4.6 million nodes across the domain. The channel has a free surface. Both sides of the wall and channel bottom are defined as solid wall. The inflow follows prescribed constant water elevation, and outflow is considered to be open boundary with free velocity and elevation. The turbine origin is set at (0,0) of the simulation domain, meaning that the span of the flume is from -10 m to 20 m in x direction, -1.68 m to 1.68 m in y direction, and -1.12 m to 1.12 m in z direction.

BlueKenue is a data visualization software developed by National Research Council Canada. This tool supports direct data input from TELEMAC-3D simulation results. Various choices of perspectives enable users to have a graphical representation of the results, serving as a sophisticated tool for data analysis.

IV. Code Implementation

A. Supporting Modules

The main Fortran subroutine for TELEMAC-3D simulations cannot work without supporting Fortran modules. These modules contain functions that handle the mathematics, geometry, objects, etc., enabling the main subroutine to be written in a more concise and efficient manner. The steering file is the most important supporting file as it governs many properties of the simulation including, fluids, boundaries, time step, and which solver equation sets to be employed. Geometry, input, and boundary files are specified in the steering file as well. There are two formatted data files as input. TELEMAC-3D will read the data from these files and store them in memory in the form of arrays for the program to use. The formatted data file 1 (`FO1`) is a text file for the turbine energy convertor (TEC) properties. This file includes the TEC's name, diameter, thickness of the blade, position in the domain, and rotations in degrees along x, y, and z axis. The number of lines indicates the number of turbines to be put in simulation. Formatted data file 2 (`FO2`) is a csv file detailing the EPTM data of the effective drag coefficient around the turbine object. The values are presented in cylindrical coordinates (radius, theta, z-elevation). The radius and theta will give us the points in '*x* and *y* direction', unfortunately, due to the different coordinate convention, the *y* component from cylindrical coordinate system is the z in Cartesian coordinate system, as it signifies the depth. The *z* component becomes the y axis after the conversion, marking the width of the channel span. The *x* component stays the same, as both are describing the flow direction. The conversion to Cartesian coordinate system, along with rotation, transformation, and scaling of the turbine object are handled by `Mat4.f90` module. To identify if a node is located within the turbine, one needs to be inside of TEC's XY and XZ (front and side view). `Qhull.f90`, adopted from Qhull, is used to draw convex hulls representing the two planes. Since the input arrays must be sorted in ascending order for the module to function, two quicksort modules were adapted and utilized, one for one-dimension array and another for two-dimension array. An intrinsic function `Inpoly.90` then identifies if the given point is within the boundary of both planes. If it is, then this point will be considered when calculating the TEC's volume and average velocities. A list of supporting files with descriptions is presented below:

- `A_qsort.f90`: provides quick sort function to a given array. "A_" was added for TELEMAC to load this file first, as the program takes in files alphabetically.
- `Check.f90`: checks if a is equal to b for different element types.
- `File_csv.f90`: writes computed 2D array results in an .csv file.
- `Geometry.f90`: handles the translation, scale, rotation, and transform of the turbine object represented a 3D array format.
- `Inpoly.f`: checks if a point is inside of a given 2D polygon
- `Mat4.f90`: handles a four by four array's reset, translation, scale, rotation, and transform function.
- `Math.f90`: uses qsort.f90, unique.f90, unique2.f90 to handle their functionalities.
- Polyarea.f: computes the area of a given enclosed polygon.
- `Qhull.f90`: computes the convex hull of a given array. See [8] for more information.
- `Quicksort_include.f03`: calculates scenarios before calling quick sort module for optimal speed.
- `Turbine.f90`: defines and creates the turbine object and its values and functions, including reading CFD values from input file.
- `Turbines.f90`: creates the formation of the turbine objects constructed.
- `Unique.f03`: Used by Math.f90 to check if the value is unique in a given series.
- `Unique2.f03`: Used by Math.f90 to check if a point is unique in a 2D array.
- `User_condi3d_uvw.f`: overwrites the default TELEMAC-3D input boundary condition of flow velocity in uvw direction. Works in conjunction with steering file.
- `User_mesh_transf.f`: overwrites the default TELEMAC-3D mesh transformation
- `User_source.f90`: the main source file which is explained in the following section

All supporting files can be found on the GitHub repository [9] : https://github.com/Julien-Cousineau/telemac3d-eptm.git.

`turbine.f90` is responsible for creating individual turbine objects, which can be considered as a 'class' in other programming languages. This type contains the numerical values read from `FO1` and `FO2`. It also contains procedures to allow users to modify its special properties, such as, rotation, translation, and scaling.

`turbines.f90` is set as an overall construction of the turbine formation. This module reads `FO1` to gain information on the number of TEC and the characteristics of each. A one-dimension array `TEC(:)` which holds objects of type `Turbine` is defined. in the module `turbine.f90` (to be differentiated from `turbines.f90`). There are two subroutines in the `turbines.f90`, `readTEC` and `readCD`. `readTEC` reads the number of turbines in `FO1` then it constructs `TEC(:)` with size equal to that number. Since each element within `TEC(:)` is of type *Turbine*, an initiation sequence called *initialize* is called with input arguments read from `FO1`. `readCD` follows the same logic, where it makes each turbine object to read the C_d^* values from `FO2`.

B. *Main Source File*

This module functions as the main program of this simulation as it overwrites the source file (user_source.f90) of TELEMAC. It uses a few intrinsic modules such as `BIEF`, `DECLARATIONS_SPECIAL`, and `DECLARATIONS_TELEMAC3D`.

a) Turbine Initialization

After declaring variables, upon the first iteration *(*`LT .EQ. 1`*)*, the program checks if the files (`FO1`, `FO2`) of turbine position and drag coefficient from the input files are in place, and will send error messages and calling the program to stop if they are not. `readTEC` *and* `readCD` are then called to read the data and store them into memory. Mesh points are saved into `meshIds` array if a given mesh point is inside of the TEC boundary in 2D view.

b) Inverse Distance Weighting (IDW)

To mimic the total drag force calculated in EPTM in TELEMAC, it is vital to use the same drag coefficient data. The spatially distributed drag coefficients data needs mapping into the coarser mesh grid of TELEMAC. To achieve this, IDW method was employed to approximate the spatially averaged drag coefficients around each node within the turbine location region in TELEMAC. Each turbine mesh node gathers scattered EPTM drag coefficient values within the defined search radius. The distance `dist` between each pair of EPTM and TELEMAC nodes is calculated using:

$$distance = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$

Since TELEMAC has an intrinsic coordinate value for each node, and EPTM node coordinates are stored in array `temptec`, the code is line 1 to 3 in the appendix. The unit is in meters.

A distance threshold `distThreshold` is defined; if the distance between two nodes is within the threshold, the value of the EPTM node is used, instead of proceeding to obtain the value from the IDW method based on the surrounding nodes within the search radius. The defined threshold is a part of the variables that can be investigated in the sensitivity analysis. This procedure is to ensure the greatest accuracy and the most efficient use of computing power are taken into account. There is no need to use IDW to map EPTM value into TELEMAC grid if the two nodes are within a defined distance threshold. Furthermore, in order to maintain the highest accuracy of the approximated value, the closest EPTM node value is used. However, in most cases, there are no nearby EPTM nodes within the distance threshold, then the second order IDW method is employed to calculate the mesh node value from interpolating nodes within the search radius using:

$$C_{d_p} = \frac{\sum_{i=1}^{n}(\frac{C_d}{d_i{}^p})}{\sum_{i=1}^{n}(\frac{1}{d_i{}^p})} \tag{4}$$

In Fortran, this is achieved by calculating the sum of both numerator and denominator on the x, y, z direction for each node, then dividing them. The search radius is the most impactful variable in IDW method, and it is crucial for the users to determine the best radius for their project needs. A logical flag `pointsInRange` is set to check if there are any points within the search radius. Sample code is in the appendix from line 4 to 37.

Parameters are reset at the end of each loop going through the nodes. Lastly, the effective C_d values along with their node IDs are added to the `updateArray` for velocity update.

c) Source term update

The velocity of each node is updated by calculating the drag forces as implicit source terms using the drag coefficients obtained earlier using $F_D = \frac{1}{2}\rho v^2 C_d A$. The implicit source terms are defined by the drag force on each node divided by the volume of the turbine. Therefore, the drag force itself is the implicit source term multiplied by the volume of the turbine object in the simulation domain. Updating the implicit source terms along all axes and the collection of drag forces are coded as shown in the appendix line 38-40.

In the code, S1U, S1V, S1W are the implicit source terms in u, v, w flow direction, and `%R` represents its value. `I3` is the specification of the node ID being updated. `effX`, `effY`, and `effZ` are the effective drag coefficient imported and calculated from the EPTM csv file. `TcenterplaneA` is the turbine's centerplane area. `TECAvgU`, `TECAvgV`, and `TECAvgW` are the turbine objective's average velocity in u, v, and w direction from the last time step. `TECVolu` stands for the turbine objective volume in the mesh.

V. Simulation Example

The CFD results are generated with the aim to validate the TELEMAC simulation results with coarser mesh against the EPTM-3DCFT results. The inflow flowrate is prescribed to be 3.763 m^3/s. With 1.68 m in width, and a prescribed fluid elevation of 1.12 m (2.24 m in flow depth), the inflow velocity is at 1 m/s, which is the reference velocity from the EPTM simulation. Several turbulence models were tested, and the most promising results were obtained when using the K-Epsilon model. This turbulence model can be activated by setting both horizontal and vertical turbulence model key words to 3 in the steering file. This model solves the balance equations for turbulent energy and turbulent dissipation [10] to determine the local eddy viscosity. The diffusion coefficient for fluid is set as 1E-6, the default, as instructed in the manual. The law of bottom friction in the steering file is set to model 3 which is the Strickler's law. And the friction coefficient is left at the default value of 60.

Figure 3(a) and (b) are the reference results generated by LFMN. Figure 3(a) is the high-fidelity CFD simulation result, and Fig. 3(b) is the EPTM version of the 3D crossflow turbine, which is scoped to be replicated using TELEMAC. Figure 4, 5, and 6 presented the simulation using TELEMAC. The color legends on the left-hand side are for all three TELEMAC figures. Fig. 4 used K-Epsilon model while the Fig. 5 and 6 employed the constant viscosity method with different diffusion coefficients. The simulation parameters are identical for all different scenarios. Time step Δt is 0.05 seconds for 1000 steps, yielding 50 seconds of simulation time. The center of the turbine is set at the origin (0,0). Spanning 1 meter long, the upstream and downstream of actuating cylinder representing the blades are located at x=-0.5 and x=0.5. The reduction of the velocity just upstream of the turbine is indicative of the blockage effect and the interaction between the turbine structure and the fluid. And the slight increase of fluid velocity near both walls in y-direction is the result of the fluid and boundary interaction. The reduction of velocity downstream of the turbine indicates energy loss in the fluid which is the energy extracted by TEC. The reference simulated wake from 3D-CFT takes a short distance to recover, and the simulated wake of EPTM-3DCFT requires more time to recover, resulting in a longer wake. The TELEMAC result of flow wake reaches to about seven to eight meters downstream of the TEC origin, depending on the choice of turbulence model. The $k - \epsilon$ model yielded a more uniform and less turbulent wake similar to the reference results. The constant viscosity model produced a more chaotic and longer wake. The simulation results are similar to the result from Laval University; however, the key difference is that the TELEMAC generated wake is longer than the wake from EPTM [1], despite using the same effective drag coefficient; different turbulence models will produce different wakes. Therefore, the balance between computational resources and the accuracy of the results is left to the latitude of the researcher and to the requirements of specific project.

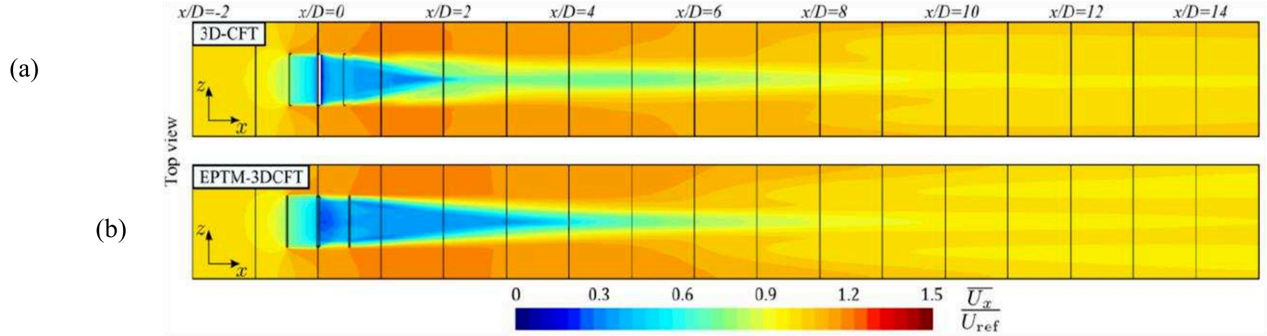
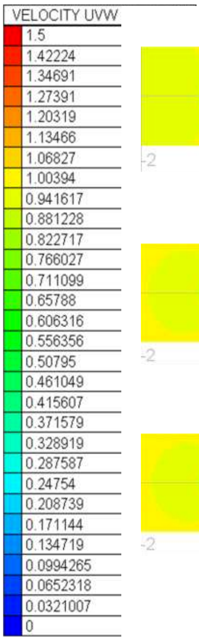


Figure 3 High-fidelity CFD and EPTM reference simulation result (Top View) [1]



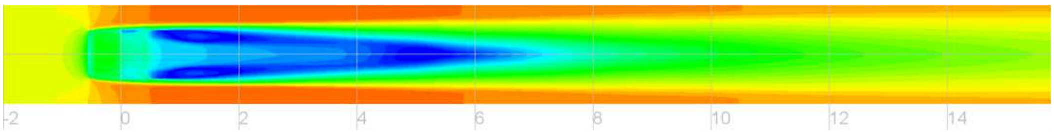


Figure 4 Temporal averaged result using K-Epsilon model (Top View)

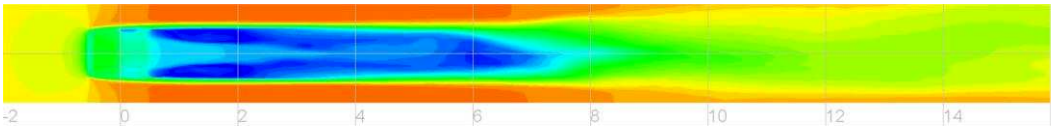


Figure 5 Temporal averaged result using constant viscosity with diffusion coefficient 1×10^{-6} (Top View)

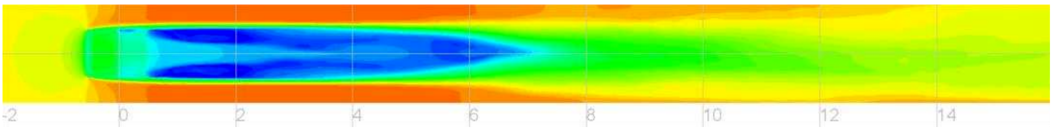


Figure 6 Temporal averaged result using constant viscosity with diffusion coefficient 1×10^{-7} (Top View)

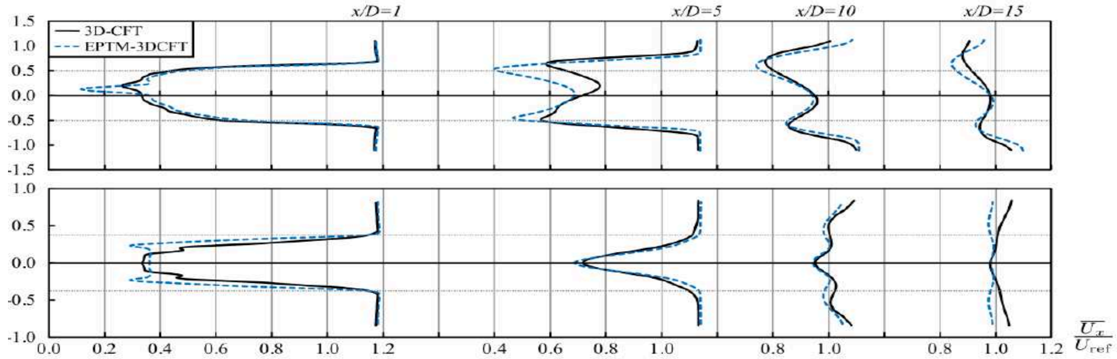


Figure 7 High fidelity and EPTM-3DCFT mean stream velocity profile at different locations, top: vertical direction, bottom: transverse direction [1]

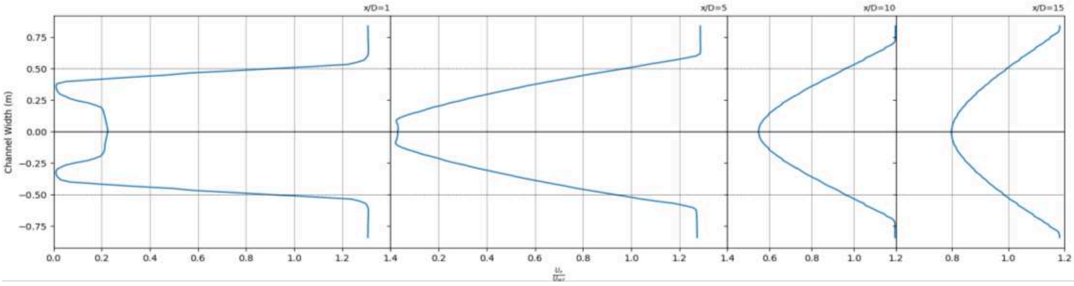


Figure 8 Velocity profile of the TELEMAC results using K-Epsilon method in vertical direction at different locations

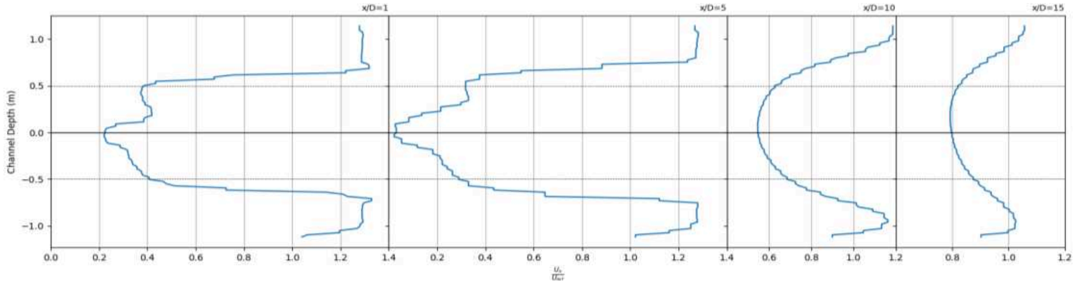


Figure 9 Velocity profile of the TELEMAC results using K-Epsilon method in transverse direction at different locations

Figure 7 showed the velocity profile of the high-fidelity reference simulation and EPTM-3DCFT velocity profile at 1, 5, 10, and 15 meters down stream of where the turbine is placed, given in this case the diameter of the turbine, D, is one meter. The black solid line represents the reference simulation while the blue dotted line shows the EPTM-3DCFT profile. Fig.8, and Fig. 9 are the velocity profile of the TELEMAC simulation using the K-epsilon method at different locations in vertical and transverse direction respectively. The jagged line in Fig. 8 is due to the fact that the data points had to be extrapolated from the 40 vertical layers defined. More vertical layers mean the result will be more refined at a cost of computational power. Both Fig. 8 and Fig. 9 resemble the velocity profile of the EPTM-3DCFT ones, except for when $x/D = 5$ in the transverse direction. As one can notice from the profiles, the simulated wake from TELEMAC takes notably longer to recover. the reduction in velocity is most prominent at both $x/D = 10$ and $x/D = 15$.

VI. Conclusions

This study utilized the Fortran programming language to add a module to TELEMAC-3D to mimic the results simulated using a high-fidelity CFD model and its simplified EPTM version. Actuating cylindrical region replaced the rotating blades, forming a time-averaged cylinder of drag coefficients around the turbine. Inverse distance weighting method was employed to distribute the input drag coefficients around the actuating region to ensure each mesh point has correct mappings of the drag coefficient, and therefore yields the correct drag force. A significant advantage of a coarser grid can reduce simulation time significantly, and the saved computing power can be allocated elsewhere or used to cover a large domain. The high-fidelity model 3D-CFT shown in Figure 3 took several months using 300 cores of a supercomputer centre. The EPTM-3DCFT model needed about 24 hours using 100 cores. The results obtained used the TELEMAC-3D model were achieved using a single core in a little more than half a day using the constant viscosity model.

The $k-\epsilon$ model adds about 25% to 40% of simulation time varying on the mesh resolution, including time step selection and the number of vertical layers. As suggested by the running time, the efficiency was significantly improved in magnitude by several hundred folds. The simulation was visualized using BlueKenue developed by the National Research Council, Canada. Results obtained were similar to the reference EPTM-3DCFT ones but exhibiting a longer wake region. Further sensitivity analysis, model improvement, and geometry adjustments are needed to achieve a better replication to the simulations obtained by [1].

Reference

[1] Bourget, Sébastien. “Development and assessment of a modeling method for hydrokinetic turbines operating in arrays.” (2018).

[2] Gauvin-Tremblay, O., Kinsey, T., Descoteaux, P. and Dumas, G., 2019. *VALIDATION OF A SIMPLIFIED NUMERICAL MODEL USING LABORATORY TESTING RESULTS OF RIVER HYDROKINETIC TURBINE ARRAYS*. Quebec City: CFD Laboratory LMFN, pp.2-5, 18-22, 64

[3] Boudreau, M., Dumas, G. 2017. Comparison of the wake recovery of the axial-flow and crossflow turbine concepts. J. Wind Eng. Ind. Aerodyn. 165:137-152. doi: 10.1016/j.jweia.2017.03.010.

[4] Antoine Joly, Chi-Tuân Pham, Marc Andreewsky, Sylvain Saviot, Lauriane Fillot. Using the DRAGFO subroutine to model Tidal Energy Converters in TELEMAC-2D. TELEMAC User Club 2015, Oct 2015, Warrington, United Kingdom. hal-01702806

[5] Provan, Mitchel & Cornett, Andrew & Knox, Paul & Cousineau, Julien & Ferguson, Sean. (2019). EXPERIMENTAL STUDY OF THE WAKE PRODUCED BY SINGLE AND MULTIPLE CROSS-FLOW TURBINE. Journal of Ocean Technology. 14. 2-20. - Joly et al. (2015)

[6] Bourget S., Gauvin-Tremblay O., Dumas G. (2018): “Hydrokinetic turbine array modelling for performance analysis and deployment optimization”, **Transactions of the CSME**, 42:370-381. http://dx.doi.org/10.1139/tcsme-2017-0088

[7] Wiki.opentelemac.org. 2020. User_Manual_TELEMAC-3D · Open TELEMAC-MASCARET. [online] Available at: <http://wiki.opentelemac.org/doku.php?id=user_manual_telemac-3d#modelling_turbulence> [Accessed 21 August 2020].

[8] Qhull.org. 2021. *Qhull code for Convex Hull, Delaunay Triangulation, Voronoi Diagram, and Halfspace Intersection about a Point*. [online] Available at: <http://www.qhull.org/> [Accessed 16 May 2020].

[9] *Man, Y., and Cousineau, J. (2021). "Julien-Cousineau/telemac3d-eptm". GitHub, <https://github.com/Julien-Cousineau/telemac3d-eptm.git> (Aug. 20, 2021).*

[10] Pham, C., & Joly, A. (2016). TELEMAC Modelling System [Ebook] (3rd ed., pp. 49-55). Opentelemac.org. Retrieved from http://www.opentelemac.org/downloads/MANUALS/TELEMAC-3D/telemac3d_user_manual_v7p1.pdf

Appendix

```
dist = DSQRT((MESH3D%X%R(I3)-
temptec%xyz(K,1))**2+(MESH3D%Y%R(I3)-
temptec%xyz(K,2))**2+(MESH3D%Z%R(I3)-
temptec%xyz(K,3))**2)

IF (dist .LT. distThreshold) THEN
IF (closest .EQ. -1.D0) THEN
closest = dist
effX = temptec%cd(K,1)
effY = temptec%cd(K,2)
effZ = temptec%cd(K,3)
ELSEIF (dist .LT. closest) THEN
closest = dist
effX = temptec%cd(K,1)
effY = temptec%cd(K,2)
effZ = temptec%cd(K,3)
ENDIF
ELSEIF (dist .LT. searchr) THEN
pointsInRange = .TRUE.
IDWSumX = IDWSumX+
     (temptec%cd(K,1))/(dist**2)
IDWSumY = IDWSumY+
     (temptec%cd(K,2))/(dist**2)
IDWSumZ = IDWSumZ +
     (temptec%cd(K,3))/(dist**2)
IDWSumDist = IDWSumDist + (1/dist**2)
ENDIF
IF (closest .EQ. -1.D0) THEN
IF (pointsInRange) THEN
L = L + 1
updateArray(L,1) = I3
updateArray(L,2) = IDWSumX/IDWSumDist
updateArray(L,3) = IDWSumY/IDWSumDist
updateArray(L,4) = IDWSumZ/IDWSumDist
ENDIF
ELSE
L = L + 1
updateArray(L,1) = I3
updateArray(L,2) = effX
updateArray(L,3) = effY
updateArray(L,4) = effZ
ENDIF

S1U%R(I3) = (0.5D0 * (-effX) *
TcenterplaneA * TECAvgU**2.D0)/TECVolu
S1V%R(I3) = (0.5D0 * (-effY) *
TcenterplaneA * TECAvgV**2.D0)/TECVolu
S1W%R(I3) = (0.5D0 * (-effZ) *
TcenterplaneA * TECAvgW**2.D0)/TECVolu
```