



ACKNOWLEDGEMENTS

We would like to thank our supervisor, Dr Alberto Aliseda, for his guidance, constructive advice and constant support throughout our study.

We are also extremely grateful to our tutor Teymour Javaherchi whose patience and support from the initial to the final level helped us to learn about numerical modeling and post treatment software.

We also thank Teymour Javaherchi and Amir Amini for their precious friendship and for the insight of the Iranian-American culture they gave us.

We are also thankful to the Lt Le Cleach as all the personal of the French Naval Academy who allowed this project.



Array Optimization of Marine Hydrokinetic Turbine

Students : Ensigns F. Rieseemann and M. Bewert, EN 2009
Department : Department of Naval Engineering
Host institute of studies : Department of Mechanical Engineering - University of Washington Seattle (WA) - USA
Director of research : Assistant Professor Dr A. Aliseda
Tutor of project : PhD-student T. Javaherchi
Corresponding officer : LT Cléach

Abstract

The development and the employment of renewable energy resources is one of the challenges of the 21st century. The exploitation of solar- and wind-energy has already become reality and is applied in a large scale. Another source of infinitive 'green' energy, which recently moved into the focus of laboratories and industry, is the tidal energy. Even though the concept of extracting power from tidal flows is not new, this technology had not yet reached its maturity. In order to produce considerable amounts of energy Marine Hydrokinetic (MHK) Turbines have to be deployed in large arrays, such as offshore wind-turbine arrays. One of the major challenges remains to understand the reciprocal interactions of tidal turbines. Wake effects of upstream turbines on downstream turbines can affect significantly the extraction of power. Experimental data about array effects are hardly available due to their high expenditures. Therefore, these effects have to be studied with the help of computational modeling. The determination of optimal turbine arrangements can be carried out by comparing the power extracted by upstream turbines and downstream ones in function of their relative position to each other.

Résumé

Le développement et l'exploitation d'énergies renouvelables est l'un des enjeux du 21^{ème} siècle. Actuellement, l'éolien et le solaire sont parfaitement maîtrisés et développés à très grande échelle. Depuis peu, les marées considérées comme source d'énergie inépuisable et 'écologique' refont l'objet de nombreuses études. Même si le concept de transformer la puissance du courant lors d'une marée en électricité n'est pas nouveau, cette technologie n'a toujours pas atteint sa maturité. Pour pouvoir produire une quantité rentable d'électricité, les turbines hydro cinétique marines doivent être déployées en parc, comme pour l'éolien offshore. L'un des plus importants challenge est de comprendre l'interaction entre les turbines. En effet la vague produite par les turbines en amont peut affecter significativement la puissance extraite des turbines en aval. Par conséquent, ces effets doivent être étudiés à l'aide de simulations sur ordinateur. L'arrangement optimal des turbines peut être déterminé en comparant la puissance extraite par les turbines en amont avec celle extraite par les turbines en aval en fonction de leur position respective.

Key words: Marine Hydrokinetic Turbine, wake, Array-optimization



GLOSSARY

Abbreviations	
ADM	Actuator Disk model
AI	Admiralty Inlet
BC	Boundary Condition
EMEC	European Marine Energy Centre
GUI	Graphical User Interface
LHS	Left-Hand-Side
MHK	Marine Hydrokinetic
NREL	National Renewable Energy Laboratory
RANS	Reynolds-Averaged Navier-Stokes
RHS	Right-Hand-Side
SRF	Single Reference Frame
SST	Shear-Stress-Transport
UDF	User Defined Function
TUI	Textual User Interface
VBM	Virtual Blade Model
List of notation	
Symbol	Designation and Unit
AOA / α	Angle of attack ($^{\circ}$)
a	Axial induction factor
$C_{(r/R)}$	Chord length (m)
$C_{L,D}$	Lift- and Drag coefficient
C_2	Inertial resistance factor (m^{-1})
D	Characteristic dimension (m)
f	External forces ($N.m^{-3}$)
$f_{L,D}$	Lift-and Drag forces/section ($N.m^{-2}$)
$F_{L,D}$	Lift and Drag forces (N)
k	Turbulent kinetic energy ($m^2.s^{-2}$)
Ma	Mach number
Δm	Thickness of the disk (m)
N_b	Number of blades
n	Factor of the power law profile
p	Pressure (Pa)
\bar{p}	Average pressure (Pa)
p'	Pressure fluctuation (Pa)
p_1	Normal pressure (Pa)
$P_{available}$	Available power (W)

List of notation	
Symbol	Designation and Unit
p_2	Pressure at the turbine (Pa)
Δp	Pressure drop (Pa)
P_{VBM}	Extracted power by VBM ($kg.m^2.s^{-3}$)
P	Power ($kg.m^2.s^{-3}$)
r	Minor axis length (m)
R	Major axis length (m)
R	Radius of the rotor disk (m)
Re	Reynolds number
S	Surface of a section (m^2)
$S_{i,j}$	Mean strain rate
S_{cell}	Sink term ($N.m^3$)
S	Additional momentum source
u	Velocity ($m.s^{-1}$)
U_{∞}, u_1, U_i	Free stream velocity ($m.s^{-1}$)
\bar{u}	Average velocity ($m.s^{-1}$)
u'	Velocity fluctuation ($m.s^{-1}$)
U	Velocity perpendicular to the turbine ($m.s^{-1}$)
u_2	Velocity at the turbine ($m.s^{-1}$)
u_3	Downstream velocity ($m.s^{-1}$)
\bar{v}	Average velocity perpendicular to a cross-section ($m.s^{-1}$)
v_{tot}	Velocity relative to the blade ($m.s^{-1}$)
v_c	Centreline velocity ($m.s^{-1}$)
v_y	y-component of flow velocity ($m.s^{-1}$)
x	Lateral coordinate direction (m)
y	Longitudinal coordinate direction (m)
z	Vertical coordinate direction (m)
$Z_{turbine}$	Distance from bottom to turbine origin (m)
Z_{depth}	Depth of AI (m)
α	Face permeability (m^2)
β	Blade pitch angle ($^{\circ}$)
θ	Angle ($^{\circ}$)
μ	Dynamic viscosity ($kg.m^{-1}.s^{-1}$)
ρ	Fluid's density ($kg.m^{-3}$)
ϕ	Twist angle ($^{\circ}$)
ψ	Rotor disk bank angle ($^{\circ}$)
ω	Rotor speed (rpm)



TABLE OF CONTENTS

1. INTRODUCTION.....	3
2. BACKGROUND OF THE STUDY.....	4
2.1 Description of the Turbine.....	4
2.2 Characteristics of potential sites of implantation (example of Admiralty Inlet).....	5
2.3 Physical notions (Navier-Stokes, RANS).....	6
2.4 Computational tools.....	8
2.4.1 <i>Virtual Blade Model</i>	8
2.4.2 <i>Actuator Disk Model</i>	10
3. ESTABLISHMENT OF A TURBINE ARRANGEMENT.....	13
3.1 Concept of the arrangement.....	13
3.2 Creation of the mesh.....	14
3.3 Implementation in FLUENT.....	15
3.3.1 <i>Inlet Conditions</i>	15
3.3.2 <i>Boundary Conditions</i>	17
3.3.3 <i>FLUENT Journal and c-script</i>	19
3.4 Exploitation of the results.....	20
3.4.1 <i>Tecplot script and Matlab program</i>	20
3.4.2 <i>Determination of the upstream plane</i>	21
3.4.3 <i>Normalization due to the shear velocity profile</i>	21
3.4.4 <i>Efficiency</i>	22
4. RESULTS.....	26
4.1 Influence of tip to tip distance on the upstream plane.....	26
4.2 Influence of the offset (at a given downstream distance).....	27
4.2.1 <i>Analysis</i>	27
4.2.2 <i>Explanation</i>	27
4.3 Influence of offset and downstream distance.....	28
4.4 Influence of tip-to-tip and downstream distance.....	29
4.4.1 <i>Analysis</i>	29
4.4.2 <i>Velocity profile near the turbine</i>	30
4.4.3 <i>Moments</i>	32
5. CONCLUSION.....	34
APPENDIXES.....	35
REFERENCES.....	52



1. INTRODUCTION

The idea of using energy from renewable sources such as the sun, wind or water seems to be very modern. Even though, already in the 5th century water mills used the kinetic energy of fluid flows in rivers or tides. During the past twenty years the development of renewable energies was mainly based on the exploitation of wind energy. Therefore, nowadays this technology is well known and applied in an industrial scale. Nevertheless the idea of extracting power from tides and rivers has moved back in the focus of industry and scientists. Apparently Marine Hydrokinetic turbines (MHK) are a promising future source of electricity. The amount of extracted power depends on the density of the fluid. Indeed the density water is thousand times bigger than the density of air. Moreover, the amount of electricity produced during one year can be estimated with a great precision. The tide is a cyclic highly predictable phenomenon contrary to the wind which is impossible to forecast one month in advance. However as the extracted power is directly proportional to the cube of velocity, it is important to find some places where the current during the tide is as strong as possible, like in the Race of Aldernay (Raz Blanchard) where the flow velocity can reach twelve knots, one of the strongest current of Europe.

As a part of a more general project, this study determines and optimizes the influence of different parameters on the efficiency of tidal turbine arrays. The final aim of the general project is to implement a field of Marine Hydrokinetic Turbines in the Admiralty Inlet, a side in the northwest of the state of Washington (USA). Because of the high costs of experimental research computational modeling is the most efficient technique to assess turbine designs and arrangements. Depending on design and the mean velocity modern turbines can generate between 1 kW and 3 MW. Therefore, to produce a relevant amount of power tidal turbines are organized in arrays.

In the continuity of the previous works of T. Javaherchi [1] [14], who investigated the feasibility of numerical modeling on wind and tidal turbines and the work of A. Gosset and G. Flouriot [2] who studied different configurations of arrangement of three or four turbines, this study defines the effect of each building parameters on the power extraction: the tip to tip distance of upstream turbines, the downstream distance and the offset of the downstream turbine.

The first step is to create a new mesh where the effect of each building parameters on the power extraction can be carried out separately. For that the mesh contains seventy turbines. In order to model a flow which approaches a realistic current profile, a shear velocity profile is used for the simulations. Post-processing the data of the simulations allows understanding the differences of power extraction of each turbine. Hereby the power extraction is defined as a function of each building parameter. Further investigations could have shown the environmental impact of turbine arrays in channels. Indeed two main aspects of the question appear. The first is the sedimentation process; in fact the wake caused by a turbine is able to change sedimentation transportation on the seabed significantly. The second is the impact on maritime species; the sudden change of pressure between intrados and extrados of a turbine-blade can affect for example small fish.

2. BACKGROUND OF THE STUDY

2.1 Description of the turbine

An economical application of Marine Hydrokinetic turbines (MHK) in a large scale is not realized. Industry and laboratory are still searching for the best way to extract tidal power from the ocean. The designs and the ideas behind tidal current energy converters vary a lot. In fact, four different types of converters with different fundamental concepts can be distinguished: Venturi Systems, variable foil systems (shown in Figure 1), vertical axis turbine and horizontal axis turbines [3].

The converter concept used for all simulations conducted in this study is the horizontal axis turbine. This turbine was developed as a wind turbine and tested in a wind tunnel in the NASA Ames Research Centre by the National Renewable Energy Laboratory (NREL). The data of these tests are freely available in contrast to many other concepts of private enterprises where the data remain unpublished. In this way, with the help of NREL phase VI data [4], researchers have the possibility to create computational models and to validate it with experimental results. Even though the hydrokinetic use of wind turbines implicates several conceptional changes especially due to different structural loading and the influence of cavitation, this turbine serve as model for the basic phenomena studied in this project. It has to be notified that the aim of this study is not to evaluate that one turbine geometry but to develop a reliable computational tool to understand the wake effects caused by horizontal axis MHK.

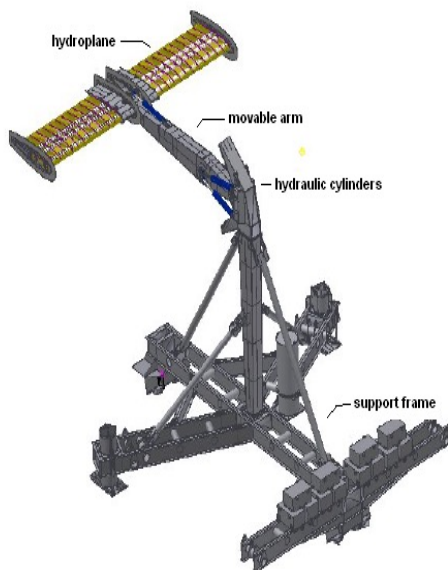


Figure 1 : Example of a variable foil system, Engineering Business Ltd

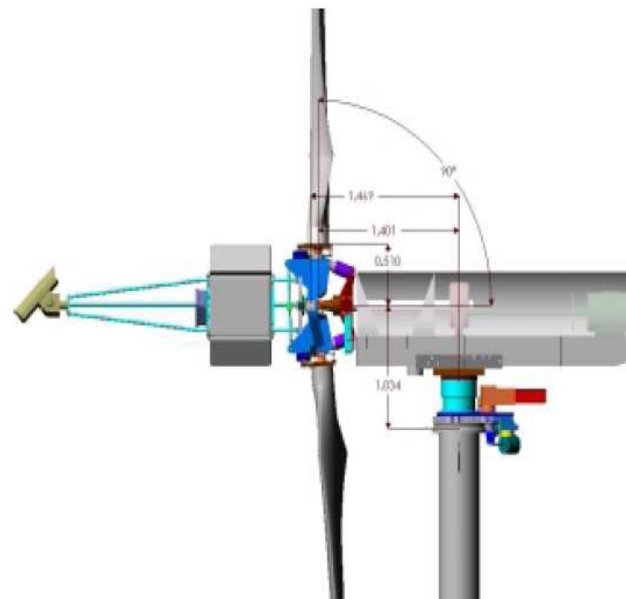


Figure 2 : Horizontal axis wind-turbine, NREL

As shown in Figure 2 the turbine consists of two blades with a radius of 5.53m. In the current simulations the rotor origin is at 16.59m over the bottom of the domain (the seabed in reality). In section 2.4.1, Virtual Blade Model, further geometrical parameters will be introduced.

2.2 Characteristics of potential sites of implantation (example of Admiralty Inlet)

Tidal movements can be observed all over the world as approximately 75% of the earth's surface is covered with water. These phenomena which are due to the gravitational attraction forces of the moon on the oceans appear in various ways. Semidiurnal tides are the most observed in European coast regions (see figure 3) whereas at the western north-American coast mixed tides occur more often.

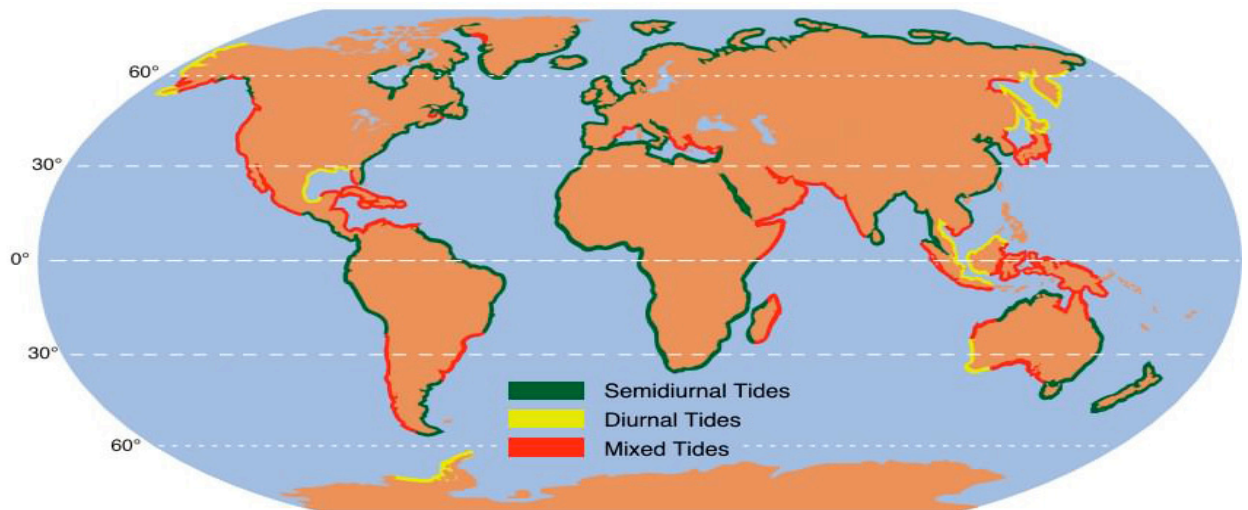


Figure 3 : Global distribution of the three tidal types [11])

Even though tides exist at almost every coast line not every place suits well for the implementation of MHK turbines. The available kinetic energy of a tidal flow in a section of the current channel is defined as follows:

$$P_{available} = \frac{1}{2} \rho S \bar{V}^3 \quad (1)$$

ρ is the density of water, S is the surface of the section and \bar{V} the average velocity perpendicular to the cross section. As the average velocity of the flow is cubed it contributes most to the available power of a potential side of MHK turbine arrays. Therefore potential sites have to be examined previously in order to determine all the parameters which are sensitive to influence power extraction capabilities of a turbine. The European Marine Energy Centre (EMEC) suggests a methodology for the assessment of Tidal Energy Resources [5]. Parameters such as mean velocity, turbulence intensity, vertical shear profile and directionality of flood and ebb flows constitute the site's bathymetry. Thus, feasibility studies based on the technical aspects carry out the efficiency of these places. But also other parameters such as the density and the types of maritime traffic or environmental issues are taken into account, especially the impact of underwater turbines and their rotating blades on fish and other maritime species.

One of these potential sites which were investigated is the Admiralty Inlet (AI) between the Strait of Juan de Fuca and the Pudget Sound next to Seattle (WA). The natural conditions are such that the flow velocity can reach over $3.25 \text{ m} \cdot \text{s}^{-1}$ [6]. Anyhow, most of the time flow velocity

can be estimated by 2 to 3m.s⁻¹. Tidal flow velocities change direction and magnitude periodically during ebb and flood. In reality, this movement cannot be described as a perfect bi-directional current. At AI the principle axis for the flood current is 143 degrees (counted clockwise from the North) whereas the main ebb flow direction is 300 degrees. In comparison with a theoretic one directional current the difference is about 23 degrees. The tidal ellipse serve as a representation (see figure 4) of the flow velocity (direction and magnitude) over one period.

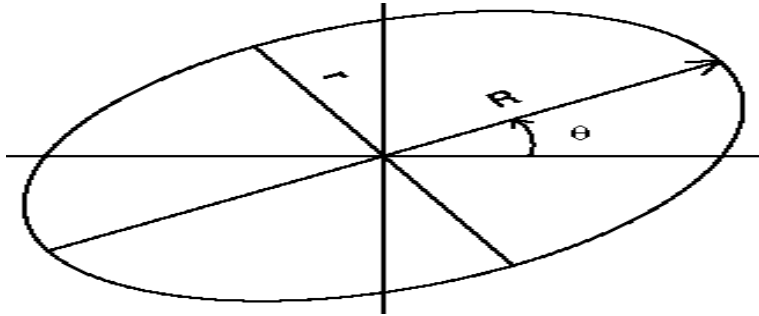


Figure 4: General tidal ellipse with a major axis (R) and a minor axis (r) and the orientation θ

Different directionalities of a tidal current can be a hindrance for horizontal axis turbine. Indeed, as the turbine's rotor section is placed perpendicular to the flow direction any changes of directionality would degrade the efficiency of the power extraction process.

Nevertheless, the simulations carried out for this study are configured in a way that the flow direction is perpendicular to the turbine surface as it should be ideally the case.

As given in Section 2.1, the center of the turbine is situated 16.59m above the ground. In order to allow swimmers or small boats to pass over the devices and to minimize damage caused by waves or floating materials EMEC suggest a top clearance to the surface of at least 5m [5, p 9]. Actually Admiralty Inlet has a mean depth of about 56m. So turbines centered at 16.59m over the seabed have a highest extent of 22.12m at the tip of the blade and an average top clearance of about 34m. All along the height of the water column the flow velocity profile is not unchanged. Due to the shear stresses of the flow and the proximity of the turbine to the seabed a shear-velocity profile must be established for a potential side.

2.3 Physical notions (Navier-Stokes, RANS)

Navier-Stokes equations are governing equations to describe the motion of Newtonian fluids. The high turbulent flow of this study is characterized with the help of these equations as well. The fluid on which they are applied is water with the density $\rho=998.2 \text{ kg.m}^{-3}$ and the dynamic viscosity $\mu=1.003 \cdot 10^{-3} \text{ kg.m}^{-1}\text{s}^{-1}$. These parameters are considered as constant. Furthermore, the fluid is supposed being incompressible:

$$\nabla \vec{u} = 0 \quad (2)$$

The Navier-Stokes equation in the case of this fluid is written as follows:



$$\rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u}\right) = -\nabla p + \mu \nabla^2 \vec{u} + \vec{f} \quad (3)$$

Where u is the velocity of the fluid and p is the pressure.

All the terms of the equations have a physical meaning and are defined as follows:

- the convective acceleration term (inertia force) $\vec{u} \cdot \nabla \vec{u}$
- the pressure gradient ∇p
- the viscous forces $\mu \nabla^2 \vec{u}$
- the external forces \vec{f} .

The behavior of a flow changes whether viscous- or inertia-forces have a dominating influence on the flow. The ratio of these two forces, so the Reynolds number (Re) is defined as follows:

$$\text{Re} = \frac{\rho U_{\infty} D}{\mu} \quad (4)$$

Where U_{∞} is the free-stream velocity and D is the characteristic dimension.

A low Reynolds number signify that the flow is laminar. The fluid flows in parallel layers, exchanges between them are rare. Viscous terms are negligible in comparison with the convective terms.

In the case of tidal current flows the Reynolds number is very high (about 10^8). The flow is turbulent. Its behavior is chaotically and the variation of pressure and velocity can occur very quickly.

Numerical modeling of flow-fields consists in predicting the behavior of the flow by solving Navier-Stokes equations. ANSYS FLUENT® is modifying these equations in order to time-average them and to split them from the fluctuating quantities, see equations (5) and (6). These modified equations are called Reynolds-Averaged-Navier-Stokes equations (RANS).

$$u = \bar{u} + u' \quad (5)$$

$$p = \bar{p} + p' \quad (6)$$

The velocity u and the pressure p can be expressed as the sum of an averaged term (\bar{u} and \bar{p}) and a fluctuating term due to the turbulence (u' and p').

By substituting equations (5) and (6) in the equation (3) we obtain the following RANS equation (in tensor notation):

$$\rho \frac{\partial \bar{u}_i}{\partial t} + \rho \frac{\partial \overline{u_i u_j}}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right) - \frac{\partial \bar{p}}{\partial x_i} + f_i - \rho \frac{\partial \overline{u_i' u_j'}}{\partial x_j} \quad (7)$$

Even though, this equation is similar to the Navier-Stokes equations (3) two more terms appear:

The mean rate of strain tensor $S_{i,j} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right)$ which represents the added term of the x, y

and z momentum equations and the non-linear Reynolds stress term $\overline{\rho u_i' u_j'}$ which can be predicted with a turbulence model in FLUENT.

The Shear-Stress-Transport (SST) $k-\omega$ model is the turbulent model used for the simulations of this project. After the $k-\varepsilon$ model it is the second most widely used two-equation model [7]. It was developed by Wilcox and other in 1993. As all two-equation models the transport equations are solved for two turbulence quantities, in this case k and ω . Timescales, lengthscales and other dimensional quantities can be formed by these two parameters. Even though the $k-\varepsilon$ model has a great range of applicability the $k-\omega$ model is more suitable for boundary-layer flows as it “is superior both in the treatment of the viscous near-wall region, and in its accounting for the effects of streamwise pressure gradient” [7, ch10.5.1 p384].

2.4 Computational tools

2.4.1 Virtual Blade Model

With the help of the software ANSYS FLUENT® which is specialized on the resolution of Navier-Stokes based equations, computational simulations of fluid flows can be realized. User-defined functions (UDF) allow the implementation of new user models or the modification of existing-ones. One of these UDFs is Virtual Blade Model (VBM). It enables ANSYS FLUENT® to calculate the extracted power of each tidal-turbine and to deduce its impact on the virtual flow-field. The circular rotor zone of a turbine is divided into twenty thin rings with momentum sources in the x, y and z directions. Each of these rings with different radii, refer to a predefined chord length and twist angle, the exact values are listed in Appendix A.

Based on the angle of attack (AOA) of the fluid just before the turbine, a special file, called s809 (see Appendix B), interpolates the corresponding lift- and drag-coefficients of the turbine. VMB averages the hydrodynamical effects of the rotating blades even without needing to model the exact geometry of the blades. This is the case in the Single Reference Frame model (SRF) where turbine's blades are drafted and

Figure 5: Rotor inputs in VBM on ANSYS FLUENT

meshed as wall boundaries with a no-slip condition so that their rotation imposes the rotation of the fluid.

Therefore, VBM represents a substantial gain of computational requirements and run time but also a simplification for the creation of the necessary mesh. Previous studies [1][14] have shown that even if VBM is able to model a realistic wake only at a distance of two radii downstream of a turbine, it furnishes a good approximation of the general flow-field studied in this project.

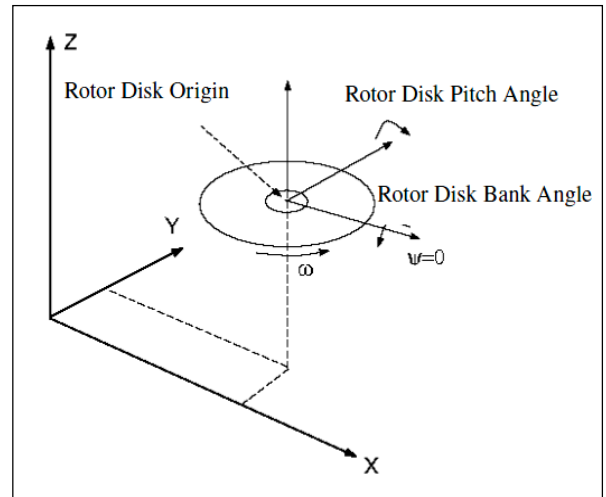


Figure 6 : Pitch and Bank angles

As seen in figure 5 VBM needs several input parameters concerning the relative position of the turbine in the flow. After setting the rotor disk origin, the turbine-disk should be perpendicular to the flow-velocity. For that the rotor disk bank angle (see figure 6) is set to 90 degrees. Another angle that has to be defined is the collective blade pitch angle. Since the turbines used in this study come up with blades which are firmly connected to the hub, all the other angles are set to zero.

Conforming to the NREL technical report and based on a previous study [2] and the angle sign convention represented in figure 7, the blade pitch and twist angle at the tip are respectively 3 and 2.5 degrees. As the collective blade pitch angle corresponds to the sum of these two, it equals -5.5 degrees.

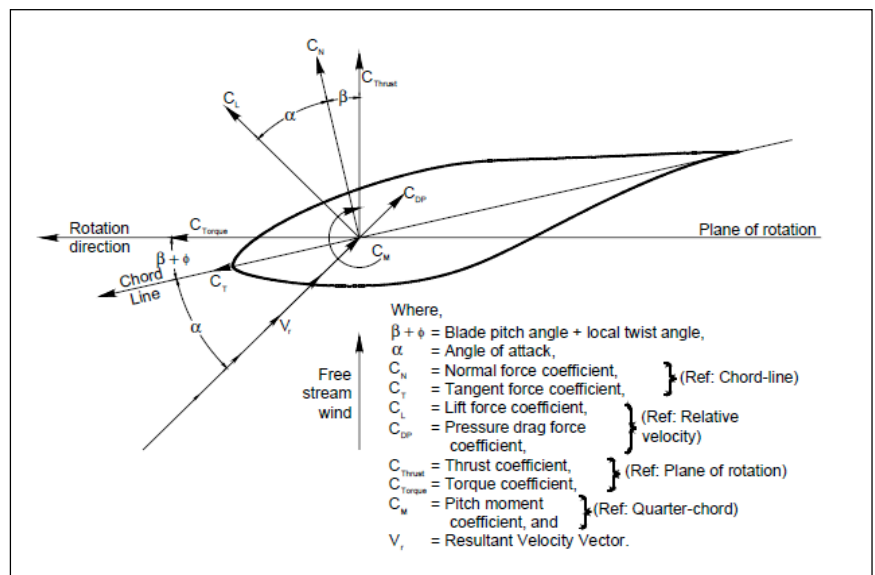


Figure 7: Forces coefficient convention

VBM calculates the Lift- and Drag-Forces for each section as following:

$$f_{L,D} = C_{L,D}(AOA, Ma, Re) \cdot c_{(r/R)} \cdot \frac{\rho V_{tot}^2}{2} \quad (8)$$

where $c_{(r/R)}$ is the chord length, ρ the density of the fluid, V_{tot} the velocity relative to the blade and $C_{L,D}$ the lift- and drag-coefficients determined as a function of Re , the number of Reynolds,

Ma, the Mach number and AOA which is defined as following:

$$AOA = \tan^{-1} \frac{U}{\omega R} \quad (9)$$

where U is the velocity perpendicular to the plane of rotation, ω is the angular velocity and R is the radius of the rotor-disk.

Finally, as the lift and drag forces are averaged over a full turbine revolution and time-averaged for each cell of the mesh, they are calculated as follows:

$$F_{L,D_{cell}} = N_b \cdot \frac{dr \cdot d\theta}{2\pi} \cdot f_{L,D} \quad (10)$$

The sink term for each cell is:

$$\vec{S}_{cell} = - \frac{\vec{F}_{cell}}{V_{cell}} \quad (11)$$

where V_{cell} is the volume of the grid cell.

At each calculation these forces are applied on the fluid. This iteration process continues until the convergence of the solution is ensured.

2.4.2 Actuator Disk Model

As seen in the previous section (2.4.1), VBM enables ANSYS FLUENT® to predict the lift- and drag-forces caused by a well-defined turbine. But these predictions are limited to the area swept by the blades and are not applicable to the hub-sections of the turbines. Due to its geometry, the hub produces still a relevant amount of drag but no lift. If VBM is used for both the rotor zone and for the hub zone, the total power extracted of this turbine would be overestimated. Therefore another modeling method must be applied to the hub-zone.

The Actuator Disk Model (ADM) stands for a further method of the simulation of hydrokinetic turbines. The hub is defined as a permeable disk placed perpendicular to the flow direction and supports a pressure difference, but not a velocity difference.

As represented in figure 8, where the whole turbine is modeled with ADM, the flow passes through the disk.

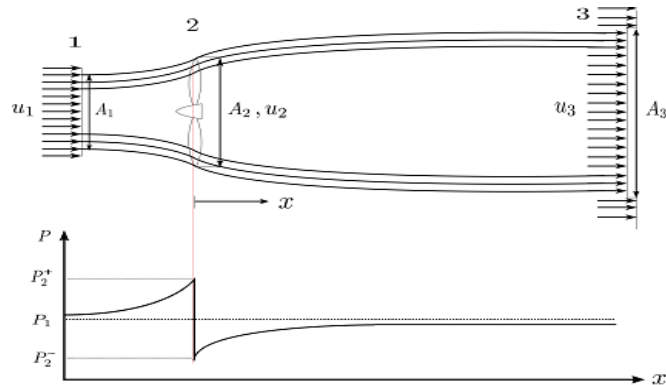


Figure 8: Streamtube around the actuator disk. Pressure drop by ADM



At the inlet of this stream tube, u_1 and p_1 are the velocity and the pressure of the unperturbed free stream condition. u_2 and p_2 represent the velocity and the pressure of the flow at the disk of the turbine, whereas p_2^+ is the pressure right before and p_2^- the pressure just behind the disk. Far downstream of this rotor zone, at the outlet, the pressure finds its ambient value of P_1 and the velocity becomes again constant on each plane of the tube.

Applying the Bernoulli equation on a streamline from a point of the inlet to another point just before the actuator disk (2^+) and from a point just behind the disk (2^-) to the outlet:

$$p_1 + \frac{1}{2} \rho u_1^2 = p_2^+ + \frac{1}{2} \rho u_2^2 \quad (12)$$

$$p_2^- + \frac{1}{2} \rho u_2^2 = p_1 + \frac{1}{2} \rho u_3^2 \quad (13)$$

As Δp is the pressure drop between p_2^+ and p_2^- ,

$$\Delta p = \frac{1}{2} \rho (u_1^2 - u_3^2) \quad (14)$$

In accordance to the theory of Froude the velocity at the actuator disk is the average velocity of the inlet and the outlet:

$$u_2 = \frac{1}{2} (u_1 + u_3) \quad (15)$$

The ratio of this diminution of velocity in comparison with the free stream velocity is called Axial Reduction Factor and is defined as:

$$a = \frac{u_1 - u_2}{u_1} \quad (16)$$

The relation of velocities at each cross section of the streamtube is given by equations (15) and (16):

$$\frac{u_2}{u_1} = 1 - a \quad (17)$$

$$\frac{u_3}{u_1} = 1 - 2a \quad (18)$$

In order to model the hub by means of the actuator disk theory as described previously it has to be implemented in ANSYS FLUENT®. This becomes possible by defining the hub as a porous media. During the simulations a momentum source is added to the standard momentum flow equation of the porous zone [9][10].

$$S = - \left(\frac{\mu}{\alpha} u_2 + C_2 \frac{1}{2} \rho u_2^2 \right) \quad (19)$$



The additional momentum S is composed of two terms on the right-hand-side of the equation: a viscous loss term, where α is the face permeability of the media ($\frac{1}{\alpha}$ is the viscous resistance factor) and an inertial loss term, where C_2 is the inertial resistance factor.

In the mesh the hub is represented as a thin disk with a thickness of Δm . Therefore the gradient of pressure drop over the porous media can be defined as:

$$\Delta p = -S \Delta m \quad (20)$$

Otherwise it can be written as following:

$$\Delta p = \left(\frac{\mu}{\alpha} u_2 + C_2 \frac{1}{2} \rho u_2^2 \right) \Delta m \quad (21)$$

The only user settings that have to be accomplished to launch the ADM simulations are the determination of the two coefficients C_2 and $\frac{1}{\alpha}$ which are unknown to FLUENT®.

Indeed, the axial induction factor is constant at each section of the streamtube and u_2 , u_3 and Δp can be calculated by varying the inlet velocity. A second degree polynomial regression of equation (22) in which u_2 is given in function of Δp permit to determine these coefficients.

$$\Delta p = K_1 u_2 + K_2 u_2^2 \quad (22)$$

Where $K_1 = \frac{\mu}{\alpha} \Delta m$ (23) and $K_2 = \frac{1}{2} C_2 \rho \Delta m$. (24)

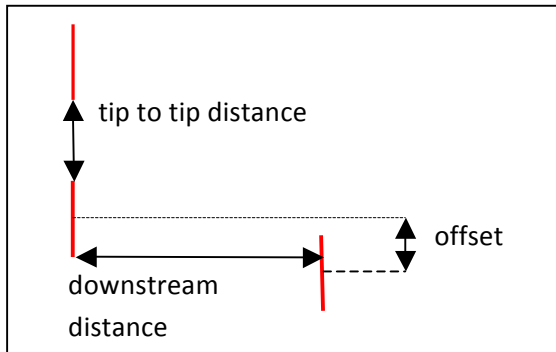
As mentioned in equation (17) a is a function of u_1 and u_2 . By consequence, a has not necessarily the same value for every device in an array of hydrokinetic turbines. Although former studies have shown that it is not necessary to define a new axial induction factor for each turbine. Therefore the values of inertial- and viscous resistance factors adopted in this study are the following:

$$C_2 = 5.633 \text{ m}^{-1} \quad \frac{1}{\alpha} = 2.5504 * 10^{-7} \text{ m}^{-2}$$

3. ESTABLISHMENT OF A TURBINE ARRANGEMENT

3.1 Conception of the mesh

The conception idea behind the mesh, built and used in this study, is to be able to alter several

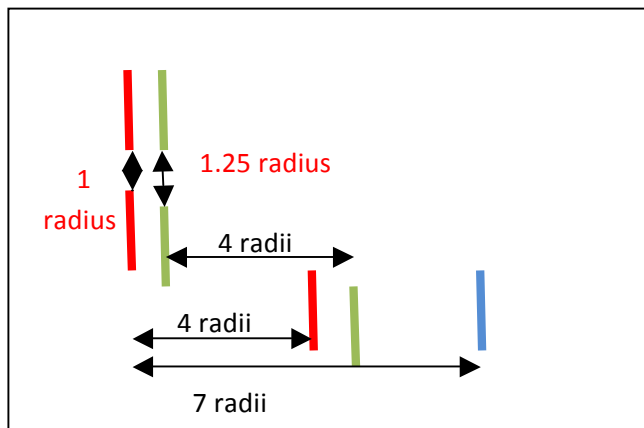


parameters of the array's geometry. There are three main parameters that can be changed: the distance tip to tip between two upstream turbines, the downstream distance, and the lateral offset for the downstream turbine (see figure 9).

At the beginning of this project, each simulation should run with only three activated turbines as shown in the figure 1. To understand the impact of

Figure 9: Main array parameters of turbines

each parameter, by comparing two simulations, only one parameter has to be changed. For example, as shown in figure 10, between the simulation with the green and red turbines there is only the tip to tip distance (in red) which



changes. Also, between the simulation with the red turbines and that with the two first red and the blue, there is just the downstream distance which changes. On those three simulations the offset is constant. That is why there are 70 turbines divided in three groups, as shown in appendix E.

Figure 10: Scheme of different simulations

tip distance of upstream turbines: 1, 1.25, 1.5 and 1.75 radius.

The first series of devices are the four couples of upstream turbines (green rectangular). They allow changing the tip to

The second part is the right-hand-side turbines which allow changing the downstream distance of the third turbine: 4, 5, 6, 7 and 8 radius (maroon rectangular).

Finally, the left-hand-side permits to play with the offset: -0.75, -0.5, -0.25, 0.25, 0.75, 1.25, 1.5, 1.75, 2, 2.5 and 3 radius (red rectangular). This offset represents the lateral distance between the origins of the upstream left-hand-side (LHS) turbines and the corresponding third turbine. A negative offset means the center of the downstream turbine is between the centers of the upstream turbines. A positive offset means that the center of the downstream turbine is between the LHS turbines and the LHS wall.

3.2 Creation of the mesh

The mesh was created with to the software Gambit®. The principle of the meshing is to divide all the volume into cells which should be as small as possible. This control of the size of the cells is the skewness. In order to have a faster convergence (number of iterations before the convergence) when simulation with fluids will run, the skewness has to be minimized. But improving the skewness means to have more cells and so computation with Fluent®, each iteration will take more time. On Gambit®, it is possible to see which elements have the worst skewness and where they are. So it is really important to have none of those elements in areas near the turbine where pressure, velocity and other data are changing greatly. The principle of the mesh is easy: a rectangular volume which is representing a portion of the tidal channel and

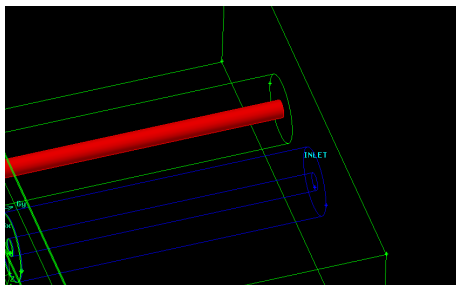


Figure 11: Constitution of the mesh

four pipes that go through the entire volume: two pipes with a bigger diameter and two smaller ones at the inside of the big ones. The small cylinders represent the hub of the turbine (red cylinder). The rotor-zones (turbines without the hubs) are represented by the area between the two coaxial cylinders (blue contour). The turbine is a

thin cylinder, the blue contour is just the mesh between two turbines (see figure 11). There are two different ways to create this mesh depending on if the turbines are coaxial or not. When they are coaxial the method of “Hex Wedge Cooper” is used. This method consists to mesh one face of the volume, called reference plane, as shown in

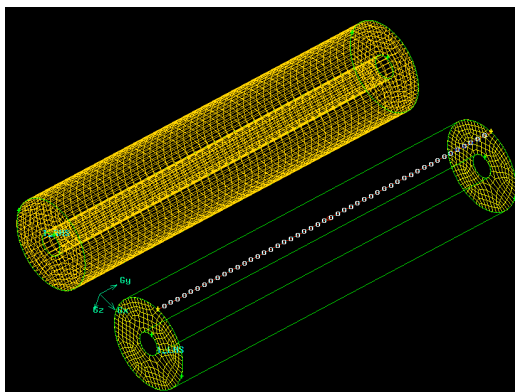


Figure 12: Hex Wedge Cooper method

figure 12 and to settle the numbers of node on a perpendicular side of the reference. With that, several planes parallel to the reference plane are created at each node. So the number of planes depends on the number of nodes. Finally, an orthogonal projection of the reference plane on the other planes is made (top cylinder of the figure 12) and the volume between two planes is meshed, so all the initial volume is meshed.

The main difficulty occurs when there is an offset between two succeeding turbines. Indeed the previous method cannot be used everywhere in the mesh: when there is an offset. So based on the previous work of Oskar Thulino, transition volumes have to be created. In those volumes there are no pipes. As shown in figure 13, where the red straights are the turbines, the pipes are ending on two faces of the transition volume.



Figure 13: transition volume of the mesh

To draw a mesh where each cell of the upper face on the figure has a corresponding cell on the lower face, the volume is meshed with an unstructured mesh (“Tet/hybrid Tgrid”) that means the computer tries to create tetrahedral cells with the lowest skewness possible. With this the user has a low control on the skewness in comparison to the “Hex/wedge Cooper” method.

In the conception, we want to have just one parameter which changes between two simulations. Due to this criterion all the turbines with different offsets corresponding to one tip to tip distance (blue circle in the appendix E) have to be on the same plane (y constant). It is not possible to do this on Gambit® because some circles overlap. So for the ten different offsets, five plans distant of 1.16 meters (0.2 radius) are created with two turbines in each. The plans are eight radius downstream the upstream turbines, so it is taken for granted that the criterion is respected. However those ‘transition volumes’ have to verify a specific ratio: the length over the width of the pave has to be superior than 0.0236. Because of this number it is impossible to create just one mesh due to the distance between some turbines. So two complementary meshes have been created: mario mesh and florian mesh, as shown in figure 14 and 15.

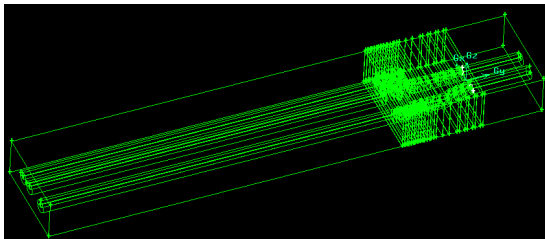


Figure 14: View of mario mesh on Gambit

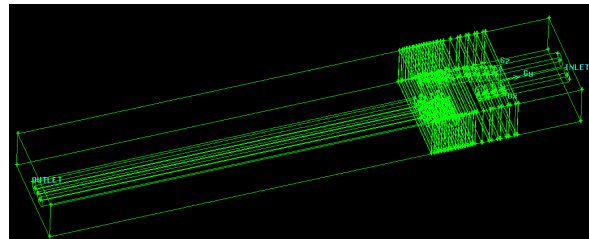


Figure 15: View of florian mesh on Gambit

Finally, the skewness of mario mesh is 0.823464 and florian mesh 0.899693. The difference of these two numbers is due to the transition volume where we have less control on the skewness and the difference of geometry between the two meshes (a more important offset). This skewness can be improved but the time spend on it will be too big in comparison to the results obtained (a simulation which converges faster).

3.3 Implementation in FLUENT

3.3.1 Inlet Conditions

The implementation of a mesh in FLUENT® needs to be completed by setting special conditions to the boundaries of the mesh. As it is described in paragraph 2.4.1 and 2.4.2 the rotor-zone and the hub-zone of a turbine are respectively defined as a VBM surface and a porous zone. Since further studies [1][2][14] employed constant velocity profiles to model tidal currents in channels (the velocity of the flow is the same in each point of a vertical plane) one of the aims of this project is to examine the influence of a shear velocity profile on turbine arrays. Because of the presence of the seabed shear stresses influence the flow near the ground. Due to the short distance of turbines to the bottom the velocity profile of the flow is still modified when it



reaches the turbine as we mentioned in section 2.2. This shear-velocity profile can be expressed in accordance to the power-law velocity profile for viscous flow in pipes [8 ch8.3.3 p426].

$$\bar{u} = V_c \left(1 - \frac{r}{R}\right)^{\frac{1}{n}} \quad (25)$$

Where \bar{u} is the time-averaged y component of the flow velocity, V_c is the centerline velocity, $\frac{r}{R}$ is the normalized z component and n is the factor of the power-law profile. The exponent of the power-law profile is increasing with the Reynolds number (Re). As well as Re is about 10^8 in the case of this study several approximations can be made for n. The assessment of the side of Admiralty Inlet [6] revealed that the velocity profile is varying with the tide. Although an average value for n of 10.1 was estimated for AI its standard derivation remains at about 5.7. Another approach for the value of n comes from the wind-turbine industry. As this economic sector is well developed their estimations of n are almost defined and based on the roughness of the ground [13]. Therefore, a one-seventh power-law profile is often used for many practical flows [12]. However, the difference between a one-seventh and a one-tenth power-law velocity profile is not substantially relevant against the background of this project, where n equals seven. Figure 16 represents the velocity profile applied to the inlet boundary of the mesh as a dark red line in comparison with the one-tenth power-law velocity profile. As the flow in our simulations is established in a rectangular volume where the origin of the coordination system is at the same vertical level than the origin of the turbine equation (25) has to be modified and becomes:

$$V_y(z) = -U_i \left(\frac{z + Z_{turbine}}{Z_{depth}}\right)^{\frac{1}{7}} \quad (26)$$

V_y is the y component of the flow velocity, U_i is the freestream velocity at the surface of the flow, Z_{depth} is the depth of AI and $Z_{turbine}$ is the distance of the turbine from the bottom. All the values of these parameters are listed in Appendix C. It shows the UDF (see paragraph 2.4.1) of this new shearvelocity profile. This UDF is necessary to implement the velocity profile in FLUENT®.

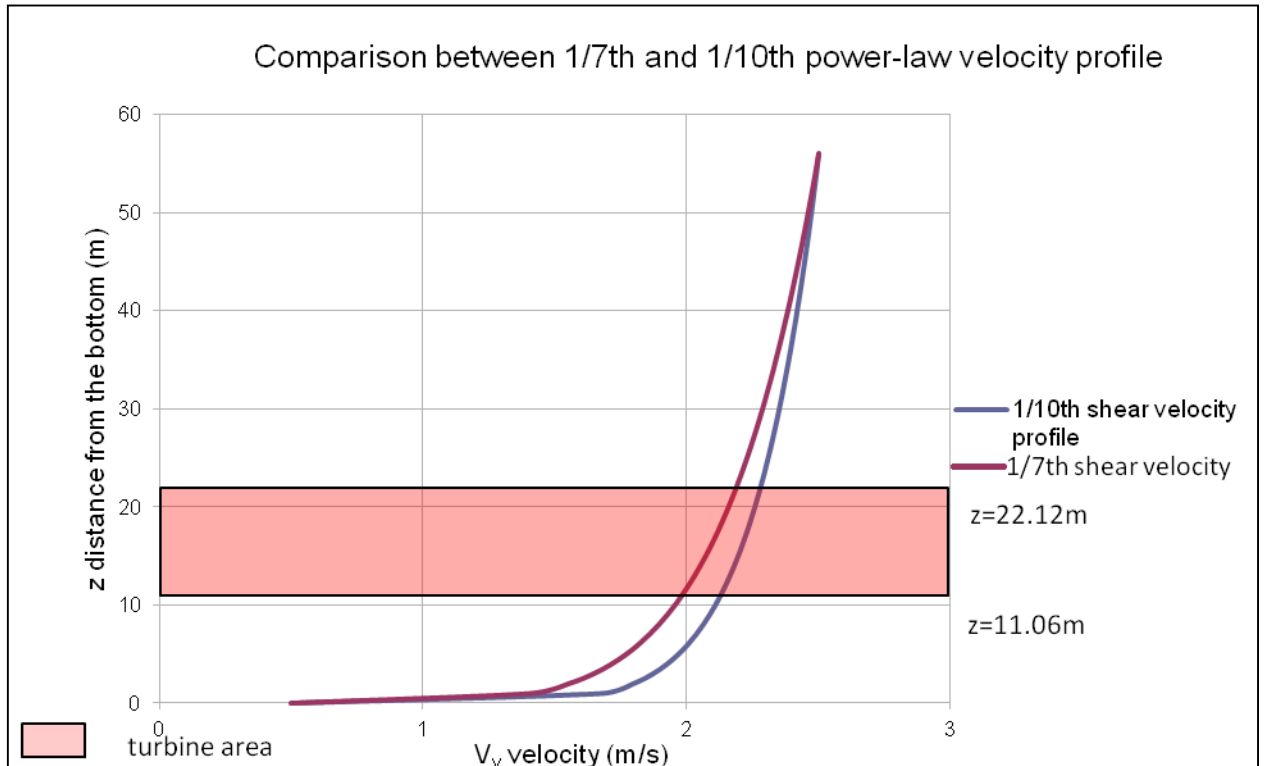


Figure 16 : Comparison between 1/7th and 1/10th power-law velocity profile. Note that V_y at the bottom equals $0.5\text{m}\cdot\text{s}^{-1}$. This value was set artificially in order to minimize the boundary-layer effect on the ground.

3.3.2 Boundary Conditions

All boundary conditions of the meshes are summarized in figure 17 below:

	Type	Momentum	Turbulence
Inlet	Velocity inlet	UDF	Intensity: 10% Length: 1m
RHS-wall			
LHS-wall			
Top-wall	Constant velocity inlet	$Y\text{-velocity}=-2.32\text{m}\cdot\text{s}^{-1}$	
Lower-wall		$Y\text{-velocity}=-0.5\text{m}\cdot\text{s}^{-1}$	
Outlet	Pressure outlet		<i>Backflow:</i> Intensity: 10% Length: 1m

Figure 17: Overview of the boundary conditions used for all simulations

All boundaries, except the outlet, are defined as velocity inlets. In order to have the same shearvelocity profile along z-axis at the outer borders of the mesh UDF had to be applied on

LHS- and RHS-walls. The top-wall is defined, with respect of equation (26) at the height of the top-wall at 33.18m, as a constant velocity inlet with a magnitude of y-velocity of $2.32\text{m}\cdot\text{s}^{-1}$. Also the lower-wall is defined as a constant velocity inlet. Although it seems more reasonable to define the lower-wall of a tidal current channel as a fixed wall since this boundary is the only one which represents a real physical obstacle (the seabed), additional simulations have shown the opposite. Actually, all the other boundary of the mesh (see figure 17) represent just the borders of the computational model which is used in this study. The comparison of simulations with different lower-wall boundary conditions (BC) with respectively wall-BC, symmetry-BC and constant velocity inlet is shown in figure 18.

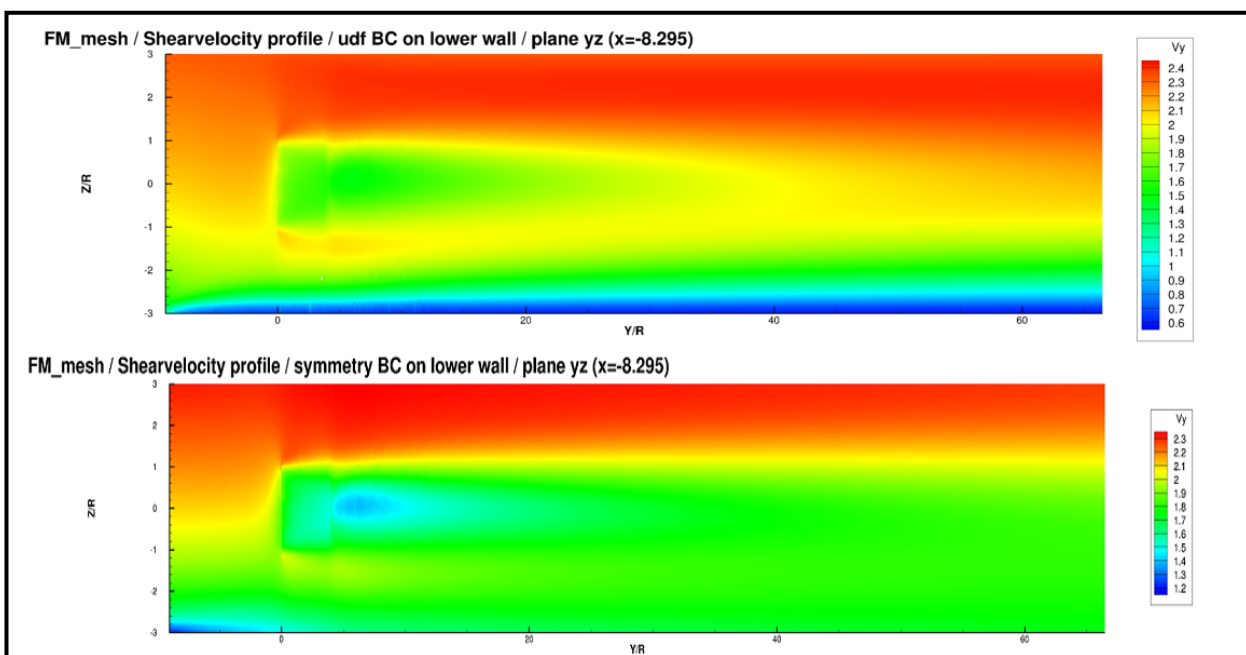


Figure 18 : Shearvelocity profiles with UDF-BC and symmetry BC on lower wall

Symmetry- and wall-BC are not conserving the initial shearvelocity profile which is imposed at the inlet. Indeed, the symmetry-BC is absorbing the lower velocities on the bottom due to its infinite momentum source “outside” of the mesh (see figure 19).

The only BC at the lower-wall that can be accepted because of its more realistic representation of a shearvelocity profile for the whole mesh remains the constant velocity inlet-BC.

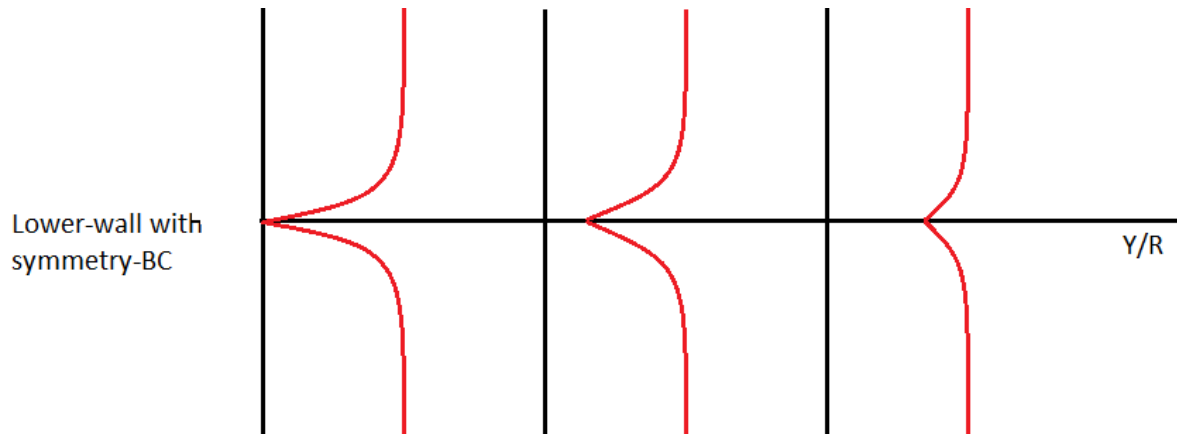


Figure 19: Scheme of the evolution of a shear velocity profile with symmetry –BC on the bottom

3.3.3 FLUENT journal and c-script

The computational models used for this project are VBM and ADM (see paragraph 2.2.1 and 2.2.2). These models are implemented in ANSYS FLUENT® and serve to produce flow-field simulations of MHK turbine arrays. The period of time that is needed to output a converged solution, so to produce one simulation, depends on the capacities of the processing unit of the computer. Obviously these capacities are limited and not available all the time also they have to be shared when several programs are executed at the same time. The server that was at our disposal has eight cores each with a clock rate of about two GHz. In ideally conditions one simulation (the convergence occurred after 1200 iterations) needs a computational time of about four hours. For the purposes of this study and because of the global turbine arrangement of the meshes more than 80 simulations had to be done. All the simulations are coming up with different turbine arrays or with different BCs therefore each of it has to be configured separately.

Due to the great amount of simulations which had to be run it was necessary to automate some of the configuration steps in order to obtain a better time-management of the computational capacities. The software FLUENT® offers two options to automate the configuration process of a simulation: to edit a journal file for a Graphical User Interface (GUI) or for a Textual User Interface (TUI). The FLUENT® user’s guide describes a journal as a “sequence of ANSYS FLUENT® commands, arranged as they would be recorded interactively into the program or entered through the GUI or TUI. The GUI and TUI commands are recorded as Scheme code lines in journal files.” As TUI commands are not able to launch or to modify UDFs, such as VBM, only the GUI method was used to create journal files. An example of GUI code lines is available in Appendix D.

One simulation needs one specific journal file where for example the coordinates of the turbine that has to be activated are defined. Even though a predefined case- and data-file, called mother-file, is always loaded at the beginning of every journal. This mother file should be predefined and compiled as follows:



- load mesh
- define viscous model
- define Hooks-function
- add zone-type --> fluid
- set all Cell-Zones-Types from solid to fluid
- set all Cell-Zones that will be Rotor-Zones to 'Source-Terms' and define X-, Y- and Z-Momentum
- unset all Rot-Cell-Zones (X-, Y- and Z-Momentum will be saved even if it is not shown)
- set Solution Methods
- set the Residuals-Monitor

NOTE: GUI journals of FLUENT® cannot set more than one Rotor-Cell-Zone to 'Source-Terms' and define at the same time X-, Y- and Z-Momentum, it is only possible to activate or deactivate 'Source-Terms'.

FLUENT® can only read one journal file at a time. By using a c-script and the following command (a) several journals can be launched one after another.

```
system("fluent 3ddp -t6 -i fm_journal_4_2p150.jou");           (a)
```

This command launches FLUENT's the three-dimensional, double-precision solver (*fluent 3ddp*), by using six cores of the processing unit (*-t6*) and executes (*-i*) the journal *fm_journal_4_2p150.jou* in a GUI-mode.

Finally, this method allows using computational time in a more efficient way without the need to launch manually all the simulations. Especially during the night or over weekends, so during periods with low server utilization, this brings noticeable advantages.

3.4 Exploitation of the results

3.4.1 Tecplot® script and Matlab® program.

To calculate a local efficiency and to understand the differences of power extraction of the turbines given by VBM, the velocity of the flow has to be known downstream and upstream the turbine. As VBM averages the turbine velocity on its entire surface, we do not have to know the velocity on a point but the average velocity. The average velocity is an average of all the velocity in the projection of a turbine in one plane. It is taken for granted that this definition of average velocity applied for the whole report. For that, the software Tecplot® is used (see appendix F).

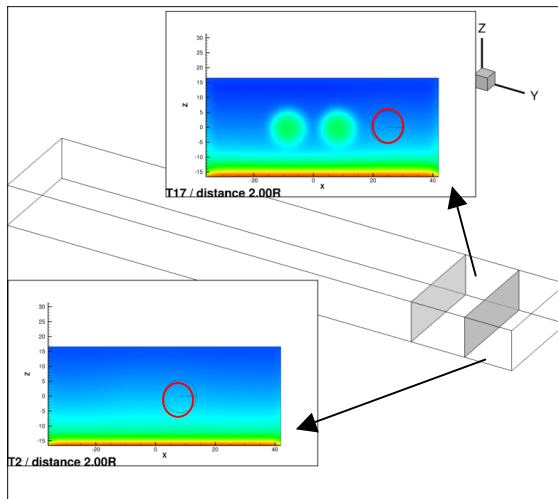


Figure 20: Tecplot export data

A script of Oskar Thulino permits to automate some steps. This script was modified to fit with thirty-eight turbines. Depending of the settings of the user, the script creates several plans before or after a turbine, and in each plane the orthogonal projection of the turbine desired is designed (red circle) and the values wanted are extracted on a data file. In the figure 20, the extracted data are two radii upstream the turbines. After the data have been extracted by the Tecplot® script, a Matlab® program of Oskar Thulino is used (see appendix G). This program calculates the average velocity, the available power and the thrust on

different planes. For some special needs of this study, other functions have to be added to this program. Indeed, major modifications to calculate the power available in different quarters of the turbine (red in the appendix G) and to calculate the average velocity in different rings with several radii (green in the appendix G).

3.4.2 Determination of the upstream plane

To explain the power extracted by each turbine, the velocity, the power available in a plan before the turbine has to be known. The choice of this upstream plane is based on the profile of the velocity in front of the turbine. Indeed when the fluid comes to the turbine, it ‘sees’ the turbine and the velocity decreases due to the resistance of the turbine to the flow. Additionally the upstream plane should not be too far from the corresponding turbine because of the geometry of the mesh which imposes only little downstream distances between turbines. Whenever the plane will be too close to another activated upstream turbine the wake effect would influence the data extracted from the plane in an unwanted way. The appendix H is a chart which shows the average velocity versus the downstream distance for several turbines with different downstream distances an offset of one radius and a tip to tip distance of one radius. Almost all the curves except one are showing similar evolutions, a constant velocity far from the turbine and a sudden decrease. The curve for the 4 radii downstream turbine does not reveal this part of constant velocity because at 4 radii upstream there are the upstream turbines, so this is why the velocity is a bit higher and decrease to reach a constant velocity of 2 m/s. For this study we choose an upstream plane at 2 radii. Although, at this distance the fluid has already ‘seen’ the turbine and the velocity has decreased of 0.45% this remains a good approximation of an unperturbed flow.

3.4.3 Normalization due to shear velocity profile

With the new profile of velocity, a boundary layer appears, as shown in figure 21, the blue line in the bottom. As the boundary layer grows with the distance, the velocity in the middle and on

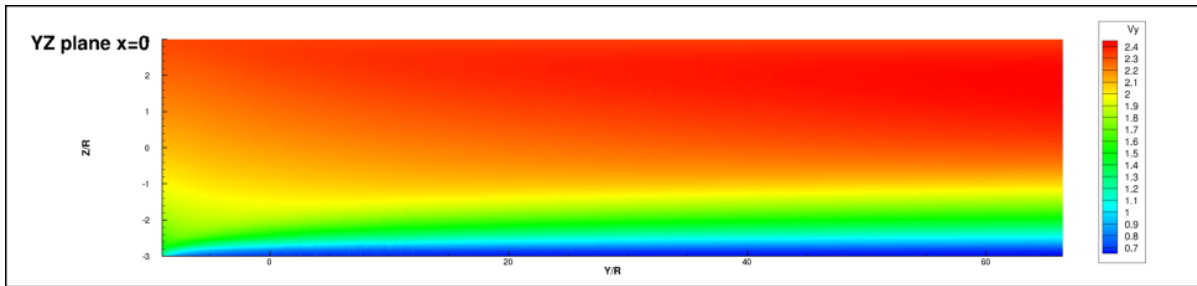


Figure 21: simulation without turbines, YZ plane x=0

the top of the field increases too. Indeed, one of the principles of the simulation on Fluent® is the conservation of mass.

$$\rho * V * A = cst \quad (27) \quad \text{where } V \text{ is the velocity of the fluid and } A \text{ the cross section.}$$

As ρ and A are constant, the velocity has to increase in the middle and top in order to offset the diminution in the boundary layer, as shown in figure 22. As it was mentioned in section 3.2, due to constructional differences between mario mesh and florian mesh the skewnesses of the meshes are not the same. So one of the preoccupations was to know whether the domains show also differences in the velocity profile or not. As shown in figure 22, the two curves are

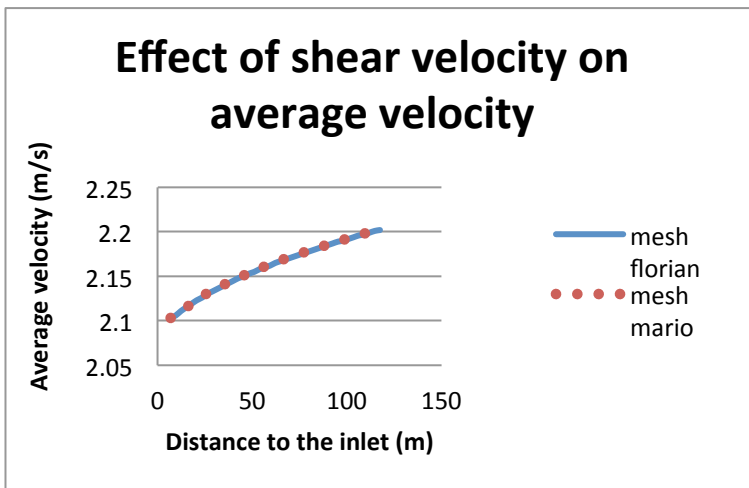


Figure 22: Effect of shear velocity on average velocity

matching perfectly also there are no problems to compare the power extracted between the turbines of the two meshes. However as shown in figure 23, there is a difference of power available for the turbines upstream and downstream of 150 kW that is not negligible. Because of this difference it is impossible to compare the power extracted by different turbines in different Y planes. For example, the farthest

turbine extracts 15 kW more than the closest turbine of the inlet. This is a not physical because the fluid cannot gain energy when it flows through the mesh because there are no sources. Obviously the best array is when all the turbines extract the same amount of power, the downstream turbines have no loss in comparison with the upstream turbine. To be able to compare the power extracted by the turbines and to see the effect of each parameter on it, the

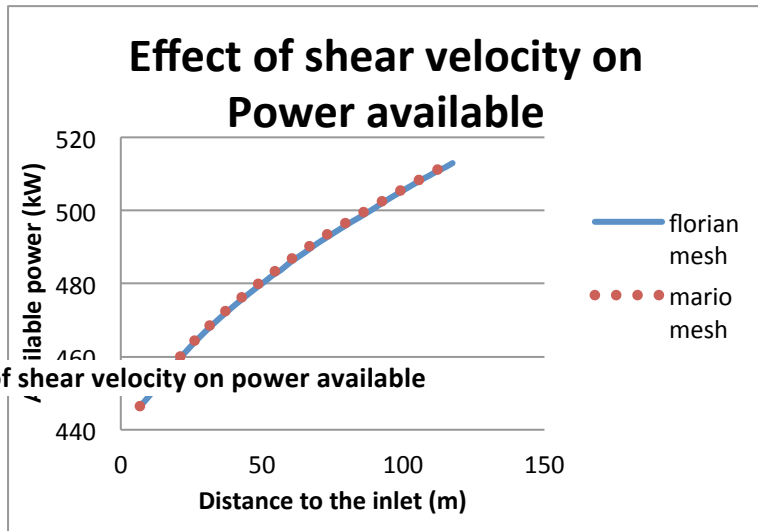


Figure 23: Effect of shear velocity on power available

extracted power has to be normalized. It has to remove the effect of the non wanted downstream distance. The normalization is the ratio between the

power extracted by the turbine when the simulation runs with three turbines over the power extracted by the turbine when the simulation runs with only this turbine. So this ratio is a kind of

loss of power of the turbine between the optimal condition (turbine alone) and the condition in a field of turbines. So to optimize the array all normalized power has to be equal to 1. During the project it was impossible to run all the simulation with just one turbine, so the efficiency was used to define the power extracted by a turbine alone.

3.4.4 Efficiency

The figure 24 shows the wake created by a turbine. On the plane XY, the axe of the turbine is an axe of symmetry for the wake. Several radii after the turbine the flow begins to reaccelerate

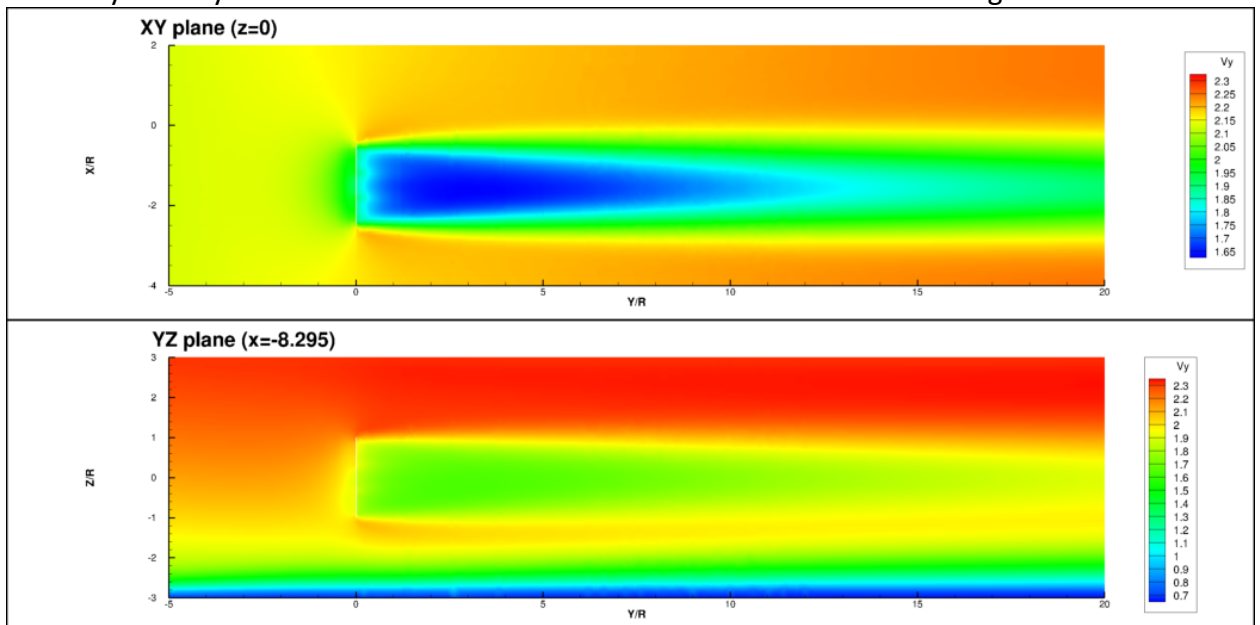
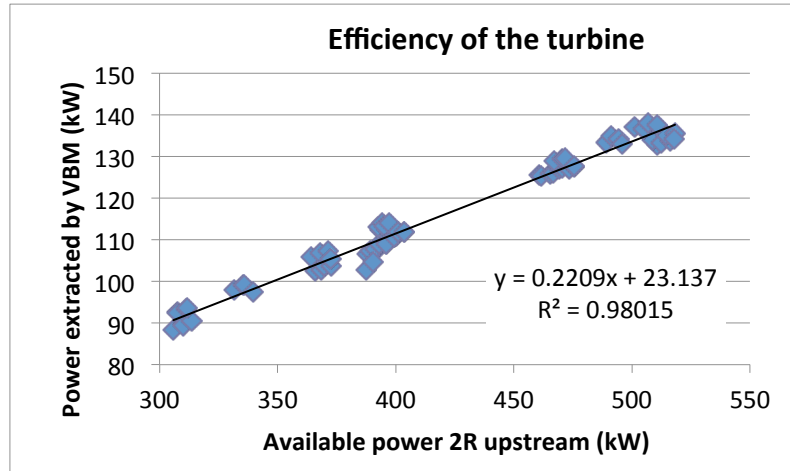


Figure 24 View of the wake of one turbine

due to the mix with the free stream flow. However the thickness of the wake decreases slowly.

The efficiency of a turbine is defined as the ratio between the power extracted by VBM and the power available two radii upstream. The figure 25 gathers the data of sixty simulations with three activated turbines. All the points are almost aligned and the linear regression gives a coefficient of correlation of



0.9801. The efficiency of the turbine is almost 22 percents.

Figure 25 Efficiency of the turbine

With this linear equation (28), we can calculate the power extracted by a turbine if we know the power available two radii upstream. Indeed we ran one simulation without any activated turbines and we extracted the available power in plans 2 radii before the turbines. And thanks to the linear equation we can calculate the power extracted by one turbine alone.

$$P_{VBM} = 0,2209 * P_{available\ 2R\ upstream} + 23,137 \quad (28)$$

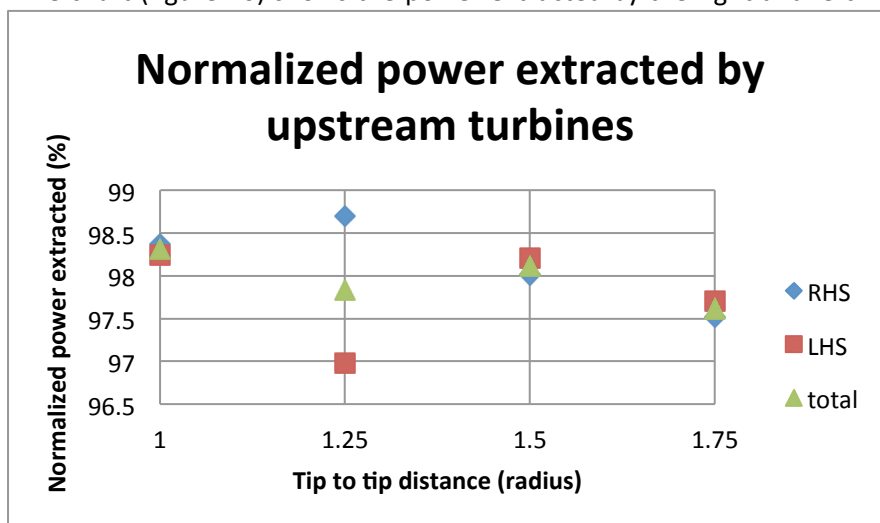
Using this equation, in average the relative error between the extracted power calculated by VBM and the estimated power is less than one percent (0.97 %).

4. RESULTS

4.1 Influence of the tip to tip distance on the upstream turbines

Between the two upstream turbines, the fluid is accelerated. This acceleration depends on the tip to tip distance. Actually when the distance increases, the acceleration decreases.

The chart (figure 26) shows the power extracted by the right and left hand-side-turbines for several tip to tip distances. The power extracted by the right-hand-side turbines increases with the distance to reach a maximum for 1.25 radius and decreases afterwards.

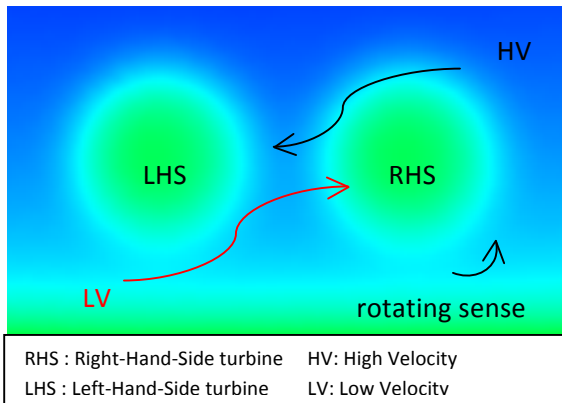


The contrary is the case for the left-hand-side turbines. Between 1.25 radius and 1.5 radius, there is an inversion between the RHS and the LHS turbines. Actually for a tip-to-tip distance smaller

Figure 26 Chart of the effect of tip to tip distance on upstream turbines

than 1.5 radius the right-hand-side turbines are extracting more power than the LHS devices This trend is inverted for bigger tip-to-tip distances. That is due to the rotational movement of the blade, and the migration of lower velocity from the ground.

As seen in figure 27 the sense of rotation of the turbines is counter-clockwise when looking from the inlet. This figure represents the effects of the rotation of the blade on the flow. Due to



the shear velocity profile, the velocity on the top is bigger than it on the bottom, a difference of 1 m/s As the RHS-turbine is rotating counter-clockwise higher velocity from the upper part of the domain is brought down towards the LHS-turbine. This migration of the flow is represented by the black arrow in figure 27. The red arrow is showing the opposite phenomena; lower velocity migrates to the RHS-turbine. Therefore, the power

Figure 27 Move of the flow due to the rotation of the turbines

extracted by the RHS-turbine is lower than than the power extracted by the LHS-device. Indeed at

1 radius tip to tip distances, the low velocity send by the LHS turbine and the high velocity send by the RHS turbine are almost compensating even if the RHS turbine produces more power, as seen in figure 26. The 1.25 radius tip to tip distance is a critical distance as it shows the figure 26. After this critical point, the turbines are too far, so the low velocity influences less the RHS turbine but the high velocity continues to influences the LHS turbine.

4.2 Influence of the offset (at a given downstream distance)

4.2.1 Analysis

This chart (figure 28) is the result of the simulation with upstream turbines and the left-hand-

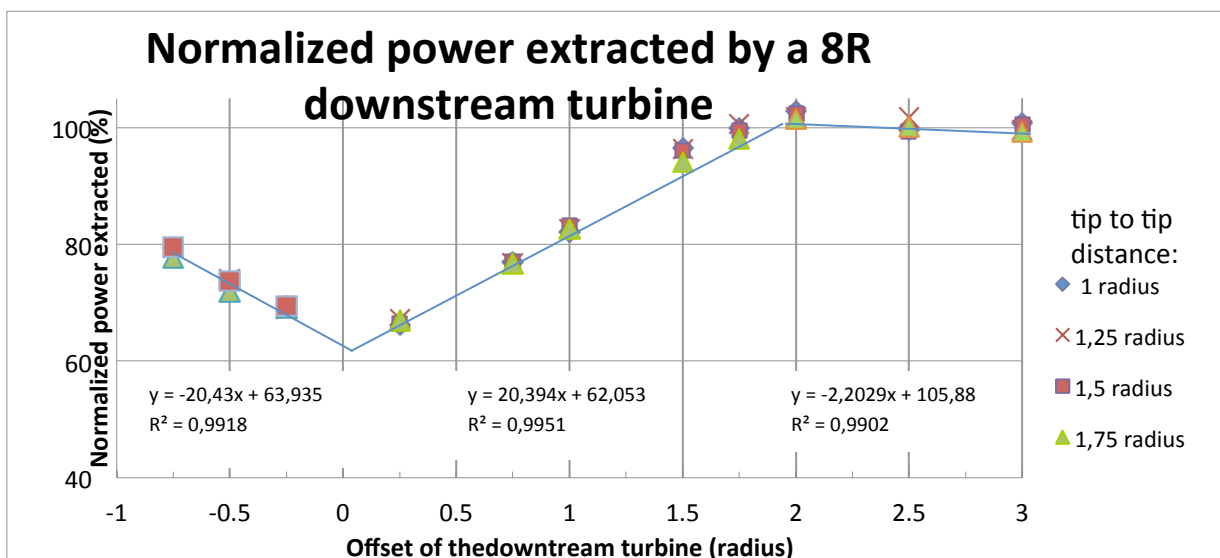


Figure 28 Chart of the effect of offset and tip to tip distance



side downstream turbines. Between each simulation just one parameter is changing, here tip to tip distance or offset. The downstream distance of the third turbine is eight radii for each simulation. As the plots are stacked, the effect of the tip to tip distance is negligible in comparison with the offset. The extracted power decreases when the offset tends toward zero. For an offset of 2R and more, the extracted power is constant. There are three different parts: the left part containing the three negative offsets, the central part containing the offsets between 0.25R and 2R and the last with offsets of 2 R and more. Those three parts can be approximated by linear equations.

$$P = -20,43 * x + 63,935 , x \in [-0,75; 0] \quad (30)$$

$$P = 20,394 * x + 62,053 , x \in [0; 2] \quad (31)$$

$$P = -2,2029 * x + 105,88 , x \in [2; 3] \quad (32)$$

For the same offset (0.25R and 0.75R), the power extracted is not the same if the offset is positive or negative. Indeed for negative offset the extracted power is bigger.

4.2.2 Explanation

The increase of power with the offset is due to the wake. The velocity on a Y axis is the lowest in the center of the wake created by the upstream turbine and increase to reach a maximum outside the wake where the flow reverts its free stream velocity. That explains the increase of power extracted with the offset.

The power extracted by the turbines with a negative offset (see paragraph 3.1) is bigger than it is with a positive offset of the same amplitude (average difference of 3%). This difference is due to the acceleration of the fluid between the two upstream turbines.

4.3 Influence of offset and downstream distance

The figure 29 is the result of the simulation with upstream turbines and the left-hand-side downstream turbines. Between each simulation just one parameter changes, here the downstream distance or offset. The tip to tip distance of the upstream turbine is 1.75 radius for each simulation.

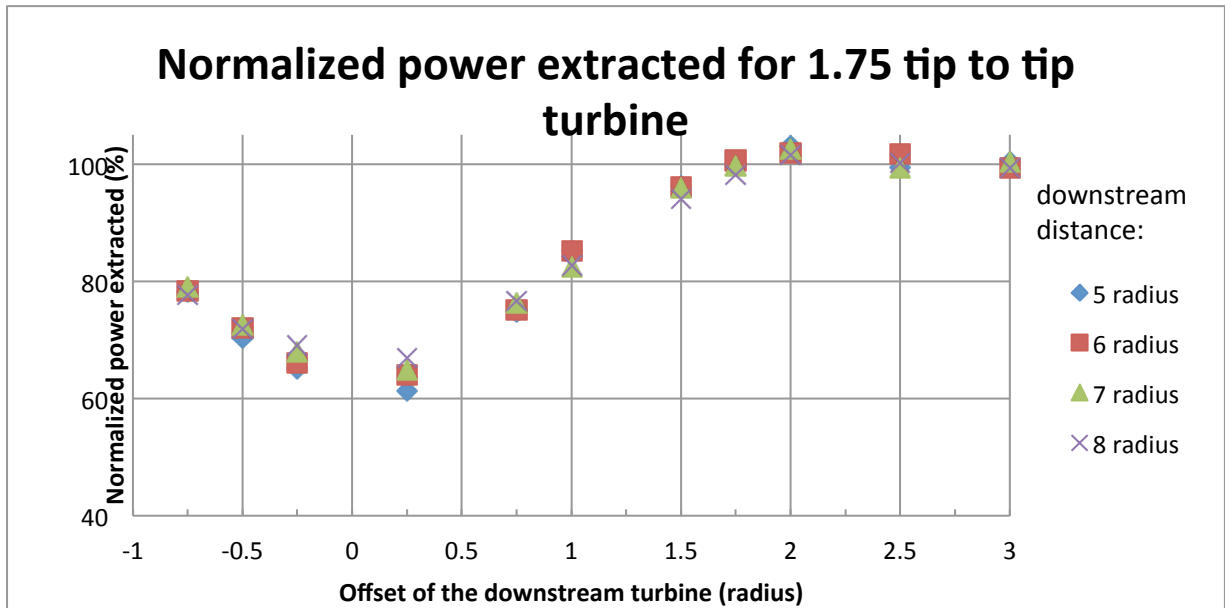


Figure 29: Chart showing the effect of downstream distance and offset

All the marks for a constant offset are superimposed. When the offset is small the 8 radii downstream turbines produce more power, but when the offset is bigger the rank is reversed and the 5 radii downstream turbines extract more power. Nevertheless, the downstream distance effect is negligible compared to the offset effect.

4.4 Influence of tip-to-tip and downstream distance.

This chart (figure 30) is the result of the simulation with upstream turbines and the right-hand-side downstream turbine. In this simulation either the tip to tip or downstream distances changes. The offset of the third for each simulation is one radius.

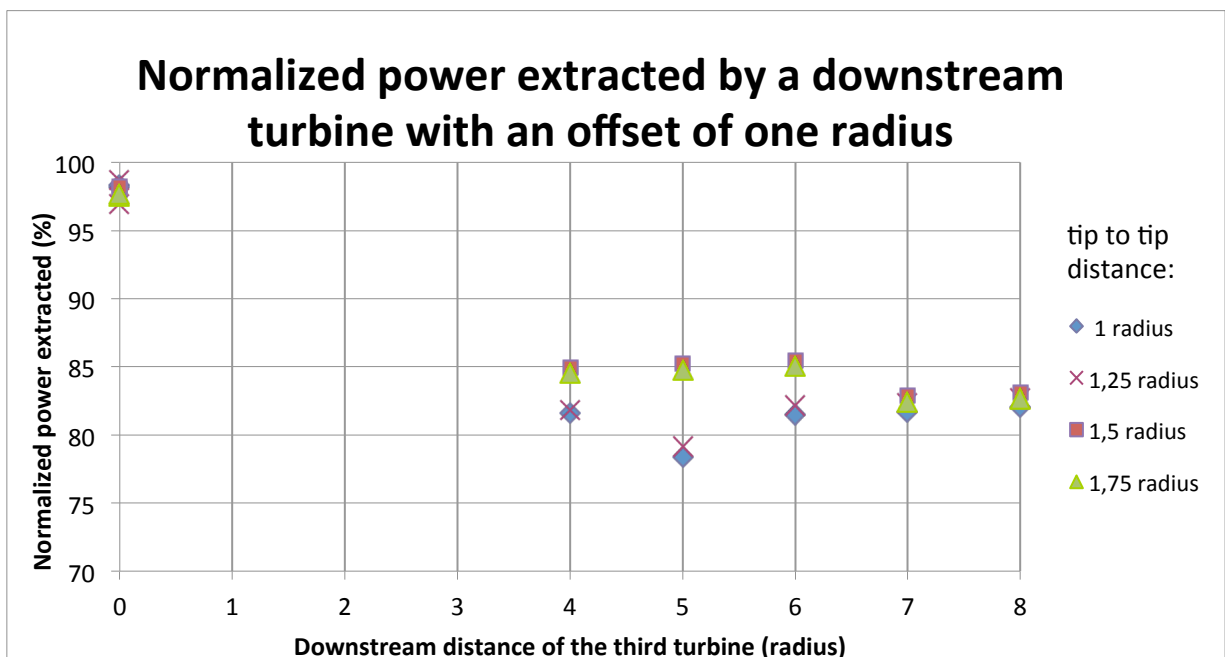


Figure 30 Chart representing the effect of downstream and tip to tip distances

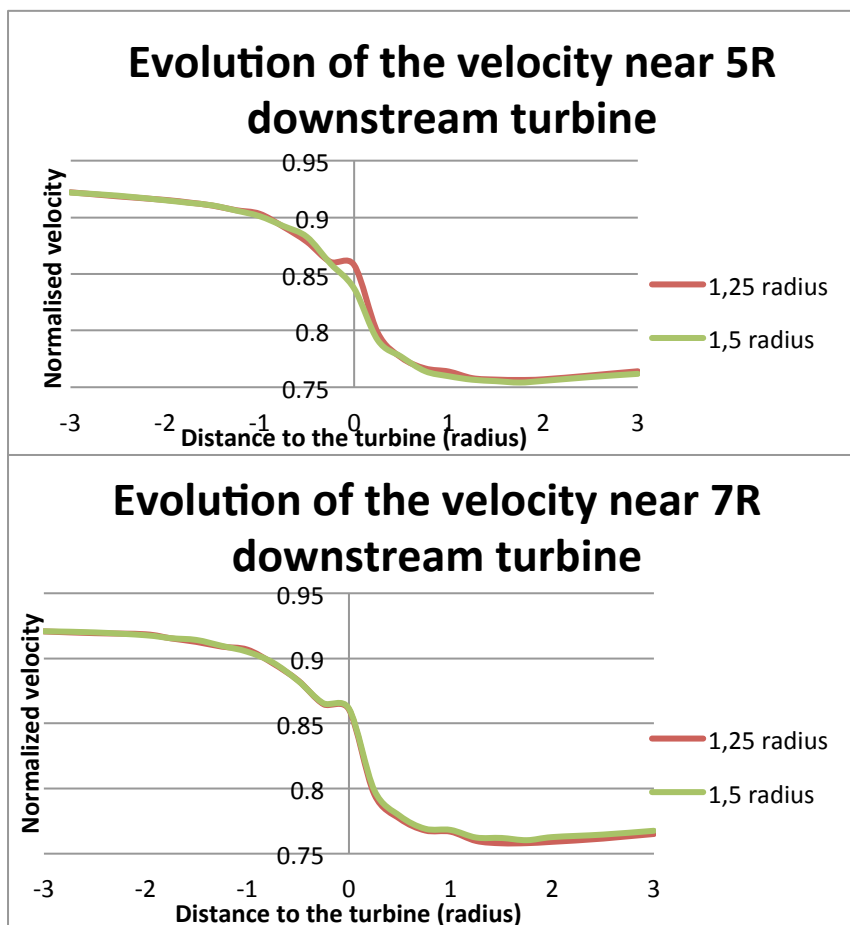
4.4.1 Analysis

At each downstream distance, the different tip to tip distances are ranked in the same order. The extracted power is increasing when the tip to tip distance is increasing too (vertical line on the figure 30). There are two groups, the first with the smallest tip to tip distance and the second with the largest. The difference between those two groups is about three percents (an amount of power of 4kW). This difference reaches a maximum of six percents (8kW) for a downstream distance of five radius and trends to approach zero. After seven radiuses, tip to tip distance has no effects anymore on the extracted power. At one given distance there is an important loss of power extraction: at 5 radii for 1 and 1.25 radius and at 7 radii for 1.5 and 1.75 radius.

The global trend for each tip to tip distance is a small augmentation of the power extraction with the increase of the downstream distance. However the normalized power extracted can be considerate as constant: eighty-two percents.

To explain those results, especially the difference of power extraction between the two biggest and the two smallest tip to tip distances, the velocity profile near the turbines has to be studied.

4.4.2 Velocity profile near the turbine



The charts (figure 31a and 31b) represent the velocity near the 5 and 7 radii downstream turbines. The velocity decreases when the fluid approaches the turbine. Behind the turbine the flow velocity is still decreasing; the turbine is extracting the available kinetic energy of the flow. After two radii the flow begins to mix with the free stream flow and so it reaccelerates. There are two types of curves one really smooth and another with a kink on the turbine plane. Actually we can see this type of smooth curve

Figure 31 Charts of the evolution of the velocity near the turbine 5R (a) and 7R (b) for several

turbines, see appendix I. The figure 32 (a and b) which shows the velocity profile near 5 and 7 radii turbines in several planes (x or z constant) explains why there are two types of curves. The red circle on the figure

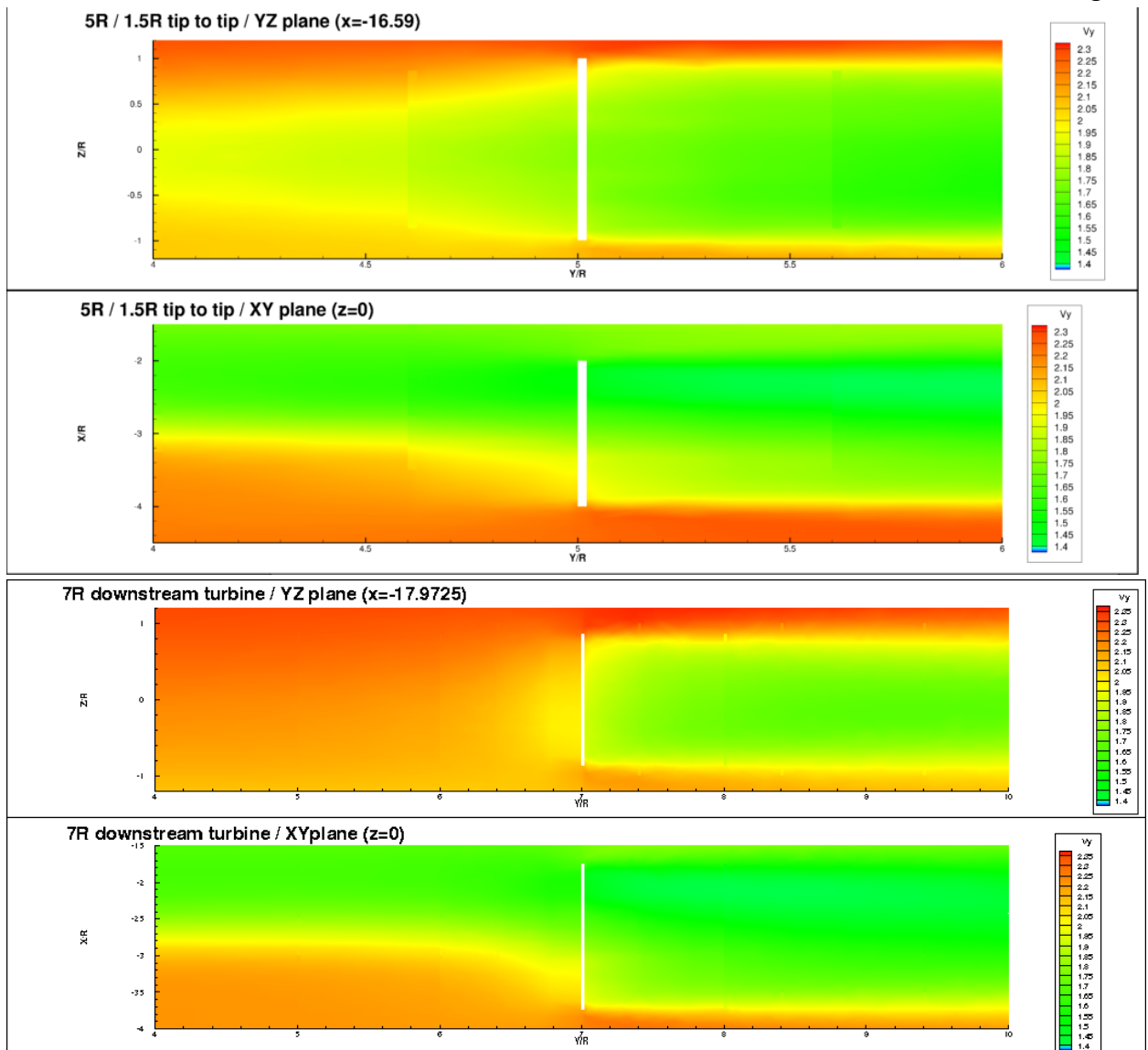


Figure 32 Velocity profile near 5 (a) and 7 (b) radii downstream turbine

shows a discontinuity in the velocity. This discontinuity corresponds to a turbine located 0.4 radius before. This turbine is not activated but it is still present in the mesh of the domain. This discontinuity causes a lower velocity on the turbine when the velocity that goes through the turbine is higher the power extraction is less efficient. Actually, the fluid has more easiness to go through the turbine so less resistance from the rotor and so the drag coefficient is lower. Turbines which have a deactivated device at 0.4 radius in front of them, their power extraction is higher than it is for the turbines that have not. The chart (figure 33) plotting some results of the simulations of Oskar Thulino. We used the results of those simulations in order to know if there was a problem of our mesh or something else. On this chart (figure 33), all the curves have a kink because there are no ,phantom'-turbine at 0.4R in front of the activated turbines. Even though the kink is not happening in realty, this shape of curve is considerate as the reference. This choice is due to the fact that even the domain of Thulino, which is less complicated as it contains only eight turbines, shows this kink-effect. There are several possible explanations for the

problem with the kink. Probably the way how the turbine-conditions are defined (with the help of VBM and ADM, see 2.4.1 and 2.4.2) causes the problem. So the figure 34 shows the normalized power extracted by the downstream turbines. The red circles show the turbines that have another phantom-turbine close in front of them. The power extraction should be lower.

With this modification, all the red plots go down (red

arrows) and as expected before the simulations the power extraction increases with the downstream distance.

However there is still the unexplained extracted power of the 4 radii downstream turbine and a tip to tip distance of 1.25 radius which stays higher than the 5 radii. Moreover three fifth of the turbines used for the study of the influence of the offset (see 6.2 and 6.3) have a turbine before them at 0.4 radius. That is why for example the turbines with an offset of 2 radii extract more power than the turbines with a bigger offset. But this difference is really small and is reducing with the downstream

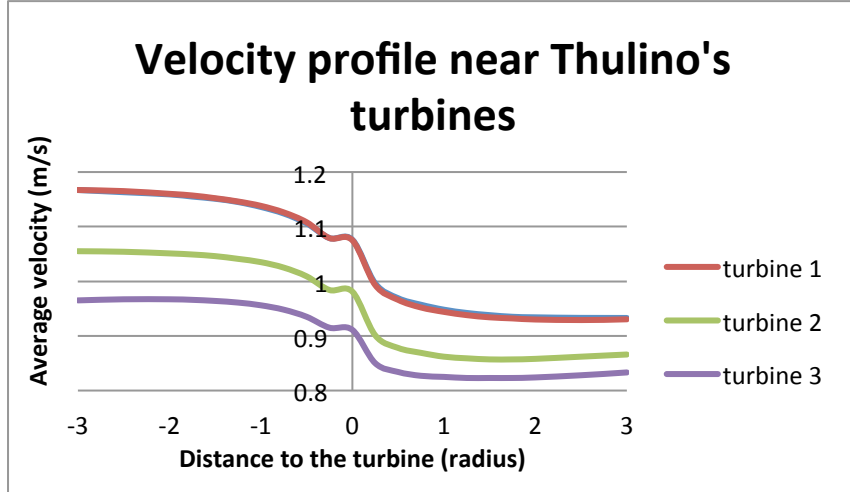


Figure 33: Velocity profile near Oskar turbines

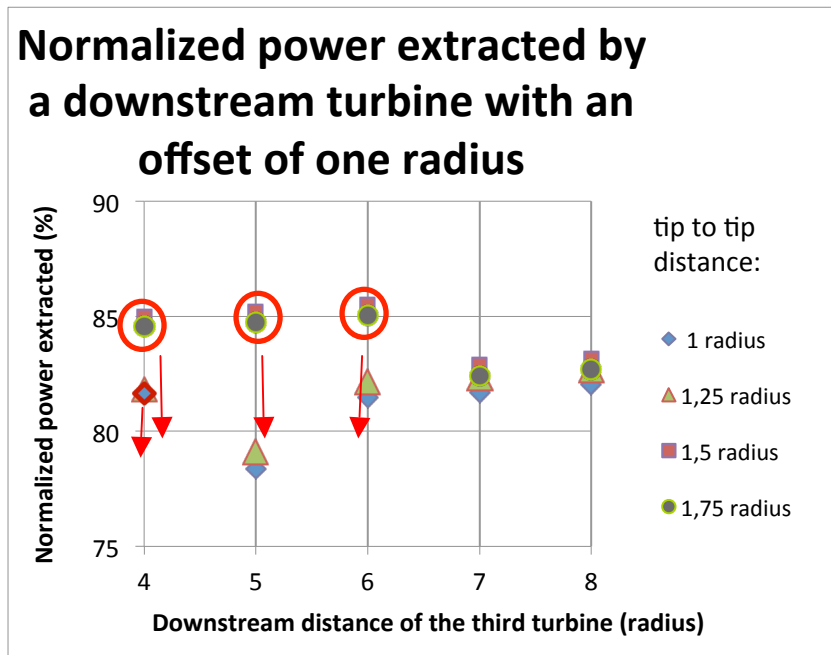


Figure 34: Chart of the power extracted showing the problem of the mesh distance.

4.4.3 Moments

The z axis moment supported by a turbine is due to the difference of velocity between the left and the right hand sides of the turbine. This difference is caused by the wake of an upstream turbine. We can link the moment and the power extraction. If the moment is equal to zero that means the velocity on the sides of the turbine is the same, so the turbine is totally in the wake

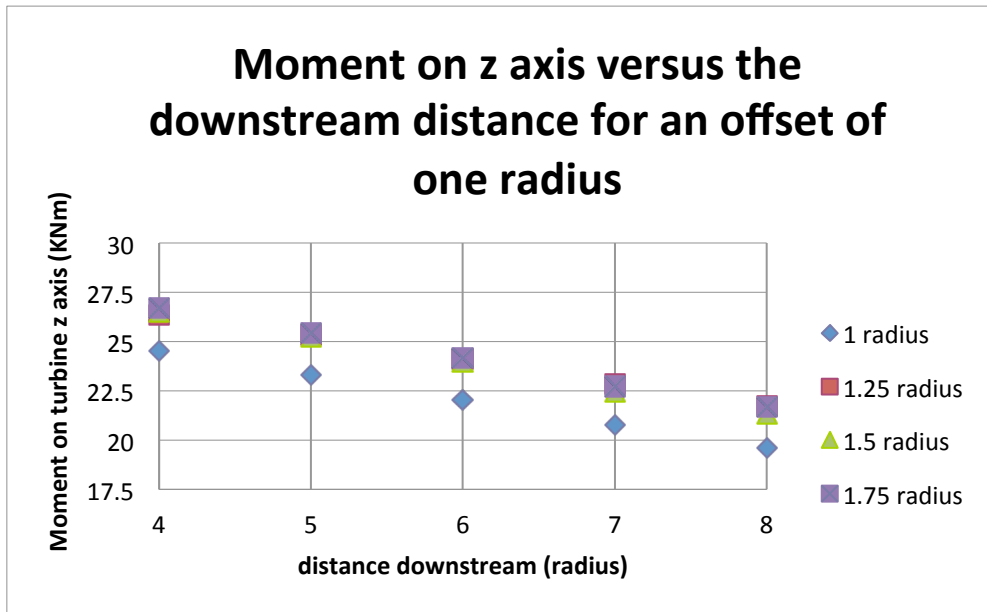


Figure 35a: Chart of the moment versus the downstream distance

one in the wake the other in the free stream flow. The figures 35 (a and b) are showing the moment on the z axis of the downstream turbines. Figure 35a shows that the moment decreases with the downstream distances. This is a normal effect because when the distance increases the mix between the free stream flow and the wake is more developed and also the power extraction is more efficient, as see in paragraph 45762762. On the figure 35b, the shape of the curve confirms the

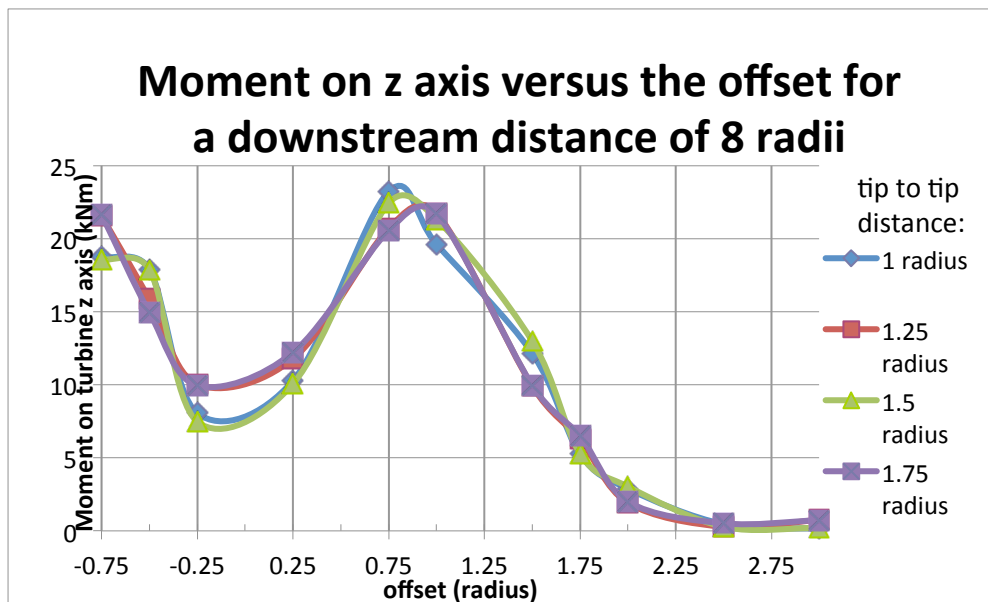


Figure 35b Chart of the moment versus the offset

or totally out of the wake. The power extraction is respectively minimum or maximum. On the opposite when the moment is maximum the turbine is divided in two equal surfaces:

one in the wake the other in the free stream flow. The figures 35 (a and b) are showing the moment on the z axis of the downstream turbines. Figure 35a shows that the moment decreases with the downstream distances. This is a normal effect because when the distance increases the mix between the free stream flow and the wake is more developed and also the power extraction is more efficient, as see in paragraph 45762762. On the figure 35b, the shape of the curve confirms the previous results of the chapter 6.2, the moment is equal to zero when the offset is bigger than 2 radii and so the power extraction is the same as the upstream turbines.

5 CONCLUSION



The Admiralty Inlet project is a feasibility study for the implementation of a field of Marine Hydrokinetic Turbines. This study is based on computational modeling due to the high cost of experimental work. Previously, the Virtual Blade Model and Actuator Disk Model were chosen to simulate respectively the rotor and the hub of a turbine. This choice turned out to be the best compromise between accuracy and computational time.

As a continuation of previous studies a computational domain was created in order to investigate on three main building parameters: the tip to tip distance between the two upstream turbines, the downstream distance and the lateral offset of the downstream turbine. By using a shearvelocity profile at the inlet of the domain undesired effects such as a boundary layer on the seabed occur. As the velocity profile and magnitude are changing with the downstream distance, a normalization of the power extraction of each turbine is necessary.

The available kinetic energy in the wake of a turbine is far less than in the free-stream current. Thus the lateral offset of downstream turbine is one of the most important building parameter for the efficiency of turbine arrays. However, as a field contains several turbines, it is not possible to place every turbine outside of all wake-effect, but to reduce its impact.

However some phenomena remain unexplained in this study. In order to determine if these effects are physical phenomena or just the consequences of the computational model, further simulations must be done by using e.g. the Single Reference Frame Model instead of Virtual Blade Model. Sedimentation processes and the environmental impact of turbine arrays on maritime species are still important topics for future studies. But also studies about the improvement of the efficiency of turbine-blades and their resistance to the sudden moment due to velocity profile should be conducted. Finally, the implementation of fields of Marine Hydrokinetic turbines is not accomplished yet; the exploitation of this renewable energy remains a challenging purpose.

APPENDIXES



Appendix A : Blade chord and Twist distribution

Section number	Span Station (r/R)	Chord Length (m)	Twist angle (°)
1	0,25	0,7728	-18,074
2	0,273	0,711	-14,292
3	0,298	0,697	-11,909
4	0,353	0,666	-7,979
5	0,408	0,636	-5,308
6	0,424	0,627	-4,715
7	0,463	0,605	-3,425
8	0,518	0,574	-2,083
9	0,573	0,543	-1,15
10	0,576	0,542	-1,115
11	0,628	0,512	-0,494
12	0,683	0,482	0,015
13	0,727	0,457	0,381
14	0,739	0,451	0,475
15	0,794	0,42	0,92
16	0,849	0,389	1,352
17	0,864	0,381	1,469
18	0,904	0,358	1,775
19	0,959	0,328	2,191
20	1	0,305	2,5

Ap



pendix B : Lift and Drag coefficients versus angle of attack

	C_l	C_d
Reynolds number	750000	750000
Mach number	0.07	0.07
Number of data points	40	54

AOA (°)	C _l	AOA (°)	C _l
-180.00	-0.5600	6.00	0.752067
-172.00	-0.5600	8.00	0.931158
-161.00	-0.5600	10.00	1.085360
-147.00	-0.5600	12.00	1.204034
-129.00	-0.5600	14.00	1.257791
-49.00	-0.5600	16.00	1.237904
-39.00	-0.5600	17.10	0.940
-20.10	-0.5600	18.10	0.850
-18.10	-0.6700	19.10	0.7000
-16.10	-0.7900	20.10	0.6600
-14.20	-0.8400	22.00	0.7000
-12.20	-0.7000	24.10	0.7900
-10.10	-0.6300	26.20	0.8800
-8.20	-0.5600	39.00	0.8800
-6.10	-0.6400	49.00	0.8800
-4.10	-0.4200	129.00	0.8800
-2.10	-0.2100	147.00	0.8800
0.00	0.134449	161.00	0.8800
2.00	0.348827	172.50	0.8800
4.00	0.556387	180.00	0.8800

AOA (°)	C _d	AOA (°)	C _d
-180.00	0.3027	8.00	0.041006
-175.00	0.3027	10.00	0.051192
-170.00	0.3027	12.00	0.064605
-165.00	0.3027	14.00	0.083792
-160.00	0.3027	16.00	0.110564
-140.00	0.3027	17.10	0.0994
-120.00	0.3027	18.10	0.2306
-110.00	0.3027	19.10	0.3142
-100.00	0.3027	20.10	0.3186
-90.00	0.3027	22.00	0.3694
-80.00	0.3027	24.10	0.4457
-70.00	0.3027	26.20	0.5260
-60.00	0.3027	30.00	0.5260
-50.00	0.3027	50.00	0.5260
-30.00	0.3027	60.00	0.5260
-20.10	0.3027	70.00	0.5260
-18.10	0.3069	80.00	0.5260
-16.10	0.1928	90.00	0.5260
-14.20	0.0898	100.00	0.5260
-12.20	0.0553	110.00	0.5260
-10.10	0.0390	120.00	0.5260
-8.20	0.0233	140.00	0.5260
-6.10	0.0112	160.00	0.5260
0.000	0.022725	165.00	0.5260
2.00	0.024253	170.00	0.5260
4.00	0.027739	175.00	0.5260
6.00	0.033265	180.00	0.5260



Appendix C : UDF of the used shear velocity profile

```
#include "udf.h"
# define Ui -2.499995429          /* Free stream velocity (m/s)*/
# define Zturbine 16.590716809   /* Depth (m)*/
# define Zdepth 56

DEFINE_PROFILE(inlet_velocity_new, thread, position)
{
    real y[ND_ND];               /* this will hold the position vector */
    real z;                      /* this identifies z as a variable */
    face_t f;                    /* defines a face */
    begin_f_loop(f, thread)      /* loop that steps through all faces in a thread */
    {
        F_CENTROID(y,f,thread);  /* returns coordinates of face centroid in array y[] */
        z = y[2];
        F_PROFILE(f, thread, position)=Ui*pow(((z+Zturbine)/Zdepth),(0.09901));
    }
    end_f_loop(f, thread)
}
```

3) : Example of GUI journal in FLUENT

```
sh

e and data file

- item "MenuBar*ReadSubMenu*Case & Data..."
- entry "Select File*FilterText" "*.cas*"
- item "Select File*Apply"
- entry "Select File*Text" "FM_mesh_mario_mother_file.cas"
- item "Select File*OK"

ned --> Functions --> Compiled UDFs

- item "MenuBar*FunctionsSubMenu*Compiled..."
- item "Compiled UDFs*Frame1*Table1*Frame1*PushButton2( Add)"

es

cx-add-list-items "Select File*Frame14*Table1*Frame2*Frame2*Frame2*List2"
model_v10.1.c") #f)
cx-add-list-items "Select File*Frame14*Table1*Frame2*Frame2*Frame2*List2"
velocity_profile.c") #f)
cx-add-list-items "Select File*Frame14*Table1*Frame2*Frame2*Frame2*List2"
_mem_v1.0.c") #f)

cx-set-text-entry "Select File*FilterText" ".*.c")
cx-activate-item "Select File*Apply"
cx-activate-item "Select File*OK"
cx-activate-item "Compiled UDFs*Frame1*Table1*Frame2*PushButton2( Add)"

les

cx-add-list-items "Select File*Frame14*Table1*Frame2*Frame2*Frame2*List2"
_mem.h") #f)

cx-set-text-entry "Select File*FilterText" ".*.h")
cx-activate-item "Select File*Apply"
cx-activate-item "Select File*OK"
cx-activate-item "Compiled UDFs*PushButton3(Build)"
cx-activate-item "Warning*OK"
cx-activate-item "Compiled UDFs*PanelButtons*PushButton1(OK)"
```

```
; File --> Read scheme (VBM)
```

```
(cx-gui-do cx-activate-item "MenuBar*ReadSubMenu*Scheme...")
(cx-gui-do cx-set-text-entry "Select File*FilterText" "*.scm")
(cx-gui-do cx-activate-item "Select File*Apply")
(cx-gui-do cx-set-text-entry "Select File*Text" "rotor_model_v10.scm")
(cx-gui-do cx-activate-item "Select File*OK")
(load "/files1/Mario/fluvent/FM_mesh/rotor_model_v10.scm")
```

```
; Define --> Models --> Select VBM
```

```
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton3(Models)")
(cx-gui-do cx-set-list-selections "Models*Frame1*Table1*Frame1*List1(Models)")
(cx-gui-do cx-activate-item "Models*Frame1*Table1*Frame1*List1(Models)")
(cx-gui-do cx-activate-item "Models*Frame1*Table1*PushButton2(Edit)")
```

```
; Rotor Inputs Turbine-1
```

```
(cx-gui-do cx-set-integer-entry "Rotor Inputs*Frame2*Table2*IntegerEntry1(Active f
(cx-gui-do cx-activate-item "Rotor Inputs*Frame2*Table2*IntegerEntry1(Active f
```

```
; Select Rotor Face Zone (Surfaces)
```

```
(cx-gui-do cx-set-list-selections "Rotor Inputs*Frame4*Frame1(General)*Frame1
Zone)*Table6(Rotor Face Zone)*Frame1*List1(Surfaces)" '( 70))
```

```
; Set Rotor Disk Origin
```

```
(cx-gui-do cx-activate-item "Rotor Inputs*Frame4*Frame1(General)*Frame1*Frz
Zone)*Table6(Rotor Face Zone)*Frame1*List1(Surfaces)")
```

```
(cx-gui-do cx-set-real-entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1
Origin)*Table5(Rotor Disk Origin)*RealEntry1(X)" '(-8.295))
```

```
(cx-gui-do cx-set-real-entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1
Origin)*Table5(Rotor Disk Origin)*RealEntry2(Y)" '( 0))
```

```
entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1*Frame5(Rotor Disk
Disk Origin)*RealEntry3(Z)" '( 0))
```

```
-item "Rotor Inputs*Frame2*Table2*PushButton2(Change/Create)")
```

uts Turbine-2

```
ger-entry "Rotor Inputs*Frame2*Table2*IntegerEntry1(Active Rotor Zone)" 2)
-item "Rotor Inputs*Frame2*Table2*IntegerEntry1(Active Rotor Zone)")
```

; Select Rotor Face Zone (Surfaces)

```
elections "Rotor Inputs*Frame4*Frame1(General)*Frame1*Frame6(Rotor Face
ace Zone)*Frame1*List1(Surfaces)" '( 74))
```

; Set Rotor Disk Origin

```
-item "Rotor Inputs*Frame4*Frame1(General)*Frame1*Frame6(Rotor Face
ace Zone)*Frame1*List1(Surfaces)")
```

```
entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1*Frame5(Rotor Disk
Disk Origin)*RealEntry1(X)" '( 8.2950))
```

```
entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1*Frame5(Rotor Disk
Disk Origin)*RealEntry2(Y)" '( 0))
```

```
entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1*Frame5(Rotor Disk
Disk Origin)*RealEntry3(Z)" '( 0))
```

```
-item "Rotor Inputs*Frame2*Table2*PushButton2(Change/Create)")
```

uts Turbine-3

```
ger-entry "Rotor Inputs*Frame2*Table2*IntegerEntry1(Active Rotor Zone)" 3)
-item "Rotor Inputs*Frame2*Table2*IntegerEntry1(Active Rotor Zone)")
```

; Select Rotor Face Zone (Surfaces)

```
(cx-gui-do cx-set-list-selections "Rotor Inputs*Frame4*Frame1(General)*Frame1
Zone)*Table6(Rotor Face Zone)*Frame1*List1(Surfaces)" '( 39))
```

; Set Rotor Disk Origin

```
(cx-gui-do cx-activate-item "Rotor Inputs*Frame4*Frame1(General)*Frame1*Fra
Zone)*Table6(Rotor Face Zone)*Frame1*List1(Surfaces)")
```

```
(cx-gui-do cx-set-real-entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1
Origin)*Table5(Rotor Disk Origin)*RealEntry1(X)" '( 5.53))
```

```
(cx-gui-do cx-set-real-entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1
Origin)*Table5(Rotor Disk Origin)*RealEntry2(Y)" '( -47.558))
```

```
(cx-gui-do cx-set-real-entry-list "Rotor Inputs*Frame4*Frame1(General)*Frame1
Origin)*Table5(Rotor Disk Origin)*RealEntry3(Z)" '( 0))
```

```
(cx-gui-do cx-activate-item "Rotor Inputs*Frame2*Table2*PushButton2(Change/
(cx-gui-do cx-activate-item "Rotor Inputs*PanelButtons*PushButton1(OK)"))
```

; Define --> Cell Zone Conditions

; Activation first hub

```
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton6(Cell Zone Cor
(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*I
(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*Pushl
(cx-gui-do cx-set-toggle-button "fluid-10427-1*Frame3*Table3*CheckButton2(P
(cx-gui-do cx-activate-item "fluid-10427-1*Frame3*Table3*CheckButton2(Prou
```

; Set Parameters for Porous Zone

; Direction-2 Vector (Y)

```
(cx-gui-do cx-set-real-entry-list "fluid-10427-1*Frame3*Table3*Frame8*Frame3l
Zone)*Frame1*Table1*Frame3*Table3*Frame5(Direction-2 Vector)*Table5(Dire
Vector)*Frame2*Table2*RealEntry2(Y)" '( -1))
```

; Viscous resistance

```
entry-list "fluid-10427-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Viscous
Table1*RealEntry2(Direction-1))" '( 2.5504e-07))
entry-list "fluid-10427-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Viscous
Table2*RealEntry2(Direction-2))" '( 2.5504e-07))
entry-list "fluid-10427-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Viscous
Table3*RealEntry2(Direction-3))" '( 2.5504e-07))
```

; Inertial resistance

```
entry-list "fluid-10427-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Inertial
Table2*RealEntry2(Direction-1))" '( 5.653))
entry-list "fluid-10427-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Inertial
Table3*RealEntry2(Direction-2))" '( 5.653))
entry-list "fluid-10427-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Inertial
Table4*RealEntry2(Direction-3))" '( 5.653))
-item "fluid-10427-1*PanelButtons*PushButton1(OK)"
```

1 second hub

```
elections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 31))
-item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)"
-item "Boundary
able1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(Edit)"
le-button "fluid-10419-1*Frame3*Table3*CheckBox2(Porous Zone)" #f)
-item "fluid-10419-1*Frame3*Table3*CheckBox2(Porous Zone)"
```

; Set Parameters for Porous Zone

; Direction-2 Vector (Y)

```
entry-list "fluid-10419-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame5(Direction-2 Vector)*Table5(Direction-2
e2*RealEntry2(Y))" '( -1))
```

; Viscous resistance

```
entry-list "fluid-10419-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Viscous
Table1*RealEntry2(Direction-1))" '( 2.5504e-07))
```

```
(cx-gui-do cx-set-real-entry-list "fluid-10419-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Visc
Resistance)*Frame2*Table2*RealEntry2(Direction-2))" '( 2.5504e-07))
(cx-gui-do cx-set-real-entry-list "fluid-10419-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Visc
Resistance)*Frame3*Table3*RealEntry2(Direction-3))" '( 2.5504e-07))
```

; Inertial resistance

```
(cx-gui-do cx-set-real-entry-list "fluid-10419-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Iner
Resistance)*Frame2*Table2*RealEntry2(Direction-1))" '( 5.653))
(cx-gui-do cx-set-real-entry-list "fluid-10419-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Iner
Resistance)*Frame3*Table3*RealEntry2(Direction-2))" '( 5.653))
(cx-gui-do cx-set-real-entry-list "fluid-10419-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Iner
Resistance)*Frame4*Table4*RealEntry2(Direction-3))" '( 5.653))
(cx-gui-do cx-activate-item "fluid-10419-1*PanelButtons*PushButton1(OK)")
```

; Activation third hub

```
(cx-gui-do cx-set-list-selections "Boundary Conditions*Frame1*Table1*Frame1*
(cx-gui-do cx-activate-item "Boundary Conditions*Frame1*Table1*Frame1*List1
(cx-gui-do cx-activate-item "Boundary
Conditions*Frame1*Table1*Frame2*Table2*Frame4*Table4*ButtonBox1*Pushl
(cx-gui-do cx-set-toggle-button "fluid-10387-1*Frame3*Table3*CheckBox2(Po
(cx-gui-do cx-activate-item "fluid-10387-1*Frame3*Table3*CheckBox2(Porou
```

; Set Parameters for Porous Zone

; Direction-2 Vector (Y)

```
(cx-gui-do cx-set-real-entry-list "fluid-10387-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame5(Direction-2 Vector)*Table5(Dire
Vector)*Frame2*Table2*RealEntry2(Y))" '( -1))
```

; Viscous resistance

```
(cx-gui-do cx-set-real-entry-list "fluid-10387-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Visc
Resistance)*Frame1*Table1*RealEntry2(Direction-1))" '( 2.5504e-07))
(cx-gui-do cx-set-real-entry-list "fluid-10387-1*Frame3*Table3*Frame8*Frame3(
Zone)*Frame1*Table1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Visc
Resistance)*Frame2*Table2*RealEntry2(Direction-2))" '( 2.5504e-07))
```



```
entry-list "fluid-10387-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame8(Viscous Resistance)*Table8(Viscous
Table3*RealEntry2(Direction-3))" '( 2.5504e-07))
```

; Inertial resistance

```
entry-list "fluid-10387-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Inertial
Table2*RealEntry2(Direction-1))" '( 5.653))
entry-list "fluid-10387-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Inertial
Table3*RealEntry2(Direction-2))" '( 5.653))
entry-list "fluid-10387-1*Frame3*Table3*Frame8*Frame3(Porous
1*Frame3*Table3*Frame9(Inertial Resistance)*Table9(Inertial
Table4*RealEntry2(Direction-3))" '( 5.653))
item "fluid-10387-1*PanelButtons*PushButton1(OK)"
```

```
elections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 73))
item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)"
item "Boundary
able1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(Edit)"
le-button "fluid-10428-1*Frame3*Table3*CheckButton4(Source Terms)" #f)
item "fluid-10428-1*Frame3*Table3*CheckButton4(Source Terms)"
item "fluid-10428-1*PanelButtons*PushButton1(OK)"
```

itor

```
elections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 69))
item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)"
item "Boundary
able1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(Edit)"
le-button "fluid-10420-1*Frame3*Table3*CheckButton4(Source Terms)" #f)
item "fluid-10420-1*Frame3*Table3*CheckButton4(Source Terms)"
item "fluid-10420-1*PanelButtons*PushButton1(OK)"
```

ir

```
elections "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)" '( 42))
item "Boundary Conditions*Frame1*Table1*Frame1*List1(Zone)"
item "Boundary
able1*Frame2*Table2*Frame4*Table4*ButtonBox1*PushButton1(Edit)"
le-button "fluid-10388-1*Frame3*Table3*CheckButton4(Source Terms)" #f)
item "fluid-10388-1*Frame3*Table3*CheckButton4(Source Terms)"
```

```
(cx-gui-do cx-activate-item "fluid-10388-1*PanelButtons*PushButton1(OK)")
```

; Initialization from Inlet

```
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton15(Solution Ini
(cx-gui-do cx-set-list-selections "Solution Initialization*Frame1*Table1*DropDow
from)" '(6))
(cx-gui-do cx-activate-item "Solution Initialization*Frame1*Table1*DropDownLis
(cx-gui-do cx-activate-item "Solution
Initialization*Frame1*Table1*ButtonBox6*PushButton1(Initialize)")
(cx-gui-do cx-activate-item "Warning*OK")
```

; Running the calculation

```
(cx-gui-do cx-activate-item "NavigationPane*Frame1*PushButton17(Run Calcula
(cx-gui-do cx-set-integer-entry "Run Calculation*Frame1*Table1*IntegerEntry8(I
Iterations)" 1200)
(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*IntegerEntry8(Num
(cx-gui-do cx-activate-item "Run Calculation*Frame1*Table1*PushButton18(Calc
(cx-gui-do cx-activate-item "Information*OK")
```

; Save case and data files after calculation

```
(cx-gui-do cx-activate-item "MenuBar*WriteSubMenu*Case & Data...")
(cx-gui-do cx-set-text-entry "Select File*FilterText" " *.*")
(cx-gui-do cx-activate-item "Select File*Apply")
(cx-gui-do cx-set-text-entry "Select File*Text"
"FM_mesh_mario_1_1m050R_shearvelocity_BC_results.cas")
(cx-gui-do cx-activate-item "Select File*OK")
```

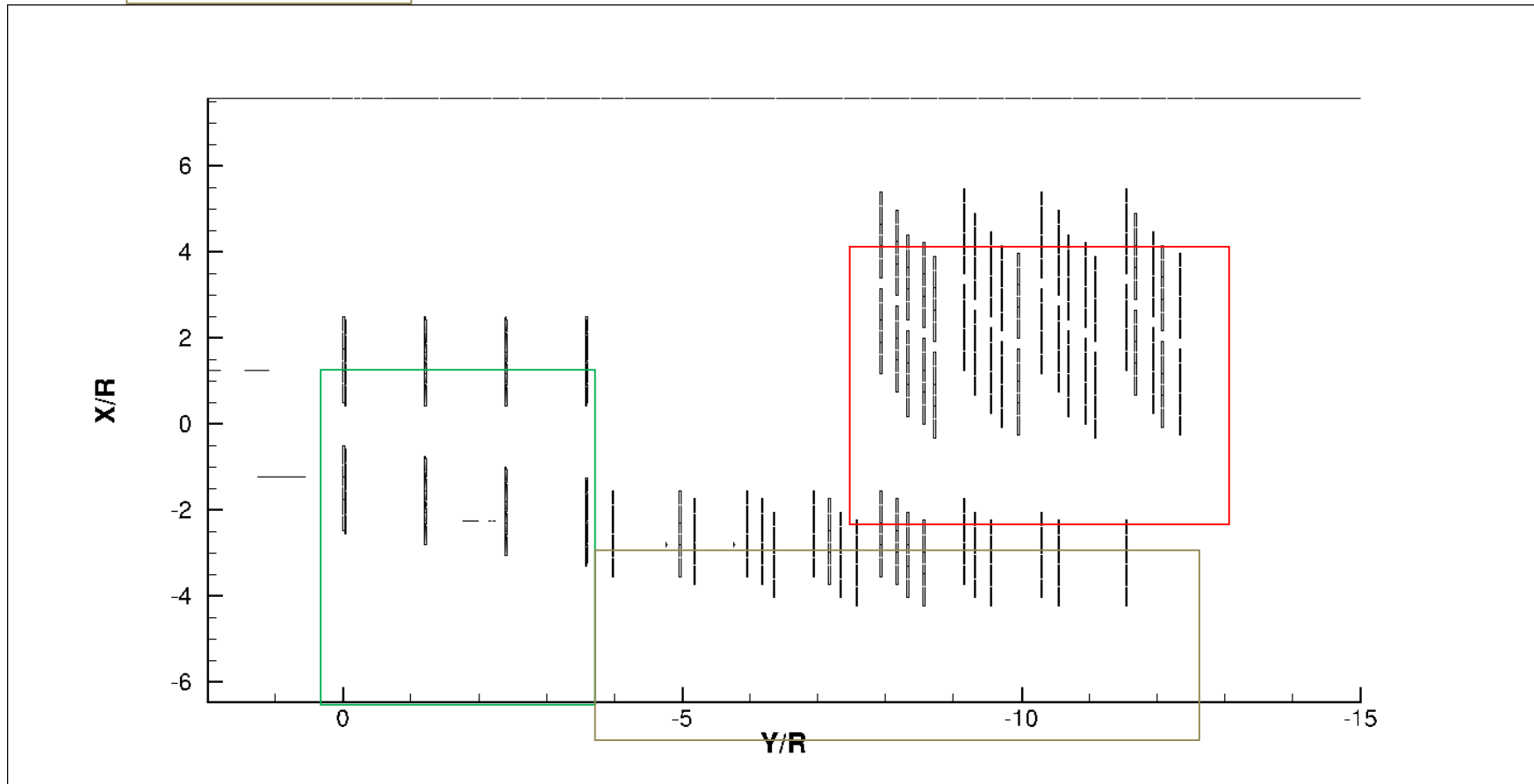
; Exit Fluent

```
(cx-gui-do cx-activate-item "MenuBar*FileMenu*Exit")
```

NOTE: This journal identifies all the turbines, hubs and blades as a number given by FLUENT and not with their own numbers given by the user. As this code is GUI-based even the numbers that correspond to these zones are characterized by numbers that have not been used out previously. All parameters that can be changed by the user are highlighted in red.



Appendix E: scheme of the array of the turbines





Appendix F Tecplot [®] script

To fit in the report, the script works for only 8 turbines. The structure is the same than the script used for the study, just the number of variable changes.

```

#!MC 1200
# Created by Tecplot 360 build 12.0.0.3454
$!VarSet |MFBD| = '.'

$!PROMPTFORTEXTSTRING |Filename|
  INSTRUCTIONS = "Choose file name of fluent cas and data you
want to load (excl. file ext):"

$!DRAWGRAPHICS FALSE
$!READDATASET ""STANDARDSYNTAX" "1.0" "LoadOption"
"CaseAndData" "FILENAME_CaseFile"
"|MFBD|/|Filename|.cas" "FILENAME_DataFile"
"|MFBD|/|Filename|.dat" "GridZones" "CellsAndBoundaries"
"IncludeParticleData" "No" "AllPolyZones" "No"
"AverageToNodes" "No" "SaveUncompressedFiles" "No"
DATASETREADER = 'Fluent Data Loader'

$!PROMPTFORYESNO |position_answer| INSTRUCTIONS = "Do
you want to use preset turbine positions?"
$!IF "|position_answer|" == "YES"

  $!WHILE 0 < 1
    $!PROMPTFORTEXTSTRING
|num_turbines|
    INSTRUCTIONS = "Enter the
number of turbines?"

    $!Varset |t1_enable| = 0
    $!Varset |t2_enable| = 0
    $!Varset |t3_enable| = 0
    $!Varset |t4_enable| = 0
  
```

```

$!Varset
|t5_enable| = 0
  $!Varset |t6_enable| = 0
  $!Varset |t7_enable| = 0
  $!Varset |t8_enable| = 0
  $!Varset |t9_enable| = 0
  $!Varset |t10_enable| = 0
  $!Varset |t11_enable| = 0
  $!Varset |t12_enable| = 0
  $!Varset |matching| = 0

  # Can be extended up to 12 turbines
without changing the loop but have to include positions then
  $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 1 included?"
  $!IF "|t_answer|" == "YES"
    $!Varset |t1_enable| = 1
    $!Varset |matching| +=
|t1_enable|
    $!IF |matching| ==
|num_turbines|
      $!BREAK
    $!ENDIF
  $!ENDIF

  $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 2 included?"
  $!IF "|t_answer|" == "YES"
    $!Varset |t2_enable| = 1
    $!Varset |matching| +=
|t2_enable|
    $!IF |matching| ==
|num_turbines|
      $!BREAK
    $!ENDIF
  $!ENDIF

  $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 3 included?"
  $!IF "|t_answer|" == "YES"
    $!Varset |t3_enable| = 1
    $!Varset |matching| +=
|t3_enable|
    $!IF |matching| ==
|num_turbines|
      $!BREAK
    $!ENDIF
  $!ENDIF

  $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 4 included?"
  $!IF "|t_answer|" == "YES"
    $!Varset |t4_enable| = 1
    $!Varset |matching| +=
|t4_enable|
    $!IF |matching| ==
|num_turbines|
      $!BREAK
    $!ENDIF
  $!ENDIF

  $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 5 included?"
  $!IF "|t_answer|" == "YES"
    $!Varset |t5_enable| = 1
    $!Varset |matching| +=
|t5_enable|
    $!IF |matching| ==
|num_turbines|
      $!BREAK
    $!ENDIF
  $!ENDIF

  $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 6 included?"
  $!IF "|t_answer|" == "YES"
    $!Varset |t6_enable| = 1
    $!Varset |matching| +=
|t6_enable|
    $!IF |matching| ==
|num_turbines|
      $!BREAK
    $!ENDIF
  $!ENDIF

  $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 7 included?"
  $!IF "|t_answer|" == "YES"
    $!Varset |t7_enable| = 1
    $!Varset |matching| +=
|t7_enable|
    $!IF |matching| ==
|num_turbines|
      $!BREAK
    $!ENDIF
  $!ENDIF
  
```



```

    $!BREAK
    $!ENDIF
$!ENDIF

    $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 8 included?"
    $!IF "|t_answer|" == "YES"
        $!Varset |t8_enable| = 1
        $!Varset |matching| +=
|t8_enable|
    $!IF |matching| ==
|num_turbines|
        $!BREAK
    $!ENDIF
$!ENDIF

    $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 9 included?"
    $!IF "|t_answer|" == "YES"
        $!Varset |t9_enable| = 1
        $!Varset |matching| +=
|t9_enable|
    $!IF |matching| ==
|num_turbines|
        $!BREAK
    $!ENDIF
$!ENDIF

    $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 10 included?"
    $!IF "|t_answer|" == "YES"
        $!Varset |t10_enable| = 1
        $!Varset |matching| +=
|t10_enable|
    $!IF |matching| ==
|num_turbines|
        $!BREAK
    $!ENDIF
$!ENDIF

    $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 11 included?"
    $!IF "|t_answer|" == "YES"

```

```

    $!Varset |t11_enable| = 1
    $!Varset |matching| +=
|t11_enable|
    $!IF |matching| ==
|num_turbines|
        $!BREAK
    $!ENDIF
$!ENDIF

    $!PROMPTFORYESNO |t_answer|
INSTRUCTIONS = "Is Turbine 12 included?"
    $!IF "|t_answer|" == "YES"
        $!Varset |t12_enable| = 1
        $!Varset |matching| +=
|t12_enable|
    $!IF |matching| ==
|num_turbines|
        $!BREAK
    $!ENDIF
$!ENDIF

    $!PAUSE "Number of turbines and
enabled turbines doesn't match. Redo, do right"
$!ENDWHILE

# Turbine parameters
$!Varset |t1_x| = -11.06
$!Varset |t2_x| = 11.06
$!Varset |t3_x| = 0
$!Varset |t4_x| = 15
$!Varset |t5_x| = -7.06
$!Varset |t6_x| = 16.59
$!Varset |t7_x| = -11.06
$!Varset |t8_x| = 16.59
$!Varset |t9_x| = 0
$!Varset |t10_x| = 0
$!Varset |t11_x| = 0
$!Varset |t12_x| = 0

$!Varset |t1_y| = 0
$!Varset |t2_y| = 0
$!Varset |t3_y| = -22.12
$!Varset |t4_y| = -22.12
$!Varset |t5_y| = -66.36
$!Varset |t6_y| = -66.36
$!Varset |t7_y| = -110.6

```

```

$!Varset |t8_y| = -110.6
$!Varset |t9_y| = 0
$!Varset |t10_y| = 0
$!Varset |t11_y| = 0
$!Varset |t12_y| = 0

$!Varset |t1_z| = 0
$!Varset |t2_z| = 0
$!Varset |t3_z| = 0
$!Varset |t4_z| = 0
$!Varset |t5_z| = 0
$!Varset |t6_z| = 0
$!Varset |t7_z| = 0
$!Varset |t8_z| = 0
$!Varset |t9_z| = 0
$!Varset |t10_z| = 0
$!Varset |t11_z| = 0
$!Varset |t12_z| = 0

$!ELSE
    $!Varset |num_turbines| = 1
    $!Varset |t1_enable| = 1
    $!Varset |t2_enable| = 0
    $!Varset |t3_enable| = 0
    $!Varset |t4_enable| = 0
    $!Varset |t5_enable| = 0
    $!Varset |t6_enable| = 0
    $!Varset |t7_enable| = 0
    $!Varset |t8_enable| = 0
    $!Varset |t9_enable| = 0
    $!Varset |t10_enable| = 0
    $!Varset |t11_enable| = 0
    $!Varset |t12_enable| = 0

    $!PROMPTFORTEXTSTRING |t1_x| INSTRUCTIONS =
    "What is the x-position of the turbine?"
    $!PROMPTFORTEXTSTRING |t1_y| INSTRUCTIONS =
    "What is the y-position of the turbine?"
    $!PROMPTFORTEXTSTRING |t1_z| INSTRUCTIONS =
    "What is the z-position of the turbine?"

$!ENDIF

```



Controls

```

$!Varset |type_data_export| = 0
$!Varset |type_upstream| = 0
$!Varset |type_wake| = 0
$!Varset |type_manual| = 0
$!Varset |decision| = 0
$!Varset |turbine_R| = 5.53

$!WHILE 0 < 1
    $!PROMPTFORYESNO |type_dist_answer|
    INSTRUCTIONS = "Do you want to use data export type?"
    $!IF "|type_dist_answer|" == "YES"
        $!Varset |type_data_export| = 1
        $!Varset |decision| += 1
    $!ELSE
        $!PROMPTFORYESNO |t_answer|
        INSTRUCTIONS = "Do you want to use upstream type?"
        $!IF "|t_answer|" == "YES"
            $!Varset |type_upstream| = 1
            $!Varset |decision| += 1
        $!ELSE
            $!PROMPTFORYESNO
            |t_answer| INSTRUCTIONS = "Do you want to use wake type?"
            $!IF "|t_answer|" == "YES"
                $!Varset
                |type_wake| = 1
                $!Varset |decision|
                += 1
            $!ELSE
                $!PROMPTFORYESNO |t_answer| INSTRUCTIONS =
                "Do you want to set one distance manually?"
                $!IF "|t_answer|" ==
                "YES"
                    $!Varset
                    |type_manual| = 1
                    $!Varset
                    |decision| += 1
                $!ENDIF
            $!ENDIF
        $!ENDIF
    $!ENDIF
$!IF |decision| > 0
    $!BREAK

```

```

$!ENDIF
$!ENDWHILE

$!Varset |start_zone| = |NUMZONES|

# Input data

# dist can be extended up to 20 number without changing the
loop
$!IF |type_upstream| == 1
    $!Varset |dist_1| = 4.0 # in radius
    $!Varset |dist_2| = 3.0
    $!Varset |dist_3| = 2.75
    $!Varset |dist_4| = 2.5
    $!Varset |dist_5| = 2.25
    $!Varset |dist_6| = 2.0
    $!Varset |dist_7| = 1.75
    $!Varset |dist_8| = 1.5
    $!Varset |dist_9| = 1.25
    $!Varset |dist_10| = 1.0
    $!Varset |dist_11| = 0.75
    $!Varset |dist_12| = 0.5
    $!Varset |dist_13| = 0.25
    $!Varset |dist_14| = 0.1
    $!Varset |num_distance| = 14
$!ELSEIF |type_wake| == 1
    $!Varset |dist_1| = 0.25 # in radius
    $!Varset |dist_2| = 0.1
    $!Varset |dist_3| = 0.0
    $!Varset |dist_4| = -0.1
    $!Varset |dist_5| = -0.25
    $!Varset |dist_6| = -0.5
    $!Varset |dist_7| = -0.75
    $!Varset |dist_8| = -1.0
    $!Varset |dist_9| = -1.25
    $!Varset |dist_10| = -1.5
    $!Varset |dist_11| = -2.00
    $!Varset |dist_12| = -2.5
    $!Varset |num_distance| = 12
$!ELSEIF |type_data_export| == 1
    $!Varset |dist_1| = 2.0 # in radius
    $!Varset |dist_2| = 1.0
    $!Varset |dist_3| = 0.25

```

```

$!Varset |num_distance| = 3
$!ELSEIF |type_manual| == 1
    $!PROMPTFORTEXTSTRING |dist_1| INSTRUCTIONS =
    "What is the distance upstream in radius?"
    $!Varset |num_distance| = 1
$!ENDIF

$!Varset |t1_name| = "T1"
$!Varset |t2_name| = "T2"
$!Varset |t3_name| = "T3"
$!Varset |t4_name| = "T4"
$!Varset |t5_name| = "T5"
$!Varset |t6_name| = "T6"
$!Varset |t7_name| = "T7"
$!Varset |t8_name| = "T8"
$!Varset |t9_name| = "T9"
$!Varset |t10_name| = "T10"
$!Varset |t11_name| = "T11"
$!Varset |t12_name| = "T12"

$!Varset |t1_undone| = |t1_enable|
$!Varset |t2_undone| = |t2_enable|
$!Varset |t3_undone| = |t3_enable|
$!Varset |t4_undone| = |t4_enable|
$!Varset |t5_undone| = |t5_enable|
$!Varset |t6_undone| = |t6_enable|
$!Varset |t7_undone| = |t7_enable|
$!Varset |t8_undone| = |t8_enable|
$!Varset |t9_undone| = |t9_enable|
$!Varset |t10_undone| = |t10_enable|
$!Varset |t11_undone| = |t11_enable|
$!Varset |t12_undone| = |t12_enable|

# Loop parameters
$!VARSET |n| = 0 # initialize counter

$!LOOP |num_turbines|

```



```

$!VARSET |n| += 1 # add one counter

# Pick data from right turbine
$!IF |t1_undone| == 1
    $!Varset |turbine_x| = |t1_x|
    $!Varset |turbine_y| = |t1_y|
    $!Varset |turbine_z| = |t1_z|
    $!Varset |turbine_name| = "|t1_name|"
    $!Varset |t1_undone| = 0

$!ELSEIF |t2_undone| == 1
    $!Varset |turbine_x| = |t2_x|
    $!Varset |turbine_y| = |t2_y|
    $!Varset |turbine_z| = |t2_z|
    $!Varset |turbine_name| = "|t2_name|"
    $!Varset |t2_undone| = 0

$!ELSEIF |t3_undone| == 1
    $!Varset |turbine_x| = |t3_x|
    $!Varset |turbine_y| = |t3_y|
    $!Varset |turbine_z| = |t3_z|
    $!Varset |turbine_name| = "|t3_name|"
    $!Varset |t3_undone| = 0

$!ELSEIF |t4_undone| == 1
    $!Varset |turbine_x| = |t4_x|
    $!Varset |turbine_y| = |t4_y|
    $!Varset |turbine_z| = |t4_z|
    $!Varset |turbine_name| = "|t4_name|"
    $!Varset |t4_undone| = 0

$!ELSEIF |t5_undone| == 1
    $!Varset |turbine_x| = |t5_x|
    $!Varset |turbine_y| = |t5_y|
    $!Varset |turbine_z| = |t5_z|
    $!Varset |turbine_name| = "|t5_name|"
    $!Varset |t5_undone| = 0

$!ELSEIF |t6_undone| == 1
    $!Varset |turbine_x| = |t6_x|
    $!Varset |turbine_y| = |t6_y|
    $!Varset |turbine_z| = |t6_z|
    $!Varset |turbine_name| = "|t6_name|"
    $!Varset |t6_undone| = 0

$!ELSEIF |t7_undone| == 1

```

```

|turbine_x| = |t7_x|
    $!Varset |turbine_y| = |t7_y|
    $!Varset |turbine_z| = |t7_z|
    $!Varset |turbine_name| = "|t7_name|"
    $!Varset |t7_undone| = 0

$!ELSEIF |t8_undone| == 1
    $!Varset |turbine_x| = |t8_x|
    $!Varset |turbine_y| = |t8_y|
    $!Varset |turbine_z| = |t8_z|
    $!Varset |turbine_name| = "|t8_name|"
    $!Varset |t8_undone| = 0

$!ELSEIF |t9_undone| == 1
    $!Varset |turbine_x| = |t9_x|
    $!Varset |turbine_y| = |t9_y|
    $!Varset |turbine_z| = |t9_z|
    $!Varset |turbine_name| = "|t9_name|"
    $!Varset |t9_undone| = 0

$!ELSEIF |t10_undone| == 1
    $!Varset |turbine_x| = |t10_x|
    $!Varset |turbine_y| = |t10_y|
    $!Varset |turbine_z| = |t10_z|
    $!Varset |turbine_name| = "|t10_name|"
    $!Varset |t10_undone| = 0

$!ELSEIF |t11_undone| == 1
    $!Varset |turbine_x| = |t11_x|
    $!Varset |turbine_y| = |t11_y|
    $!Varset |turbine_z| = |t11_z|
    $!Varset |turbine_name| = "|t11_name|"
    $!Varset |t11_undone| = 0

$!ELSEIF |t12_undone| == 1
    $!Varset |turbine_x| = |t12_x|
    $!Varset |turbine_y| = |t12_y|
    $!Varset |turbine_z| = |t12_z|
    $!Varset |turbine_name| = "|t12_name|"
    $!Varset |t12_undone| = 0

$!ENDIF

$!VARSET |m| = 0 # reset counter

$!LOOP |num_distance|

```

```

$!FRAMECONTROL ACTIVATEBYNUMBER

FRAME=1

    $!VARSET |m| += 1 # add 1 to counter
    $!VARSET |Current_slice| = (|start_zone|
+ 1 + 2*(|n|-1)*|num_distance| + |m| - 1))
    $!VARSET |Current_circle| = (|start_zone|
+ 2*(|n|-1)*|num_distance| + |m|)
    $!VARSET |Current_frame| = (1 + (|n|-
1)*|num_distance| + |m|)

    $!IF |m| == 1
        $!VARSET |distance| = |dist_1|
    $!ELSEIF |m| == 2
        $!VARSET |distance| = |dist_2|
    $!ELSEIF |m| == 3
        $!VARSET |distance| = |dist_3|
    $!ELSEIF |m| == 4
        $!VARSET |distance| = |dist_4|
    $!ELSEIF |m| == 5
        $!VARSET |distance| = |dist_5|
    $!ELSEIF |m| == 6
        $!VARSET |distance| = |dist_6|
    $!ELSEIF |m| == 7
        $!VARSET |distance| = |dist_7|
    $!ELSEIF |m| == 8
        $!VARSET |distance| = |dist_8|
    $!ELSEIF |m| == 9
        $!VARSET |distance| = |dist_9|
    $!ELSEIF |m| == 10
        $!VARSET |distance| =
|dist_10|
    $!ELSEIF |m| == 11
        $!VARSET |distance| =
|dist_11|
    $!ELSEIF |m| == 12
        $!VARSET |distance| =
|dist_12|
    $!ELSEIF |m| == 13
        $!VARSET |distance| =
|dist_13|
    $!ELSEIF |m| == 14
        $!VARSET |distance| =
|dist_14|
    $!ELSEIF |m| == 15
        $!VARSET |distance| =
|dist_15|

```



```

$!ELSEIF |m| == 16
$!VARSET
|distance| = |dist_16|
$!ELSEIF |m| == 17
$!VARSET |distance| =
|dist_17|
$!ELSEIF |m| == 18
$!VARSET |distance| =
|dist_18|
$!ELSEIF |m| == 19
$!VARSET |distance| =
|dist_19|
$!ELSEIF |m| == 20
$!VARSET |distance| =
|dist_20|
$!ENDIF

$!GLOBALTHREED SLICE{NORMAL{Y = 1}}
$!GLOBALTHREED SLICE{NORMAL{Z = 0}}
$!GLOBALTHREED SLICE{ORIGIN{Y =
(|turbine_R|*|distance| + |turbine_y|)}}
$!CREATESLICEZONEFROMPLANE
SLICESOURCE = VOLUMEZONES
FORCEEXTRACTIONTOSINGLEZONE = YES
COPYCELLCENTEREDVALUES = NO

$!VARSET |slice_name| =
"|turbine_name| / distance |distance%.2f|R"

$!CREATENEWFRAME
XYPOS
{
X = -2.75
Y = (5.0 -
0.25*|Current_frame|)
}
WIDTH = 5
HEIGHT = 3
$!PLOTTYPE = CARTESIAN2D
$!TWOAXIS YDETAIL{VARNUM = 3}
$!GLOBALCONTOUR 1 VAR = 6
$!CONTOURLEVELS RESETTONICE
CONTOURGROUP = 1
APPROXNUMVALUES = 15
$!FIELDLAYERS SHOWCONTOUR = YES

```

```

$!GLOBALCONTOUR 1
COLORMAPFILTER{COLORMAPDISTRIBUTION = CONTINUOUS}
$!ACTIVEFIELDMAPS = [|Current_slice|]

$!ATTACHTEXT TEXT = "|slice_name|"

$!FRAMECONTROL ACTIVATEBYNUMBER
FRAME=|Current_frame|

$!CREATECIRCULARZONE
IMAX = 20
JMAX = 80
KMAX = 1
X = |turbine_x|
Y = |turbine_z|
Z1 = 0
Z2 = 0
RADIUS = |turbine_R|

$!RENAMEDATASETZONE
ZONE = |Current_slice| # For slice
NAME = "|slice_name| SLICE"

$!RENAMEDATASETZONE
ZONE = |Current_circle| # For turbine
NAME = "|slice_name| CIRCLE"

$!LINEARINTERPOLATE
SOURCEZONES = [|Current_slice|]
DESTINATIONZONE = |Current_circle|
VARLIST = [2,4-7]
LINEARINTERPCONST = 0
LINEARINTERPMODE = SETTOCONST

$!IF |distance| >= 0
$!Varset |Dist_int| =
(floor(|distance|))
$!Varset |Dist_dec| =
((|distance| - |Dist_int|)*100)
$!Varset |Fname_distR_Tnum|
= "|Dist_int%d|point|Dist_dec%d|R_|turbine_name|"
$!ELSE
$!Varset |Dist_int| =
(abs((ceil(|distance|))))

```

```

$!Varset |Dist_dec| = (-
(|distance| + |Dist_int|)*100)
$!Varset |Fname_distR_Tnum|
= "-|Dist_int%d|point|Dist_dec%d|R_|turbine_name|"
$!ENDIF

$!WRITEDATASET
"|MFB|/|Filename|_|Fname_distR_Tnum|.dat"
INCLUDETEXT = NO
INCLUDEGEOM = NO
INCLUDECUSTOMLABELS = NO
ASSOCIATELAYOUTWITHDATAFILE = NO
ZONELIST = [|Current_circle|]
VARPOSITIONLIST = [1-7]
BINARY = NO
USEPOINTFORMAT = YES
PRECISION = 9
TECPLOTVERSIONTOWRITE =

TECPLOTCURRENT

$!ENDLOOP
$!ENDLOOP

$!DRAWGRAPHICS YES
$!REDRAW

```



Appendix G Matlab program

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% file: integration_turbines_auto.m
% author: Oskar Thulin
% description: program that integrates
the Vy^3 over circular disk
% input is .txt file from Tecplot
where the data is sorted
% pointwise in ASCII code. Can
handle asymmetric flow. Input
% file has to have the turbine
number in the end of the file
% name (before file extension).
% created: 1/29/2011
% last change: 5/19/2011
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clc;
close all;
clear all;

%% control
plot_on_off = 0; % on/off for
functions 1 - on, 0 - off
calculations_on_off = 1;
choice = 7; % 1-> average
velocity
% 2-> average PA
% 3-> average T
% 4-> TSR
% 5-> all
% 6-> average PA on
different quarter
% 7-> average velocity
on different circles
% with different radii
%% opens the txt file
n_data = 7; % n_data is the number of
data columns in the file exported
% from Tecplot

Turbines = [1,2];
Distance = [{'2point0'}];
l_Turbines = length(Turbines);
l_Distance = length(Distance);

for T_num=1:l_Turbines

for D_num=1:l_Distance

file=['mesh_florian_shear_velocity_1_1_
4R_', Distance{D_num}, 'R_T',
num2str(Turbines(T_num)), '.dat'];
fid=fopen(file,'r');

for j=1:n_data+15; % skipping
definition lines in data text file
fgetl(fid);
end
% knows where it shall write it
% [numeric,txt] =
xlsread('/Users/thulino/Dropbox/UW
G44 -
Me600/Matlab/Turbine_Efficiency_old.xl
s','Extended_channel','A')

%% reading and loading variables
X,Y,Z,Vy, Vy^3
[data,count] =
fscanf(fid,'%f',[n_data inf]); %
fclose(fid);

l = length(data); %
number of nodes

X_x = data(1,:);
X_y = data(2,:);
X_z = data(3,:);
rho = data(4,:);
V_x = data(5,:);
V_y = -data(6,:); % only
interested in the positive value
V_z = data(7,:);

%% initializing constants
rho = 998.2;
R = 5.53;
A = pi * R^2;
Omega=1.780235837;
type_turbine_str =
Turbines(T_num);
turbine_position_matrix = [-8.295
0 0;... % matrix with positions of
8.295 0 0;... % turbines,
used to subst-
-9.6775 -6.636 0; ... % ract
turbine center from
8.295 -6.636 0;... % data in
order to generate
-11.06 -13.272 0;... % polar
data from the X-Z
8.295 -13.272 0;... % plane
-12.4425 -19.908 0;...
8.295 -19.908 0;...
-13.8250 -22.120 0;...
-13.8250 -27.650 0;...
-13.8250 -33.180 0;...
-16.5900 -35.392 0;...
-13.8250 -38.710 0;...
-16.5900 -40.922 0;...
-13.8250 -44.240 0;...
12.4425 -44.240 0;...
24.885 -44.240 0;...
-16.59 -46.452 0;...
6.9125 -46.452 0;...
19.3550 -46.452 0;...
4.1475 -48.664 0;...
16.5900 -48.664 0;...
-16.5900 -51.982 0;...
9.6775 -51.982 0;...
22.120 -51.982 0;...

5.5300 -54.194 0;...
17.9725 -54.194 0;...
-16.5900 -57.512 0;...
12.4425 -57.512 0;...
24.8850 -57.512 0;...
6.9125 -59.724 0;...
19.355 -59.724 0;...
4.1475 -61.936 0;...
16.5900 -61.936 0;...
9.6775 -65.254 0;...
22.120 -65.254 0;...
5.5300 -67.466 0;...
17.9725 -67.466 0;];
turbine_center_X_x =
turbine_position_matrix(type_turbine_s
tr,1);
turbine_center_X_z =
turbine_position_matrix(type_turbine_s
tr,3);

%% plot cartesian
if plot_on_off
% plot nodes in X-Z plane
clf
figure(1)
plot(X_x,X_z,'*')
title('node positions')
xlabel('x [m]')
ylabel('z [m]')

% plot Vy distribution in 3D
figure(2)
tri = delaunay(X_x,X_z);
[rows,columns] = size(tri);
h = trisurf(tri, X_x, X_z, V_y);
axis vis3d
lighting phong
shading interp
colorbar EastOutside
xlabel('x [m]')
ylabel('z [m]')
title('V_y [-m/s]')
end

%% calculating polar coordinates
[theta,r] = cart2pol(X_x-x
turbine_center_X_x,X_z-
turbine_center_X_z);
% recalculation of cart. to polar
coord. syst.

if plot_on_off
% plot nodes, theta and r in
cart. plane
figure(4)
plot(r,theta,'*')
xlim([-1 6])
ylim([-4 4])
xlabel('r [m]')
ylabel('\theta [rad]')
title('polar coordinates in
cartesian coord. syst')

% plot nodes, theta and r in
polar plane
figure(5)
polar(theta,r,'*')
xlabel('r [m]')

```




```

ylabel('\theta [rad]')
title('polar coordinates in polar
coord. syst')
end

%% data analysis/processing
if calculations_on_off
    Data = [data; theta; r]; %
concatenation of old data and
                                % new polar
data for respective node
    k = 1;
    new_row_index = [];

    % analyze when the radial set
of points ends and the next take over
    for i=1:-1
        if r(i) > r(i+1)
            new_row_index(k) = i;
            k = k + 1;
        end
    end

    % number of nodes in
circumferential and radial direction
(product = l)
    circumferential_elements =
length(new_row_index) + 1;
    radial_elements =
l/circumferential_elements;
end

%% calculations
if calculations_on_off

    % initializing vectors for faster
calculations
    V_x_ave = zeros(l,1);
    V_y_ave = zeros(l,1);
    V_z_ave = zeros(l,1);
    r_ave = zeros(l,1);
    V_theta = zeros(l,1);
    V_y_3_ave = zeros(l,1);
    dA = zeros(l,1);
    PA_ave = zeros(l,1);
    V_y_ave_area = zeros(l,1);
    T_ave = zeros(l,1);
    TSR_average = zeros(l,1);

    % Vy, dA & available power
calculations (3 methods, 3rd prob. best)
    for j=1:-1
        if sum(j == new_row_index)
            % neglects all elements
where the following radius is less than
            % present
        else
            % averageing Vy between
the two closest nodes in the
            % radial direction
            V_x_ave(j,1) = (0.5 *
(V_x(j) + V_x(j+1)));
            V_y_ave(j,1) = (0.5 *
(V_y(j) + V_y(j+1)));
            V_y_3_ave(j,1) = (0.5 *
(V_y(j)^3 + V_y(j+1)^3));
            V_z_ave(j,1) = (0.5 * (V_z(j)
+ V_z(j+1)));
            theta_ave(j,1) = (0.5 *
(theta(j) + theta(j+1)));

            r_ave(j,1) = (0.5 * (r(j) +
r(j+1)));

            V_theta(j,1) = -
V_x_ave(j,1)*sin(theta_ave(j,1)) +
V_z_ave(j,1)*cos(theta_ave(j,1));

            % calculates the dA
segments
            dA(j,1) =
pi/circumferential_elements *
((r(j+1))^2 - ...
            (r(j))^2);

            % calculates the available
power for each dA segment
            V_y_ave_area(j) = dA(j) *
V_y_ave(j);
            PA_ave(j) = (0.5 * rho *
dA(j)*V_y_ave(j));
            T_ave(j) = (rho *
V_y_ave(j) * r_ave(j) * V_theta(j) *
dA(j));
        end

        end

        % summation of available
power over all dAs
        V_y_ave_total_sum =
sum(V_y_ave_area/sum(dA));
        PA_sum = sum(PA_ave);
        T_sum = sum(T_ave);
        TSR = Omega * R /
V_y_ave_total_sum;

        % summation of power for
different quarter
        j = 1:400;
        Pa_1 = sum(PA_ave(j));
        j = 401:800;
        Pa_2 = sum(PA_ave(j));
        j = 801:1200;
        Pa_3 = sum(PA_ave(j));
        j = 1201:1600;
        Pa_4 = sum(PA_ave(j));
        % summation of velocity for
different radii
        V_y_1 = 0;
        V_y_2 = 0;
        V_y_3 = 0;
        V_y_4 = 0;
        V_y_5 = 0;
        dA_1 = 0;
        dA_2 = 0;
        dA_3 = 0;
        dA_4 = 0;
        dA_5 = 0;
        j = 0;
        for i=1:79
            V_y_1 = V_y_1 +
V_y_ave_area(j+1) + V_y_ave_area(j+2)
+ V_y_ave_area(j+3) +
V_y_ave_area(j+4);
            dA_1 = dA_1 + dA(j+1) +
dA(j+2) + dA(j+3) + dA(j+4);
            V_y_2 = V_y_2 +
V_y_ave_area(j+5) + V_y_ave_area(j+6)
+ V_y_ave_area(j+7) +
V_y_ave_area(j+8);
            dA_2 = dA_2 + dA(j+5) +
dA(j+6) + dA(j+7) + dA(j+8);

            V_y_3 = V_y_3 +
V_y_ave_area(j+9) +
V_y_ave_area(j+10) +
V_y_ave_area(j+11) +
V_y_ave_area(j+12);
            dA_3 = dA_3 + dA(j+9) +
dA(j+10) + dA(j+11) + dA(j+12);
            V_y_4 = V_y_4 +
V_y_ave_area(j+13) +
V_y_ave_area(j+14) +
V_y_ave_area(j+15) +
V_y_ave_area(j+16);
            dA_4 = dA_4 + dA(j+13) +
dA(j+14) + dA(j+15) + dA(j+16);
            V_y_5 = V_y_5 +
V_y_ave_area(j+17) +
V_y_ave_area(j+18) +
V_y_ave_area(j+19) +
V_y_ave_area(j+20);
            dA_5 = dA_5 + dA(j+17) +
dA(j+18) + dA(j+19) + dA(j+20);
            j = j + 20;
        end
        V_y_1 = V_y_1 / dA_1;
        V_y_2 = V_y_2 / dA_2;
        V_y_3 = V_y_3 / dA_3;
        V_y_4 = V_y_4 / dA_4;
        V_y_5 = V_y_5 / dA_5;
    end

    % printing the result in the
command window
    current_turbine =
num2str(Turbines(T_num));
    fprintf('%s\nV_ave_total_sum = %3.3f
[m/s]\n\n', current_turbine,...
            V_y_ave_total_sum);

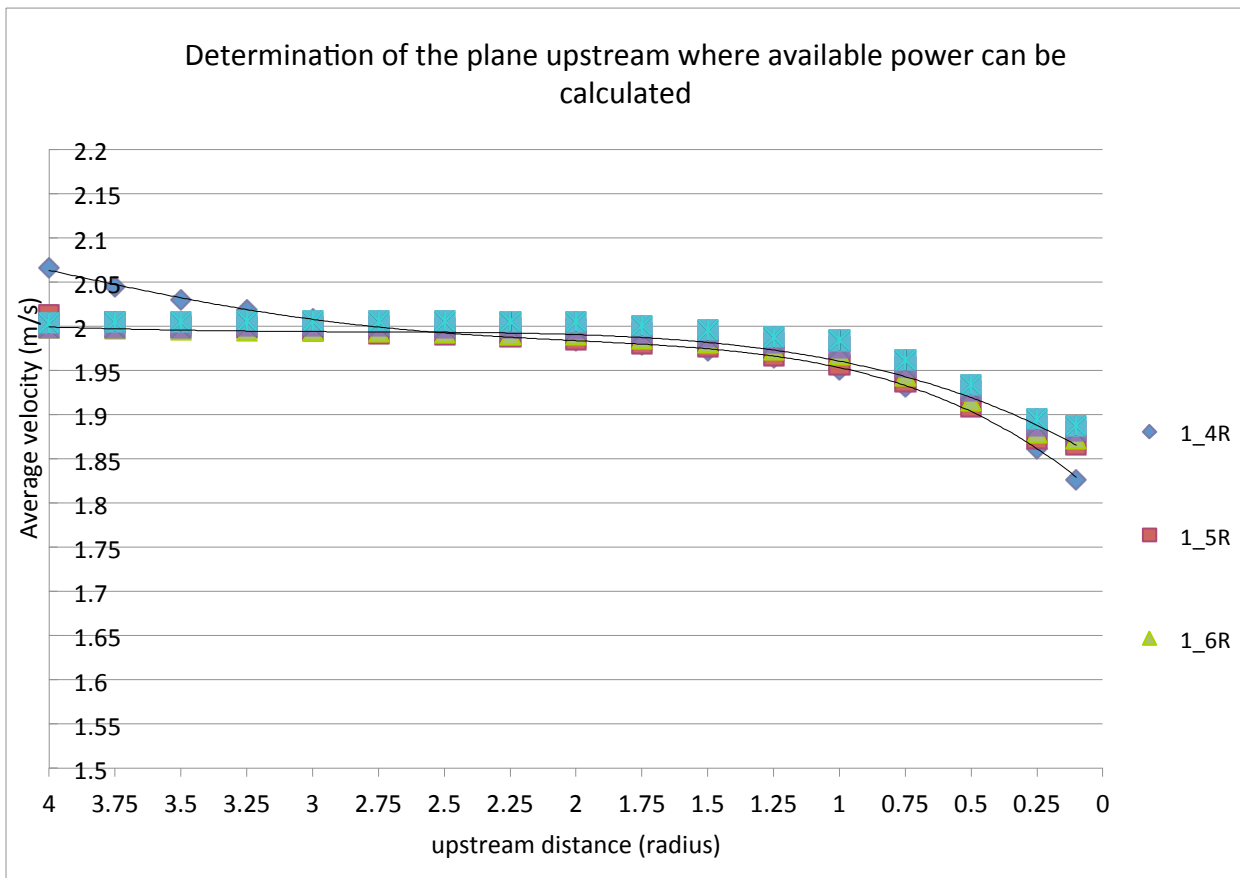
    fprintf('%s\nPA_indirect_ave_sum =
%3.3f [kW]\n\n',current_turbine,...
            PA_sum/1000);

    fprintf('%s\nT_indirect_ave_sum =
%3.3f [kNm]\n\n',current_turbine,
            T_sum/1000);
    printf('%s\nTSR = %3.3f
[kNm]\n\n',current_turbine, TSR);

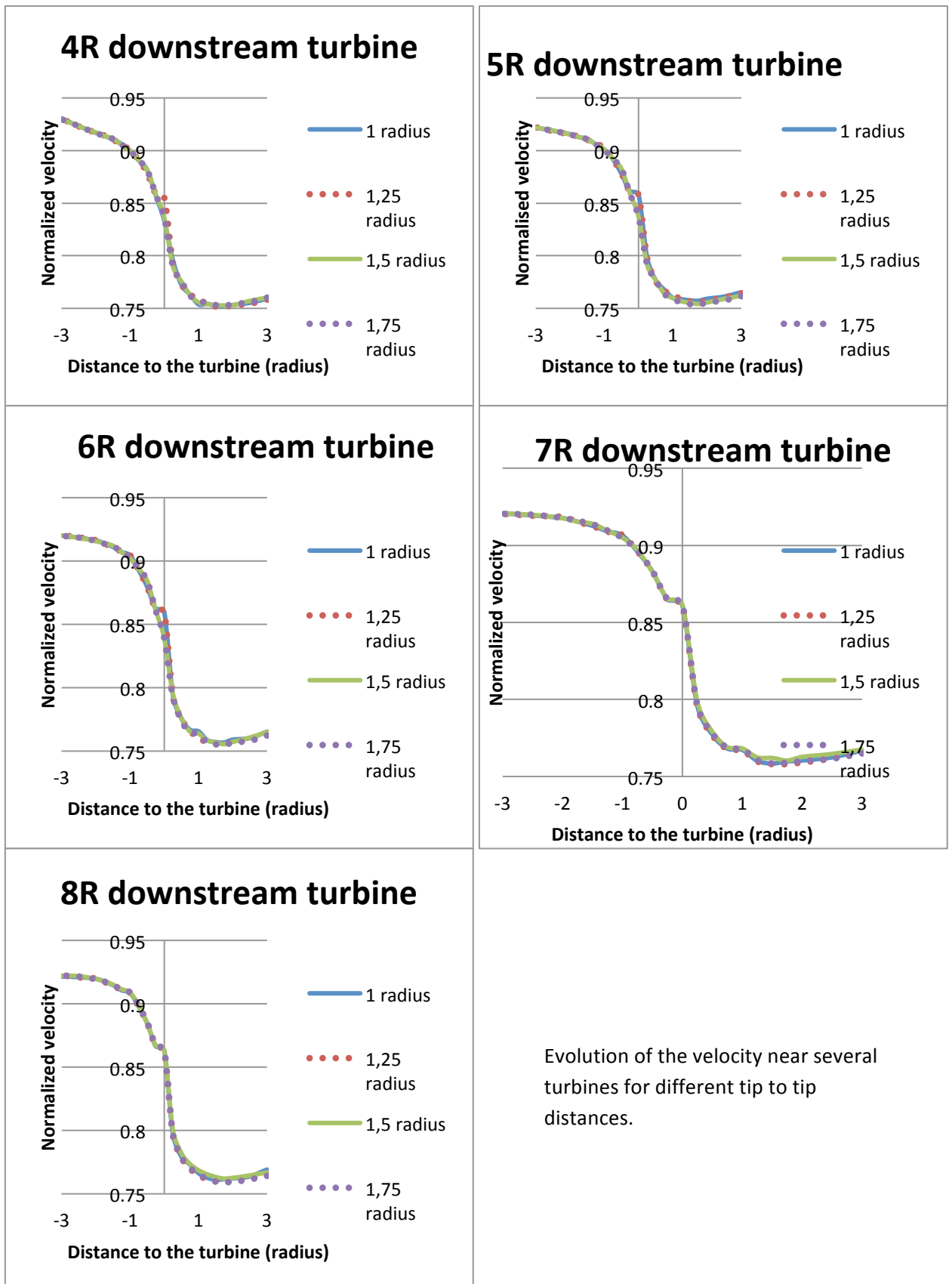
    fprintf('%s\nPa_LUP = %3.3f [kW]\n\n',
current_turbine,Pa_1/1000);
    fprintf('%s\nPA_RUP = %3.3f
[kW]\n\n',current_turbine,Pa_2/1000);
    fprintf('%s\nPA_RDn = %3.3f
[kW]\n\n',current_turbine,Pa_3/1000);
    fprintf('%s\nPA_LDn = %3.3f
[kNm]\n\n',current_turbine,Pa_4/1000);
    fprintf('%s\nV_ave_1 = %3.3f
[m/s]\n\n', current_turbine, V_y_1);
        fprintf('%s\nV_ave_2 = %3.3f
[m/s]\n\n',current_turbine,V_y_2);
        fprintf('%s\nV_ave_3 = %3.3f
[m/s]\n\n',current_turbine,V_y_3);
        fprintf('%s\nV_ave_4 = %3.3f
[m/s]\n\n',current_turbine, V_y_4);
        fprintf('%s\nV_ave_5 = %3.3f
[m/s]\n\n',current_turbine, V_y_5);
    fprintf('%s\nV_ave = %3.3f
[m/s]\n\n',current_turbine,...
            V_y_ave_total_sum);
    end
end

```

Appendix H Chart of the velocity before the turbine



Appendix I Chart of the velocity near downstream turbines





References

- [1] Amir Teymour Javaherchi. *Numerical Modeling of Tidal Turbines: Methodology Development and Potential Physical Environmental Effects*. Master thesis, November 2010
- [2] Aurélie Gosset and Guillaume Floriout. *Optimization of Power Extraction in an array of Marine Hydrokinetic Turbines*. Projet de fin d'étude, November 2010
- [3] Xiaojing Sun. *Numerical and experimental investigation of tidal current energy extraction*. University of Edinburgh, 2008.
- [4] L.J. Fingersh D.W. Jager J.R. Cotrell S. Schreck M.M. Hand, D.A. Simms and S.M. Larwood. *Unsteady aerodynamics experiment phase VI: Wind tunnel test configurations and available data campaigns*. National Renewable Energy Laboratory (NREL) / TP-500-29955, December 2001.
- [5] Claire Legrand. *Assessment of Tidal Energy Resource*. The European Marine Energy Center, London, 2009.
- [6] Samuel Gooch. *Siting Methodologies for Tidal In-Stream Energy Conversion (TISEC) Systems*. University of Washington, 2009.
- [7] Stephen B. Pope. *Turbulent flows*. Cambridge University press, 2000.
- [8] B. Munsen, D. Young and T. Okiishi. *Fundamentals of Fluid Mechanics, fifth edition*. Wiley, (Iowa State University), 2006.
- [9] Theory guide. ANSYS *Fluent*® 13.0, November 2010.
- [10] User's guide. ANSYS *Fluent*® 13.0, November 2010.
- [11] Michael Pidwirny & Scott Jones. *Fundamentals of Physical Geography*. Available on the Internet: <http://www.physicalgeography.net/fundamentals/8r.html>
- [12] P. Kundu and I. Cohen. *Fluid Mechanics, forth edition*. Academic Press 2008.
- [13] Eric Meindl and David Gilhousen. *Determining the Power-Law Wind-Profile Exponent under Near-Neutral Stability Conditions at Sea*. National Data Buoy Center, Stennis Space Center, Mississippi, 1994
- [14] T. Javaherchi, S. Antheaume, A. Aliseda. *Hierarchical methodology for the numerical simulation of the flow field around and in the wake of horizontal axis wind turbines: Rotating reference frame, blade element method and actuator disk model*, Wind Eng. 38[2], PP 181-202, 2014.

