REVISTA
INTERDISCIPLINAR DE
PESQUISA EM
ENGENHARIA

CILAMCE
2016

XXXVII IBERIAN LATIN AMERICAN CONGRESS
ON COMPUTATIONAL METHODS IN ENGINEERING
BRASÍLIA - DF - BRAZIL

STUDIES ON HYDRODYNAMIC TORQUE ON GUIDE VANES OF FRANCIS TURBINES UNDER STEADY STATE OPERATION

Pedro Henrique Bastos Menezes

phbmenezes@gmail.com

ELETROBRAS - ELETRONORTE & UNIVERSITY OF BRASILIA

Brasilia, Distrito Federal, Brazil

Abstract. *According to the Brazilian National Electric Energy Agency (ANEEL), over 60% of electricity produced in Brazil comes from hydraulic resources. Most turbines in Brazilian hydroelectric power plants are Francis type, in which the flow is controlled by rotating guide vanes. Any resistive hydraulic torque on these vanes are overcome by servo motors. The estimation of resulting torque was performed by Computational Fluid Dynamics (CFD), using techniques to simulate the flow from the vanes to the exit of the draft tube. The geometries and experimental data were collected for free on Francis-99 website, held by the Norwegian University of Science and Technology (NTNU). Based on the axi-periodic geometry, the flow was divided into channels. One of them was divided into three isolated tracks with independent fluid meshes and suitable boundary conditions, considering steady state operation. In order to realise the simulation, the softwares OpenFOAM and ParaView were used. Computational techniques as Mixing Plane, General Grid Interface and Multiple Rotating Frames of Reference were applied to make the interfaces between sections and to emulate effects of rotation. The pressure field on the surface of a guide vane was then integrated with respect to area so that the resulting hydraulic torque was found.*

Keywords: *CFD, Turbines, Francis, OpenFOAM, ParaView*

1 INTRODUCTION

According to ANEEL (2015), most of the Brazilian electric power grid is based on hydraulic sources, like Hydroelectric Power Plants (HPP) using turbines exhibiting efficiency above 90%. This value is linked to a best efficiency point of operation and when the turbine operates in a different load, the amount of hydraulic energy effectively converted to electric energy decreases. When in part or high load the efficiency goes down, representing cumulative loss of energy production and thus less profit from energy selling.

1.1 Flow rate control on Francis turbines

In some Francis turbines, the speed of operation in the best efficiency point is constant and directly related to the volumetric flow rate and the resistive torque on the runner, both variable along the day. The flow rate must be constantly controlled by several synchronised rotating guide vanes operated by servo motors. In order to decrease the flow rate, the hydraulic torque must be overcome. Thus good estimations for the value of torque are essential to guarantee the guide vanes whole range of operation.

1.2 Project Francis-99

According to the Norwegian Hydropower Centre, located at the Norwegian University of Science and Technology (NTNU):

> ”Francis-99 is a series of three workshops, which provides an open access of the complete design and data of a model Francis turbine. It provides an open platform to the hydropower researchers and it gives the possibility to explore their capabilities and enhance their skills. In fact, this is the main objective of the workshops. The researchers can use these data and perform numerical studies by applying different tools and techniques.”

The content available on-line includes the complete geometry and experimental data of a real Francis turbine scaled model. Three different points of operation were tested: Part Load, Best Efficiency Point and High Load. For this work, just the Best Efficiency Point was analysed.

1.3 Interface techniques

In order to simulate the whole turbine, the flow dominion was divided into three different sections and meshes were created separately for each one. The techniques used to provide interface between two surfaces were the General Grid Interface (GGI) and the Mixing Plane.

Since the nodes of two meshes sharing a common surface may not be in the same points, they may face geometrical incompatibility. The GGI makes this coupling based on an area-weighted division of a property flux through elements of one control volume of a mesh to the intersecting ones of the other mesh (Beaudoin et al., 2008). Using this interface technique, the flux through different meshes can be calculated even if the common surface is offsetted or rotated in space. A more specific version of the GGI is the cyclicGgi, which enables axi-periodic rotating machines to be dismembered into single repeatable sections. The cyclicGgi is used to include a cyclic boundary condition on both the sliced surfaces, making their properties equal, point by point.

Considering a fixed non-rotating frame of reference linked to the guide vanes and a rotating one linked to the runner, the interface between these meshes are fundamental to the simulation of turbines. In order to emulate a steady-state regime and provide coupling to geometrically incompatible meshes, the Mixing Plane technique was used. The base of it is the averaging of properties along the circumferences around the rotating axis (Beaudoin et al., 2014). The discretization of it creates ribbons of average value in which both surfaces takes values from. For purely radial turbines, the discretization is made along the axial direction and for purely axial ones, it is made along the radius.

1.4 Softwares

Two different softwares were used for the solving and the post-processing steps of this project:

- *FOAM-Extend 3.1*: open-source solver; a branch containing community contributions based on the original *OpenFOAM*;
- *ParaView 4.1.0*: open-source software for post-processing.

The open-source softwares were used on the Ubuntu 14.04 LTS version of Linux OS. For the sake of simplicity *OpenFOAM* will be referred instead of FOAM-Extend 3.1 along this document.

1.5 Objective

Based on Computational Fluid Dynamics (CFD) and using techniques like General Grid Interface (GGI), Mixing Plane and Multiple Rotating Frames of Reference (MRF), find an estimative for the hydrodynamic torque on guide vanes of Francis turbines under a steady-state approximation.

2 METHODOLOGY

Some simplifications were adopted in order to avoid the analysis of an unsteady-operating complete Francis turbine and any runner-vanes interactions:

- steady-state regime approximation using Mixing Plane;
- axi-periodic tracks simulation with cyclic condition instead of the whole turbine;
- due to the lack of proper knowledge of turbulence models, laminar flow was used in the simulation;
- free-slip condition on every wall.

2.1 Preparing the geometries

The geometry for the whole turbine was provided by Project Francis-99 and then sliced into three different tracks:

- Track 1: Guide vanes
  This Track begins right after the stay vanes, includes the suction side of one guide vane and the pressure side of a second one, ending on runner's entrance. The resulting mesh represents 1/28 of the system's guide vanes and its patches can be seen in Fig.1. Its corresponding unstructured mesh is made of tetrahedral volumes with higher density on skewed sections of its inlet and outlet.
- Track 2: Runner blades
  The second Track starts on runner's inlet, includes the suction side of a runner blade, the pressure side of another one and a whole splitter, ending in the draft tube's inlet. The resulting mesh represents 1/15th of the system's runner blades an its unstructured mesh is made of tetrahedral volumes with higher density on the cyclic surfaces. This Track's patches can be seen in Fig.2.
- Track 3: Draft tube
  The last Track starts on runner's outlet and includes the whole draft tube, ending in its outlet. Its structured mesh is made of hexahedral volumes with higher density on the cyclic surfaces. This Track's patches can be seen in Fig.3.

After the geometries for the three isolated Tracks were ready, they were put together and exported to *ANSYS FLUENT* `.msh` format, using *ASCII* codification. Since the interface patches are internal to the resulting mesh, the following command must be used in order to properly rescale, convert the geometry to a *OpenFOAM* compatible format and generate a log file of the process:

```
fluent3DMeshToFoam -scale 0.001 2>&1 | tee log.fluent3D
```

After the conversion, some characteristics as max skewness and non-orthogonality can be check with the command:

```
checkMesh 2>&1 | tee log.checkMesh
```

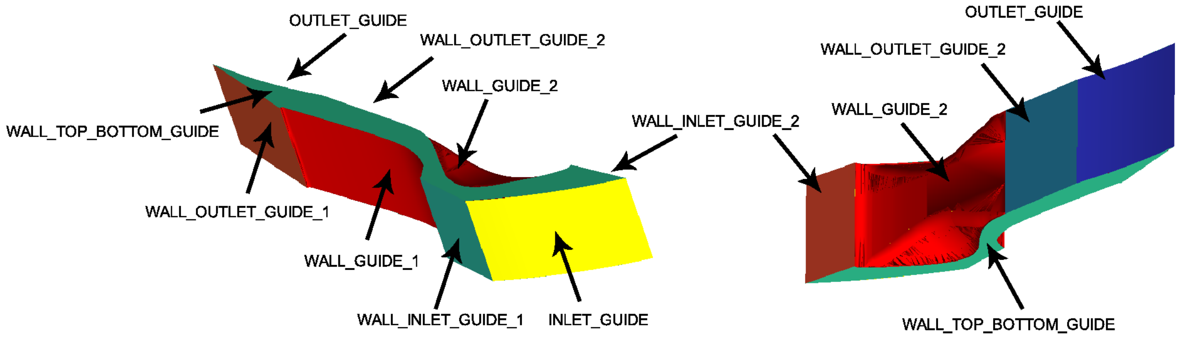

Figure 1: Patches for Track 1 - Guide vanes

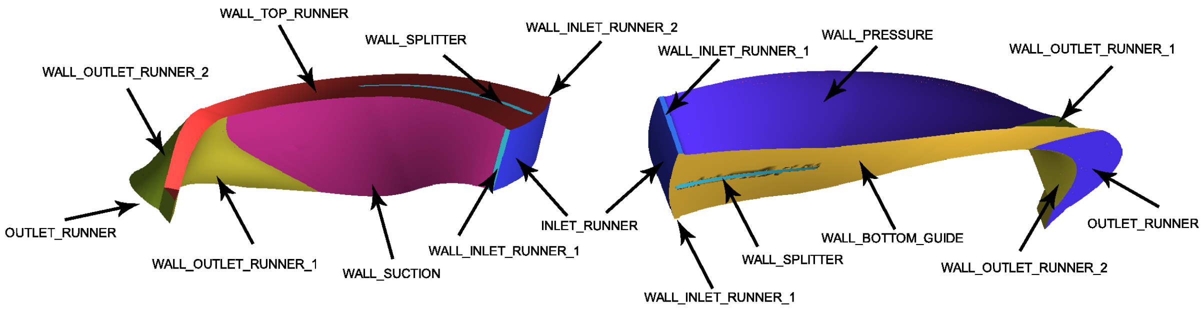

Figure 2: Patches for Track 2 - Runner blades

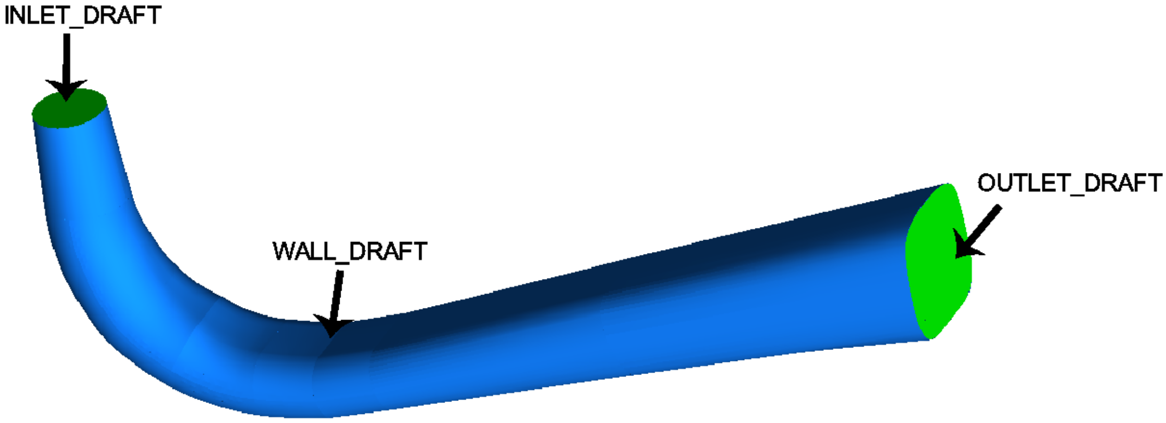

Figure 3: Patches for Track 3 - Draft tube

2.2 Applying interfaces

Patch types were declared on `constant/polyMesh/boundary` file as `patch`, `wall`, `cyclicGgi` and `mixingPlane` types, depending on the surface location.

The interface used to connect Tracks 1 and 2 was the `mixingPlane` with axial discretization (`stackAxis Z`), while between Tracks 2 and 3 it was used the radial format (`stackAxis R`).

Since Track 1 is only 1/28 of the complete guide vanes system, the angle value used in its `cyclicGgi` setting is defined by `rotationAngle 12.857` with `separationOffset (0 0 0)`. The Track 2 represent 1/15 of the runner, so it was used `rotationAngle 24`.

2.3 Boundary conditions

For the inlet and outlet boundary conditions, settings used on Francis-99 were adopted and the types used on every patch can be found in Table 1.

A customized `groovyBC` boundary type was created so velocity at inlet could be set in terms of radial and tangential components:

```
INLET_GUIDE
{type groovyBC;
variables (
"vr=-1.408650; vt=-2.099457;"
"vresult=sqrt(pos().x*pos().x+pos().y*pos().y);"
"vxr= vr*pos().x/vresult; vyr= vr*pos().y/vresult;"
"vxt=-vt*pos().y/vresult; vyt= vt*pos().x/vresult;"
"vxtotal=vxr+vxt; vytotal=vyr+vyt;"
);
valueExpression "vector(vxtotal,vytotal,0)";
value uniform (0 0 0);}
```

2.4 Applying the Multiple Rotating Frames of Reference (MRF)

In order to avoid an increase in complexity due to a dynamic mesh, the effects of rotation on Track 2 were emulated using a rotating frame of reference while Tracks 1 and 3 used a fixed in space frame of reference. The Coriollis and centrifugal pseudo-forces are added to the Navier-Stokes equations resulting on rotating effects, even though the mesh is actually static.

For that, the `MRFSimpleFoam` is used as a solver for *OpenFOAM* and the whole Track 2 is defined as a cellzone rotating at 335.4rpm, set in file `constant/MRFZone`.

2.5 Parameters used

The choosen linear equation solver for both velocity and pressure was the Biconjugate Gradient Stabilized Method (BiCGSTab). The relaxation factor for velocity was set as 0.6 and 0.3 for pressure, while the tolerances were set as 10^{-8} and 10^{-7}, respectively.

2.6 Resulting torque

The torque contribution $\vec{d_T}$ due to a discrete control volume can be estimated as the cross product of vectors distance $\vec{d_{xy}}$ from the rotating axis of the guide vane (with null Z component) and force $\vec{F}$ due to pressure acting in its face, as seen in Eq.1 . The magnitude of vector $\vec{F}$ is given by pressure times area and its direction by an unit vector $\vec{n}$ orthogonal to the face of the analysed control volume, resulting on Eq. 2.

$$d_T = \vec{d_{xy}} \times \vec{F} \tag{1}$$

$$d_T = \vec{d_{xy}} \times \vec{n}(pS) \tag{2}$$

After the pressure field is established, the torque contribution $\vec{d_T}$ from every control volume on the surface of the guide vane can be integrated with respect to its area. That value is the resulting torque T_T applied to the guide vanes. Due the discretization of the surface the integration process become a sum, as represented on Eq.3.

$$T_T = \sum_{i=1}^{k} \vec{d}_{xy,i} \times \vec{n}(p_i S_i) \tag{3}$$

All these operations were made in the post-processing step of the simulation using the software *ParaView*. The resulting pipeline and filters used are ilustrated on Fig. 4.

Table 1: Boundary conditions for all patches

PATCH	BOUNDARY CONDITIONS (U)	BOUNDARY CONDITIONS (p)
INLET_GUIDE	type groovyBC;	type zeroGradient;
OUTLET_GUIDE	type mixingPlane;	type mixingPlane;
INLET_RUNNER	type mixingPlane;	type mixingPlane;
OUTLET_RUNNER	type mixingPlane;	type mixingPlane;
INLET_DRAFT	type mixingPlane;	type mixingPlane;
OUTLET_DRAFT	type zeroGradient;	type fixedValue; value 0;
WALL_INLET_GUIDE_1	type cyclicGgi;	type cyclicGgi;
WALL_INLET_GUIDE_2	type cyclicGgi;	type cyclicGgi;
WALL_OUTLET_GUIDE_1	type cyclicGgi;	type cyclicGgi;
WALL_OUTLET_GUIDE_2	type cyclicGgi;	type cyclicGgi;
WALL_INLET_RUNNER_1	type cyclicGgi;	type cyclicGgi;
WALL_INLET_RUNNER_2	type cyclicGgi;	type cyclicGgi;
WALL_OUTLET_RUNNER_1	type cyclicGgi;	type cyclicGgi;
WALL_OUTLET_RUNNER_2	type cyclicGgi;	type cyclicGgi;
WALL_GUIDE_1	type slip;	type zeroGradient;
WALL_GUIDE_2	type slip;	type zeroGradient;
WALL_TOP_BOTTOM_GUIDE	type slip;	type zeroGradient;
WALL_PRESSURE	type slip;	type zeroGradient;
WALL_SUCTION	type slip	type zeroGradient;
WALL_TOP_RUNNER	type slip	type zeroGradient;
WALL_BOTTOM_RUNNER	type slip	type zeroGradient;

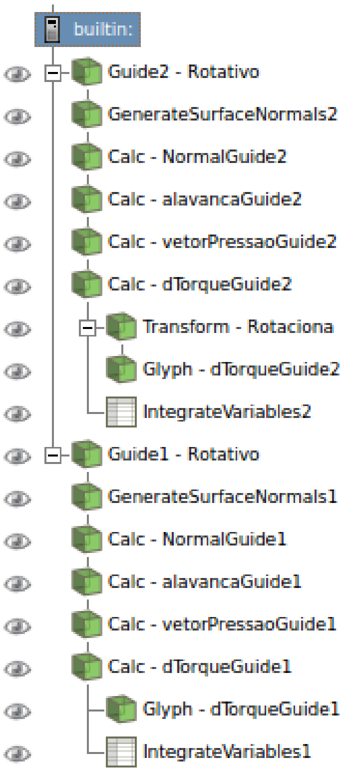

Figure 4: *ParaView* pipeline containing filters used for torque estimation in the surfaces of a guide vane

3 RESULTS

Simulations with isolated and static Tracks were made prior to the one considering the complete rotating system in order to check errors in each mesh.

3.1 Complete rotating channel

The slicing process for the geometries was made in such a way that the sensors were located outside the fluid dominion. Due to shortage of time, the simulation couldn't be executed until its complete convergence, the `MRFSimpleFoam` process was stopped after 49 steps and, instead of rotating the probes on *OpenFOAM* or *ParaView*, the results were compared to the values in a similar work - Lucien Stoessel's master thesis (2014). Table 2 contains information about the last time step and Figs. 5 to 8 illustrate the post-processing operation on *ParaView*. Some vectors on Fig.8 doesn't follow the tendencies of their surroundings due to misdirecting faces on the respective control volume during the meshing process.

Applying the methodology shown on Section 2.6, the following resulting torques were found for each surface:

- T_T (@ `WALL_GUIDE_1`) = $-5,036.11Nm$
- T_T (@ `WALL_GUIDE_2`) = $-8,992.21Nm$

and thus the resulting hydraulic torque for a single guide vane is:

- $T_{T,Resulting} = -14.028,32Nm$

Table 2: Information about last time step on the simulation of the complete rotating channel

Final time step	Ux (Final residual)	Uy (Final residual)	Uz (Final residual)	P (Final residual)
49	1.6938e-10	1.9129e-09	9.7931e-10	3.6096e-05

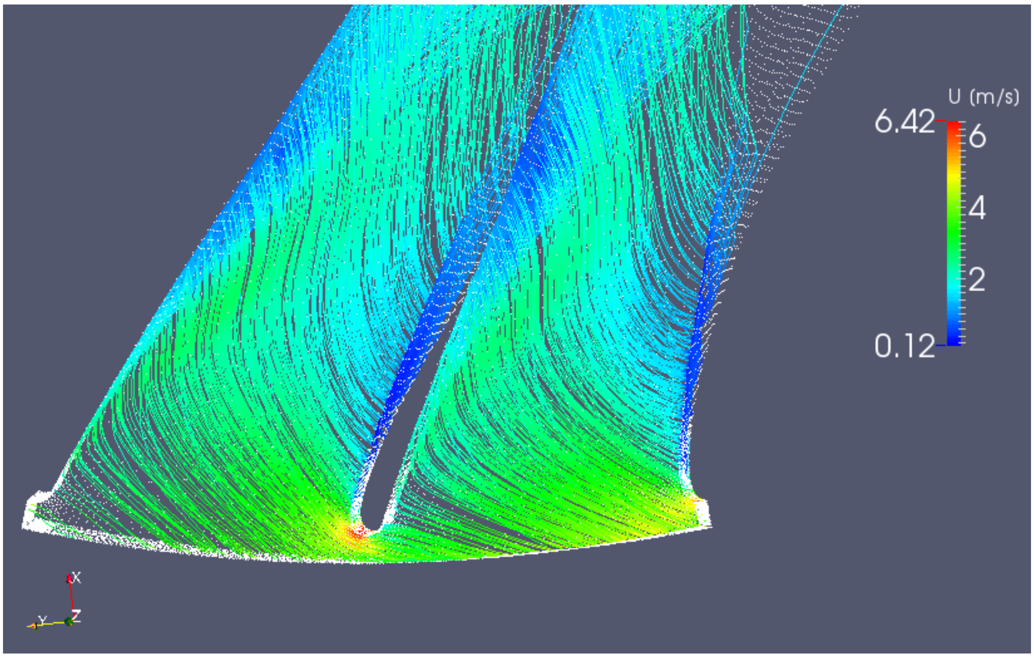

Figure 5: *Stream Tracer* filter applied to the inlet (`INLET_RUNNER`) of Track 2

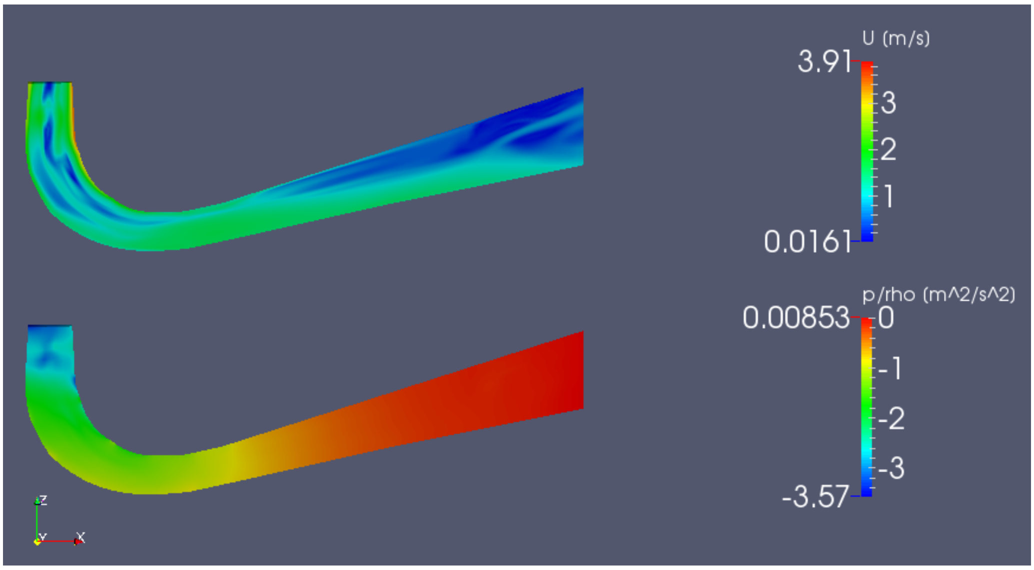

Figure 6: Velocity and pressure fields on Track 3. *Slice* filter used in a plane orthogonal to the Y axis and intersecting point (0 0 0)

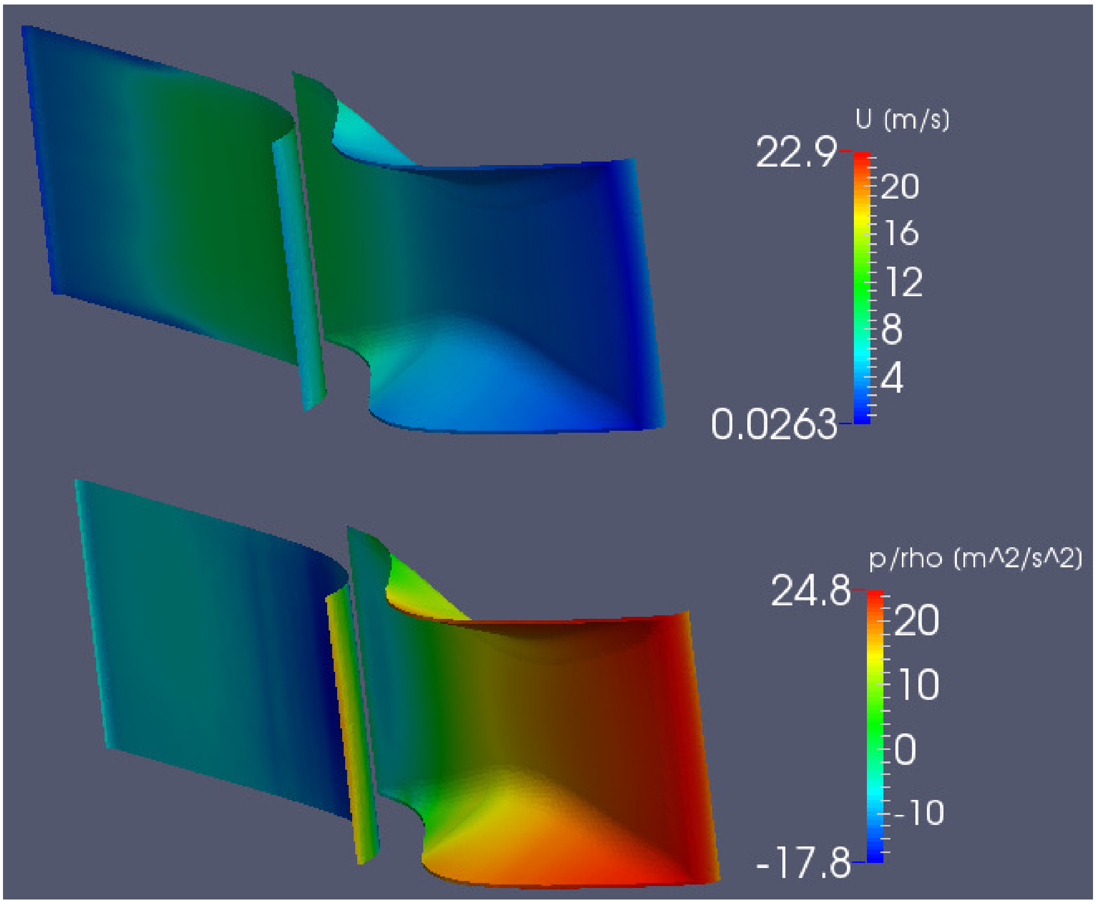

Figure 7: Velocity and pressure fields on the surfaces (`WALL_GUIDE_1` and `WALL_GUIDE_2`) of a guide vane

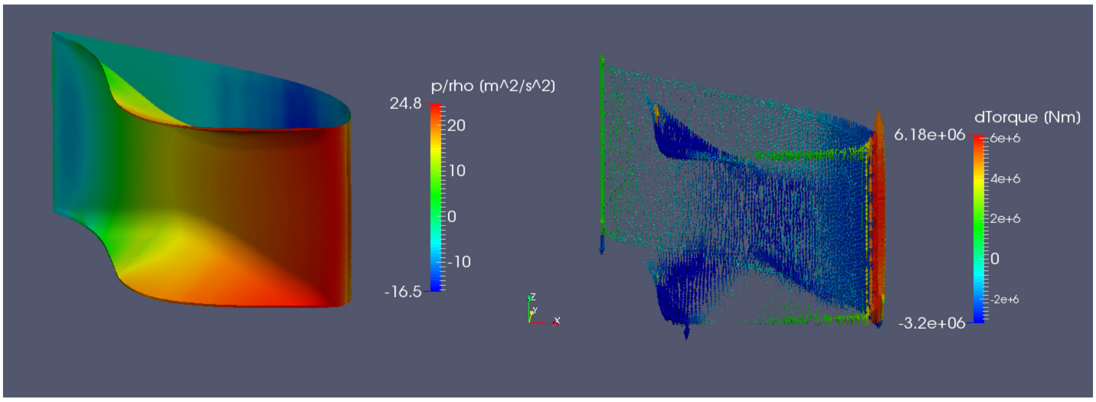

Figure 8: Distribuition of `dTorque` vectors on a complete guide vane under steady-state regime

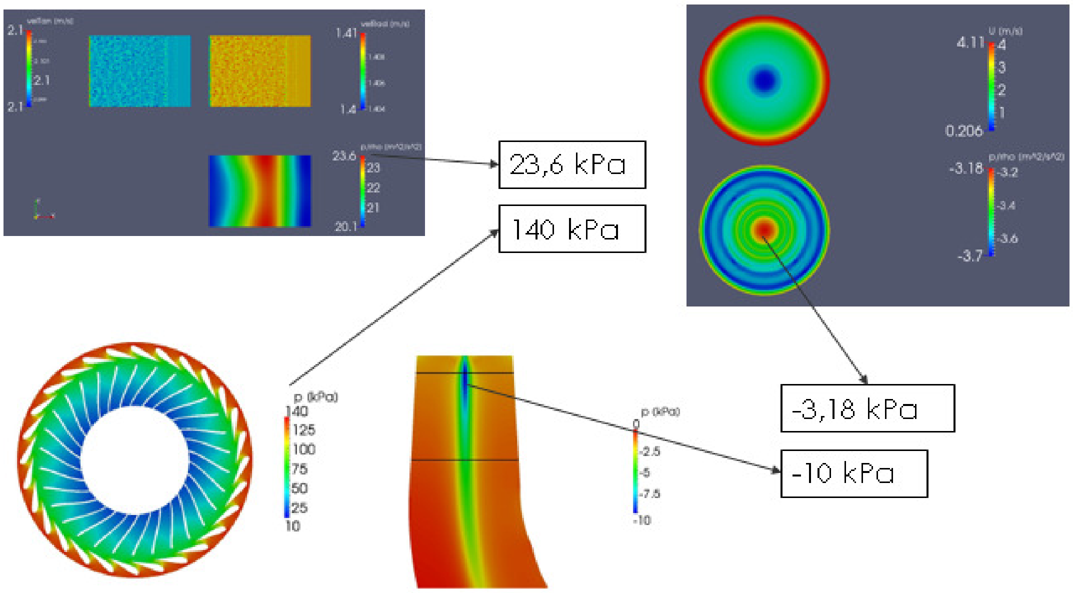

Figure 9: Comparison between results from this research and the ones presented on Lucien Stoessels' masters thesis (2014)

3.2 Conclusion

Values obtained on the simulation diverged from the ones used as reference but have similar order of magnitude and exhibited physical tendencies as expected due to coupling between velocity and pressure. However, in the present state, the results aren't good enough to be used as a trustful and safe prediction to hydrodynamic torque.

The difference between values was influenced by problems on meshing quality, usage of laminar flow model, slight difference in the velocity on inlet adopted by the reference and the time of simulation.

Results could be used as initial boundary fields for more complex and robust simulations in order to achieve faster convergence.

4 Future improvements

In order to generate safer and more stable results improvements for this methodology are suggested as follows:

- Better mesh quality:
  - Track 1 has very skewed parts in its inlet and outlet surroundings, more effort on reducing its skewness is required;
  - Use structured mesh;
- Use parallel processing:
  - Speed up convergence and results checking;
- Use a proper turbulence model:

  - Stop using laminar model for the flow;
  - Test convergence using different models;
- Collect data on the fly, in each time step:
  - Use probe function and gnuplot for plotting of real time curves.

The used methodology is based on significant simplifications in such a manner that even if the mentioned improvements are applied the results may present limited usability. For a more robust simulation, with results compatible to the unsteady regime intrinsical to Francis turbines, the following improvements are suggested:

- Avoid using *Mixing Plane*:
  - It doesn't consider the circumferential variation on properties;
  - Not compatible to the analysis of *Blade Pass Frequency*;
- Consider possibility of cavitation:
  - Use the `interFoam` solver for multi-phase flow;
- Less simplifications:
  - Do not consider an axi-periodic approach but analyse the whole geometry;
  - Avoid using MRF but effectively rotate the grid while using GGI to make an interface.

ACKNOWLEDGEMENTS

This research was made as part of my final assessment for the degree on Bachelor of Mechanical Engineering in University of Brasilia, Brazil, in 2016. It was supported by ELETROBRAS - ELETRONORTE during my internship in the Engineering Management for Electromechanical Power Generation (EEGE) from 2015 to 2016.

REFERENCES

Agência Nacional de Energia Elétrica (ANEEL), 2015, November. *Banco de Informações de Geração - Capacidade de Geração do Brasil*. Retrieved from
http://www.aneel.gov.br/aplicacoes/capacidadebrasil/capacidadebrasil.cfm

Beaudoin, M. & Jasak, H., 2008. *Development of a Generalized Grid Interface for Turbomachinery simulations with OpenFOAM*. Open Source CFD International Conference.

Beaudoin, M., Nilsson, H., Page, M., Magnan, R. & Jasak, H., 2014. *Evaluation of an improved mixing plane interface for OpenFOAM*. 27th IAHR Symposium on Hydraulic Machinery and Systems (IAHR 2014

Norwegian Hydropower Centre. (n.d.) *Francis-99*. Retrieved from
https://www.ntnu.edu/nvks/francis-99

OpenFOAM Wiki. (n.d.) *Contrib/groovyBC*. Retrieved from
https://openfoamwiki.net/index.php/Contrib/groovyBC

OpenFOAM Wiki. (n.d.) *See the MRF development*. Retrieved from http://openfoamwiki.net/index.php/See_the_MRF_development

OpenFOAM User Guide. (n.d.) *3.6 Standard utilities*. Retrieved from http://cfd.direct/openfoam/user-guide/standard-utilities/

OpenFOAM User Guide. (n.d.) *3.5 Standard solvers*. Retrieved from http://cfd.direct/openfoam/user-guide/standard-solvers/

OpenFOAM User Guide. (n.d.) *5.2 Boundaries*. Retrieved from http://cfd.direct/openfoam/user-guide/boundaries/

Stoessel, L., 2014. *Numerical simulations of the flow in the Francis-99 turbine*. Master thesis, Chalmers University of Technology.