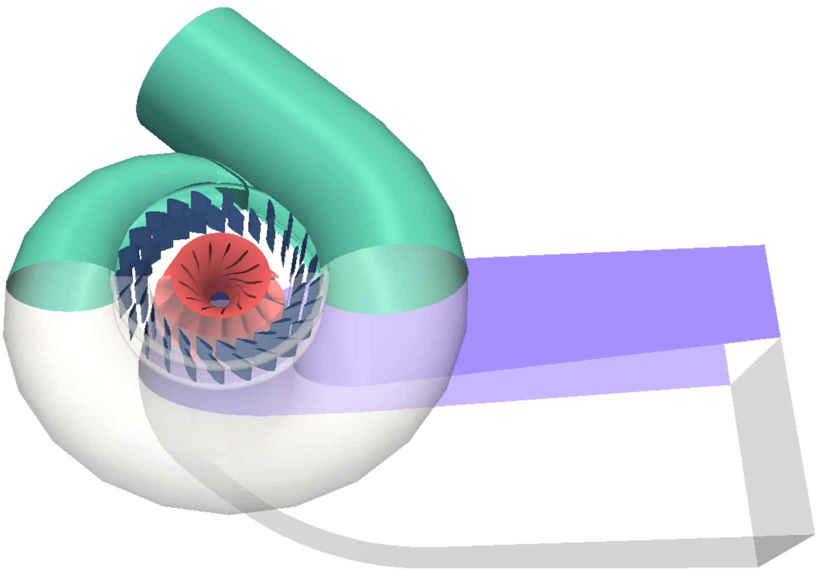
CFD support s.r.o.
Sokolovská 270/201, 19000 Praha 9, Czech Republic
info@cfdsupport.com | Tel: +420 212 243 883 | www.cfdsupport.com

TCFD 17.06 parallel scaling Francis Hydro-Turbine

Summary of current results



June, 2017

Contents

Introduction

The aim of this work was to evaluate the parallel performance of the TCFD® 17.06 (Turbomachinery CFD) software produced by the company CFD Support.

TCFD® is a comprehensive CFD workflow for turbomachinery simulations. This workflow covers complete process from the basic (usually CAD) data over CFD analysis to significant engineering results.

TCFD® is based on the OpenFOAM® software. It is the final outcome of a many year development of the team of CFD Support engineers and developers. TCFD® is not dependent on other software but it is fully compatible with standard OpenFOAM® and other software packages. It was originally designed for simulating rotational machines, nevertheless it can be used for a wide range of various CFD simulations.

- Pumps
- Fans
- Compressors
- Turbines
- Hydro Turbines
- Turbochargers
- Nozzles & Diffusers
- Steam Turbines
- Both axial and radial machines
- Both compressible and incompressible flows
- Ventilators
- Engine flows

Francis Hydro-Turbine CFD Model

The geometry of the Francis Hydro-Turbine model is presented in the figures 1, 2 and 3. The model consists of the inlet spiral casing followed by 24 stay vanes and the same amount of the guides vanes which route the water into the runner with 13 blades. And the turbine model ends with the draft-tube.
The operating conditions for this model were following:

Volumetric flow-rate: $6.75\ m^3 \cdot s^{-1}$
RPM of the runner: $600\ min^{-1}$
Fluid type: *Water*

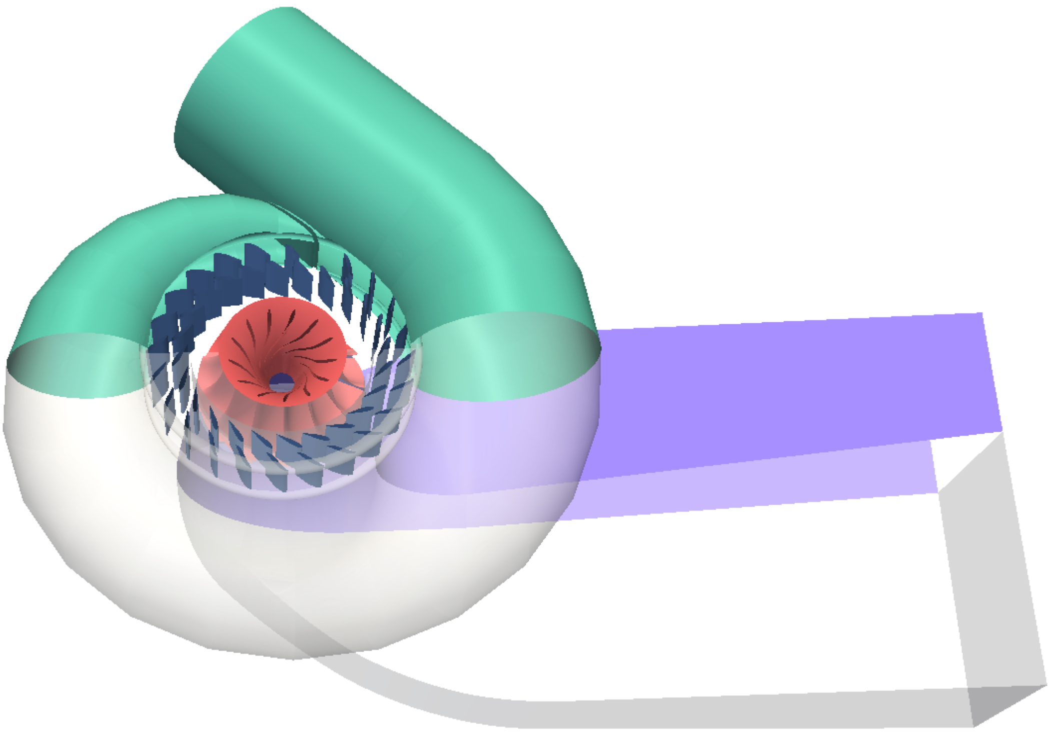


Figure 1: Geometry the Francis Hydro-Turbine

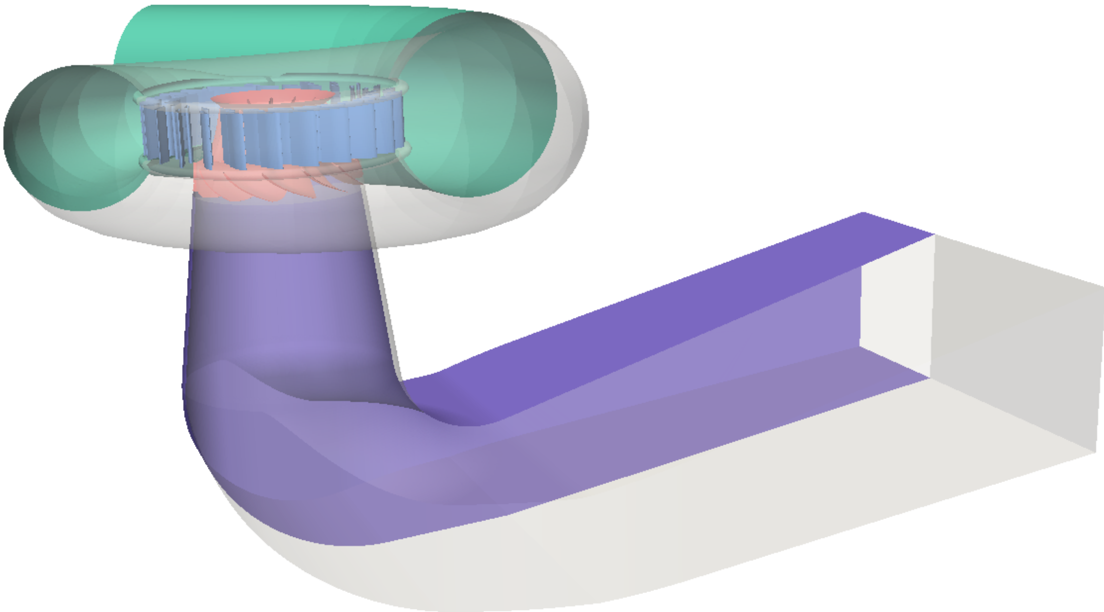


Figure 2: Geometry the Francis Hydro-Turbine

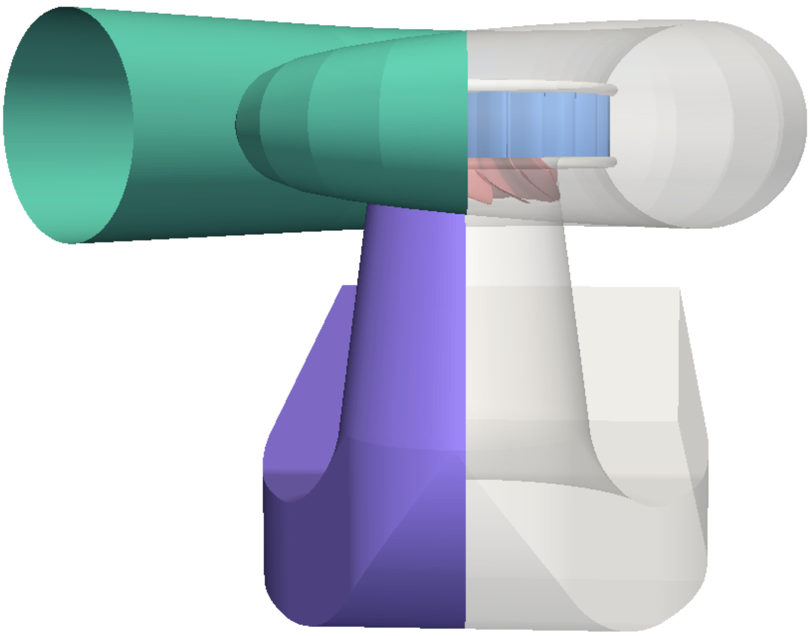


Figure 3: Geometry the Francis Hydro-Turbine

Computational Mesh

The computational mesh consists of 5 components in this case - inlet spiral casing, stay vanes, guide vanes, a runner and a draft-tube - see the figure 4.

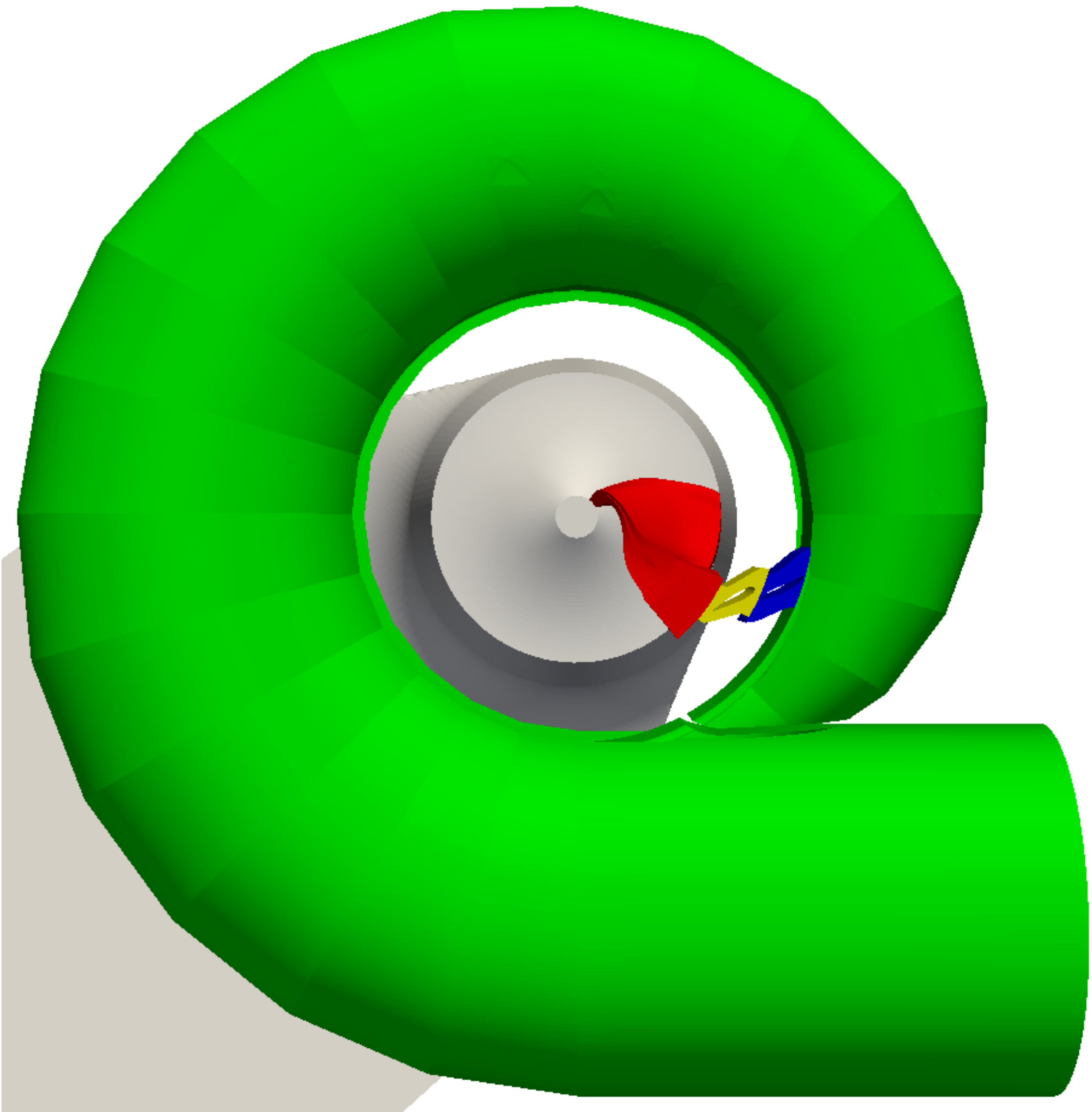


Figure 4: Components of the computational mesh of the Francis Hydro-Turbine

There were considered three variants of the computational mesh for the parallel scaling test. They differ only in the mesh size and their topologies were completely the same. One can find their cells count in the following table 1:

component	Mesh 1 # of cells	Mesh 2 # of cells	Mesh 3 # of cells
Spiral	689 004	2 219 516	9 950 188
Stay	71 808	574 464	4 595 712
Guide	70 400	563 200	4 505 600
Runner	168 960	1 351 680	10 813 440
Draft-tube	971 776	7 774 208	7 774 208
Totals	**2.0 million**	**12.5 million**	**37.6 million**

Table 1: Mesh size table

The inlet spiral casing was meshed by the automatic mesh generator snappyHexMesh, which is a part of the OpenFOAM Toolbox. It generates so called hexa-dominant unstructured meshes and it is very efficient in meshing of complex geometries.

The rotationally symmetrical periodic segments were extracted from the stay, the guide and the runner - there is considered just one blade with periodic boundary conditions on its sides. All of these periodic segments were meshed by the block-structured hexahedral mesh.
Mesh 1 - stay, guide and runner topology - see figures 5, 6.
Mesh 2 - stay, guide and runner topology - see figures 7, 8.
Mesh 3 - stay, guide and runner topology - see figures 9, 10.

And the last component - the draft-tube - was meshed also with the block-structured hexahedral mesh.

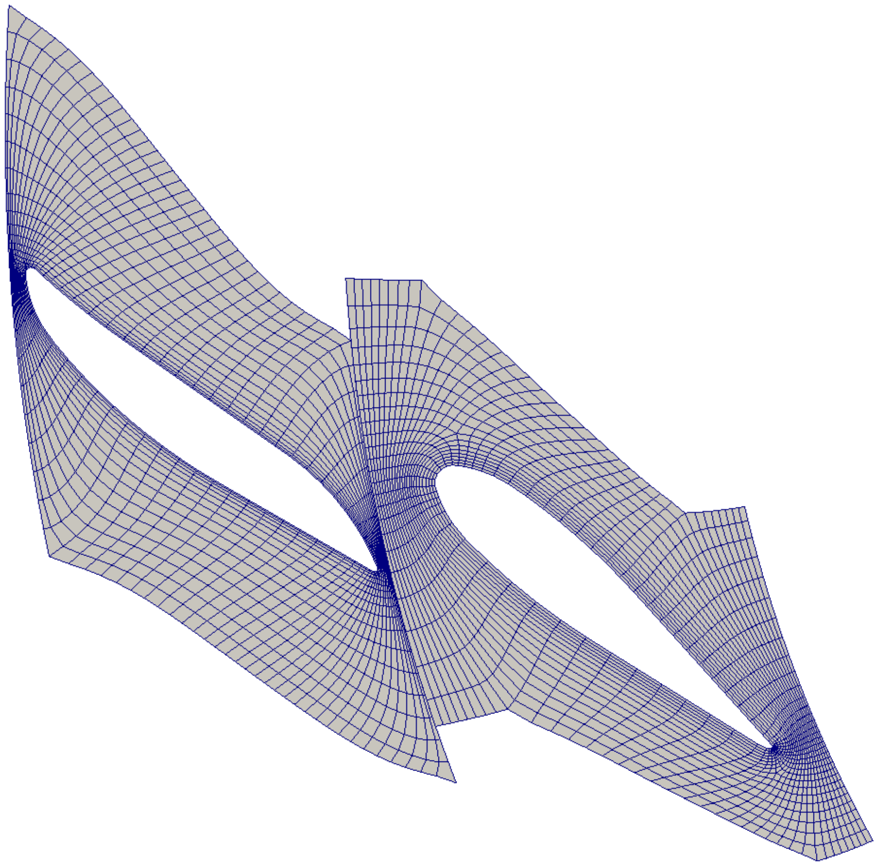


Figure 5: Mesh 1 of the Francis Hydro-Turbine stay and guide vanes - 2 million cells mesh

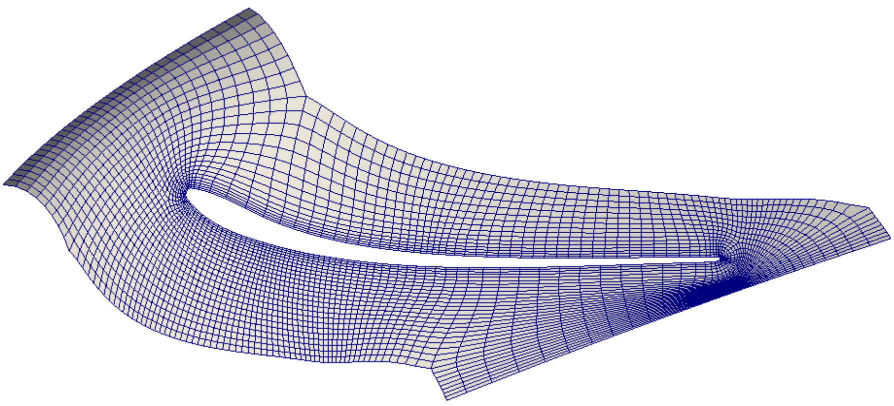


Figure 6: Mesh 1 of the Francis Hydro-Turbine runner blade - 2 million cells mesh

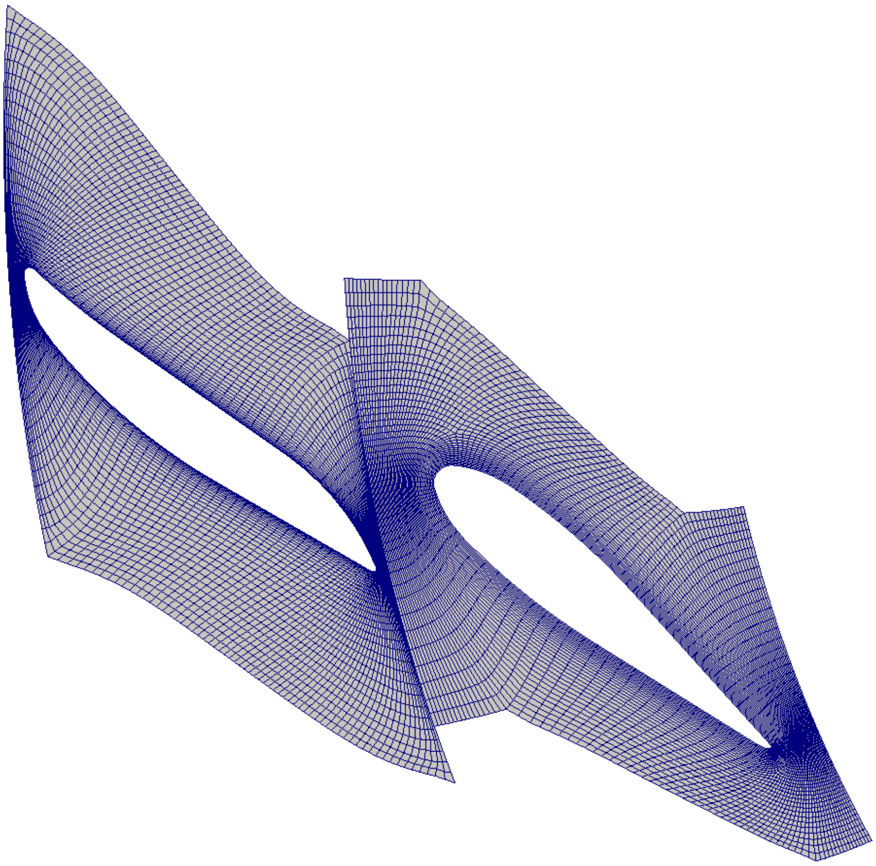


Figure 7: Mesh 2 of the Francis Hydro-Turbine stay and guide vanes - 12.5 million cells mesh

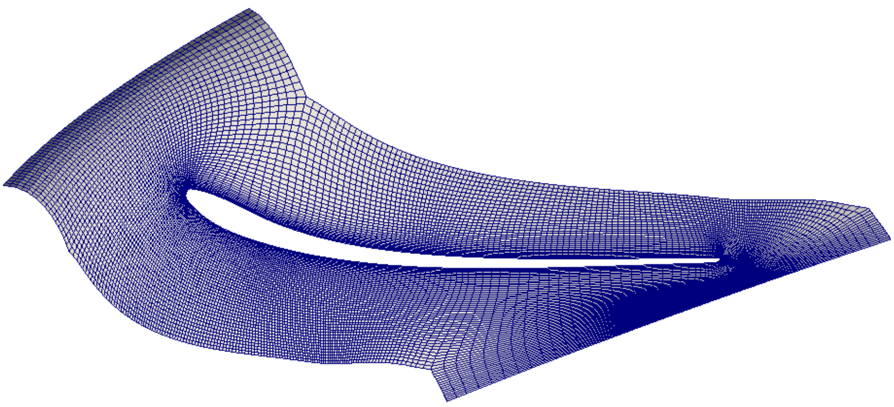


Figure 8: Mesh 2 of the Francis Hydro-Turbine runner blade - 12.5 million cells mesh

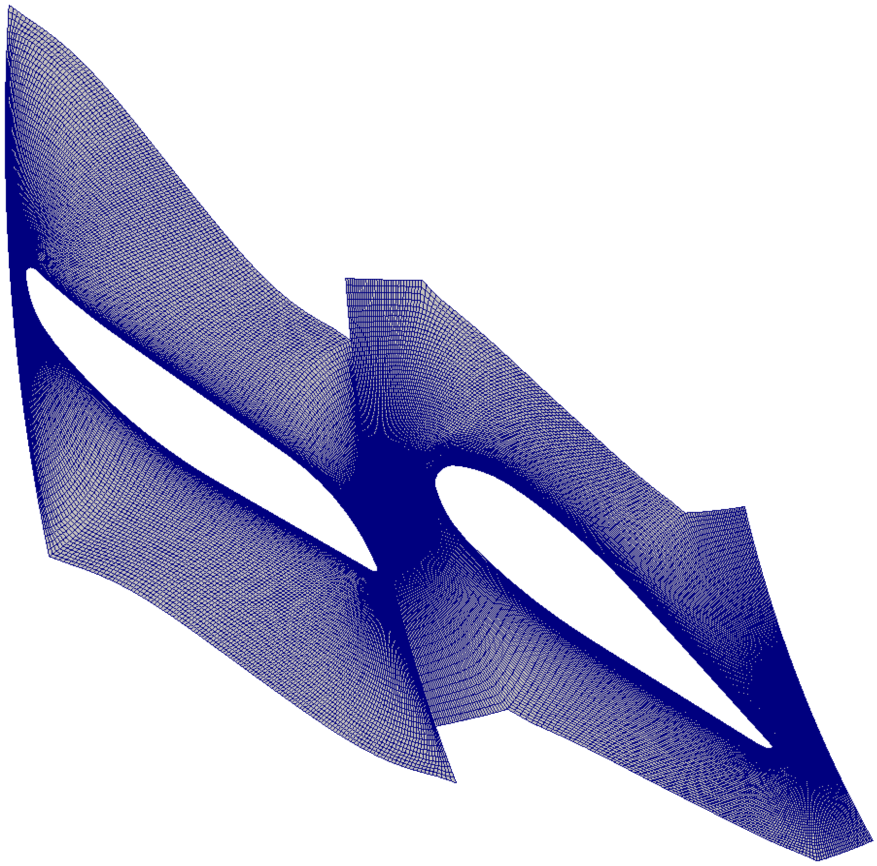


Figure 9: Mesh 3 of the Francis Hydro-Turbine stay and guide vanes - 37.6 million cells mesh

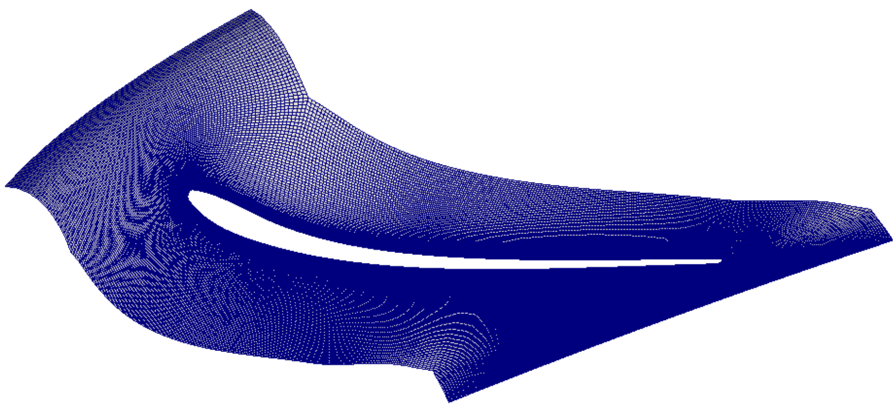


Figure 10: Mesh 3 of the Francis Hydro-Turbine runner blade - 37.6 million cells mesh

Case Setup

The steady state incompressible turbulent fluid flow including MRF approach for rotation modeling was considered in this case.

The fluid was modeled as an incompressible water with the reference density $\rho_{ref} = 998\ kg/m^3$ and the kinematic viscosity $\nu = 1.002 \cdot 10^{-6}\ m^2/s$.

The general case setup is summarized in the following table 2:

Time marching:	steady state
Fluid type:	incompressible (water)
Turbulence:	RANS - k-ω SST
Runner rotation:	MRF (Multiple Reference Frame) method - 600 RPM
Volumetric flow-rate:	$Q = 6.75\ m^3/s$
Reference density:	$\rho_{ref} = 998\ kg/m^3$
Kinematic viscosity:	$\nu = 1.002 \cdot 10^{-6}\ m^2/s$

Table 2: General case setup table

Types of Boundary Conditions

The boundary conditions setup was done as usual for such a case.

There was set the *volumetric flow-rate* at the inlet of the spiral casing with velocity vectors perpendicular to the inlet boundary (patch). The resulting velocity magnitude was uniformly distributed across the inlet boundary (patch).

There were set the *mixing-plane* (mixingInterface - developed uniquely for TCFD®) boundary conditions at every interface - i.e. at the spiral-stay interface, at the stay-guide interface, at the guide-runner interface and also at the runner-draft-tube interface.

All periodic segments (stay, guide, runner) had been set the cyclicAMI on their periodic (cyclic) boundaries.

All wall boundaries (patches) had been set so called no-slip boundary conditions, i.e. all components of velocity vector was set to 0.

A fixed kinematic static pressure was set at the outlet from the draft-tube - $p_{s_{out}} = 0$.

The boundary conditions setup is summarized in the following table 3:

Inlet:	volumetric flow-rate $Q = 6.75\ m^3/s$
Walls:	no-slip $U = (0, 0, 0)$
Periodic boundaries:	cycliAMI - rotational
Interfaces (between components):	mixingInterface - *mixing-plane*
Outlet:	fixed kinematic pressure $p_{s_{out}} = 0$

Table 3: Boundary conditions setup table

Turbulence Modeling

The steady state Reynolds Averaged Navier-Stokes approach completed by k-ω SST turbulence model was used to model the turbulence.

Turbulence intensity was set to 5% at the inlet to the spiral casing and a fixed value of $\omega = 100\ s^{-1}$ at the inlet.

Standard wall-functions were used on boundaries of type wall.

Postprocessing

The Efficiency function Object was used for postprocessing. The Efficiency function Object is a unique part of TCFD® which does the postprocessing of integral variables on selected boundaries - e.g. efficiency, total pressure drop, runner torque, flow-rate, static pressure per interfaces, total pressure per interfaces and many others.

The efficiency was examined in this case, whether it is not changing with the changing number of parallel processes.

Hardware Overview

TCFD® 17.06 parallel scaling was tested on 1, 2, 3 and 4 nodes of following hardware (see table 4):

Mother board:	SUPERMICRO X10DRT (dual socket)
CPU:	2x Intel Xeon E5-2680v3 at 2.5 GHz (12 COREs each)
Memory:	8x 16GB DDR4 2133 MHz (128 GB in total)
Interconnection (between nodes):	Mellanox Infiniband FDR 56 Gbps

Table 4: Boundary conditions setup table

One can see the server chasis in the figure 11 - two nodes from table 4 are in 1U server. There were two such servers available for parallel scaling test - 96 COREs in total.

Figure 11: SUPERMICRO SuperServer 6018TR-TF

Results

The parallel efficiency and the parallel speedup were evaluated for the 12 simulations in total. There were running simulations with the 3 different mesh sizes (see the table 1 for more information) on 1, 2, 3 and 4 nodes of the server (see the table 4 for information about the hardware configuration).

The simulation run-time was measured and the run-times from all the simulation are presented in the following table 5

# of Cores	**Parallel Efficiency on different meshes**		
	2 Mil. cells	12.5 Mil. cells	37.6 Mil. cells
24 (1 node)	2385 s	93930 s	307295 s
48 (2 nodes)	1205 s	46165 s	156264 s
72 (3 nodes)	858 s	30994 s	104814 s
96 (4 nodes)	648 s	22894 s	81251 s

Table 5: Run-times of all performed simulations

Parallel Efficiency

The parallel efficiency η is defined by the following formula:

$$\eta = \frac{t_1}{n \cdot t_n} \tag{1}$$

where t_1 is the run-time of the simulation which run on one node of the server and t_n is the run-time of the simulation which run on the n nodes of the server. The "ideal" scaling is when the efficiency remains 100% with increasing number of parallel processes.

See the results of the parallel efficiency for all mesh sizes in the table 6 and see the plot of parallel efficiency in the figure 12.

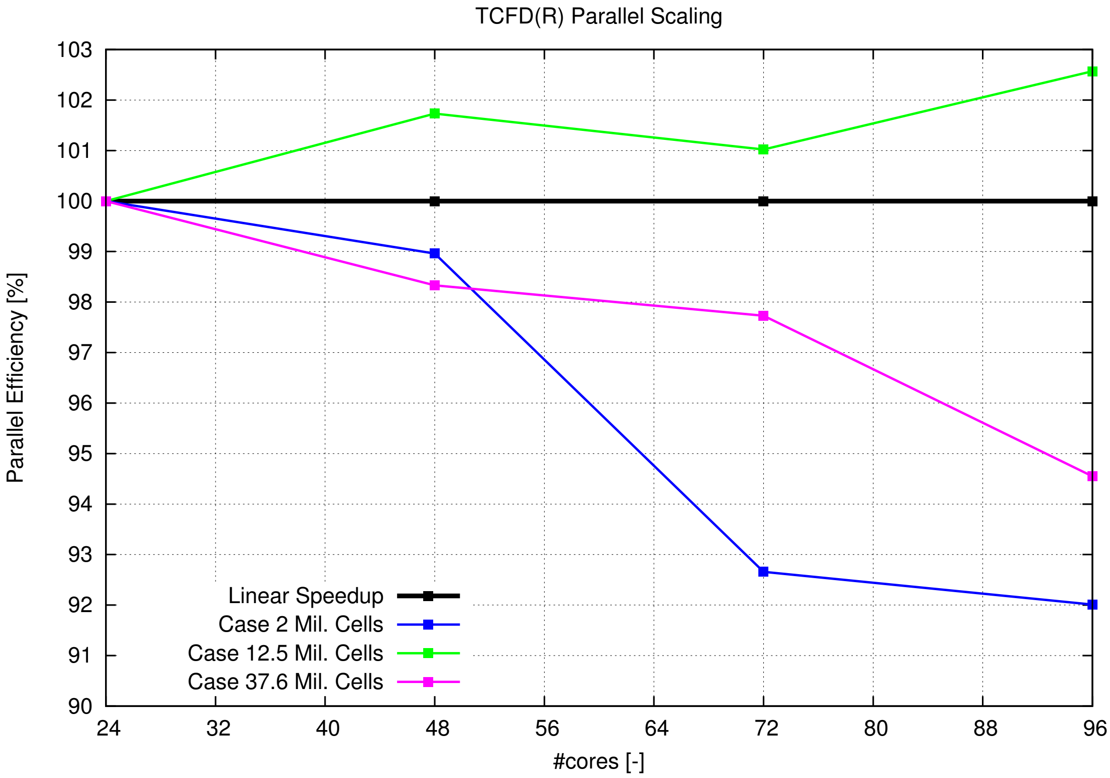


Figure 12: Parallel efficiency depending on # of cores

# of Cores	**Parallel Efficiency on different meshes**			
	2 Mil. cells	12.5 Mil. cells	37.6 Mil. cells	Theory (ideal)
24 (1 node)	100.00%	100.00%	100.00%	100.00%
48 (2 nodes)	98.96%	101.73%	98.33%	100.00%
72 (3 nodes)	92.66%	101.02%	97.73%	100.00%
96 (4 nodes)	92.01%	102.57%	94.55%	100.00%

Table 6: Parallel efficiency depending on # of cores

Parallel Speedup

The parallel speedup S is defined by the following formula:

$$S = \frac{t_1}{t_n} \tag{2}$$

where t_1 is the run-time of the simulation which run on one node of the server and t_n is the run-time of the simulation which run on the n nodes of the server. The "ideal" scaling is when the speedup is equal n.

See the results of the parallel speedup for all mesh sizes in the table 7 and see the plot of parallel speedup in the figure 13.

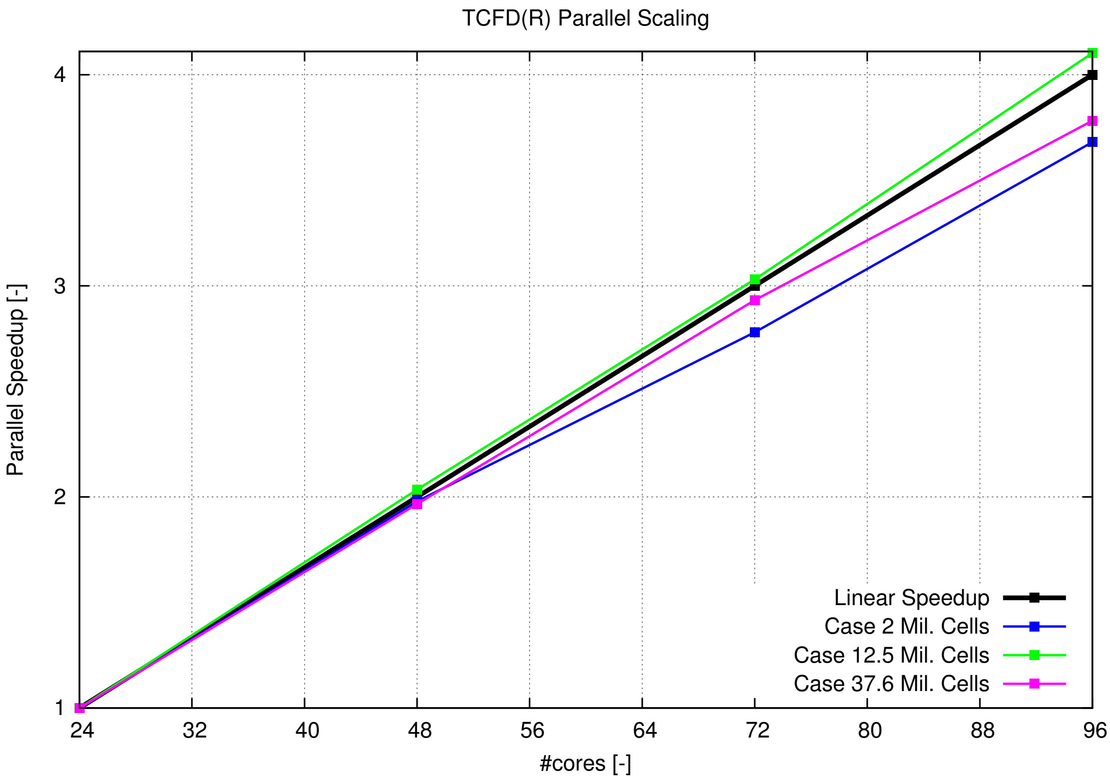


Figure 13: Parallel speedup depending on # of cores

# of Cores	Parallel Speedup on different meshes			
	2 Mil. cells	12.5 Mil. cells	37.6 Mil. cells	Theory (ideal)
24 (1 node)	1.000	1.000	1.000	1.000
48 (2 nodes)	1.979	2.035	1.967	2.000
72 (3 nodes)	2.780	3.031	2.932	3.000
96 (4 nodes)	3.681	4.103	3.782	4.000

Table 7: Parallel speedup depending on # of cores

CFD Results Comparison

The results of CFD simulations are consistent. They do not depend on number of nodes (CPUs, Cores) used for the parallel calculation.

The results depend just on the mesh size and its quality.

See table 8 - the comparison of the Francis Hydro-Turbine efficiency and see table 9 - the comparison of the Francis Hydro-Turbine head (Hydro-Turbine head is proportional to the total pressure difference between the inlet and outlet of the machine).

# of Cores	Hydro-Turbine Efficiency comparison		
	2 Mil. cells	12.5 Mil. cells	37.6 Mil. cells
24 (1 node)	92.44%	92.67%	92.77%
48 (2 nodes)	92.44%	92.67%	92.77%
72 (3 nodes)	92.44%	92.66%	92.77%
96 (4 nodes)	92.44%	92.67%	92.78%

Table 8: Hydro-Turbine Efficiency depending on # of cores

# of Cores	Hydro-Turbine Head comparison		
	2 Mil. cells	12.5 Mil. cells	37.6 Mil. cells
24 (1 node)	44.20 m	43.55 m	43.62 m
48 (2 nodes)	44.19 m	43.55 m	43.62 m
72 (3 nodes)	44.19 m	43.56 m	43.63 m
96 (4 nodes)	44.20 m	43.55 m	43.62 m

Table 9: Hydro-Turbine Head depending on # of cores

Conclusion

As one can see in the results (see table 6 and 7 and figure 12 and 13) it is evident that TCFD® 17.06 scales very well. The worst observed parallel efficiency presented in this report is about 92% for the smallest case (smallest in sense of mesh size) which is still very high number for such a small case. All other results are even better. Moreover based on these results it can be very reliably estimated that there can be achieved the parallel efficiency higher than 90% with doubling or tripling number of nodes (or cores - parallel processes) for Mesh 2 - 12.5 million cells and for Mesh 3 - 37.6 million cells.

There are two regimes in the parallel computing, could be said - one is the single process computation (independent of the rest of processes) and the second one is the processes' synchronization. The case with higher number of cells has much lower ratio of time needed for process synchronization over the time needed for single process computation. So that's why typically applies - the more cells in the mesh the higher parallel efficiency or parallel speedup, which one can observe more or less also in this case.

The results of the smallest mesh and the biggest mesh are quite consistent - the higher number of cell in the mesh the better parallel efficiency.

There are also cases where the opposite behavior is observed (so called super-linear speedup) - mainly on the medium size mesh (Mesh 2 - 12.5 million). This phenomenon can occur due to the efficient CPU's cache usage and it is strongly dependent on the parallel decomposition of the mesh and the specific CPU - so i.e. such behavior is possible to achieve but it strongly depends on the specific CFD case and the specific hardware.

The results of CFD simulations are consistent. They do not depend on number of nodes (CPUs, Cores) used for the parallel calculation. The results depend just on the mesh size and its quality. See table 8 - the comparison of the Francis Hydro-Turbine efficiency and see table 9 - the comparison of the Francis Hydro-Turbine head (Hydro-Turbine head is proportional to the total pressure difference between the inlet and outlet of the machine).