

# Some Experiences on the Accuracy and Parallel Performance of OpenFOAM for CFD in Water Turbines

Håkan Nilsson

Department of Applied Mechanics, Div. of Fluid Dynamics,  
Chalmers, SE-412 96 Gothenburg, Sweden

`hani@chalmers.se`

<http://www.tfd.chalmers.se/~hani>

**Abstract.** 50% of the electric power in Sweden is generated by water power. Many of the power plants in Sweden are getting old and some major refurbishments are coming up. Due to the development of numerical methods and computer power the last decades Computational Fluid Dynamics (CFD) is to a large extent used as a design tool for this purpose. The general features of the flow in water turbines can be resolved with today's methods and computational power, but in order to study the flow in detail enormous HPC (High Performance Computing) facilities and new methods are required.

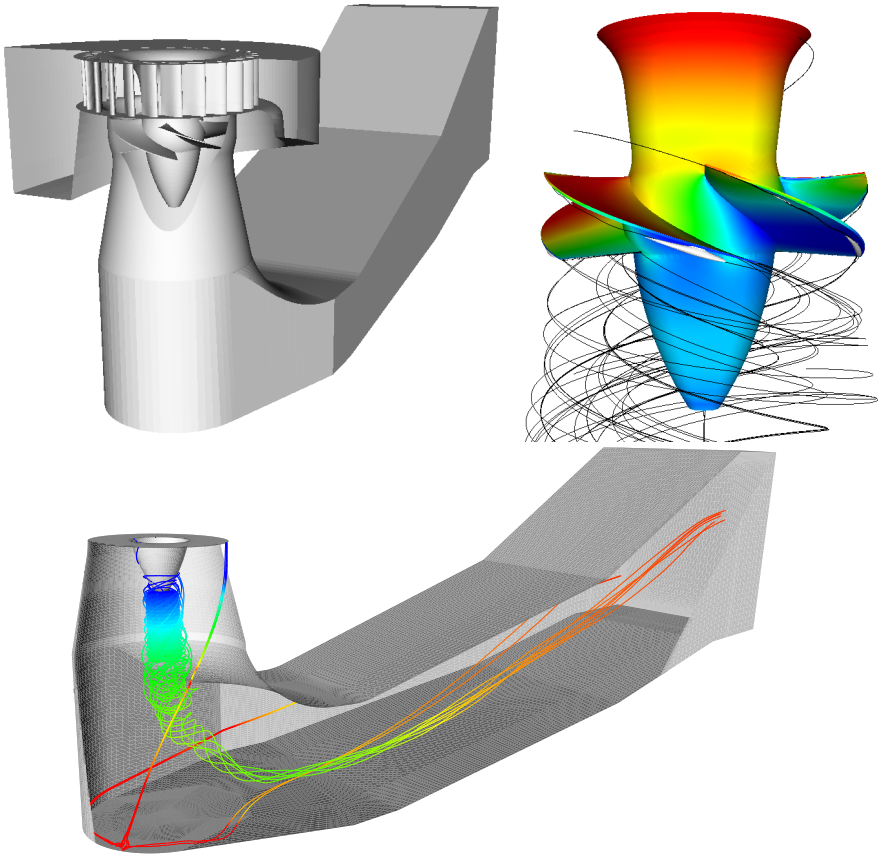
The present work presents the water turbine field with its HPC requirements, shows some state-of-the-art results from OpenFOAM CFD analysis, and presents a parallel performance analysis on a Linux cluster using an ordinary gigabit interconnect v.s. an Infiniband interconnect.

## 1 Water Turbines

Water turbines are designed to extract energy from the water flowing through the water turbine runner. The available power is determined by the difference in the elevation of the tail water and head water multiplied with the water mass flow and gravity. In reaction turbines the flow enters the runner with a swirl and the runner is designed to remove that swirl before the water enters the draft tube (see Figure 1). The draft tube is a diffuser which recovers the static pressure and leads the water towards the tail water.

### 1.1 HPC Requirements

Water turbines have complicated geometries where the flow in different parts of the turbine influences the flow in other parts of the turbine. To be able to set valid boundary conditions and to model the flow correctly it would thus be ideal to include all the geometry from head water to tail water. The Reynolds numbers in water turbine applications are always high, so the resolution of the computational mesh must be fine where large gradients in the flow occur. These



**Fig. 1.** CAD model of the Hölleforsen Kaplan turbine model and visualizations of the flow in the runner and draft tube

requirements together yields an enormously large mesh. In addition to this there are both stationary and rotating parts in water turbines, which require rotor-stator interaction and unsteady computations. The thin wakes after stayvanes, guide vanes and runner blades affect the details of the flow and should also be resolved. Close to the runner blades cavitation often occurs, and the numerical methods to resolve those effects require even more cells and shorter time steps. There is probably no limit on the HPC requirements for flow in water turbines if all the important flow features should be appropriately predicted using CFD.

## 2 The OpenFOAM CFD Tool

The newly released OpenFOAM (Open Field Operation and Manipulation, [www.openfoam.org](http://www.openfoam.org)) tool has been used in the present study. OpenFOAM is an Open-Source object oriented C++ tool for solving various partial differential equations

(PDEs). It includes preprocessing (grid generator, converters, manipulators, case setup), postprocessing (using OpenSource Paraview) and many specialized CFD solvers are implemented. Some of the more specialized features that are included in OpenFOAM, and which are important to the flow in water turbines are: sliding grid, moving meshes, two-phase flow (Langrange, VOF, Euler-Euler) and fluid-structure interaction. OpenFOAM runs in parallel using automatic/manual domain decomposition. In addition to the source code, OpenFOAM gives access to an international community of OpenFOAM researchers through the discussion board at the OpenFOAM home page.

### 3 The Studied Cases

In the present work two parts of the Hölleforsen turbine model have been modeled, the runner and the draft tube (see Figure 1). The two parts have been modeled separately, but the aim is to simultaneously model all parts including the spiral casing, the runner and the draft tube. The present runner computations have an inlet boundary condition from a previous computation of the flow through the guide vanes. Steady computations have been made for both a periodic part of the runner (1/5) and the whole runner. The present draft tube computations have inlet boundary conditions from detailed Laser-Doppler Anemometry (LDA) measurements. Both steady [1] and unsteady computations [2] have been made for the draft tube. In all cases the standard  $k - \epsilon$  turbulence model is used. The sizes of the meshes used are, 450 000 cells for the periodic runner computation, 2 240 000 cells for the full runner computation, and 1 000 000 cells for the draft tube computations. These are all wall-function grids that do not resolve the boundary layers in detail.

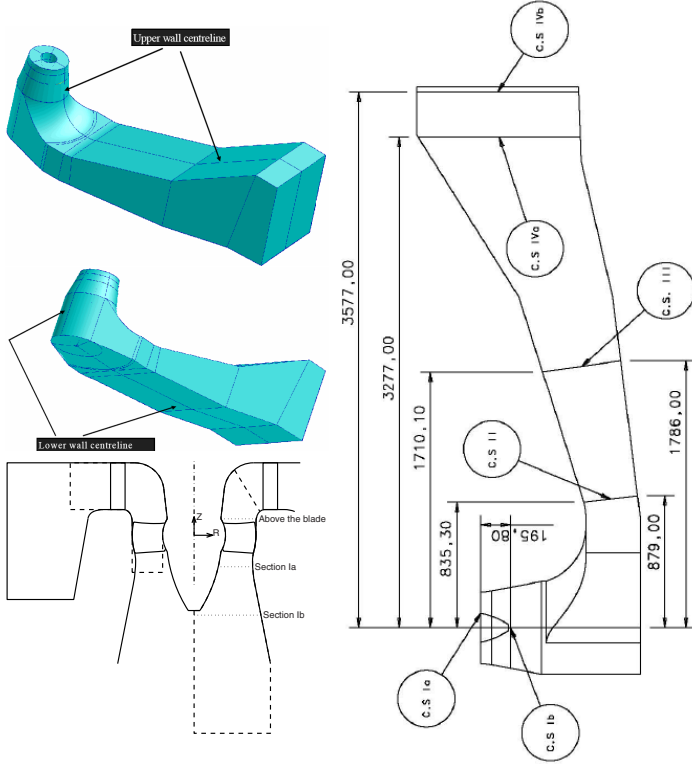
### 4 Results and Validation

The results from the present computations have been presented and validated in detail in [2]. Some of these results and validations against measurements and CFX-5 results are presented here also. The locations of the available measurements [3] of the flow in the present cases are shown in Figure 2, and the names of the measurement sections are used when presenting the results.

The dimensionless coefficients used in the validation are the velocity coefficient ( $C_v$ ), the pressure coefficient ( $C_p$ ), the mean pressure recovery ( $C_{prm}$ ) and the energy loss coefficient ( $\zeta$ ).

The velocity coefficients,  $C_v$ , are the velocities normalized by  $Q/A_i$ , where  $Q = 0.522\text{m}^3/\text{s}$  is the volume flow and  $A_i$  is the area of each cross-section ( $A_{Ia} = 0.15\text{m}^2$ ,  $A_{Ib} = 0.23\text{m}^2$ ). The tangential velocity component is positive when the flow is co-rotating with the runner, and the axial velocity component is positive in the main flow direction.

The pressure coefficient,  $C_p$ , is the local static pressure divided by the dynamic pressure at section Ia,  $P_{dyn,Ia} = \rho Q^2 / (2A_{Ia}^2)$ .



**Fig. 2.** Geometry and measurement sections. (Pictures from the Turbine-99 workshop guidelines and from previous work).

The mean pressure recovery is defined as

$$C_{prm} = \frac{\frac{1}{A_{CS}} \int \int_{A_{CS}} P dA - \frac{1}{A_{Ia}} \int \int_{A_{Ia}} P dA}{P_{dyn, Ia}}, \quad (1)$$

where  $P_{dyn, Ia} = \rho Q^2 / (2A_{Ia}^2)$  is the dynamic pressure at cross-section Ia.

The energy loss coefficient is defined as

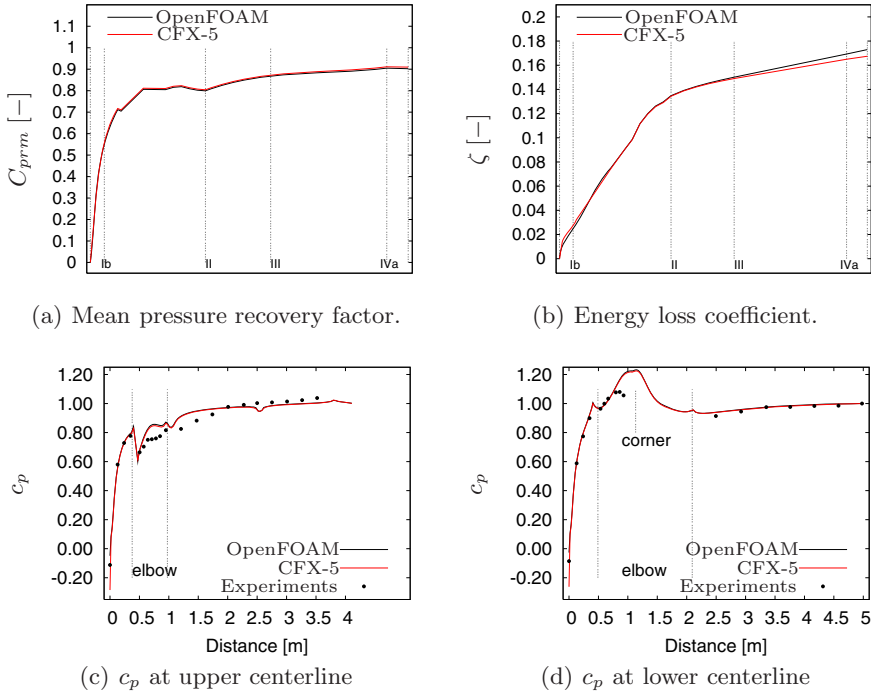
$$\zeta = \frac{\int \int_{A_{Ia}} \left( P + \rho \frac{|U|^2}{2} \right) \mathbf{U} \cdot \hat{\mathbf{n}} dA - \int \int_{A_{CS}} \left( P + \rho \frac{|U|^2}{2} \right) \mathbf{U} \cdot \hat{\mathbf{n}} dA}{\left| \int \int_{A_{Ia}} \rho \frac{|U|^2}{2} \mathbf{U} \cdot \hat{\mathbf{n}} dA \right|}. \quad (2)$$

#### 4.1 Integral Quantities

The integral quantities described in eqs. (1) and (2) are here used to validate the steady draft tube computation. Figure 3 shows through-flow developments of the mean pressure recovery factor and energy loss coefficient. The cross-section

integrals are computed for a number of cross-sections perpendicular to the flow, while keeping the reference integral at section Ia constant [1] (see Figure 2). The figure shows that the integrated result from the present computation is almost identical to the result from a CFX-5 computation [4].

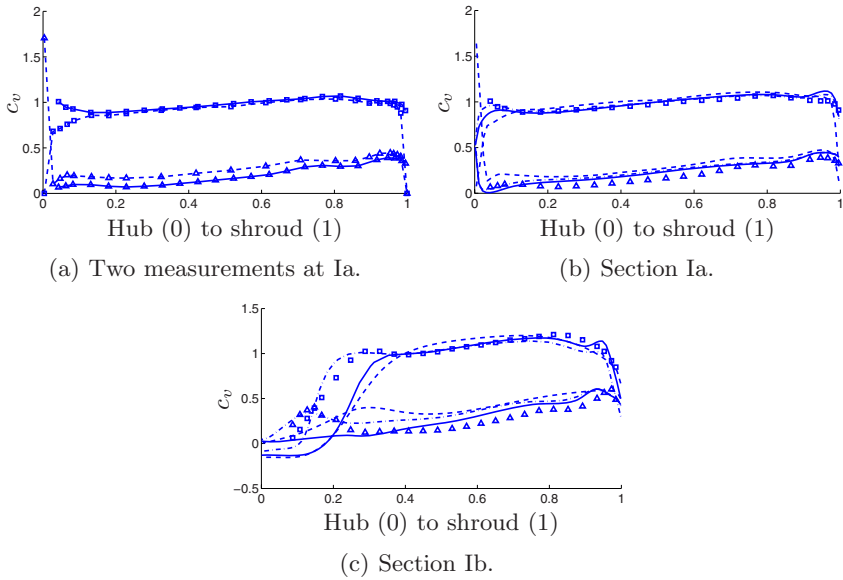
The pressure coefficient at the upper and lower center lines (see Figure 2) are compared with both the experimental data and the CFX-5 computation in Figure 3. The two computations are almost identical, and they are close to the experimental results.



**Fig. 3.** Computed integral quantities between section Ia and the end of the draft tube (a & b), and pressure coefficient distributions along the upper and lower centerlines (c & d). The only difference that can be seen is in (b), where OpenFOAM gives a slightly higher value at the end of the draft tube.

### 4.2 Velocity Distributions at Ia and Ib

Figure 4 compares the velocity coefficient distributions at cross-sections Ia and Ib (see Figure 2) with the experimental results. Figure 4(a) shows two different results from the measurements at cross-section Ia. This gives an indication of the accuracy of the measurements. For the draft tube computations it was recommended (Turbine-99) to use the results corresponding to the dashed lines as inlet boundary condition. Thus the dashed lines in Figures 4(a) and (b) are



**Fig. 4.** Velocity coefficient distributions. Markers: Measurements of axial (squares) and tangential (triangles) velocity components. In (a) two different measurements are shown, and the markers of each measurement are connected with solid lines and dashed lines respectively. In (b) and (c): Dashed lines: steady draft tube, Solid lines: runner without hub clearance, Dash-dotted lines: runner with hub clearance.

equivalent, and the measurement markers in Figures 4(b) and (c) correspond to the solid line measurements in Figure 4(a). The choice of those measurements as inlet boundary condition however doesn't seem to give the correct behaviour of the flow in the boundary layer at the hub. The solid lines in Figure 4(a) show that the axial velocity has a local maximum close to the hub, which is important for the development of the flow down to section Ib and throughout the draft tube.

Figures 4(b) and (c) compare the computational results with the experimental results. The dashed line corresponds to the steady draft tube computation while the dash-dotted and solid lines correspond to the runner computations with and without a runner blade hub clearance respectively. It is shown that the runner blade hub clearance produces an increase in the axial velocity close to the hub. This effect is very important for how well the separation at the hub is modelled. In Figure 4(c) the runner computation with the runner blade hub clearance resembles the experimental results much better than the other results, which is an effect of the small increase in axial velocity at section Ia. The velocity distribution at section Ib is very important for the flow development in the draft tube, and it is thus likely that the boundary condition for the draft tube computation is inadequate.

For both runner computations the effect of the tip clearance flow can be seen as a local maximum in the axial and tangential velocity profiles close to the shroud.

## 5 Parallel Performance

A parallel performance test has been made using the draft tube case described above, with about  $10^6$  cells. The decompositions of the domain into 2, 4, 8 and 16 subdomains were made using the automatic load-balanced decomposition (Metis) in OpenFOAM. The Linux cluster was a 4 node Dual socket AMD Opteron 280 (2.4 GHz, dual core) with 4GB DDR400 RAM, i.e. 4 cores (CPUs) per node and a total of 16 cores (CPUs). Two different interconnects were tested, a Gigabit Ethernet through an HP ProCurve 2824 Switch, and an Infiniband (PCI-X) through a Silverstorm 9024 Switch. The SuSE Linux Enterprise Server, Service pack 3 operating system was used. The analysis has been made together with Peter Rundberg at Gridcore ([www.gridcore.se](http://www.gridcore.se)).

The wall clock times were measured for three iterations, and the tests were repeated several times. No I/O was included in the performance test. The wall clock times are presented in Table 1. The table also shows the speed-up normalized by the single CPU run for each configuration, and the speed-up when using the Infiniband interconnect instead of the Gigabit interconnect (based on the speed-up columns in the table).

**Table 1.** Parallel performance using 1Gbit Ethernet (ETH) and Infiniband (IBA) interconnects. Packed vs. spread CPU distribution (the distribution of the processes on the nodes)

# CPU	# nodes	ETH (s)	IBA (s)	ETH (speed-up)	IBA (speed-up)	$\frac{IBA \text{ (speed-up)}}{ETH \text{ (speed-up)}}$ (based on speed-up)
1	1	165	163	1.0	1.0	1.0
2	1	86	78	1.9	2.1	1.1
2	2	85	81	1.9	2.0	1.0
4	1	76	72	2.2	2.3	1.0
4	2	64	62	2.6	2.6	1.0
4	4	53	56	3.1	2.9	0.9
8	2	43	41	3.8	4.0	1.0
8	4	41	35	4.0	4.7	1.2
16	4	23	20	7.2	8.2	1.1

Table 1 shows that the execution time of the computations decreases as more CPUs are used. When going from one to two CPUs the parallel speed-up is linear, but when using more CPUs the parallel efficiency is significantly reduced. When using 16 CPUs the parallel efficiency is 45% for the gigabit interconnect and 51% for the Infiniband interconnect. Table 1 shows no significant difference between the parallel runs using the different interconnects except when the computations are distributed on as many nodes as possible. This can most clearly be seen when using 8 CPUs on 2 and 4 nodes, where there is a 20% gain in using the Infiniband interconnect when distributing the computation on 4 nodes instead of

2 nodes. When distributing the processes on different nodes the computational requirement for each node is reduced and the requirement of the communication speed between the nodes is increased. When the computations are packed as much as possible on each node, the effect of the interconnect between the nodes is reduced. This effect could unfortunately not be further investigated for more nodes since there was only four nodes available during the present work.

Here follows some additional remarks on the results shown in Table 1. There is a small difference between the single CPU runs. Such differences are expected. Many things can influence computational speeds to this order of magnitude ( $\sim 1\%$ ). There is also a difference between the runs with 2 CPUs on 1 node. The interconnect should not be important in this case. It has however been observed that sometimes when running two processes on the same node they end up at the same socket, which influences the computational speed to this order of magnitude ( $\sim 10\%$ ). Later versions of the Linux kernel should have fixed this problem. An effect related to this can be seen in Table 1, where the speed-up is increased if the computations are spread over as many nodes as possible. Table 1 suggests that IBA has a worse tendency than ETH when distributing the 4 CPU case on different numbers of nodes. The more nodes involved, the more important is the interconnect. The table however suggests that the Gigabit interconnect is the best in the 4 CPU case.

## 6 Conclusions

This work has presented the water turbine field and its huge HPC requirements when the flow in a complete water turbine is to be predicted in detail. With complex geometries, high Reynolds numbers, turbulence, cavitation, and interaction between rotating and stationary parts, the computer power of today is still far from sufficient. By modeling parts of the physics and focusing the computations on parts of the water turbine it is however feasible to get reasonably accurate and useful solutions.

The OpenFOAM OpenSource tool was introduced and applied to CFD problems in the water turbine field. The results proved to be of the same accuracy as those of commercial CFD solvers. OpenFOAM is a true competitor to both commercial tools and in-house research codes. The major benefits of the OpenFOAM tool are that it is free of charge, and that the complete source code is available.

The parallel performance of OpenFOAM was analysed on a Linux cluster using an ordinary gigabit Ethernet interconnect and an Infiniband interconnect. The problem used for the analysis was the flow in a water turbine draft tube, and the grid consisted of approximately  $10^6$  control volumes. The Infiniband interconnect did not significantly improve the speed-up for this problem. The largest improvement is observed when the problem is distributed on as many CPUs as possible. When distributing the problem on 16 CPUs the Infiniband interconnect has a  $\sim 10\%$  speed-up compared with the corresponding gigabit Ethernet interconnect computation. The parallel performance of OpenFOAM



for the present problem was modest, with a parallel efficiency of approximately 50% when run on 16 CPUs.

## References

1. Nilsson, H., Page, M.: OpenFOAM Simulation of the flow in the Hölleforsen draft tube model. In: Proceedings of Turbine-99 III (2005)
2. Nilsson, H.: Evaluation of OpenFOAM for CFD of turbulent flow in water turbines. In: Proceedings of the 23rd IAHR Symposium in Yokohama (2006)
3. Andersson, U.: Turbine 99 - Experiments on draft tube flow (test case T). In: Proceedings of Turbine 99 - Workshop on Draft Tube Flow (2000) ISSN: 1402 - 1536
4. Page, M., Giroux, A.-M., Nicolle, J.: Steady and unsteady computations of Turbine99 draft tube. In: Proceedings of Turbine-99 III (2005)