Transient Simulation of Speed-No Load Conditions With An Open-Source Based C++ Code

E Casartelli, L Mangani, G Romanelli, T Staubli

Lucerne University of Applied Sciences & Arts, Department of Mechanical Engineering, Technikumstrasse 21, 6048, Luzern, CH.

E-mail: ernesto.casartelli@hslu.ch

Abstract. Modern reversible pump-turbines can start in turbine operation very quickly, i.e. within few minutes. Unfortunately no clear design rules for runners with a stable start-up are available, so that certain machines can present unstable characteristics which lead to oscillations in the hydraulic system during synchronization. The so-called S-shape, i.e. the unstable characteristic in turbine brake operation, is defined by the change of sign of the slope of the head curve. In order to assess and understand this kind of instabilities with CFD, fast and reliable methods are needed. Using a 360 degrees model including the complete machine from spiral casing to draft tube the capabilities of a newly developed in-house tool are presented. An ad-hoc simulation is performed from no-load conditions into the S-shape in transient mode and using moving-mesh capabilities, thus being able to capture the opening process of the wicket gates, for example like during start-up. Beside the presentation of the computational methodology, various phenomena encounterd are analyzed and discussed, comparing them with measured and previously computed data, in order to show the capabilities of the developed procedure. Insight in detected phenomena is also given for global data like frequencies of vortical structures and local flow patterns.

1. Introduction

Reversible pump-turbines are often used to cover peak-energy demand in turbine mode, while during periods with excess production they can be switched to pump-mode and use the available energy to fill reservoirs. With the deregulation of the electricity market, the switching from pump in turbine mode occurs more often and has to be accomplished very quickly, in order to fully exploit the machine flexibility.

During machine design special care has to be taken into account for the pump-mode, since increasing pressure is by far a more difficult task than accelerating the flow as in turbine mode. This can unfortunately lead to undesired behavior in turbine mode, around the no-load condition, as indicated in various publications [1, 2, 3, 4] and thus leading to machine or even system oscillations which can retard the synchronization procedure in turbine mode. Being able to have an insight on these phenomena is therefore very important, in order to understand and afterward develop suitable design guide-lines in order to avoid this kind of phenomena during operation.

The focus of the present paper is to demonstrate the ability of a in-house developed open-source based C++ code for numerically-tackling industry-relevant problems with complex unsteady flow-features, when both rigid and deformable dynamic motion of the physical domain are involved. To this end a transient simulation from the no-load condition into the S-shape

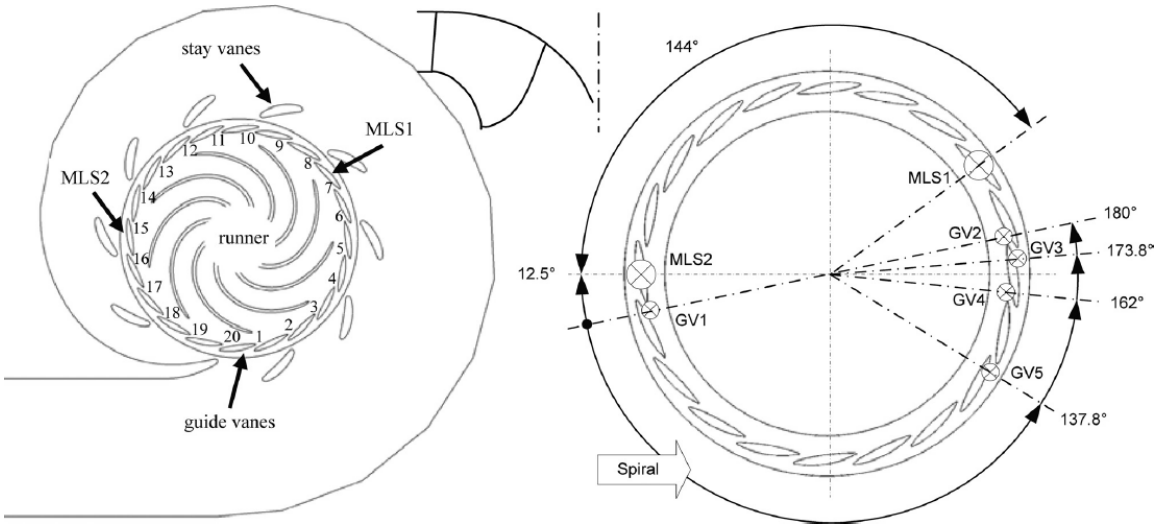


Figure 1. Outline of the investigated pump-turbine with numbered GV channels (left) including the meridional view (middle) and sensor arrangement in the guide vanes (right).

region was performed for a modern reversible pump-turbine, gradually increasing the guide-vanes opening angle. In such a way it is possible to investigate a) the potential occurrence of purely hydraulic (not induced by control system activation) unsteady phenomena at the no-load conditions and b) successively analyze the behaviour of the system during the unsteady motion of the guide-vanes. For a) the numerical results are compared with experimental data of the test campaign presented in [5]. For b) the numerical results are compared with those of a widely adopted commercial solver within industry.

2. Governing equations

The Unsteady Reynolds Averaged Navier-Stokes (URANS) equations governing the dynamics of an incompressible, viscous flow can be written within an Arbitrary Lagrangian Eulerian (ALE) framework in differential form as follows:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}_\mathrm{r}\mathbf{u}) = -\frac{1}{\rho}\nabla p + \nabla \cdot (\nu_\mathrm{eff}\nabla \mathbf{u}) \tag{2}$$

where $\mathbf{u}$ is the absolute velocity, p the pressure and ρ the density of the fluid. The convecting flux is written in terms of the relative velocity $\mathbf{u}_\mathrm{r} = \mathbf{u} - \mathbf{u}_\mathrm{ALE}$, where $\mathbf{u}_\mathrm{ALE}$ is the velocity associated to the dynamic motion (either rigid and/or deformable) of the physical domain $V = V(t)$ and it is evaluated enforcing the so-called Geometric Conservation Law (GCL). For instance, for a purely rigid rotation with constant angular velocity $\mathbf{\Omega}$, it corresponds to $\mathbf{u}_\mathrm{ALE} = \mathbf{\Omega} \times \mathbf{r}$. Finally the effective kinematic viscosity is the sum of the laminar and turbulent kinematic viscosities ($\nu_\mathrm{eff} = \nu + \nu_\mathrm{t}$).

3. Numerical discretization

Speed and robustness are among the most important requirements for any software that has to be used for the time-accurate numerical analysis of complex highly unsteady phenomena. When programming a CFD code the best way of combining both requirements is to couple the governing equations implicitly, since resolving efficiently the pressure-velocity coupling is essential for the performance of any CFD code. However up until today the SIMPLE family of algorithms [6], which couples the governing equations only by means of sub-looping, solving sequentially each governing equation, still remains the predominant methodology used in the CFD community. Therein a segregated approach in resolving the pressure velocity coupling is followed. Compared to block coupled implicit algorithms, segregated algorithms lack scalability with mesh size, robustness and calculation speed, which is inherent due in part to the under-relaxation needed to stabilize the algorithm. This is even more critical for unsteady simulations, as a set of inner non-linear iterations must be accounted for in order to drive the solver to the desired accuracy level in time.

In order to overcome these shortcomings Mangani et al. [7] developed a block coupled incompressible solver using the open-source CFD library OpenFOAM® as programming platform. Therein the algebraic equations resulting from the Navier-Stokes equations are solved simultaneously with the following system of equations, where 'C' are the cell values while 'NB' the neighbour contributions:

$$\begin{bmatrix} a_C^{uu} & a_C^{uv} & a_C^{uw} & a_C^{up} \\ a_C^{vu} & a_C^{vv} & a_C^{vw} & a_C^{vp} \\ a_C^{wu} & a_C^{wv} & a_C^{ww} & a_C^{wp} \\ a_C^{pu} & a_C^{pv} & a_C^{pw} & a_C^{pp} \end{bmatrix} \cdot \begin{bmatrix} u_C \\ v_C \\ w_C \\ p_C \end{bmatrix} + \sum_{NB} \begin{bmatrix} a_{NB}^{uu} & a_{NB}^{uv} & a_{NB}^{uw} & a_{NB}^{up} \\ a_{NB}^{vu} & a_{NB}^{vv} & a_{NB}^{vw} & a_{NB}^{vp} \\ a_{NB}^{wu} & a_{NB}^{wv} & a_{NB}^{ww} & a_{NB}^{wp} \\ a_{NB}^{pu} & a_{NB}^{pv} & a_{NB}^{pw} & a_{NB}^{pp} \end{bmatrix} \cdot \begin{bmatrix} u_{NB} \\ v_{NB} \\ w_{NB} \\ p_{NB} \end{bmatrix} = \begin{bmatrix} b_C^u \\ b_C^v \\ b_C^w \\ b_C^p \end{bmatrix} \tag{3}$$

To further enhance computational performance an algebraic multi-grid solver has been implemented and used for the solution of the block-coupled system of linear(ized) equations presented above.

While segregated algorithms operate using many under-relaxed sub-loops to account for inter-equation coupling, the inter-variable coupling is much stronger and less sub-loops are needed. Moreover no under-relaxation is needed. The solution of such a discretized system of equations therefore results to be numerically much more stable than that of segregated algorithms and also turns out to be significantly faster in terms of calculation time, which has been demonstrated by Mangani et al. [7]. The solver described in [7] has been used for the unsteady simulations carried out in this work and compared with the results of a reference commercial solver. More details regarding the modeling choices, the numerical set-up and the initial and boundary conditions are provided below.

Turbulence modeling All the simulations were performed using a $k-\omega$ SST turbulence model which requires the numerical solution of two scalar equations for the turbulent conservation variables (not coupled) in addition to momentum and continuity equations. Although two-equation turbulence models are known to have shortcomings at off-design conditions, the investigated flow instabilities were captured and fairly agree with experimental data [5, 4].

Domain discretization The simulation domain ranges from the spiral case inlet, the stay vanes, the guide vanes and runner to the draft tube as shown in Figure 2. Rotating and stationary domains were matched using transient rotor stator Arbitrary Mesh Interfaces (AMI). The global mesh encompassed approximately 6 million cells distributed in 4 cell zones (spiral/stay-vanes, guide-vanes, rotor and draft tube). As displayed in Figure 2, special refinement was applied in the rotor (approx. 2.5 million cells) and in the guide-vane regions (approx. 1.5 million cells).

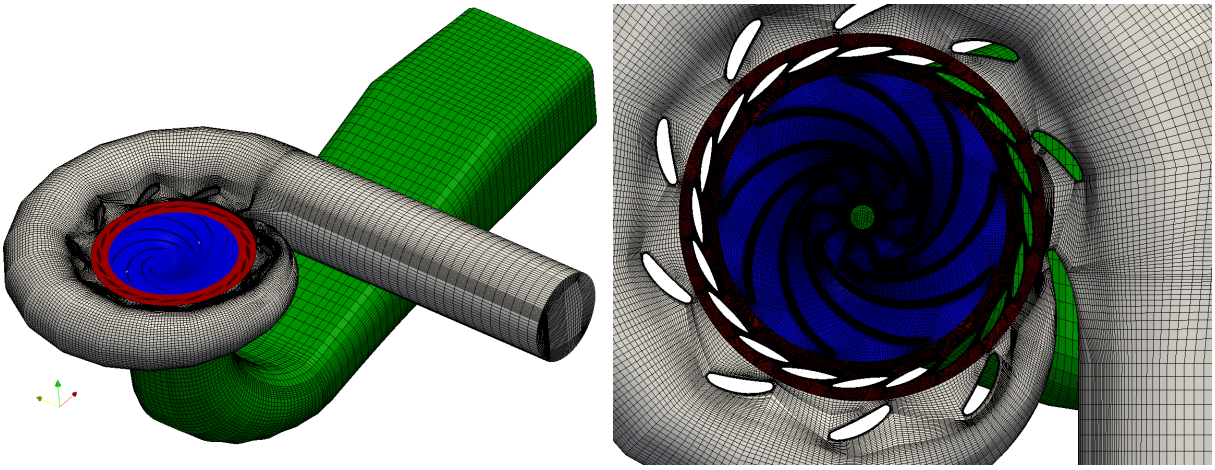


Figure 2. Computational mesh used for the transient time-accurate simulations with spiral, guide-vane, rotor, draft tube zones highlighted (left) and zoom in the guidevane/rotor interface zone (right).

Disc friction and leakage flow were neglected. The positions of the pressure sensors in the measurement configuration were also used as probe positions in the simulation.

Numerical schemes All the simulations were performed with fully second order upwind space discretization operators. Special schemes were developed in order to keep the stability and robustness also for the accelerated algorithm. Moreover ad-hoc switching to spatial first order was not necessary at the beginning of the simulations for stability issues as instead usually required by the standard simpleFoam solver [8]. A fully second-order time discretization is achieved by means of inner non-linear iterations (for the present simulations 4-6 sub-iterations were sufficient to achieve a drop of 5 orders of magnitude of the inner residuals), using a timestep of $\Delta t = 8.3333\,10^{-4}$ s, corresponding to a resolution of the rotor motion of 5 deg. In order to assess the quality of the results and the robustness of the numerical scheme adopted, the simulations were also run using a timestep of $\Delta t = 2.5\,10^{-3}$ s, corresponding to an angular resolution of 15 deg, without any numerical instability problems and leading to an only sligthly more diffusive solution.

Initial and boundary conditions All unsteady simulations have been initialized with a steady-state Multiple Reference Frames (MRF) solution. At the inlet of the computational domain the inlet velocity $\mathbf{u}_{\text{inlet}}$ is imposed, according to the experimental test matrix described in [5]. At the outlet of the computational domain, an average constant pressure of $p_{\text{outlet}} = 0$ Pa is used. The turbulence variables k, ϵ and ω are initialized as $k = 3/2(I|\mathbf{u}_{\text{inlet}}|)^2$, $\epsilon = C_\mu^{0.75} k^{1.5}/\ell$ and $\omega = \epsilon/k/C_\mu$ respectively, where I is the prescribed turbulence intensity, $C_\mu = 0.09$ is a constant turbulence model parameter and ℓ is the turbulent mixing length scale.

4. Mesh deformation

Once the structural displacements and velocities are suitably evaluated on the relevant boundary patches, it is necessary to tackle the problem of adjusting the internal hydrodynamic mesh to the newly computed boundary nodes in such a way that grid quality is not degraded significantly, e.g. with non-negative cell volumes, moderate levels of stretching and non-orthogonality. This task of moving what can amount up to millions of nodes has to be performed thousand of times during an unsteady FSI computation (for both one-way or two-way coupling). Therefore

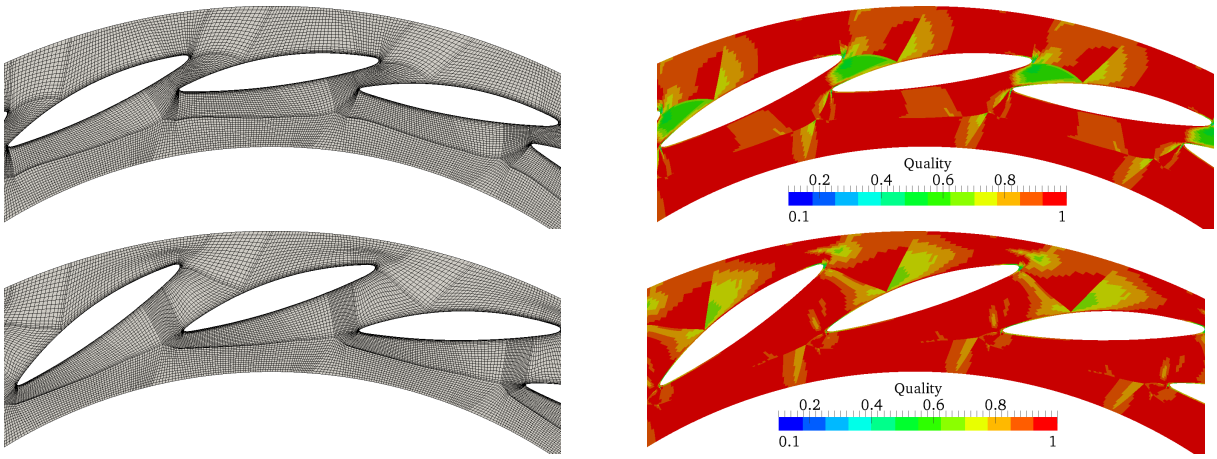


Figure 3. Comparison of normalized Jacobian quality index before (top) and after (bottom) mesh deformation via SIDW interpolation for $\delta = 10$ deg guide-vane rotation.

the availability of efficient, massively parallel and robust mesh deformation tools and topology modifiers is crucial.

A significant body of research has been performed in the area of grid motion schemes for aeroelastic analysis. These range from simple algebraic shearing methods to those that model the grid as a physical field equation that can be coupled with the analysis [9]. Some of the most appealing strategies are based on the continuum analogy between the computational grid and an elastic solid [10, 11] and multivariate interpolation schemes such as RBF or IDW [12, 13]. However all these options differ significantly in terms of accuracy, robustness and (parallel) efficiency.

In order to achieve the best compromise between those opposing requirements, rather than using algorithms that require solving a systems of equations (computationally expensive), we choose an explicit mesh deformation technique. Given the structural displacement $\boldsymbol{s}_j$ of each j-th solid boundary d.o.f. $\boldsymbol{x}_j$, the displacement $\Delta\boldsymbol{x}_k$ relative to the reference undeformed configuration $\boldsymbol{x}_k$ for each k-th internal d.o.f. can be evaluated by performing the following matrix-array multiplication with an asymptotic computational cost of $\mathcal{O}(N_{a,\,v}\,N_{a,\,b})$:

$$\Delta\boldsymbol{x}_k = \sum_{j=0}^{N_{a,\,b}} \frac{\mathbf{IDW}_{(k,\,j)}}{|\mathbf{IDW}_{(k,\,:)}|}\,\boldsymbol{s}_j \qquad \forall \quad k = [1,\,N_{a,\,v}], \tag{4}$$

where $\mathbf{IDW}_{(k,\,j)} = |\boldsymbol{x}_k - \boldsymbol{x}_j\|^{-p}$. The parameter p can be used as a knob to adjust the size of the computational stencil and therefore the smoothness of the results. In the present work we adopted $p = 3$. As it is, this strategy can be either time-efficient or memory-efficient, depending on whether matrix $\mathbf{IDW}$ is computed only once in the pre-processing stage and then stored with a large memory overhead. In order to achieve the best trade-off we propose a simple and effective modification to the original interpolation algorithm [14, 12] called Sparse Inverse Distance Weighting (SIDW) in which matrix $\mathbf{IDW}$ is stored in compressed sparse column format as follows:

$$\mathbf{SIDW}_{(k,\,j)} = \mathbf{IDW}_{(k,\,j)} \qquad \text{if} \qquad \frac{\mathbf{IDW}_{(k,\,j)}}{|\mathbf{IDW}_{(k,\,:)}|} > \xi \tag{5}$$

where ξ is a threshold parameter to be set by the user. However if its value is too large and there is no active j-th column for k-th row, it is iteratively doubled until a safe number of active columns per row is reached. The computational cost and memory requirements can be estimated as $\mathcal{O}(c\,N_{a,\,v})$, where $c \ll N_{a,\,b}$ is the average number of active columns per row. In practice the

present strategy makes the effort associated to mesh deformation almost negligible with respect to the effort associated to the solution of the governing equations. This method is powerful also because it does not require any knowledge of the connectivity of the mesh and it is particularly suited for massively parallel simulations.

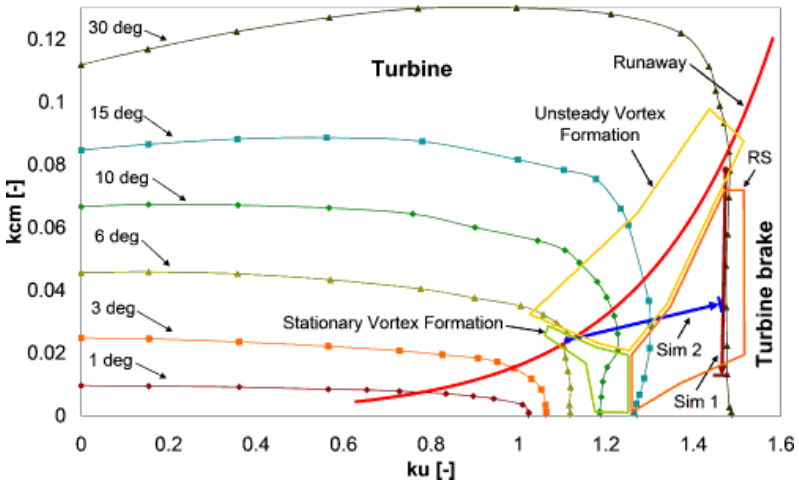
With reference to the investigated pump-turbine geometry, in Figure 3 a detail of the computational grid is shown before and after a rigid $\delta = 10$ deg rotation is applied to each guide-vane, thus changing the GVO angle from 6 deg to 16 deg. It is possible to appreciate that the present mesh deformation algorithm is able to handle efficiently the provided displacements input, but it also yields an average improvement of overall normalized Jacobian mesh quality index (which measures the deviation of a generic hexahedron from the reference element, thus combining non-orthogonality and skewness information), especially in the opening region, where the cell quality shifts from average values of approximately 0.6 to almost 1.0.

5. Numerical results

In the present section the numerical results of the transient simulation in a modern reversible pump-turbine are presented, starting from speed-no load conditions and gradually increasing the guide-vanes opening angle accordingly in order to reach the S-shape region in the turbine brake quadrant, as indicated in Fig. 4. The results are then compared with experimental and numerical data available.

The simulations were run on 64 cores (4 nodes with 16 Intel Xeon E5-2660 (2.2 GHz) processors each). Each core has 32 GB DDR3-1600 MHz RAM. With 5 deg angular resolution it took 23 hours to complete 25 revolutions. The pre-processing routines for building the IDW operator required 15 minutes, while during runtime the mesh deformation effort is negligible.

With reference to Figure 5, the mass-flow is initially set to 31 kg/s and the guide-vane opening angle to 6 deg. This operating condition is kept constant for 10 revolutions, only having rigid mesh motion, in order to appreciate the unsteady phenomena related to the no-load condition. Successively, using mesh deformation, the guide vanes are opened by approximately 10 deg in 15 revolutions (simultaneous GV opening by 0.0014 degree and mass flow increase by $0.558\,10^{-3}$ kg/s per 1 degree runner revolution) in order to reach the S-shape instability (see Fig. 4). The angular velocity is kept constant to $n = 1000$ rpm during the whole process.



$$k_u = \frac{n\pi D_1}{\sqrt{2gh}}$$

$$k_{cm} = \frac{4Q}{D_1^2\pi\sqrt{2gh}}$$

Figure 4. Turbine quadrant with measured characteristics for different guide vane angles and with the simulated conditions with varying GVO and flow rate (sim 2).

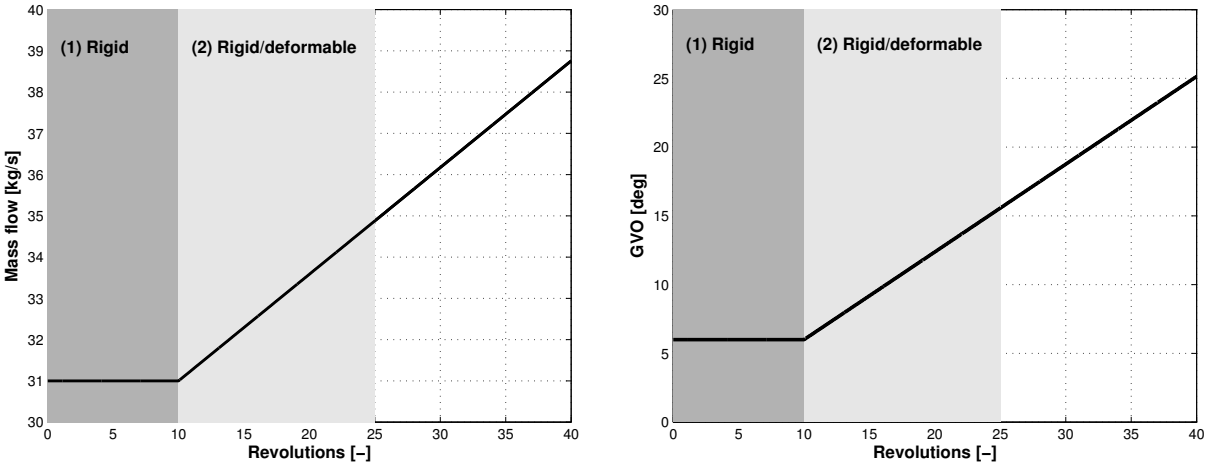


Figure 5. Curves describing the mass-flow (left) and guide-vane opening angle (right) as a function of number of revolutions.

5.1. Speed-no load phase

Figure 6 illustrates the time histories of the 5 pressure sensors (measurement), and probes (simulation) in the guide vanes within 7 runner revolutions. As in [5], the blade passing frequency (150 Hz) is damped by means of a running average procedure, namely $p_k = \sum p_{k+j}/N$ where $j \in (0, N)$ and the number of data N (inversely proportional to filtering band) corresponds to 5 blade passing periods and depends on the time resolution of acquired and simulated data. Moreover it is convenient to represent the numerically computed and experimentally measured pressure data in adimensional form as the normalized turbine head H/H_0, where the reference turbine head H_0 is evaluated as the average value during 10 revolutions.

It is possible to observe, both in the measurement as well as in the CFD simulation, small pressure fluctuations at the probes GV1 to GV5. Especially the sensors 3 and 5, which are less influenced by the blade passing frequency, confirm the pressure fluctuations in the simulation. The amplitude in the simulation ($\pm 6\%$) is significantly higher than that of the measurement ($\pm 4\%$). This discrepancy is also observed in the numerical results presented in [5], obtained with a commercial solver and it can be related to different system properties and boundary conditions (neglected system elasticity and fixed massflow instead of fixed head). The Fourier analysis of the signals shown in Figure 7 also confirms the presence of a small pressure fluctuation. In particular the simulation data feature for all probe points GV1 to GV5 a significant peak at a frequency of approximately 6 Hz, which corresponds to approximately 1/3 of the blade-passing frequency. Also experimental data show a peak at a frequency of approximately $5 - 6$ Hz even if weaker in amplitude, especially for probe points GV3 and GV5. This phenomenon was attributed in [5] to unsteady vortex formation in the vaneless space.

In Figure 8 an example of the detected vortical structures is given. As it can be seen, the flow pattern at speed-no load is very complex and far away from a sound flow like that encountered around the best efficiency operation. The hydraulic energy input into the turbine is completely dissipated in a highly unsteady process to achieve no-load conditions on the shaft. As it can be seen in Figure 8, on both pressure and suction side of the runner blades the flow is in reverse direction (pumping) while large part of the passage is blocked. The partially pumping turbine generates a flow back into the vaneless space along the blades, absorbing the torque generated in the downstream part of the passage and thus leading to an overall no-load condition on the runner. In fact, due to the observed vortices, which have a tangentially-oriented axis spanning from pressure to suction side, the flow is only able to enter the turbine passage at hub and shroud, as shown in Figure 9. Here flow separation, recirculation, and vortex formation were observed in every channel during operation at speed no load at 6 degree GVO.

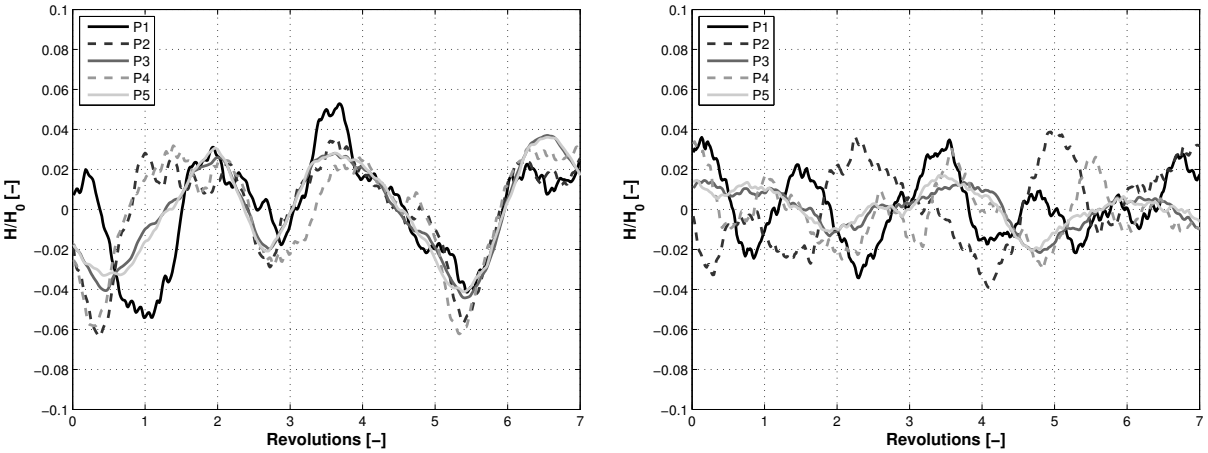


Figure 6. Comparison of the adimensional turbine head time history H/H_0 between numerical (left) and experimental (right) data.

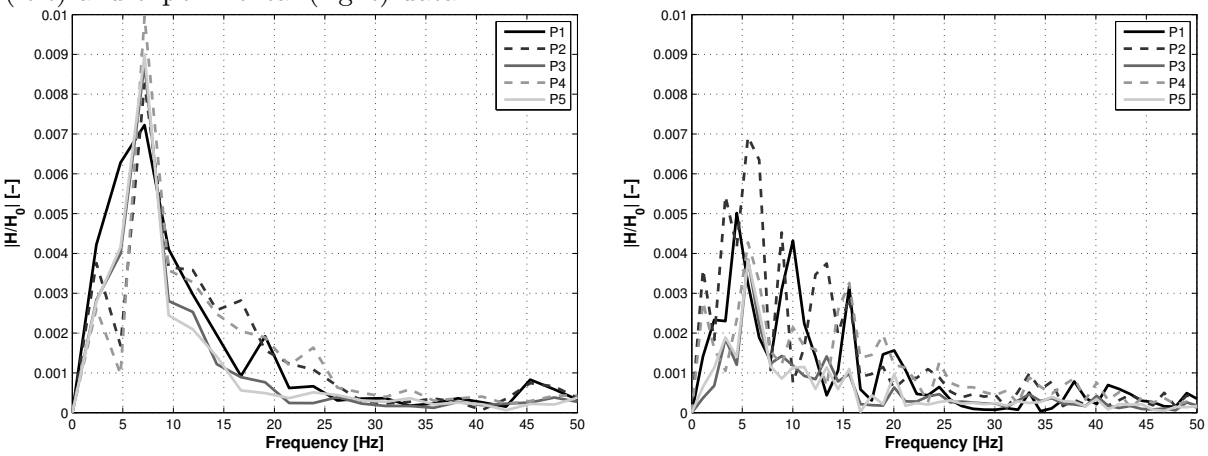


Figure 7. Comparison of the Fourier coefficients magnitude of adimensional turbine head H/H_0 as a function of frequency f between numerical (left) and experimental (right) data.

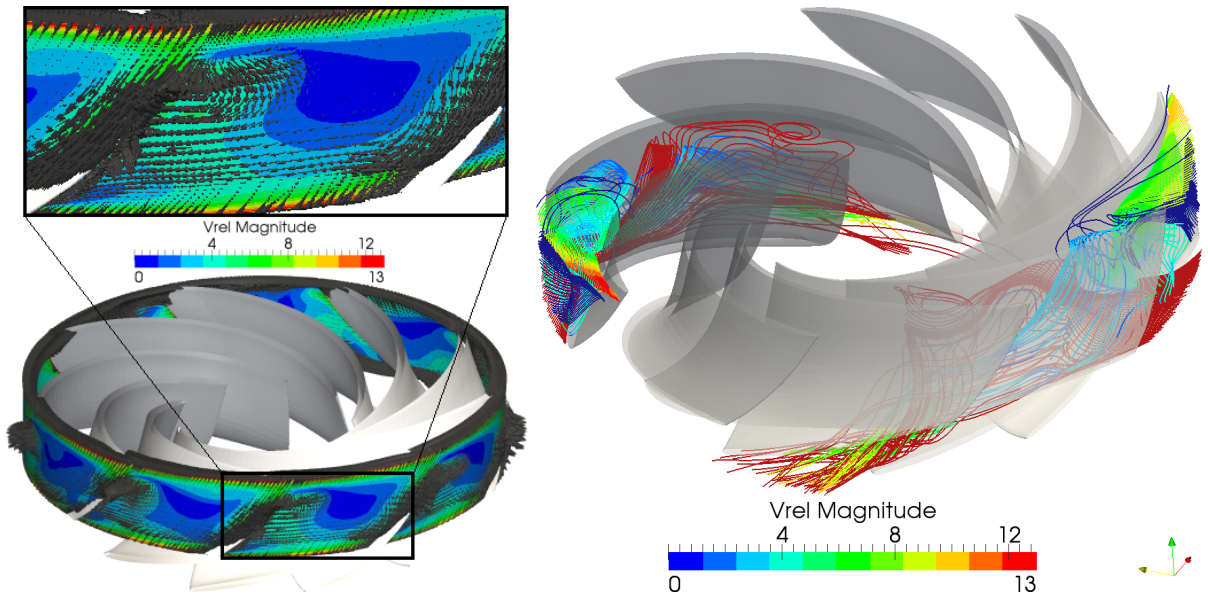


Figure 8. Flow-field of the relative velocity at the interface between rotor/guide-vane regions (left) and streamlines within two passages, highlighting the blockage phenomenon (right).

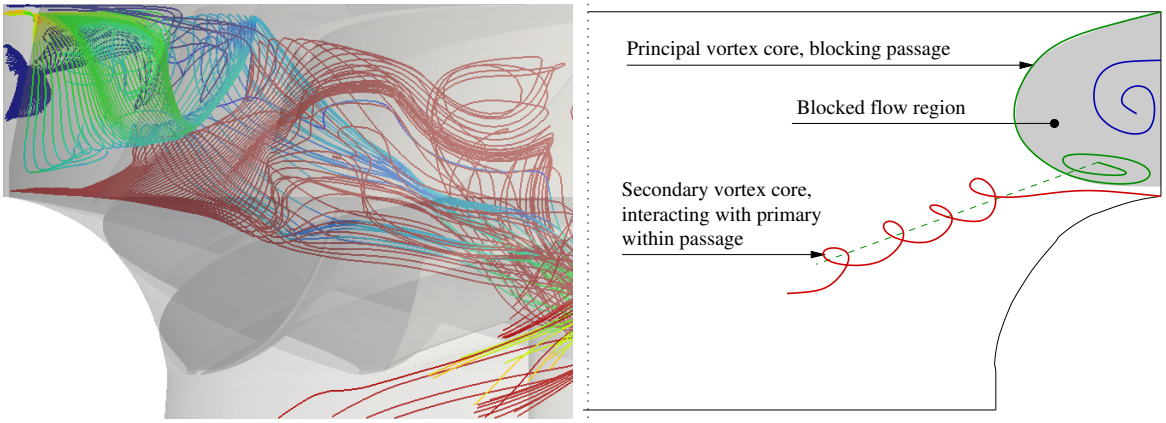


Figure 9. Streamlines of the relative velocity (left) and conceptual representation illustrating the blocking passage phenomenon (right).

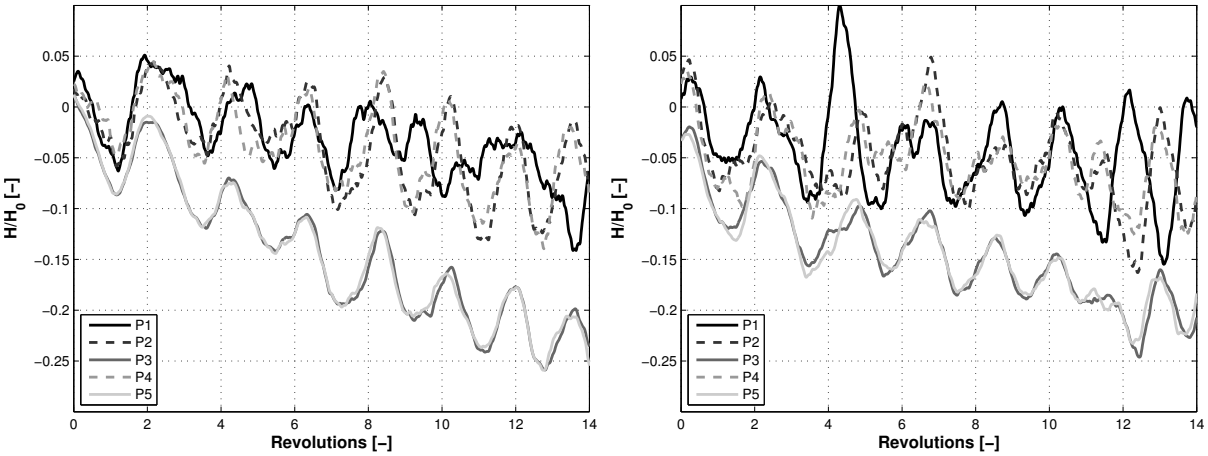


Figure 10. Comparison of the adimensional turbine head time history H/H_0 between present (left) and reference numerical (right) data.

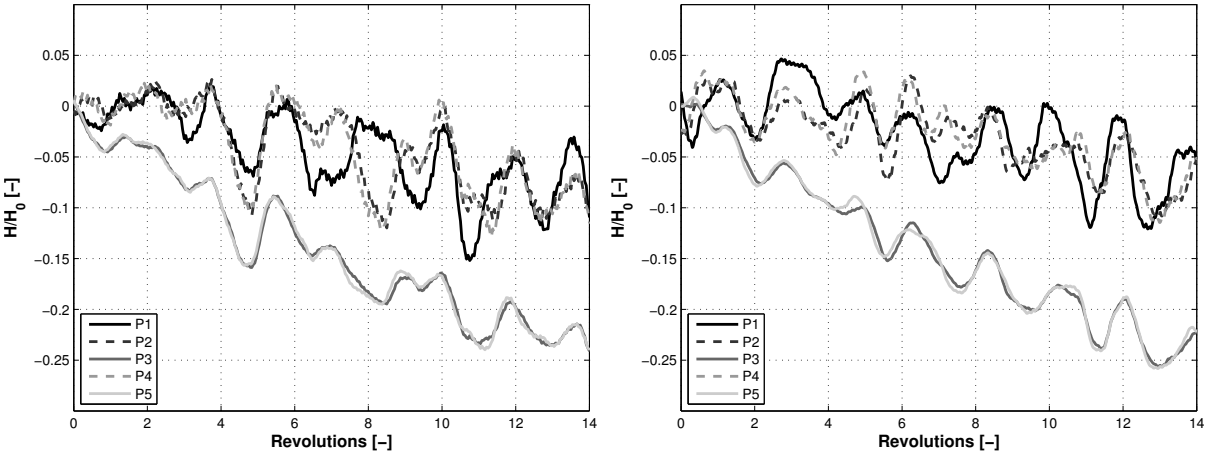


Figure 11. Adimensional turbine head time history H/H_0 with 2-nd order accurate time-scheme with 15 deg angular resolution (left) and 1-st order accurate time-scheme with 5 deg angular resolution (right).

5.2. Guide-vane opening phase
In Figure 10 the predicted dimensionless turbine-head oscillations are compared with the results of a reference commercial solver for the first 14 revolutions of guide-vane opening phase, with GVO angle passing from 6 deg to approximately 16 deg. In general there is a fairly good agreement. In particular the average decreasing trends shown by the two groups of pressure probes GV1/GV2/GV4 and GV3/GV5 are similar, underlining the validity of the new solver. Moreover in Figure 11 we present the numerical results obtained increasing the angular resolution (and therefore the timestep) up to 15 deg. First it is worthwhile to highlight that such a choice did not yield any robustness issues within the CFD code and allowed to reduce the overall computational time for completing 25 revolutions down to approximately 10 hours. The results are consistent with those obtained above in terms of average trends. Of course the oscillations amplitude are slightly damped, with a damping effect similar to that of switching to 1-st order accurate time scheme with 5 deg angular resolution.

Conclusions
In the present paper a methodology to compute the flow field at no-load conditions and in turbine brake operation has been presented. The methodology is based on the open source toolbox OpenFOAM. Using a novel coupled-solver (compared to the standard segregated OpenFOAM solvers) and extending it with moving-mesh capabilities gained from aeronautical applications, a fast procedure for the analysis of this kind of unsteady conditions could be developed and successfully tested. The results, both global and local, are in line with measurements and previous computations with a commercial solver, showing the maturity of the proposed procedure. The flow analysis shows a strong interaction of different flow patterns in the vaneless space: the highly tangential flow out of the guide-vanes produces vortices at the runner inlet which largely block the flow passage. While at hub and shroud the flow can enter the runner and produce torque, at both pressure and suction-side flow is pumped out of the runner, thus using the generated torque and leading to a no-load condition.

References
[1] Martin C S 1986 *5th Intl. Conf. on Pressure Surges*
[2] Martin C S 2000 *Proc. 20th IAHR Symp. Hydraulic Machinery and Systems*
[3] Pejovic S, Krsmanovic L, Jemcov R and P C 1976 *Proc. of 8th IAHR Symp. on Hydr. Mach. and Cavitation*
[4] Yan J P, Seidel U and Koutnik J 2012 *26th IAHR Symposium on Hydraulic Machinery and Systems*
[5] Widmer C, Staubli T and Ledergerber N 2011 *Journal of Fluids Engineering* **133** 041101
[6] Patankar S 1980 *Numerical Heat Transfer and Fluid Flow* (Hemisphere Publishing, Washigton)
[7] Mangani L, Buchmayr M and Darwish M 2014 *Numerical Heat Transfer, Part B: Fundamentals* **66** 1–20
[8] OpenCFD 2014 *OpenFOAM User Guide* OpenCFD Limited
[9] Schuster D M, Liu D D and Huttsell L J 2003 *Journal of Aircraft* **40** 5
[10] Cavagna L, Quaranta G and Mantegazza P 2007 *Computers & Structures* **85** 11
[11] Jasak H 2009 *47-th AIAA Aerospace Sciences Meeting*
[12] Witteveen J 2009 *47-th AIAA Aerospace Sciences Meeting*
[13] Romanelli G, Castellani M and Ricci S 2012 *53-rd AIAA/ASME/ASCE Structural Dynamics and Materials Conference*
[14] Shepard D 1968 *ACM National Conference*