

Evaluation of an improved mixing plane interface for OpenFOAM

M Beaudoin¹, H Nilsson², M Page¹, R Magnan¹, H Jasak^{3,4}

¹ Institut de recherche d'Hydro-Québec (IREQ), Varennes, Qc, Canada

² Chalmers University of Technology, Gothenburg, Sweden

³ Wikki Ltd., London, United Kingdom

⁴ FSB, University of Zagreb, Croatia

E-mail: beaudoin.martin@ireq.ca

Abstract. A mixing plane interface provides a circumferentially averaging rotor-stator coupling interface, which is extremely useful in practical turbomachinery simulations. It allows fundamentally transient problems to be studied in steady-state, using simplified mesh components having periodic properties, and with the help of a multiple reference frames (MRF) approach. An improved version of the mixing plane interface for the community-driven version of OpenFOAM is presented. This new version of the mixing plane introduces a per-field, user-selectable mixing option for the flow fields at the interface, including the possibility to use a mass-flow averaging algorithm for the velocity field. We show that the quality of the mass-flow transfer can be improved by a proper selection of the mixing options at the interface. This paper focuses on the evaluation of the improved mixing plane interface for various steady-state simulations of incompressible flows, applied to a simple 2D validation test case, and to more complex 3D turbomachinery cases.

1. The OpenFOAM CFD simulation platform and its main versions

OpenFOAM is a high-quality open-source CFD simulation platform, which offers new R&D opportunities by providing direct access to models and solver implementation details. OpenFOAM's well-designed object-oriented C++ architecture allows the toolbox to be enhanced efficiently by focusing on specific libraries and re-using existing capabilities [1] [2]. Since its public release back in 2004, the OpenFOAM software has seen many revisions and enhancements. The source code of OpenFOAM is accessible in full and released under the GPL license [3], thus granting full rights to distribute and modify the source code as per the requirements and obligations of this license. Two major versions of OpenFOAM are available nowadays: one version being developed and maintained by a corporation, and a community-driven version that is evolving independently of the so-called main branch.

The community-driven version of OpenFOAM, called OpenFOAM-1.6-ext, has been a driving force for innovation for many years. Given the growing popularity of the OpenFOAM simulation platform in academia and in the industry, many new and very interesting developments has sprung out outside of the main development branch since 2004. Many of those developments are now integrated into OpenFOAM-1.6-ext, a version released under the community-driven open-source project called OpenFOAM-extend, currently hosted on the public collaboration site SourceForge.Net [4].

More recently, a new fork of the OpenFOAM software was released under the name and version foam-extend-3.0. This new project will keep offering to the community of CFD researchers and engineers a sound and stable platform over which new contributions and innovations can be built and contributed openly, in the open-source movement spirit, and with full and public recognition of the authors' copyrights and intellectual property.

2. Enhancing OpenFOAM for turbomachinery simulation

Some of the numerous developments contributed under the OpenFOAM-extend project in recent years are related to the usage of OpenFOAM for turbomachinery simulation. Turbomachinery simulation represents a significant portion of contemporary Computational Fluid Dynamics (CFD) [5] [6]. The simulation of steady-state, Multiple Reference Frames (MRF) capabilities for hydroturbines has driven the development of numerous missing components for OpenFOAM [7], namely turbomachinery-specific validation test cases [8] [9], specialized boundary conditions, file format converters based on the CGNS industry standard, interpolation interfaces for non-conformal meshes [10] [5], MRF solvers and improved parallel decomposition tools [11].

More specifically, the availability of interpolation interfaces for non-conformal meshes is critical for the simulation of turbomachinery components. One very common practice for the simulation of 3D components with complex geometries like turbine runners or distributors (stay vanes and wicket gates) is to use a single passage or sector of these components, which can be spatially discretized using meshes with non-matching periodic boundaries. The GGI interface accounts for the coupling of those boundaries in an implicit manner, without necessitating any remeshing at the interface. The General Grid Interface (GGI) and cyclicGGI interface for OpenFOAM [10] were the first community-contributed design to serve this purpose.

When connecting these simplified turbomachinery flow passages (distributor, runner) to full-geometry components like a diffuser and a spiral casing in order to evaluate the steady-state simulation of the full assembly, a non-conformal interpolation interface like a mixing plane is also necessary. When using a multiple rotating reference frame approach, this interface performs a circumferential averaging of the solution at the rotor-stator interface [5] [7].

An example of the assembly of a steady-state turbine problem is shown in figure 1, with a distributor passage (blue), a runner passage (green) and a full draft tube geometry (red). Such a configuration requires two mixing plane interfaces, one between the distributor and the runner sectors and one between the runner sector and the full draft tube geometry. Periodicity of the runner and distributor sector requires cyclicGGI interfaces.

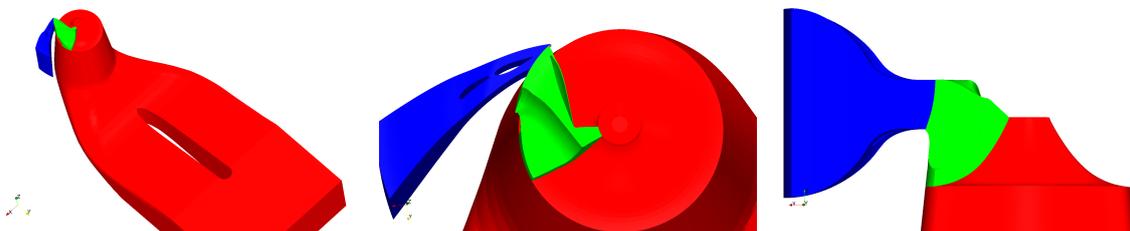


Figure 1. Steady-state problem requiring two mixingPlane interfaces.

3. A quick overview of the mixingPlane interface

The mixing plane interface, hereafter called the mixingPlane, is a recent addition to the community-driven versions of the OpenFOAM simulation toolbox. The mixingPlane is essentially a circumferential averaging interpolation interface.

The mixingPlane interpolator is internally constructed using an intermediary cylindrical or radial patch surface made of a stack of 360° ribbons shared between two specially crafted GGI interfaces. Care was taken in the discretization of the ribbon patch to appropriately interpolate both the coarse and fine regions of the upstream and downstream patches from the mixingPlane interface. A

discretization profile is constructed based on the 2D upstream and downstream patch point distribution in cylindrical space. This 2D profile serves as the discretization control curve for generating the intermediary 360° ribbons patch. The mixingPlane averaging factors are derived directly from the two GGI interfaces weighting factors, as a consequence of the shape of the mixing patch and of the facet area-based assembly of GGI weights.

A simplified illustration of the mixingPlane interpolation ribbons is shown in figure 2. Two GGI interfaces are created to interpolate the flow values between the downstream and upstream domains; one GGI interface using the downstream patch D and ribbon patch R, and the other GGI interface using the ribbon patch R and the upstream patch U.

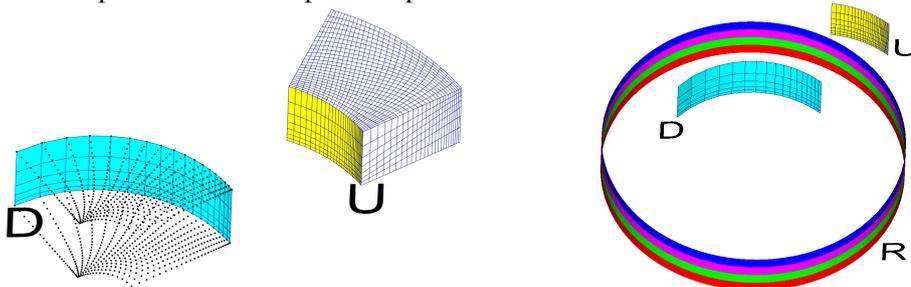


Figure 2. Illustration of a simple mixingPlane interface configuration.

4. Improving the quality of the mixingPlane interpolation using a synthetic validation test case

The mixingPlane source code for OpenFOAM was first released publicly in 2012, just before the 7th OpenFOAM Workshop held in Darmstadt, Germany. Since then, numerous improvements have been contributed to this development.

4.1. The RC1 version

The first implementation of the mixingPlane, dubbed the *RC1* version, laid out the basic infrastructure necessary for this kind of coupling interface [12]. This initial design is reusing the generalized grid interface (GGI) as the underlying area-weighted interpolator and introduced a novel mesh-driven discretization algorithm for the determination of the circumferential interpolation bands. This first basic mixingPlane interpolation scheme is called *areaAveraging*. Still, extensive validation tests of this initial design have shown the limitations of an area-weighted-only interpolator for the circumferential averaging of flow fields. In the presence of large flow gradients very close to the interface, it was shown that the mixingPlane interface was still mass-conservative, but at the expense of disturbances of the flow fields in the vicinity of the interface, mostly for the pressure field [13]. The behaviour of the mixingPlane was demonstrated using a synthetic 2D test case named *domAdomB* [14], in which two stationary flow passages are artificially coupled by a mixingPlane. The test case was published and documented in details [13] [14] [15] to illustrate the current limitations of the RC1 design, and to help in the validation and improvement of this development.

4.2. The RC2 version

A major improvement to the mixingPlane code came in 2013, shortly before the 8th OpenFOAM Workshop held in Jeju, South Korea, thanks to the help and sound advices of the Turbomachinery Working Group members. For this RC2 version, a mass-flow averaging interpolator was introduced in order to complement the area-weighted interpolator algorithm. Additional mixingPlane interpolation schemes were introduced in this version, namely the *fluxAveraging* and *zeroGradient* mixingPlane schemes. Using various mixingPlane specific schemes or "mixing recipes", it was now possible for example to apply the mass-flow averaging interpolator only to the velocity field, while still using the area-averaging interpolator for the other fields.

Using the same *domAdomB* 2D test case, we were able to show that the use of the mass-flow interpolator for the velocity field greatly diminishes the disturbances observed close to the

mixingPlane interface, but at the expense of a significant mass-flow loss at the interface. Basically, the RC2 version of the mixingPlane was no longer mass conservative when using the mass-flow averaging interpolator for the velocity field.

4.3. The RC3 version

Based on the results obtained with the RC1 and RC2 versions of the mixingPlane interface, a new RC3 version was designed shortly after the Jeju OpenFOAM Workshop. The RC3 version basically combines the mass conservation quality of the RC1 version with the mass-flow averaging algorithm of the RC2 version.

4.3.1. A new mass-conservative mixingPlane velocity scheme

A new mixingPlane interpolation scheme was introduced for the velocity field. Under that new scheme, the velocity field first goes through the mass-flow averaging interpolator, and the resulting mass-flow balance is computed across the mixingPlane interface. Then the area-averaging interpolator is invoked for the same field in order to compute an adjusted mass-flow balance. The ratio between the two mass-flow values provides a scaling factor to be applied to the component of the velocity field normal to the mixingPlane patches. Basically, the RC3 version of the mixingPlane is applying a mass conservative area averaging scheme to the component of the velocity field normal to the mixingPlane patches while a mass-flow averaging scheme is being applied to the other components. The name of this new mixingPlane scheme is *fluxAveragingAdjustMassFlow*.

4.3.2. A new zero-gradient-like mixingPlane scheme for the pressure field

In addition to this new velocity scheme, a new mixingPlane scheme was introduced for the pressure field. This new scheme basically behaves like a zero-gradient scheme, while keeping the average pressure level the same on both sides of the interface. The name of this new mixingPlane scheme is *zeroGradientMatchAverageValue*.

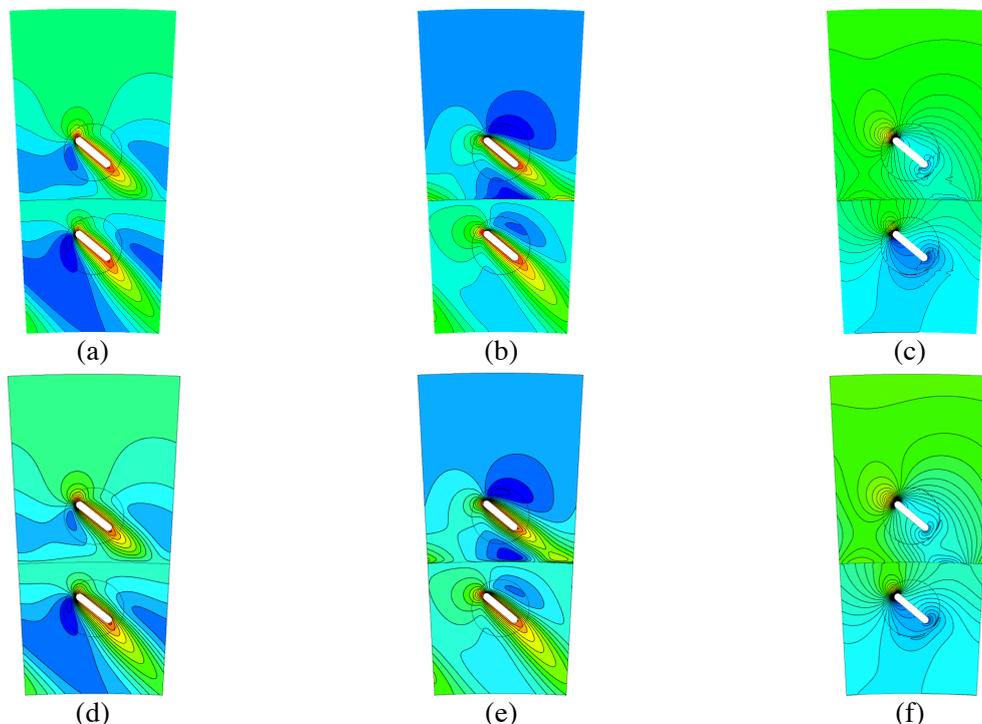


Figure 3. Comparison between the RC3 version of the mixingPlane code and a commercial solver for the domAdomB test case. RC3 version - (a): radial velocity, (b): tangential velocity, (c): pressure. Commercial solver - (d): radial velocity, (e): tangential velocity, (f): pressure.

4.3.3. Comparison of the RC3 version results with a commercial software

Figure 3 above shows a comparison of the results we get for the domAdomB test case, when using the RC3 version of the mixingPlane for OpenFOAM and a commercial solver. The same contours are used for both codes. It can be seen that the pressure and velocity distributions are almost identical between the RC3 version of the mixingPlane for OpenFOAM and the commercial solver implementation.

5. Using the mixingPlane interface with turbomachinery test cases

Based on the experience with the RC3 version of the mixingPlane for the domAdomB test case, an analysis of its behavior and performance for more complex 3D turbomachinery test cases is here presented. The two test cases are a Francis hydroturbine application and the Turbine-99 Kaplan turbine model.

5.1. Francis hydroturbine application

The RC3 version of the mixingPlane for OpenFOAM is used for a 195-MW Francis turbine computation, as illustrated in figure 4. The computational domain couples one of the 20 distributor passages with one of the 13 runner passages. The discretization of the distributor and runner passages is using 0.12M and 0.27M hexahedral cells respectively. The periodic sides of each passage are coupled using cyclicGgi interfaces. The distributor and the runner passages are connected by a mixingPlane interface with interpolation ribbons generated in the axial direction. The upstream and downstream patches of the mixingPlane interface are shown in figure 4. It should be noted that the axial ribbons are not defined at a constant radius.

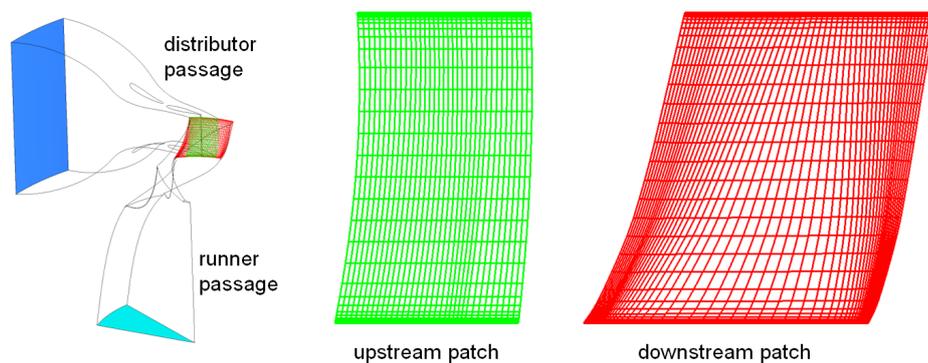


Figure 4. Upstream (green) and downstream (red) patches of the mixingPlane interface between the distributor and the runner passages of a Francis hydroturbine.

Computations are performed at a constant guide vane opening by specifying the flow rate and the turbulent quantities at the inlet of the distributor passage. A constant zero pressure is specified at the outlet of the runner passage. The steady-state Reynolds-Averaged Navier-Stokes equations, formulated in the multiple frames of reference, are used for the computations [7]. The standard $k-\epsilon$ model with log-law near the wall is used. The second-order linearUpwind scheme is selected for the discretization of the convection term of the momentum equations.

The mixingPlane discretization is using the *fluxAveragingAdjustMassFlow* scheme for the velocity, the *zeroGradientMatchAverageValue* scheme for the pressure, and *areaAveraging* scheme for the turbulent quantities. Figures 5 and 6 show a comparison between the results obtained with RC3 version of the mixingPlane interface with OpenFOAM and results obtained with the mixing plane interface of a commercial CFD code.

Figure 5 compares the absolute radial, tangential and axial velocity components, the static pressure, the total pressure in the stationary frame, and the turbulent kinetic energy on the upstream and downstream patches of the mixingPlane interface. Globally, the fields are similar on the downstream

side but we observe that the static and total pressure, the radial velocity, and the turbulent kinetic energy are captured differently by the two codes on the upstream side of the mixingPlane. Figure 6 compares the fields at the mid-height of the distributor and runner passages across the mixingPlane interface. Again, the results are globally similar. The same contour limits have been selected to highlight the variation of the fields near the interface. There are significant differences between the two codes near the wicket gate for the radial and tangential component of the velocity. It may be related to the fact that the pressure treatment is different on the upstream patch of the interface as shown in figure 5.

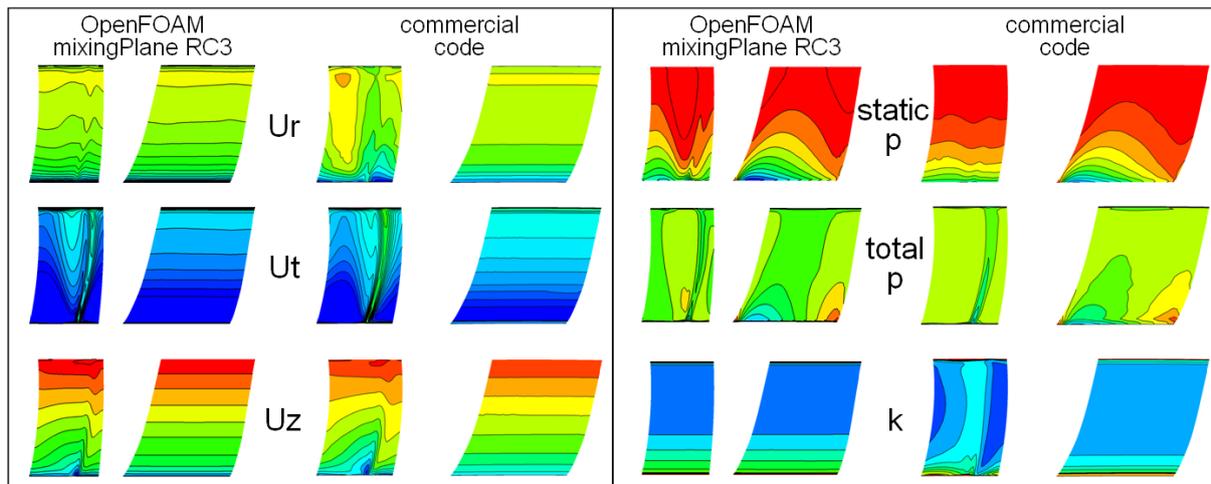


Figure 5. Field distributions on the upstream and downstream patches of the mixingPlane interface for the absolute radial, tangential and axial components of the velocity, for the static pressure, for the total pressure in the stationary frame, and for the kinetic turbulent energy. The same contour levels are used for both codes.

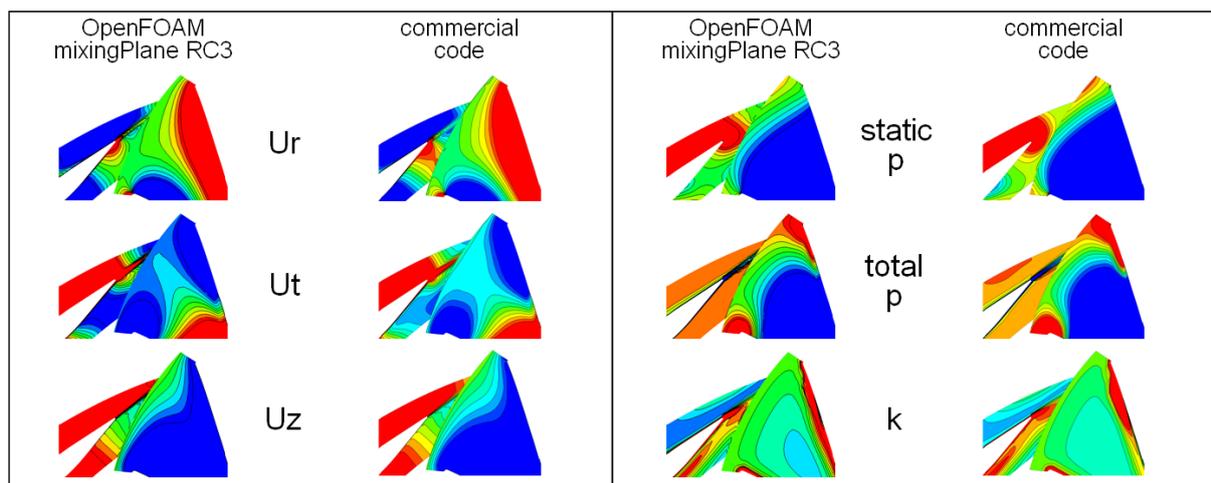


Figure 6. Field distributions at mid-height of the distributor and runner passages across the mixingPlane interface for the absolute radial, tangential and axial components of the velocity, for the static pressure, for the total pressure in the stationary frame, and for the turbulent kinetic energy. The same contour levels are used for both codes, except for the static and total pressure.

5.2. The Turbine-99 test case

The Turbine-99 workshops focused on the flow in a Kaplan turbine draft tube. The inlet boundary conditions were applied from the experimental data. Nilsson and Page [16] showed that the results using OpenFOAM were similar as those of all major CFD tools on the market, and also similar to the experimental data. Nilsson [17] made further OpenFOAM simulations of the Turbine-99 draft tube, and also of the guide vanes and runner. The runner domain included the axi-symmetric part of the draft tube, and the simulations were done in a rotating frame of reference. In particular, it was shown that the hub clearance flow is extremely important to delay a separation at the runner hub. At the Turbine-99 workshops it was discussed how important the unsteadiness due to the runner blade wakes are for the capturing of correct mean velocity profiles in the draft tube. Nilsson and Cervantes [18] performed a study using OpenFOAM and CFX, where different draft tube inlet boundary conditions were based on the runner results of Nilsson [17]. The inlet boundary conditions were based on the experimental data, the computed symmetric data, the computed angular resolved data with coarse resolution, and the computed angular resolved data with coarse resolution but with refined boundary layers. In the case of angular resolved inlet data, it was rotated with the runner rotational direction and speed. It was shown that the radial resolution of the boundary layers at the inlet of the draft tube is by far the most important feature of the inlet boundary condition. There were no differences in the time-averaged downstream flow between the results when using symmetric inlet data or angular resolved inlet data. This is very promising for the use of a circumferentially averaged coupling, such as that of a mixing plane.

Figure 7 shows the Turbine-99 guide vane, runner and draft tube geometry and numerical regions of the present work. It also shows the experimental cross-sections Ia and Ib, and the two mixingPlane interfaces that are used in the present work. Only the best efficiency operating condition is studied here.

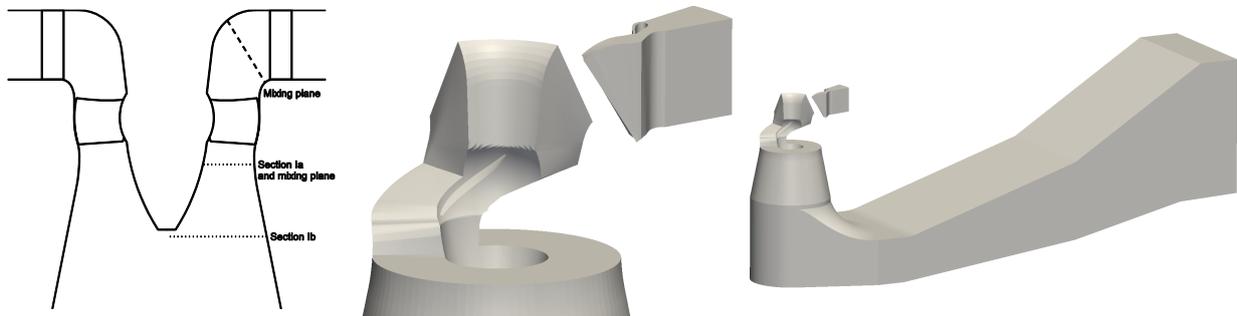


Figure 7. Turbine-99 guide vane, runner and draft tube geometry and numerical regions, with experimental cross-sections Ia and Ib, and two mixingPlane interfaces.

A constant inlet boundary condition is set for the velocity, from the flow rate, $Q=0.522\text{m}^3/\text{s}$, and the guide vane angle, 36.5 degrees from the tangential direction. The inlet turbulence, for the k-omega SST model, is specified using a turbulence intensity of 5%, and a turbulent viscosity ratio of $\nu_t/\nu=10$. No-slip conditions are used at all walls, and a solid-body rotation is applied on walls that are rotating with the runner rotational speed $N=595\text{RPM}$. A zero gradient outlet boundary condition is applied for the velocity and turbulent quantities, with a restriction ensuring the flow to be purely out of the computational domain. The static pressure is set to zero at the outlet.

The computational domain is divided in three regions, as seen in figure 7. The rotation of the runner region is handled using the steady-Multiple Reference Frames (MRF) methodology, where source terms for the rotation are added to the momentum equations. Two mixingPlane interfaces are applied at the connections between the rotating and stationary regions, one between the guide vanes and runner blades, and one between the runner blades and the draft tube. The mixingPlane discretization is using the *fluxAveragingAdjustMassFlow* scheme for the velocity, the

zeroGradientMatchAverageValue scheme for the pressure, and the *areaAveraging* scheme for the turbulent quantities. The mixingPlane ribbon patch discretization is done using the mesh distribution on both patches of the interfaces. This makes sure that the boundary layers are resolved to the highest possible level across the mixingPlane interface. A cyclic GGI is applied on the cyclic sides of the runner and guide vane regions, although the mesh is periodically conformal. The convective terms are discretized using the second-order linear-upwind scheme for the velocity, and the first-order upwind scheme for the turbulent quantities. The governing equations are discretized on a block-structured hexahedral mesh with 0.23M, 0.61M and 1.2M cells in the guide vane, runner and draft tube regions, respectively. The mesh is adapted for the low-Reynolds k- ω SST turbulence model, integrating the boundary layers to the wall and resolving the hub and tip clearances.

Figure 8 shows the predicted velocity distributions, normalized with the bulk velocity at the draft tube inlet, which is the same as the mixingPlane between the runner and draft tube regions. The profiles are predicted similarly as those by Nilsson [17] at cross-section Ia, although there are some differences in the prediction of the effects of the clearance flow. The profiles at cross-section Ib show qualitatively the same behaviour as the experimental profiles, but is not as good as the ones presented by Nilsson [17]. The discrepancy may be due to the mixingPlane interface, but also due to other modifications in OpenFOAM since 2006. It is also likely that the small discrepancies at section Ia, between the present clearance effects with those of Nilsson [17], may cause the large discrepancies at cross-section Ib. However, the actual cause needs to be investigated thoroughly before any final conclusions can be drawn.

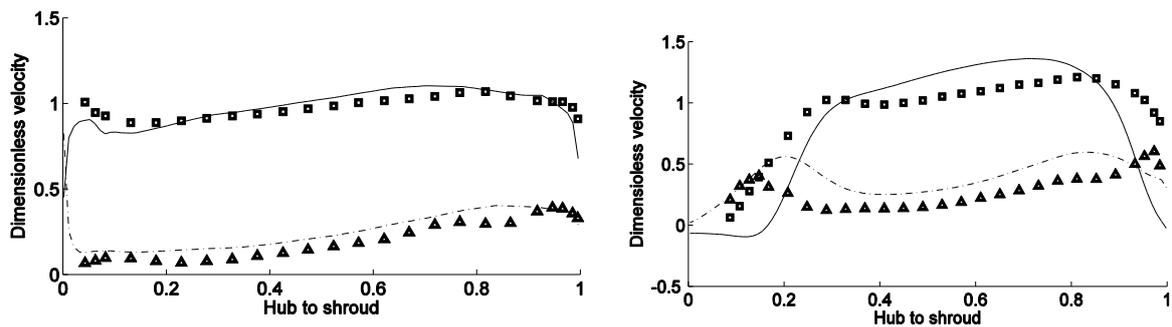


Figure 8. Velocity distributions at cross-sections Ia and Ib. Curves: Numerical. Markers: Experimental. Solid: Axial. Dash-dotted: Tangential. Squares: Axial. Triangles: Tangential.

Figure 9 shows the kinematic pressure distributions at the two interfaces of the Turbine-99 results. The currently employed mixingPlane discretization of the pressure acts as a zero gradient boundary condition at both sides, while preserving the pressure level. The pressure is thus not forced to have the same mean radial distribution on both sides.

Finally, figure 10 shows the velocity magnitude distributions at the mixingPlane interfaces of the Turbine-99 results. The effects of the guide vane and runner blade can be seen on the upstream side of the interface, while it is circumferentially averaged at the downstream side of the interface.

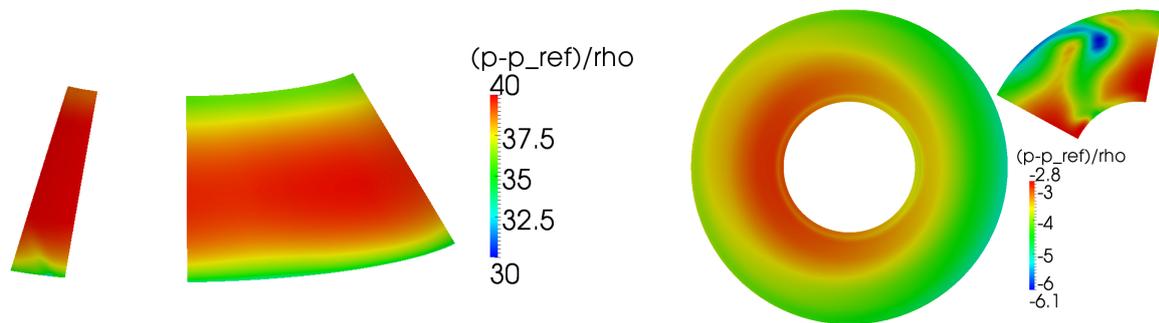


Figure 9. Kinematic pressure distributions at guide vane – runner mixingPlane interface (left) and runner – draft tube mixingPlane interface (right, top view). The reference pressure is that at the outlet, which is set to zero in the simulation.

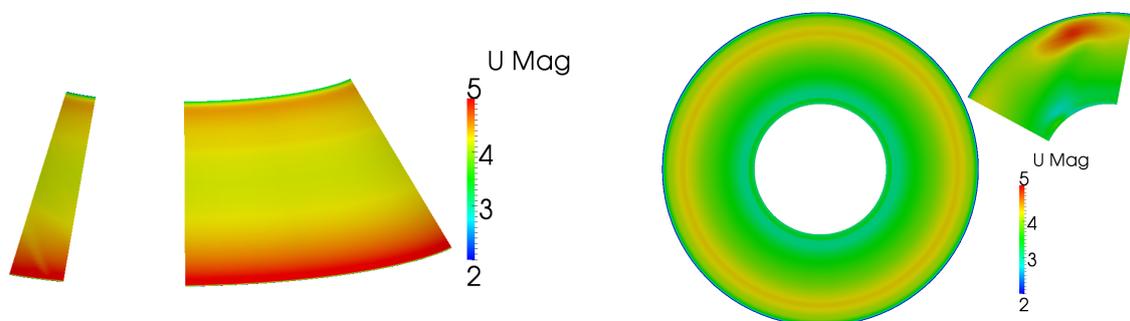


Figure 10. Velocity magnitude distributions at guide vane – runner mixingPlane interface (left) and runner – draft tube mixingPlane interface (right, top view).

6. Conclusion and future work

An improved version of the mixingPlane for the community-driven version of OpenFOAM was presented, including simulation results obtained from complex 3D turbomachinery cases. The RC3 version of the mixingPlane is introducing new schemes that address the mass conservation of the flow across the interface. Comparisons were made against simulation results obtained from a commercial solver and also against experimental data. Future work will target additional validation test cases in order to improve the overall performance of this interface. This will probably lead to the development of new mixingPlane schemes for the velocity, pressure and turbulent quantities.

Acknowledgements

Prof. Håkan Nilsson is funded by the “Swedish Hydropower Centre – SVC”. SVC is established by the Swedish Energy Agency, Elforsk and SvenskaKraftnät together with Luleå University of Technology, The Royal Institute of Technology, Chalmers University of Technology and Uppsala University, www.svc.nu. The computational facilities for the Turbine-99 simulation are provided by C3SE, the center for scientific and technical computing at Chalmers University of Technology, and SNIC, the Swedish National Infrastructure for Computing.

References

- [1] Weller H, Tabor G, Jasak H and Fureby C 1998 A numerical approach to continuum mechanics using object-oriented techniques *Computers in Physics*. **12(6)** 620-631
- [2] Chen G, Xiong Q, Morris P J and Paterson E G 2014 OpenFOAM for Computational Fluid Dynamics *Notices of the American Mathematical Society*. **61**, Issue 4, p 354
- [3] GNU General Public License, Version 3, 29 June 2007 *Free Software Foundation*

- [4] Beaudoin M and Gschaider B 2008 The OpenFOAM-extend project on SourceForge: current status *Third OpenFOAM Workshop (Politecnico di Milano, Milan, Italy, July 2008)*
- [5] Jasak H and Beaudoin M 2011 OpenFOAM Turbo Tools: From general purpose CFD to Turbomachinery simulation *Proceedings of the ASME-JSME-KSME Joint Fluids Engineering Conference (Hamamatsu, Shizuoka, Japan, July 24-29, 2011)*. Paper no. AJK2011-05015
- [6] Giroux A-M, Torriano F, Page M, Nicolle J, Morissette J-F, Labbé P and Beaudoin M 2013 An Overview of CFD-based Research Projects at Hydro-Québec *21st Annual Conference of the CFD Society of Canada (Sherbrooke, Canada, May 2013)*
- [7] Page M, Beaudoin M and Giroux, A-M 2011 Steady-state Capabilities for Hydroturbines with OpenFOAM *Int. J. of Fluid Machinery and Systems*. **4(1)** 160–170
- [8] Nilsson H, Page M, Beaudoin M, Gschaider B and Jasak H 2008 The OpenFOAM Turbomachinery working group, and conclusions from the Turbomachinery session of the third OpenFOAM workshop *IAHR 24th Symposium on Hydraulic Machinery and Systems (Foz do Iguassu, Parana, Brazil, October 27-31 2008)*
- [9] Petit O, Page M, Beaudoin M and Nilsson H 2009 The ERCOFTAC Centrifugal Pump OpenFOAM Case-Study *3rd IAHR International Meeting of the Workgroup of Cavitation and Dynamic Problems in Hydraulic Machinery and Systems (Brno, Czech Republic, October 2009)*. pp 523–532
- [10] Beaudoin M and Jasak H 2008 Development of a General Gid Interface for turbomachinery simulations with OpenFOAM *Open Source CFD International Conference (Berlin, Germany, November 2008)*
- [11] Beaudoin M 2013 DecomposeZoltan: Doing domain decomposition in parallel *8th International OpenFOAM Workshop (Jeju, South Korea, June 11-14 2013)*
- [12] Beaudoin M 2012 Community Day: SIG Turbomachinery – Inside the mixingPlane interface *7th OpenFOAM Workshop (Darmstadt, Germany 2012)*
- [13] Page M 2012 Community Day: SIG Turbomachinery – domAdomB testcases - Behaviour of the mixingPlane *7th OpenFOAM Workshop (Darmstadt, Germany 2012)*
- [14] Page M 2012 Community Day: SIG Turbomachinery – domAdomB testcases - Or various ways to compute and connect periodic flow passages *7th OpenFOAM Workshop (Darmstadt, Germany 2012)*
- [15] Beaudoin M 2012 Community Day: SIG Turbomachinery – The mixingPlane interface: A quick introduction *7th OpenFOAM Workshop (Darmstadt, Germany 2012)*
- [16] Nilsson A and Page M 2005 OpenFOAM simulation of the flow in the Hölleforsen draft tube model. Turbine-99 III workshop (*Porjus, Sweden, 2005*)
- [17] Nilsson H 2006 Evaluation of OpenFOAM for CFD of turbulent flow in water turbines *23rd IAHR Symposium on Hydraulic Machinery and Systems, (Yokohama, Japan, 2006)*
- [18] Nilsson H and Cervantes M 2012 Effects of inlet boundary conditions, on the computed flow in the Turbine-99 draft tube, using OpenFOAM and CFX. *26th IAHR Symposium on Hydraulic Machinery and Systems, (Beijing, China, 2012)*. *IOP Conf. Ser.: Earth Environ. Sci.* **15** 032002, *15 (PART 3) s. Article number 032002*