International Journal of Fluid Machinery and Systems
Vol. 4, No. 1, January-March 2011

DOI: 10.5293/IJFMS.2011.4.1.161
ISSN (Online): 1882-9554

Original Paper (Invited)

Steady-state Capabilities for Hydroturbines with OpenFOAM

Maryse Page, Martin Beaudoin and Anne-Marie Giroux

Hydro-Québec, Institut de recherche
1800 Lionel-Boulet, Varennes, Québec, Canada J3X 1S1
page.maryse@ireq.ca, beaudoin.martin@ireq.ca, giroux.anne-marie@ireq.ca

Abstract

The availability of a high quality open source CFD simulation platform like OpenFOAM offers new R&D opportunities by providing direct access to models and solver implementation details. Efforts have been made by Hydro-Québec to adapt OpenFOAM to hydroturbines for the development of steady-state capabilities. This paper describes the developments that have been made to implement new turbomachinery related capabilities: multiple frames of reference solver, domain coupling interfaces (GGI, cyclicGGI and mixing plane) and specialized boundary conditions. Practical use of the new turbomachinery capabilities are demonstrated for the analysis of a 195-MW Francis turbine.

Keywords: OpenFOAM, hydroturbine, GGI interface, mixing plane interface, multiple frames of reference

1. Introduction

This paper presents on-going developments for performing steady-state simulations of hydroturbines with OpenFOAM. Even though OpenFOAM offers an extensive suite of computational fluid dynamic (CFD) solvers and capabilities [1], some important components are still missing for turbomachinery applications. The objectives of this work are given in the Section 2, which describes the use of CFD at Hydro-Québec. A new long-term research program has been started which will integrate advanced (cavitation, air injection, etc) in order to analyze the hydraulic behaviour of hydroturbines for all operating conditions. OpenFOAM is a powerful simulation platform on which the required models can be added. Section 3 describes the developments that have been made to implement new turbomachinery related capabilities, namely multiple frames of reference (MFR) solver, domain coupling interfaces (GGI, cyclicGGI, and mixing plane) and specialized boundary conditions. Section 4 discusses the computational performance and the parallelization of the domain coupling interfaces. This work represents a first step in the building of our new simulation platform. To demonstrate the new capabilities, results are shown for predicting the flow field and the performances of a 195-MW Francis turbine for various operating conditions. On-site efficiency measurements are available at both nominal and low operating heads for comparison with numerical results.

2. Computational fluid dynamics at Hydro-Québec

For more than 20 years, Hydro-Québec has used numerical simulation technology to analyze existing equipments. At the Hydro-Québec's Research Institute, IREQ, computational fluid dynamics helps to improve the performances of old turbines [2] or optimize power plant operation [3]. CFD is also being used to study generator and transformer cooling systems [4][5][6]. Important benefits have already been reaped from those projects.

Since 2007, a new high-performance computing center is providing sizeable computational capacities and possibilities to CFD users. In 2008, in order to coordinate development efforts in numerical simulations applied to the analysis of the mechanical and hydraulic behaviour of hydroturbines, a new research program called SAMH was started [7]. This long-term program aims at developing numerical technology in order to:

1. analyze the hydraulic behaviour of turbines for all operating conditions;
2. simulate fluid-structure interactions for stable and transient operating regimes;
3. do multi-physics rotor dynamic analyses of hydro-generator units.

Computational fluid dynamics is at the heart of this research program. Our work is characterized by its application to existing equipments. In order to run CFD simulations to analyze old hydroturbines, we have to model runner geometries using state-of-the art 3D digitizing technology. Surfaces are often degraded by wear, corrosion and cavitation erosion over time, and geometric differences between one runner blade to another must sometimes be taken into account [8].

Accepted for publication December 29, 2010: Paper number O10061S
Corresponding author: Maryse Page, Researcher, page.maryse@ireq.ca

This manuscript was presented at the 25th IAHR Symposium on Hydraulic Machinery and Systems, September 20-24, 2010, Politechnica University of Timisoara, Romania .

The following is a list of the main challenges we will be facing to achieve the first objective of the SAMH research program :

- Cavitation modelling
- Air injection characterization and modelling
- Leakage flows
- Surface roughness characterization and modelling
- Free surface flows at intakes
- Trash rack modeling (with fouling characterization)

Our CFD simulation toolbox contains various software packages, including commercial tools, open source tools and in-house developments. Over the years, the commercial software packages we have been using have proven to be quite robust and reliable. However, with the high cost of running license fees, commercial software packages do not allow us to use all the computing resources now available with modern supercomputers. On the other hand, for open source software packages, even though there are no license running fees, they usually require important development, validation and maintenance efforts that are often underestimated. Still, open source software offers the possibility of integrating our own numerical models and of better understanding the inner workings of numerical algorithms. To complement our CFD simulation toolbox, in-house developments are also made as required.

3. OpenFOAM in hydroturbines

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. The well-designed object-oriented C++ architecture of OpenFOAM allows us to efficiently enhance the toolbox by focusing on specific libraries and by re-using existing capabilities. Thanks to its open design, OpenFOAM is a powerful simulation platform to which we can add the required models to better simulate complex flow behaviour through hydroturbines such as cavitation modelling, air injection modelling, or surface roughness modelling. For forecasting the performance of the entire power station, it is also important to model the free-surface flow and the trash rack modelling impact at the water intake.

The simulation of the steady-state multiple frames of reference (MFR) capabilities for hydroturbines with OpenFOAM represents a first step in building this new simulation platform for Hydro-Québec. OpenFOAM represents a new paradigm where code development and experience can be shared within a community of CFD experts. Hence, it is worth mentioning that the add-ons described in this paper will be contributed to the OpenFOAM community. The OpenFOAM Turbomachinery Working Group [9], co-chaired by Chalmers University and Hydro-Québec, has been active in contributing valuable enhancements for hydroturbine applications and in validating these developments. Steady-state and unsteady simulations in diffuser and draft tube flows with OpenFOAM, based on Reynolds-Averaged Navier-Stokes equations with standard *k*-ε turbulence models, have already been validated [10][11], leading to results similar to those obtained with commercial CFD codes. An unsteady turbomachinery computation is another area that has been extensively covered by the OpenFOAM Turbomachinery Working Group with the ERCOFTAC centrifugal pump case-study [12] and the Timisoara swirl generator case-study [13].

In this section, we will describe the developments that have been made to implement these new steady-state MFR capabilities, namely a new MFR solver, new domain coupling interfaces (GGI and mixingPlane) and some specialized boundary conditions.

3.1 Steady-state MFR solver

The OpenFOAM toolbox already provides a solver called simpleFoam for solving the steady-state Reynolds-Averaged Navier-Stokes equations with turbulence modelling, such as the standard *k*-ε model. The coupling between velocity and pressure is treated using the SIMPLE method [14]. The Turbine-99 draft tube case study has shown that OpenFOAM provides results comparable to those of commercial CFD codes [10]. The simpleFoam solver is discretized by the finite-volume technique of the Navier-Stokes conservation equations in the absolute frame :

$$\nabla \cdot (u_I \otimes \vec{u}_I) = -\nabla\left(p / \rho\right) + \nu \, \nabla \cdot \nabla \vec{u}_I \tag{1}$$

$$\nabla \cdot \vec{u}_I = 0 \tag{2}$$

To compute the steady-state flow through a turbine, it is important to take into account that the runner is in relative motion with respect to the distributor and the draft tube. To be able to compute simultaneously between these rotating and fixed components, a MFR solver must be used. The Navier-Stokes equations in the inertial frame are rewritten in the rotating reference frame. They lead to the classical formulation :

$$\nabla \cdot (\vec{u}_R \otimes \vec{u}_R) + 2\vec{\Omega} \times \vec{u}_R + \vec{\Omega} \times \vec{\Omega} \times \vec{r} = -\nabla\left(p / \rho\right) + \nu \, \nabla \cdot \nabla \vec{u}_R \tag{3}$$

$$div(\vec{u}_R) = 0 \tag{4}$$

These conservation equations can be rewritten such as the convected velocity is the velocity in the absolute frame $\vec{u}_I$ instead of the velocity in the rotating frame $\vec{u}_R$:

$$\nabla \cdot (\vec{u}_R \otimes \vec{u}_I) + \vec{\Omega} \times \vec{u}_I = -\nabla\left(p / \rho\right) + \nu \, \nabla \cdot \nabla \vec{u}_I \tag{5}$$

$$\nabla \cdot \vec{u}_I = 0 \tag{6}$$

The implementation of a new solver in OpenFOAM is quite straightforward since existing solver source code can be used as an example. A new solver called simpleTurboMFRFoam was developed from the simpleFoam solver. This new solver adds the contribution of the term $\vec{\Omega} \times \vec{u}_I$ in the momentum equation to the basic simpleFoam solver. It also computes the flux with the rotating velocity instead of the absolute velocity.

3.2 Coupling interfaces for hydroturbine simulations

For the CFD simulation of hydroturbines, separate 3D mesh passages or full geometry are generally connected together in order to simulate the flow of water through a succession of complex meshes. The requirement to fit all the meshes with conformal matching interfaces is often very difficult or leads to geometric compromises that would affect the numerical quality of the simulation results. Non-conformal meshes are thus generated separately for each part of the whole geometric model and joined using one or many specialized interfaces.

The Generalized Grid Interface (GGI) is a type of coupling interface used for joining multiple non-conformal regions where the patch nodes on each side of the interface do not match. Specialized versions of the GGI are also needed in order to simplify the computational domains and hence reduce the computer time needed to run the simulations. One such example is the cyclicGGI used for the simulation of periodic geometries. For unsteady hydroturbine simulations, the GGI interface is also appropriate because the relative rotation of the mesh parts produces non-conformal interfaces between the stationary and the moving sections.

Another important coupling interface for hydroturbine simulation is the mixing plane [15]. For steady-state simulations in hydroturbines, it is common to use a mixing plane interface between a guide vane passage and a runner passage, or between a runner passage and a draft tube. The flow through the interface is averaged circumferentially.

3.3 GGI and cyclicGGI interfaces

The basic Generalized Grid Interface or GGI is a recent addition to OpenFOAM [16]. This interface uses weighted interpolation to evaluate and transmit flow values across a pair of conformal or non-conformal coupled patches called master and shadow GGI patches. The basic GGI interface is similar to a case of a static sliding interface with the advantage that no re-meshing is required for the neighbouring cells of the interface.

Figure 1 (left side) illustrates the basic concept of a GGI interface where the flow values are transferred between two non-conformal domains using a master/shadow boundary patch configuration and weighted interpolation. Figure 1 (right side) illustrates the intersection areas between a shadow facet and its four neighbouring master facets. The robust detection of facets neighbours and the precise computation of the master and shadow facets intersection areas are at the core of the evaluation of the GGI weighting factors.

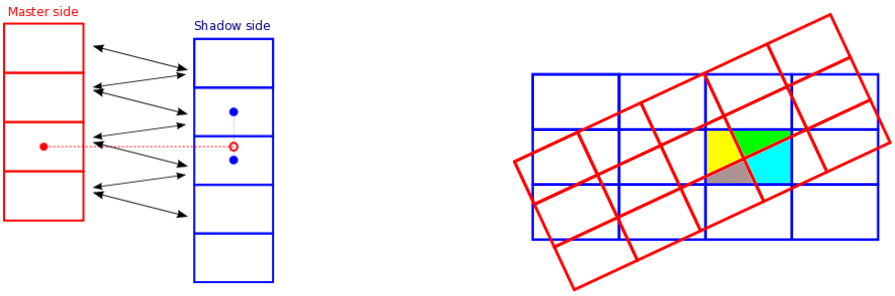

Fig. 1 Coupling two non-conformal patches using a GGI interface

The equations that control the flow values between the GGI master and shadow patches are derived from basic FVM discretization reasoning. They state that consistent and conservative discretization across the interface is achieved using weighted interpolation. Hence, flow values ϕ_S on the shadow patch can be represented as a summation of weighting factors and master patch variables :

$$\phi_{Si} = \sum_n W_{M_n\_to\_S_i} * \phi_{Mn} \tag{7}$$

And, reciprocally, flow values ϕ_M on the master patch can be represented as a summation of weighting factors and shadow patch variables:

$$\phi_{Mj} = \sum_m W_{S_m\_to\_M_j} * \phi_{Sm} \tag{8}$$

In order for the interface discretization to remain conservative, it is required that the weighting factors respect the following constraints:

$$\sum_n W_{M_n\_to\_S_i} = 1 \tag{9}$$

$$\sum_m W_{S_m\_to\_M_j} = 1 \tag{10}$$

$$W_{M_n\_to\_S_i} * \left|S_{M_n}\right| = W_{S_m\_to\_M_j} * \left|S_{S_m}\right| = \left|S_{\cap M\_to\_S}\right| \tag{11}$$

Also, it is required that

$$if\ W_{M_n\_to\_S_i}\ \exists \quad \Rightarrow \quad W_{S_i\_to\_M_n}\ \exists \tag{12}$$

It should be noted that in general,

$$W_{M_n\_to\_S_i} \neq W_{S_i\_to\_M_n} \tag{13}$$

Equations 12 and 13 simply state that if, for a given master patch facet, a master to shadow GGI weighting factor exists, then a corresponding shadow to master GGI weighting factor must exist as well for the neighbour shadow patch facet. However both factors do not necessarily have the same values, unless the two master and shadow facets have the same surface area.

Given two overlapping GGI master and shadow patch faces, the GGI weighting factors are computed using the ratio of the surface intersection between the faces, and with their respective surface area.

From eq. (11), we get, for the master to shadow patch faces the following expression for the weighting factors :

$$W_{M\_to\_S_i} = \frac{|S_{\cap_{M\_to\_S_i}}|}{|S_{M_n}|}, \quad with \quad W_{M\_to\_S_i} \in]0.0, 1.0] \tag{14}$$

And for the shadow to master patch faces:

$$W_{S\_to\_M_j} = \frac{|S_{\cap_{S\_to\_M_j}}|}{|S_{S_m}|}, \quad with \quad W_{S\_to\_M_j} \in]0.0, 1.0] \tag{15}$$

The determination of the number of neighbours for each facet and the computation of the GGI weighting factor values require robust and precise algorithms in order to make sure that the flow values are correctly distributed across the interface. A reliable weights calculation uses geometrical cutting to determine the facets intersection area. Hence the efficient Sutherland-Hodgman algorithm [17] is used as the cutting algorithm for evaluating facets surface intersection values. Some quick rejection algorithms have been implemented to speed up the search for potential face neighbours. These algorithms are based on an Axis Aligned Bounding Box (AABB) overlapping detection test, an efficient implementation of the Hormann-Agathos point-in-polygon algorithm [18], and the Separating Axis Theorem algorithm [19] that will handle the final non-overlapping filtering test. Finally, discretization effects are being taken into account in order to properly scale the GGI weighting factors to handle the possible presence of non-overlapping faces and hence keep the GGI interface conservative.

The cyclicGGI is a variation of the basic GGI developed for handling periodic non-conformal meshes. It uses the same internal algorithms as the basic GGI. The cyclicGGI adds an internal transform to be applied to the shadow patch data in order to internally superimpose them over the master patch data. This transformation is required in order to determine the cyclicGGI patch faces neighbourhood, to compute the cyclicGGI weighting factors, and also to transform any shadow patch vector or tensor field values before computing the GGI weighted interpolation across the interface. The cyclic transformation can be a 3D translation for translational-only periodic meshes, or a 3D rotation for rotational-only periodic meshes. Figure 2 illustrates a cyclicGGI interface using a rotational transformation.

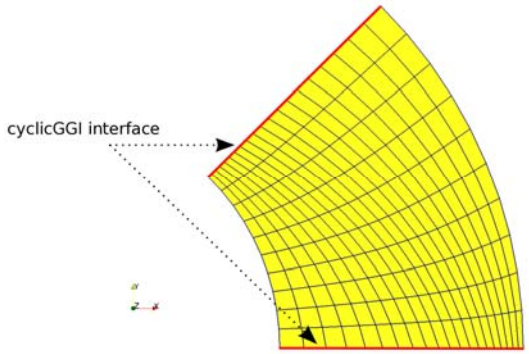

Fig. 2 Synthetic mesh using a CyclicGGI interface

3.4 MixingPlane interface

The mixingPlane interface is a circumferential averaging interface. Figure 3 is an illustration of a synthetic periodic mesh where the upstream (U) and downstream (D) parts of the mesh are coupled using a mixingPlane.

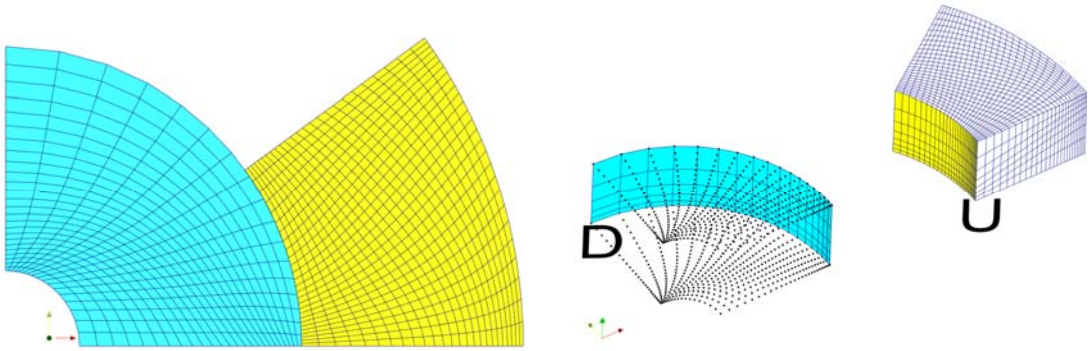

Fig. 3 Synthetic meshes of two flow passages connected by a mixingPlane interface

The main component of the mixingPlane interface in OpenFOAM is a circumferential averaging interpolator that computes and transfers average field values across the upstream and downstream patches. The interpolator is constructed using an intermediary cylindrical patch surface made of a stack of 360° ribbons shared between two specially crafted GGI interfaces. In Fig.4, we can see a synthetic example of a mixingPlane interface where an intermediary cylindrical patch surface, labelled G, was built from three 360° ribbons. That intermediary surface will be connected to the upstream patch labelled U and the downstream patch labelled D using two separate GGI interfaces. By sharing a common intermediary patch G, the two GGIs will serve as a coupling mechanism in order to transmit the flow values between the upstream and downstream patches. It is important to note that the upstream and downstream patches are usually defined with different spatial discretization. The GGI interface interpolation algorithms are specifically designed to handle such mesh topology. Furthermore, by using 360° ribbons for the intermediary patch, each ribbon will serve as a local flow values accumulator, a necessary component in order to compute averaged flow values. The GGI weighting factors will provide the necessary scaling factors for computing the final averaged upstream or downstream flow values. In order to simplify the computation of the GGI weighting factors, all the necessary GGI patch facets intersections are being computed in a cylindrical coordinates reference frame.

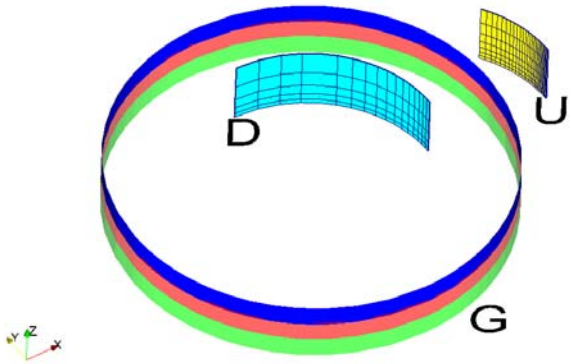

Fig. 4 Intermediary 360° ribbons patch of the mixingPlane interface

Special care has been taken for the discretization of the cylindrical ribbons patch in order 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 upstream and downstream patch point distribution in cylindrical space. This 2D profile will then serve as the discretization control curve for generating the intermediary 360° ribbons patch.

The current design of the mixingPlane requires a selection between a radial or axial discretization profile. The radial and axial discretization scheme is based on a simple leapfrog algorithm where the next profile discretization coordinate will be driven by the largest jump from either the upstream or downstream patch discretization. In order to model the upstream and downstream patches discretization, two histograms need to be built, based on the axial or radial distribution of the upstream and downstream patches coordinates mesh points evaluated from a cylindrical coordinates reference frame. This algorithm ensures that none of the upstream or downstream patches will be over sampled, and that the final discretization profile will provide a good compromise between the finer and coarser regions of both the upstream and downstream patches of the mixingPlane interface. Figure 5 illustrates the leapfrog algorithm being used to construct a radial discretization profile based on a pair of theoretical upstream and downstream patch coordinates histograms.

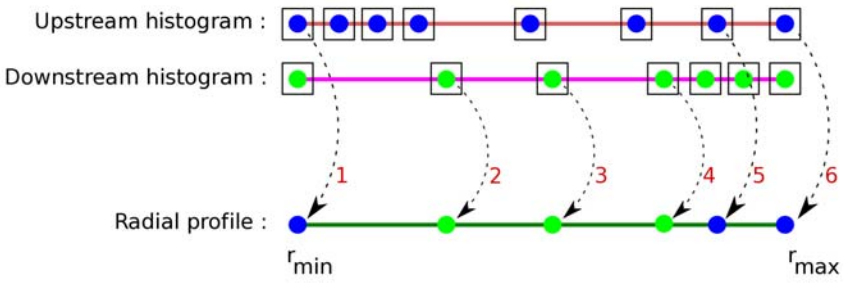

Fig. 5 Determination of a discretization profile for the mixingPlane intermediary ribbons patch

The choice between a radial or axial profile is obviously suboptimal when confronted to curved patch geometries where neither an axial or radial profile is perfectly suitable. Figure 6 illustrates such geometries where a mixingPlane is being used to couple the outlet of a runner passage to a full 360° draft tube inlet. In such a case, either an axial or radial discretization profile could be used as long as the normal of some of the mixingPlane patches facets do not get closely aligned with the chosen discretization direction, axial or radial. In such a case, the discretization profile might become too coarse on those regions. In order to better handle those curved patches geometries, a new algorithm based on a k-nearest neighbours (k-NN) technique will be developed.

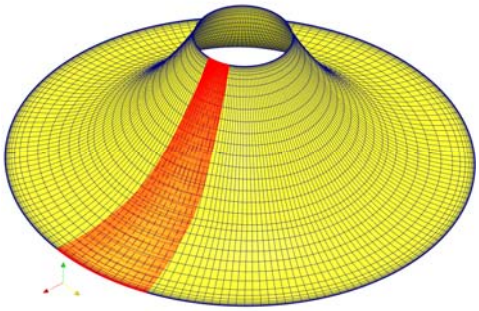

Fig. 6 Discretization of the mixingPlane interpolation patch: neither the axial or radial scheme is optimal for curved patches

It is important to note that the mixingPlane interface design presented here is totally bi-directional, meaning that no discrimination is being made between the upstream or the downstream flow values. In our current design, all of the flow values from a given mixingPlane patch are being averaged, and transmitted to its neighbour patch. We have found out that this simple design offers surprisingly interesting results. Further analysis and possible adjustments will be made in order to make sure that this design is sufficiently conservative in all situations.

3.5 Boundary conditions

In addition to the GGI and mixingPlane coupling interfaces, two new boundary conditions have been developed for the simulation of hydroturbines with OpenFOAM. The profile1DfixedValue boundary condition is used for applying a radial or axial profile of a specific field onto a mesh boundary patch, such as the velocity field at the inlet of a draft tube. The series of 1D profile values are provided through an ASCII file using a Comma Separated Values (CSV) file format. These profile values are interpolated along an axial or radial circular pattern at the boundary patch face centers. The turboWallFixedValue boundary condition is used for simulating a rotating wall at a mesh boundary patch. This boundary condition simply defines a rotation velocity in RPMs, as well as a type of reference frame, which is either stationary or rotating.

4. Computational performance and parallelization of the domain coupling interfaces

The computational performance of the GGI interface and of the mixingPlane interface is crucial when applied for the simulation of hydroturbine cases. The accurate modelling of those cases usually requires meshes of very large size. Hence, for both the GGI and mixingPlane interfaces, some quick internal algorithms where chosen or developed, and specific parallelization techniques were designed in order to run the simulations over high performance computing hardware.

4.1 Improving the performance of the GGI interface

To improve the performance of the GGI and cyclicGGI interfaces, the Axis Aligned Bounding Box overlapping detection test can be replaced by a multi-resolution search algorithm based on octrees. The internal quick reject test of the AABB algorithm is based on simple bounding box comparisons, which are very fast. However, the algorithm is also using a simple linear search for the detection of possible overlapping between GGI master and shadow facets. This simple search technique becomes very inefficient when large numbers of GGI facets are involved.

The more efficient octree search will recursively subdivide the search space into eight octants, until the final search space becomes small enough for switching to a linear search algorithm. In order to apply this technique for the search of GGI facet neighbours, the size of the octree was parameterized in order to keep the cost of building the octree relatively small, even for large meshes. Compared to the cost of the AABB algorithm, the new octree-based algorithm is bringing a constant tenfold speedup for the search of GGI facets neighbours. For steady-state computations, the evaluation of the GGI weighting factors is done only once, but for unsteady computations with moving meshes [12][13], the evaluation needs to be done at every mesh updates.

4.2 Parallelization of the GGI interface

An efficient parallelization of the GGI interface is also necessary for running parallel simulations over multiple processors. The actual implementation of the GGI is already parallelized. The GGI patch face information is shared globally among all the processors using a global zone so that every processor can compute and have access to its necessary GGI weighting factors. Still, sharing this information globally means that the flow values that need to be evaluated on the GGI interface patches also need to be shared globally among all the processors. This global sharing of information can severely limit the performance scaling of the overall parallel simulation.

The next generation of the GGI interface for OpenFOAM will provide a fully parallelized interface where only the processors sharing GGI facets will participate in the parallel exchange of information for the flow values going through the GGI interfaces. We expect that the removal of a global zone for sharing GGI information among all the processors will dramatically cut back on the number of communications and on the amount of informations that needs to be shared.

4.3 Parallelization of the mixingPlane interface

Contrary to the GGI interface, the mixingPlane interface is not fully parallelized in order to run large hydroturbine simulations on powerful computing equipment. Based on the experience obtained from the parallelization of the GGI interface, a different approach was explored for the mixingPlane in order to still be able to use it for large parallel simulations. The technique being used in this case is

to force the mesh cells on each side of the mixingPlane patches on a single processor or domain, hence eliminating the need of parallel communication for sharing the circumferentially averaged values from the mixingPlane interface between the processors.

A new automatic mesh partitioning tool was developed for OpenFOAM based on the Zoltan parallel partitioning library [20]. The Zoltan library aims at providing a number of tools that simplify the development and improve the performance of parallel, unstructured and adaptive applications. Most noticeably, this library offers a suite of dynamic load balancing and parallel repartitioning algorithms, including geometric, hypergraph and graph methods. It is also possible with the Zoltan library to force a list of specific cells onto specific processors or partitions, hence offering an interesting alternative for running the mixingPlane interface in parallel.

4.4 Selecting parallelization techniques for optimal performance

Even though the simple decomposition technique developed for the mixingPlane interface enables the simulation of large turbomachinery problems on a parallel computer, forcing mesh cells onto a single processor is not always beneficial for every type of interfaces. For instance, applying that decomposition strategy to a cyclicGGI interface often generates suboptimal or very large communication interfaces between the partition hosting the cyclicGGI interface and the rest of the mesh. So even though the evaluation of the flow values through the cyclicGGI interface is done rapidly by accessing to the local processor memory, the overall cost of communicating flow values with the rest of the mesh outweigh the benefice of running the cyclicGGI locally on one processor. The optimal strategy to efficiently run large cases with a mix of GGI and mixingPlane interfaces on a parallel computer will probably be a combination of the improved parallelization of the GGI interface while forcing other interfaces like the mixingPlane to run locally on one processor.

5. Hydroturbine results

The steady-state MFR capabilities described in Section 3 were used to simulate various operating conditions for a 195-MW Francis turbine. The main interest for this hydroturbine is that previous computations have been made with our commercial tools. Moreover, on-site efficiency measurements are available for various operating conditions. Apart from the on-site efficiency measurements performed by Hydro-Québec's test engineers at nominal operating head, data are also available at a lower operating head. Losses and efficiency predicted by CFD simulations can be compared to these valuable on-site data.

In this section, we present our CFD results computed with the simpleTurboMFRFoam solver, the cyclicGGI interface, the mixingPlane interface and the profile1DfixedValue boundary condition, as described in Section 3. Computations were facilitated by the fact that in-house tools were developed by our research team for pre-processing meshes to OpenFOAM (cgnsToFoam) and for post-processing solutions from OpenFOAM to the CGNS format (foamToCGNS).

5.1 Computational domain and methodology

The computational domain couples a distributor passage, a runner passage and a draft tube, as illustrated in Fig. 7. The meshes of each component were discretized using hexahedral elements. The ICEM-Hexa meshing tool from Ansys has been used for meshing the distributor passages (0.4 M elements) at various guide-vane opening angles and for meshing the draft tube (1 M elements). The Autogrid meshing tool from Numeca has been used to generate the runner passage (0.8 M elements). Close-up views of the meshes are shown in Fig. 8.

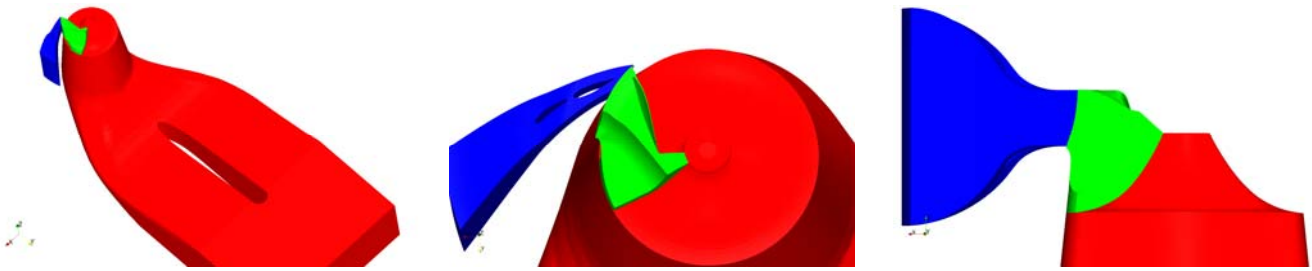

Fig. 7 Computational domain for MFR simulations

CyclicGGI interfaces are used to specify the rotational cyclic conditions on both sides of the distributor passage (1 of 20) and the runner passage (1 of 13). For both passages, the cyclicGGI interfaces connect conformal coupled patches. The distributor and the runner passages are connected by a mixing plane interface (Fig. 9, left) with interpolation ribbons defined in the axial direction. The runner passage and the draft tube are connected by another mixingPlane interface (Fig. 9, right) with interpolation ribbons defined along the radial direction.

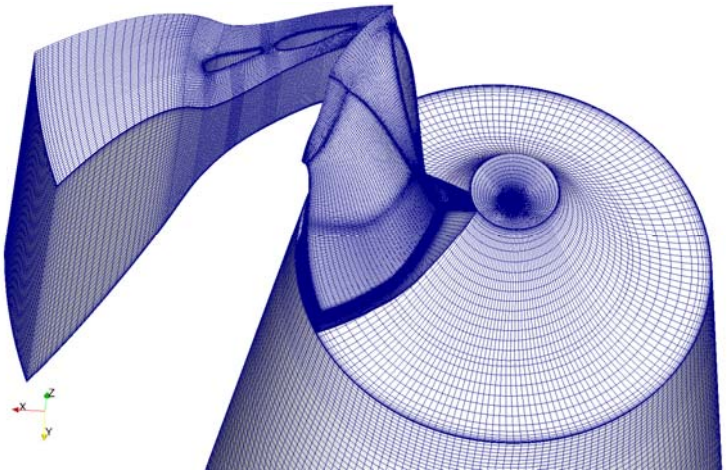

Fig. 8 Computational meshes for the distributor passage, the runner passage and the draft tube

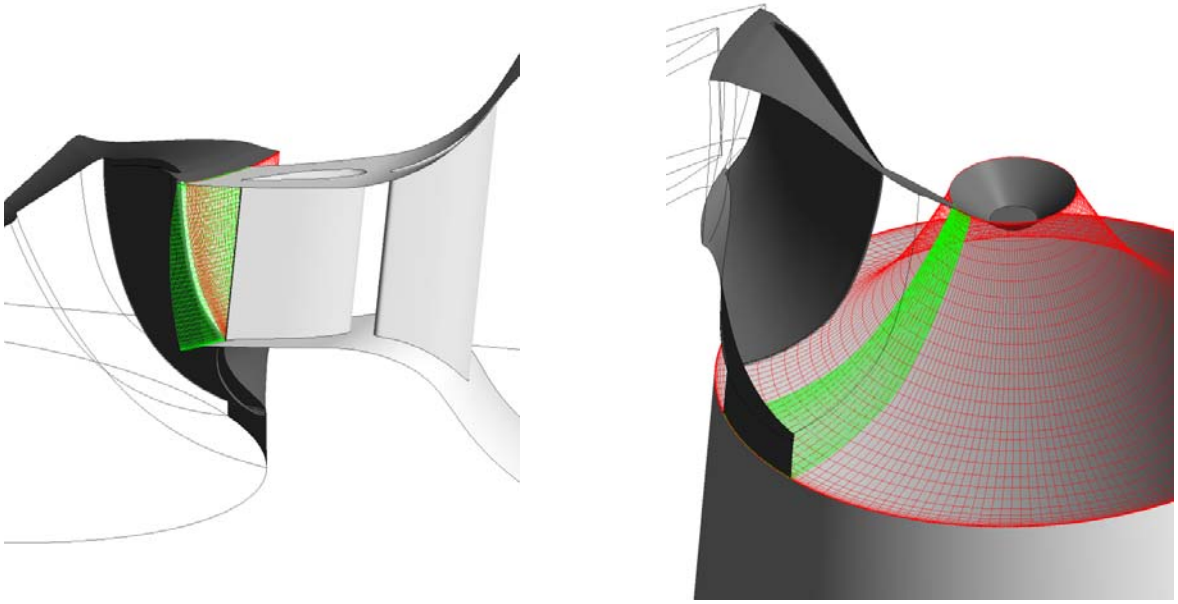

Fig. 9 Upstream (red) and downstream (green) patches of the mixingPlane interface between the distributor and the runner passages (left) and of the mixingPlane interface between the runner passage and the draft tube (right)

For the mixingPlane interface between the distributor and runner passages, 60 circumferential interpolation ribbons were automatically generated based on the upstream and downstream patches discretization. For the mixingPlane interface between the runner passage and the draft tube, 55 circumferential interpolation ribbons were automatically generated. Finer and coarser regions of both upstream and downstream patches were adequately discretized.

Figure 10 illustrates a close-up view of the domain decomposition generated by an OpenFOAM/Zoltan partitioning tool. The cells of the mixingPlane and the cyclicGGI interfaces were forced to reside on the same processor.

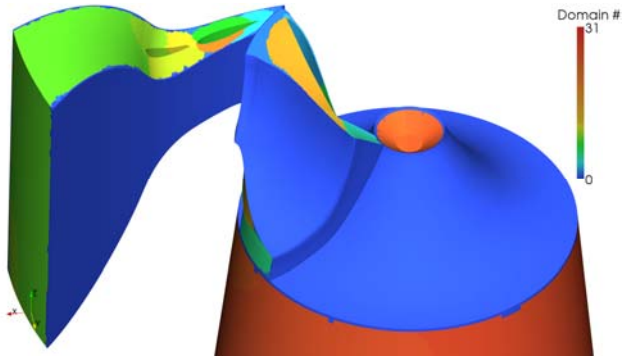

Fig. 10 Domain decomposition using the Zoltan parallel partitioning library

Computations were performed at various guide-vane opening angles by specifying the flow rate at the inlet of the distributor passage for both nominal and low heads. Losses were estimated by computing the mass flow averaged total pressure variation in each component.

The steady RANS equations, formulated in the multiple frames of reference, were solved. The standard k-ε model with log law near the wall was used. The scheme linearUpwind was selected for the discretization of the convection term of the momentum equation; it provides good convergence rate and adequate results for diffuser flows [9].

5.2 Flow field visualizations

Results are presented for three characteristic operating conditions for a Francis runner: part load, top efficiency and maximum output. Typical results are obtained with equivalent commercial codes computing the same steady-state MFR flow fields. In Fig. 11, pressure contours at distributor mid-height are plotted in the distributor-runner passages. In Fig. 12, pressure contours are plotted on the pressure and suction sides of the Francis runner blade. Minimum pressure peak near the leading edge and the band region is well predicted for this type of design.

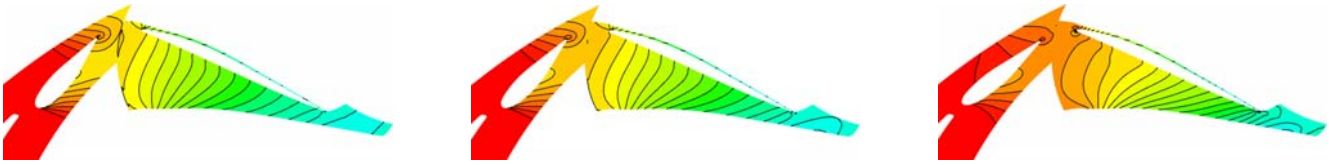

Fig. 11 Pressure contours at distributor mid-height in the distributor-runner passages for part-load (left), top efficiency (center) and maximum output (right) operating conditions

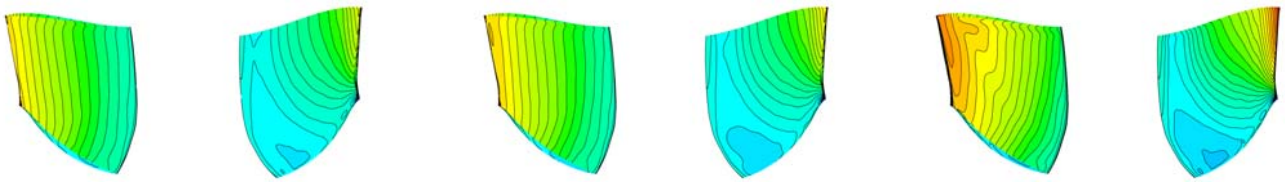

Fig. 12 Pressure contours for the pressure and suction sides on the blade for part-load (left), top efficiency (center) and maximum output (right) operating conditions

The flow field in the draft tube is also typical (Fig. 13), with a co-rotating flow vortex entering at the right side of the pier. At the maximum power operating condition, there is a counter-rotating vortex. Backflow is predicted at the back side of the elbow for the maximum efficiency and maximum output operating points (Fig. 14).

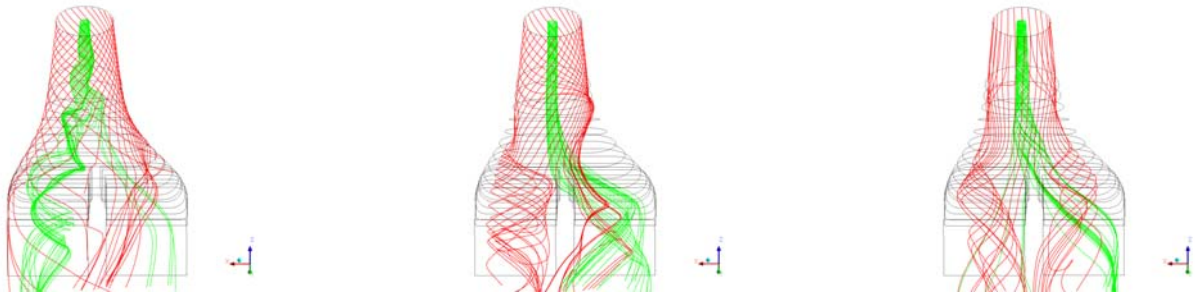

Fig. 13 Streaklines in the draft tube for part-load (left), top efficiency (center) and maximum output (right) operating conditions

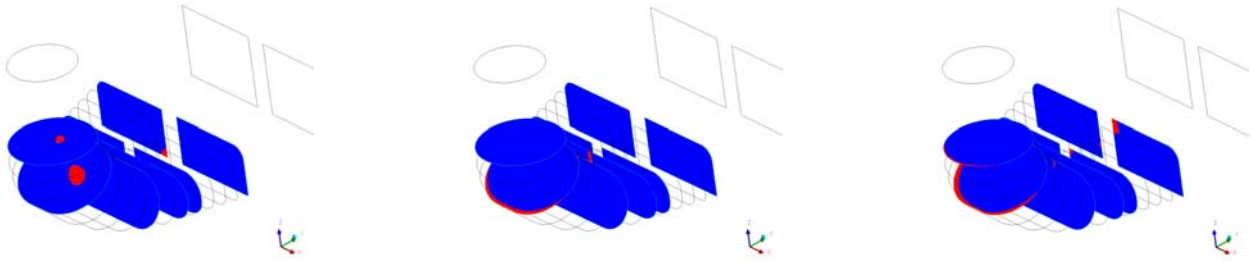

Fig. 14 Normal velocity at various sections in the draft tube for part-load (left), top efficiency (center) and maximum output (right) operating conditions. Red indicates a backflow in the section

5.3 Loss predictions

Losses were computed for the simulated operating points at nominal and low heads. Absolute loss predictions are not yet possible since not all flow phenomena are modelled. Figure 15 shows the efficiency curves measured on site and the relative efficiency calculated for eight operating conditions. It can be seen that trends are adequately predicted but there is still room for improvement. These are preliminary results and further analyses will be done to better evaluate losses. A closer look at the influence of numerical schemes and convergence criteria is also required.

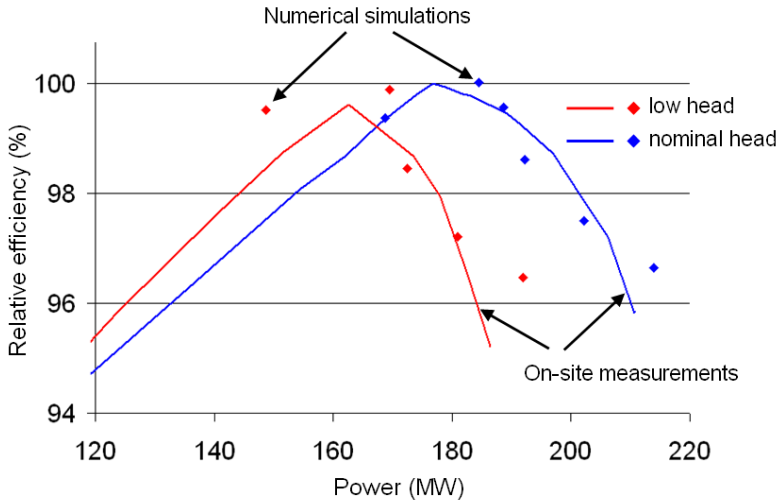

Fig. 15 Normalized numerical efficiency compared to on-site efficiency measurements

6. Conclusion

New capabilities have been added to the OpenFOAM toolbox to perform steady-state MFR simulations in hydroturbines, including a MFR solver, coupling interfaces (GGI, cyclicGGI and mixingPlane) and specialized boundary conditions. Simulation results have shown that the main characteristics of the flow fields in Francis turbine components are well predicted. Loss predictions were also compared against on-site efficiency measurements.

In the true sense of the open source spirit, the added capabilities we have developed will be contributed to the OpenFOAM community in order to stimulate cooperation and sharing of knowledge among CFD specialists. We expect this work to be enhanced through the activities of the OpenFOAM Turbomachinery Working Group.

Acknowledgments

The authors would like to thank Dr. Hrvoje Jasak, professor at the Faculty of Mechanical Engineering and Naval Architecture of the University of Zagreb, Croatia and director at Wikki Ltd. for his precious collaboration while developing the coupling interfaces for OpenFOAM. We would also like to thank Dr. Håkan Nilsson, associate professor at the Applied Mechanics department of Chalmers University of Technology in Göteborg, Sweden for his valuable discussions and contributions to the OpenFOAM Turbomachinery Working Group.

Nomenclature

p	Fluid pressure [Pa]	$W_{S\_to\_M}$	Shadow facet to master facets weighting factors
$\lvert S_M \rvert$	Surface area of master facet [m]	ν	Kinematic viscosity [m^2/s]
$\lvert S_S \rvert$	Surface area of shadow facet [m]	ρ	Fluid density [kg/m^3]
$\lvert S_{\cap M\_to\_S} \rvert$	Intersection surface area between master and shadow facets	$\vec{\Omega}$	Rotating frame speed [rad/s]
$\vec{u}_I$	Flow velocity in inertial frame (absolute) [m/s]	ϕ_M	Master patch variable
$\vec{u}_R$	Flow velocity in rotating frame [m/s]	ϕ_S	Shadow patch variable
$W_{M\_to\_S}$	Master facet to shadow facets weighting factors		

References

[1] Weller, H.G., Tabor, G., Jasak, H. and Fureby, C., 1998, "A numerical approach to continuum mechanics using object-oriented techniques," Computers in Physics, Vol. 12, No. 6, pp. 620-631.

[2] Giroux, A.-M., Houde, S., Laroche, Y. and Dubois, R., 2008, "Improving the performance of a 20 year-old Francis turbine using numerical simulations and robotized intervention," Proceedings of 24th IAHR Symposium on Hydraulic Machinery and Systems, Foz Do Iguassu, Brazil.

[3] Houde, S., Page, M., Trousseau, P. and Perri, A.T., 2008, "Evaluating the impact of a collector type surge chamber on the overall efficiency of a 5616 MW hydraulic powerplant through free surface RANS simulations," Proceedings of the 11th International Conference on Multiphase Flow in Industrial Plants, Palermo, Italy.

[4] Houde, S., Hudon, C. and Vincent, P.B., 2008, "Simulation strategies of the cooling flow for large hydro-generators," Proceedings of Hydro2008, Ljubljana, Slovenia.

[5] Torriano, F., Chaaban, M. and Picher, P., 2010, "Numerical study of parameters affecting the temperature distribution in a disc-type transformer winding," Applied Thermal Engineering, Vol. 30, No. 14-15, pp. 2034-2044.

[6] Picher, P., Torriano, F., Chaaban, M., Gravel, S. And Rajotte, B., 2010, "Optimization of transformer overload using advanced thermal modelling," CIGRE conference, Paris, A2-305.

[7] Giroux, A.-M., Magnan, R., Page, M., Labbé, P., Morissette, J.-F., Nicolle, J., Gauthier, G., Cupillard, S., Beaudoin, M., Bannari, R., Marcouiller, L., Merkhouf, A., Beauregard, S., 2010, "A new research program to study the hydraulic and mechanical behavior of existing turbines at Hydro-Québec," 16th International Seminar on Hydropower Plants, Vienna, Austria.

[8] Nicolle, J., Labbé, P., Gauthier, G. and Lussier, M., 2010, "Impact of blade geometry differences for CFD performance analysis of existing turbines," 25th IAHR Symposium on Hydraulic Machinery and Systems, IOP Conf. Ser.: Earth Environmental Science, 12, 012028.

[9] 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," 24th IAHR Symposium on Hydraulic Machinery and Systems, Foz Do Iguassu, Brazil.

[10] Nilsson, H and Page, M., 2005, "OpenFOAM simulation of the flow in the Holleforsen draft tube model," Turbine-99 – Workshop III, Porjus, Sweden.

[11] Muntean, S., Nilsson, H. and Susan-Resiga, R.F., 2009, "3D numerical analysis of the unsteady turbulent swirling flow in a conical diffuser using Fluent and OpenFOAM," 3rd IAHR International Meeting of the Workgroup on Cavitation and Dynamic Problems in Hydraulic Machinery and Systems, Brno, Czech Republic.

[12] Petit, O., Page, M., Beaudoin, M. and Nilsson, H., 2009, "The ERCOFTAC centrifugal pump OpenFOAM case-study," 3rd IAHR International Meeting of the Workgroup on Cavitation and Dynamic Problems in Hydraulic Machinery and Systems, Brno, Czech Republic.

[13] Petit, O., Bosioc, A.I., Nilsson, H., Muntean, S. and Susan-Resiga, R.F., 2010, "Unsteady Simulations of the Flow in a Swirl Generator, Using OpenFOAM," International Journal of Fluid Machinery and Systems, Vol. 4, No. 1 (Jan.-Mar.).

[14] Ferziger, J.H. and Peric, M., 1999, "Computational Methods for Fluid Dynamics," 2nd edition, Springer, Heidelberg, Germany.

[15] Keck, H. and Sick, M., 2008, "Thirty years of numerical flow simulation in hydraulic turbomachines," Acta Mech, 201, pp. 211-229.

[16] Beaudoin, M. and Jasak H., 2008, "Development of a Generalized Grid Interface for Turbomachinery simulations with OpenFOAM," Open Source CFD International Conference, Berlin, Germany.

[17] Sutherland, I.E. and Hodgman, G.W., 1974, "Reentrant polygon clipping," Commun. ACM 17 (1), pp. 32-42.

[18] Hormann, K. and Agathos, A., 2001, "The point in polygon problem for arbitrary polygons," Computational Geometry 20 (3), pp. 131-144.

[19] Gottschalk, S., 1996, "Separating axis theorem," Tech. Rep. TR96-024, Dept. of Computer Science, UNC Chapel Hill.

[20] Devine, K. Boman, E., Heaphy, R., Hendrickson, B. and Vaughan, C., 2002, "Zoltan Data Management Service for Parallel Dynamic Applications," Computing in Science and Engineering, Vol. 4, No. 2, pp. 90-97.