SPH Method used for Flow Predictions at a Turgo Impulse Turbine: Comparison with Fluent

Phoevos K. Koukouvinis, John S. Anagnostopoulos, Dimitris E. Papantonis

***Abstract*—**This work is an attempt to use the standard Smoothed Particle Hydrodynamics methodology for the simulation of the complex unsteady, free-surface flow in a rotating Turgo impulse water turbine. A comparison of two different geometries was conducted. The SPH method due to its mesh-less nature is capable of capturing the flow features appearing in the turbine, without diffusion at the water/air interface. Furthermore results are compared with a commercial CFD package (Fluent®) and the SPH algorithm proves to be capable of providing similar results, in much less time than the mesh based CFD program. A parametric study was also performed regarding the turbine inlet angle.

***Keywords*—**Smoothed Particle Hydrodynamics, Mesh-less methods, Impulse turbines, Turgo turbine.

I. Introduction

IMPULSE water turbines operation is based on the interaction of a high velocity water jet with the rotating turbine runner. The turbine runner changes the direction of the flow and torque develops on its blades due to the change of water jet's angular momentum. The static pressure does not change before and after the turbine runner. This enables the operation of the turbine in atmospheric environment, without the need of a sealed casing. The water jet is formed at the turbine nozzle, where the water hydraulic head is transformed into kinetic energy. An impulse turbine may have more than one nozzles, at the turbine casing around the turbine, directing the jet tangentially at the runner blades (Turgo turbine), or buckets (Pelton turbine). The nozzles are equipped with a needle, which is used to adjust the water flow rate through the nozzle to match the requested power at the generator coupled to the water turbine. Apart from the needle the nozzles often have a deflector, used to deflect the water jet from the runner, in case of an emergency to quickly reject load from the turbine. The most well known impulse turbines are the Pelton and Turgo turbines. Turgo turbines are designed for medium head applications. They have a flat efficiency curve and provide excellent part load efficiency, thus they can be used as an alternative of other turbine types, especially if there are large flow rate variations. The simplest Turgo turbine runner looks like a Pelton runner split in half at the plane of symmetry. The water jet enters from the one side of the runner and exits from the other (Fig. 1). Because of that, the escape of the water does not interfere with the incoming jet, or the other turbine blades, enabling the turbine to handle larger flow rates and jet diameters than a Pelton runner of the same runner diameter. As a result the Turgo turbine has higher specific speed and smaller size than a Pelton turbine of the same power. The smaller diameter allows the operation at higher angular velocities, which in turn, makes the coupling between the turbine and the generator easier, avoiding the use of a mechanical transmission system decreasing costs and increasing the mechanical reliability of the system [1].

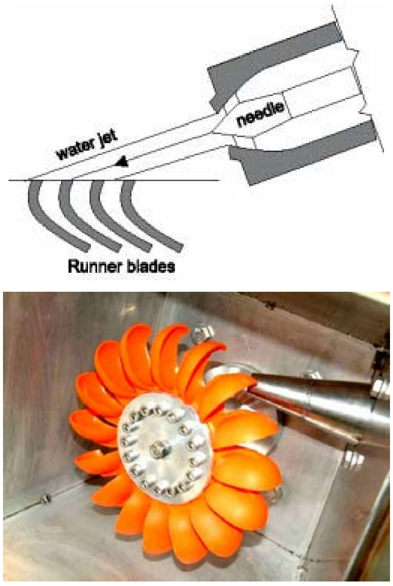

Fig. 1 Sketch of the flow in a Turgo turbine (up) and Turgo runner (down)

In impulse water turbines the developing flow of the impinging jet on the runner is unsteady, free-surface with moving boundaries, due to the runner rotation. Simulation with Eulerian methods is difficult [2], since special treatments are required for capturing the underlying phenomena. To be more specific, the treatments required are the Volume Of Fluid (VOF) method, combined with mesh refinement, for tracking the free-surface, and sliding meshes, for the connection between the moving and stationary meshes. The above treatments increase the computational cost and requirements of the simulation considerably.

An alternative way of simulating the flow would be by adopting a Lagrangian point of view, using and tracking particles which represent the water jet and interact with the turbine runner. Several attempts of using Lagrangian framework exist in literature, with Lagrangian particle tracking [3] or with a Moving Particle Semi-Implicit method [4]. A relatively new and promising method is the Smoothed Particle Hydrodynamics (SPH) method which will be used in the present work for the simulations.

The SPH method was initially developed by Lucy, Gingold and Monaghan (1977) and has been used for modeling

P. K. Koukouvinis, J. S. Anagnostopoulos and D.E. Papantonis are with the National Technical University of Athens, 15780, Heroon Polytechneiou 9 (e-mails: fivoskouk@gmail.com, j.anagno@fluid.mech.ntua.gr, papan@fluid.mech.ntua.gr, respectively).

astrophysical problems in three dimensional open space [5]. Today SPH is extended beyond its initial purpose of astrophysical phenomena, for modelling the behaviour of solids and fluids. The application of SPH to a wide range of scientific areas has led to significant extensions and improvements of the original method [6]. SPH is a Lagrangian, particle, mesh-less method and has the advantages of tracing and resolving the free-surface without any special treatment [7] and describing moving/deforming boundaries easily.

II. Standard SPH Formalism

The SPH formalism relies on the use of kernel approximation of field functions for the calculation of the operators appearing in the discretization of the flow equations, instead of using a computational grid. In this way it is able to approximate derivatives or functions from unconnected and randomly scattered computation points. The basis of the SPH approximations originates from the following identity:

$$f(x)=\int_{\Omega} f(x')\delta(x\text{-}x')dx' \qquad (1)$$

where $f(x)$ is a function of a three dimensional position vector **x**, $\delta(x\text{-}x')$ is the Dirac delta function and Ω is the volume of the integral that contains **x**. The above relation can be approximated using a smoothing kernel function $W(x\text{-}x',h)$:

$$\langle f(x)\rangle=\int_{\Omega} f(x')W(x-x',h)dx' \qquad (2)$$

A similar equation can be derived for the gradient of a function:

$$\langle\nabla f(x)\rangle=-\int_{\Omega} f(x')\cdot\nabla W(x-x',h)dx' \qquad (3)$$

In order the above approximations to be valid, the kernel function $W(x-x',h)$ has to fulfil certain requirements, such as:

- Unity or normalization condition : $\int_{\Omega} W(x-x',h)dx=1$
- Dirac function property : $\lim_{h\to 0} W(x-x',h)=\delta(x-x')$
- Compact condition : $W(x-x',h)=0$, for $|x-x'|>k\cdot h$, where $k.h$ is the kernel's support domain
- Also the kernel function has to be even, positive and monotonically decreasing function.

There are many types of kernel functions. In the present work the fourth–order kernel is used [8] for reasons which will be explained later.

In the kernel formulation (eq. 4) q=|r|/h, with |r| the distance between two computational points and h a characteristic smoothing length. Subscript d denotes the dimension of the problem and the constant term α is $\frac{1}{20\pi}$ for 3-D cases, in order to fulfil the normalization condition.

$$W(q)=\frac{a_d}{h^d}\begin{cases}\left(\frac{5}{2}-q\right)^4-5\left(\frac{3}{2}-q\right)^4+10\left(\frac{1}{2}-q\right)^4 & 0\le \mathrm{q}\le 0.5\\ \left(\frac{5}{2}-q\right)^4-5\left(\frac{3}{2}-q\right)^4 & 0.5\le \mathrm{q}\le 1.5\\ \left(\frac{5}{2}-q\right)^4 & 1.5\le \mathrm{q}\le 2.5\\ 0 & \mathrm{q}\le 2.5\end{cases} \qquad (4)$$

In the SPH method the entire system is represented with a finite number of particles that carry individual mass, occupy individual space and the characteristic quantities of the flow (e.g. velocity, density, pressure etc.). Thus the continuous integral relations can be written in the following form of discretized particle approximation:

$$\langle f(x_i)\rangle=\sum_{j=1}^{N}\frac{m_j}{\rho_j}f(x)W_{ij} \qquad (5)$$

$$\langle\nabla f(x_i)\rangle=\sum_{j=1}^{N}\frac{m_j}{\rho_j}f(x)\cdot\nabla W_{ij} \qquad (6)$$

In the above equations $W_{ij}=W(x_i-x_j,h)$, m_j is the j particle's mass and ρ_j is the j particle's density. Apart from the above relation for the derivative, there are the following alternative formulations [6], which tend to give better approximations than equation 6 [9]:

$$\langle\nabla f(x_i)\rangle=\frac{1}{\rho_i}\left[\sum_{j=1}^{N}m_j\left[f(x_j)-f(x_i)\right]\cdot\nabla W_{ij}\right], \qquad (7)$$

$$\langle\nabla f(x_i)\rangle=\rho_i\left[\sum_{j=1}^{N}m_j\left[\frac{f(x_j)}{\rho_j^2}+\frac{f(x_i)}{\rho_i^2}\right]\cdot\nabla W_{ij}\right] \qquad (8)$$

Considering the above procedure for the derivation of the SPH flow equations, one can notice two approximations. The first approximation has to do with the approximation of the Dirac delta with the kernel function W (equations 1, 2). As it was proved by Monaghan this interpolation is of, at least, second order of accuracy [10], due to the requirement of the kernel function being even. The second approximation has to do with the summation representation of the integral (equations 5, 6, 7, 8). It is proved [10] that, provided that the kernel function is smooth enough and particles are equi-spaced, the error of the summation approximation is nearly negligible. Particle disorder and kernel inconsistency tends to degrade the accuracy of the method. On the other hand, smooth kernel functions give better approximations and are less sensitive to particle disorder [6, 10]. This is the main reason for using the specific kernel (equation 4), since higher order functions are smoother than lower order ones.

Choice of the smoothing length h is also crucial for the SPH method, since it directly affects the support domain of the kernel approximation for each particle and the accuracy of the approximations. A small smoothing length would result in low accuracy due to very few particles in the support domain. On the other hand a large smoothing length would smooth out local properties. We have found that a smoothing length of

$\sqrt{3} \cdot dx$ is able to give satisfactory results for 3D simulations (*dx* is the inter-particle distance, or the particle discretization).

Using the above approximations for a function and the derivative of a function, one can derive the SPH flow equations [8]:

Continuity equation:

$$\frac{d\rho_i}{dt} = \sum_j m_j \mathbf{u}_{ij} \cdot \nabla_i \mathbf{W}_{ij} \tag{9}$$

Momentum equation:

$$\frac{d\mathbf{u}_i}{dt} = -\sum_j m_j \left[\left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla_i \mathbf{W}_{ij} - \Pi_{ij} \right] + g \tag{10}$$

where Π_{ij}, is a viscosity term suggested by Morris [11]. This term will be omitted in the rest simulations since they will be considered inviscid.

An obvious consequence of the above formulations is that SPH approximations are symmetric. This means that the contribution of particle *i* to *j* in the momentum equation is equal and opposite to the contribution of particle *j* to *i*. This enables SPH to conserve both linear and angular momentum of the system of particles under consideration.

III. Implementation Details

Pressure is calculated from an equation of state, thus the method is weakly compressible. The Tait equation of state is commonly used for modelling incompressible flows.

$$p_a = \frac{\rho_0 c_0^2}{\gamma} \left[\left(\frac{\rho_a}{\rho_0} \right)^\gamma - 1 \right] \tag{11}$$

In the above equation ρ_0 is the reference density and c_0 is an artificial speed of sound, since the real speed of sound would require a very small timestep. In order to keep density variations less than 1%, the value of c_0 is chosen ~$10 \cdot V_{\max}$, according to Monaghan [10]. SPH is known to produce unphysical pressure oscillations due to the stiffness of the equation of state [12]. In order to reduce the large pressure oscillations at the pressure field of particles, a density re-initialization technique [12] is adopted, and for moving the particles, the XSPH variant [6] is used, in which the velocities of the nearby particles are taken into account:

$$\frac{d\mathbf{r}_a}{dt} = \mathbf{u}_a - \varepsilon \sum_b \frac{m_b}{\rho_b} \mathbf{u}_{ab} W_{ab} \tag{12}$$

where *ε* is a parameter set at 0.3 for incompressible flows.

Since in the SPH method there is no connectivity between individual calculation points, a static matrix neighbor list [13] is used to find neighboring particles. For the time integration of the equations the fourth–order Runge–Kutta method is used. Also to speed up calculations and to utilize properly multi-core hardware, OpenMP is used for parallelization. Due to the nature of SPH, calculation of continuity and momentum terms can be done independently for each particle. Thus SPH algorithm is easily parallelized, achieving great speed – up.

IV. Boundary Conditions

There are various ways to define solid boundaries in SPH, such as fictitious particles, ghost particles or boundary particles ([6], [8], [10]). In the present work, the method of boundary forces will be used [14], [15]. The boundary is described with one layer of particles which exerts Lennard-Jones forces when fluid particles come close enough to the boundary.

$$\vec{\mathbf{f}} = D\left[\left(\frac{r_0}{\|r\|} \right)^{12} - \left(\frac{r_0}{\|r\|} \right)^{6} \right] \cdot \frac{\vec{\mathbf{r}}}{\|r\|^2} \quad \left[m/s^2 \right] \tag{13}$$

$$\text{for } {r_0}/{\|r\|} \le 1$$

In the above formula r_0 represents the range of influence of the Lennard-Jones forces and is considered to be equal to the fluid particle spacing *dx*. Also $\|r\|$ represents the inter-particle distance and *D* is a problem dependent parameter, usually set in the same scale as the maximum velocity in the simulation [6]. Apart from the boundary force, the boundary particles do not interact with the fluid particles in any other way.

The bucket geometry is represented using layers of force particles that exert the boundary forces prescribed by the above formula. The boundary particles on the bucket are positioned at a closer inter-particle distance than the fluid particles in order to ensure no penetration of the boundary and also to make the boundary force as smooth as possible. This is due to the overlapping influence areas of the boundary particles which, for fine boundary resolution, approximate the actual boundary surface.

The force that the fluid exerts on the solid boundary will be calculated from the reaction of the boundary forces on the wall. Since the forces during particle interactions are symmetric, the total force on a wall can be calculated by summing the reaction force on all wall particles. Furthermore, torque on each boundary particle can be calculated from the definition: $\vec{T}_i = \vec{r}_i \times \vec{F}_i$.

V.Geometry Description

The Turgo runner geometry was created using specialized software (Tools for Turbomachinery – T4T [16]), which uses as input the blade angles at each edge of the blade and the distance between the edges. The geometry generator forms a quadrilateral surface grid of points which represents the surface of the blade, using NURBS (fig. 2). Each quadrilateral element is filled up with boundary particles with inter-particle distance equal to *dx/3* (where *dx* is the inter-particle distance for fluid particles). Apart from filling the grid with particles, the algorithm extrudes the surface grid in the normal direction, at a prescribed depth, in order to form a full 3D body, if required.

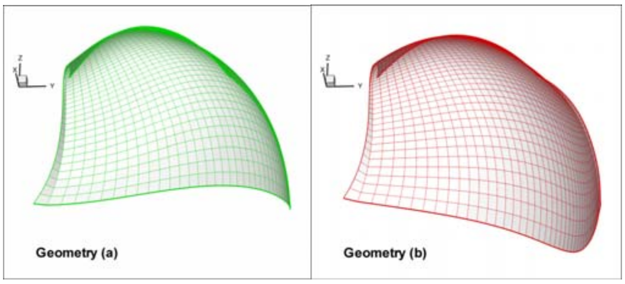

Fig. 2 Two indicative Turgo blade geometries created with the software T4T

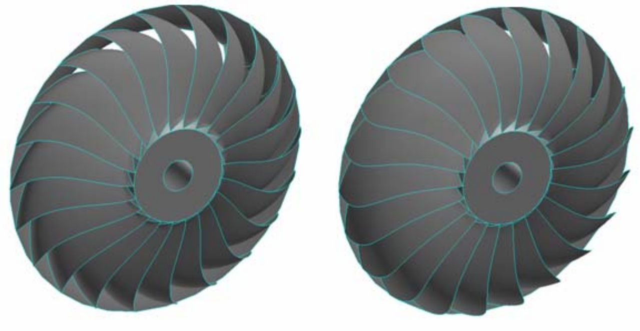

Fig. 3 Turgo runner geometries for the above blade geometries (viewed from the outlet)

VI. Case Set- Up

At first two different geometries were tested and compared with the results of Fluent in order to determine the performance and the accuracy of the SPH method. The two geometries modeled are those presented in fig. 2. The two geometries differ both at the inlet and the edges and, consequently, at the respective angles. In order to keep simulation cost at a minimum, only the space between two turbine blades was modeled, assuming periodic flow conditions. Also, since the geometry used in Fluent had no wall thickness, the blade was modeled with only a wall layer with the SPH method.

The runner hub was at 260mm radius and the tip at 500mm radius from the center of rotation. The nominal diameter of the runner was at 770mm. Also the runner consisted of 22 blades. The water jet had a diameter of 154mm and velocity 32m/s (discharge 596m^3/s) and entered the runner at 25° degrees from the runner rotation plane. The rotational speed of the runner was set to 44.8rad/s. Initially the two blades were positioned at -32° from the Y-axis (fig. 5), just before the impact of the water jet on the runner. The simulation duration was 42ms, in which the runner performs a rotation of 107°, enough to cover the whole interaction between the jet and the runner.

The particle size used in SPH was 5mm, which proved fine enough after a particle dependence study [15]. The numerical speed of sound was set to 350m/s, larger than 10.V$_{jet}$ in order to ensure the incompressibility of the simulated water. The computational mesh used in the Fluent program had a discretization size of 7.5mm, but refined to 5mm in areas of interest, such as the wall where pressure gradients appear, or the interface, consisting of ~500000 elements totally. Also the mesh consisted of two different areas (one rotating and one static), connected with an interface region. The VOF method was used to track the interfaces and second order discretization was used, to limit numerical viscosity.

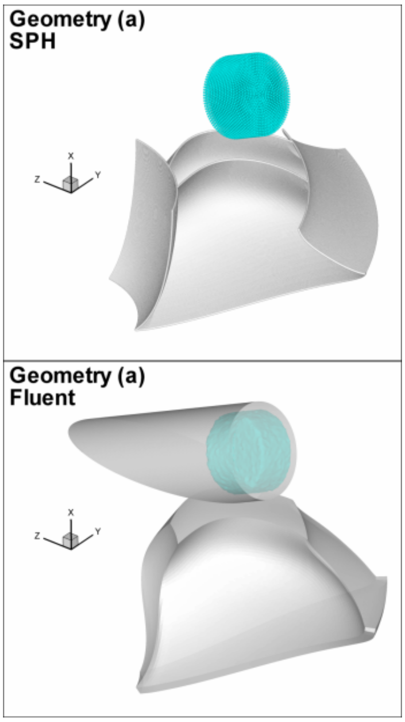

Fig. 4 The initial set – up for the simulated cases using SPH and Fluent

In the following figures indicative results of the simulations are shown. Figures 5 and 6 show a general view of the free surface flow developing between the blades at a time instance at the middle of the water jet and turbine blade interaction. By comparing the respective figures one can notice some differences, especially at the area of jet impingement, but generally both the velocity distribution and the free surface shapes are similar. As it is expected the velocity at the turbine runner outlet is much lower comparing to the water jet velocity, due to the energy transfer of kinetic energy from the jet to the turbine runner. Another important remark, regarding the behavior of the flow, is that a portion of the water jet leaks out from the inlet edge of the turbine blade (fig. 5, 6). This detail could not be effectively captured with Fluent, since the mesh did not extend beyond the blade (fig. 4).

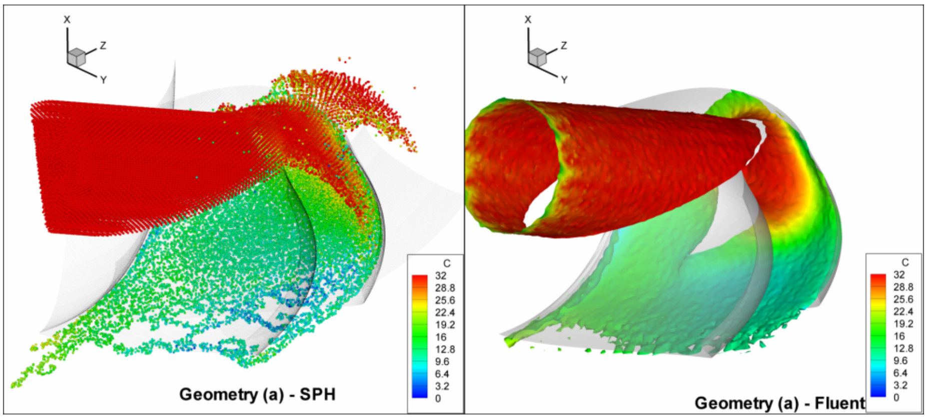

Fig. 5 Simulation results for geometry (a) using SPH (left) and Fluent (right), at time 21ms. SPH: Fluid particles are coloured according to velocity magnitude, wall particles are translucent to allow visualization of the flow between blades. Fluent: Isosurface is colored according to velocity magnitude.

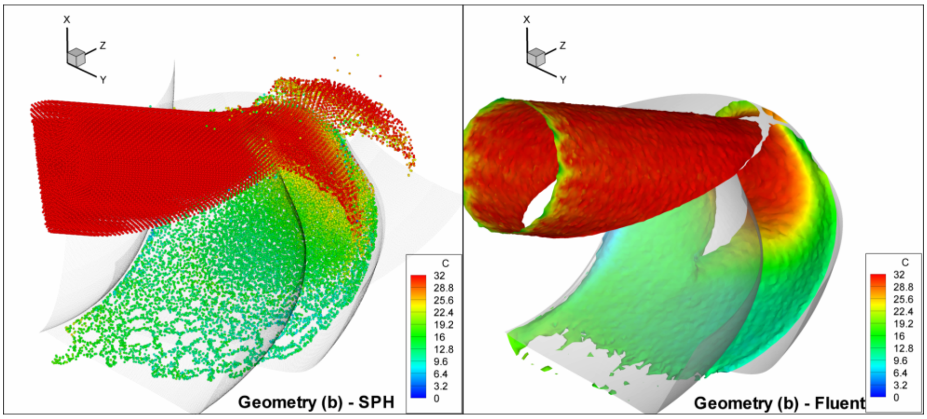

Fig. 6 Simulation results for geometry (b) using SPH (left) and Fluent (right), at time 21ms. SPH: Fluid particles are coloured according to velocity magnitude, wall particles are translucent to allow visualization of the flow between blades. Fluent: Isosurface is colored according to velocity magnitude.

In figures 7 and 8 the area of the blade, for geometry (b), covered by water is shown at two different time instances. Again results by both programs are very close. Similar results are obtained for geometry (a) too. The sharpness of the SPH results is attributed to the coarseness of the interpolation grid used to obtain the respective data, note though that this mesh was not involved in the flow field calculations.

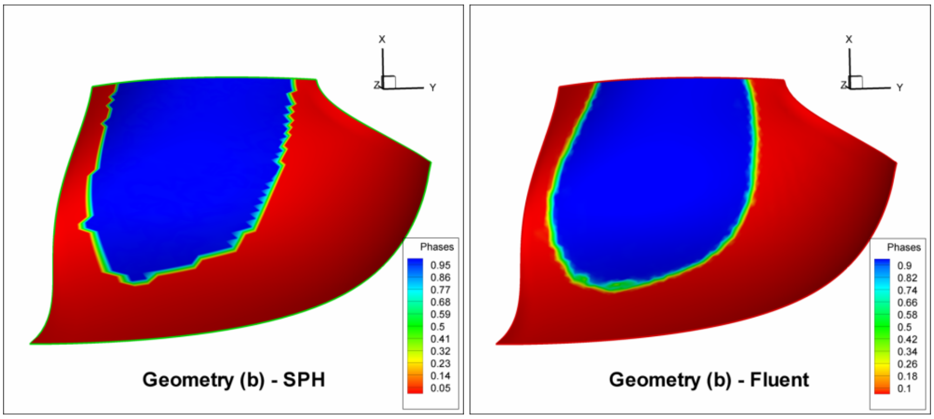

Fig. 7 Free surface evolution on the wall using SPH and Fluent, at time 12.6ms. Water is represented with blue, air with red

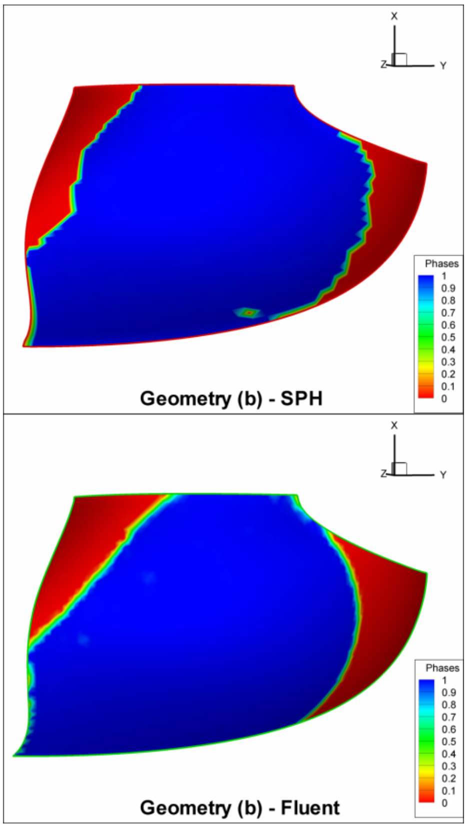

Fig. 8 Free surface evolution on the wall using SPH and Fluent, at time 21ms. Water is represented with blue, air with red

In figure 9 the developing torque (in respect to the angle of the blade from the Y-axis) on the turbine blade is shown for the two geometries tested with both programs. The torque calculated using SPH exhibits some oscillations, but the general trends are similar. For each geometry, the work of the torque, calculated by SPH and Fluent, on the blade is approximately the same (~0.5% difference). Also by comparing the work of the torque for the two geometries, it was found that blade (b) performs better, approximately by ~4%. Torque on geometry (a) develops earlier due to the shape of the inlet edge; the inlet edge in this case is more curved and cuts the water jet earlier.

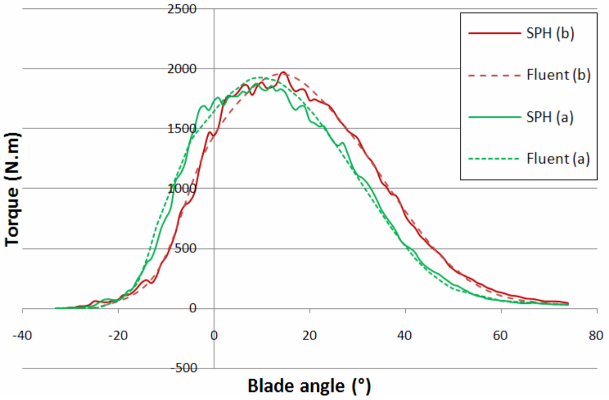

Fig. 9 Torque developed on the blade surface for the two geometries

Finally in fig. 10 the average torque distribution on the turbine blade is shown for the two geometries tested. In both cases maximum torque develops in curved areas where the direction of the flow changes. Geometry (b) has a smoother torque distribution at a larger area due to its design.

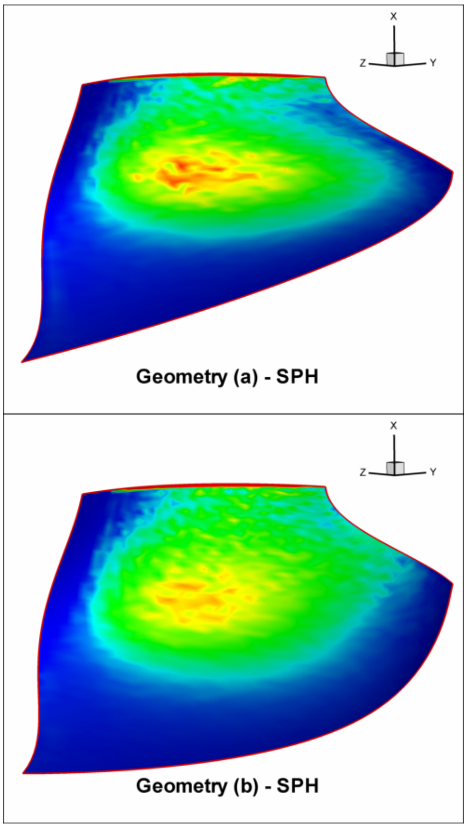

Fig. 10 Average torque distribution through the water jet – bucket interaction

Considering the time needed for the execution of the two programs, the SPH method is much faster than fluent, due to the 'embarrassingly parallel' nature of the SPH algorithm. The SPH algorithm only needed 10hrs on a 2xQuad – Core Xeon 2.Ghz computer (80 CPUhrs), for each simulation. On the other hand Fluent needed ~10days using 4 parallel processes on a i7 2.97Ghz computer (960 CPUhrs).

VII. Inlet Angle Dependence

From the previous results it has been shown that the geometry (b) performs better than geometry (a). For that reason geometry (b) was used to perform a further design test regarding the water jet inlet angle. Simulation conditions will be the same with those mentioned at part 6 of the current paper, apart from the jet inlet angle. The inlet angles which were tested are 20°, 30°, 35° and 40°.

In figure 11 a comparison between the torque graphs for the different water jet impingement angles is made; from this figure it is shown that, by increasing the water jet impact angle, the developing torque curve becomes narrower and exhibits a higher peak value. This is further illustrated in figure 12, where the average torque distribution is shown; again for larger impact angle the peak value is higher and the peak torque value is moved towards the center of the blade. By calculating the integral of the torque curve it was found that the maximum efficiency is achieved for the intermediate inlet angle of 30°, as shown in figure 11 in the respective graph of normalized efficiency (efficiency is normalized by the maximum value, i.e. the efficiency of the turbine for 30° inlet angle). A similar trend has been observed with our in-house Fast Lagrangian Simulation algorithm [1], used to estimate the efficiency for the same conditions.

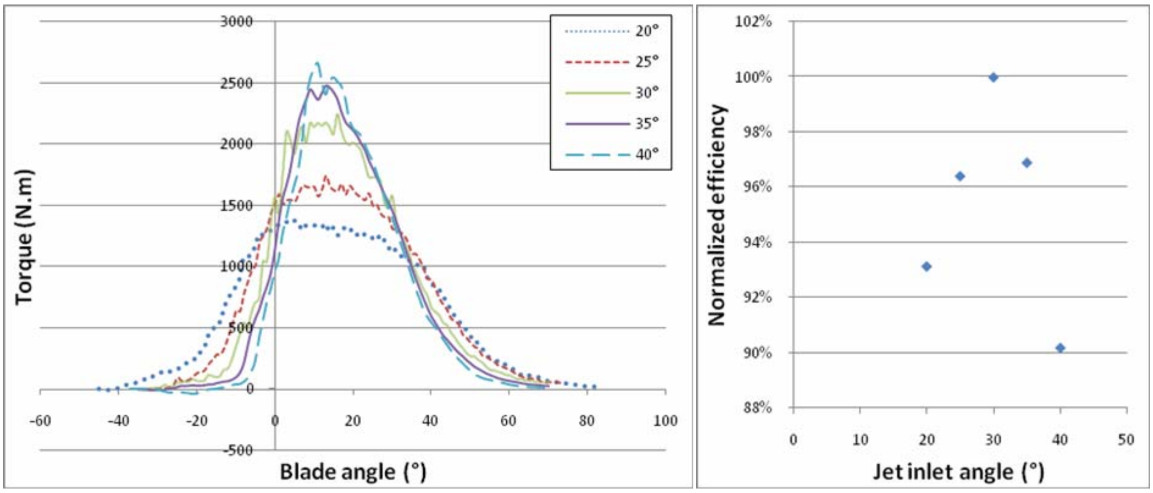

Fig. 11 Comparative torque (left) and normalized efficiency (right) graphs depending on the water jet inlet angle

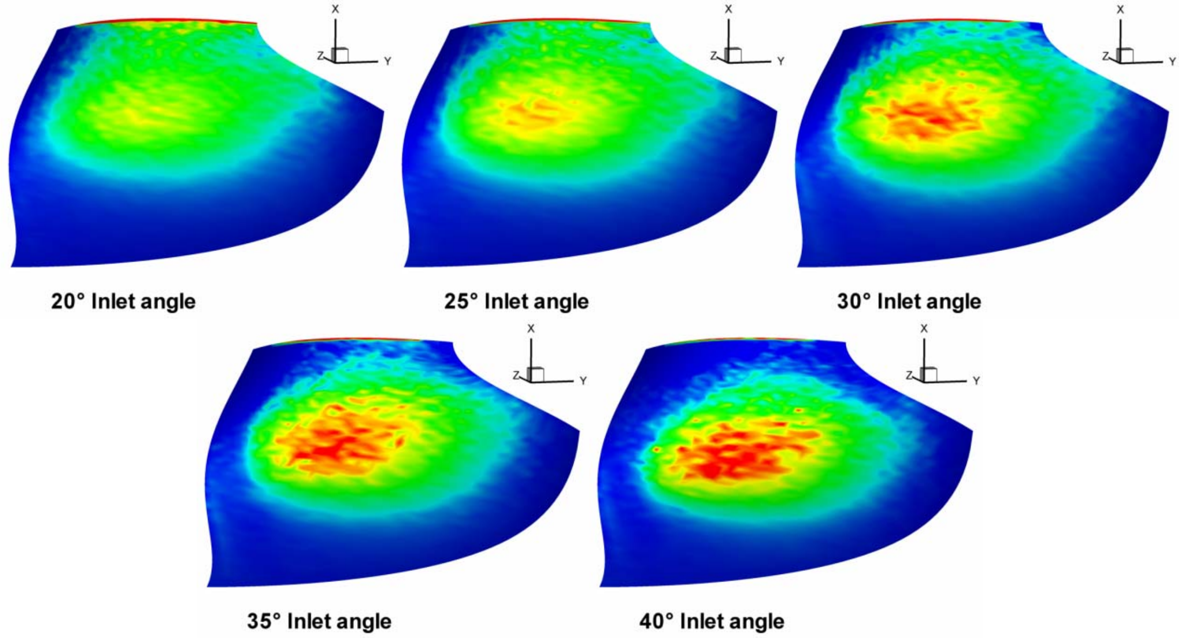

Fig. 12 Comparative torque distribution on the turbine blade. Red corresponds to higher torque values, blue to lower

VIII. Conclusion

In this work the SPH method was used to assess the performance of two different Turgo turbine runners, in comparison to a commercial CFD solver. The SPH method proves to be able to produce similar results in much less time than the mesh based program. For this reason it is an attractive alternative for parametric studies or even design optimization in the complex flows appearing in Turgo and other impulse turbines.

An important advantage of SPH algorithm is that its inherent structure permits the easy parallelisation and adaption to run in GPU technology, which can dramatically enhance execution speed. This work is currently underway in our laboratory.

REFERENCES

[1] J.S. Anagnostopoulos, D.E. Papantonis, “Flow modeling and runner design optimization in Turgo water turbines”, Interantional Journal of Applied Science, Engineering and Technology, Vol. 4 Number 3 2007 ISSN 1307-4318

[2] J.C. Marongiu, F. Leboef, E. Parkinson (2007), “Numerical simulation of the flow in a Pelton turbine using the meshless method smoothed particle hydrodynamics: a new simple boundary treatment”, Journal of Power and Energy, Vol. 221, Part A, p. 849-856.

[3] J. S. Anagnostopoulos, D. E. Papantonis (2006), “A numerical methodology for design optimization of Pelton turbine runners”, Proceedings of the Hydro2006 conference, Porto Carras, Cyprus, 25-27 September 2006.

[4] Y. Nakanishi, T. Fujii, M. Morinaka, K. Wachi (2006), “Numerical simulation of the flow in a Pelton bucket by a particle method”, Proceedings of the 23rd IAHR Symposium, Yokohama, October 2006.

[5] Price D.J., (2004), “Magnetic Fields in Astrophysics”, PhD thesis, University of Cambridge, UK.

[6] G.R. Liu, M. B. Liu (2003), “Smoothed particle hydrodynamics: a meshfree particle method”, World Scientific Publishing Company, December 2003.

[7] P. Koukouvinis, J. Anagnostopoulos, D. Papantonis (2009) “Flow modelling in the injector of a Pelton turbine”, Proceedings of the 4th Spheric workshop, Nantes, France.

[8] D. Violeau, R. Issa (2006), “Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview”, International Journal of Numerical Methods for Fluids.

[9] F. Colin, R. Egli, F.Y. Lin (2005), “Computing a null divergence velocity field using smoothed particle hydrodynamics”, Journal of Computational Physics.

[10] J. Monaghan (2005), “Smoothed Particle Hydrodynamics”, Reports on Progress in Physics, Volume 68 No.8, p. 1703-1759.

[11] J. Morris, P. Fox, Y. Zhu (1997), “Modelling low Reynolds number incompressible flows using SPH”, Journal of Computational Physics, Volume 136, p. 214-226.

[12] M. Gomez-Gesteira, B. Rogers, R. Dalrymple, A. Crespo, M. Narayanaswamy (2010), Users guide for the Sphysics code v2.0, http://www.sphysics.org.

[13] J. M. Dominiguez, A. Crespo (2009), “Improvements on SPH neighbour list”, Proceedings of the 4th Spheric workshop, Nantes, France.

[14] J. Monaghan, J. Kajtar (2009), “SPH boundary forces”, Proceedings of the 4th Spheric workshop, Nantes, France.

[15] P. Koukouvinis, J. Anagnostopoulos, D. Papantonis (2010) “Flow modelling in a Turgo turbine using SPH”, Proceedings of the 5th Spheric workshop, Manchester, United Kingdom.

[16] G. Koini, S. Sarakinos, I. Nikolos, (2009) “A software tool for parametric design of turbomachinery blades”, Advances in Engineering Software, p.41-51, DOI : 10.1016/j.advengsoft.2008.03.008