GPU-accelerated Pelton turbine simulation using finite volume particle method coupled with linear eddy viscosity models

S Alimirzazadeh1*, T Kumashiro2, S Leguizamón1, A Maertens4, E Jahanbakhsh1,3, K Tani2, F Avellan1

1 Laboratory for Hydraulic Machines, École polytechnique fédérale de Lausanne, Lausanne, Switzerland
2 Hydraulic R&D Laboratory, Hitachi Mitsubishi Hydro Corporation, Hitachi, Japan
3 Institute of Computational Science, Università della Svizzera italiana, Lugano, Switzerland
4 siaam.alimirzazadeh@epfl.ch

Abstract. The numerical investigation of the unsteady flow patterns around a Pelton bucket can be helpful to improve the overall turbine efficiency by optimizing the bucket design based on identified loss mechanisms. Since the flow is highly turbulent, modeling the effect of turbulence can bring about improved predictions. In this paper, two RANS-based eddy viscosity models (namely the standard and realizable k-ε) have been implemented as a module in a particle-based in-house solver, GPU-SHPEROS. A scalable wall function based on the log-law has been utilized to model the flow in the near-wall region. The solver has been accelerated on GPUs and is based on the Finite Volume Particle Method (FVPM), which is a locally conservative and consistent particle-based method including many of the attractive features of both particle-based methods (e.g. SPH) and conventional mesh-based methods (e.g. FVM). As a mesh-free method based on the Arbitrary Lagrangian Eulerian (ALE) formulation, FVPM is robust in handling free surface flows and large boundary deformations, such as the ones found in rotating Pelton buckets. The validation of the turbulence models implementation within FVPM is presented for internal and free surface flows. Then, the effectiveness of the turbulence models in the case of rotating Pelton buckets is assessed by comparing the predicted torque time histories to experimental data acquired on a model-scale test rig.

1. Introduction

The Pelton turbine is an impulse turbine invented to extract energy from the momentum of high-speed water jets. The Pelton turbines are suitable for high head and low flow rate hydropower plants. The kinetic energy of the rotating runner is converted to electricity. However, despite more than 130 years of Pelton turbines use, there is still room for improvement [1]. The fluid flow inside the Pelton bucket is a transient flow featuring free surface and moving boundaries and a particle-based method is appropriate to simulate such physics.

FVPM is a mesh-free particle-based method, which is both locally conservative and consistent. FVPM includes many of the attractive features of both particle-based methods (e.g. SPH) [2] and conventional mesh-based Finite Volume Methods (FVM) [3]. FVPM relies on particle interaction vectors to weight conservative fluxes exchanged between particles. These vectors are equivalent to the intercell area vectors in mesh-based FVM [4],[5]. In 2014, Quinlan et al. [6] presented an exact method for computing FVPM interaction vectors in 2-D. This method was then extended to 3-D by Jahanbakhsh et al. [4],[5] and implemented into the SPHEROS code to simulate the interaction between fluids, solid and silt. SPHEROS has been parallelized featuring the MPI library, efficient algorithms, and Adaptive Domain Decomposition (ADD) for load balancing, such that it can be executed on massively parallel machines.

* Corresponding author.
[7],[8]. However, computing the particle interaction vectors is expensive and usually has to be performed at each time step. To further speed up the software and perform realistic and large simulations, the unique architecture of Graphics Processing Units (GPUs) is taken advantage of. The solver is almost 6 times faster on NVIDIA® Tesla™ P100 GPU than a dual-setup CPU node equipped with two Intel® Xeon® E5-2690 v4 Broadwell CPUs. The reader is referred to [9] for more details on the computational performance and parallel algorithms featured by GPU-SPHEROS.

Turbulence is a flow regime, which is characterized by chaotic changes in pressure and flow velocity. Generally, unsteady vortices – containing a wide range of scales – appear in turbulent flows. Turbulent flows are challenging to simulate due to unsteady aperiodic motions and random spatial variations in the flow field. Direct Numerical Simulation (DNS) is a method to solve time-dependent Navier-Stokes equations resolving the full spatial and temporal scales of the turbulence. Although there is no need for any turbulence model in DNS methods, the computational cost of these methods is prohibitive. There are alternative methods to extract time-averaged and large-scale quantities at a reasonable computational time based on the turbulence modeling. Reynolds-Averaged Navier-Stokes (RANS) methods and large eddy simulation (LES) are the main alternatives, which can respectively give the time-averaged mean value and the small-scales filtered eddies solution based on filtering eddies. In the present research, two RANS eddy viscosity models, namely standard and realizable k-ε, have been implemented and coupled to Arbitrary Lagrangian Eulerian (ALE) FVPM in order to simulate turbulent flows. These models have been widely used and validated for both internal and free surface flows by different researchers [10],[11].

In the next sections, the FVPM discretized equations with the standard and realizable k-ε models are presented. Then the validation test cases for both internal and free surface flows (flow inside a pipe and open channel flow) are shown. The result of a Pelton turbine simulation using FVPM coupled with the aforementioned eddy viscosity models is shown in the final part.

2. Governing equations

The mass and momentum conservation equations can be written in the following conservative form:

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0$$

(1)

where \( U \) and \( F \) represent the conserved variables and the flux functions, respectively. For fluid flow, these variables are:

$$U = \left( \begin{array}{c} \rho \\ \rho C \end{array} \right)$$

(2)

and,

$$F = \left( \begin{array}{c} \rho C \\ \rho C \otimes C - s + pI \end{array} \right)$$

(3)

where \( \rho \) is the density, \( C \) is the fluid velocity vector, \( s \) the deviatoric stress tensor, \( p \) the pressure, and \( I \) is the identity tensor. The pressure is based on the Tait equation of state:

$$p = \rho_0 a \gamma \left( \frac{\rho}{\rho_0} \right)^\gamma - 1$$

(4)

where \( a \) is the speed of sound, \( \rho_0 \) is the reference density and \( \gamma \) is a constant coefficient which here is set to 7. In weakly compressible flow simulations, the speed of sound \( a \) is considered at least ten times greater than the maximum fluid velocity to reduce the computational cost [4],[5].
2.1. Finite Volume Particle Method

FVPM can be interpreted as a generalization of conventional mesh-based FVM. In FVM, the computational domain is partitioned into finite control volumes with defined surfaces. The area vector of the surfaces is used as a weight for the flux exchanged between the control volumes. In FVPM, control volumes are replaced by overlapping particles and the exchange occurs through the interfaces defined by overlapping regions. For each pair \((i,j)\) of overlapping particles (referred to as neighbors), two interaction vectors \(\Gamma_{ij}\) and \(\Gamma_{ji}\) are defined and their difference \(\Delta \Gamma_{ij} = \Gamma_{ij} - \Gamma_{ji}\) is analogous to the area vector in FVM.

Readers are referred to [4] and [5] for more details.

Being a mesh-free method, FVPM is primarily intended for problems where mesh-based methods may fail or have difficulties, such as moving or free boundaries, or fluid-structure interaction (FSI) problems.

The discretized FVPM formulation for conservation laws (1) for particle \(i\), reads [4]:

\[
\frac{d}{dt}(U_i V_i) = \sum_j \left( U_j \otimes \dot{x}_j - F_j \right) \cdot A_{ij} + \left( U_b \otimes \dot{x}_b - F_b \right) \cdot B_i
\]

and,

\[
\frac{dV_i}{dt} = \sum_j \dot{x}_j \cdot A_{ij} + \dot{x}_b \cdot B_i
\]

\[
\dot{x}_j = \left( \dot{x}_j \cdot \Gamma_{ij} - \dot{x}_i \cdot \Gamma_{ji} \right) \frac{A_j}{A_{ij} \cdot A_j}
\]

Boundary term \(B_i\) is computed as:

\[
B_i = -\sum_j A_{ij}
\]

where \(U_i\) is the conserved variable of the \(i^{th}\) particle, \(V_i\) is its volume, \(U_j\) and \(F_j\) are respectively the conserved variable and flux function at the interface of particles \(i\) and \(j\), whereas \(\dot{x}_j\) is the velocity at which the interface moves. Similarly, \(U_b\), \(F_b\) and \(\dot{x}_b\) are the conserved variable, flux function and particle velocity at the boundary. \(B_i\) and \(A_{ij}\) are the vectors which weight exchanged fluxes with the boundary and between particles, respectively. These vectors are computed from interaction vectors \(\Gamma_{ij}\) for each pair of neighbor particles. See [4] for the detailed derivation of these formulae.

Computing the interaction vectors requires that the supporting border of \(i^{th}\) particle, \(\partial \Omega_i\), be partitioned into sub-surfaces defined by a unique set of intersecting neighbors. Each elementary sub-surface \(e\) is characterized by \(\sigma_e\), the number of neighbors covering it. Then, the interaction vector \(\Gamma_{ij}\) is computed exactly as [4],[5],[6]:

\[
\Gamma_{ji} = \sum_{\sigma \in \{\Omega_j \cap \Omega_i\}} \mathbf{S}_e \left( \frac{1}{\sigma_j + 1} - \frac{1}{\sigma_j} \right)
\]

where \(\mathbf{S}_e\) denotes the area vector of the elementary surface \(e\) and \(\Omega_i\) denotes the supporting volume of \(i^{th}\) particle. Elementary surfaces can have complex or even disjointed shapes, making partitioning challenging. The detailed algorithm can be found in [5].

2.2. Boundary conditions

Regarding the boundary conditions, all the employed approaches to enforce boundary conditions have been presented in details in [4]. For no-slip wall boundaries, a layer of boundary particles whose motion is fixed by the wall boundary dynamics is overlaid. Their density is initially set to the fluid reference density. For the inlet, the same approach is used to update the mass and the volume of the boundary particles. The inlet boundary particles move with the prescribed velocity and are fed to the system as soon...
as they cross the inlet border. In this case, new boundary particles replace the fed ones. For the free surface, no boundary particles are used but the boundary fluxes are set to zero.

3. Turbulence modeling

3.1. Reynolds-Averaged Navier–Stokes

The idea behind the Reynolds-Averaged Navier–Stokes (RANS) is Reynolds decomposition, where the instantaneous velocity $\mathbf{C}(x,t)$ is decomposed to time-averaged $\bar{\mathbf{C}}(x)$ and fluctuating quantities $\mathbf{C}'(x,t)$ [12]:

$$\mathbf{C}(x,t) = \bar{\mathbf{C}}(x) + \mathbf{C}'(x,t)$$ (10)

By substituting the decomposed velocity in the momentum equation, a new term $-\rho \mathbf{C}_a \mathbf{C}_\beta$ appears which is called the Reynolds stress. The Reynolds stress tensor can be defined based on equation (11), which is called Boussinesq approximation. This approximation relates the Reynolds stress tensor to the velocity gradients through the eddy viscosity $\mu_t$, which is assumed isotropic:

$$\tau_{turb} = -\rho \bar{\mathbf{C}}_a \mathbf{C}_\beta = 2\mu_t \mathbf{S}_{\alpha\beta} - \frac{2}{3} \rho k \delta_{\alpha\beta} \quad (\alpha, \beta = 1, 2, 3)$$ (11)

where $\tau_{turb}$ is the Reynolds stress and $\mathbf{S}_{\alpha\beta}$ is the strain rate tensors. $\delta_{\alpha\beta}$ is the Kronecker delta and $k$ is the turbulence kinetic energy. $k$ can be written as a function of velocity fluctuations:

$$k = \frac{1}{2} \left( c_x'^2 + c_y'^2 + c_z'^2 \right) = \frac{3}{2} c_x'^2$$ (12)

In equation (12), $c_x$, $c_y$ and $c_z$ are the three velocity components. The Boussinesq approximation is reliable in a wide range of simulations. However, in some physics such as predicting flows with sudden and abrupt changes in the strain of the averaged flow and swirling flows [13],[14] as well as in the cases which the velocity fluctuations are highly-anisotropic, the approximation is error-prone. This approximation allows to model turbulence using eddy viscosity models instead of solving the system of equations for 6 unknown Reynolds stress terms.

3.2. Standard k-ε model coupled with FVPM

The standard k-ε is the most common model to simulate mean flow characteristics for turbulent flows. It is a two-equation model which uses the following transport equations for turbulence kinetic energy $k$ and turbulence dissipation rate $\varepsilon$ [15]:

$$ \frac{\partial (\rho k)}{\partial t} + \nabla \cdot (\rho k \mathbf{C}) = \nabla \cdot \left( \mu \left( \frac{H}{\sigma_k} \right) \nabla k \right) + P_k - D_k $$ (13)

$$ \frac{\partial (\rho \varepsilon)}{\partial t} + \nabla \cdot (\rho \varepsilon \mathbf{C}) = \nabla \cdot \left( \mu \left( \frac{H}{\sigma_\varepsilon} \right) \nabla \varepsilon \right) + \frac{\varepsilon}{k} \left( C_1 P_k - C_2 k D_k \right) $$ (14)

where $P_k$ and $D_k$ are respectively the turbulence kinetic energy production and destruction, with:

$$ P_k = 2\mu_t \mathbf{S}_{\alpha\beta} \mathbf{S}_{\alpha\beta} $$ (15)

and,

$$ D_k = \rho \varepsilon $$ (16)

The eddy viscosity $\mu_t$ is a function of the density $\rho$, turbulence kinetic energy $k$ and turbulence dissipation rate $\varepsilon$:
\[ \mu_i = \rho C_\mu \frac{k^2}{\varepsilon} \]  

The values for the model constants have been obtained by comprehensive data fitting based on a wide range of turbulent flows [15]:

\[ C_\mu = 0.09 \quad C_{1e} = 1.44 \quad C_{2e} = 1.92 \quad \sigma_k = 1.00 \quad \sigma_\varepsilon = 1.30 \]  

The discretized form of equations (13) and (14) are derived for FVPM as follows:

\[ \frac{d}{dt} (\rho V k_i) = \sum_j \left( \left( \mu + \frac{\mu_1}{\sigma_k} \right) \nabla k_i \right) \cdot \Delta j_i + \left( P_i - D_i \right) V_i \]  

\[ \frac{d}{dt} (\rho V \varepsilon_i) = \sum_j \left( \left( \mu + \frac{\mu_1}{\sigma_k} \right) \nabla \varepsilon_i \right) \cdot \Delta j_i \quad \left( \rho C_\mu \frac{k^2}{\varepsilon} \right) \left( C_{1e} P_i - C_{2e} D_i \right) V_i \]  

3.3. Boundary conditions for \( k \) and \( \varepsilon \)

For wall boundaries, the zero gradient boundary condition has been imposed for turbulence kinetic energy \( k \) and a scalable wall function approach is used to set turbulence variables near the wall. The production of \( k \) near the wall is computed as:

\[ P_{wall} = \frac{\tau^2}{\kappa \Delta y_i C_i^4 k_{wall}^2} \]  

The turbulence dissipation equation is not solved at the wall-adjacent particles, but instead is computed using equation (22).

\[ \varepsilon_{wall} = \frac{C_i^3 k_{wall}^2}{\kappa \Delta y_i} \]  

In equation (21) \( \kappa \) is von Kármán constant (\( \kappa \approx 0.41 \)) and \( \Delta y_i \) is normal distance of the center of particle \( i \) to the wall. \( \tau_w \) is the wall shear stress and is computed based on the scalable wall function approach. Thus,

\[ \tau_w = \frac{\rho C_p \frac{1}{\kappa} k^{\frac{1}{3}}}{u^*} \]  

where,

\[ u^* = \frac{1}{\kappa} \log(E y^*) \]  

In equation (23) \( C_p \) and \( y^* \) are respectively the velocity component parallel to the wall and the non-dimensional distance of the center of the particle \( i \) to the wall. Equation (24) is known as the logarithmic law, with \( E = 9.793 \), and \( y^* \) is computed as follows:

\[ y^* = \max \left( \frac{\rho \Delta y_i C_i^4 k_{wall}^2}{\mu} \right)^{\frac{1}{3}}, 11.06 \]  

The logarithmic-law velocity profile is not accurate for \( y^* < 30 \), including the buffer layer, where \( y^* \) is between 5 and 30, or the viscous sublayer (where \( y^* < 5 \)). The scalable wall function approach is used to produce consistent results for arbitrary particle refinements. Indeed, the scalable wall functions is used
avoid errors coming from applying the log-law in the laminar and buffer regions of the boundary layer by shifting the near-wall particle to \( y^+ = 11.06 \). If the boundary layer is not fully resolved, we will be relying on the logarithmic wall function approximation to model the boundary layer without losing the validity of the wall function approach. The value of \( y^+ = 11.06 \) in equation (25) is derived based on intersection of linear and logarithmic \( u^+ \) profiles [16].

For the free surface boundaries, the \( k \) and \( \varepsilon \) gradients normal to the free surface are set to zero, i.e. \( \partial k / \partial n = 0 \) and \( \partial \varepsilon / \partial n = 0 \) where \( n \) is a unit vector normal to the free surface. For the inlet boundary, the values of \( k \) and \( \varepsilon \) are identified and set based on the turbulence intensity and length scale.

3.4. Realizability

Based on the Boussinesq approximation equation (11) and the eddy viscosity equation (17), the normal stresses can be written as:

\[
\overline{C^2_u} = \frac{2}{3} k \delta_{\alpha\alpha} - 2 \nu S_{\alpha\alpha} = 2k \left( \frac{1}{3} \delta_{\alpha\alpha} - C_\mu \frac{k}{\varepsilon} S_{\alpha\alpha} \right)
\]  

(26)

The right side of the equation (26) should never be negative, as the normal stresses \( C^2_u \) are always non-negative. However, one can see that when the strain rate becomes large, the following condition can occur which is non-realizable [17]:

\[
\frac{k}{\varepsilon} S_{\alpha\alpha} > \frac{1}{3C_\mu} \approx 3.7
\]  

(27)

Realizability is the minimum requirement to prevent the turbulence model to generate non-physical results. The realizable model, proposed by Shih et al. [19], have shown improvement over the standard \( k-\varepsilon \) model for the cases with vortices, rotational flow and strong streamline curvature. The realizable \( k-\varepsilon \) model differs from the standard model in two important ways. First, the eddy viscosity equation has a non-constant \( C_\mu \) [18],[19] and second, the dissipation equation is modified based on the dynamic equation of the mean-square vorticity fluctuations. In the realizable model, the turbulence production equation remains the same as in the standard model (see equation (13)). Equation (28) is the dissipation equation for the realizable model:

\[
\frac{\partial (\rho \varepsilon)}{\partial t} + \nabla \cdot (\rho \varepsilon \mathbf{C}) = \nabla \cdot \left( (\mu + C_\mu \frac{k}{\varepsilon}) \nabla \varepsilon \right) + \rho \varepsilon \left( C_1 S - C_2 \varepsilon - \frac{\varepsilon^3}{k^2 + \sqrt{k \varepsilon}} \right)
\]  

(28)

where \( S = \sqrt{2 \sigma_{\alpha\beta} S_{\alpha\beta}} \) and \( C_1 = \max \left[ 0.43, Sk/(Sk + 5\varepsilon) \right] \). \( C_2 \) is set to 1.9 and \( C_\mu \) is a function of the strain rate tensor and turbulence variables (i.e. \( k \) and \( \varepsilon \)). The full formulation of \( C_\mu \) can be found in [18].

4. Validation

Equations (19) and (20) are solved for FVPM to compute turbulence kinetic energy \( k \) and dissipation rate \( \varepsilon \) in standard \( k-\varepsilon \) model. For the realizable model, the modified dissipation equation is discretized similarly to equation (20). Both models have been validated for two test cases: i) Flow inside a circular pipe and ii) open channel flow. All the validation tests have been done for \( Re = 10'000 \) and the results have been compared to the FVM-based ANSYS CFX and Fluent results. The schematic of test cases is shown in Figure 1.

4.1. Flow inside a circular pipe

The turbulence implementation in FVPM has been first validated for turbulent flow inside a circular pipe. Here, the length of the pipe is \( L = 10 \) m while the inlet velocity in \( C_{inlet} = 1.0 \) m-s\(^{-1}\), allowing the turbulent boundary layer to develop [20]. The walls are treated as no-slip impermeable boundaries. Using FVPM with both moving and fixed particles, the authors could produce velocity and turbulence kinetic energy...
profiles in good agreement with Eulerian FVM-based ANSYS CFX and Fluent results (see Figure 2 and Figure 3). For pipe internal flow tests, there is almost 30 particles per pipe diameter, thus, $D/X_{ref} \approx 30$.

![Figure 1](image1.png)

**Figure 1.** Pipe (top-left) and open channel flow (bottom-left) validation workbenches and mesh at the outlet of the pipe (right). In the FVPM solver, there is no particle to cover the second phase (air). Therefore, only the water particles are injected to the domain as inlet and the effect of air on the water is neglected.

![Figure 2](image2.png)

**Figure 2.** Velocity (left) and $k$ (right) of developed flow inside a circular pipe with standard $k$-$\varepsilon$ at $x = L$; fixed and moving particles FVPM vs. FVM.

![Figure 3](image3.png)

**Figure 3.** Velocity (left) and $k$ (right) of developed flow inside a circular pipe with realizable $k$-$\varepsilon$ at $x = L$; FVPM moving particles vs. FVM.

![Figure 4](image4.png)

**Figure 4.** Velocity (left) and $k$ (right) of open channel flow with standard and realizable $k$-$\varepsilon$ models at $x = L$; FVPM with moving particles vs. FVM with fixed-size grid. The data have been extracted from the wall (i.e. $r/D_{channel} = 0.5$) to the free surface ($r/D_{channel} \approx 0.25$).
Cp (left) and velocity (right) along the X-axis at Z/Djet = 0.033 above the wall for a jet impinging a flat plate with Re = 120'000.

Since the solver has been verified for fixed particle case in standard model (see Figure 2), the realizable model has been directly verified for moving particles. The authors use scalable wall function to avoid the y⁺ values become smaller than 11.06, which the logarithmic law is not valid for. Since the particles positions are not fixed in FVPM while in FVM the method features fixed-position elements, the distance of the particles from the wall yₚ varies during the simulation due to the particles motion. The y⁺ values then is changed and in FVPM solver vs. the FVM, which can produce different results for turbulence variables.

4.2. Free surface flow

The second test case is an open channel flow, which includes free surface effects. A schematic of the test case is shown in Figure 1 (bottom). The velocity magnitude C and turbulence kinetic energy k have been compared to the ANSYS CFX results for both standard and realizable k-ε models coupled with FVPM. The results show reasonably good agreement, despite having moving particles (see Figure 4). Since the FVPM solver is single-phase, there are no particles covering the upper part of the channel while in the CFX the solution is two-phase flow (including air and water) and the free surface is modelled based on the volume of fraction (VoF) method. The spatial resolution for the open channel simulation is Rchannel/Xref ≈ 15, where Rchannel = 0.5Dchannel.

FVPM is an ALE method, in which the particle velocity term is used in the convective flux computation and a velocity correction approach is then utilized to apply a proper particle velocity while FVM features fixed-position elements during the whole simulation with no element motion. All the discretization schemes used in ANSYS CFX for both the pipe and open channel simulations were 2nd order accurate while this is not the case for FVPM. These all could make the results of ANSYS CFX and GPU-SPHEROS to be different. The authors will also validate the models against the experimental data in the future to assess the model accuracy vs. reliable experimental results.
Schematic of Pelton buckets and water jet. The buckets are fed by injected water jet.

Figure 7. Schematic of Pelton buckets and water jet. The buckets are fed by injected water jet.

Figure 8. The jet inlet velocity and turbulence variables computed based on distributor simulation using ANSYS CFX. The $X$ and $Y$ correspond to horizontal and vertical axes, respectively.

5. Impinging jet on a flat plate

The code performance and accuracy has been assessed for an impinging jet on a flat plate as a dynamic free surface problem and the results have been compared to the ANSYS CFX results. The jet velocity at the inlet is $C_{\text{inlet}} = 4.0 \text{ m}\cdot\text{s}^{-1}$. The jet particles (with the resolution of $D_{\text{jet}}/X_{\text{ref}} = 30$) are injected as a circular inlet boundary and the plate acts as a no-slip wall boundary. The center of the plate corresponds to the center of the Cartesian coordinate system and the plate is perpendicular to the jet. The inlet of the jet has the diameter of $D_{\text{jet}} = 0.03 \text{ m}$ and is located at $Z_{\text{jet}} = 2.5D_{\text{jet}}$ above the flat plate. The pressure coefficient $C_p$ and the velocity profile along the $X$–axis and the visualized water jet impinging on the flat plate are respectively shown in Figure 5 and Figure 6. $C_p$ is averaged in time over the period ranging for $0.04 \text{ s} < t < 0.05 \text{ s}$ to filter out the pressure oscillations caused by compressibility. As shown in the Figure 5, there is a very good agreement between FVPM and CFX results using the same turbulence model for both, so the solver is now validated and ready to move on to the Pelton simulation.

6. Pelton turbine simulation

The fluid flow inside a rotating Pelton bucket is a transient free-surface flow with large deformation of boundaries. This physics has motivated many engineers and scientists to use particle-based methods for Pelton turbine simulation, since they are robust for modelling free surfaces. The FVPM capability and accuracy have already been validated by Jahanbakhsh et al. [4],[5] for different test cases. In the present work, a water jet feeding a rotating Pelton turbine has been simulated using GPU-SPHEROS. The
simulation has been done using both aforementioned eddy viscosity models. The results are compared to the experimental data measured by the authors.

In the present study, a high-speed water jet impinges on three rotating buckets. The turbine dimension is in model reduced scale (not the prototype) and the bucket width is \( B_2 = 0.08 \text{ m} \). The runner reference diameter is \( D_1 = 0.45 \text{ m} \). Three buckets rotate around the \( X \)-axis with a rotation speed \( N = 534.0 \text{ min}^{-1} \). The buckets are separated by an angle of 16.36 degrees, corresponding to a full Pelton runner featuring 22 buckets. The water jet is oriented in the \( Z \) direction and its inlet center is located at: \((x, y, z) = (0.0 \text{ m}, -0.5D_1, 0.2201 \text{ m})\). The jet diameter is \( D_2 = 0.0332 \text{ m} \). The schematic of the buckets and jet is shown in Figure 7.

The discharge velocity of the water jet, as well as turbulence variables including turbulence kinetic energy \( k \) and turbulence dissipation rate \( \varepsilon \), are computed based on simulation of the whole distributor using ANSYS CFX. Since there is no volume fraction parameter in FVPM and the solver, the particles’ velocity profile extracted from FVM solution are weighted by the water volume fraction and the jet diameter is then found in the sense of having the same water discharge as the CFX data. There are around 34 particles per jet diameter, therefore, \( D_2/X_{ref} = 34 \) to have solver \( y^+ \) around 300. The solver \( y^+ \) is initially estimated before launching the simulation based on the equation (29) in order to have a rough estimation for particle spacing. In this equation, \( \Delta y \) and \( L \) are respectively the distance between the wall and an adjacent fluid particle and the characteristic length \( L \). This formula is derived based on fluid mechanics flat-plate boundary layer theory [21]. The jet inlet velocity, \( k \) and \( \varepsilon \) are shown in Figure 8.

\[
y^+ = f \left( \frac{\Delta y}{L}, Re \right) = \left[ \sqrt{0.0135 \frac{\Delta y}{L}} \right] Re^{13/4}
\]

The time-history of the computed overall torque \( T_{ED} \) is shown in Figure 9. The average torque predicted by GPU-SPHEROS is underestimated (by almost 15\%) compared to the experimental torque measured by the authors during the model tests. This numerical errors is due to both numerical models and the spatial/time discretization. Since the FVPM is both (locally) conservative and consistent, the results, in theory, should be converged to the exact solution by refining the grid. However, the modelling errors are not negligible for many numerical models and the numerical results do not then converged to the exact solution. However, for the Pelton turbine case, the convergence will be studied for the future extension of the present research.

As a Central Difference Scheme (CDS) has been used for spatial discretization, which is 2\textsuperscript{nd} order accurate but leads to oscillations, the raw torque in Figure 9 is noisy. Here, standard and realizable models

![Figure 9](image-url)
produce almost the same torque. In fact, the realizable model is expected to have superior performance compared to standard model in the cases of rotational and swirling flow which is not the physic for Pelton turbine application.

7. Conclusion
Particle-based methods can be a robust approach for free surface problems simulation, specifically with moving boundaries, such as rotating Pelton turbine. In the present work, two linear eddy viscosity models have been implemented and coupled with the FVPM method. The implemented turbulence models are in good agreement with FVM-based CFX solver regardless of particle motion. While the realizable model is known to provide more accurate results than the standard model for swirling or separated flows, both models give similar results for the case of the Pelton turbine simulation. However, the realizable model has almost the same computational effort compared to the standard one while addressing the physical realizability issues of standard $k$-$\varepsilon$. A scalable wall function approach has been utilized for near wall region computations, to avoid errors caused by using the log-law relation in the buffer layer or viscous sublayer where the velocity profile is not logarithmic. In general, $k$-$\varepsilon$ models are simple, affordable and reasonably accurate for a wide range of flows. The next step will be to implement the $k$-$\omega$ Shear Stress Transport ($k$-$\omega$ SST) model to integrate with FVPM for problems in which the $k$-$\varepsilon$ models tend to be erroneous, such as in low-Reynolds region turbulence, flows with separation, etc.

Acknowledgment
This work was done in the context of a collaboration between the École Polytechnique Fédérale de Lausanne (EPFL), Laboratory for Hydraulic Machines (LMH), and Hitachi Mitsubishi Hydro Corporation (HMHydro).

REFERENCES
[1] Patel K, Patel B, Yadav M and Foggia T, 2010 Romania. Development of Pelton turbine using numerical simulation, 25th IAHR Symposium on Hydraulic Machines and Systems
[2] Gingold R A, Monaghan J J, 1977 Smoothed particle hydrodynamics-theory and application to non-spherical stars Mon. Not. R. Astron. Soc. 181, pp. 375–389,
[3] LeVeque R J 2002, Finite Volume Methods for Hyperbolic Problems, Cambridge university press, 31.
[4] Jahanbakhsh E, Vessaz C, Maertens A, Avellan F, 2016, Development of a Finite Volume Particle Method for 3-D fluid flow Computer Methods in Applied Mechanics and Engineering, 298, pp. 80-107.
[5] Jahanbakhsh E, Maertens A, Quinlan N J, Vessaz C, Avellan F, 2017, Exact finite volume particle method with spherical-support kernels, Computer Methods in Applied Mechanics and Engineering, 317, pp. 102–127.
[6] Quinlan N J, Lobovsky L, Nestor R M, 2014 Development of the meshless finite volume particle method with exact and efficient calculation of interparticle area, Computer Physics Communications, 185, pp. 1554–1563.
[7] Vessaz C, 2015, Finite Particle Flow Simulation of Free Jet Deviation by Rotating Pelton Buckets, École Polytechnique Fédérale de Lausanne (EPFL), doctoral thesis N° 6470.
[8] Vessaz C, Jahanbakhsh E, Avellan F, 2015, Flow Simulation of Jet Deviation by Rotating Pelton Buckets Using Finite Volume Particle Method, Transactions of the ASME, Journal of Fluids Engineering, 137 (7). 074501
[9] Alimizrazadeh S, Jahanbaksh E, Maertens A, Leguizamon S, Avellan F, 2018, GPU-accelerated 3-D Finite Volume Particle Method, Computers & Fluids, 171, pp. 79-93
[10] Pope S, 2000, Turbulent Flows, Cambridge University Press
[11] Chen C J, Jaw S, 1998, Fundamentals of turbulence modeling, Taylor & Francis, ISBN 1-56032-405-8
[12] Adrian R J, Christensen K T, Liu Z C, 2000 Analysis and Interpretation of instantaneous turbulent velocity fields, Experiments in Fluids, 29 pp. 275–290.
[13] Wilcox D C, 1988 Multiscale Model for Turbulent Flows, AIAA Journal, 26 (11) pp. 1311-1320.
[14] Kazuhiko S 1998 Recent development in Eddy Viscosity Modeling of Turbulence, R&D Review of Toyota CRDL 33 (1).
[15] Versteeg H K, Malalasekera W, 2007 England An introduction to computational fluid dynamics: The finite volume method, Pearson Education Ltd.
[16] ANSYS Fluent 12.0 Theory Guide 2009, ANSYS Inc.
[17] Speziale C G 1991 Analytical methods for the development of Reynolds-stress closures in turbulence, Ann. Rev. Fluid Mechanics 23 pp. 107-157.
[18] Reynolds W C 1987 Fundamentals of turbulence for turbulence modeling and simulation, Lecture Notes for Von Karman Institute Agard Report No. 755.
[19] Shih T H, Liou W W, Shabbir A, Yang Z, Zhu J, 1995 A New k-ε Eddy-Viscosity Model for High Reynolds Number Turbulent Flows - Model Development and Validation, Computers Fluids, 24 (3):227-238.
[20] Cimbala Y, Çengel A, John M, 2006. Fluid mechanics: fundamentals and applications (1st Ed.). McGraw-Hill Higher Education. pp. 321-329. ISBN 0072472367.
[21] Frank M, White, 2003 Fluid Mechanics, 5th edition, McGraw-Hill.