344

Numerical modelling of supercritical flow in circular conduit bends using SPH method[*]

Nikola M. Rosić[1], Milena B. Kolarević[2], Ljubodrag M. Savić[1], Dejana M. Đorđević[1], Radomir S. Kapor[1]
1. *Faculty of Civil Engineering, University of Belgrade, Belgrade, Serbia, E-mail: nrosic@grf.bg.ac.rs*
2. *Energoprojekt-Hidroinzenjering, Belgrade, Serbia*

Abstract: The capability of the smoothed-particle hydrodynamics (SPH) method to model supercritical flow in circular pipe bends is considered. The standard SPH method, which makes use of dynamic boundary particles (DBP), is supplemented with the original algorithm for the treatment of open boundaries. The method is assessed through a comparison with measured free-surface profiles in a pipe bend, and already proposed regression curves for estimation of the flow-type in a pipe bend. The sensitivity of the model to different parameters is also evaluated. It is shown that an adequate choice of the artificial viscosity coefficient and the initial particle spacing can lead to correct presentation of the flow-type in a bend. Due to easiness of its implementation, the SPH method can be efficiently used in the design of circular conduits with supercritical flow in a bend, such as tunnel spillways, and bottom outlets of dams, or storm sewers.

Key words: Smoothed-particle hydrodynamics (SPH), super critical bend flow, helical flow, artificial viscosity

Introduction

The smoothed-particle hydrodynamics (SPH) is the most popular meshfree particle method for fluid flow simulations. Although it was initially developed and used for astrophysical computations[1], nowadays, it is gaining increasing popularity in solving different engineering problems. A range of possible engineering applications can be found in Refs.[2-4]. Capabilities to model free-surface and non-Newtonian flows that are described with complex rheological models are the main advantages of SPH over Eulerian, grid-based methods in solving hydraulic engineering problems. The fact that it does not require free surface tracking and gridding techniques, makes it attractive for modelling these flows. As a meshfree method, the SPH is especially suitable for the simulation of discontinuous free-surface flows, where free surface tracking and gridding are challenging tasks for grid-based methods. Since the numerical solution is sought for moving particles, the SPH method is computationally very demanding when compared with grid-based methods.

However, recent development of parallel computing techniques enabled faster computations and wider application of the method in solving engineering problems[5].
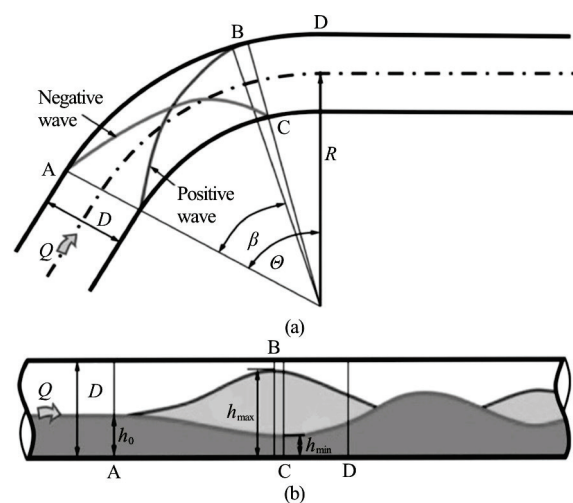

Fig.1 Standing waves in the pipe bend

A supercritical flow in a closed conduit bend is a good example for testing capabilities of the SPH method in solving hydraulic engineering problems. The

supercritical bend flow may develop in tunnel spillways and bottom outlets of dams, in storm sewers and in diversion tunnels. Due to high uncertainty about flow conditions that might develop therein, bends are commonly avoided, even at the expense of an increase in the construction costs.

The supercritical flow in a bend causes abrupt depth and velocity disturbances that spread further downstream (Fig.1). A positive standing wave develops along the outer wall, and a negative wave develops along the inner wall. In the case of a significant disturbance, the positive wave reaches the conduit invert and flow becomes helical (Fig.2(a)), or, for even larger disturbances, the flow turns to a choking flow (Fig.2(b)).
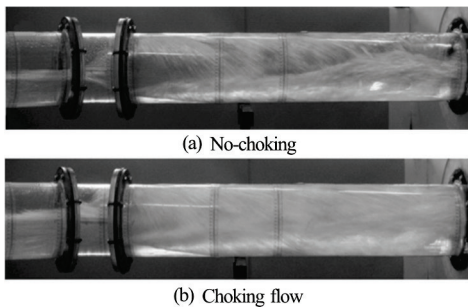

(a) No-choking

(b) Choking flow

Fig.2 Helical and choking flow in the pipe bend

The choking flow must be avoided, due to its periodicity, which is characterized by successive changes from pressurized to free-surface flow, and the entrainment of large amount of air into flowing water. This reduces conduit capacity and provokes vibrations. Even though no-choking helical flow may be permissible, it should be avoided for the design flow in order to stay on the safe side.
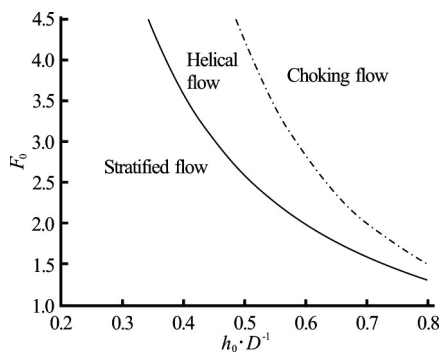

Fig.3 An example of limit curves for onset of helical and choking flow[6]

Previous experimental studies[6,7] have shown that the flow-type in the pipe bend depends on the deflection angle Θ, bend curvature R/D and the flow characteristics upstream of the bend (i.e., the relative depth h_0/D, and Froude number $Fr_0 = Q/\sqrt{gDh_0^4}$). Moreover, relationships which predict conditions for the development of the helical and choking flows were defined (Fig.3).

This paper aims at estimating a capability of the SPH method to predict the flow-type in the circular pipe bend. An open source code DualSPHysics[5,8,9] is modified for the purpose of modelling supercritical flow in circular pipe bends. The code is supplemented with an original algorithm which provides correct interpretation of the upstream boundary condition. The modified code is tested and verified against the experimental data.

The SPH method is applied to the so-called "clear" water flow, i.e., the flow with no air entrained. Since the significant entrainment of air can be expected only after the flow becomes helical, it seems appropriate to use the one-phase flow assumption to estimate the inception of the helical flow. With this assumption computations are markedly less demanding.

One experimental dataset[6] is used for model calibration, and another one[6,7] for its verification. Special attention is paid to the sensitivity analysis, in which the quality of the model performance is assessed by varying artificial viscosity coefficient and initial particle spacing.

1. Numerical model

The DualSPHysics code based on the standard SPH method is used in this study. This version of the SPHysics code is developed for GPU computations. A more detailed explanation of this code can be found in Refs.[5,8,9].

As it is already known the SPH method uses a set of particles to present a moving fluid. Hence, the SPH method solves equations of fluid dynamics that are written using material derivatives. The method applied in this study solves the continuity and the momentum equations in this form

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \mathbf{v} , \quad \frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\nabla \mathbf{P} + \mathbf{g} + \mathbf{\Pi} \qquad (1)$$

where $\mathbf{v}$ is the velocity, ρ and $\mathbf{P}$ are the density and the pressure respectively, $\mathbf{g}$ is the gravity acceleration and $\mathbf{\Pi}$ is the viscosity term. Time derivatives of the density are computed for each particle (i) using SPH interpolation rules. The numerical procedure is based on the interpolation of the weighting function W that is used as follows

$$\frac{D\rho_i}{Dt} = \sum_j m_j \mathbf{v}_{ij} \nabla W_{ij} ,$$

$$\frac{\mathrm{D}v_i}{\mathrm{D}t} = -\sum_j m_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} + \Pi_{ij} \right) \nabla W_{ij} + g \qquad (2)$$

where j refers to all particles located at a distance less than $2h$ from the particle i (in this study $h = \Delta p_0$, where Δp_0 is the initial spacing between particles), m_j is the mass of the particle j and v_{ij} is the relative velocity of particle j with respect to particle i, Π_{ij} is the artificial viscosity term proposed by Monaghnan[10]:

$$\Pi_{ij} = \frac{-\alpha \bar{c}_{ij} \mu_{ij}}{\bar{\rho}_{ij}}, \quad v_{ij} r_{ij} < 0 \qquad (3a)$$

$$\Pi_{ij} = 0, \quad v_{ij} r_{ij} > 0 \qquad (3b)$$

where r_{ij} is the relative position vector of the particle j with respect to i, $\mu_{ij} = (hv_{ij} r_{ij})/(r_{ij} r_{ij} + 0.01h^2)$, $\bar{c}_{ij} = (c_i + c_j)/2$ (c is the speed of sound in water, which is calculated as a square root of the partial derivative of pressure with respect to density), α is a free parameter that is problem dependent (the common value is of the order of 10^{-1}). The formulae used for the weighting function are:

$$W_{ab}(q,h) = \frac{1}{\pi h^3} = 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3, \quad 0 \le q \le 1 \qquad (4a)$$

$$W_{ab}(q,h) = \frac{1}{\pi h^3} = \frac{1}{4}(2-q)^3, \quad 1 \le q \le 2 \qquad (4b)$$

$$W_{ab}(q,h) = \frac{1}{\pi h^3} = 0, \quad q \ge 2 \qquad (4c)$$

where $q = |r_{ij}|/h$.

In the standard SPH method, the fluid is treated as weakly compressible, and the pressure is calculated from the equation of state. This approach is computationally less demanding than the incompressible fluid model, where Poisson equation has to be solved. The following equation of state (Tait's equation) is used

$$P = B \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \qquad (5)$$

where $\gamma = 7$ is the polytropic constant, $B = c_0^2 \rho_0 / \gamma$ is a constant related to the bulk viscosity, $\rho_0 = 1\,000 \text{ kg/m}^3$ is the density of the fluid at the free

surface, and c_0 is the speed of sound at the free surface. The speed of sound could be artificially reduced in order to enable a larger time step.

The SPH simulations with the presented equations are followed by pressure oscillations. To avoid this spurious effect the following correction, so-called "smoothing of density" may be periodically applied

$$\tilde{\rho}_i = \sum_j \rho_j \tilde{W}_{ij} \frac{m_j}{\rho_j} \qquad (6)$$

This formulation is known as Shepard's filter[11]. The kernel weighting function is corrected as follows

$$\tilde{W}_{ij} = \frac{W_{ij}}{\sum_j W_{ij} \frac{m_j}{\rho_j}} \qquad (7)$$

In this study, the filtering was repeated after 35 steps. Results have shown, that smoothing of the density (pressure) field can stabilise computation in the absence of the artificial viscosity term.

Time integration of Eqs.(2) was performed using a predictor-corrector scheme that consists of three steps. The predictor values ($F_i = \mathrm{d}v_i / \mathrm{d}t$ and $D_i = \mathrm{d}\rho_i / \mathrm{d}t$) in the first step, are calculated according to:

$$v_i^{p,n+1/2} = v_i^n + \frac{\Delta t}{2} F_i^n, \quad \rho_i^{p,n+1} = \rho_i^n + \frac{\Delta t}{2} D_i^n,$$

$$r_i^{p,n+1} = r_i^n + \frac{\Delta t}{2} v_i^n \qquad (8)$$

These values are then corrected in the corrector step:

$$v_i^{n+1/2} = v_i^n + \frac{\Delta t}{2} F_i^{p,n+1/2}, \quad \rho_i^{n+1/2} = \rho_i^n + \frac{\Delta t}{2} D_i^{p,n+1/2},$$

$$r_i^{n+1/2} = r_i^n + \frac{\Delta t}{2} v_i^{p,n+1/2} \qquad (9)$$

Final values at the end of the time step are calculated in the last computational step, as follows:

$$v_i^{n+1} = 2v_i^{n+1/2} - v_i^n, \quad \rho_i^{n+1} = 2\rho_i^{n+1/2} - \rho_i^n,$$

$$r_i^{n+1} = 2r_i^{n+1/2} - r_i^n \qquad (10)$$

The mechanical energy loss coefficient in the pipe bend was calculated using Bernoulli's equation for compressible flow

$$\xi = \frac{\left(\int_p \frac{dp}{\rho} + \frac{V^2}{2} + gz\right)_u - \left(\int_p \frac{dp}{\rho} + \frac{V^2}{2} + gz\right)_d}{\left(\frac{V^2}{2}\right)_u} \quad (11)$$

Terms in the equation are time averages, and subscripts u and d denote the upstream and downstream cross-sections respectively.

In the version of the SPH method applied in this study, a pipe wall is modelled using the–so-called–dynamic boundary particles (DBPs)[12]. These particles remain still (their position is fixed) during the computation. However, they appear in relationships for the approximation to the derivatives of the fluid particles that are located within the distance of $2h$ from the boundary. For DBPs, the density is computed from the continuity equation, and the pressure change is computed from the equation of state. When a fluid particle approaches the wall particles (DBPs), the pressure increases, thus creating a repulsing mechanism that prevents the fluid from penetrating through the solid boundary.

In the case of a supercritical flow, boundary conditions must be prescribed at the upstream end of the computational domain. The treatment of an inflow/outflow boundary with the SPH method is still not satisfactorily resolved. For instance, Federico et al.[13] proposed a technique based on fixed ghost particles[14,15] for solving this problem. However, the use of fixed ghost particles is not as simple as that of DPBs. Moreover, particles in the buffer zone, from which they enter the physical domain considered are constantly generated by additional algorithm, which makes the whole computational procedure cumbersome, and thus not suitable for parallelization using GPUs.

Since the steady state is reached fairly fast in the problem under investigation (in just a few seconds), it is more efficient to load all particles in the buffer zone in front of the upstream boundary at the beginning of computation (i.e., no new particles are generated during the computation). Hence, a new procedure is developed in this study. Particles in the buffer zone form a water column with prescribed velocity and pressure. The column is translated through the upstream boundary into the flow domain. As with the Federico et al., there is no influence of particles from the considered flow domain on particles in the buffer zone (while the buffer zone particles influence those in the flow domain). This basically means that only a position of the buffer particle is calculated from the given velocity value. Consequently, the computational time is rather independent of the number of particles in the buffer zone. Furthermore, the authors suggest that the water column in the buffer zone is positioned such

that the distance from boundary particles (δz) on the inflow boundary is greater than $\Delta p_0 (\Delta p_0 = h)$ as illustrated on Fig.4. This distance (δz) affects the pressure difference between the DBP at the inflow boundary and the fluid particle that enters the flow domain and, consequently, the repulsion between the two particles, which may provoke numerical instability.
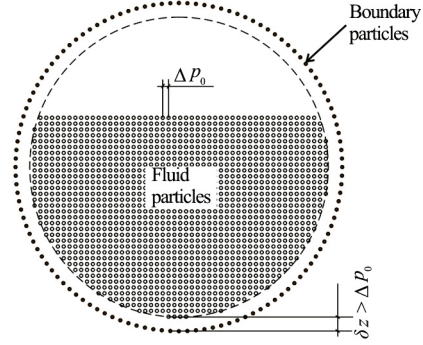

Fig.4 Cross-section beyond real upstream boundary of the physical domain

Repulsion problems might arise due to the fact that DBPs have no influence on buffer particles, which start to approach the solid boundary as they enter the flow domain, thus provoking a greater increase of pressure in DBPs than that developed when the flow domain is already populated with particles. To estimate the pressure rise in the DBP, a movement of an arbitrary, single fluid particle towards the solid boundary is analyzed. The change of pressure in DBPs may be calculated from expressions (2) and (5) using the following relation

$$\frac{d\boldsymbol{P}}{dt} = \frac{\partial \boldsymbol{P}}{\partial \rho}\frac{d\rho}{dt} = \frac{B\gamma}{\rho_0^\gamma}\rho^\gamma m_f \boldsymbol{v}_{bf}\nabla W_{bf} \quad (12)$$

The relation shows that the change in pressure for the given flow velocity depends on compression parameters of the fluid on one side, and the particle spacing, via gradient of the weighting function, on the other side. Authors suggest an increase in distance of the buffer zone from the boundary (δz) as a solution to the problem of the pressure discontinuity, since compression parameters are overall constants. The increase in distance means that the gradient of the weighting function is reduced. The weighting function is a monotonously decreasing function within the interval $1 < q < 2$. Figure 5 shows the particle distribution for three distinct values of δz. Boundary particles (DBPs) are set at the bottom, below fluid particles of the flow domain. In these examples the particle velocity is 2 m/s, the water column height is 0.05 m and the initial particle spacing is $\Delta p_0 = 1$ mm.
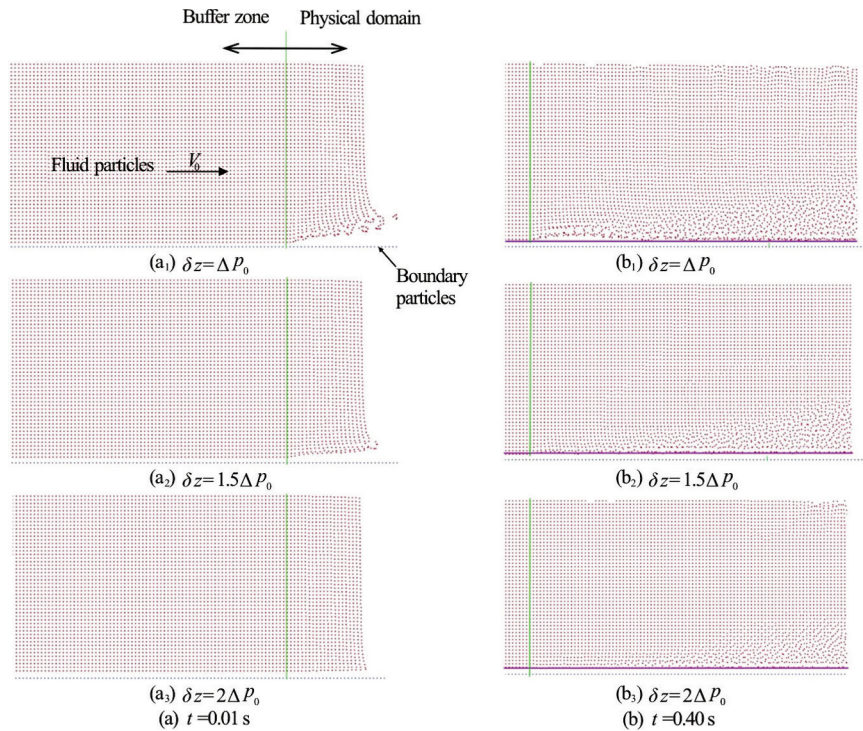
Fig.5 (Color online) Influence of δz on the flow pattern

Figures 5(a_1)-5(a_3) undoubtedly confirm that the increase in the buffer zone distance δz results in reduced bouncing of fluid particles from the boundary (when $\delta z = \Delta p_0 = 2h$ there are no bounced particles, Fig.5(a_3)). Distributions of particles 0 s, 40 s after the beginning of simulation (Figs.5(b_1)-5(b_3) show that the distance at which fluid particles are kept away from DBPs is approximately $2\Delta p_0$ (horizontal line) in either case. It is also interesting to take a look at the particle distribution near the free surface. It is readily noticeable that there is no disturbance of the free surface only when $\delta z = 1.5\Delta p_0$, and that notable disturbances are developed when $\delta z = \Delta p_0$. In the case when $\delta z = 2\Delta p_0$ disturbances are small. Therefore, the choice of the optimal value of δz for arbitrary flow geometries is left for further investigation. For all simulations in this study (circular pipe bend) initial distance is $\delta z = 2\Delta p_0$, since oscillations of the depth and the velocity are negligible.

2. Experimental setup

The study of the supercritical flow in a pipe bend with the SPH method described in the previous section is based on the experimental results of Kolarević et al.[6]. The experimental setup consists of the upstream reservoir, followed by a 0.45 m long transitional section and a horizontal 0.15 m diameter pipe (Fig.6). The circular pipe is made of 2 m long upstream, and 1.5 m long downstream, straight sections with the circular bend in between.
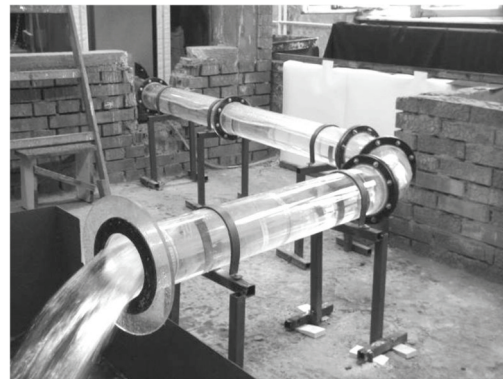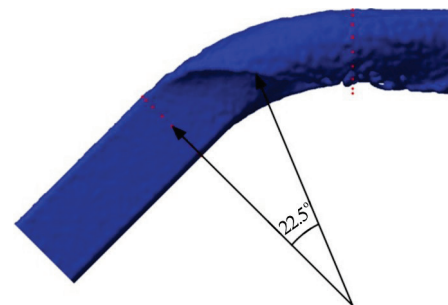

Fig.6 Experimental setup

Fig.7 (Color online) Location of the cross-section where the wave reaches the pipe invert ($\alpha = 0$ and $\Delta p_0 = 4.5$ mm)
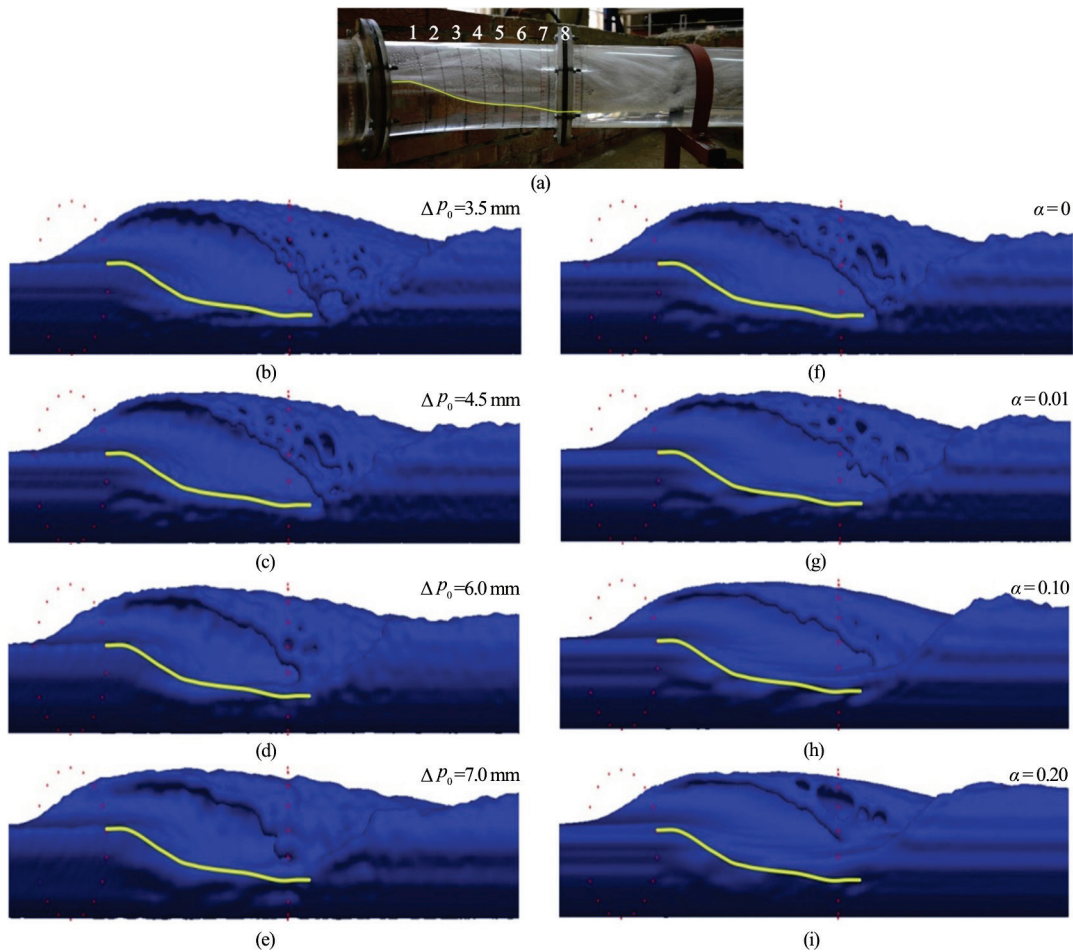
Fig.8 (Color online) (a) Helical flow in the pipe bend in the experiment[6], (b)-(e) influence of Δp_0 (for $\alpha = 0$) on the flow in the pipe bend, (f)-(i) influence of α (for $\Delta p_0 = 4.5$ mm)

Kolarević et al.[6] performed a series of experiments to study the influence of the deflection angle and bend curvature on the type of the bend flow. In this paper the bend curvature of $D/R = 1/3$ and the deflection angle of $45°$ are considered.

3. Numerical simulations

To enable comparison between numerical and physical model results, it is necessary to provide the same values of the flow discharge and the depth at the gauging location. Since the described method for the definition of the upstream boundary condition requires shifting of the fluid particles from the solid boundary, the flow depth cannot be assigned at the gauging section, but some distance upstream. Hence, the depth of the water column at the fictitious upstream boundary must be determined by a trial and error.

Courant number value of 0.3 and the sound celerity of approximately 30 m/s are used in all numerical simulations to provide a stable computation and to prevent fluid particle penetrating through the solid boundary. The initial particle spacing is varied in the range between 3.5 mm and 7.0 mm. In addition to the particle spacing, the most important parameter in these simulations proved to be the artificial viscosity coefficient α (Eq.(3)).

The open source code DualSPHysics is modified to enable inclusion of the open boundary condition. This requires that all particles are loaded into GPU memory at the beginning of computation. Initially, the pipe is empty, and particles form the water column upstream of the boundary (buffer zone). The length of the water column (i.e., the number of particles) is determined from the simulation time necessary to reach the steady state. The maximum number of particles was approximately 8×10^6, which resulted in a GPU time of 8 hours that was necessary to simulate 10 s on GTX 970 GPU.

4. Results

The numerical model is calibrated against the experimental data from the scenario with helical flow.

The flow discharge in this experiment was 22.4 l/s and the flow depth was 0.09 m. The calibration criterion is a requirement to match the location of the section where the positive standing wave reaches the pipe invert in the experiment. The wave reached the pipe invert in the physical model in the cross-section with $\beta = 22.5^\circ$, measured from the beginning of the bend (Fig.7). The best agreement between the two models (numerical and physical) is achieved with no artificial viscosity. It should be noted that in the absence of the artificial viscosity, the Shepard filter is required for stable computation. As it was expected, a better agreement with the measurements is attained with lower values of Δp_0, with the remark that no significant improvements are achieved for $\Delta p_0 < 4.5$ mm.

Simulated water surface profiles for different α and Δp_0 values are compared to the measured one in Figs.8(a)-8(i). The measured water surface profile along the inner bend is presented with the line. It is readily noticeable that the increase in both α and Δp_0 deteriorates the agreement with measurements, i.e. that the best agreement is accomplished with no artificial viscosity added ($\alpha = 0$) and small Δp_0-value. The normalized RMS errors CV (RMSE) in cross-sections 1 to 8 (Figs.8(b)-8(i)) for the analyzed α and Δp_0 values are presented in Tables 1 and 2, respectively. The error is calculated from the following expression

$$CV(RMSE) = \sqrt{\frac{\sum_{i=1}^{8}(z_{i,SPH} - z_{i,m})^2}{\sum_{i=1}^{8} z_{i,m}}} \qquad (13)$$

where $z_{i,SPH}$ is the calculated water surface elevation on the inner bend and $z_{i,m}$ is the measured one. A location of the cross section in which the wave reaches the pipe invert when $\alpha = 0$ and $\Delta p_0 = 4.5$ mm is indicated in Fig.7.

Table 1 Normalized RMS errors CV(RMSE) in cross-sections 1 to 8 for the analyzed α - values and $\Delta p_0 = 4.5$ mm

α ($\Delta p_0 = 4.5$ mm)	CV (RMSE)/%
0	8.5
0.01	13.0
0.10	16.7
0.20	25.5

Table 2 Normalized RMS errors CV (RMSE) in cross-sections 1 to 8 for the analyzed Δp_0 - values and no artificial viscosity added ($\alpha = 0$)

Δp_0 / mm ($\alpha = 0$)	CV (RMSE)/%
3.5	8.3
4.5	8.5
6.0	15.4
7.0	17.4

The sensitivity of the location where the wave reaches the pipe invert, i.e. the sensitivity of the invert angle β to the α-value is presented in Fig.9. It is evident that the β-angle is rather insensitive to the change in α-value. The sensitivity of the invert angle β to the initial particle spacing Δp_0 is presented in Fig.10. One can notice that the agreement between the computed and measured values is better for lower values of Δp_0 ($\Delta p_0 \leq 4.5$ mm).
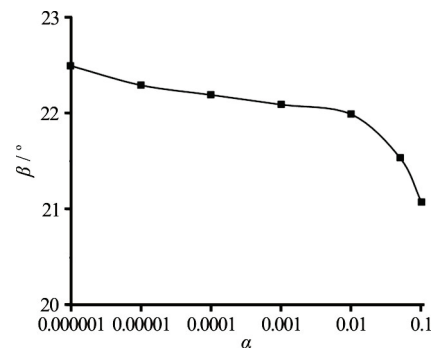

Fig.9 The influence of the artificial viscosity coefficient α on the invert angle β
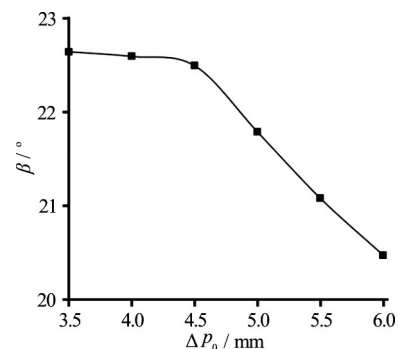

Fig.10 The influence of the initial particle spacing Δp_0, on the invert angle β

The influence of the two parameters on the bend loss coefficient ξ is presented in Figs.11 and 12. It is obvious that the bend loss coefficient increases with

an increase in both artificial viscosity (Fig.11) and the particle spacing Δp_0 (Fig.12). Additionally, there is an abrupt increase in ξ when $\alpha \geq 0.01$.
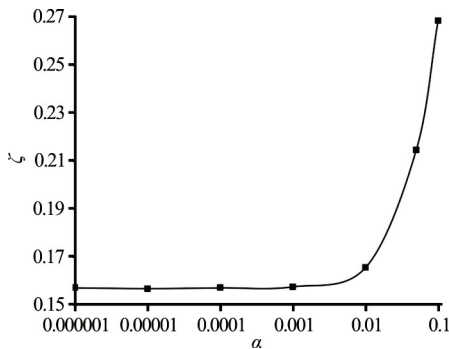

Fig.11 The influence of the artificial viscosity coefficient α on the bend loss coefficient ξ
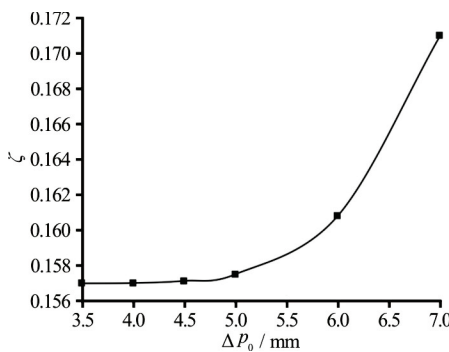

Fig.12 The influence of the initial particle spacing Δp_0 on the bend loss coefficient ξ

(a) Physical model[6]

(b) Numerical model
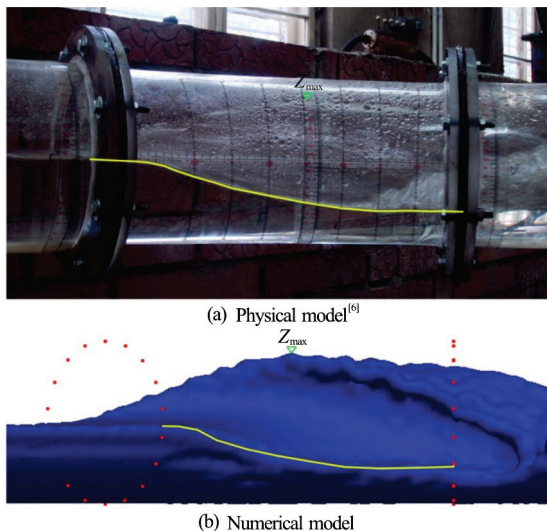
Fig.13 (Color online) Flow without turning ($\alpha = 0$ and $\Delta p_0 = 4.5$ mm)

The model is verified against the data from the scenario in which the flow had not yet become helical (Fig.13). The photograph of no-turning flow in the experiment is given in Fig.13(a), while the numerical simulation results are presented in Fig.13(b). The agreement between numerical and experimental results in the bend is even better than in the case with the helical flow (the corresponding CV (RMSE) is 6.1%, when compared with 8.5% for helical flow, Table 1). Moreover, the simulated and observed maximum depths on the outer bend are almost identical. Since the air-water flow is not modeled in this study, better prediction for the case without turning of the flow, where less air was entrained than for the case with the helical flow, would be expected.

To assess the capability of the SPH method to predict conditions for the establishment of the helical flow in the pipe bend, scenarios with various values of the relative flow depth and Froude number are simulated, using parameters defined in the calibration process (i.e., for $\alpha = 0$ and $\Delta p_0 = 4.5$ mm). These simulations may be understood as additional model verification. Numerical simulation results are presented in Fig.14 together with the limit curves for the onset of helical flow, based on experiments of Kolarević et al.[6], Hager and Gisonni[7]. Both limit curves are regression curves derived from the experimental data.
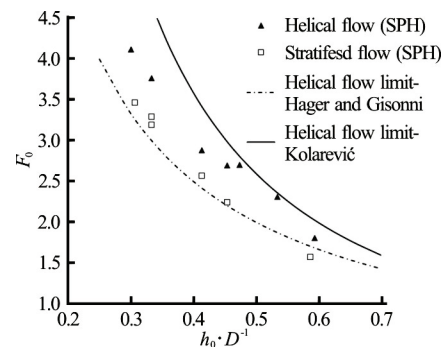

Fig.14 Comparison between numerical and experimental results

Since numerical simulation results lie midway between the two curves in the whole range of relative inflow depths and Froude numbers, the authors do believe that the SPH method results can be used as a good estimate for the preliminary design of closed conduits with bends in the case of supercritical flow, when the physical model is not affordable.

5. Conclusions

The SPH method is applied for the simulation of supercritical flow in the circular pipe bend. In this study the standard SPH method is supplemented with the original algorithm for the treatment of the upstream boundary condition. The comparison against experimental data has proven that the SPH method is a reliable tool for the prediction of the helical-type flow in the circular pipe bend.

Satisfactory performance of the model is achieved with no artificial viscosity (i.e., for $\alpha = 0$) and the initial particle spacing of $\Delta p_0 = 4.5$ mm. It turned out that in the absence of the artificial viscosity, the Shepard filter must be applied to stabilize computation.

An acceptable agreement between the numerical simulation results and the experimental ones, and the easiness of the model implementation, suggest that the SPH method can be efficiently used in the preliminary design of circular conduit bends in tunnel spillways, bottom outlets of dams, and storm sewers where supercritical flow might be expected to develop. It is also, reasonable to assume that the model will produce reliable results for closed conduit with cross section of different shapes (rectangular, horse-shoe, etc.).

As for the future research the model should be modified to cope with the air-water mixture flow, and it should be tested in conduits with different cross-sectional shapes.

Acknowledgements

References

[1] Gingold R. A., Monaghan J. J. Smoothed particle hydrodynamics: Theory and application to non-spherical stars [J]. *Monthly Notices of the Royal Astronomical Society*, 1977, 181(3): 375-389.

[2] Liu M. B., Liu G. R. Smoothed particle hydrodynamics (SPH): An overview and recent developments [J]. *Archives of Computational Methods in Engineering*, 2010, 17(1): 25-76.

[3] Violeau D. Fluid mechanics and the SPH method: Theory and applications [M]. Oxford, UK: Oxford University Press, 2012.

[4] Hughes K., Vignjevic R., Campbell J. et al. From aerospace to offshore: Bridging the numerical simulation gaps–Simulation advancements for fluid structure interaction problems [J]. *International Journal of Impact Engineering*, 2013, 61: 48-63.

[5] Crespo A. J. C., Domínguez J. M., Barreiro A. et al. GPUs, a new tool of acceleration in CFD: Efficiency and reliability on Smoothed Particle Hydrodynamics methods [J]. *Plos One*, 2011, 6(6): e20685.

[6] Kolarević M., Savić L., Kapor R. et al. Supercritical flow in circular closed-conduit bends [J]. *Journal of Hydraulic Research*, 2015, 53(1): 93-100.

[7] Gisonni C., Hager W. Bend flow in bottom outlets [C]. *Proeedings of 28th IAHR Congress.* Graz, Austria, 1999.

[8] Gómez-Gesteira M., Rogers B. D., Crespo A. J. C. et al. SPHysics-development of a free-surface fluid solver-Part 1: Theory and formulations [J]. *Computers and Geosciences*, 2012, 48: 289-299.

[9] Gómez-Gesteira M., Crespo A. J. C., Rogers B. D. et al. SPHysics-development of a free-surface fluid solver-Part 2: Efficiency and test cases [J]. *Computers and Geosciences*, 2012, 48: 300-307.

[10] Monaghan J. J. Simulating free surface flow with SPH [J]. *Journal of Computational Physics*, 1994, 110(2): 399-440.

[11] Panizzo A. Physical and numerical modelling of subaerial landslide generated waves [D]. Doctoral Thesis, L'Aquila, Italy: Universita degli Studi di L'Aquila, 2004.

[12] Dalrymple R. A., Knio O. SPH modelling of water waves [C]. *Proceedings of Coastal Dynamics'01.* Lund, Sweden, 2000, 779-787.

[13] Federico I., Marrone S., Colagrossi A. et al. Simulating 2D open-channel flows through an SPH model [J]. *European Journal of Mechanics-B/Fluids*, 2012, 34(7): 35-46.

[14] Marrone S., Antuono M., Colagrossi A. et al. Enhanced boundary treatment in 2D smoothed particle hydrodynamics models [C]. *Proceedings of XIX Congress AIMETA.* Ancona, Italy, 2009.

[15] Marrone S., Antuono M., Colagrossi A. et al. δ-SPH model for simulating violent impact flows [J]. *Computer Methods in Applied Mechanics and Engineering*, 2011, 200(13): 1526-1542.