Modelling of transport and collisions between rigid bodies to simulate the jam formation in urban flows

S. Hadji(1), A. Ouahsine(1)*, H. Naceur(1) and P. Sergent(2)
(1)Université de Technologie de Compiègne, Laboratoire Roberval, UMR-CNRS 6253
B.P. 20529, 60205 Compiègne Cedex, France
(2)CETMEF, Département DRIM, 2 Bd Gambetta, 60200 Compiègne, France

ABSTRACT
This study deals with the simulation of transport and interaction between bodies considered as a rectangular shape particles, in urban flow. We used an hydrodynamic two-dimensional finite elements model coupled to the particles model based on Maxey-Riley equations, and taking into account of contact between bodies. The finite element discretization is based on the velocity field richer than pressure field, and the particles displacements are computed by using a rigid body motion method. A collision strategy is also developed to handle cases in which bodies touch.

1. INTRODUCTION

Many observations of damages caused by flash floods or dam-break floods indicate that floating debris and debris jams are a substantial additional risk during flooding. The jams result from accumulations of materials (Brushes, uprooted trees, carried cars. . .), which temporarily causes a water reserve to the upstream. When these jams yield or breakup, the rupture causes the violent discharge of a great quantity of water and floating bodies. With the downstream, the characteristics of the flood change then brutally (acceleration of the current, abrupt rise of water, solid transport), and this in a not easily foreseeable way. Thus, they often to the origin of the principal damages generated by the quick swellings (small water courses to strong slope, pluvial urban streaming).

The jam generally occurs in the rivers and to proximity of the bridges, and often generates a flood (Fig. 1). Thus, the motion and the orientation of transported bodies, which form this jam, is a problems of practical interest to understand how these natural barriers are formed. In particular for shallow water, the orientations for example of tree's branches in river and their collisions can form stopping between bridge piers and then perturb the flow (Fig. 2).

In the past decade, many researchers developed numerical methods for direct simulation (DNS) of fluid-particle interaction (see [1], [2], [3], [4]). These problems are fluid and solid interaction problem and employ remeshing techniques or fictitious domain methods which involve a lot of computational storage and long calculation time at each time step.

The purpose of this paper is to simulate the Debris-Jam formation by floating bodies. The method used is based on the computation of the pathway of bodies of rectangular shape, by including the translation and rotation movements, without the remeshing technique. Thus, a numerical simulation of incompressible viscous flow past moving rigid bodies, when the motion of the bodies is not known in advance but results from the hydrodynamical coupling and external forces such as gravity and collisions, is performed. A simple strategy to take

[*] Address for correspondence: Email : ouahsine@utc.fr

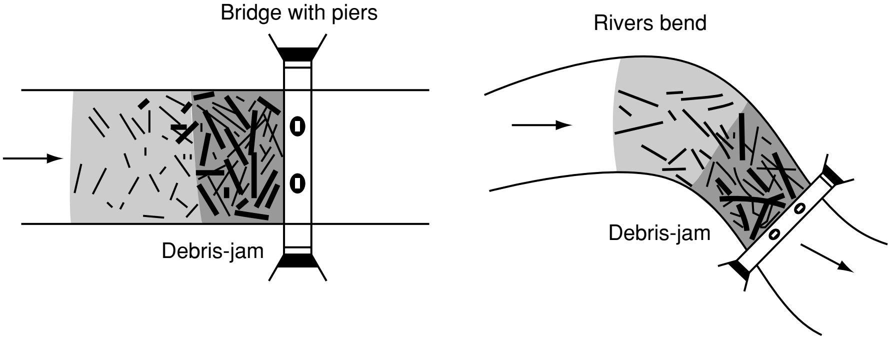

Figure 1 Schematic presentation of potential Debris-jam sites.

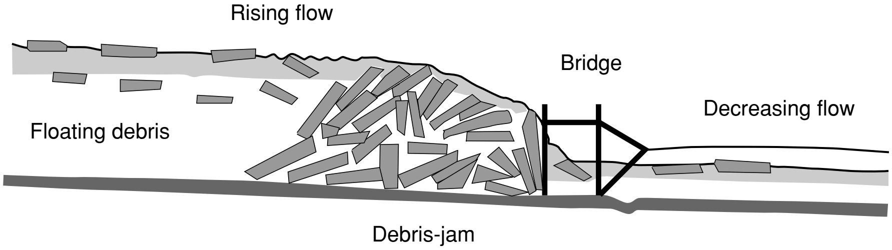

Figure 2 Conceptual sketch of the formation and breaching of debris-jams at potential locations (Bridges with piers).

into account body/body and body/wall collisions is then used. The outline of this paper is as follows. The hydrodynamic finite element model is presented in section 2. The modelling of the Fluid-Rigid body interaction is presented in section 3. Section 4 is devoted to Particle-Particle collisions for rigid bodies. The computational procedure is presented in section 5 and the numerical tests are presented in section 6. Results and discussions are given in 7.

2. HYDRODYNAMIC FLOW MODEL

The two-dimensional hydrodynamical model was based on the resolution of shallow-water equations, which were obtained by using hydrostatic and Boussinesq approximations and by integrating Navier–Stokes equations over total water depth. Notice, that for many practical surface-water flow applications, knowledge of the full three dimensional flow behaviors is not needed and it is sufficient to use the depth average flow quantities in two perpendicular horizontal directions. Thus, the depth-averaged momentum, and continuity equations lead to the following Saint-Venant set of equations

$$\begin{cases} \dfrac{\partial \mathbf{u}_f}{\partial t} + \mathbf{u}_f.\nabla \mathbf{u}_f + g\nabla h - \nu \nabla \mathbf{u}_f = \mathbf{F} \\ \dfrac{\partial h}{\partial t} + \nabla(h\mathbf{u}_f) \qquad\qquad = 0 \end{cases} \tag{1}$$

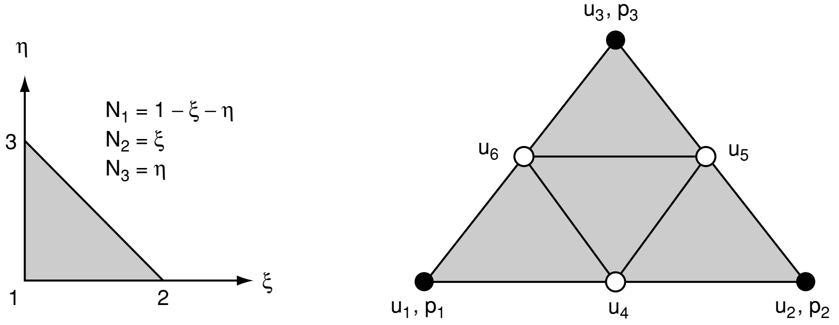

Figure 3 P2-P1 element: Six-nodes base triangle composed of 4 sub-triangles.

where $\mathbf{u}_f$= (u_f , υ_f) the depth average velocity with u_f and υ_f are the components in the x and y coordinate directions, defined by:

$$\mathbf{u}_f = \frac{1}{H}\int_{z_b}^{h}(u,\upsilon)dz \tag{2}$$

H (x, y, t) is the water depth, $z_b(x, y, t)$ the bed elevation and $h(x, y, t) = z_b(x, y, t) + H$ (x, y, t) is the water surface elevation, v is the effective viscosity, which includes the dispersion and the turbulence contributions. **F** = (Fx, Fy) integrate volume forces (Coriolis), actions exerted on the bottom (friction) or on the free surface (wind) and also forces as radiation constraints. Thus:

$$\mathbf{F} = \underbrace{-\beta\mathbf{u}_f}_{\textit{Friction forces}} + \underbrace{2\mathbf{u}_f \times \mathbf{k}\omega\ \sin\varphi}_{\textit{Coriolis forces}} - \underbrace{r\frac{Q_a}{Q}\frac{1}{H}\left\|\mathbf{W}\right\|\mathbf{W}}_{\textit{Wind forces}} \tag{3}$$

where **k** = (0, 0, 1) is the vertical unit vector*;* ω is the earth velocity in *rd*/*sec*, φ the latitude in degrees, r is a coefficient equal to 0.0026, **w** = (ω_x, ω_y) are wind components in x and y directions, Q_a and Q are densities of air and water respectively. β depends on the value of the following coefficient γ defined by:

$$\gamma = \left[19.8\,\mathrm{In}\left(\frac{915}{v}\right)\right]^{-1} \tag{4}$$

Thus:

$$\begin{cases} \beta = \dfrac{g}{\gamma^2}\dfrac{\sqrt{\mathbf{u}_f^2}}{H} & \textit{if}\ \ c > 1 \\ \beta = g\gamma^2\dfrac{\sqrt{\mathbf{u}_f^2}}{H^{3/4}} & \textit{if}\ \ c > 1 \end{cases} \tag{5}$$

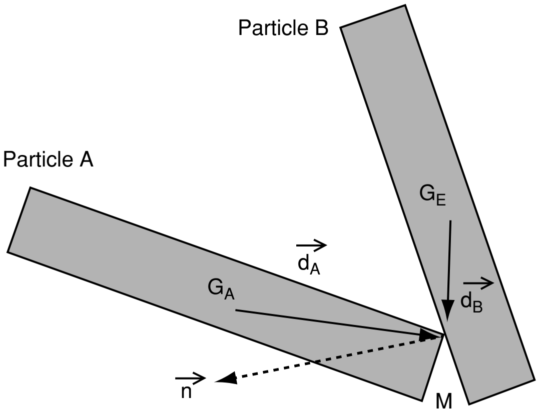

Figure 4 Collision between 2 particles.

For each finite element, the turbulent viscosity is calculated using the Prandtl-Smagorinsky formula:

$$\nu_t = l_m^2 \left[\frac{\sqrt{\bar{\mathbf{u}}^2}_f}{\nu_t} .C_f \frac{\bar{\mathbf{u}}^2{}_f}{\bar{h}} + 2(\nabla \bar{\mathbf{u}}_f)^2 + (\nabla .\bar{\mathbf{u}}_{friction})^2 \right]^{1/2} \tag{6}$$

For each finite element, $\bar{\mathbf{u}}_f$ is the averaged velocities, $\bar{h}$ the averaged free surface, $\bar{\mathbf{u}}_{friction}$ the friction velocity and l_m the mixing length defined by:

$$l_m = \alpha \bar{h}\ ,\ \alpha = 1.612 \frac{\sqrt{C_f}}{\bar{h}^{3/2}} \left[\frac{\bar{\mathbf{u}}_f^2}{2(\nabla \bar{\mathbf{u}}_f)^2 + (\nabla .\bar{\mathbf{u}}_f)^2} \right]^{3/4} \tag{7}$$

2.1. FINITE ELEMENT DISCRETIZATION AND VARIATIONAL FORM

The weak form associated to the set of equations (1) is:

$$w = w_{\mathbf{u}f} + w_h = 0\ \forall \quad \delta \mathbf{u}_f, \delta h \tag{8}$$

with:

$$\begin{cases} w_{\mathbf{u}f} = \int_\Omega \delta \mathbf{u}_f \left(\frac{\partial \mathbf{u}_f}{\partial t} + (\mathbf{u}_f .\nabla)\mathbf{u}_f + g\nabla h + \frac{g\left|\mathbf{u}_f\right|}{\gamma^2 H} + \mathbf{F} \right) d\Omega \\ \qquad + \int_{\partial\Omega} \nabla \delta \mathbf{u}_f \frac{\left[\tau\right]}{Q} ds - \int_{\partial\Omega} \delta \mathbf{u}_f \frac{\left[\tau\right]_n}{Q} ds \\ w_h = \int_\Omega \left(\delta h \frac{\partial h}{\partial t} - \nabla(\delta h).H\mathbf{u}_f \right) d\Omega + \int_{\partial\Omega} \delta h H u_n ds \end{cases} \tag{9}$$

$\delta \mathbf{u}_f$, δh are the Galerkin test functions, [*r*] the viscous constraints tensor given by:

$$\left[\tau\right] = 2\upsilon \begin{bmatrix} \dfrac{\partial u_f}{\partial x} & \frac{1}{2}\left(\dfrac{\partial u_f}{\partial y} + \dfrac{\partial \upsilon_f}{\partial x}\right) \\ \frac{1}{2}\left(\dfrac{\partial u_f}{\partial y} + \dfrac{\partial \upsilon_f}{\partial x}\right) & \dfrac{\partial \upsilon_f}{\partial y} \end{bmatrix} \tag{10}$$

$\left[\tau\right]_n$ the edge friction tensor and *S* is the Neumann stress boundary.

The discretised model associated to the set of equations (9) is obtained by using a $P_{2\_}P_1$ element for finite element approximation [10]-[13] (Fig 3).

3. MODELLING OF THE FLUID-RIGID BODY INTERACTION

A rigid body (Particle) in a non-uniform time-dependent fluid flow responds to the forces imposed by fluid velocity and body forces. A delineation of the particle motion and the interaction between the particle and fluid depends on the correct calculation of the forces. Thus, the Lagrangian equation of a rigid body moving in an arbitrary 2D flow are governed by the Newton's laws, that includes in the force balance Buoyancy, pressure gradient of ambient flow, added mass and drag forces.

$$\begin{cases} \dfrac{d\mathbf{OG}}{dt} = \mathbf{u}_p \\ m_p \dfrac{d\mathbf{u}_p}{dt} = \mathbf{F}^p \\ I_p \dfrac{d\omega_p}{dt}\mathbf{k} = \mathbf{F} \times \mathbf{GM} \end{cases} \tag{11}$$

m_p is the particle mass, $\mathbf{u}_p = (u_x, u_y)$ is the translation velocity vector of the particle mass-center is the moment of inertia, I_p, is the moment of inertia, ω_p is the angular velocity and $\mathbf{F}^p = \left(F_x^p, F_y^p\right)$ are total forces acting on the particle. *G* is the position of the center of mass and *M* the position of the center of pressure where are concentrate hydrodynamic forces.

3.1. FORCES ACTING ON THE RIGID BODY

The essence of the present simulations is to treat the total forces imposed on the fluid by particles of non spherical form, which can be obtained from the particle equation of motion—equation (11), as a body force. Since the effect of the particle on the fluid is to generate a disturbance in the fluid velocity field, it would also be appropriate to couple the particle effect into the fluid in terms of a velocity disturbance. Since there is usually a relative motion between the particle and the fluid, and also the particle can be rotating, the liquid flow around individual particle leads to local variations in pressure and shear stress. Here fives different contributions for the interaction force term can be taken into account, a friction (Stokes drag) force , an added mass force, the pressure gradient force, the Basset history force, and the Buoyancy force, leading to the following approximation for the force of interaction, governed by the Maxey-Riley equation [12]:

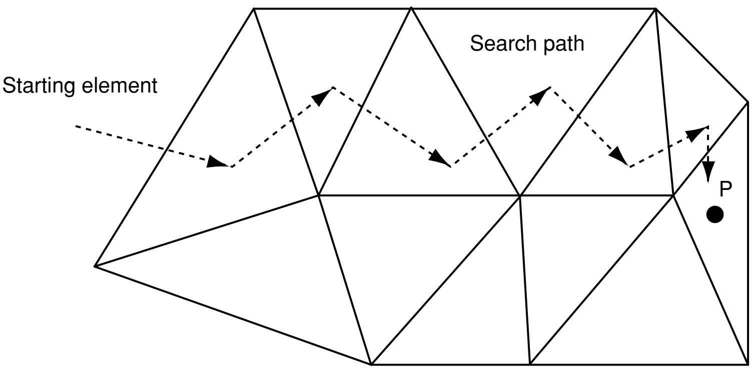

Figure 5 Sketch of the search line algorithm.

$$\mathbf{F}^p = \mathbf{F}_{Drag} + \mathbf{F}_{Pg} + \mathbf{F}_{Vm} + \mathbf{F}_{Gr} + \mathbf{F}_{Bh} \quad (12)$$

where:

- $\mathbf{F}_{Drag} = -\frac{1}{2}\mathrm{C}_d m_f [\mathbf{u}_p - \mathbf{u}_f]$, is the Stokes drag term, where C_d is the drag coefficient which depends on the particle form $\mathbf{u}_p$ and m_f are the velocity and the masse of the particle respectively, whereas, $\mathbf{u}_f$ is the fluid velocity.
- $\mathbf{F}_{pg} = m_f \frac{D\mathbf{u}_f(t)}{Dt}$, is the pressure gradient related to fluid acceleration,
- $\mathbf{F}_{Vm} = -\frac{1}{2} m_p \left[\frac{d\mathbf{u}_p(t)}{dt} - \frac{d\mathbf{u}_f(t)}{dt}\right]$, is the virtual mass contribution (or added mass effect). It arises because acceleration of the particle requires acceleration of the fluid surrounding the body, and the pressure gradient force accounts for the acceleration of the displaced fluid (i.e. the force a fluid particle of the same size would experience in absence of the particle).
- $\mathbf{F}_{Bo} = -g(m_f - mp)\mathbf{k}$, is the Buoyancy force,
- $\mathbf{F}_{Bh}$, is the Basset history term. It adjusts the particle acceleration by taking into account the past acceleration on the particle motion, including the effect of the conditions that prevailed during development of the flow. In the two-dimensional study, where gravity is absent, this term may be negligible, especially because the particle is not released from rest but with an initial velocity corresponding to the ambient fluid velocity.

Elsewhere, in the above equation, the Buoyancy term: $\mathbf{F}_{Bo}$ as well as the pressure gradient $\mathbf{F}_{pg}$ because of the 2D horizontal space problem. We also neglect neglect the Faxen correction term, which becomes significant only in the event of large curvature in the velocity profile.

We notice that the derivative *d/dt* is used here to denote a time derivative following the moving particle, thus:

$$\frac{d}{dt}=\frac{\partial}{\partial t}+(\mathbf{u}_f.\nabla) \tag{13}$$

and the derivative *D*/*Dt* denotes the time following a fluid element, so that

$$\frac{d}{dt}=\frac{\partial}{\partial t}+(\mathbf{u}_p.\nabla) \tag{14}$$

is the fluid acceleration as observed at the instantaneous center of the particle. In what follows, we assume the following assumption :

$$\frac{D\mathbf{u}}{Dt}=\frac{d\mathbf{u}}{dt}$$

then from (11), the set of equations for calculating the particle location, and the linear and angular velocities in vector form becomes:

$$\begin{cases} \dfrac{d\mathbf{OG}}{dt}=\mathbf{u}_p \\ m_p\dfrac{d\mathbf{u}_p}{dt}=A\left[\dfrac{\partial\mathbf{u}_f}{\partial t}+(\mathbf{u}_f.\nabla\mathbf{u}_f)\right]+B\,[\mathbf{u}_f.\nabla\mathbf{u}_p]+C[\mathbf{u}_f-\mathbf{u}_p] \\ I_p\dfrac{d\omega_p}{dt}\mathbf{k}=\left(A\left[\dfrac{\partial\mathbf{u}_f}{\partial t}+(\mathbf{u}_f.\nabla\mathbf{u}_f)\right]+B\,[\mathbf{u}_f.\nabla\mathbf{u}_p]+C\,[\mathbf{u}_f-\mathbf{u}_p]\right)\times\mathbf{GM} \end{cases} \tag{15}$$

where $A=\frac{2}{3}\left(\frac{1}{2}m_p+m_f\right), B=-m_p$ and $C=-\frac{3}{4}m_pC_d$. C_d is commonly expressed over the entire Reynolds number spectra as:

$$C_d=\ \beta S_{eff}\left\|\mathbf{u}_f-\mathbf{u}_p\right\|(\mathbf{u}_f-\mathbf{u}_p) \tag{16}$$

where S_{eff} is the particle area normal to the direction of the drag force. It changes with the incidence angle α between relative velocity ($\mathbf{u}_f-\mathbf{u}_p$) and particle major axis direction and is determined by

$$S_{eff}=\pi l^2\sqrt{\sin^2\alpha+(4L/\pi)^2\cos^2\alpha} \tag{17}$$

L and *l* are the length and large of the particle respectively, β is the drag coefficient which is determined by [5]:

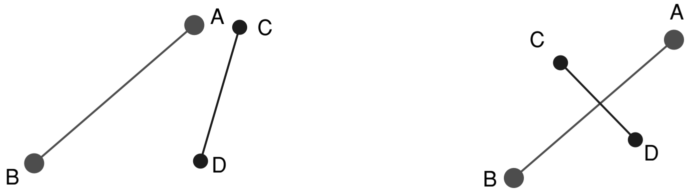

Figure 6 Schematic presentation of the sweep line algorithm.

$$\frac{\beta}{k_2}=\frac{24}{k_1k_2R_e}(1+0.1118(k_1k_2R_e^{0.6567})+\frac{0.4305}{1+0.3305/(k_1k_2R_e)}$$
$$k_1=(d_n/3d+2/3\Theta^{0.5})^{-1};\ k_2=10^{1.8148(-\log\Theta)^{0.5743}} \tag{18}$$

where $d_n=\sqrt{4S_{eff}/\pi}$ is the equal projected area circle diameter, Re is the particle Reynolds number, defined on the basis of the relative velocity between particle and the surrounding fluids and equal-volume sphere diameter; $\Theta=s/S$ is the particle sphericity, where s is the surface of a sphere having the same volume as the particle and S is the actual surface area of the non spherical particle. The equations (15) calculate the translation velocity vector at the center of mass. The velocity vector for an arbitrary point M on a rotating and translating rigid body is then:

$$\mathbf{u}\,(M,T)=\mathbf{u}_p(t)+\omega_p\times\mathbf{GM} \tag{19}$$

G is the position of the center of mass.

As the center of pressure P does not coincide with the center of mass at nonzero incidence angles, hydrodynamics forces described above, which act at the center of pressure P rather than at the center of mass G, will give rise to a torque on the particle. Thus, the distance $//d//$ between G and P is given by (see [5]):

$$\|d\|(G,M)=0.1L(1-e^{3(1-L/l)})\left|\cos^3\alpha\right| \tag{20}$$

4. PARTICLE-PARTICLE COLLISIONS FOR RIGID BODIES

The inter-particle collision depends mainly on the body motion and on it's size, and on the fluid dynamic transport effects, that is, drag and pressure gradient forces. However, the calculation of the particle linear (u_p) and angular (ω_p) velocities change caused by an inter-particle collision is related to the following assumptions:

- Bodies are assumed to be rigid, and no shape deformation of the bodies during the collision process is considered.
- Bodies move in a two-dimensional plane. There is no linear movement and rotation in a third-dimensional space.
- Collisions are instantaneous, and during the collision calculation, only impulsive forces are considered.

Thus, in what follows and for the sake of simplicity, we will present only the case of contact between two bodies. This approach can be extended to more complex cases by treating them as series of contacts between two bodies. Suppose a vertex on particle *A* is colliding into an edge of particle *B* at the point *M*, (see Fig. 4)

GA and *GB* are the centers of mass of particles A and B respectively $\mathbf{d}_A$, and $\mathbf{d}_B$ are the distances vectors from *GA* and *GB* to point *P* for particles *A* and *B* respectively. **n** is the normal vector to edge of particle B. Two solutions exist to model collision between particles. The first one is to introduce an impulsion force directly in equation (15) when collision occurs. That means to know details about the materials of the particles, their exact geometry and how they deform under stress, etc. This approach is recommended when one wants to know more details on collisions but require lot of *CPU* time cost. The second approach consists on modifying the directions and magnitudes of velocities vectors after the impact of the particles by introducing the concept of impulse [5]. An impulse is the change in momentum of an object when a force is applied over a very brief period of time. We imagine that during the collision there is a force acting for a very brief period of time. If you integrate that force over that brief time, you get the impulse. This is resumed on the following equation:

$$\left(\mathbf{u}_A^{final} - \mathbf{u}_B^{final}\right)\mathbf{n} = -\,\mathbf{e}(\mathbf{u}_A^{intial} - \mathbf{u}_B^{initial})\mathbf{n} \tag{21}$$

$\mathbf{u}_A^{intial}, \mathbf{u}_B^{intial}, \mathbf{u}_A^{final}, \mathbf{u}_B^{final}$ are the initial and final translational velocities of particles *A* and *B* respectively. This equation says that the velocity at which the particles fly apart is proportional to the velocity with which they were coming together. The proportionality factor is the parameter *e*. The elasticity contact parameter *e* is equal to 0 if the collision is inelastic and 1 if the collision is perfectly elastic. Our goal is to identify the collisions between particles and to know approximately how stopping are formed in river for example when tree's branch contact bridge piers and perturb the flow. Because of its relative simplicity, our choice is focussed for the second method. If we neglect the friction force (which is parallel to the edge) and assume that the only force during the collision is in direction perpendicular to the edge, which is given by the vector **n**, and using the equation (21), the pre- and post-velocities are related by the following equations:

$$\begin{cases} \mathbf{u}_A^{final} = \mathbf{u}_A^{intial} + \dfrac{\alpha_{AB}}{m}\mathbf{n}; \quad \omega_A^{final} = \omega_A^{intial} + \mathbf{d}_A \times \dfrac{\alpha_{\mathrm{AB}}}{I_A}\mathbf{n} \\ \mathbf{u}_B^{final} = \mathbf{u}_B^{intial} - \dfrac{\alpha_{AB}}{m}\mathbf{n}; \quad \omega_B^{final} = \omega_B^{intial} + \mathbf{d}_B \times \dfrac{\alpha_{AB}}{I_B}\mathbf{n} \end{cases} \tag{22}$$

ω_A^{intial}, ω_B^{intial}, ω_A^{final}, ω_B^{final} are respectively the initial and final angular velocities of particles *A* and *B*. is I_A is the moment of inertia od the particle A, α_{AB} is the impulse parameter and it's expression is calculated by introducing equations (21) into equation (22) and assuming that $\mathbf{n}.\mathbf{n} = 1$. The expression of α_{AB} is then:

$$\alpha_{AB} = \frac{-(1-e)(\mathbf{u}_A^{intial} - \mathbf{u}_B^{initial})\mathbf{n}}{1/m_A + 1/m_{AB} + (\mathbf{d}_A \times \mathbf{n})^2/I_A + (\mathbf{d}_B \times \mathbf{n})^2/I_B} \tag{23}$$

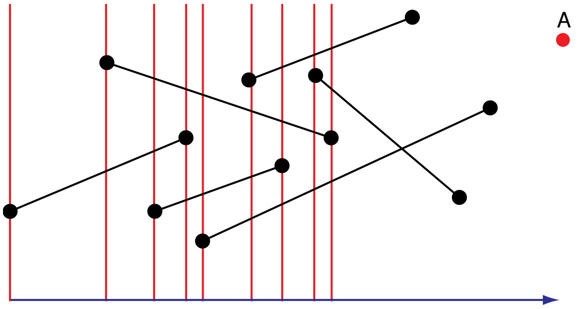

Figure 7 Sweep direction.

We can use this last equation (22) for calculating collisions with a wall by assuming the mass of the wall is infinite, hence it becomes:

$$\alpha_{AB} = \frac{-(1-e)\mathbf{u}_A^{intial}\mathbf{n}}{1/m_A + 1/m_{AB} + (\mathbf{d}_A \times \mathbf{n})^2 / I_A} \tag{24}$$

Notice that all those equations calculate velocities at center of mass. For an arbitrary point C in particle, one uses the following equation to estimate the velocities:

$$\mathbf{u}_G^C = \mathbf{u}_G + \ \omega \times \mathbf{d}_G^C \tag{25}$$

5. COMPUTATIONAL PROCEDURE

We discretize the system of equations (15) using a finite element procedure, with an additional constraint given by equation (19) and by taking into account of inter-particles collisions.

5.1. THE TIME-INTEGRATION SCHEME

The time-integration scheme consists on using the Runge-Kutta scheme with an adaptive step size control algorithm. The goal of this adaptive time discretization step control is to achieve some predetermined accuracy in the solution with minimum *CPU*-time. Thus, after writing the set of equations (15) in the form:

$$\frac{\partial y}{\partial y} = f(t,y) \tag{26}$$

the basic idea of the Runge-Kutta with adaptive step size control algorithm, may be given as follow:

- Combine a 4^{th} order Runge-Kutta method and a 5^{th} order Runge-Kutta method;
- Estimate error for each local step combining solution of the 4th and 5th order;
- If the local truncation error is larger than a given accuracy, reduce the step size;
- If the local truncation error is smaller than a given accuracy, increase the step size

For the 5^{th} order Range-Kutta, the solution is:

Table 1 Numerical values

i	a_i	$a_i – b_i$	c_i	d_{ij}
1	$\frac{16}{135}$	$\frac{1}{360}$	0	0
2	0	0	$\frac{1}{4}$	$\frac{1}{4}$
3	$\frac{6656}{12825}$	$\frac{-128}{4275}$	$\frac{3}{8}$	$\frac{3}{32}, \frac{9}{32}$
4	$\frac{28561}{56430}$	$\frac{-2197}{75240}$	$\frac{12}{13}$	$\frac{1932}{2197}, \frac{7200}{2197}, \frac{7296}{2197}$
5	$\frac{19}{50}$	$\frac{1}{50}$	1	$\frac{439}{216}, -8, \frac{3680}{513}, \frac{-845}{4104}$
6	$\frac{2}{55}$	$\frac{2}{55}$	$\frac{1}{2}$	$\frac{-8}{27}, 2, \frac{3544}{2565}, \frac{1859}{4104}, \frac{-11}{40}$

$$y(t+\Delta t) = y(t) + \sum_{i=1}^{6} a_i S_i \tag{27}$$

For the 4^{th} order Range-Kutta, the solution is:

$$\overline{y}(t+\Delta t) = y(t) + \sum_{i=1}^{6} b_i S_i \tag{28}$$

where Δt is the time discretisation step, y is the function to be approximated, S_i are computed as regular Range-Kutta methods:

$$S_i = \left(f(t + c_i, y + \sum_{i=1}^{6} d_{ij} S_j \right) \Delta t \tag{29}$$

f is the right hand side, the parameters introduced in this method are given in the table 1, where the error reads:

$$err = y(t+\Delta t) - \overline{y}(t+\Delta t) = \sum_{i=1}^{6} (a_i - b_i) S_i \tag{30}$$

Thus, the adaptative time discretisation step Range-Kutta algorithm is as follow:

1. Input f, $t_{0,}$ $t_{last,}$ Δt, $y_{\ 0,}$ tol

$$\begin{cases} \Delta t_{\min} = 0 \\ \Delta t_{\max} = (t_{last} - t_0) / 16 \\ y(1) = y_0 \\ t = t_0 + \Delta t \end{cases}$$

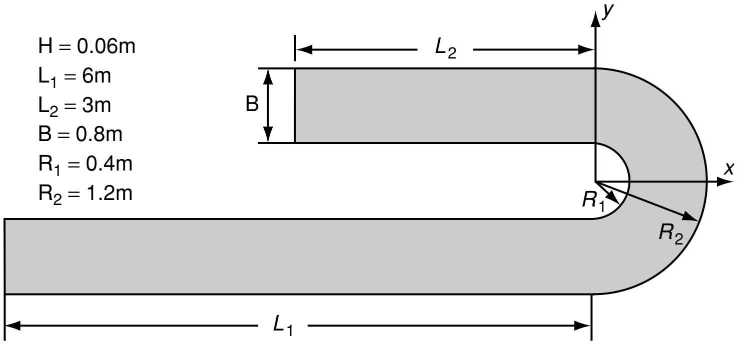

Figure 8 Geometry of the bend tube.

```
2.  while ($t < t_{last}$)
        $\begin{cases} \text{Calculate } S_i \ as\ (29) \\ \text{Calculate } y(i)\ as\ (27) \\ \text{Calculate error } err\ as\ (30) \end{cases}$

3.  Estimate optimal time step size as: $\Delta t_{opt} = 0.8\Delta t\left(\frac{tol}{|err|}\right)^{0.2}$

•   if $|e| > tol$ , then $\Delta t = Max\left(\Delta t_{opt}, \frac{1}{2}\Delta t, \Delta t_{\min}\right)$ and goto 1

•   else if $|e| \preceq tol$
        $\begin{cases} \Delta t = Min\left(\Delta t_{opt}, \frac{3}{2}\Delta t, \Delta t_{\max}\right) \\ t = t + \Delta t, i = i + 1 \text{and go to 1} \end{cases}$
```

Remark 1 *In the above algorithm,* $\bar{y}(t + \Delta t)$ *is never calculated*

5.2. GENERAL ALGORITHM

The computational algorithm mainly consists of the following steps:

1. Flow simulating
   - Solve Saint-Venant flow equations
   - Set initialization conditions of particles and time step size Δt
2. particle simulating
   - Calculate all forces and torques acting on the particles
   - Solve the basic equations of motion using Eqs. (12) Over Δt
3. Update particles positions and orientations.

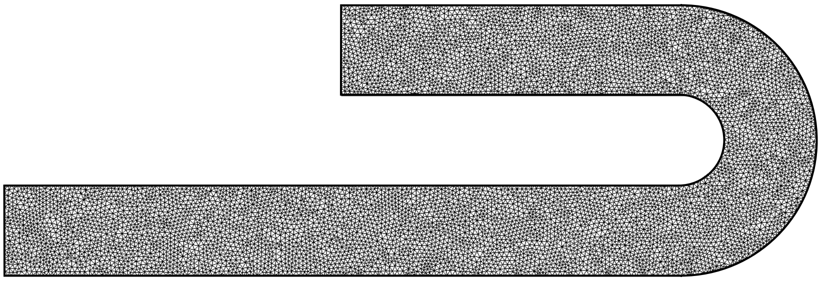

Figure 9 Mesh of the bend tube.

5.3. SEARCH ALGORITHM FOR INTERPOLATION

To get information for each particle moving through the finite element mesh and solve numerically equations (15) and (21) – (26) at each time step, one need to interpolate data associated with unstructured grids. This interpolation can take a non-negligible portion of total CPU-time, especially for large applications. To optimize this *CPU*-time, we use a fast neighbor-to-neighbor technique [6] that accelerates speedup by a factor 5 compared to usual techniques.

Consider an unstructured finite element mesh (see Fig. 5), as well as a point P with coordinates (*xp, yp*). A straightforward way to determine if the point *M* is inside a given **element i** is to determine the shape-function values of P (*Ni*) with respect to the coordinates of the points belonging to **element i** as follow:

$$\begin{cases} X_P = \sum N_i X_i \\ Y_p = \sum N_i Y_i \end{cases} \tag{31}$$

where N_i are the shape functions, that verify:

- for a 3 nodes triangle:

$$\begin{cases} N_1 = \Big((y_3 - y_2)(x_2 - x_p) - (x_3 - x_2)(y_2 - y_p)\Big) / \; Area \\ N_2 = \Big((y_1 - y_3)(x_3 - x_p) - (x_1 - x_3)(y_3 - y_p)\Big) / \; Area \\ N_3 = \Big((y_2 - y_1)(x_1 - x_p) - (x_2 - x_1)(y_1 - y_p)\Big) / \; Area \\ \quad Area = (x_3 - x_2)(y_1 - y_2) - (x_1 - x_2)(y_3 - y_2) \end{cases} \tag{32}$$

- for six nodes finite element, we split each element into four triangles and evaluate each of these sub-triangles. Then the point *M* is in *element i*, if and only if:

$$Min\ (N_i, 1 - N_i) \geq 0\ \ \forall i \tag{33}$$

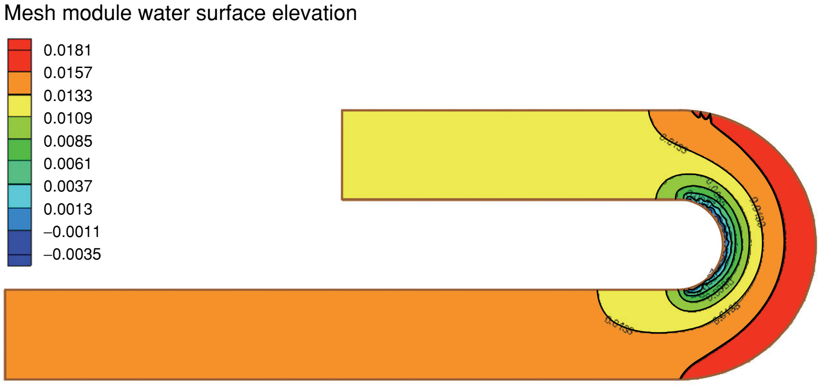

Figure 10 Water surface elevation.

Starting with "starting element", the algorithm jump from an element to its neighbor in the known grid. If the element into which x falls can be found in a few attempts (< 10), this procedure, will outperform all interpolation algorithms. The algorithm is as follows:

```
1.  Form the list of elements adjacent to element for the given mesh
2.  Loop over all particles
3.  Obtain good starting element "starting element"
4.  For "starting element", evaluate $Ni$
If ( $Min\ (N_{i,}\ 1 - N_i) \geq 0,\ i = 1, 3$
    Exit
    else
    Set "starting element" to neighbor associated with $Min(N_i)$
Goto 4
```

5.4. FAST LINE-SEGMENT INTERSECTION

For each time step, one must detect if the particles, considered as rigid bodies, intersect each other. Especially for large number of particles, this intersection's search can take a lot of total CPU-time and must be optimized to reduce computational time. A simple way to calculate intersection between two segments is to construct the convex hull of the segments. Two segments intersect if and only if the convex hull is a quadrilateral whose vertices alternate between the two segments. Nevertheless, this technique could be used only when the number of particles is small. In this section, we propose the sweep line algorithm witch is more complicated to program than the first one, but accelerates drastically speedup especially for large number of particles. Two segments AB and CD intersect (Fig. 6) if and only if :

- the endpoints A and B are on opposite sides of the line CD, and
- the endpoints C and D are on opposite sides of the line AB.

To test whether two points are on opposite sides of a line through two other points, we use a counter clockwise (*CCW*) test. *A* and *B* are on opposite sides of line *CD* if and only if exactly one of the two triples *A*, *C*, *D* and *B*, *C*, *D* is in counter clockwise order. To test the order of the triple *A*, *B*, *C*, we use the simple algorithm *CCW*:

1. CCW (A, B, C)

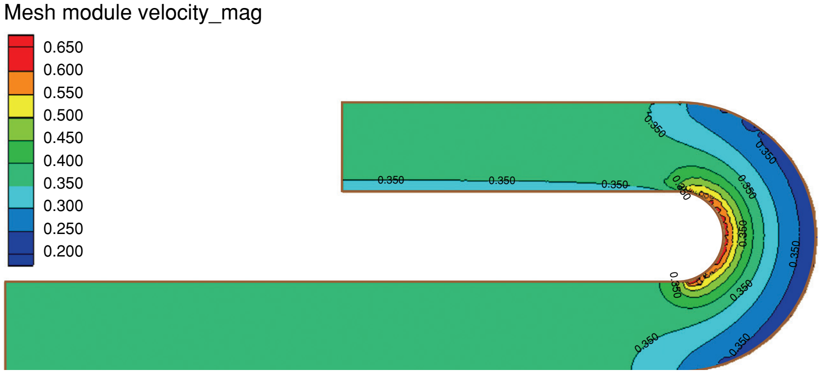

Figure 11 Isco-contour of the velocity magnitude.

```
if $(y_C - y_A)\ (x_B - x_A) > (y_B - y_A)\ (x_C - x_A)$
return TRUE
else
return FALSE
```

x_A, y_A, x_B, y_B, x_C, y_C are the coordinates of *A, B, C* points respectively. So the intersection algorithm of two segments is:

2. INTERSECTION (A, B, C, D)

```
If CCW (A, C, D) = CCW (B, C, D)
return FALSE
else if CCW (A, B, C) = CCW (A, B, D)
return FALSE
else
return TRUE
```

In order to detect whether there is an intersection in a set of more than just two segments, we use the sweep line algorithm (Fig. 7). The segments which the sweep line intersect form a "working set" of line segments, amongst which we test for intersections. If the working set is ordered from left to right, then only neighbors in the working set could intersect each other. As the sweep line moves down, segments enter and leave the working set, and the working set is maintained in left-to-right order. Every time the working set changes, we search again for possible intersections.

6. NUMERICAL TESTS AND RESULTS

6.1. VALIDATION OF THE HYDRODYNAMICAL MODEL

The validation of the hydrodynamical model (Eq.1) is carried out by using a bend tube as a benchmark test. The geometry used in this study was a 180° U-bend, with a diameter B, as shown in Fig. 8, where the mesh grid used in the present 8 calculations is given in Fig. 9. This geometry has been used in several experimental 9 and numerical investigations [7], [8], [9]. The computed water elevation, the velocity magnitude contours and iso-vectors are presented respectively in Fig. 10, Fig. 11 and Fig. 12. As illustrated in Fig. 10, in the external part of the curve, the pressure force is in balance with the centrifugal force, however, the

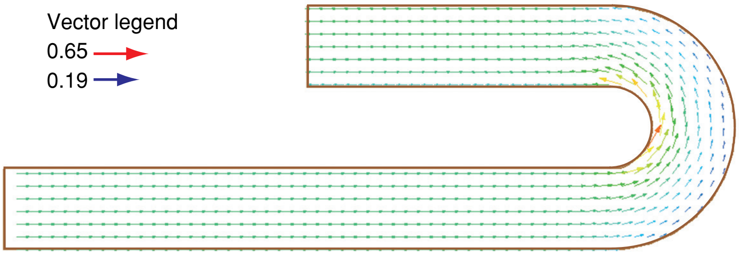

Figure 12 isco-vectors of the velocity.

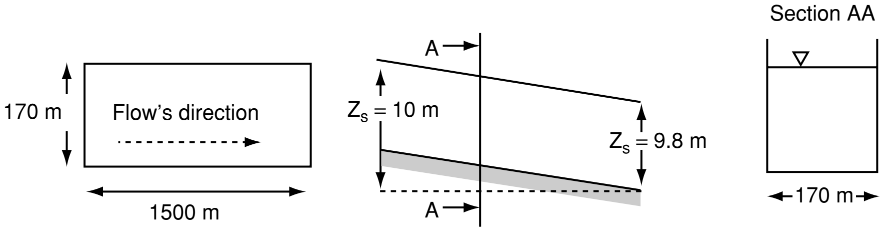

Figure 13 Geometry of the channel.

fluid particle velocity decreases. Consequently, the centrifugal force acting on it declines. The maximum velocity tends to be found closer to the inner curve (Fig. 11 and Fig. 12). It is also worthwhile to mention that for square-sectioned ducts or for non-circular ducts, the velocity vectors may give rise to secondary motion, which may be strong enough to generate small vortexes near the boundary layer.

6.2. TRANSPORT AND JAM FORMATION: RESULTS AND DISCUSSIONS

We consider the simulation of the motion of tree's branches in river and their collisions with four bridge piers. The tree's branches have rectangular shape with 1*m* large and 20 *m* long. The channel has 1500*m* length and 170 *m* large and inclined at slope 0.01 (Fig. 13). The obstacles have circle shape and their radius equal 30 *m*. The computational domain is discretized using triangular mesh for a total of 4468 finite elements and 9357 nodes with 15000 degrees of freedom (Fig. 14). At each node, the unknowns of the problem are the water surface elevation $h(x,t)$ and the velocities u and v. The physical parameters for the fluid are as follows: the density, $Q = 1{:}Kg/m^3$; the Reynolds number, $Re = 1000$ and the Manning coefficient's friction= $0.03m^{1/3}/s^{-1}$. Initial solution, $u = v = h = 0$; boundary conditions

- on AB, h = 10m,
- on CD, h = 9.80m,
- on AC and BD, a reflection boundary condition is imposed.

The simulation takes about 300 second CPU, on a Windows based *PC* with 2:8*Ghz* Pentium.

The computed water elevation, the velocity magnitude contours are presented respectively in Fig. 15 and Fig. 16. Snapshots of the positions and orientations of the particles are shown in Fig. 17. The total number of particles is 7 (the number of particles released is not restrictive). Initially the velocity of the particle is given $u_p^0 = 0$ and angular speed of the particle is given $\omega_p^0 = 0$. The time increment is $\Delta t = 0.1$ second, and the repulsion coefficient $e = 1.0$

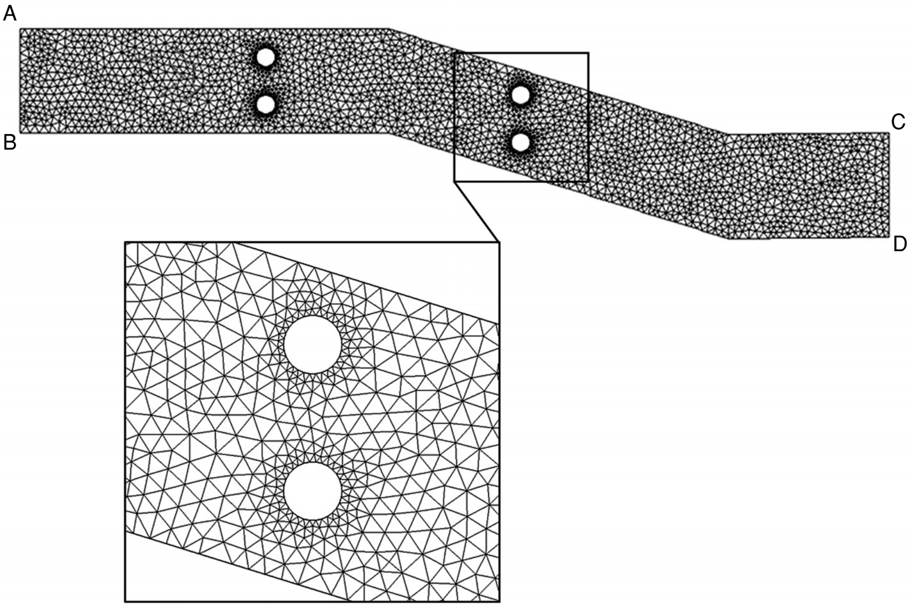

Figure 14 Mesh of the channel: two piers are located in the zoomed picture.

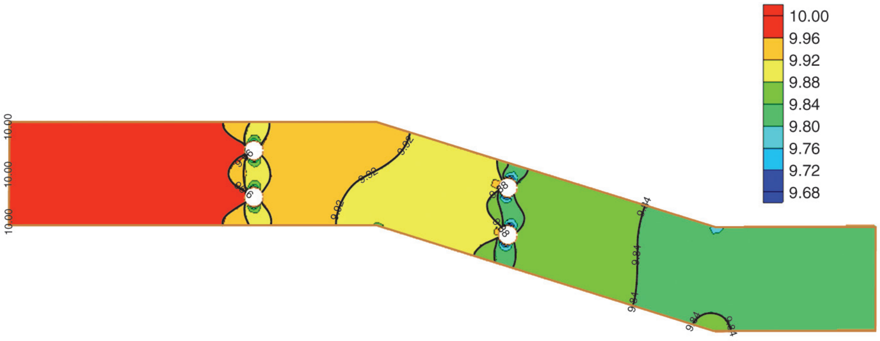

Figure 15 Water surface elevation.

Fig. 16 shows that complex flow around the piers reflects eddies in the hydrodynamical flow field and Fig. 17 shows the collisional effects from the particles accumulation around the piers, and that particles tend to orient with long axes parallel to flow in the center of the channel. Furthermore, Fig. 17 shows that interactions among multiple particles have tendency to both increase and decrease particle transport: particle-to-particle collisions entrain previously deposited particles, while immobile particles can obstruct moving ones, causing deposition, and formation of log jams.

Nevertheless, in order to examine interactions among hydraulics, channel geometry, transport distance and deposition of floating bodies, and to assess the potential effects of floating bodies on the dynamic morphology of streams, we need to understand where these

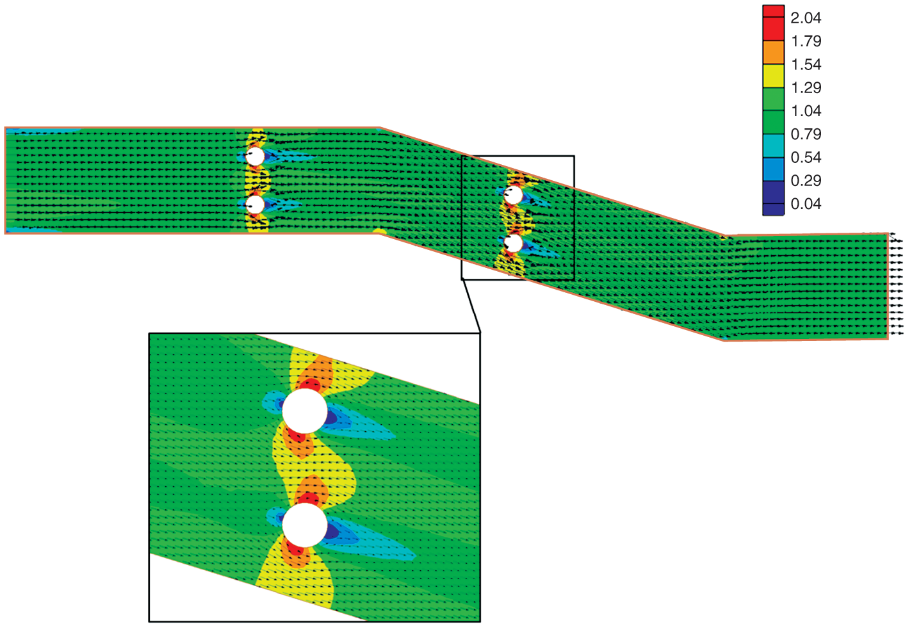

Figure 16 Velocity magnitude through de bridge.

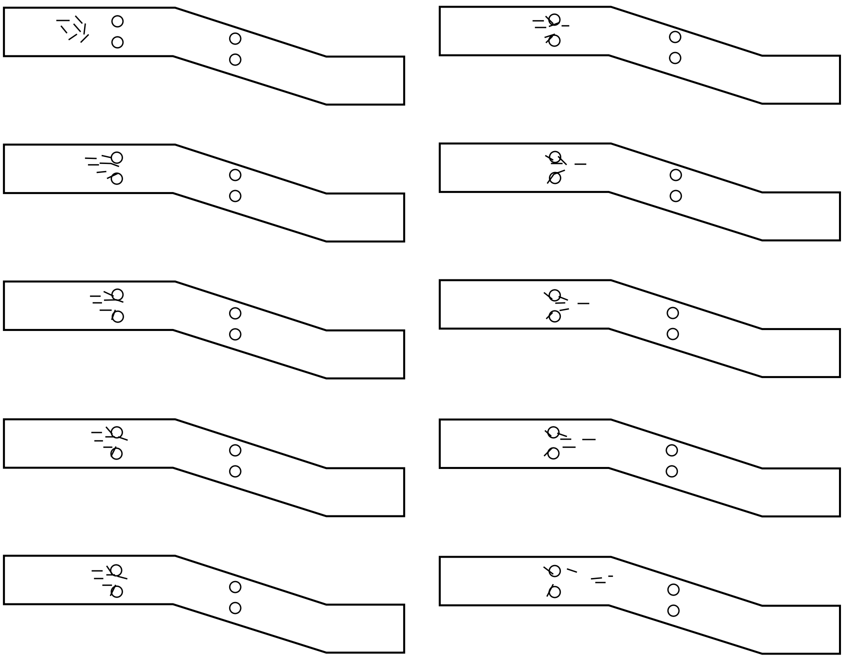

Figure 17 Particle transport passing through piers, located in the channel. Particles consist of floating long sticks.

bodies will deposit and how long they will remain there. Furthermore, the natural formation of debris jams, is a build-up of floating bodies of variable sizes and quantities. Thus it is necessary to simulate numerically the pathway of particles of different size and shapes, while taking into account channel geometries.

7. CONCLUSION

In this paper, a model is successfully derived to track the motion of non spherical rigid-bodies in a non-uniform flow field. The model reproduces successively the bodies pathway by including translation and rotation movements, and by taking into account of the Particle-Particle collisions. By using the fast neighbor-to-neighbor technique [6], the results showed the capacity of the model to reproduce the collisions between bodies and walls without increasing the computing time.

This model can help to identify potential problems for flood management in urban areas as well as to improve the accuracy of flow modelling of these areas. It can also be useful to researchers during investigations of the interaction between the floating debris, the flood wave propagation and the jam formation. Indeed, the information available consists mainly of after-flood observations of the remains of debris jams.

ACKNOWLEDGEMENTS

This study was supported by the ANR Program: project RIVES (Risque d'Inondation en Ville et Evaluation des scenarios

REFERENCES

[1] Hu H. H., Joseph D. D., Crochet M. J., “Direct simulation of fluid particles motions”, Theoret Comput. Fluid Dynamics, 3, 285–306, (1992)

[2] Glowinsky R., Pan T. W. and Perieux J. , “A fictitious domain method for Di-rechlet problem and applications”, Computer Methods in Applied Mechanics and engineering, 111, 283–303, (1994).

[3] Johnson A. , Tezduyar T. , “3-D simulation of fluid-rigid body interactions with the number of rigid bodies reaching 100”, Comp Meth Appl Mech Eng, 145, 301–321, (1997)

[4] Pan T. W., Glowinsky R., Joseph J.D., “Simulating the dynamics of fluid-cylinder interactions”, J. Zhejiang Univ. SCI, 6A(2), 97–109, (2005).

[5] Yin C., Rosendahl L., Kaer S. K., Sorensen H., “Modelling the motion of cylindrical particles in a nonuniform flow”, Chemical Engineering Science, 58, 3489–3498, (2003)

[6] Lohner R., “Robust, Vectorized Search Algorithms for Interpolation on Unstructured Grids”, Journal of Computational Physics, 118, 380–387, (1995)

[7] Chang, S.M., Humphrey, J.A.C. and Modavi, A. (1983) “Turbulent flow in a strongly curved U-bend and downstream tangent of square cross-section”, PCH Physical Hydrodynamics, Vol. 4, No. 3, pp.243–269.

[8] Choi, Y.D., Moon, C. and Yang, S.H. (1990) “Measurement of turbulent flow characteristics of square duct with a 180^o bend by hot wire anemometer”, in Rodi, W. and Ganic, E.N. (Eds.): Intl. Symp. on Engng. Turbulence Modelling and Measurements, Elsevier, pp.429–438.

[9] Johnson, R.W. and Launder, B.E. (1985), ‘Local Nusselt number and temperature field in turbulent flow through a heated square-sectioned U-bend’, Int. J. Heat and Fluid Flow, Vol. 6, No 3, pp.171–180.

[10] S.Hadji, (1997) G. Dhatt, “Asymptotic-Newton method for solving incompressible flows”, Int. J.Numer. Methods fluids, 44, 861–878, 1997.

[11] Kim, J. & Moin, P. 1985 “Application of a fractional-step method to incompressible Navier-Stokes equations”. J. Comput. Phys. 59, 308–323.

[12] Maxey M. R., Riley J. J., 1983 “Equation of motion for a small rigid sphere in a nonuniform flow”, Phys. Fluids 26, 883–889 (1983).

[13] Soulaimani A., Fortin M., Dhatt G., Ouellet Y., and Bertrand F., 1991. “ Finite element simulation of two- and three-dimensional free surface flows”. Comp. Meth. Appl. Mech. Eng., Volume 86, Issue 3. pp 265–296.