Dam break Smoothed Particle Hydrodynamic modeling based on Riemann solvers

L. Minatti[1] & A. Pasculli[2]
[1]Department of Civil and Environmental Engineering, University of Florence, Italy
[2]Department of Sciences, University of Chieti-Pescara, Italy

Abstract

The Smoothed Particle Hydrodynamic (SPH) method, originally developed during the 1970s to solve astrophysical problems, has shown many attractive features that have led many authors to try to use it to solve fluid flows problems. Its free surfaces tracking capabilities and its straightforward implementation of multi-materials interactions make it well suited for complex flows modeling.

The first part of the paper is devoted to a general overview of the method. In particular, a brief description of a recently proposed flux term, implemented in this paper, is introduced. The addition of the flux terms, based on a Riemann solvers approach, enhances the stability and smoothness of field variables, leading to more accurate pressure fields.

In order to test the effectiveness of the selected approach, in the second part of the paper, the Dam break problem is discussed. Furthermore, the classical Poiseuille flow problem is considered as well.

The numerical results are very satisfactory, exhibiting the effectiveness of the implemented approach regarding, at least, the selected kind of problems.

Keywords: SPH method, Riemann solvers, Dam break, Poiseuille flow.

1 Overview of the SPH method

The Smoothed Particle Hydrodynamic (SPH) method is a numerical technique that was initially developed during the 1970s to solve astrophysical problems (Monaghan [1]). It is a fully meshless particle method that is easy to code. Its meshless and Lagrangian nature make it very attractive for solving fluid flow problems where free surface boundary conditions and large strain rates may be

WIT Transactions on Engineering Sciences, Vol 69, © 2010 WIT Press
www.witpress.com, ISSN 1743-3533 (on-line)
doi:10.2495/AFM100131

involved. The computational domain is filled with particles carrying flow field information (e.g. pressure, velocity, density) and is capable of moving in space. Particles are the computational frame used in the method to solve the flow describing PDEs, as a grid or a mesh to calculate spatial derivatives is not needed.

We shall refer to 2D cases throughout the rest of the paper, even though all the assumptions and results can be extended to a 3D case with little effort. The key idea on which the method is based is the well-known use of a convolution integral with a Dirac delta function to reproduce a generic function *f($\underline{x}$)*:

$$\hat{f}(\underline{x}) = \int_D f(\underline{x}') \cdot \delta(\underline{x} - \underline{x}') d\underline{x}' \tag{1}$$

In the SPH method, the Dirac function is replaced by a "bell-shaped" kernel function *W* (it 'mimics' the Dirac delta function), and the generic function *f($\underline{x}$)* is reproduced with the following convolution integral:

$$\hat{f}(\underline{x}) = \int_{\Omega_x} f(\underline{x}') \cdot W(\underline{x} - \underline{x}') d\underline{x}' \tag{2}$$

The kernel function is chosen to be non-negative, even and with a support domain Ω_x (usually circular) whose radius is a multiple of a length *h*, named *smoothing length*. The kernel function is zero outside the support domain and the smoothing length serves as a scaling parameter for its arguments. It also has the property of converging to the Dirac function as the smoothing length approaches to zero.

The kernel function must satisfy some conditions in order to correctly reproduce functions up to a given order *k*, in a Taylor series expansion. Let us consider a 1D case where a function *f(x)* is approximated about point *x* by a Taylor series up to the order *k*:

$$f(x') = \sum_{i=0}^{k} \frac{f^{(i)}(x)}{i!} \cdot (x' - x)^i \tag{3}$$

If eqn (3) is substituted in eqn (2), the SPH approximation of function *f(x)* takes the form:

$$\begin{aligned} \hat{f}(x) &= \int_{\Omega_x} \sum_{i=0}^{k} \frac{f^{(i)}(x)}{i!} \cdot (x' - x)^i \cdot W(x - x') dx' = \\ &= \sum_{i=0}^{k} \frac{f^{(i)}(x)}{i!} \cdot \int_{\Omega_x} (x' - x)^i \cdot W(x - x') dx' \end{aligned} \tag{4}$$

If a correct reproduction of function *f(x)* is searched up to the order *k*, then the kernel must satisfy the following conditions:

$$M_i(x) = \int_{\Omega_x} (x - x')^i \cdot W(x - x') dx' = \delta^{i0} \quad i = 0 \ldots k \tag{5}$$

i.e. every kernel moment has to be equal to zero (except for the 0 order one that has to be equal to 1).

It is possible to obtain the expression for the SPH approximation of a function gradient by using eqn (2) and the Gauss-Green formula:

$$\nabla\hat{f}'(\underline{x}) = \int_{\Omega_x} \nabla f(\underline{x}') \cdot W(\underline{x}-\underline{x}')d\underline{x}' = \int_{\partial\Omega_x} f(\underline{x}') \cdot W(\underline{x}-\underline{x}') \cdot \underline{n}ds - \int_{\Omega_x} f(\underline{x}') \cdot \nabla' W(\underline{x}-\underline{x}')d\underline{x}' \qquad (6)$$

where:

- the kernel is differentiated with respect to the $\underline{x}'$ coordinate;
- $\underline{n}$ represents the normal to the support domain boundaries;

The first term of the RHS of eqn (6) can be zero if the support domain is not truncated by the computational domain boundaries, as the kernel is zero on the support domain boundaries. Another case when the term can be zero is when the support domain is truncated by the computational domain boundaries but there exists a boundary condition forcing the function $f(\underline{x})$ to vanish on the boundaries (it may be the case when $f(\underline{x})$ represents a velocity and a no-slip condition has to be enforced on the computational domain boundaries). If the first term of the RHS of eqn (6) is zero, then the SPH approximation of $f(\underline{x})$ gradient takes the form:

$$\nabla\hat{f}(\underline{x}) = -\int_{\Omega_x} f(\underline{x}') \cdot \nabla' W(\underline{x}-\underline{x}')d\underline{x}' \qquad (7)$$

Eqn (7) is often used, even when the first term of the RHS of eqn (6) does not vanish.

It is possible to find the conditions the kernel must meet in order to correctly reproduce the first derivative of a given function *f(x),* up the order k of its Taylor series expansion. They are similar to the conditions of eqn (5) and it can be shown that they are related. The reproducing conditions for a function first derivative are expressed as follows:

$$M_i^{'}(x) = \int_{\Omega_x} (x-x')^i \cdot W'(x-x')dx' = \delta^{i1} \quad i = 0...k \qquad (8)$$

The most frequently used kernels involve truncated Gaussian and spline curves. The kernel used in this paper is the cubic spline function with compact support, whose expression is as follows (see also the plot reported in Figure 1):

$$W(R) = \frac{2}{3} - R^2 + \frac{1}{2}R^3 \quad 0 \le R < 1$$

$$W(R) = \frac{1}{6}\,(2\text{-}R)^3 \quad 1 \le R \le 2 \qquad (9)$$

$$R = \frac{|\underline{x}\text{-}\underline{x}'|}{h}$$

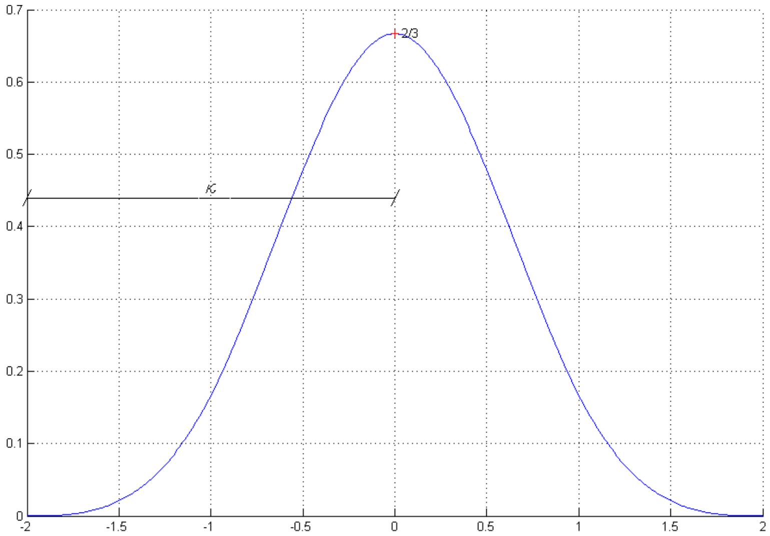

Figure 1: Third order cubic spline kernel plot.

The condition forcing the kernel to be non-negative makes it impossible to meet condition of eqn (5) for k = 2 and condition of eqn (8) for k = 3. Furthermore, the use of an even function as a kernel automatically satisfies conditions for exact reproduction of linear functions and second order function derivatives. Therefore, it is possible to correctly reproduce this kind of functions only with a non-negative kernel like the cubic spline.

2 Particle approximation related problems

The previous equations need to be evaluated in a discrete manner in order to develop a numerical technique from the theoretical framework shown above. In the SPH method, the discrete evaluation is made by means of the particle approximation. The continuous space is then replaced by a finite set of particles, each one of them carrying a mass, an area and other problem related information.

Hence, particle approximations of eqns (2) and (7) take the form:

$$\hat{f}_h(\underline{x}_i) = \sum_{j=1}^{n} f(\underline{x}_j) \cdot W(\underline{x}_i - \underline{x}_j) \cdot \Delta A_j \tag{10}$$

$$\nabla \hat{f}_h(\underline{x}_i) = -\sum_{j=1}^{n} f(\underline{x}_j) \cdot \nabla_j W(\underline{x}_i - \underline{x}_j) \cdot \Delta A_j \tag{11}$$

where:

- $\underline{x}_i$ and $\underline{x}_j$ represent the *i* and *j* particle positions in the given frame of reference;
- $\varDelta A_j$ represents the tributary area associated with particle *j*;

- Summations are extended to all the particles located within the support domain of particle *i*;

The ability of the kernel to exactly reproduce a function and its derivative up to a certain order vanishes when the particle approximations shown in eqns (10) and (11) are performed. This means that the particle approximations of the consistency conditions of eqns (5) and (8), which are shown below for a 1D case:

$$m_p(x_i) = \sum_{j=1}^{n} (x_i - x_j)^p \cdot W\left(x_i - x_j\right) \cdot \Delta x_j = \delta^{p0} \quad p = 0...k \tag{12}$$

$$m_p{}'(x_i) = \sum_{j=1}^{n} (x_i - x_j)^p \cdot W'\left(x_i - x_j\right) \cdot \Delta x_j = \delta^{p1} \quad p = 0...k \tag{13}$$

are no longer exactly satisfied. The afore-mentioned problem is often referred to as the *particle inconsistency* problem.

Another issue related -but not limited to the particle inconsistency problem arises when particle approximation of eqn (11), for evaluating a function gradient, is performed at a particle whose support domain overlaps with the computational domain boundaries. As the first term of the RHS of eqn (6) is often neglected, the particle approximation shown in eqn (11), besides suffering from the particle inconsistency related errors, suffers from the ones arising from this further approximation. This problem is often referred to as the *particle deficiency* problem and it may cause relevant errors both in evaluating a function derivative close to boundaries and in imposing a boundary condition on a field variable.

Many authors have proposed corrective strategies to tackle particle approximation related problems. Randles and Libersky [2] used ghost particles to treat a symmetrical surface boundary condition. Ghost particles have also been used in a various manners for particle approximations near boundaries by Takeda *et al.* [3], Morris *et al.* [4] and Ferrari *et al.* [5].

An approach that tries to overcome these issues without the use of any kind of ghost particles has been proposed by Liu *et al.* [6]. The method proposed by the authors (briefly named RKPM) consists in multiplying the kernel function by corrective coefficients in the particle approximation, in order to restore the particle approximation of the consistency conditions of eqns (12) and (13). By using the RKPM method, it is possible to find corrective coefficients to attain higher order consistency. The computational cost of the SPH simulation increases as it is necessary to solve least squares problem and to invert symmetrical square matrixes.

Another corrective approach is the one proposed by Liu and Liu [7]. The authors use the Taylor expansion series to offset and correct the standard particle approximations of eqns (10) and (11) for reproducing a function and its first derivative. Computational cost is increased even in this case, as it is necessary to perform matrix inversion in order to obtain a corrective matrix for the standard SPH approximation.

Nevertheless, it is possible to obtain realistic and accurate simulations with SPH despite these issues. The main reasons relies on the fact that the momentum equation can be set up so that the interaction terms between each pair of particles are symmetrical, thus allowing for momentum conservation.

3 SPH for fluid flow problems

In this paper, uncompressible fluids are treated using the weakly compressible fluid approximation. This means that pressure is not obtained by solving a Poisson equation, as in the uncompressible case but using a stiff equation of state instead. The equation used here has successfully been employed before by Monaghan [8] and Ferrari *et al.* [5], among others, and computes the particle pressure p_i this way:

$$p_i = k_0 \cdot \left(\left(\frac{\rho_i}{\rho_0} \right)^7 - 1 \right) \tag{14}$$

where:

- k_0 is a reference pressure;
- ρ_i represents the density at particle *i*;
- ρ_0 represents a reference density of the fluid when relative pressure is zero.

The k_0 value must be chosen in order to have a speed of sound, which is at least ten times higher than the highest fluid velocity involved in the problem: in this way it is possible to limit the density variations to around 1% of reference density and to not introduce prohibitively small time steps (see Monaghan [8]). Finally, please note that if eqn (14) is employed then particle sound speed c_i has the following expression:

$$c_i = \sqrt{\frac{7k_0}{\rho_0} \cdot \left(\frac{\rho_i}{\rho_0} \right)^6} \tag{15}$$

The particle approximation used for the continuity equation is the following:

$$\frac{D\rho_i}{Dt} = \rho_i \sum_{j=1}^{n} \frac{m_j}{\rho_j} \cdot (\underline{v_i} - \underline{v_j}) \cdot \nabla_i W_{ij} \tag{16}$$

where:

- $\underline{v}_j$ represents the velocity for particle *j*;
- m_j represents the mass of particle *j*.

In eqn (16), velocity derivatives are calculated slightly differently from what would be suggested by equation (11). It is possible to show that with this modification, involving velocity differences in the equation instead of just the term $\underline{v}_j$, it is possible to exactly reproduce gradients of constant velocity fields. It can also be shown that using equation (15) is like using a modified kernel, which satisfies the particle approximation for consistency conditions (13) up to the 0th order. The use of such equation, which has proved to be successful in many cases, enhances the accuracy in velocity field divergence calculations, especially near boundaries (see Liu and Liu [9]).

In this paper, we also selected an approach proposed by Ferrari *et al.* [5], which uses a Riemann solvers based modification of the continuity equation where a central Rusanov flux term is added to the equation. Adding the Rusanov central flux term to eqn (16) leads to the following form for the continuity equation:

$$\frac{D\rho_i}{Dt} = \rho_i \sum_{j=1}^{n} \frac{m_j}{\rho_j}\cdot(\underline{v_i}-\underline{v_j})\cdot\nabla_i W_{ij} + \sum_{j=1}^{n}\frac{m_j}{\rho_j}\cdot(\underline{n}_{ij}\cdot\nabla_i W_{ij})\cdot c_{ij}\cdot(\rho_j-\rho_i) \qquad (17)$$

where:

- $\underline{n}_{ij}$ is the unit vector pointing from particle *i* towards particle *j*;
- c_{ij} represents highest sound speed between particle *i* and *j*.

The addition of the flux terms results in more accurate and smoother density field calculations, which lead to more accurate pressure fields. The correction acts also as a penalty term for density fluctuations helping in enforcing weak compressibility condition.

Finally, the particle approximation that has been used throughout the computations for momentum equation is the following one:

$$\frac{D\underline{v}_i}{Dt} = \underline{f_i} + \sum_{j=1}^{n} m_j \cdot \left(\frac{\underline{\underline{\sigma_i}}}{\rho_i^2}+\frac{\underline{\underline{\sigma_j}}}{\rho_j^2}\right)\cdot\nabla_i W_{ij} \qquad (18)$$

where:

- $\underline{f_i}$ is the force/mass ratio of the external forces for particle *i*;
- $\underline{\underline{\sigma}}_i$ is the stress tensor at particle *i*.

It is possible to show that this form of momentum equation does not satisfy the particle consistency conditions of eqn (13), neither for the 0th order: therefore, it cannot be used to reproduce gradients of constant stress fields exactly. However, if the terms in the summation of the right hand side of eqn (18) are interpreted as forces (per unit of mass) exchanged by each pair of interacting particles, it is easy to notice that they are symmetrical, thus allowing for particle momentum conservation (when no external forces are present).

3.1 Inviscid flows computations

In order to simulate the behavior of inviscid flows we use the following isotropic constitutive equation for the stress tensor:

$$\underline{\underline{\sigma}}_i = -p_i \cdot \underline{\underline{I}} \qquad (19)$$

where:

- $\underline{\underline{I}}$ is the unit tensor.

By using eqn (19), the momentum eqn (18) takes this form:

$$\frac{D\underline{v}_i}{Dt} = \underline{f}_i - \sum_{j=1}^{n} m_j \cdot \left(\frac{p_i}{\rho_i^2}+\frac{p_j}{\rho_j^2}\right)\cdot\nabla_i W_{ij} \qquad (20)$$

As it has been pointed out in paragraph 3, the momentum equation conserves momentum exactly, when there are no external forces acting on the system.

3.2 Viscous flows computations

In order to simulate the behavior of a Newtonian viscous fluid at a laminar regime, we use the well-known constitutive equation:

$$\underline{\underline{\sigma}}_i = -p_i \cdot \underline{\underline{I}} + \mu \cdot \underline{\underline{\varepsilon}}_i \tag{21}$$

where:

- $\underline{\underline{\varepsilon}}_i$ is the strain rate tensor at particle *i*;
- μ is the dynamic viscosity of the fluid.

We then used the following expression for the particle approximation of the strain rate tensor components (see Liu and Liu [9]):

$$\varepsilon_i^{\alpha\beta} = \sum_{j=1}^{n} (v_j^{\beta} - v_i^{\beta}) \cdot \frac{\partial W_{ij}}{\partial x_i^{\alpha}} \cdot \Delta A_j + \sum_{j=1}^{n} (v_j^{\alpha} - v_i^{\alpha}) \cdot \frac{\partial W_{ij}}{\partial x_i^{\beta}} \cdot \Delta A_j - \left(\sum_{j=1}^{n} (\underline{v_i} - \underline{v_j}) \cdot \nabla_i W_{ij} \cdot \Delta A_j \right) \delta^{\alpha\beta} \tag{22}$$

By using eqn (21), the momentum eqn (18) takes this form:

$$\frac{D\underline{v}_i}{Dt} = \underline{f}_i - \sum_{j=1}^{n} m_j \cdot \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij} + \sum_{j=1}^{n} m_j \cdot \left(\frac{\mu \cdot \underline{\underline{\varepsilon}}_i}{\rho_i^2} + \frac{\mu \cdot \underline{\underline{\varepsilon}}_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij} \tag{23}$$

4 Numerical examples

In the following paragraphs we show numerical examples regarding the solution of two test cases, one for inviscid flow and the other for a laminar viscous flow.

4.1 Water column

The test case regards the solution of the classical Water column (or Dam break) problem where a column of water collapses under the effect of gravity and a breaking wave impinging on a vertical wall is created thereafter.

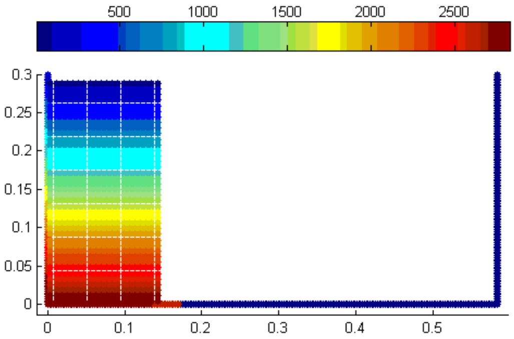

Figure 2: Initial particles distribution for the SPH solution of the water column problem.

The SPH scheme used to solve this numerical test is the inviscid scheme given by eqns (14), (17) and (20). A free-slip boundary condition has been enforced on the boundaries by using special ghost particles that are created via point symmetry about a layer of boundary particles as in Ferrari *et al.* [5].

Time integration has been performed by using a Runge-Kutta third order TVD scheme (see Gottlieb and Shu [10]).

At the initial time step, particles are placed on the left side of the tank with zero velocity and a hydrostatic pressure distribution, which has been calculated analytically according to the equation of state of eqn (14). The initial particle distribution can be seen in Figure 2, where the particles are color coded according to pressure values (measured in Pa) and distances are in meters.

The water column size is 14.6 cm x 28.9 cm, while the tank length is 58.4 cm. 800 particles have been used in the simulation with spacing of 7.3 mm in both vertical and horizontal directions.

In Figures 3–5, the results obtained with SPH are shown. Particles are color coded according to pressure values (measured in Pa).

Even though no comparison with experiments or other numerical solutions has been made at this stage, the results, in terms of pressure and particles displacement seem very realistic and reasonable.

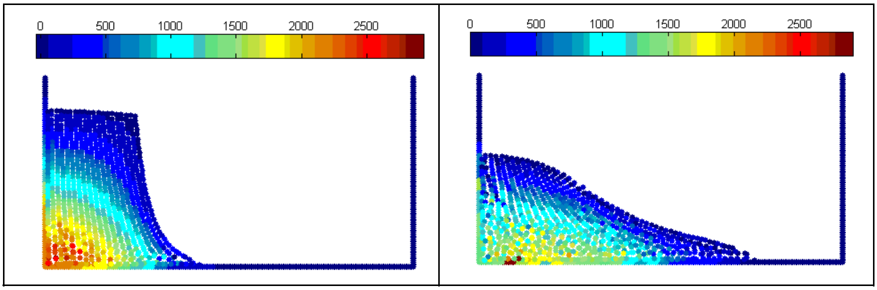

Figure 3: SPH solution at time 0.10 s (left) and 0.20 s (right) (color online only).

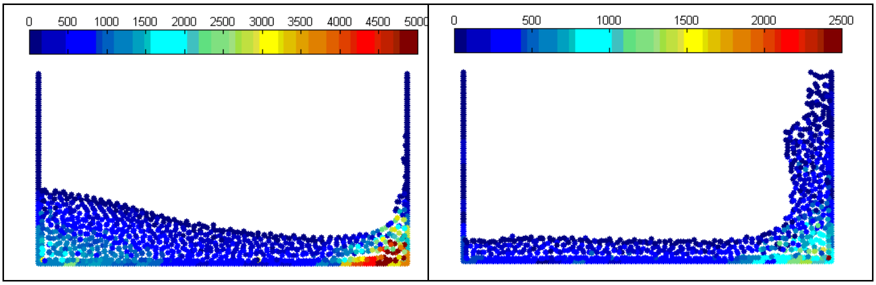

Figure 4: SPH solution at time 0.30 s (left) and 0.60 s (right) (color online only).

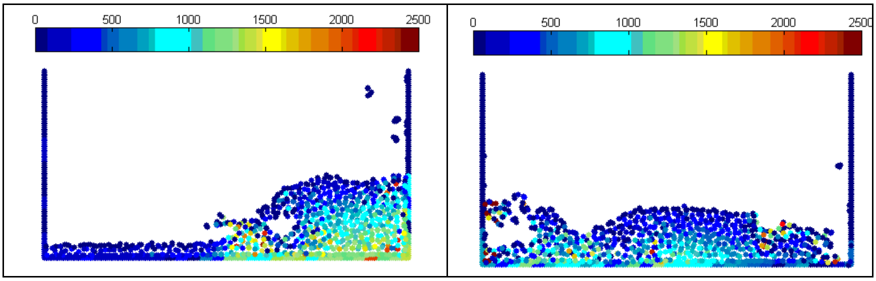

Figure 5: SPH solution at time 0.80 s (left) and 1.00 s (right).

4.2 Poiseuille viscous flow

The test case is the classical Poiseuille flow problem, solved with a layer of water flowing between two infinite parallel planes. Motion is created by a horizontal body force/mass $\underline{F}$, whose module has been set to $2\cdot10^{-4}$ m/s in the SPH simulation.

The analytical solution of the problem is known as a series solution. Its expression for horizontal velocity v_x as a function of time and space is as follows:

$$v_x(z,t)=\frac{F}{2\nu}z(z-L)+\sum_{n=0}^{+\infty}\frac{4FL^2}{\nu\pi^3(2n+1)^3}\cdot\exp\left(-\frac{(2n+1)^2\pi^2\nu}{L^2}t\right)\cdot\sin\left(\frac{\pi z}{L}(2n+1)\right) \qquad (24)$$

where:

- L is the distance between the two planes (set to 10^{-3} m in the SPH simulation);
- ν is the kinematic viscosity of water (10^{-6} m^2/s);
- z is the transversal coordinate between the planes (z=0 on the lower plane).

From the first spatial derivative of eqn (24), it is possible to obtain the analytical expression for the tangential stresses.

The SPH scheme used to solve this numerical test is the viscous scheme given by eqns (14), (16), (22) and (23). Boundary conditions have been enforced by using ghost particles that are being created by mirroring the liquid particles about the boundaries as in Morris *et al.* [4]. For viscous flows, a no-slip condition is required and so ghost particles are assigned an opposite velocity to the liquid particles they are created from. To simulate an infinite extension of the domain on the horizontal direction, periodic boundary conditions have been used, where particles exiting the domain on one side are re-entered in the system from the opposite side.

Time integration has been performed by using the simple Euler forward-in-time scheme.

At the initial time step, particles are placed between the two planes with zero velocity and a uniform zero pressure distribution. The motion is triggered by the horizontal body force/mass component $\underline{F}$, which is constant in every instant of the simulation. Four hundred particles have been used in the simulation with spacing of 0.025 mm in both vertical and horizontal directions.

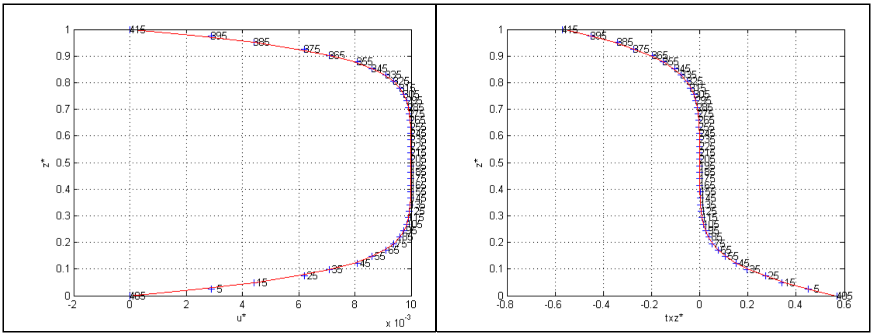

Figure 6: SPH solution of unsteady Poiseuille flow at an intermediate time step.

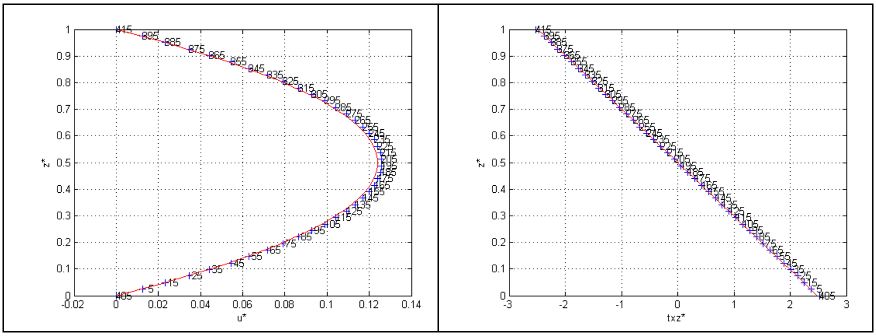

Figure 7: SPH solution of unsteady Poiseuille flow at the steady state.

In Figures 6 and 7, a comparison between the SPH solution of the problem and the analytical solution of the problem is shown, both for the horizontal velocity component (left hand side) and the tangential stresses (right hand side). Variables values are non-dimensional and scaled, with $l_0 = 1 \cdot 10^{-3}$ m being the scaling length, $u_0 = 2 \cdot 10^{-4}$ m/s being the scaling velocity and $\tau_0 = 4 \cdot 10^{-5}$ Pa being the scaling tension.

As can be seen from Figures 6 and 7, the SPH solution well agrees with the analytical solution. Steady state was considered reached when the average SPH solution error was lower than 2%. The error was also lower than this bound during all the intermediate instants of the simulation, except for the first ones, when the error was found to be higher.

5 Conclusions

SPH is a powerful and easy to code numerical tool that can be useful in order to solve many fluid flow problems. Its Lagrangian nature makes it particularly suitable for simulating free surface flows, but some drawbacks still wait to be solved. To this purpose, in order to enhance stability and smoothness of field variables, a Riemann based modification approach of continuity equation has been selected and implemented. The numerical elaborations show the capability of the method to capture the general features of the Dam break problem. Poiseuille test has shown very satisfactory behavior as well.

Next step will be the comparison of the results obtained with the selected method with experimental and other kind of numerical approaches.

References

[1] Monaghan J.J., Smoothed particle hydrodynamics. *Reports on Progress in Physics*, **68**, pp. 1703-1759, 2005.

[2] Randles P.W., Libersky L.D., Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, **138**, pp 375-408, 1996.

[3] Takeda H., Miyama S.M., Sekiya M., Numerical simulation of viscous flow by Smoothed particle hydrodynamics. *Progress of Theoretical Physics*, **92(5)**, pp. 939-960, 1994.

[4] Morris J.P., Fox P.J., Zhu Y., Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, **136**, pp. 214-226, 1997.

[5] Ferrari A., Dumbser M., Toro E.F., A new 3D parallel scheme for free surface flows. *Computers & Fluids*, **38**, pp. 1203-1217, 2009.

[6] Liu W.K., Jun S., Zhang Y.F., Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, **20**, pp. 1081-1106, 1995.

[7] Liu M.B., Liu G.R., Restoring particle consistency in smoothed particle hydrodynamics. *Applied Numerical Mathematics*, **56**, pp.19-36, 2006.

[8] Monaghan J.J., Simulating free surface flows with SPH. *Journal of Computational Physics*, **110**, pp. 399-406, 1994.

[9] Liu G.R., Liu M.B., *Smoothed Particle Hydrodynamics a meshfree particle method*, World Scientific Publishing, pp. 114-117, 2003.

[10] Gottlieb S., Shu C.-W., Total Variation Diminishing Runge-Kutta Schemes. *Mathematics of Computation*, **67(221)**, pp. 73-85, 1998