Particle refinement for fluid flow simulations with SPH

Yaidel Reyes López[1] **and Dirk Roose**[2*]

[1]*Department of Computer Science, Katholieke Universiteit Leuven*
Celestijnenlaan 200A - bus 2402. 3001 Heverlee, Leuven, Belgium.

Centro de Investigaciones de Métodos Computacionales y Numéricos en la Ingeniería (CIMCNI),
Universidad Central "Marta Abreu" de Las Villas (UCLV)
Road to Camajuaní Km. $5\frac{1}{2}$, 54830 Santa Clara, Villa Clara, Cuba.
e-mail: yaidel.reyeslopez@cs.kuleuven.be

[2]*Department of Computer Science, Katholieke Universiteit Leuven*
Celestijnenlaan 200A - bus 2402. 3001 Heverlee, Leuven, Belgium.
e-mail: dirk.roose@cs.kuleuven.be

Abstract

In this paper, we present a refinement algorithm for the SPH method. A particle is refined by replacing it with smaller daughter particles. The position of the new particles is calculated by using a square pattern centered at the position of the refined particle. We study the possibility of scaling and rotating this pattern according to the local distribution of the particles to reduce the overlap with the newly created daughter particles. The results of the simulations using the fully refined domain and the simulations using the dynamic refinement starting from the unrefined domain are compared and are in a good agreement. Kinetic energy as well as linear and angular momentum are conserved by the refinement procedure. The algorithm is presented in 2D, but its extension to 3D is straightforward.

Keywords: meshless methods, adaptivity, Smoothed Particle Hydrodynamics, particle refinement

1. Introduction

The Smoothed Particle Hydrodynamics (SPH) method has been successfully applied in several types of problems including fluid dynamics and deformation of solids. Because SPH is a fully mesh-free Lagrangian method it is an attractive, suitable and robust choice to simulate problems involving large deformations and moving boundaries. Comprehensive reviews of the method can be found in [7, 8, 11].

SPH is a kernel based method where the domain is discretized by a set of particles (or interpolation points) that independently carry the material properties and the local resolution or smoothing distance (h) of the particle. Varying the smoothing length allows to vary the spatial resolution. Nevertheless, it is not common in SPH simulations to have different levels of refinement, as in e.g. the Finite Element Method (FEM), to improve the accuracy and/or to reduce computational cost.

Static refinement (refinement during the initialization of the particles) has been performed already in SPH simulations, e.g. Oger et al. [14]. However, during the simulation the particle distribution is dictated by the flow and control over the resolution, set during the initialization, can be lost. For this reason static refinement is not sufficient for some problems.

The first steps in dynamic adaptivity in SPH were performed for astrophysical simulations, where the density was used as the criterion to change the resolution [1, 4, 5, 10, 13]. In other fields Lastiwka and co-workers [6] proposed a method for adaptively inserting and removing particles and they tested the method in a shock tube problem.

More recently, Feldman and Bonet [3] proposed a particle refinement procedure, where daughter particles are located using axis aligned hexagonal and triangular patterns that are scaled according to a spread parameter defined before the simulation starts. In this paper we present a refinement procedure using a square pattern for the daughter particles and for which the spread parameter and rotation are calculated according to the local distribution of the particles.

2. SPH for general fluid dynamics

The SPH method provides a numerical solution for integral equations and partial differential equations (PDEs). The state of the system is represented by a set of particles used to discretize the governing equations. These particles move with the flow and interact with all the particles within a range controlled by a smoothing function or kernel (W).

2.1. Integral and particle approximation of a function

The SPH approach starts from the following identity:

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

where f is a continuous function of the position vector x in Ω, and δ is the Dirac delta function. Replacing $\delta(x - x')$ by a smoothing function $W(x - x', h)$ with a finite spatial size h, a kernel (integral) approximation of $f(x)$ is obtained as

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

where h is the smoothing length defining the support domain of the kernel. Major properties and requirements of W are summarized and described in [7, 8]. For the simulations presented in this work, the piecewise cubic spline is used as the kernel function.

Eq. 2 is approximated by discretizing the domain into a finite set of particles where particle i has volume V_i, and by replacing

[*] This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its author(s).

the volume to introduce the mass (m_i) and the density (ρ_i), the integral can be approximated by a summation, becoming

$$\langle f(x)\rangle = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(x_j) W(x - x_j, h), \tag{3}$$

where N is the number of particles. This form is known as the particle approximation. A similar expression can be derived for approximating the derivative of the function

$$\langle \nabla f(x)\rangle = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(x_j) \nabla W(x - x_j, h). \tag{4}$$

2.2. *Approximation of the governing equations*

The SPH approximation, to be applied in general fluid dynamics, is derived by discretizing the Navier-Stokes equations, which state the conservation of mass, momentum and energy. Various formulations have been obtained as result of different transformations in the derivation of the particle approximation. In this work, we use the formulation described by Monaghan [12] to simulate free surfaces flows.

The particle approximation of the density (ρ) is obtained by using the continuity approach leading to the equation

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^{N} m_j (v_i^\alpha - v_j^\alpha) \frac{\partial W_{ij}}{\partial x_i^\alpha}, \tag{5}$$

where v and x are the velocity and position respectively (the Einstein notation is applied herein).

For the momentum equation, the following expression is used

$$\frac{Dv_i^\alpha}{Dt} = -\sum_{j=1}^{N} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} - \Pi_{ij}\right) \frac{\partial W_{ij}}{\partial x^\alpha} + F_i^\alpha, \tag{6}$$

where p is the pressure, m is the mass, Π_{ij} is the artificial viscosity and F_i^α represents the external force, which is the gravity for the problems considered in this paper.

Following Monaghan [12], an incompressible flow is considered to be slightly compressible. The relative fluctuation in density is proportional to $(\frac{v_{typ}}{c})^2 = M^2$, where v_{typ} is the typical bulk velocity, c is the speed of sound, and M is the Mach number. The value for c is chosen to give a density fluctuation of $\sim 1\%$ resulting in $M = 0.1$. Here the pressure is calculated by applying the following equation of state

$$p_i = B\left(\left(\frac{\rho_i}{\rho_0}\right)^\gamma - 1\right), \tag{7}$$

where γ is a constant commonly taken as 7, ρ_0 is the reference density, and B is a problem dependent constant related with the speed of sound by

$$c = \sqrt{\frac{\partial p}{\partial \rho}} \approx \sqrt{\frac{7B}{\rho_0}}. \tag{8}$$

The artificial viscosity is calculated using

$$\Pi_{ij} = \begin{cases} \frac{-\alpha c \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}} & \text{if } \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\mu_{ij} = \frac{h \mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{\mathbf{x}_{ij}^2 + 0.01h^2}$, $\bar{\rho}_{ij} = \frac{\rho_i + \rho_j}{2}$, the notation $A_{ij} = A_i - A_j$ is used, and α and β are constants that change with the problem.

Different expressions for the artificial viscosity have been described in the literature [9]. However, since in the problems treated in this work the viscosity of the fluid is low, *a priori* we consider that using a different artificial viscosity approach shouldn't affect the results.

2.3. *Symmetrization of particle interaction*

The formulation used herein keeps the value of h fixed for each particle, but due to the dynamic refinement procedure, while increasing the local resolution, "smaller" particles with smaller smoothing distance are introduced. This leads to situations where two particles, e.g. i and j, with unequal h_i and h_j interact. In such cases, it may happen that the influence domain of particle i contains particle j but not vice versa. Hence it is possible to have particle i exerting forces on particle j without the corresponding reaction on i, which is a violation of Newton's third law.

To overcome this, some approaches for preserving the symmetry have been used (see [7]). In this work the smoothing distance used for computing the interaction between particles is modified to produce a symmetric smoothing length using the arithmetic mean $h_{ij} = (h_i + h_j)/2$.

2.4. *Boundary conditions*

For implementing the boundary conditions, two types of virtual particles are used. Virtual particles of the first type are located at fixed positions on the solid boundaries. These particles only exert a repulsive Lennard-Jones force, avoiding penetration through the boundaries. The applied force is calculated by the expression

$$f(r) = \begin{cases} D\left(\left(\frac{r_0}{r}\right)^{12} - \left(\frac{r_0}{r}\right)^{6}\right) \frac{x_{ij}^\alpha}{r^2} & \text{if } r < r_0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where r_0 is the reference distance and r is the distance between the real and the fixed particle.

Virtual particles of the second type are ghost particles mirrored to the outside from the real particles. Their physical properties are copied from real particles, except for the velocity which is calculated depending on the boundary condition (e.g. slip or no slip).

3. Refinement procedure

In SPH simulations, the finest resolution needed to obtain the required accuracy is commonly used in the whole domain. However, often zones are observed where the flow of the particles is slower and the properties behave smoother. In these zones, fewer particles can be used while a similar accuracy can be achieved.

For using different resolutions, the regions where more particles are needed have to be identified, either before starting the simulation (statically) or during the simulation (dynamically), depending on the dynamics of the problem. Then, a method to change the resolution must be applied.

Notice that for dynamic refinement the additional computational cost introduced by the refinement procedure have to be carefully considered, since both the identification of the particles to be refined and the procedure to increase the resolution at these particles are performed during the simulation loop. Dynamic refinement only makes sense if there is a noticeable gain in execution time compared with the fully refined domain, and there is a little loss in accuracy.

3.1. *Refinement criterion*

For static refinement, more particles should be placed in those regions where more precision is required and where the particle disorder is larger due to the flow. These regions should be selected carefully, taking into account the problem dynamics. However, this is not enough for some problems where the resolution initially set is lost due to the movement of the particles. For that reason, in this paper we focus on dynamic refinement.

Several criteria could be used for determining which particles should be refined during the simulation, depending on the type of the problem. Feldman and Bonet [3] used refinement zones, splitting all the particles that go inside those zones. The number of neighbors can also be used and particles with few neighbors can be split to maintain the local accuracy. Physical properties have been successfully used too as criteria for the refinement. Kitsionas and Whitworth [4, 5] applied a refinement criterion based on a physical requirement known as "Jeans Condition" and Lastiwka et al. [6] used a criterion based on the velocity gradient.

For the simulations in this work, a criterion based on the velocity of the particles is used for dynamic refinement. Particles with a velocity greater than a pre-established threshold are refined. This refinement criterion was found to be adequate in the problems that we considered. Note that the refinement algorithm described below is totally independent of the refinement criterion.

3.2. *Refinement procedure*

The refinement algorithm will be described for a 2D domain, but its extension to 3D is straightforward. We start with the simple case of particles located in a lattice during static refinement, if the particle i, with level of refinement l_i, is to be refined, it will be replaced by four new daughter particles $a = 0, ..., 3$, placed in a square pattern as shown in Fig. 1a, with $h_a = h_i/2$, $m_a = m_i/4$, $d^{l_a} = d^{l_i}/2$ and $l_a = l_i + 1$, where d^l is the inter-particle space in the lattice at the level l (we will call it reference separation).

All particles at the same refinement level have the same smoothing distance, then h is related to d according to $h^l = rd^l$. With the current kernel function, for having a sufficient and necessary number of particles within the support domain, r should be about 1.2 [7]. This procedure produces a sufficiently quiet initial state, i.e. if the particles placed in a lattice with level of refinement l are refined, a new lattice of level of refinement $l + 1$ is obtained (Fig. 1b).

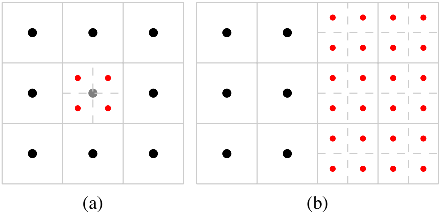


Figure 1: Refinement of particles in a lattice. Red points represent the new daughter particles

For dynamic refinement we use the same square pattern with some additional considerations. The properties of the daughter particles should be related to the properties of the replaced particle in a way that the change of the local properties after their insertion is small and that global properties like mass, kinetic energy and linear and angular momentum are preferably conserved.

The properties of the daughter particles are assigned as follows: $m_a = \frac{m_i}{4}$, $h_a = h_i/2$, $v_a = v_i$ and $\rho_a = \rho_i$. In [2, 3] it is shown that the only possible velocity refinement strategy that simultaneously conserves kinetic energy and linear momentum is to give to all daughter particles the original unrefined velocity value which, combined with preservation of mass and a symmetric pattern to place the daughter particles, also preserves angular momentum. Since density does not affect the conservation of global properties it could be interpolated at the daughter particles, but no attempt has been done in this work.

We studied two approaches for locating the daughter particles using the square pattern. In the first approach, if a particle i is to be refined, then a cell with side d^{l_i} is centered at the position of the particle i ($\mathbf{x}_i$). This cell is divided in four equal cells and the daughter particles are placed at the center of the new cells forming the square pattern as shown in Fig. 2a. In this work the cell is axis aligned, but it could be randomly rotated. This procedure ignores the distribution of the neighbor particles. However, during the evolution of the simulation, the distance between a particle and its neighbors can become smaller than d^{l_i}. We observed in some simulations that using an axis aligned cell with sides d^{l_i} can lead to newly created daughter particles placed very close to neighbor particles, introducing large overlaps.

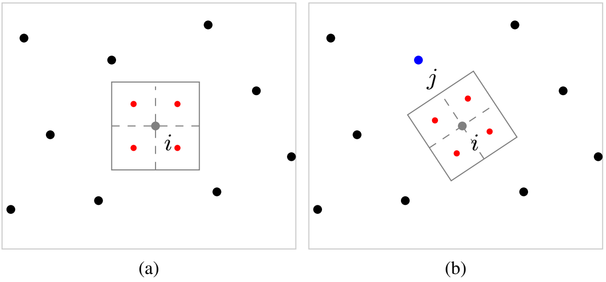


Figure 2: Square pattern for dynamic refinement. Red points represent the new daughter particles

In the second approach the cell centered at the position of the particle i, and therefore the square pattern, is scaled and rotated to reduce the overlap between the neighbors and the newly created daughter particles (Fig. 2b). For this, a reference particle j is selected. This is the particle that gives the highest value of F_{ij}, which is a scalar factor calculated in terms of the distance as $F_{ij} = \frac{d_{ij}}{r_{ij}}$, where $d_{ij} = \frac{d^{l_i}+d^{l_j}}{2}$ and r_{ij} is the distance between the particles. Once the reference particle is selected, the cell centered at $\mathbf{x}_i$ is aligned to the line segment $\overline{\mathbf{x}_i\mathbf{x}_j}$ (see Fig. 2b) and it is scaled with $s_i = \frac{d^{l_i}}{2F_{ij}}$, where s_i is the length of the side of the cell.

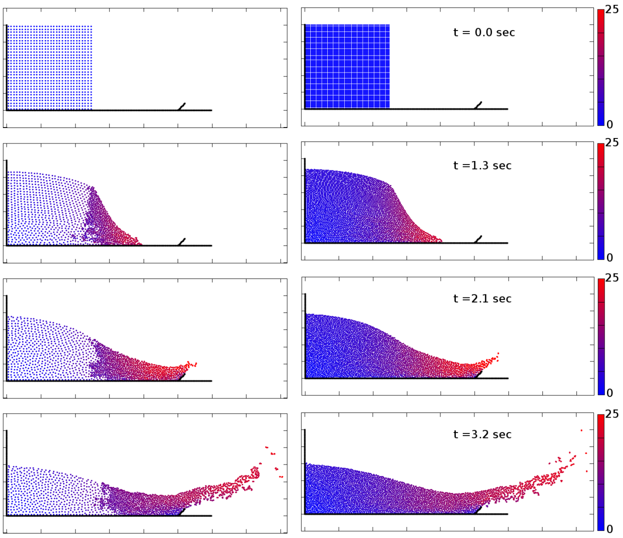
4. Results

The dynamic refinement procedure is applied to simulate the well-known breaking dam and the splash of a drop of water test cases. The results obtained while using the dynamic refinement with the axis aligned pattern (AAP) and with the rotated and scaled pattern (RSP) are compared with the results obtained with the simulations using the fully refined domain. While discussing the results, no figures showing the evolution of the simulations using dynamic refinement with AAP are included because they are very similar to the corresponding figures obtained from the results of the simulation using RSP and no significant difference can be noticed from them. Differences between the simulations that use the AAP and the ones that use the RSP can be observed better after post-processing the results and compare, for the two problems studied, the evolution of the height of the water and also the evolution of the surge front in the case of the breaking dam problem.

A challenging aspect in the considered problems is the inclusion of free boundaries because the perturbations arising from the refinement of particles are more slowly damped there, since particles at the surface are not completely surrounded by other particles. In the simulations run in this work daughter particles are not allowed to be refined, leading to only two different levels of refinement, and the refinement criterion is checked after a prescribed number of iterations to reduce the computational cost of this task. Here, the criterion is checked after every 100 time steps. Free-slip boundary conditions are applied in all the simulations.

4.1. *Breaking dam*

In the breaking dam problem a 25m side square block of water is considered. Initially the water is at rest in a container until the right wall is removed suddenly. An obstacle is placed at 50m from the origin. Four different simulations are performed for this problem. In the first simulation the block of water is discretized by 900 particles located in a lattice (unrefined domain). For the second simulation, the 900 particles are statically refined leading to a finer lattice of 3600 particles (fully refined domain). Finally, two simulations using dynamic refinement are performed, one with AAP and the other with RSP. The two simulations using dynamic refinement start from the unrefined domain and the threshold used for the refinement criterion was set to $10m/s$.



(a) (b)

Figure 3: Breaking dam simulations: a) with dynamic refinement (RSP), b) with the fully refined domain. Color represents velocity.

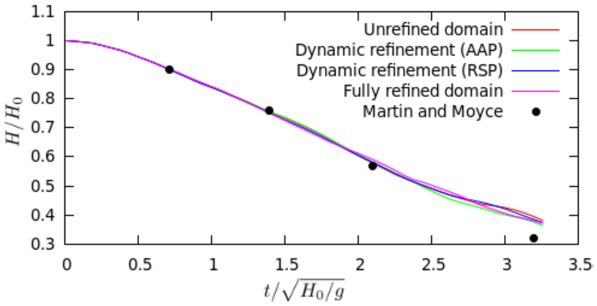


Figure 4: Evolution of the height (H) of the water in the breaking dam simulations.

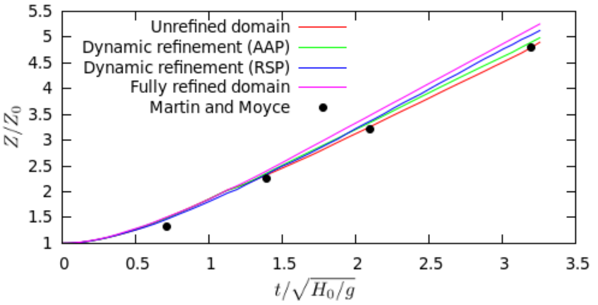


Figure 5: Evolution of the front (Z) of the water in the breaking dam simulations.

Fig. 3 shows the evolution of the particles during the simulation using dynamic refinement with RSP and using the fully refined domain. It can be seen that there is a good agreement in the distribution and the velocity of the particles. By visualizing the evolution of the density we found that there are no large density fluctuations in the interface between the particles of different sizes, this is using either AAP or RSP.

The evolution of the water height (H) and the surge front (Z) of the water particles in the four simulations are shown in Fig. 4 and Fig. 5 respectively, where also the experimental result of Martin and Moyce [12] are presented. Because this experimental result has a timing error of ~ 0.1, the comparison with it is only approximate. The values for H and Z are normalized by the initial values $H_0 = 25m$ and $Z_0 = 25m$ and the time is normalized by $\sqrt{H_0/g}$ with g the gravity force.

The main differences are observed in the evolution of the surge front where, contrary to what we expected, the simulation with the lowest resolution gives the results that are closest to the experiment. However, the slower behavior of the surge front in the experimental result could be due to a drag force between the fluid and the bottom which is neglected in the simulations when the free-sleep boundary condition is applied. As was expected, the results of the simulations using dynamic refinement are between the unrefined and the fully refined simulations. Fig. 5 shows that when using RSP the results are slightly closer to the results of the simulation with the fully refined domain than when using AAP.

4.2. *Splash of a drop of water*

For the simulations of the splash of a drop into rest water, the following situation is considered. A $10m$ high block of water is at rest in a $25m$ wide rectangular container, when a water drop with radius $2.0m$ with center at $(0m, 13.3m)$ and downward velocity of $2m/s$ starts moving until it splashes into the rest water. Four different simulations are performed using the same strategy employed for the breaking dam problem to obtain different resolutions. For the simulation with the unrefined domain 360 particles discretize the rest water and 21 particles the water drop. In the simulations using the dynamic refinement the threshold of the refinement criterion was set to $1.5m/s$.

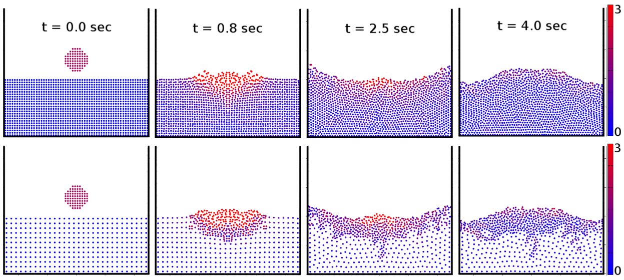


Figure 6: Simulations of the water drop splashing into rest water using the fully refined domain (top row) and the dynamic refinement with RSP (bottom row). Color represents velocity.

Fig. 6 shows the evolution of the simulations with dynamic refinement using RSP and with the fully refined domain. It can be seen that the movement of the surface as well as the distribution of the velocity are similar. In Fig. 7 we show the evolution of the height of the water in each simulation. The simulation with dynamic refinement using the AAP better reproduces the wavy behavior of the surface than the one using the RSP and gives a result that is closer to the one obtained with the simulation using the fully refined domain. The line corresponding to the simulation using RSP shows a damping introduced by the refinement in the velocity of the particles that leads to a faster decrease of the waves produced due to the splashing.

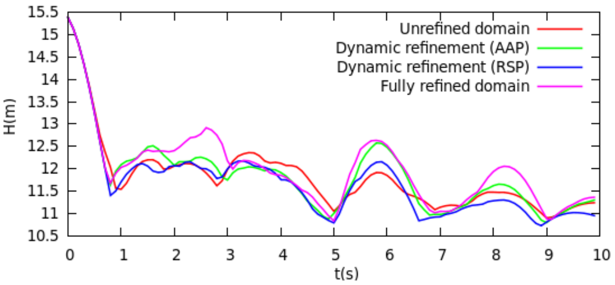


Figure 7: Evolution of the height (H) of the water in the simulation of the splash of the drop.

5. Conclusions

The results obtained with the simulations using dynamic refinement with AAP are in good agreement with those obtained when using the fully refined domain. In the case of the simulations using the RSP, the results obtained for the breaking dam problem are very similar to the results obtained with the AAP, but in the simulation of the splash of the water drop the use of the RSP introduces to much damping on the velocity of the particles. The damping may be produced by the scaling of the pattern. Considering also that the use of the RSP is more computationally expensive than AAP, because the reference particle have to be found among the neighbors of the particle to be refined, we conclude that rotating and scaling the pattern is not worthwhile for the implemented model.

The reduction in the number of particles decreases the calculations required resulting in a shorter total time of the simulation. The total time needed when using dynamic refinement was $\sim 35\%$ and $\sim 56\%$ of the time required when using the fully refined domain for the breaking dam and the splash of the drop of water respectively.

Dynamic refinement proved to be a valid strategy to reduce the computational time while keeping a good accuracy. However, the error introduced due to the refinement should be analyzed in detail. Further issues to be studied are: a) dynamic coarsening, as the opposite of refinement; b) the implementation of the method in 3D; c) the application to other test cases.

References

[1] Bate, M.R., Bonnell, I.A. and Price, N.M., Modelling accretion in protobinary systems, *Monthly Notices of the Royal Astronomical Society*, 277, pp. 362-376, 1995.

[2] Feldman, J., Dynamic refinement and boundary contact forces in Smoothed Particle Hydrodynamics with applications in fluid flow problems, *University of Wales Swansea, School of Engineering*, 2006.

[3] Feldman, J. and Bonet, J., Dynamic refinement and boundary contact forces in SPH with applications in fluid flow problems, *International Journal for Numerical Methods in Engineering*, 72, pp. 295-324, 2007.

[4] Kitsionas, S. and Whitworth, A.P., Smoothed Particle Hydrodynamics with particle splitting, applied to self-gravitating collapse, *Monthly notices of the Royal Astronomical Society* 330, pp. 129-136, 2002.

[5] Kitsionas, S. and Whitworth, A.P., High-resolution simulations of clump-clump collisions using SPH with particle splitting, *Monthly notices of the Royal Astronomical Society* 378, pp. 507-524, 2007.

[6] Lastiwka, M., Quinlan, N. and Basa, M., Adaptive particle distribution for smoothed particle hydrodynamics, *International Journal for Numerical Methods in Fluids*, 47, pp. 1403-1409, 2005.

[7] Liu, G. and Liu, M. *Smoothed particle hydrodynamics: a meshfree particle method*, World Scientific Publishing, 2003.

[8] Liu, M. and Liu, G., Smoothed Particle Hydrodynamics (SPH): an Overview and Recent Developments, *Archives of Computational Methods in Engineering*, 17, pp. 25-76, 2010.

[9] Lombardi, J.C., Sills, A., Rasio, F.A. and Shapiro, S.L., Tests of spurious transport in smoothed particle hydrodynamics, *Journal of Computational Physics*, 152, pp. 687-735, 1999.

[10] Meglicki,Z., Wickramasinghe, D. and Bicknell, G.V., 3D structure of Truncated accretion disks in close binaries, *Monthly Notices of the Royal Astronomical Society*, 264, pp. 691-704, 1993.

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

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

[13] Monaghan, J. and Varnas, S., The dynamics of interstellar cloud complexes, *Monthly Notices of the Royal Astronomical Society*, 231, pp. 515-534, 1988.

[14] Oger, G., Doring, M., Alessandrini, B. and Ferrant, P., Two-dimensional SPH simulations of wedge water entries, *Journal of Computational Physics* 213, pp. 803-822, 2006.