ERF2013-044

Simulation of Helicopter Ditching using Smoothed Particle Hydrodynamics

Mark A. Woodgate¹, George N. Barakos¹, Nigel Scrase² and Tim Neville²

¹ CFD Laboratory, Department of Engineering University of Liverpool, L69 3GH, U.K. http://www.liv.ac.uk/flightscience/PROJECTS/CFD/ROTORCRAFT/index.htm

> ² Fuselage Aerodynamics, Helicopter Systems Design AgustaWestland Ltd, Yeovil, BA20 2YB, UK University of Liverpool, L69 3GH, U.K.

Abstract

This paper explores the potential use of smoothed particle hydrodynamics methods for helicopter ditching. The method appears suitable for the task since it is mesh-free and can accommodate the interaction between a floating object and the free-surface of water. Simple cases of objects dropped on water were first studied to establish confidence on the method and quantify the effect of the numerical parameters of SPH including the boundary condition between the water and solid, the effect of the number and type of smoothed particles as well as the generation of different sea-states for the ditching. Once confidence on the method was established, experiments for the ditching of a model-scale helicopter were used for further validation. It appears that smoothed particle hydrodynamics has good potential for use in ditching simulation, provided the parameters of the method can be carefully selected.

1 INTRODUCTION

Ditching is defined to be an emergency surfacing on water, deliberately executed, with the intent of abandoning the helicopter as soon as practical. After ditching the helicopter either floats upright, floats inverted or sinks inverted. Between 2000 and 2003 the CAST project, Crashworthiness of Helicopter on Water: Design of Structures using Advanced Simulation Tools, assessed methods that could simulate ditching, with Smoothed Particle Hydrodynamics (SPH) being one of these. This work was continued in the GARTEUR HC/AG-15, Improvement of SPH methods for application to helicopter ditching, and the follow on program Smart Aircraft in Emergency Situations (SMAES). These included looking into adding air entrapment, cavitation and suction force effects to improve both analytical and numerical fluid dynamics models.

Water impact was first studied by von Karman [1] in the late twenties where he developed a theoretical formula for water impact and compared it to experimental data from sea plane floats. The problem was idealised to the calculation of forces generated during a vertical impact of a wedge shape onto water in two dimensions. Several years later Wagner [2, 3] increased the fidelity of the model by taking into account the free surface in the form of the local uprise. The Wagner model was then extend into axisymmetric cases by Chuang [4]. More recently Scolan and Korobkin [5,6] looked at the energy distribution from the vertical impact of 3D objects on calm water.

An assessment of the models of Zhoa and Faltinsen [7], the simplified generalised Wagner model and the modified Logvinovich model proposed by Korobkin [8] has been carried out by Tassin *et. al.* [9]. In all these analytical models the body is assumed to be rigid and the fluids inertia dominates the forces acting on it during the impact. The effects of viscosity, surface tension, compressibility, gravity are neglected. The flow is also assumed to be irrotational.

Regardless of the promising results obtained with these models, the need to study the impact of complex shapes on water requires a different approach that can accommodate changes of the geometry as well as multiple surfaces impacting the water at the same time. For this reason most of the modern efforts are directed towards Computational Fluid Dynamics methods that offer a general framework for ditching studies even if their computational cost is considerably higher.

2 NUMERICAL METHOD

2.1 Smoothed Particle Hydrodynamics Overview

SPH is a mesh free method originally formulated by Lucy [10], and Gingold and Monaghan [11] that solves a set of partial differential equations both accurately and stably without using any mesh connecting the particles. SPH is an interpolation method which approximates values and derivatives of continuous variables using a set of discrete sample points. These points are smoothed particles which have a position, velocity and mass, and these are calculated as some weighted average from all adjacent particles. This work builds on the SPHyscis/DualSPHyscis solver [12, 13] that has been extended to include a rotor model since when ditching the helicopter can still have substantial lift from the main rotor.

SPH is a computational fluid dynamics method but takes

a different approach to mainstream mesh based methods like the Helicopter Multi-Block solver of Liverpool. In mesh based methods the continuum domain is divided into discrete small sub-domains called cells. The edges if these cells them form a lattice which connects the mesh points together. The governing equations are then discretised over these cells. Although mesh based methods have been very successful they are not well suited for all types of problems. The difficulties occur when trying to keep the mesh compatible with the physical continuum and hence problems with free surfaces, deformable boundaries or moving interfaces all present complications for mesh based schemes.

The outline of the basic SPH method is shown in figure 1. The fluid is treated as a set of particles each of which has physical properties associated to them like mass, density, position and velocity. Next a neighbour list is constructed to find the adjacent particles. This is done by cutting the computational domain into boxes of size 2h. A list is then built of all the particles which are in that box. For any given particle only the interaction between itself and adjacent particles closer than 2h are to be considered so a particle can only interact with particles in the same or adjacent boxes. All particles in these 9 boxes are checked to find the ones within 2h. The particles interaction can now be calculated and these forces can be used to update the physical properties of each particle.

As stated above the SPH is an interpolation method. The interpolation is based on the theory of integral interpolants using kernels that approximate a delta function. The integral interpolant reads:

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}'$$
(1)

where f is a function of the three dimensional position vector \mathbf{x} , $\delta(\mathbf{x} - \mathbf{x}')$ is the Dirac delta function and Ω is the volume of the integral containing the point \mathbf{x} . If the Dirac delta function is replaced by a smoothing function $W(\mathbf{x} - \mathbf{x}', h)$ with width h then equation 1 becomes

$$\langle f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}') d\mathbf{x}'$$
 (2)

The width h is a scaling factor that controls the smoothness/roughness of the kernel whist using $\langle \rangle$, that is the standard SPH convention. The smoothing function W is normally an even function which satisfies the following conditions. Firstly, the integration of the smoothing function Wmust be normalised to unity

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}') d\mathbf{x}' = 1.$$
(3)

Secondly, in the limit as $h \rightarrow 0$ it must equal the Dirac delta

$$\lim_{h \to 0} W(\mathbf{x} - \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$$
(4)

and lastly W should be compact

$$W(\mathbf{x} - \mathbf{x}') = 0 \quad |\mathbf{x} - \mathbf{x}'| > \kappa h \tag{5}$$

for some constant κ . This implies that only particles close to the point x are used in the average.

The approximation of the gradient of f is obtained by replacing $f(\mathbf{x})$ with $\nabla \cdot f(\mathbf{x})$ in equation 2

$$\langle \nabla \cdot f(\mathbf{x}) \rangle = \int_{\Omega} [\nabla \cdot f(\mathbf{x}')] W(\mathbf{x} - \mathbf{x}') d\mathbf{x}'$$

$$= \int_{\Omega} \nabla \cdot [f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}')] d\mathbf{x}'$$

$$- \int_{\Omega} f(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}')] d\mathbf{x}'$$

$$= \int_{S} f(\mathbf{x}')W(\mathbf{x} - \mathbf{x}') \cdot \mathbf{n} dS$$

$$- \int_{\Omega} f(\mathbf{x}') \cdot \nabla W(\mathbf{x} - \mathbf{x}')] d\mathbf{x}'$$

$$(6)$$

using the divergence theorem where S is the surface of the domain of integration Ω . If Ω lies within the problem domain and since the function W has compact support the surface integral is zero. However, if Ω overlaps the problem domain for example close to the fluid body boundary the function W is truncated and so non zero.

If the infinitesimal volume $d\mathbf{x}'$ is replaced with the volume of the particle ΔV_i that has corresponding mass m_i then

$$m_j = \Delta V_j \rho_j \tag{7}$$

for each of the N particles in the support domain Ω then the numerical approximation to equation 2 is:

$$\langle f(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}') W(\mathbf{x} - \mathbf{x}') d\mathbf{x}'$$

$$\approx \sum_{j}^{N} f(\mathbf{x}_{j}) W(\mathbf{x} - \mathbf{x}') \Delta V_{j}$$

$$= \sum_{j}^{N} \frac{m_{j}}{\rho_{j}} f(\mathbf{x}_{j}) W(\mathbf{x} - \mathbf{x}').$$
(8)

The effectiveness of the SPH method depends on the choice of the weighting function. Kernels are expressed as a function of a non dimensional distance between particles given by q = r/h where r is the distance between particles, and h controls the number of particles that the interactions are calculated over. There are a huge number of possible functions and some of the more common are outlined below and are shown in figure 2.

The Gaussian

$$W(r,h) = \alpha_d \exp(-q^2) \tag{9}$$

The Quadratic

$$W(r,h) = \alpha_d \left[\frac{3}{16} q^2 - \frac{3}{4} q + \frac{3}{4} \right] \quad 0 \le q \le 2$$
 (10)

The Cubic spline

$$W(r,h) = \alpha_d \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & 0 \le q \le 1\\ \frac{1}{4}(2-q)^3 & 1 \le q \le 2\\ 0 & q \ge 2 \end{cases}$$
(11)

The Quintic

$$W(r,h) = \alpha_d \left[1 - \frac{q}{2}\right]^4 (2q+1) \quad 0 \le q \le 2$$
 (12)

2.2 SPH for the Navier-Stokes Equations

The continuity equation in Lagrangian form is written as

$$\frac{D\rho}{Dt} = -\rho\nabla.u.$$
(13)

There are two commonly used SPH continuity formulations used in computations which are derived by applying different approximation rules.

Considering equation 13, one can write:

$$\left\langle \frac{D\rho}{Dt} \right\rangle = - \left\langle \rho \nabla \cdot v \right\rangle$$

$$\approx - \left\langle \rho \right\rangle \left\langle \nabla \cdot v \right\rangle$$

$$\approx - \left\langle \rho \right\rangle \nabla \cdot \left\langle v \right\rangle$$
(14)

with

$$(\nabla, \langle v \rangle)_i = \sum_j \Delta V_j v_j \cdot \nabla_i W_{ij} = \sum_j \frac{m_j}{\rho_j} v_j \cdot \nabla_i W_{ij}$$
(15)

and

$$\langle \nabla W \rangle_i = \sum_j \frac{m_j}{\rho_j} \nabla_i W_{ij}.$$
 (16)

Substituting equations (15) and (16) into equation (14) gives

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j \frac{m_j}{\rho_j} (v_i - v_j) \cdot \nabla_i W_{ij} = \rho_i \sum_j \frac{m_j}{\rho_j} v_{ij} \cdot \nabla_i W_{ij}.$$
(17)

The second continuity equation can be derived by applying the approximation rule for the dot product as follows

$$\langle \rho \nabla \cdot v \rangle_i \approx \sum_j (v_j - v_i) \cdot \nabla_i W_{ij} m_j$$
 (18)

and hence

$$\frac{D\rho_i}{Dt} = \rho_i \sum_j \frac{m_j}{\rho_i} (v_i - v_j) \cdot \nabla_i W_{ij} = \rho_i \sum_j \frac{m_j}{\rho_i} v_{ij} \cdot \nabla_i W_{ij}.$$
(19)

It is easy to spot the difference between equation (17), that represents the summation density approximation, and (19) that represents the continuity density approximation.

The momentum equation in a continuum field, with no body force, is

$$\frac{Dv^{\alpha}}{Dt} = \frac{1}{\rho} \frac{\partial \sigma^{\alpha\beta}}{\partial x^{\beta}}$$
(20)

where σ is the total stress tensor made up of two parts, the isotropic pressure p and the viscous stress τ

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \tau^{\alpha\beta}.$$
 (21)

For Newtonian fluids the viscous shear stress should be proportional to the shear strain rate via the dynamic viscosity μ , and consequently

$$\tau^{\alpha\beta} = \mu \left(\frac{\partial v^{\beta}}{\partial x^{\alpha}} + \frac{\partial v^{\alpha}}{\partial x^{\beta}} - \frac{2}{3} (\nabla \cdot v) \delta^{\alpha\beta} \right).$$
(22)

Below are the two most common ways in the literature to approximate the moment equation. First consider the equation

$$\left\langle \rho \frac{Dv}{Dt} \right\rangle_i = \left\langle \nabla \cdot \sigma \right\rangle_i. \tag{23}$$

As

$$\rho_i \frac{Dv_i}{Dt} \approx \sum_j \left(\sigma_i + \sigma_j\right) \cdot \nabla_i W_{ij} \Delta V_j \tag{24}$$

then we have:

$$m_{i} \frac{Dv_{i}}{Dt} = \sum_{j} \Delta V_{i} \Delta V_{j} \left(\sigma_{i} + \sigma_{j}\right) \cdot \nabla_{i} W_{ij} = \sum_{j} \frac{m_{i} m_{j}}{\rho_{i} \rho_{j}} \left(\sigma_{i} + \sigma_{j}\right) \cdot \nabla_{i} W_{ij}.$$
 (25)

Secondly, using a different SPH gradient approximation

$$\left\langle \frac{Dv}{Dt} \right\rangle_i = \left\langle \frac{1}{\rho} \nabla \cdot \sigma \right\rangle_i \tag{26}$$

and as

$$\frac{Dv_i}{Dt} \approx \sum_j \left(\frac{\sigma_i}{\rho_i^2} + \frac{\sigma_j}{\rho_j^2}\right) \cdot \nabla_i W_{ij} m_j \tag{27}$$

then

$$m_i \frac{Dv_i}{Dt} = \sum_j m_i m_j \left(\frac{\sigma_i}{\rho_i^2} + \frac{\sigma_j}{\rho_j^2}\right) \cdot \nabla_i W_{ij}.$$
 (28)

Both equations (25) and (28) are symmetric with respect to the indices i and j which reduces the errors arising from the particle inconsistency problem, see for example Monaghan [14–16].

Due to its simplicity, the artificial viscosity outlined in Monaghan [17] is normally used. It provides the correct amount of viscosity to convert kinetic energy into heat at shocks and also helps to prevent unphysical penetration when two particles become close. When this term is added to the momentum equation of (28) the following equation is obtained

$$\frac{D\mathbf{u}_i}{Dt} = -\sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} + \Pi_{ij}\right) \cdot \nabla_i W_{ij} \qquad (29)$$

where Π_{ij} is given by

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{c}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}} & \mathbf{u}_{ij} \cdot \mathbf{r}_{ij} < 0\\ 0 & \mathbf{u}_{ij} \cdot \mathbf{r}_{ij} \ge 0 \end{cases}$$
(30)

where $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$,

$$u_{ij} = \frac{h\mathbf{u}_{ij} \cdot \mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2 + \nu^2}, \quad \bar{c}_{ij} = \frac{1}{2}(c_i + c_j), \quad \bar{\rho}_{ij} = \frac{1}{2}(\rho_i + \rho_j).$$

The expression Π_{ij} contains a linear difference in the velocity which produces both a shear and bulk viscosity. The quadratic term is required to handle high Mach number shocks, and hence we will be always setting β to zero. This viscosity has a number of good features. Firstly it is invariant in Galilean transformations, secondly it conserves total linear and angular momentum and finally it vanishes for rigid body rotations. The parameter ν is to prevent the denominator going to zero and is taken so that $\nu = 0.1h$. The laminar viscous stresses in the momentum equation can be formulated as a hybrid of a standard SPH first derivative with a finite difference approximation for the first derivative.

$$(\nu \nabla^2 \mathbf{u})_i = \nu \sum_j \frac{4m_j \mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{(\rho_i + \rho_j)(|\mathbf{r}_{ij}|^2 + \nu^2)} \mathbf{u}_{ij}$$
(31)

where ν is the kinetic viscosity. So the final momentum equation is

$$\frac{D\mathbf{u}_{i}}{Dt} = -\sum_{j} m_{j} \left(\frac{p_{i}}{\rho_{i}^{2}} + \frac{p_{j}}{\rho_{j}^{2}} \right) \cdot \nabla_{i} W_{ij} + \nu \sum_{j} \frac{4m_{j} \mathbf{r}_{ij} \cdot \nabla_{i} W_{ij}}{(\rho_{i} + \rho_{j})(|\mathbf{r}_{ij}|^{2} + \nu^{2})} \mathbf{u}_{ij}. \quad (32)$$

By contrasting equations (31) and (30) it is possible to compare the scaling of the kinetic viscosity to the parameter α .

The Sub-Particle Scale (SPS) model was first introduced by Gotoh *et al.* [18, 19]. The conservation of momentum equation can be written as:

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla P + g + \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla\tau \qquad (33)$$

and here the laminar term is treated in equation (31) and τ represents the SPS stress tensor. Boussinesq's hypothesis for the eddy viscosity states that the Reynolds stress tensor τ_{ij} is proportional to the trace-less mean strain rate tensor.

$$\frac{\tau_{ij}}{\rho} = \mu_t \left\{ 2S_{ij} - \frac{2}{3}k\delta_{ij} \right\} - \frac{2}{3}C_l \Delta^2 \delta_{ij} |S_{ij}|^2$$
(34)

where τ_{ij} is the sub-particle stress tensor, μ_t is the turbulence eddy viscosity, k the SPS turbulence kinetic energy, C_l is a constant equal to 0.0066 and S_{ij} the element of SPS strain tensor. DualSPHyscis uses the implementation suggested in [20]:

$$\frac{D\mathbf{u}_{i}}{Dt} = -\sum_{j} m_{j} \left(\frac{p_{i}}{\rho_{i}^{2}} + \frac{p_{j}}{\rho_{j}^{2}} \right) \cdot \nabla_{i} W_{ij}
+ \nu \sum_{j} \frac{4m_{j}\mathbf{r}_{ij} \cdot \nabla_{i} W_{ij}}{(\rho_{i} + \rho_{j})(|\mathbf{r}_{ij}|^{2} + \nu^{2})} \mathbf{u}_{ij}
+ \sum_{j} m_{j} \left(\frac{\tau_{i}}{\rho_{i}^{2}} + \frac{\tau_{j}}{\rho_{j}^{2}} \right) \cdot \nabla_{i} W_{ij}.$$
(35)

In Monaghan [21] the fluid in the SPH formulation was treated as a weakly compressible and hence an equation of state was used to determine the pressure in the fluid. The idea behind using this artificial compressibility is to reduce the prohibitively small time steps required to a reasonable level by slowing the speed of sound in the fluid. This reduced speed of sound should, however, be at least an order of magnitude faster than the maximum fluid velocity which keeps the density variations close. Monaghan applied the following equation of state for water to model free surface flows:

$$p = B\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \tag{36}$$

where γ is a constant taken to be 7 in more circumstances, ρ_0 is the reference density, and *B* is a problem dependent parameter, which limits the maximum change in density. The subtraction of 1 can remove the boundary effect for free surfaces and it can be seen that a small oscillation in the density may result in a large variation of the pressure. In the current code

$$B = \frac{c_0^2 \rho_0}{\gamma} \tag{37}$$

where c_0 is the speed of sound at the reference density.

The particles are updated using the XSPH variant according to Monaghan [22] which was introduced to stop SPH particles pass through each other. The idea Monaghan used was that each particle is moved with an average of the velocities of its neighbours. This reduces or even eliminates the number of particles passing through each other. The method is non-dissipative and conserves linear and angular momentum. This smoothing also has the further advantage of reducing local disorder.

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{u}_i + \epsilon \sum_j m_j \frac{2}{\rho_i + \rho_j} \mathbf{u}_{ji} W_{ij}$$
(38)

where ϵ is a user defined parameter usually taken to be 0.5.

Near the boundary and free surfaces, particles have a cut down smoothing kernel due to the absence of neighbouring particles. To correctly handle these conditions the kernel function W_{ij} or its gradient are modified. Two of the possible methods are kernel correction and kernel gradient correction. This is based on the work of Bonet and Lok [23] and Liu *et al.* [24]. The kernel is changed to enable polynomial functions of a given degree to be interpolated exactly. However Bonet and Lok consider the linear correction is unsuitable for computational purposes the constant correction

$$\mathbf{v}_{i} = \sum_{j} \frac{m_{j}}{\rho_{j}} \mathbf{v}_{j} W_{ij} / \sum_{j} \frac{m_{j}}{\rho_{j}} W_{ij}$$
(39)

Another option is to modify the kernel gradient used in the equation of motion.

$$\widetilde{\nabla} = L_j \nabla W_{ij} \tag{40}$$

$$L_i = M_i^{-1} \tag{41}$$

$$M_i = \sum_j \frac{m_j}{\rho_j} \nabla W_{ij} \otimes (\mathbf{r}_i - \mathbf{r}_j)$$
(42)

It should be noted that when the particle i is way from the boundaries and free surface that M_i is equal to the identity matrix and hence no correction is made to the kernel gradient. However, when the particle to close the distribution of particles around it does not remain symmetric and the correction kicks in. This correction is anisotropic since the off diagonal terms of the L_i involve both spatial coordinates.

In SPH while the simulations are realistic the pressure field of the particles can exhibit large pressure oscillations. Many approaches have been used to try and reduce the problem. These include correcting the kernel via equation (39) and development of incompressible solvers. However its also possible to apply a filter over the density of the particles and then use this new smoothed value. The Shepard filter is a correction which is applied after a user specified number of steps. The correction is as follows

$$\rho_i^{new} = \sum_j \rho_j \widetilde{W}_{ij} \frac{m_j}{\rho_j} = \sum_j m_j \widetilde{W}_{ij}$$
(43)

where the kernel has been corrected using a zeroth-order correction of equation 39

$$\widetilde{W}_{ij} = W_{ij} / \sum_{j} \frac{m_j}{\rho_j} W_{ij}.$$
(44)

A first order correction called moving least squares (MLS) was first developed by Dilts [25, 26]. Since it is first order a linear variation of the density field can be exactly reproduced.

$$\rho_i^{new} = \sum_j \rho_j W_{ij}^{MLS} \frac{m_j}{\rho_j} = \sum_j m_j W_{ij}^{MLS} \qquad (45)$$

where the corrected kernel is called by

$$W_{ij}^{MLS} = W_{ij}^{MLS}(\mathbf{r}_i) = \beta(\mathbf{r}_i) \cdot (\mathbf{r}_i - \mathbf{r}_j) W_{ij}.$$
 (46)

2.3 Time Marching SPH

To perform time-marching simulations each particle is updated using a global fixed time step Δt . For clarity, consider the following system of equation for density momentum and position:

$$\frac{d\rho_i}{dt} = D_i \tag{47a}$$

$$\frac{d\mathbf{u}_i}{dt} = \mathbf{F}_i \tag{47b}$$

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{U}_i. \tag{47c}$$

If U_i represents the the velocity contribution from particle *i* only then $U_i = u_i$. However it can also include the contribution of the neighbouring particles (via the XSPH correction).

The simplest method considered is the semi implicit Euler scheme. The scheme is semi implicit since only the position \mathbf{r} is updated in an implicit manner.

$$\rho_i^{n+1} = \rho_i^n + \Delta t^n D_i^n \tag{48a}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}^n + \Delta t^n \mathbf{F}_i^n \tag{48b}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{U}_i^{n+1} \tag{48c}$$

The leap-frog scheme gets it name by updating the positions \mathbf{r} and the velocities \mathbf{u} at interleaved points. The leapfrog scheme is second order in time and is written as:

$$\rho_i^{n+1} = \rho_i^n + \Delta t^n D_i^n \tag{49a}$$

$$\mathbf{u}_i^{n+1/2} = \mathbf{u}^{n-1/2} + \Delta t^n \mathbf{F}_i^n \tag{49b}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{U}_i^{n+1/2}.$$
 (49c)

The initial velocity is given by

$$\mathbf{u}_{i}^{-1/2} = \mathbf{u}_{i}^{0} - \frac{1}{2}\Delta t^{0}\mathbf{F}_{i}^{0}.$$
 (50)

The velocity at time step n is required when computing the forces at time step n and can be approximated using the midpoint rule

$$\mathbf{u}_{i}^{n} = \frac{1}{2} \left(\mathbf{u}_{i}^{n-1/2} + \mathbf{u}_{i}^{n+1/2} \right).$$
(51)

The Verlet integration [27] is a very common time integration scheme used in molecular dynamics. The basic idea is two expand two Taylor series for the position \mathbf{r}_i one forward and one backward in time.

$$\mathbf{r}_{i}^{n+1} = \mathbf{r}_{i}^{n} + \mathbf{u}_{i}^{n}\Delta t + \frac{1}{2}\mathbf{F}_{i}^{n}\Delta t^{2} + \frac{1}{6}\mathbf{s}_{i}^{n}\Delta t^{3} + O(\Delta t^{4})$$
(52a)
$$\mathbf{r}_{i}^{n-1} = \mathbf{r}_{i}^{n} - \mathbf{u}_{i}^{n}\Delta t + \frac{1}{2}\mathbf{F}_{i}^{n}\Delta t^{2} - \frac{1}{6}\mathbf{s}_{i}^{n}\Delta t^{3} + O(\Delta t^{4})$$
(52b)

The scheme employed for this work is split into two parts. Normally the variables are calculated using

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^{n-1} + 2\Delta t^n \mathbf{F}_i^n \tag{53a}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{u}_i^n + 0.5 (\Delta t^n)^2 \mathbf{F}_i^n$$
(53b)

$$\rho_i^{n+1} = \rho_i^{n-1} + 2\Delta t^n D_i^n \tag{53c}$$

Since these equations are not couple every few iterations (10 to 40) the variables are calculated using the explicit Euler scheme

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \Delta t^n \mathbf{F}_i^n \tag{54a}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \Delta t^n \mathbf{u}_i^n + 0.5 (\Delta t^n)^2 \mathbf{F}_i^n$$
(54b)

$$\rho_i^{n+1} = \rho_i^n + \Delta t^n D_i^n \tag{54c}$$

Symplectic time integration algorithm are designed for the numerical solution of Hamilton's equations and since these conserve the Hamiltonian and are widely applied in molecular dynamics where long term evolution is required. These schemes are also reversible in the absence of friction or viscous forces [28]

2.4 Moving Objects

In the application of SPH, there are two possible types of objects interacting with the fluid. The first have pre defined movement and the second objects that are moved by the fluid. For the first type, the objects interact with the fluid in such a way that the fluid is displaced by their movement, however, the motion of the object is independent to the fluid that is is moving thought. Objects of the second type have a two way interaction. For their motion the equations of rigid body dynamics are required.

Using Newton's second law the resulting force \mathbf{F} acting on a rigid body of mass m becomes

$$\mathbf{F} = m \dot{\mathbf{u}}_{cg}.$$
 (55)

The general moment equation about the centre of gravity is given by

$$G = h \tag{56}$$

there **G** is the resulting moment of the force **F** and **h** is the resulting angular momentum of the body about the centre of gravity. Now, considering the body has angular velocity ω with components ω_x , ω_y , and ω_z

$$\boldsymbol{\omega} = \mathbf{i}\omega_x + \mathbf{j}\omega_y + \mathbf{k}\omega_z \tag{57}$$

the velocity of a mass point of the rotating body becomes

$$\mathbf{V} = \mathbf{V}_{cg} + \boldsymbol{\omega} \times \mathbf{r} \tag{58}$$

hence the angular momentum of a rigid body about the centre of gravity is

$$h = \int \mathbf{r} \times (\mathbf{V}_{cg} + \boldsymbol{\omega} \times \mathbf{r}) \, dm =$$
$$\int \mathbf{r} \times \mathbf{V}_{cg} \, dm + \int \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm =$$
$$\int \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm. \quad (59)$$

In addition,

$$\int \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm =$$

$$\int (\boldsymbol{\omega} (\mathbf{r} \cdot \mathbf{r}) - \mathbf{r} (\boldsymbol{\omega} \cdot \mathbf{r})) dm =$$

$$\int (\boldsymbol{\omega} \mathbf{r}^2 - \mathbf{r} (\boldsymbol{\omega} \cdot \mathbf{r})) dm \quad (60)$$

and by substituting $\mathbf{r} = \mathbf{i}x + \mathbf{j}y + \mathbf{k}z$ and equation (60) into equation (59),

$$\mathbf{h} = \boldsymbol{\omega} \int (x^2 + y^2 + z^2) dm - \int \mathbf{r} (x\omega_x + y\omega_y + z\omega_z) dm \quad (61)$$

$$\mathbf{h} = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \int \begin{bmatrix} (y^2 + z^2) & -xy & -xz \\ -xy & (x^2 + z^2) & -yz \\ -xz & -yz & (x^2 + y^2) \end{bmatrix} dm, = \mathbf{I}\boldsymbol{\omega}$$
(62)

where **I** is defined as the inertia matrix. The diagonal terms I_{xx} , I_{yy} and I_{zz} are the moments of inertia while the off diagonal terms $-I_{xy}$, $-I_{xz}$ and $-I_{yz}$ are the products of inertia.

When a reference frame is fixed to the body (x_b, y_b, z_b) the inertia matrix remains constant. However the frame of reference now rotates with angular velocity ω . So in the body frame of reference equations (55) and (56) become

$$\mathbf{F} = m \frac{\partial \mathbf{V}_{cg}}{\partial t} + m\boldsymbol{\omega} \times \mathbf{V}_{cg}$$
(63)

and

$$\mathbf{G} = \frac{\partial \mathbf{h}}{\partial t} + \boldsymbol{\omega} \times \mathbf{h}.$$
 (64)

If the forces and moments are transformed into the body reference frame $(x_b, y_b, z_b) \mathbf{F} = \mathbf{i}F_{x_b} + \mathbf{j}F_{y_b} + \mathbf{k}F_{z_b}$ and $\mathbf{G} = \mathbf{i}G_{x_b} + \mathbf{j}G_{y_b} + \mathbf{k}G_{z_b}$ this yields

$$F_{x_b} = m \left(\dot{u} + \omega_y w - \omega_z v \right)$$

$$F_{y_b} = m \left(\dot{v} + \omega_z u - \omega_x w \right)$$

$$F_{z_b} = m \left(\dot{w} + \omega_x v - \omega_y u \right)$$
(65)

and

$$G_{x_b} = h_x + \omega_y h_z - \omega_z h_y$$

$$G_{y_b} = \dot{h}_y + \omega_z h_x - \omega_x h_z$$

$$G_{z_t} = \dot{h}_z + \omega_r h_y - \omega_y h_x$$
(66)

$$G_{x_b} = I_{xx}\dot{\omega}_x + (I_{zz} - I_{yy})\omega_y\omega_z - I_{xy}(\dot{\omega}_y - \omega_x\omega_z) - I_{xz}(\dot{\omega}_z - \omega_x\omega_y) - I_{yz}(\omega_y^2 - \omega_z^2) G_{y_b} = I_{yy}\dot{\omega}_y + (I_{xx} - I_{zz})\omega_x\omega_z - I_{xy}(\dot{\omega}_x - \omega_y\omega_z) - I_{xz}(\omega_x^2 - \omega_z^2) - I_{yz}(\dot{\omega}_z - \omega_x\omega_y) G_{z_b} = I_{zz}\dot{\omega}_z + (I_{yy} - I_{xx})\omega_x\omega_y - I_{xy}(\omega_x^2 - \omega_y^2) - I_{xz}(\dot{\omega}_x - \omega_y\omega_z) - I_{yz}(\dot{\omega}_y - \omega_x\omega_z)$$

$$(67)$$

2.5 Particle Interactions

In general the support of the kernel function is compact and only a finite number of particles are within this domain of this support. Three possible way to calculate the nearest neighbours, all-pair search, linked-list search algorithm and tree search algorithm are discussed below. The all-pair search, or brute force method, is a direct and simple method. For any given particle *i* calculate the distance r_{ij} to each particle *j*. If the distance r_{ij} is smaller than the dimension of the support for *i* then the particles *i* and *j* interact. This search is carried out on all *N* particles and so $O(N^2)$ operations are required. Hence the all-pair search is only computationally efficient of the total number of particles is very small.

The linked-list search algorithm works best for cases where the support radius is constant across all particles. It was shown by Monaghan and Gingold [29] that by using cells as a bookkeeping device the computational cost of particles interactions could be reduced. If all particles are assigned to bins and identified through a linked-list the computational time is reduced as only certain bins need to be checked. A temporary mesh is overlaid on the problem domain. The mesh spacing is selected to match the dimension of the support domain. Then for the particle *i*, its nearest neighbouring particles can only be in the same grid cell or the adjoining cells. Domínguez et al. [30] compared the performance for the two different methods to create list of neighbours namely the cell-linked list (CLL) and the Verlet list (VL). The CLL method set up a list linked to every cell and is the method shown in figure 1. The VL method creates a linked list for each vertex. It is usually implemented by generating a simple linked list to contain which particles are in each cell and a one dimensional array describing which cell each particle is in. They also looked into renumbering the particles so they are "close" in memory for better cache usage. They concluded that for parallel computations VL was better.

The main drawback of the linked-list search algorithm is when a variable smoothing length h is used. In this case the mesh spacing used to define the bin may not be optimal for every particle and hence the efficiency drops. This problem is overcome by using a tree search algorithm. Order trees are created according to particle position which are then searched to find the nearest neighbour particles. The tree method recursively splits the domain until only a single particle is in each leaf. The search is performed by centring a cube on the particle and checking the overlap of this cube with the volume represented by the node. Finally a check is required to see if the particle is in the support domain. The complexity of a tree search algorithm is order $N \log N$.

2.6 SPH Formulation

An open source version of an SPH solver is DualSPHysics [12, 13, 31] that was also used in this work. A simple flowchard of the employed SPH method can be seen in figure 3. The formulation for floating objects within DualSPHysics is based the work of Monaghan et al. [32] and examples in the literature have shown a good agreement between the SPH results and experimental data. However the method only contains a very basic formulation which does not include the correct physics when floating objects interact with other objects or solid walls and the friction forces are not taken into account. Further, the standard SPH method suffers from a lack of stability and hence uses an artificial viscosity term π_{ij} or by applying a density renormalisation. Both these fixes help to increase the regularity of the pressure field within the computational domain however as can been seen from figure 4 even with options used there are still defects in the pressure field making the pressure calculation at a point a non trivial task. Colagrossi [33] improved this pressure field via using a second order accurate interpolation with moving least square kernel.

3 RESULTS AND DISCUSSION

3.1 Simpler Cases

Simple flow cases were initially considered with SPH to allow the tuning of the various method parameters and assess the effect of different boundary conditions on the obtained results. SPH method, require careful use and systematic assessment of their numerical parameters since otherwise the obtained results may violate the conservation laws and lead to solutions with incorrect physics. Only some of the studies conducted in the preparation of the SPH method for helicopter ditching are presented in this work.

3.1.1 Effects of applying smoothing to an idealised problem

The effect of different filters to improve the smoothness of the solution was first investigated. Figures 5 and 6 show the effect of the Shepard and the moving Least-Squares filters on the obtained particle density for the case of a cube dropped on the surface of the water. The results suggest that for the cases of ditching filtering of the solution may be necessary to smooth out the pressure oscillations but global behaviour of the object remain largely unchanged.

3.1.2 Effect of boundary condition on the solution to an idealised problem

In addition to the solution smoothness, higher frequency oscillations may be present on the force and accelerations of floating bodies as a result of the applied boundary condition between the fluid particles and the particles attached to floating objects. The dynamic boundary condition of Dalrymple was first assessed. According to this condition, boundary particles are forced to satisfy the same equations as fluid particles. However, they do not move freely and so remain fixed in position, unless their position changes due to some external function, or rigidly under loading for floating objects. The repulsive boundary condition can also be used. This condition uses a repulsion function to ensure that a fluid particle can never cross a solid boundary. This is a much more involved boundary condition which also requires the geometric normals for every point on the boundary.

Figure 7 shows the results of a two dimensional cube drop. As can be seen the Dalrymple boundary condition produces a large oscillation in the velocity of the cube where the Repulsive force boundary condition does not. This also leads to a much higher deceleration in the initial phase of the impact which smaller oscillations at later times. It is therefore better for the Repulsive force boundary condition to be used, provided that the complex task of computing the surface normals for each particle can be performed. This task is trivial for a simple object with flat surfaces but can be harder for the case of ditching a helicopter due to the complex fuselage surface and its representation as a cloud of points with little or no connectivity information.

3.1.3 Choice of lattice type for floating objects and the fluid

Different lattice types can be used to represent the boundary and the fluid domain and the surface of floating object. This leads to a total of four possible combinations. The two different lattice types are shown in figure 8. The type first lattice has just a single row of particles representing the object while the type two lattice has a double row. For a given weight of object the type two lattice particles will have half the mass of the type one lattice. The type two lattice will also roll over quicker since in effect the cube has had two of its corners rounded off and hence the forces on either side of the square will lead to a moment causing the object to roll. The reason why a double layer of lattice is normally used for objects is that it is much harder for the fluid particles to penetrate the boundary walls.

A simple test case was used of dropping a $10 \times 10 \times 10$ m cube into the middle of a 30m square tank containing water 15 meters deep. The density of the cube is half that of the water and hence its equilibrium position will have half of the cube sitting out of the water. The final height should be at 15.55 m due to the small size of the tank making the displaced water increase the water level by about four percent. Figure 9 shows the four different combinations of lattice. The two cases where the fluid and boundary have the same type of lattice give very consistent results but the final position of the cube is about 1.5 m too high. The normal type two lattice boundary, and type one lattice fluid in this case gave a better final position at around 15m which is a little low. However the type one lattice boundary with a type two lattice fluid did something very different. For the first four seconds it run correctly even if it resulted with at a much higher position in the water. However, after this time fluid particles start leaking from the domain causing the height to drop. This is more clearly seen in the velocities as this configuration is not converging to zero.

Figure 10 shows the effect of increasing the number of particles. For the type one lattice boundary and fluid the solution becomes more oscillatory. However, the final equilibrium position was only 0.4m too high. The normal type two lattice boundary and type one lattice fluid nearly hit the bottom of the tank in this case, and also showed a drop in the centre

of gravity equilibrium under particles refinement, drifting further away from the correct answer. Extrapolating the results it would appear that the type two lattice boundary and type one lattice fluid will have an equilibrium position approximately as if the cube was of the same density as that of the water. The type one lattice boundary and fluid gave the correct behaviour under particle refinement and the particles did not leak. It was therefore used for helicopter ditching simulations.

3.1.4 Obtaining the correct equilibrium position with coarse particle density

As discussed in the previous section, even when the masses of the fluid and the body were correct the body did not have the correct buoyancy. This was because the body displaced too much fluid and hence ended to high in the water. Consider a tank of $3 \times 3 \times 3m$ with 2m of fluid in it. The results can be seen in table 1 for different particles sizes. As the particle size is reduced the mass of the fluid in the container converged to the modelled condition (18,000Kg). This is because particles are not placed exactly on the tank walls. This means that for he in the 0.1 particle case, instead of $30 \times 30 \times 20$ particles the SPH method is started with $29 \times 29 \times 19 = 15979$. For a particles size of 10cm there is a 11% error. This error scales linearly with the particles size so at a 1cm scale the error has been reduced to one percent.

Something similar happens when a fully submerged floating body is added to the tank. Particles are now placed on the faces of the cube. So for lattice type one, the number of particles in the floating object is

$$6 \times (n-1)^2 + 12 \times (n-1) + 8 \tag{68}$$

where n is the size of the cube divided by the particle size. The number of displaced fluid particles is

$$(n+1)^3$$
. (69)

For the buoyancy to be correct the amount of displaced fluid has to equal 1000Kg which is n^3 particles and hence the error is

$$\frac{3n^2 + 3n + 1}{n^3} \tag{70}$$

which again is order n^{-1} . The results of this can be seen in table 2. For a 10cm particles size a one meter cubed object will displace 1331Kg of fluid and hence buoyancy would cause the cube to move upwards. In this case the equilibrium position would have about 25cm of cube height above the water line. As the particle size is reduced this discrepancy is reduced linearly. The difference is slightly smaller for partially submerged objects as shown in table 3 because the discrepancy due to the upper surface has been removed.

3.1.5 Using complex geometries

Using simple geometric shapes like a cube is easy with either lattice type onto the body. However, for complex general cases, this is a non trivial problem. Consider for example an approximate fuselage of the AW101 helicopter shown in figure 11. Four different particle resolutions, 20cm, 10cm, 5cm and 2cm, were used to represent the fuselage and the results can be seen in figures 12 and 13. The computer code written to generate the surface is not discussed here but it is clear from the pictures how the algorithm works. Firstly a grid of particles is set up with the correct spacing. If the point is greater than half the particle resolution away from the surface then the point is discarded else it is kept. This means that for any give surface the particles may over- or under-approximate it by half their resolution. This effect can be seen in the closeup view near the radar dome and the bottom of the fuselage (bottom of figure 12). For the 20cm resolution (black squares) the points lie outside the radar dome and inside the bottom of the fuselage making the effective height of the radar dome bigger whereas in the 10cm resolution case this is reversed. The other drawback of this method is that surfaces with curvature will be represented by straight line segments. Even the 2cm resolution in the high curvature region of the radar dome has a very pronounced "staircase" effect. At present, the surface particle generation method can work with STL files produced by standard CAD systems and can extract a representation of any helicopter fuselage suitable for ditching computations with the SPH method.

3.2 Demonstration of SPH for Helicopter Ditching

The representation of the AW159 fuselage in the format used for ditching computations can be seen in figure 14. Based on the discussion of the previous paragraph a resolution of 5cm was used as a starting point, and the surface included 77 thousand particles.

Validation of the SPH method was then carried out against experiments, conducted at both the basins of DGA/TH (Val de Reuil) and ECN (Nantes) for a scaled model of the AW159 fuselage. The experiments provided data for the motion of the model as well as pressure and accelerometer readings from a few points on the model. Figure 15 shows the employed model as well as a still photograph of the model 0.2 seconds after a drop on the surface of water at sea-state zero and with the main rotor providing 67% of lift. This particular condition has also been simulated using the SPH method.

Figure 16 shows the pressure on the fuselage at different times. The pressure is scaled with the maximum value seen during the run. It can be seen that the water line does not get very far up the fuselage and hence most of the fuselage has zero pressure on it. However as can be seen from Figure 17 the comparison with the vertical velocity between the experimental data and the SPH simulation is good. The results show that the velocity and acceleration are predicted fairly well if the correct size of particles is used. The SPH results appear to capture well the peaks of the vertical acceleration and velocity with some noise present in the solution. The acceleration in particular, is reasonably well predicted apart from the initial impact for the 67% rotor model.

Figure 18 shows the AW159 fuselage drop at sea state 4 where the regular waves are 4 meters high with a wave slope of 0.1. The particle size was 1.5cm on the model scale which is larger than ideal but still required 24 hours of CPU for 2.6 seconds of real-time simulation. It can be seen in the figures that the fuselage sits high in the water. This is due to the fact that the rotor model is active during the whole of the computation producing lift in the vertical direction meaning in effect that the fuselage only has about 1/3 of its correct weight. Another reason for the high position is the larger particle size since the iso surface of the fluid is this distance below the real surface.

Figure 19 shows the motion of the fuselage more clearly. The main effect of the impact is in the Z direction while the impact effect is reduced by a quarter for the X direction. The initial vertical velocity looks very similar to a ditch into sea state zero but the effects of the waves can be clearly seen. The size of the waves is slightly low and the fuselage moves slowly towards the beach over time. The roll rate is also increasing but this can be due to the fact that the fuselage sits relatively high in the water.

4 CONCLUSIONS AND FUTURE WORK

In this paper SPH has been demonstrated for helicopter ditching. The method is mesh-free and in comparison to traditional CFD methods appears to be easier to use due to the lack of the mesh-generation step. On the other hand, the results of SPH depend heavily on the use of appropriate particle resolution, flow model parameters, and correct boundary conditions between the solid and fluid particles. Simple cases like the drop of a cube on the surface of water were initially used for the investigation of the effects of all the aforementioned parameters.

Once the effect of the boundary conditions and flow model parameters were quantified, the simulation of an AW159 ditching was attempted. The case of a vertical drop of the fuselage at sea-state zero was attempted and the results showed an overall fair agreement with the measure mens regarding the velocity, acceleration and position of the fuselage versus time. Further cases included a vertical drop on the crest of a wave to demonstrate the potential of the method.

Overall, SPH was found satisfactory for the ditching task even though some user experience and careful selection of the model parameters were necessary. In the future, efforts will be directed towards establishing a practical list of criteria for the selection of the numerical parameters of the SPH method so that routine analyses of helicopter ditching can be performed.

Acknowledgements

The financial support via the NGVL project of AgustaWestland and the Business Innovation and Skills Department of UK is gratefully acknowledged.

REFERENCES

- T. von Karman, "The impact of seaplanes floats during landing," Tech. Rep. NASA Technical note no 321, NASA, 1929.
- [2] H. Wagner, "Phenomena associated with impacts and sliding on liquid surfaces," *Journal of Applied Mathematics and Mechanics*, vol. 12, pp. 193–235, 1932.
- [3] H. Wagner, "Landing of seaplanes," Tech. Rep. NASA Technical Memorandum no 622, NASA, 1932.
- [4] C. S. L., "Theoretical investigations on slamming of cone-shaped bodies," *Journal of Ship Research*, vol. 13, pp. 276–283, 1969.

- [5] Y. Scolan and A. A. Korobkin, "Three-dimensional theory of water impact. part 1. inverse wagner problem," *Journal of Fluid Mechanics*, vol. 440, pp. 293–326, 2001.
- [6] Y. Scolan and A. A. Korobkin, "Energy distribution from vertical impact of a threedimensional solid body onto the flat free surface of an ideal fluid," *Journal of Fluids and Structures*, vol. 17, pp. 275–286, 2003.
- [7] R. Zhao and O. M. Faltinsen, "Water entry of twodimensional bodies," *Journal of Fluid Mechanics*, vol. 246, pp. 593–612, 1993.
- [8] A. A. Korobkin, "Analytical models of water impact," *European Journal of Applied Mathematics*, vol. 15, pp. 821–838, 2004.
- [9] A. Tassin, N. Jacques, A. El Malki Alaoui, A. Nême, and B. Leblé, "Assessment and comparison of several analytical models of water impact," *International Journal of Multiphysics*, vol. 4, no. 2, pp. 125–140, 2010.
- [10] L. B. Lucy, "A Numerical Approach to Testing the Fission Hypothesis," *The Astronomical Journal*, vol. 82, no. 12, pp. 1013–1924, 1977.
- [11] R. A. Gingold and J. J. Monaghan, "Smoothed Particle hydrodynamic: theory and application to non-spherical stars," *Monthly Notices of the Royal Astronomical Society*, vol. 181, pp. 375–389, 1977.
- [12] M. Gomez-Gesteira, B. Rogers, A. Crespo, R. Dalrymple, M. Narayanaswamy, and J. Dominguez, "SPHysics development of a free-surface fluid solver part 1: Theory and formulations," *Computers & Geosciences*, vol. 48, pp. 289 299, 2012.
- [13] M. Gomez-Gesteira, A. Crespo, B. Rogers, R. Dalrymple, J. Dominguez, and A. Barreiro, "SPHysics - development of a free-surface fluid solver - part 2: Efficiency and test cases," *Computers & Geosciences*, vol. 48, pp. 300 – 307, 2012.
- [14] J. J. Monaghan, "Why particle methods work," SIAM Journal of Scientific and Statistical Computing, vol. 3, no. 4, pp. 422–433, 1982.
- [15] J. J. Monaghan, "Particle Methods for Hydrodynamics," *Computer Physics Reports*, vol. 3, pp. 71–124, 1985.
- [16] J. J. Monaghan, "An introduction to SPH," *Computer Physics Communications*, vol. 48, pp. 89–96, 1988.
- [17] J. J. Monaghan, "Smoothed Particle Hydrodynamics," *Annual review of Astronomy and Astrophysics*, vol. 30, pp. 543–574, 1992.
- [18] H. Gotoh, T. Shibahara, and T. Sakai, "Sub-particlescale turbulence model for the MPS method - Lagrangian flow model for hydraulic engineering," Advanced Methods for Computational Fluid Dynamics, vol. 9-4, pp. 339–347, 2001.

- [19] H. Gotoh, S. Shao, and T. Memita, "SPH-LES model for numerical investigation of wave interactions with partially immersed breakwater," *Coastal Engineering Journal*, vol. 46, pp. 39–63, 2004.
- [20] R. A. Dalrymple and B. D. Rogers, "Numerical modeling of water waves with the SPH method," *Coastal Engineering*, vol. 53, pp. 141–147, Feb. 2006.
- [21] J. J. Monaghan, "Simulating free surface flows with SPH," *Journal of Computational Physics*, vol. 110, pp. 399–406, 1994.
- [22] J. J. Monaghan, "On the problem of penetration in particle methods," *Journal of Computational Physics*, vol. 82, pp. 1–15, May 1989.
- [23] J. Bonet and T. S. L. Lok, "Variational and momentum preservation aspects of Smooth Particle Hydrodynamic formulations," *Computer Methods in applied mechanics and engineering*, vol. 180, no. 1-2, pp. 97–115, 1999.
- [24] W. K. Liu, S. F. Li, and T. Belytschko, "Moving least squares Galerkin methods (I) methodology and convergence," *Computer Methods in Applied Mechanics and Engineering*, vol. 143, pp. 113–154, 1997.
- [25] G. A. Dilts, "Moving least-squares particle hydrodynamics I: consistency and stability," *International Journal for Numerical Methods in Engineering*, vol. 44, no. 8, pp. 1115–1155, 1999.
- [26] G. A. Dilts, "Moving least-squares particle hydrodynamics II: conservation and boundaries," *International Journal for Numerical Methods in Engineering*, vol. 48, no. 10, pp. 1503–1524, 2000.

- [27] L. Verlet, "Computer "experiments" on classical fluids. I. thermodynamical properties of Lennard-Jones molecules," *Physical Review*, vol. 159, pp. 98–103, July 1967.
- [28] B. J. Leimkuhler, S. Reich, and R. D. Skeel, "Integration methods for molecular dynamics," in *IMA Voulmes in Mathematics and its applications*, pp. 161–186, Springer Verlag, 1996.
- [29] J. J. Monaghan and R. A. Gingold, "Shock simulation by the particle method SPH," *Journal of Computational Physics*, vol. 52, no. 2, pp. 374–389, 1983.
- [30] J. M. Domínguez, A. J. C. Crespo, M. Gómez-Gesteira, and J. C. Marongiu, "Neighbour lists in smoothed particle hydrodynamics," *International Journal for Numerical Methods in Fluids*, 2010.
- [31] A. C. Crespo, J. M. Dominguez, A. Barreiro, M. Gómez-Gesteira, and B. D. Rogers, "GPUs, a new tool of acceleration in CFD: Efficiency and reliability on smoothed particle hydrodynamics methods," *PLoS ONE*, vol. 6, no. 6, 2011.
- [32] J. J. Monaghan, A. Kos, and N. Issa, "Fluid motion generated by impact," *Journal of waterway, port, coastal and ocean engineering*, vol. 129, pp. 250–259, 2003.
- [33] A. Colagrossi and M. Landrini, "Numerical simulation of interfacial flows by Smoothed Particle Hydrodynamics," *Journal of Computational Physics*, vol. 191, pp. 448–475, 2003.

Particle	Number of fluid	Mass of	Total Mass	Percentage
size(m)	particles	particle	of fluid (kg)	error
0.1	15979	1.000	15979	11.2
0.05	135759	0.125	16970	5.7
0.02	2197899	0.008	17583	2.3
0.01	17790799	0.001	17791	1.16

Table 1: Number of particles in 18 cubic meters of water.

Table 2: Mass of displaced fluid for a fully submerged $1 \times 1 \times 1$ m object.

Particle	Number of fluid	Mass of	Number of body	Particles	Mass	Percentage
size (m)	Particles in tank	Particle	Particles	displaced (kg)	displaced	Error
0.1	15979	1.000	602	1331	1331	33.1
0.05	135759	0.125	2402	9261	1158	15.8
0.02	2197899	0.008	15002	132651	1061	6.1
0.01	17790799	0.001	60002	1030301	1030	3.0

Table 3: Mass of displaced fluid for a partially (20%) submerged $1 \times 1 \times 1$ m object.

Particle	Number of fluid	Mass of	Number of body	Particles	Mass	Percentage
size (m)	Particles in tank	Particle	Particles	displaced	displaced (kg)	Error
0.1	15979	1.000	602	242	242	21.0
0.05	135759	0.125	2402	1764	221	10.5
0.02	2197899	0.008	15002	26010	208	4.0
0.01	17790799	0.001	60002	204020	204	2.0



Figure 1: Overview of the SPH Method.



Figure 2: Example of 4 commonly used kernels and there Gradients used in SPH methods.



Figure 3: Schematic of the SPH Code.



Figure 4: The non regular density field in an SPH computation for the case of a cube dropped on the surface of water.



Figure 5: Effect of the Shepard filter on the smoothness of the solution.



Figure 6: Effect of the moving least squares filter on the smoothness of the solution.



Figure 7: Effect of the boundary condition on the Acceleration and velocity of the dropped cube.



Figure 8: Two different lattices for a square floating object.



Height above the bottom of the tank

Vertical velocity of the cube

Figure 9: Effect of the four different methods on the height and vertical velocity of the centre of gravity of a cube.



Figure 10: Effect of increasing the number of particles in the consistent lattice 1,1 and lattice 2,2 cases.



Approximate surface of the AW101 fuselage

Mesh density in STL file





Closeup of the radar dome with all resolutions

Figure 12: Effect of particle resolution on a section through the fuselage.



 $5~\mathrm{cm}$ particle resolution

 $2\ {\rm cm}$ particle resolution

Figure 13: Effect of particle resolution on the approximation of the fuselage shape near the middle of the AW101 fuselage.



Figure 14: Particles used to represent the AW159 fuselage.



(a) The Experimental model



(b) Still of drop test at time 0.2 seconds

Figure 15: Vertical drop of the AW159 Fuselage with 69% lift into sea state zero.



Figure 16: Pressure on the underside of the AW159 fuselage with 67% lift from the basic rotor model.



(a) Velocity in the vertical direction (b) Acceleration in the vertical direction

Figure 17: Comparison between SPH and experimental data for a vertical AW159 Fuselage drop into sea state 0.





Figure 18: A vertical drop into sea state 4 and different times with the fuselage hitting the crest of the wave.



Figure 19: Vertical drop on sea state 4 and different times with the fuselage hitting the crest of the wave.