

Whale To Turbine Impact Using The GPU Based SPH-LSM Method

S. M. Longshaw, P. K. Stansby & B. D. Rogers
School of Mech., Aero., & Civil Engineering
The University of Manchester
Manchester, UK

Abstract—Underwater turbines are an interesting solution for future energy demands. An important design consideration is the potential for impact with their surrounding natural environment. This work focuses on the dynamic event of impact with a large sea creature, the minke whale is chosen as a test case as it is a well-established species in the waters being considered. The simulation goals are to capture the macroscopic event of whale to turbine impact whilst both are submerged in fluid, with the primary focus being the hydrodynamic effects of the fluid on the whale.

A particle based elastic modelling technique named lattice spring modelling (LSM) that is well suited to implementation on a graphics processing unit (GPU) is presented, this is coupled with smoothed particle hydrodynamics (SPH) to create the SPH-LSM method. The proposed solution offers unique capabilities when compared to alternatives. This approach dynamically assigns identity to particles depending upon their current position, with an interface formed by particles that move according to different sets of governing equations. The LSM presented aims to capture a linearly elastic model using first order Hookean response, however the basic methodology can be used as a framework to support more complete constitutive models.

The SPH and LSM methods are described and their GPU implementation detailed, including a method to derive a spring constant based on a Young's modulus and Poisson's ratio. A test case is provided in the form of a 3-D beam deflection test and results compared favourably against Timoshenko beam theory. A preliminary case showing an impact event between a geometrically realistic minke whale described using LSM and a rigid rotating turbine is presented, showing the method to be technically viable.

I. INTRODUCTION

As underwater turbines grow in popularity for harvesting tidal energy, it becomes necessary to consider potentially damaging operational events such as impact between the rotor and surrounding debris and wildlife. Underwater impacts demand different simulation considerations than those in air because the damping effect of the surrounding fluid has more effect, also because of the nature of life that has evolved within the ocean, impact between a turbine and large creatures of notable mass is more likely.

This work aims to consider the specific case of impact between a minke whale and a large underwater turbine. The minke whale has been chosen as it is the dominant large species within the considered waters. This initial work aims to gain an understanding of the macroscopic features of an impact between whale and turbine and seeks insight into

considerations such as how the impact event will affect the kinematics of the whale and what influence the fluid dynamics of the ocean will have, with questions such as *is there a minimum speed that the whale can travel at before it is no longer swept clear of the turbine due to the wake it creates.*

This problem demands a number of simulation elements to be combined, the whale must be modelled as a deformable body consisting of a number of material properties (i.e. skin, blubber and bone), which must be transient throughout the simulation domain. The surrounding fluid also needs to be simulated. It is also necessary to be able to model the turbine and capture the interaction between its moving rotor blades and the surrounding fluid and whale. While impact modelling is often undertaken using mesh-based methods, such as Finite Element Modelling (FEM) [1], [2], the intricacies of this scenario require a different approach.

For this model, certain simplifications have been assumed. Primarily this means that the whale is considered to be homogenous within each of its discrete materials (i.e. all bone has the same material properties); the model is also designed such that the whale is considered to be a linearly elastic object that cannot be fractured. In reality this simplification would not hold in a number of scenarios, such as when part of the skeleton breaks or skin tears. In these scenarios a number of constituent models would be required, such as the Mooney-Rivlin [3], [4] formulation to capture non-linear hyperelastic deformation, followed by an appropriate model to capture plastic deformation as well as a model to determine failure. Modelling to this level of detail was considered beyond the scope of this work.

The Lagrangian meshless method smoothed particle hydrodynamics (SPH) is used here as the starting point, as its discrete nature is well suited to the problem. It is able to capture highly dynamic fluid mechanics, as well as fluid structure interaction; however the method to simulate a deformable whale in a manner that can be easily coupled with SPH is less clear. While there currently exist commercial software packages that provide general frameworks with which to define scenarios using SPH and deformable boundaries, their SPH implementations are not yet sufficiently refined to perform simulations involving upwards of a million particles within a reasonable time frame. This is a requirement here due to the large computational domain demanded by the scale of

the turbine and whale. The starting point for this work is therefore the graphics processing unit (GPU) based SPH solver *DualSPHysics* [5]. This code is open-source and tested against cases such as *SPHERIC Test-case 2* (3-D Dam-breaking) [5], importantly it is also able to process models with upwards of a million particles on modest computing hardware within a reasonable time-frame.

To enable fluid structure interaction (FSI) between a deformable whale and its surrounding fluid, as well as rigid SPH boundaries, there are two potential approaches. The first is to utilise a method able to capture linear elastic deformation, such as Finite Element Modelling (FEM) or the Immersed Boundary Method (IBM) [6], and then couple this with the SPH formulation via a boundary method. This was discounted however, as it was decided that coupling by way of a known boundary method would introduce extra considerations in terms of how effectively it captured a moving and deforming boundary, computational considerations were also made in that the existing *DualSPHysics* framework consumes the majority of the processing capacity available within a GPU and therefore finding a way to incorporate an elastic model depended upon re-utilisation of the existing structure rather than the significant addition of a distinct second model. The option of developing a particle based method was therefore selected.

The inspiration for this comes from the Lattice Spring Model (LSM) [7], [8], which involves the interconnection of rigid bodies in a conceptual lattice and allows material properties, such as Young's modulus or Poisson's ratio to be used to define elastic behaviour. The lattice is used as a way to determine neighbour interactions and stress and strain tensors are solved locally for both normal and shear forces. In using a particle with associated mass instead of a rigid body it is possible to take the LSM concept and apply it directly within an SPH framework. However, when implementing the method on a GPU it becomes necessary to modify the manner in which local deformation forces are calculated per particle. Solving the full stress and strain tensors in a lattice formed from hundreds of thousands of particles is best avoided as this constitutes a notable number of sparse matrix operations per time-step, which in turn introduces an undesirable memory access pattern for a GPU based solution. The method presented here uses a methodological simplification by considering only normal forces and capturing the effects of shear through the selected lattice, different applications of the method may wish to restore a more rigorous approach at the expense of computational overhead.

A novel mechanism to allow particles involved in LSM computations to interact with SPH fluid and boundary particles is specified, this is achieved by dynamically defining the purpose of particles within the simulation such that LSM particles that fall within the smoothing kernel of an SPH particle become SPH-LSM. This special type of particle experiences forces from both the underlying LSM calculations as well as the Navier-Stokes SPH formulations, thus producing an

intrinsic interface between the two methods.

Results showing the verification of the LSM portion of the model are provided in the form of a 3-D beam deflection experiment; where a cantilever beam, discretised at varying levels and with two associated strengths, is subjected to a point force and the results compared against Timoshenko beam theory [9], where good agreement is seen. A more dynamic case involving interaction between LSM and SPH particles is also included and shows an impact between a realistically modelled minke whale and turbine. This initial case demonstrates the model and forms the basis for future comparison works with upcoming laboratory data.

This paper is comprised first of a section that describes the SPH formulations used. This is followed by a section that provides an introduction to the SPH-LSM method and then describes its specific detail, including the LSM model, calculation of global spring and damping constants and how LSM and SPH are integrated. Two sections follow this, the first providing beam deflection results and the second the whale to turbine impact. Finally the method and results are concluded.

II. SPH FORMULATION

The SPH formulations used within this work are unchanged from those found in the standard release of *DualSPHysics*, using weakly-compressible SPH. The formulations solve a discretised variant of the Navier-Stokes equations for a Newtonian fluid, designed to conserve momentum and mass for each particle in the system. The equation for the conservation of momentum in SPH terms is

$$\frac{D\vec{v}_a}{Dt} = - \sum_b m_b \left(\frac{P_b}{\rho_b^2} + \frac{P_a}{\rho_a^2} \right) \vec{\nabla}_a W_{ab} + \vec{f} + \vec{\Theta}_{ab} \quad (1)$$

where a and b are a particle pair, \vec{v} is the velocity, P the pressure, ρ the density, m the mass, W_{ab} is a distance based kernel function, \vec{f} represents additional external forces such as gravity and $\vec{\Theta}_{ab}$ is a term to represent the effects of viscosity.

While viscous effects are typically handled using an *artificial* scheme [10], it is generally accepted that the scheme can be difficult to quantify in terms of its realism. Models that aim to define viscous effects realistically have been developed, a notable example is the laminar scheme [11], which is used here. The laminar viscosity term replaces $\vec{\Theta}_{ab}$ in equation 1 and is written as

$$\vec{\Theta}_{ab} = \sum_b m_b \left(\frac{4v_0 r_{ab} \vec{\nabla}_a W_{ab}}{(\rho_a + \rho_b) |\vec{r}_{ab}|^2} \right) \vec{v}_{ab} \quad (2)$$

where v_0 is the kinematic viscosity.

The smoothing kernel (W_{ab}) used is the quintic Wendland kernel [13] as it has been shown to better capture fluid dynamics in some scenarios [12] and is used here to maximise the potential quality of SPH results.

As the mass of each particle remains constant, density change is calculated according to the conservation of mass equation, which can be expressed in SPH form as [10]

$$\frac{D\rho_a}{Dt} = \sum_b m_b \vec{v}_{ab} \cdot \vec{\nabla}_a W_{ab} \quad (3)$$

in order to determine pressure, the model is closed using an equation of state [14]

$$P_a = B \left[\left(\frac{\rho_a}{\rho_0} \right)^\gamma - 1 \right] \quad (4)$$

where γ represents the compressibility of the fluid and $B = \frac{c_0^2 \rho_0}{\gamma}$, where c_0 is equal to the speed of sound at the initial density ρ_0 .

Time integration is performed using a second-order explicit symplectic scheme, consisting of two distinct predictor and corrector steps [16] and includes the XSPH correction [15]. Variable time-stepping is used to maintain stability. Primarily this is based on the typical Courant-Friedrich-Levy (CFL) condition and forcing terms, however a new criterion is added to maintain LSM stability, therefore the complete mechanism is discussed in further detail in Section III-D.

III. THE SPH-LSM METHOD

The SPH-LSM method sees weakly-compressible SPH combined with a particle based method able to handle elastic deformation, lattice spring modelling (LSM). LSM is suitable for this application as it is able to handle deformation due to stress and strain localised at discrete locations. The method relies on a conceptual lattice which interconnects bodies or particles in a pattern suitable to capture deformation in each of the six degrees of freedom.

The method, as presented here, is also designed specifically for processing via GPU. This is an important consideration as the computational demand of combining both SPH and LSM is significant. The highly parallel nature of the GPU is ideally suited to solving this problem as long as the elastic method is also designed to parallelise well (i.e. it relies on a system of particles that can be considered independently of each other). Another important consideration that GPU implementation brings is that of numerical accuracy. Typically the magnitude of forces involved in an SPH simulation are reasonable within the precision available according to single precision floating point. However, solid simulations designed to capture elasticity can produce significantly higher forces, the reason for which can be identified when the case of $F = EL$ is considered where E is a Young's modulus and L a descriptive length. Even at a relatively low Young's modulus of 0.8 GPa (typical for polyethylene HDPE) and for a small descriptive length of 0.001 m, the forces involved are likely to be of the order of magnitude of 1×10^5 . One solution for this is to utilise double precision, however this is undesirable when programming for the GPU, therefore all work presented uses *compensated summation* [19] to reduce floating point round-off error in both LSM and SPH iterative summation processes.

A. Lattice Spring Modelling

The premise of LSM is to discretise the volume of an elastic object in to a set of interconnected discrete particles, each of which represents a portion of the total volume and therefore carries an associated mass. These move according to an underlying set of governing equations, in this case the

first-order linear approximation, Hooke's law (plus a damping term). The second-order symplectic numerical integration scheme described in Section II is an appropriate solution for this.

The most obvious consideration when using this kind of technique is the form that the lattice takes. As the intention is to use the functionality of the pre-processing tool that exists as part of the *DualSPPhysics* package, it is necessary to start from a point of cubic particle distribution, it is also necessary that any methods used are able to scale according to different particle spacings. In a cubic lattice a 3-D volume is comprised of n particles, which form a set of cubes with side length equal to L , the particle spacing. The first step therefore is to connect each of the particles to its nearest neighbours, these connections allow axial distortion to be captured and are henceforth named structural connections. In the case of shear forces, only having structural connections will result in an object which simply collapses, it is therefore necessary to also have a way to capture shear.

In work defining a 3-D LSM method based around a Discrete Element Model, named DLSM [8], Zhao et al. handle axial and shear forces independently and examine a number of possible interconnect patterns. They utilise structural connections to capture axial loads and then add additional connections specifically to capture shear forces. They experiment with patterns defined by creating connections for each particle for all of its neighbours that fall within a distance of $\sqrt{2}L$, they call this *Cubic II*. Their work shows that, for cubic based lattices this layout produces the best results in terms of its ability to represent Poisson's ratio. As it is also the connection pattern with the least redundancy and therefore the least computational overhead, it is used here. An example of this pattern can be seen in Figure 1 where the case of 8 particles is shown in 3-D, producing 12 structural interconnects of length L and 12 of length $\sqrt{2}L$ across each face, only two of which are shown in the figure for brevity.

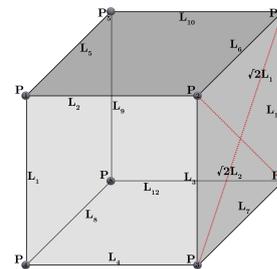


Fig. 1. The *Cubic II* lattice structure for 8 points in a cubic layout. 12 structural connections of length L are shown as solid black lines and 2 of the 12 shear connections of length $\sqrt{2}L$ are shown as dotted red lines, with the rest omitted for clarity.

The lattice is calculated as a pre-processing step before the simulation evolves. This is achieved using a k-dimensional tree structure [17] by performing a search around each particle considered to be part of a set contributing to a single elastic object. Neighbouring particles found to fall within a radius

of $\sqrt{2}L$ are interconnected, this ensures a computational complexity of $O(n)$. The resulting lattice is stored as a set of linked lists suitable for use within a GPU implementation [18].

Although this work builds on that of the DLSM method, for reasons mentioned in Section I, it is undesirable to fully resolve the stress and strain tensors for each particle in the LSM within this GPU based solution. Therefore the method handles linear elasticity by considering all interconnects to be structural and capturing the effects of shear by modifying a global spring constant locally for each particle based on the geometric layout of its interconnects, as well as relying on the interconnect pattern. Due to this, a single spring constant is calculated according to the algorithm defined for the DLSM to find the macroscopic normal constant. While this simplification undoubtedly introduces a compromise between computational performance and model accuracy, it is deemed appropriate, as capturing minute deformation is less important than quickly determining the macroscopic effects of elasticity, such as the effects on bulk motion.

Calculating the force F that is imposed on each particle due to any external factors and its interconnects is conceptually simple in that it follows the basic principle of Hooke's law $F = -kx$, where k is a spring constant and x a change in distance. The key question is how to derive a local value for k at each interconnect that will produce the correct global elastic response for the object as a whole. This is achieved here by first deriving a set of global spring constants that represent the entire interconnect system in each direction and then locally correcting these per particle, also in each direction. The basis behind this process is the manner in which sets of springs behave in terms of the strength of a system (i.e. springs in parallel are additive and those in series deductive), the same principle holds for sets of dampers therefore the same process is also applied to find damping coefficients. Calculation of these values is described further in Sections III-B and III-C, where they are found according to a set of pre-computed global constants (k_{global} and b_{global}) and they are referred to here as k_{local} and b_{local} respectively.

The force \underline{F}_{ab} is calculated at particle a according to its connected particles b using

$$\underline{F}_{ab} = \sum_b \left(-[K]_{ba} \underline{r}_{ba} \right) - \left([B]_{ba} (\underline{v}_{ba} \cdot \underline{r}_{ba}) \left(\frac{\underline{r}_{ba}}{|\underline{r}_{ba}|} \right) \right) \quad (5)$$

where r_{ba} is the difference between the position of the two particles and v_{ba} their velocity difference. $[K]_{ba}$ and $[B]_{ba}$ are diagonal matrices that represent local spring and damping constants locally to particle a in each direction and can be described as

$$[K]_{ba} = \begin{bmatrix} (k_{local})_x & 0 & 0 \\ 0 & (k_{local})_y & 0 \\ 0 & 0 & (k_{local})_z \end{bmatrix} \quad (6)$$

$$[B]_{ba} = \begin{bmatrix} (b_{local})_x & 0 & 0 \\ 0 & (b_{local})_y & 0 \\ 0 & 0 & (b_{local})_z \end{bmatrix} \quad (7)$$

As k_{local} and b_{local} are derived from the pre-calculated global values k_{global} and b_{global} , it is possible for these to vary per particle. In the case of two particles being connected and having different global values, the values are averaged per connection so as to provide a smooth interface within the model. This provides the ability to define different sections of an object as having different elastic properties and therefore enables the functionality to meet the requirement of modelling a whole comprised of different connected materials.

B. Locally Corrected Spring and Damping Constants

To ensure the spring constant of each particle gives an appropriate contribution to the global constant, it is necessary to consider the combinatory effects of systems of springs. When linked in series, the strength (and therefore the effective constant $k_{effective}$), for a system of n springs follows the rule $\frac{1}{k_{effective}} = \frac{1}{k_1} + \frac{1}{k_2} + \frac{1}{k_{...n}}$, therefore in order to ensure the actual spring constant used is as per the global definition, this needs to be compensated for by increasing the value of k_{local} so that it equals $n k_{effective}$, the same applies to dampers in series.

For the problem presented here it is possible to state that, for each particle there exists a system of n springs connected to each other in series, however each does not have an equal weighting within the system from a force resolution perspective. This can be resolved by deriving a value for k_{local} in each direction and is achieved for each particle a , according to its neighbours b , using

$$(k_{local})_x = (k_{global})_x \sum_b |\cos((\theta_{ab})_x)| \quad (8)$$

$$(k_{local})_y = (k_{global})_y \sum_b |\cos((\theta_{ab})_y)| \quad (9)$$

$$(k_{local})_z = (k_{global})_z \sum_b |\cos((\theta_{ab})_z)| \quad (10)$$

where k_{global} represents the global constants associated with particle a and θ_{ab} is the angle formed between them in each direction, found according to the direction cosine. While this is computationally significant in 3-D, it is less so than fully resolving stress and strain tensors per particle. An important programming consideration is to ensure that errors in the calculated angles (i.e. those greater than one or less than zero) caused by floating point rounding error are determined and corrected. The same process occurs to determine the local damping constants b_{local} .

C. Deriving a Global Spring Constant

In order to be able to capture linear elastic response, it is necessary to define a global spring constant that describes the nature of the lattice. In the work of Griffiths & Mustoe [7] and later Zhao et al. [8], the authors derive a method to correlate macroscopic material properties, such as Young's modulus and Poisson's ratio, to a microscopic spring constant for resolving normal elastic forces at each discrete location in the lattice. The same method is used here but considers the problem differently so as to allow for the discrepancy introduced by not considering shear forces directly.

Zhao et al. consider the effects of the lattice on the resultant spring constant by taking the square length of all interconnects for each discrete location over its volume, they refer to this value as α^{3D} and use it in the formulation

$$k_{normal} = \frac{3E}{\alpha^{3D}(1-2\nu)} \quad (11)$$

where E is the Young's modulus and ν the Poisson's ratio. In utilising the cubic nature of the lattice as presented here and considering the LSM as a collection of discrete cubes, this method can be used to determine the constant of one of the cubes in the system. Once this value is known the same principles of how systems of springs relate to each other (as used in Section III-B) can be applied by considering each cube to represent a single spring. This simplification is then utilised to calculate a single global value, in each direction, that approximates the complete system for any given object.

When calculating the spring density value, it is necessary to remember that lattice connections are designed to only capture normal forces, however to ensure that shear is still physically captured, cross connections of length $\sqrt{2}L$ are still present. These connections do not provide the same force weighting as structural members (for the reasons seen in Section III-B) and therefore a modification is needed to account for this. This is achieved by considering a difference in length between the different spring types and modifying the volume that the spring density is taken over. The result (ϵ) is found such that it also accounts for connections of length $\sqrt{2}L$ that cross outside of the given volume at the edge (connections of length L will always reside within the given volume), this is calculated according to

$$\epsilon = \frac{L}{x\sqrt{2}L} \quad (12)$$

where x accounts for the aforementioned edge effects and is found using $x = \frac{1}{12} \frac{n_{\sqrt{2}L}}{n_p}$ in 3-D, where $n_{\sqrt{2}L}$ is the total number of interconnects of that length and n_p the total number of particles.

In the case of certain objects, such as a rectangular beam, this process is sufficient, as the elastic response for a beam with the same dimensions but differing strengths is linear. However, when considering more complex geometries it is known that elastic response is not always linear, i.e. in the case of a sphere response follows $E^{\frac{2}{3}}$ [20]. This will vary depending upon the object being modelled and so it is considered here to be a case specific shape coefficient and is referred to as λ . Future work should look to derive this value from the geometry of the object as part of the process of building the lattice. The initial value ($k_{initial}$) for a given lattice is therefore found according to

$$k_{initial} = \left(\frac{3E}{\frac{\sum l_{ab}^2}{\epsilon V} (1-2\nu)} \right)^\lambda \quad (13)$$

where $\sum l_{ab}^2$ is the sum of the squared length of all lattice connections, V is the total cubic volume of the object, found by first counting the number of connections of length $\sqrt{3}L$ (where L is the initial particle spacing) that the lattice forms

and then (in 3-D) dividing this by 8, giving the number of cubic volumes n , the total volume is then found according to $V = nL^3$.

This results in a spring constant for a single cubic volume from the lattice, calculation of k_{global} then follows in each direction. To make the process tenable for any case, this is performed by first defining a bounding box around the lattice and then using it to calculate the maximum number of cubic volumes the geometry produces in any one direction (n_v). This also helps to explain λ as it is needed to account for complex shapes that do not fit perfectly in to their bounded domain. A value is derived in each direction by considering the nature of a 2-D plane taken through the bounding box, as each degree of freedom of the system has the potential for acting according to two planes, the average of these values is taken in order to simplify the calculations needed on the GPU. It would otherwise be necessary to determine the direction of motion of a particle to decide which value to use, introducing undesirable parallel divergence. The values of k_{global} are found according to

$$(k_{global})_x = \frac{1}{2} \left(\frac{(n_v)_x k_{initial}}{(n_v)_z} + \frac{(n_v)_x k_{initial}}{(n_v)_y} \right) \quad (14)$$

$$(k_{global})_y = \frac{1}{2} \left(\frac{(n_v)_y k_{initial}}{(n_v)_x} + \frac{(n_v)_y k_{initial}}{(n_v)_z} \right) \quad (15)$$

$$(k_{global})_z = \frac{1}{2} \left(\frac{(n_v)_z k_{initial}}{(n_v)_x} + \frac{(n_v)_z k_{initial}}{(n_v)_y} \right) \quad (16)$$

D. Combining LSM and SPH

With definitions of the LSM and SPH methods in place, a mechanism to allow particles governed by one model to interact with those from the other is needed. One solution is typical when coupling dissimilar methods to SPH, which is to introduce boundary particles that represent the outermost points of the LSM into the SPH simulation, these are then moved according to the LSM dynamics and also act as a way to transfer resultant fluid pressure from the SPH simulation to the LSM. The problem with this approach is finding a way to implement it such that it introduces little computational overhead within the GPU. Also of importance is the effect that the chosen boundary condition will have on the results, the best way to implement a boundary is not yet fully understood within the SPH community and one which can both move and deform provides a significant challenge.

This work therefore offers a different approach by disregarding the need for a specific boundary condition to represent the LSM and instead reusing existing fluid particles to form a dynamically generated implicit boundary. To handle external forces, the LSM method requires only a linear force per particle, which is then transferred through the structure via the interconnecting lattice. This simplicity allows for a scheme in which certain particles are treated as belonging to both simulations and their momentum calculated according to both sets of governing equations. This does not violate the SPH method as it introduces an additional external force rather than alter the underlying method, similarly the same can be said for the LSM portion as the particles within a weakly-compressible

SPH simulation maintain a steady mass, therefore the total mass of the elastic object is preserved as the simulation evolves.

This is achieved on the GPU by defining a hexadecimal masking system in which each SPH particle has an identifying code which when combined, using fast Boolean logic, with the identifying codes (i.e. LSM) of other particles that fall within its smoothing kernel, determines the particles type. If the particle is LSM then it is considered to be SPH-LSM for as long as it remains within an SPH particle's smoothing kernel, after which it becomes LSM again. This process is depicted in Figure 2 where a simplified 2-D case is considered.

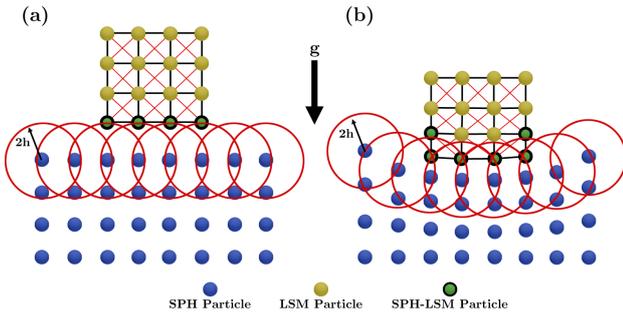


Fig. 2. A 2-D depiction of the SPH-LSM mechanism. The left of the figure (a) shows an elastic box dropping towards a fluid surface due to gravity. The right (b) shows the box interacting with the fluid due to the SPH-LSM particles, which in turn causes the LSM and SPH models to deform and a further two SPH-LSM particles to be defined as they enter the smoothing kernels of SPH particles.

Conservation of momentum is performed for each particle type using

$$\frac{D\vec{v}_a}{Dt}_{SPH} = F_{SPH} \quad (17)$$

$$\frac{D\vec{v}_a}{Dt}_{LSM} = \vec{F}_{LSM} \quad (18)$$

$$\frac{D\vec{v}_a}{Dt}_{SPH-LSM} = F_{SPH} + \vec{F}_{LSM} \quad (19)$$

where F_{SPH} denotes the right-hand side of Equation 1 and \vec{F}_{LSM} the right-hand side of Equation 5. Conservation of mass is performed according to

$$\frac{D\rho_a}{Dt}_{SPH} = \rho_{SPH} \quad (20)$$

$$\frac{D\rho_a}{Dt}_{LSM} = 0 \quad (21)$$

$$\frac{D\rho_a}{Dt}_{SPH-LSM} = \rho_{SPH} \quad (22)$$

where ρ_{SPH} denotes the right-hand side of 3. It is important to note that SPH interactions only occur between SPH and SPH-LSM particles and not between SPH-LSM particles.

As mentioned in Section II, time-step size is determined according to a number of stability criteria. These include those typical to an SPH simulation and one specific to LSM. The size of time-step taken is determined by the minimum value derived from all of these and is found according to $\min(C\min(\Delta t_f, \Delta t_{ev}) \Delta t_{LSM})$ where C is the CFL condition and Δt_f and Δt_{ev} are found according to the force per unit mass and calculated speed of sound plus viscosity scheme

respectively. These are found according to

$$\Delta t_f = \min\left(\sqrt{\frac{h}{|f_a|}}\right) \quad (23)$$

$$\Delta t_{ev} = \min_a\left(\frac{h}{\max(C_s)_b \frac{h\vec{v}_{ab}\vec{r}_{ab}}{r_{ab}^2}}\right) \quad (24)$$

where h is the smoothing length, f_a is the force per unit mass of the SPH particles and C_s is the currently calculated speed of sound. The third criteria Δt_{LSM} is based on the minimum time-step needed to maintain stability for the simple harmonic oscillation of an undamped spring, which can be found using $2\pi\sqrt{\frac{m}{k}}$, where k is the spring constant and m the mass. However, 2π only applies to the case of a simple harmonic system, therefore this needs to change depending upon the complexity of the lattice. In this initial work it is defined as a case specific coefficient c_t , recommended to be no larger than 0.1 with a value closer to 0.05.

As the maximum value for a spring constant for a particle within the system is constantly changing, due to the mechanism described in section III-B, also because it cannot be assumed that the particles associated with a connection will have the same mass, this is calculated per time-step, after locally corrected spring constants have been found, using

$$\Delta t_{LSM} = \min\left(c_t \sqrt{\frac{m_{ab}}{\max((k_{local})_x, (k_{local})_y, (k_{local})_z)}}\right) \quad (25)$$

where m_{ab} is the combined mass of the connected particles and k_{local} the local spring constant for a lattice connection.

IV. BEAM DEFLECTION TESTING

To verify the LSM method's ability to capture linear elastic deformation a 3-D test case with a well defined analytical solution is required. This case measures the deflection of a cantilever beam due to a point force loading on one end for four levels of discretisation and at two strengths ($E = 10$ MPa and $E = 100$ MPa), with a Poisson's ratio of $\nu = 0.3$. The case is depicted in Figure 3.

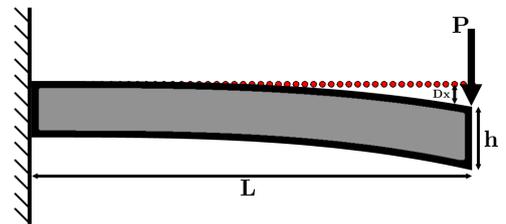


Fig. 3. 3-D beam deflection test case, the problem is shown in 2-D for simplicity as the beam is formed using an equal height and depth.

The beam has a length of 0.5 m and a height and depth of 0.1 m. It is fixed at one end by connecting it to a set of fixed position boundary particles using lattice interconnects and then disabling all SPH interactions. The simulation occurs over 3 s, with the point force P being linearly increased from 0 N to 10 N over the first 2 s and then held at 10 N, the time-stepping coefficient c_t is set at 0.06.

Results for the maximum deflection (Dx) at the free end of the beam are compared against Timoshenko beam theory, found analytically using

$$Dx = \frac{P(L-x)}{\kappa AG} + \frac{PL^3}{3EI} \quad (26)$$

where κ is the Timoshenko shear coefficient and is found according to $\frac{10(1+\nu)}{12+11\nu}$, A is the area of the beam and G the shear modulus. L is the length of the beam and I its second moment of area about the x axis. Damping is set to zero in all cases and the beam is defined as having a density of 1000 kg/m³.

Results can be seen in Table I, where examples using more interconnects show a trend towards the correct result. The analytical solution (Dx_a) is 0.005153 m for $E = 10$ MPa and 0.0005153 m for $E = 100$ MPa, except when particle spacing is at 0.006 m as this produces a particle representation of the beam that is slightly smaller due to rounding error, therefore the analytical solution becomes 0.00473 m for $E = 10$ MPa and 0.000429 m for $E = 100$ MPa. The simulated solution (Dx_s) is recorded in the table and error is calculated using $\frac{Dx_s - Dx_a}{Dx_a}$. It can be noted that solution error reduces with

TABLE I
BEAM DEFLECTION RESULTS COMPARED AGAINST A TIMOSHENKO
DERIVED ANALYTICAL SOLUTION.

Spacing (m)	Interconnects	E (MPa)	Dx_s (m)	Error
0.025	7434	10	0.003211	-0.3769
0.02	13,832	10	0.003621	-0.2973
0.01	100,062	10	0.004475	-0.1316
0.006	459,984	10	0.004564	-0.035
0.025	7434	100	0.0003248	-0.3697
0.02	13,832	100	0.0003616	-0.2983
0.01	100,062	100	0.0004467	-0.1332
0.006	459,984	100	0.0004498	-0.04882

the particle spacing, suggesting a level of convergence, this is not linear due to ϵ , which quickly converges as the number of interconnects increases. The high error at lower spacings is expected due to the simplifications employed to maintain GPU performance and also due to ϵ , which is small at low interconnect densities. It is reasonable to hypothesise that rate of convergence is in fact linked to the number of interconnects in the lattice rather than directly to particle spacing. This was also noted by Zhao et al. who stated that their method gave good results as long as enough connections were used, though they do not quantify this. This assumption means that the level of discretisation needed to retrieve good results from the method will be case specific rather than just discretisation level specific. It is also likely that deriving the shape specific coefficient λ rather than assuming a constant would reduce the error seen at lower levels of discretisation.

V. MINKE WHALE TO TURBINE IMPACT SIMULATION

To demonstrate the practical implementation of the SPH-LSM method the scheme is applied to a realistic case of a minke whale striking a turbine. This work is part of a larger,

ongoing investigation, laboratory experiments are in development and are being designed such that direct comparisons will be able to be drawn with simulations in the future.

In order to capture the dynamics of the minke whale, efforts have been made to model the geometry of the creature as accurately as possible. This has been achieved by creating a 3-D scan of the skeleton of a mid-sized whale using LiDAR technology, a surrounding skin has then been defined according to the skeletal scan and the space in between the two assumed as connective tissue. The various model components can be seen in Figure 4 where the transition from LiDAR point cloud to fully formed 3-D geometry is shown.

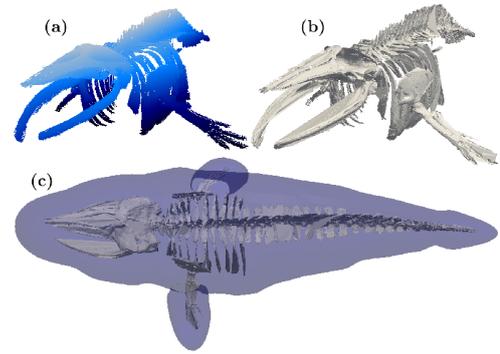


Fig. 4. The creation of a realistic minke whale model, (a) shows the 3-D point cloud captured using LiDAR containing approximately 2.7 million points, (b) shows the resultant 3-D mesh created from the cloud and (c) shows the meshed skeleton surrounded by a derived skin.

This model is used in the formation of an impact case involving a turbine rotating at 1.5 rad/s and treated as a rigid object, the aforementioned whale and surrounding water. The case involves several simplifications which will be removed in later simulations; the fluid domain is a non-periodic box, the turbine rotors are moved according to a fixed angular velocity and do not react to external forces and finally the whale is travelling at approximately 2.5 m/s at the point of impact, a figure chosen arbitrarily for this case. Future simulations will see the whale's kinematics controlled using a forcing function. The three main components of the whale (skeleton, connective tissue and skin) are modelled according to current best estimates of the animals material properties according to their Young's modulus (E) and Poisson's ratio (ν), all salient modelling parameters are given in Figure 5 where the starting point of the scenario is also shown. The time-step coefficient c_t is set to 0.06 and shape coefficient λ estimated using the value for a sphere of $\frac{2}{3}$. A nominal amount of damping is applied to the elastic model with b_{global} values set to 0.05.

The resulting case is comprised of 3,377,206 particles and 1 s of time was simulated. This took approximately 20.37 hours to complete using an NVIDIA GTX680 GPU and involved 153,019 time-steps. This simulation time shows that introducing an elastic element has an impact on computational times and justifies the use of the GPU, snapshots of six periods of time from the simulation can be seen in Figure 6, where the elastic response of the whale behaves as expected and offers

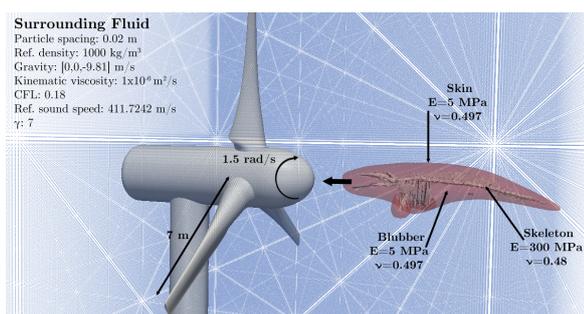


Fig. 5. Initial case for minke whale impact, SPH fluid properties and rigid turbine set up defined. The three distinct areas of the whales makeup are also shown.

reassurance that the method is a suitable approach for future work.

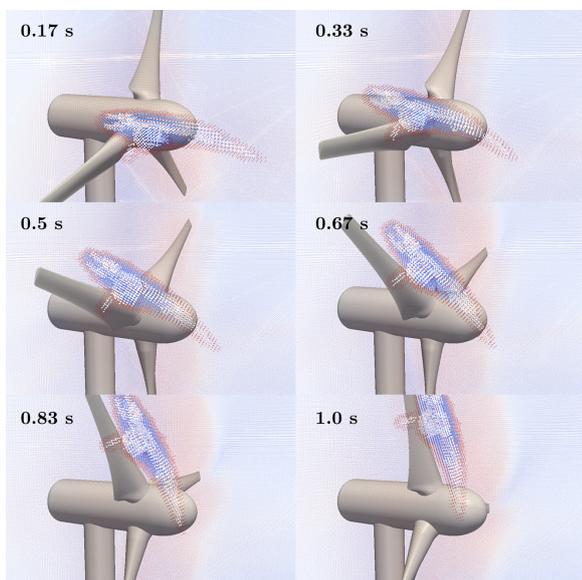


Fig. 6. Six time-steps from the minke turbine impact case. Particles are coloured according to velocity, pressure values are omitted due to the arbitrary nature of the case.

VI. CONCLUSIONS

A method named SPH-LSM has been presented, this provides a way to integrate a particle based elastic simulation technique named lattice spring modelling (LSM) with the weakly-compressible smoothed particle hydrodynamics (SPH) method. SPH-LSM, as presented here, has been designed specifically for calculation using a graphics process unit (GPU) and as such includes a number of modelling simplifications, which are detailed. Testing of the elastic LSM method is provided in the form of a 3-D beam deflection case where results compare favourably against Timoshenko beam theory and show a level of convergence based on particle spacing. Full testing of the SPH-LSM method is provided in the form of a complex scenario involving a multi-material, realistically modelled, minke whale and its impact with a rotating turbine

whilst both are submerged in fluid. This offers a demonstration of the methods suitability for future investigation.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the EPSRC under grant number EP/J010235/1 and also Dr Phillip Manning, Dr William Sellers and Ms. Charlotte Brassey (School of Earth, Atmospheric & Environmental Sciences, The University of Manchester, UK) for capturing the minke whale skeleton using LiDAR as well as Mr Georgios Fourtakas (School of Mechanical, Aerospace & Civil Engineering, The University of Manchester, UK) for developing the minke whale skin geometry.

REFERENCES

- [1] Haug, E., *Engineering safety analysis via destructive numerical experiments*, Engineering Transactions, 29(1) pp. 39-49, 1981.
- [2] LSTC, *LS-DYNA Keyword User's Manual Volume 1: Version R7.0*, Livermore Software Technology Corporation (LSTC), Retrieved 30-04-2014, 2013.
- [3] Mooney, M., *A theory of large elastic deformation*, Journal of Applied Physics, 11(9) pp. 582-592, 1940.
- [4] Rivlin, R.S., *Large elastic deformations of isotropic materials. IV. Further developments of the general theory*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences, 241(835) pp. 379-397, 1948.
- [5] Crespo A.J.C., Dominguez J.M., Barreiro A., Gomez-Gesteira M. and Rogers B.D., *GPUs, a new tool of acceleration in CFD: Efficiency and reliability on Smoothed Particle Hydrodynamics methods*, PLoS ONE, 6(6) ref. e20685, 2011.
- [6] Peskin, C.S., *The immersed boundary method*, Acta Numerica, 11 pp. 479-517, 2002.
- [7] Griffiths, D.V. and Mustoe, G.G.W., *Modelling of elastic continua using a grillage of structural elements based on discrete element concepts*, Int. J. Numer. Meth. Engng., 50 pp. 1759-1775, 2001.
- [8] Zhao, G., Fang, J. and Zhao, J., *A 3D distinct lattice spring model for elasticity and dynamic failure*, Int. J. Numer. Anal. Meth. Geomech., 35(8) pp. 859-885, 2011.
- [9] Timoshenko, S.P., *On the correction factor for shear of the differential equation for transverse vibrations of bars of uniform cross-section*, Philosophical Magazine, pp. 744, 1921.
- [10] Monaghan, J.J., *Smoothed particle hydrodynamics*, Annual Rev. Astron. Appl., 30 pp. 543-574, 1992.
- [11] Lo, E.Y.M. and Shao, S., *Simulation of near-shore solitary wave mechanics by an incompressible SPH method*, Applied Ocean Research, 24 pp. 275-286, 2002.
- [12] Macià, F., Colagrossi, A., Antuono, M. and Souto-Iglesias, A., *Benefits of using a Wendland kernel for free-surface flows*, Proceedings of 6th international SPHERIC workshop, pp. 30-37, Hamburg, Germany, 2011.
- [13] Wendland, H., *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Advances in computational Mathematics, 4(1) pp. 389-396, 1995.
- [14] Monaghan, J.J., *Simulating free surface flows with SPH*, Journal of Computational Physics, 110 pp. 399-406, 1994.
- [15] Monaghan, J.J., *On the problem of penetration in particle methods*, Journal Computational Physics, 82 pp. 1-15, 1989.
- [16] Monaghan, J.J., *Smoothed Particle Hydrodynamics*, Rep. Prog. Physics, 68 pp. 1703-1759, 2005.
- [17] Bentley, J.L., *Multidimensional binary search trees used for associative searching*, Communications of the ACM, 18(9) pp. 509-517, 1975.
- [18] Longshaw, S.M., Rogers, B.D. and Stansby, P.K., *Integration Of Spring Physics With The SPH Method For Quasi-Solid To Fluid Interaction Using GPGPU Programming*, Proc. 8th Int. SPHERIC Workshop, pp. 233-240, 2013.
- [19] Kahan, W., *Further remarks on reducing truncation errors*, Communications of the ACM, 8(1) pp. 40, 1965.
- [20] Puttock, M.J. and Thwaite, E.G., *Elastic compression of spheres and cylinders at point and line contact*, Melbourne: Commonwealth Scientific and Industrial Research Organization, 1969.