A Numerical Scheme for Coupled CFD and CSD Simulation of 3-D Fluid-Membrane Interactions

Yong Gan[1], Zhen Chen[1,2,*,†] and Stephen Montgomery-Smith[3]

[1]Department of Civil & Environmental Engineering, University of Missouri, Columbia, MO 65211, USA
[2]Department of Mechanical & Aerospace Engineering, University of Missouri, Columbia, MO 65211, USA
[3]Department of Mathematics, University of Missouri, Columbia, MO 65211, USA

SUMMARY

For coupled computational fluid dynamics (CFD) and computational solid dynamics (CSD) simulation of the three-dimensional fluid-membrane interaction, a numerical scheme is developed by improving the material point method (MPM). In order to compute the stress state at any membrane point, a plane stress assumption is made in the local tangent plane consisting of membrane points, and a simple procedure is proposed to find the effective point connectivity information for determining the orientation of the local tangent plane. With an iterative algorithm, the existing MPM is improved for formulating fluid points so that fluid dynamics problems with strong shocks could be better simulated. The use of an Eulerian background mesh for solving the momentum equations enables the MPM to automatically handle fluid-membrane interactions without requiring special treatment. Three examples are used to demonstrate the features of the proposed numerical scheme, and to compare them with the analytical and FEM solutions. It appears from the comparison that the proposed procedure is robust and efficient to simulate the three-dimensional fluid-membrane interactions.

KEY WORDS: 3-D fluid-membrane interactions; material point method; meshless; explicit

1. INTRODUCTION

Model-based simulation, which allows detailed parametric studies and complements experiments, is playing an increasingly important role in designing explosion/penetration-resistant structures which are subjected to extreme loading conditions. If the fidelity of model-based simulation is assured, the number of field and/or in-lab experiments could be reduced when verifying and improving new designs. However, existing simulation approaches combine computational fluid dynamics (CFD) with computational solid/structural dynamics (CSD) codes so that the interplay among different spatial and temporal scales would affect the fidelity of simulation results due to the de-coupling effect. In particular, the evolution of failure in the

* Correspondence to: Zhen Chen, 1Department of Civil & Environmental Engineering, University of Missouri, Columbia, MO 65211, USA
† Email: chenzh@missouri.edu

structural system subjected to various loading conditions cannot be predicted objectively with a de-coupled CFD and CSD discretization procedure.

The most challenging task with coupled CFD and CSD simulations is how to treat the interface among different material phases. The past few decades have seen tremendous efforts towards developing numerical methods for fluid-structure interaction problems. One simple simulation approach is to couple the Eulerian fluid and the Lagrangian structure, by imposing appropriate boundary conditions on the interface between Eulerian and Lagrangian meshes. However, the algorithms for identifying the fluid-structure interface and for obtaining convergent boundary conditions can be complicated and computationally costly. The arbitrary Lagrangian Eulerian (ALE) method has also been widely employed for fluid-structure interaction simulation [1-5]. In the ALE approach, the nodes of the computational mesh are allowed to move in an arbitrary way, which can lead to a smooth mesh in highly distorted regions. When compared with the simple Eulerian-Lagrangian coupling approach, the ALE method provides a more convenient fluid-structure interface description. Nevertheless, the mesh-update procedure in the ALE approach may require a tremendous amount of computation time.

Peskin [6] introduced the immersed boundary (IB) method, in the context of simulating the blood flow around heart valves. This method uses the Eulerian description for the fluid, and the Lagrangian description for the structure. Specifically, the immersed structure is modeled as a set of elastic fibers, whereas the fluid flow is governed by the Navier-Stokes equations, which are solved on a fixed Eulerian mesh. The interaction between fluid and structure is achieved by transmitting the fiber stresses and interpolating the nodal velocities through a smooth distribution function [7]. Due to its advantage of automatically handling the fluid-structure interface, the IB method has become a popular tool in the study of various phenomena involving fluid-structure interaction, and has also made a great contribution to the development of other novel numerical approaches for fluid-structure interaction. One such example is the immersed finite element method (IFEM) [8-10].

The IFEM is fundamentally based upon the IB method, along with some features of the extended immersed boundary method [11] and the reproducing kernel particle method (RKPM) [12-14]. In the IFEM, the structure totally immersed in the fluid is described using the Lagrangian formulation, and the Eulerian description is used to define the fluid. An artificial fluid is introduced in the structure domain to generate independent fluid and solid meshes, and fluid and solid equations are solved separately. Within the overlapping domain where the solid and the artificial fluid coexist, the RKPM delta function is employed to couple fluid and solid velocities, and distribute the solid interaction forces onto fluid nodes. However, some assumptions made in the development of the IB method and the IFEM limit their potential in simulating fluid-structure interactions in general. For example, the use of the immersed elastic fiber-like structure in the IB method obstructs the modeling of structures with a nonlinear constitutive relation including failure evolution, and the IFEM may not be able to simulate air-structure interaction problems due to the incompressible fluid assumption.

The material point method (MPM) is an extension to solid dynamics problems of a hydrodynamics code called FLIP which, in turn, evolved from the Particle-in-Cell Method [15-18]. The motivation of the development was to simulate the problems such as impact/contact, penetration, and perforation with history-dependent internal state variables, as shown in the early publications about the MPM [19-20]. The essential idea is to take advantage of both the Eulerian and Lagrangian methods while avoiding the shortcomings of each. In comparison with the other recently developed numerical methods, the MPM appears to be less complex, with a cost factor

of at most twice that associated with the use of corresponding finite elements, and could be easily combined with the finite element codes for large-scale simulations involving different problem domains [21]. In addition, the use of the single-valued mapping functions in the MPM results in a natural no-slip contact/impact scheme so that no inter-penetration would occur. By modifying the original MPM algorithm, York II et al. [22-23] improved the MPM formulation for the interaction between thin membranes and compressible fluids. However, their work only considered two-dimensional cases.

To better simulate the fluid-membrane system response involving failure evolution, an effort is made here to develop the MPM for the three-dimensional fluid-membrane interaction. To determine the movement of membrane points in their local tangent plane, a simple numerical scheme is designed to compute the stress state at any membrane point by adopting the plane stress assumption. An efficient algorithm for determining the orientation of the three-dimensional MPM membrane is developed based on the discretization technique of the finite element method (FEM). In this way, the implementation of the MPM membrane formulation becomes straightforward. In addition, the MPM fluid is formulated to accommodate the cases with strong shocks. The interaction between fluid and membrane points is accomplished by transferring stresses at material points and mapping accelerations at mesh nodes. The remaining sections of this paper are organized as follows. The improved MPM for fluid-membrane interaction in three dimensions is presented in the next section. Three numerical examples are discussed in Section 3 to verify and demonstrate the proposed procedure. Finally, concluding remarks are made in Section 4.

2. THE THREE-DIMENSIONAL MPM FORMULATION FOR FLUID-MEMBRANE INTERACTION

The membrane is one kind of solid structure in that it only possesses the stretching stiffness in the plane tangent to the membrane and has no rigidity in bending. To be computationally robust, the original MPM algorithm is enhanced for the simulations of membranes, fluids and fluid-structure interaction, as described below.

2.1. Computation of membrane stresses

In the MPM, material points in a continuum body are connected via the Eulerian grid nodes. No connection exists between any two points separated by one or more grid cells. It is known that membranes have stresses only in the local tangent plane, while other stress components are negligible. Thus, to model the three-dimensional membrane with the MPM, the algorithm of computing stresses at points in the original MPM must be modified to ensure the movement of membrane material points in their local tangent plane. Otherwise, unrealistic membrane rupture may occur due to the absence of effective connection between neighboring membrane points through grid nodes.

Figure 1 illustrates one three-dimensional membrane in the global x-y-z Cartesian coordinate system. The local Cartesian coordinate system at membrane point p is defined as the x'-y'-z' with the x'-y' plane being the local tangent plane and the z' axis being along the thickness direction. There is one layer of material points through the membrane thickness. Since the membrane has stresses in the tangent plane only, the plane stress assumption is made in the x'-y' plane. Moreover, all other stress components are set to be zero. Unlike the two-dimensional MPM

membrane model, the MPM membrane formulation in three dimensions no longer nullifies strains of $\varepsilon_{y'}$ and $\varepsilon_{x'y'}$. If the membrane is linear elastic, stress components in the local coordinate system can then be simply computed as

$$\varepsilon_{z'} = -\frac{\nu\left(\varepsilon_{x'} + \varepsilon_{y'}\right)}{1-\nu} \tag{1}$$

$$\begin{Bmatrix} \sigma_{x'} \\ \sigma_{y'} \\ \sigma_{z'} \\ \sigma_{x'y'} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 \\ \nu & 1-\nu & \nu & 0 \\ \nu & \nu & 1-\nu & 0 \\ 0 & 0 & 0 & 1-2\nu \end{bmatrix} \begin{Bmatrix} \varepsilon_{x'} \\ \varepsilon_{y'} \\ \varepsilon_{z'} \\ \varepsilon_{x'y'} \end{Bmatrix} \tag{2}$$

where E and ν are the Young's modulus and the Poisson's ratio of the membrane, respectively.

In the MPM, the equations of motion are solved in the x-y-z coordinate system. The strain rate at membrane point p, $\dot{\boldsymbol{\varepsilon}}_p$, is calculated by

$$\dot{\boldsymbol{\varepsilon}}_p = \frac{1}{2}\left[\nabla \boldsymbol{v}_p + \left(\nabla \boldsymbol{v}_p\right)^T\right] \tag{3}$$

where $\boldsymbol{v}_p$ is the velocity vector of point p. At time level k+1 (k=1, 2, 3, …), the total strains at point p are

$$\boldsymbol{\varepsilon}_p^{k+1} = \boldsymbol{\varepsilon}_p^{k} + \dot{\boldsymbol{\varepsilon}}_p^{k+1}\Delta t \tag{4}$$

in which the superscript denotes the time level, and Δt is the time step. To evaluate stresses at membrane points, membrane strains computed in the x-y-z coordinate system must be transformed to the x'-y'-z' coordinate system. According to tensor theory [24], the strains in the local x'-y'-z' coordinate system can be found by the following transformation formulation

$$\boldsymbol{\varepsilon}'_p = \boldsymbol{Q}^T \boldsymbol{\varepsilon}_p^{k+1} \boldsymbol{Q} \tag{5}$$

where $\boldsymbol{\varepsilon}'_p$ is strains at point p in the x'-y'-z' coordinate system, and $\boldsymbol{Q}$ is the direction cosine matrix for the transformation of coordinates in three dimensions and can be expressed as

$$\boldsymbol{Q} = \begin{bmatrix} Q_{xx'} & Q_{xy'} & Q_{xz'} \\ Q_{yx'} & Q_{yy'} & Q_{yz'} \\ Q_{zx'} & Q_{zy'} & Q_{zz'} \end{bmatrix} \tag{6}$$

with Q_{ij} (i=x, y, z and j=x', y', z') being the directional cosine between the global basis vector e_i and the local basis vector e_j. Once the local membrane stresses are computed with the local strains, the plane stress assumption, and the constitutive equation, they should be transformed back to the x-y-z coordinate system for the next MPM computation cycle by

$$\boldsymbol{\sigma}_p^{k+1} = \boldsymbol{Q}\boldsymbol{\sigma}'_p\boldsymbol{Q}^T \tag{7}$$

where $\boldsymbol{\sigma}_p^{k+1}$ and $\boldsymbol{\sigma}'_p$ are the symmetric stress tensor in the global and local coordinate systems at point p, respectively. It should be noticed that the local stresses rather than the local strains are rotated back to the global coordinate system.

As can be seen from the above derivation, the key point of evaluating stresses at membrane points is the determination of $\boldsymbol{Q}$. To find elements of $\boldsymbol{Q}$, it is proposed that the basis vectors of

the local x'-y'-z' coordinate system, $\boldsymbol{v}_{x'}$, $\boldsymbol{v}_{y'}$ and $\boldsymbol{v}_{z'}$, be found by two steps: (1) calculate the vector normal to the membrane surface, $\boldsymbol{v}_{z'}$, and then (2) determine vectors $\boldsymbol{v}_{x'}$ and $\boldsymbol{v}_{y'}$, as discussed below.

In addition to the point connectivity method used by York II in the two-dimensional MPM membrane model [22], many other approaches have been proposed to determine the material point normal, such as simple color function approach, interpolation method, mass matrix approach and point-set method [25-26]. These methods, however, are not effective for complex membrane shapes, and need a large number of material points to obtain a satisfactory point normal. Moreover, the numerical implementation of these approaches is much more complicated than that of the point connectivity method. In fact, the connectivity method is quite simple and convenient except for the disadvantage of additional storage space for the point connectivity data, which is not a problem with current computer hardware.

The original algorithm to set material points is cell-based. Material points are regularly distributed in grid cells, and each point is assigned a fraction of the mass of the associated cell. Since membrane points have no ordered relationship with grid cells, the initialization of membrane points is performed in a different way to construct the effective connectivity information of membrane points.

The membrane surface is first approximated by a collection of triangles, and then the membrane material points are defined on the vertices. Let s be the surface area of the membrane, ρ^m denote the mass per unit area of the membrane, and N^P represent the total number of vertices. Then, the mass of each material point is simply set to be $s\rho^m / N^P$. Insufficient membrane points may result in unrealistic membrane rupture due to the separation of membrane points by mesh cells. On the other hand, more membrane points require more computation time and hardware. Thus, the triangulation of the membrane surface should be performed based on the available hardware and the characteristics of the problems.

Because of its initialization on the triangle vertex, each membrane material point will be shared by several triangles. The normal vector of each triangle can be easily found with the coordinates of its three vertices. As illustrated in Figure 2, the point normal at point p is simply taken as the average of vectors normal to the triangles to which membrane point p belongs, i.e.,

$$\boldsymbol{v}_{z'} = \sum_{i=1}^{N_{Tri}} \boldsymbol{n}_i \Big/ N_{Tri} \tag{8}$$

where $\boldsymbol{n}_i$ is the unit vector normal to triangle i, and N_{Tri} is the total number of triangles surrounding point p. Conventionally, the normal to a triangle is calculated by the right-hand rule and the outward-pointing normal is used for closed surfaces.

Now, let us consider how to determine $\boldsymbol{v}_{x'}$ and $\boldsymbol{v}_{y'}$. In three dimensions, the relation between the values of $\boldsymbol{v}_{z'}$ at time levels k+1 and k can be expressed as

$$\boldsymbol{v}_{z'}^{k+1} = \boldsymbol{S} \cdot \boldsymbol{v}_{z'}^{k} \tag{9}$$

where $\boldsymbol{S}$ is the 3×3 orthogonal matrix, that is, $\boldsymbol{S}^{-1} = \boldsymbol{S}^{T}$, which maps $\boldsymbol{v}_{z'}^{k}$ to $\boldsymbol{v}_{z'}^{k+1}$, but preserves all vectors perpendicular to both $\boldsymbol{v}_{z'}^{k}$ and $\boldsymbol{v}_{z'}^{k+1}$. Due to the use of the Cartesian coordinate system, the vectors of $\boldsymbol{v}_{x'}^{k+1}$ and $\boldsymbol{v}_{y'}^{k+1}$ can also be written as

$$\boldsymbol{v}_{x'}^{k+1} = \boldsymbol{S} \cdot \boldsymbol{v}_{x'}^{k} \tag{10}$$

$$\boldsymbol{v}_{y'}^{k+1} = \boldsymbol{S} \cdot \boldsymbol{v}_{y'}^{k} \tag{11}$$

Therefore we are reduced to the problem of finding $\boldsymbol{S}$. Let $\overline{\boldsymbol{w}}$ denote the cross product of $\boldsymbol{v}_{z'}^{k}$ to $\boldsymbol{v}_{z'}^{k+1}$, and $\hat{\boldsymbol{w}}$ be the unit vector in the same direction, i,e,

$$\overline{\boldsymbol{w}} = \boldsymbol{v}_{z'}^{k} \times \boldsymbol{v}_{z'}^{k+1} \tag{12}$$

$$\hat{\boldsymbol{w}} = \overline{\boldsymbol{w}} / \left|\overline{\boldsymbol{w}}\right| \tag{13}$$

Then

$$\boldsymbol{P}_1 = \hat{\boldsymbol{w}} \otimes \hat{\boldsymbol{w}} \tag{14}$$

$$\boldsymbol{P}_2 = \boldsymbol{I}_3 - \hat{\boldsymbol{w}} \otimes \hat{\boldsymbol{w}} \tag{15}$$

are the orthogonal projections that map onto subspace perpendicular to $\boldsymbol{v}_{z'}^{k}$ and $\boldsymbol{v}_{z'}^{k+1}$, and the subspace spanned by $\boldsymbol{v}_{z'}^{k}$ and $\boldsymbol{v}_{z'}^{k+1}$, respectively. Here, $\boldsymbol{I}_3$ denotes the 3 × 3 identity matrix. Then, $\boldsymbol{S}$ will be the identity on the range of $\boldsymbol{P}_1$, and a two dimensional rotation on the range of $\boldsymbol{P}_2$. Using Gram-Schmidt orthogonalization [27], it may be seen that the range of $\boldsymbol{P}_2$ has orthonormal basis consisting of the two vectors $\hat{\boldsymbol{b}}_1$ and $\hat{\boldsymbol{b}}_2$, i.e.,

$$\hat{\boldsymbol{b}}_1 = \boldsymbol{v}_{z'}^{k} \tag{16}$$

$$\hat{\boldsymbol{b}}_2 = \boldsymbol{b}_2 / \left|\boldsymbol{b}_2\right| \tag{17}$$

$$\boldsymbol{b}_2 = v_{z'}^{k+1} - \left(\hat{\boldsymbol{b}}_1 \cdot v_{z'}^{k+1}\right)\hat{\boldsymbol{b}}_1 \tag{18}$$

and with respect to this basis, $\boldsymbol{S}$ performs the following two dimensional rotation

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{19}$$

where $\cos\theta = \boldsymbol{v}_{z'}^{k} \cdot \boldsymbol{v}_{z'}^{k+1}$, and $\sin\theta = \sqrt{1-\cos^2\theta}$. Here, $\sin\theta$ can always be taken to be non-negative, so the square root calculation is unambiguous. Therefore

$$\boldsymbol{R} = \left(\cos\theta\hat{\boldsymbol{b}}_1 - \sin\theta\hat{\boldsymbol{b}}_2\right)\otimes\hat{\boldsymbol{b}}_1 + \left(\sin\theta\hat{\boldsymbol{b}}_1 + \cos\theta\hat{\boldsymbol{b}}_2\right)\otimes\hat{\boldsymbol{b}}_2 \tag{20}$$

Now

$$\boldsymbol{S} = \boldsymbol{R}\boldsymbol{P}_2 + \boldsymbol{P}_1 \tag{21}$$

which after simplification becomes

$$\boldsymbol{S} = \left(\hat{\boldsymbol{w}} \otimes \hat{\boldsymbol{w}}\right) + \left(\boldsymbol{v}_{z'}^{k} \cdot \boldsymbol{v}_{z'}^{k+1}\right)\left(\boldsymbol{I}_3 - \hat{\boldsymbol{w}} \otimes \hat{\boldsymbol{w}}\right) + \left(\boldsymbol{v}_{z'}^{k+1} \otimes \boldsymbol{v}_{z'}^{k} - \boldsymbol{v}_{z'}^{k} \otimes \boldsymbol{v}_{z'}^{k+1}\right) \tag{22}$$

The only way a singularity can happen in this calculation is when $\overline{\boldsymbol{w}} = 0$, in which case we have $\boldsymbol{S} = \boldsymbol{I}_3$. After the orientation of the local coordinate axes has been updated, the elements of matrix $\boldsymbol{Q}$ are calculated based on their definition.

Based on the procedure described above, the three-dimensional MPM membrane model can be easily implemented by modifying the existing three-dimensional MPM code. If the material point is a membrane point, its total strains in the global coordinate system are rotated to the local coordinate system at each time step, and then the plane stress assumption is applied. With an appropriate constitutive model, the local stresses at membrane points are computed and transformed back to the global coordinate system for the evaluation of the internal forces at mesh nodes.

2.2. Fluid constitutive equation

As demonstrated by Sulsky et al. [20], the development of the MPM discrete momentum equations does not invoke any constitutive equations at all. Hence, the discretization procedure and the numerical scheme of the standard MPM hold for solid points as well as fluid points. The key difference between fluid and solid material points is the various constitutive relations they respectively follow. For fluid points, the relation between the stresses and rate of strains is given by

$$\boldsymbol{\sigma}_f = 2\mu\dot{\boldsymbol{\varepsilon}}_f + \lambda \operatorname{tr}\left(\dot{\boldsymbol{\varepsilon}}_f\right)\underline{\underline{\boldsymbol{I}}} - P\underline{\underline{\boldsymbol{I}}} \tag{23}$$

where the *f* subscript denotes the fluid point, λ is bulk viscosity of the fluid, μ is shear viscosity of the fluid, *P* is pressure of fluid points and $\underline{\underline{\boldsymbol{I}}}$ is the second-order unit tensor.

The strain rate of fluid material points can be obtained by Equation (3). To find stresses at fluid points with Equation (23), an equation of state (EOS) is required for fluid pressure *P*, namely,

$$I_f = I\left(P, \rho_f\right) \tag{24}$$

where I_f and ρ_f are the specific internal energy and the density of fluid points, respectively. Based on the mass conservation, the density of material points at time level *k*+1 can be updated by the following equation

$$\rho_f^{k+1} = \frac{\rho_f^k}{1 + \Delta t\left(\nabla \cdot \boldsymbol{v}_f^{k+1}\right)} \tag{25}$$

where $\boldsymbol{v}_f$ is the velocity vector of fluid points. The EOS is dependent on the internal energy as well as density. The energy equation is considered at each material point. If the thermal effect is negligible, the conservation of energy implies that the change of internal energy is equal to the rate of mechanical work done by stresses on the system. Thus, the internal energy of fluid points is updated by

$$I_f^{k+1} = I_f^k + \frac{\Delta t}{\rho_f^{k+1}} \boldsymbol{\sigma}_f^{k+1} : \dot{\boldsymbol{\varepsilon}}_f^{k+1} \tag{26}$$

It can be found from Equations (24) and (26) that Equation (23) is nonlinear because both sides contain the stress term. Thus, an iterative procedure must be employed to obtain convergent internal energy and pressure. At time level *k*+1, the iteration steps are described as follows:

(1) Calculate ρ_f^{k+1} by Equation (25)

(2) Initially set the internal energy of fluid points as

$$I_f^{k+1,m} = I_f^k + \frac{\Delta t}{\rho_f^{k+1}} \boldsymbol{\sigma}_f^k : \dot{\boldsymbol{\varepsilon}}_f^{k+1} \tag{27}$$

where *m* (*m*=1, 2, …) denotes the *m*th iteration substep.

(3) Solve the equation of state for the pressure of fluid points

$$I_f^{k+1,m} = I\left(P^{k+1,m}, \rho_f^{k+1}\right) \tag{28}$$

(4) Compute the stress tensor of fluid points

$$\boldsymbol{\sigma}_f^{k+1,m} = 2\mu\dot{\boldsymbol{\varepsilon}}_f^{k+1} + \lambda \operatorname{tr}\left(\dot{\boldsymbol{\varepsilon}}_f^{k+1}\right)\underline{\underline{\boldsymbol{I}}} - P^{k+1,m}\underline{\underline{\boldsymbol{I}}} \tag{29}$$

(5) Update the internal energy of fluid points

$$\begin{cases} I_f^{k+1,m} = I_f^k + \dfrac{\Delta t}{\rho_f^{k+1}} \boldsymbol{\sigma}_f^{k+1,m} : \dot{\boldsymbol{\varepsilon}}_f^{k+1} & m = 1 \\ I_f^{k+1,m} = I_f^k + \dfrac{\Delta t}{\rho_f^{k+1}} \left(\dfrac{\boldsymbol{\sigma}_f^{k+1,m-1} + \boldsymbol{\sigma}_f^{k+1,m}}{2} \right) : \dot{\boldsymbol{\varepsilon}}_f^{k+1} & m > 1 \end{cases} \tag{30}$$

(6) Repeat steps (3)-(5) until the internal energy and the stresses converge, and then update the internal energy and the stresses at fluid points using

$$\begin{cases} I_f^{k+1} = I_f^{k+1,m}, \boldsymbol{\sigma}_f^{k+1} = \boldsymbol{\sigma}_f^{k+1,m} & m = 1 \\ I_f^{k+1} = \dfrac{I_f^{k+1,m-1} + I_f^{k+1,m}}{2}, \boldsymbol{\sigma}_f^{k+1} = \dfrac{\boldsymbol{\sigma}_f^{k+1,m-1} + \boldsymbol{\sigma}_f^{k+1,m}}{2} & m > 1 \end{cases} \tag{31}$$

The use of artificial viscosity in fluid dynamics simulations has proven itself to be able to smooth the oscillation at the shock front and give more accurate results. The artificial viscosity employed in the MPM, q, is added to the pressure of fluid points and expressed as

$$\begin{cases} q = \rho L_c \left\{ c_0 L_c \left[\mathrm{tr}\left(\dot{\boldsymbol{\varepsilon}}_f \right) \right]^2 - c_1 a \, \mathrm{tr}\left(\dot{\boldsymbol{\varepsilon}}_f \right) \right\} & \mathrm{tr}\left(\dot{\boldsymbol{\varepsilon}}_f \right) < 0 \\ q = 0 & \mathrm{tr}\left(\dot{\boldsymbol{\varepsilon}}_f \right) \geq 0 \end{cases} \tag{32}$$

where a is the local sound speed of the fluid point, c_0 and c_1 are constants, and L_c is the characteristic length. In the three-dimensional MPM, L_c is calculated as

$$L_c = \sqrt[3]{V_{\mathrm{cell}}} \tag{33}$$

where V_{cell} is the volume of the grid cell. Usually, the grid cell in the three-dimensional MPM is cubic. Thus, the characteristic length is identical to the side length of the background mesh cell. The artificial viscosity given in Equation (32) is also used in LS-DYNA [28], in which c_0 and c_1 default to 1.5 and 0.6, respectively. In general, the values of c_0 and c_1 should be determined through numerical tests.

2.3. Fluid-Membrane interaction in the MPM

In the MPM, material points provide the Lagrangian description of the fluid and the membrane, and the equations of motion are solved on the nodes of the computational mesh instead of the material points. The stresses at membrane and fluid material points are transformed into nodal forces, and the accelerations are imposed on the material points through interpolation of the nodal accelerations. Therefore, the pressure from the fluid points is not directly applied to the membrane points. Instead, the interaction between the fluid and the membrane is indirectly coupled via the Eulerian grid nodes without any consideration of the fluid-membrane interface. As a result, there is no need for the MPM to identify the fluid-membrane interface and apply correct boundary conditions, in contrast to what mesh-based methods generally do. In other words, the original MPM is able to automatically handle the fluid-membrane interaction without requiring special treatment.

3. NUMERICAL EXAMPLES

3.1. One-dimensional shock tube

The simulation of one dimensional shock propagation in the fluid is an ideal example to test the MPM fluid model, because analytical solutions are available for this problem. As shown in Figure 3, a shock tube is divided into two halves by a diaphragm. Initially, the left region is full of perfect gas with high pressure p_L and density ρ_L, and the right region contains ideal gas with low density ρ_R and pressure p_R. The diaphragm is suddenly broken at time t=0, and then the shock wave due to the pressure discontinuity propagates to the right.

The length of the tube, l, is 1 m and the initial velocities for the air in both regions are zero. Other initial conditions are ρ_L= 1 kg/m^3, p_L=1 Pa, ρ_R= 0.125 kg/m^3, and p_R=0.001 Pa. The ideal gas EOS is applied to the gases in both regions. This one-dimensional problem is solved with the three-dimensional MPM. The x-axis is chosen to be the wave propagation direction. The nodal velocities along the other two directions, i.e., the y- and z-directions, are nullified. The background mesh is composed of 800 cubic cells with a side length of 0.00125 m. The initialization of material points in each cell is demonstrated in Figure 4. Thus, each cell has 25 points and there are 20000 points in total. The time step is 2.0×10^{-5} s, and the artificial viscosity defined in Equation (32) is applied with c_0=2.0 and c_1=1.0.

To verify the proposed iterative algorithm, the non-iterative MPM algorithm for fluids presented by York II et al. [23] is also used to simulate this example. Figure 5 gives the profiles of pressure, density, velocity and internal energy along the wave propagation direction at time t=0.143 s using the non-iterative algorithm, and the corresponding profiles obtained by the presented iterative algorithm are illustrated in Figure 6. Figures 5 and 6 are both plotted according to the averages of material points initialized in every four consecutive mesh cells. In Figure 5, poor agreement is observed between the MPM and analytical solutions. In particular, the density at the shock front in the MPM solution is up to 47 kg/m^3, whereas the corresponding analytical solution is 0.73 kg/m^3. This demonstrates that the non-iterative algorithm is not able to give satisfactory solutions for problems with high shock strength. From Figure 6, it can be seen that the results for the iterative algorithm favorably agree with the analytical solutions.

3.2. Impact between a membrane and an elastic solid

An example of a cuboid impacting a net is used to validate the MPM formulation for membranes. As shown in Figure 7, a cuboid solid is initially positioned above the center of a stationary net. At time t=0, the cuboid moves toward the net with a velocity of 1 m/s along the z-direction. Both the cuboid and the net are elastic, and their dimensions and material properties are listed in Tables I and II, respectively.

The MPM model is composed of 53500 material points with 40000 for the net and 13500 for the cuboid. The net is triangulated with 77922 triangles. The computational mesh is built with cubic cells, and three cell sizes are employed, namely, 0.05 m, 0.025 m and 0.02 m. All simulations are performed with a time step of 1×10^{-5} s. Figure 8 presents the time history of the z-directional displacement at the central point of the cuboid by the MPM and the LS-DYNA. It can be observed from the figure that the solutions by the MPM and the LS-DYNA agree well, and the MPM solutions are convergent as the mesh size becomes smaller. The deformations of the net at various times are shown in Figure 9 (0.02 m mesh size).

3.3. Membrane expansion

This example demonstrates the MPM formulation for fluid-membrane interaction. As shown in Figure 10, a gas-filled box is covered by a flat membrane, and other walls of the box are rigid. One elastic cube is put at the center of the membrane. The membrane perimeter is fixed, and the gas in the box has an initial pressure of 800 Pa. Due to the pressure difference between the inside and outside of the box, the gas will expand and the membrane will be displaced. Eventually, the cube will be released from the membrane. During the time period of this simulation, no release is considered, and the cube always has close contact with the membrane.

The box is 0.2 m in length, 0.2 m in width, and 0.1 m in height. Hence, the 0.01m-thick square membrane has a side length of 0.2 m. The cube has a side length of 0.04 m. The membrane is triangulated with 19602 triangles, and represented by 10000 material points. The material points for the cube and the gas are initialized by the standard cell-based algorithm [25], with 13824 solid points for the cube, and 48000 fluid points for the gas. Linear elasticity is used for the membrane and cube with elastic parameters given in Table III. The ideal-gas EOS is adopted for the gas, and artificial viscosity is used with c_0=2.0 and c_1=1.0. The computational grid is constructed of cubic elements, with the side length being 0.01m, and a time step of 1×10^{-5} s is employed. The *z*-directional displacement at the central point of the cube is given in Figure 11. The good match between the MPM and LS-DYNA solutions demonstrates that the MPM can solve fluid-membrane interaction problems without using additional algorithms. The deformed shapes of the membrane at various times are shown in Figure 12. Due to the inertia of the cube, the membrane points interacting with the cube have smaller velocities than other membrane points. Therefore, a concavity at the center of the membrane is observed in the figure.

4. CONCLUDING REMARKS

The improved MPM algorithms for three-dimensional simulation of membranes, fluids and fluid-membrane interaction have been presented in this paper. To implement the MPM for membranes, the plane stress assumption is made in the local tangent plane of membrane points so that stresses at membrane points are consistent with the membrane orientation. Based on the mesh-generation technique from the FEM, a simple and effective method of initializing membrane points is developed to build the point connectivity used to determine the local normal-tangential coordinate for membrane points. By using an iterative algorithm, the MPM fluid model has been improved to enable the simulation of strong-shock fluid dynamics problems. The interaction between the fluid and the membrane is indirectly coupled via the Eulerian grid nodes without any consideration of the fluid-membrane interface. No modification to the original MPM algorithm is needed for the treatment of fluid-membrane interaction.

Three examples of the shock tube test, a cuboid impacting a membrane, and the membrane expansion are solved with the proposed MPM formulations. The reasonable agreement between the MPM results and the analytical and FEM solutions demonstrates the efficiency and robustness of the presented MPM algorithm for fluid-membrane interaction problems. With the basic MPM code, the proposed MPM model for fluid-membrane interaction can be easily implemented by simply adding subroutines for the evaluation of stresses at membrane and fluid points. Due to the inherent features of the MPM, the proposed method should be well suited to the analysis of fluid-membrane interaction problems involving failure evolution, which is the focus of the ongoing work.

ACKNOWLEDGEMENTS

The authors are grateful to the partial support from the NSF and to the University of Missouri Bioinformatics Consortium for super-computing time.

REFERENCES

1. Liu WK. *Development of finite-element procedures for fluid-structure interaction*. Ph.D. thesis, California Institute of Technology, 1981.
2. Liu WK, Ma DC. Computer implementation aspects for fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering* 1982; **31**: 129-148.
3. Huerta A, Liu WK. Viscous flow with large free surface motion. *Computer Methods in Applied Mechanics and Engineering* 1988; **69**: 277-324.
4. Hu HH, Patankar NA, Zhu MY. Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian–Eulerian technique. *Journal of Computational Physics* 2001; **169**: 427-462.
5. Zhang LT, Wagner G, Liu WK. Modeling and simulation of fluid structure interaction by meshfree and FEM. *Communications in Numerical Methods in Engineering* 2003; **19**: 615-621.
6. Peskin CS. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics* 1972; **10**: 220-252.
7. Mittal R, Iaccarino G. Immersed boundary method. *Annual Review of Fluid Mechanics* 2005; **37**: 239-261.
8. Zhang L, Gerstenberger A, Wang X, Liu WK. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**: 2051-2067.
9. Liu WK, Liu Y, Farrell D, Zhang L, Wang SX, Fukui Y, Patankar N, Zhang Y, Bajaj C, Lee J, Hong J, Chen X, Hsu H. Immersed finite element method and its applications to biological systems". *Computer Methods in Applied Mechanics and Engineering* 2006; **195**: 1722-1749.
10. Zhang L, Gay M. Immersed finite element method for fluid-structure interactions. *Journal of Fluids and Structures* 2007; **23**: 839-857.
11. Wang X, Liu WK. Extended immersed boundary method using FEM and RKPM. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**: 1305-1321.
12. Liu WK, Jun S, Zhang YF. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**: 1081-1106.
13. Liu WK, Chen YJ. Wavelet and multiple scale reproducing kernel method. *International Journal for Numerical Methods in Fluids* 1995; **21**: 901-932.
14. Zhang LT, Wagner GJ, Liu WK. A parallized meshfree method with boundary enrichment for Large-Scale CFD. *Journal of Computational Physics* 2002; **176**: 483-506.
15. Brackbill JU. On modeling angular momentum and vorticity in compressible fluid flow. Computer Physics Communications 1987; **47**: 1-16.
16. Brackbill JU. The ringing instability in particle-in-cell calculations of low speed flow. *Journal of Computational Physics* 1988; **75**: 469-492.
17. Brackbill JU. FLIP-MHD: A particle-in-cell method for magneto-hydrodynamics. *Journal of Computational Physics* 1991; **96**: 163-192.
18. Brackbill JU, Ruppel HM. FLIP: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics* 1986; **65**: 314-343.
19. Sulsky D, Chen Z, Schreyer HL A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1994; **118**: 179-196.
20. Sulsky D, Zhou SJ, Schreyer HL. Application of a particle-in-cell method to solid mechanics. *Computer Physics Communications* 1995; **87**: 236-252.

21. Chen Z, Hu W, Shen L, Xin X, Brannon R. An Evaluation of the MPM for Simulating Dynamic Failure with Damage Diffusion. *Engineering Fracture Mechanics* 2002; **69**: 1873-1890.
22. York II AR, Sulsky D, Schreyer H. The Material Point Method for Simulation of Thin Membranes. *International Journal for Numerical Methods in Engineering* 1999; **44**: 1429-1456.
23. York II AR, Sulsky D, Schreyer H. Fluid-Membrane Interaction Based on the Material Point Method. *International Journal for Numerical Methods in Engineering* 2000; **48**: 901-924.
24. Lai WN, Rubin D, Krempl E. *Introduction to Continuum Mechanics (Third Edition)*. Butterworth-Heinemann: Burlington, 1993.
25. York II AR. Development of modifications to the material point method for the simulation of thin membranes, compressible fluids, and their interactions. *Technical Report SAND97-1983*, Sandia National Laboratories, 1997.
26. Torres DJ, Brackbill JU. The point-set method: Front-Tracking without connectivity. *Journal of Computational Physics* 2000; **165**: 620-644.
27. Golub GH, Van Loan CF. *Matrix Computations (Third Edition)*. The Johns Hopkins University Press: Baltimore, 1996.
28. Hallquist JO. *LS-DYNA Theoretical Manual*. Livermore Software Technology Corporation: Livermore, California, 1998.

List of Tables

Table I Dimensions of the cuboid and the net

Table II Material properties of the cuboid and the net

Table III Material properties of the cube and the membrane

Table I Dimensions of the cuboid and the net

Continuum body	Length (m)	Width (m)	Height or Thickness (m)
Cuboid	0.2	0.2	0.1
Net	1	0.2	0.0125

Table II Material properties of the cuboid and the net

Continuum Body	Young's Modulus (Pa)	Poisson's ratio	Density (kg/m^3)
Cuboid	2×10^7	0.2	4000
Net	2×10^7	0.0	2000

Table III Material properties of the cube and the membrane

Continuum Body	Young's Modulus (Pa)	Poisson's ratio	Density (kg/m^3)
Cube	1×10^7	0.2	1000
Membrane	1×10^5	0.45	1000

List of Illustrations

Figure 1 One three-dimensional membrane and the definition of the local normal-tangential coordinate system

Figure 2 Calculation of the point normal

Figure 3 One-dimensional shock tube problem

Figure 4 Initialization of points in one cell

Figure 5 MPM solutions by the non-iterative algorithm

Figure 6 MPM solutions by the iterative algorithm

Figure 7 An elastic cuboid impacting a net

Figure 8 Time history of the *z*-directional displacement at the central point of the cuboid

Figure 9 Material point position plots for the simulation of a cuboid impacting a net: (a) t=0.05 s, (b) t=0.1 s, (c) t=0.15 s, (d) t=0.2 s (0.02 m mesh size)

Figure 10 Setup of the fluid-membrane interaction problem

Figure 11 Time-history of the *z*-directional displacement at the central point of the cube

Figure 12 Deformed shapes of the membrane at various times: (a) t=0.005 s, (b) t=0.001 s, (c) t=0.015 s, (d) t=0.02 s, (e) t=0.025 s, (f) t=0.03 s