ADAPTIVE MESHING IN THE PARTICLE FINITE ELEMENT METHOD

Thomas Leyssens[1] Jonathan Lambrechts[1] Jean-François Remacle[1]

[1] *Université Catholique de Louvain, Louvain-la-neuve, Belgium.*
thomas.leyssens@uclouvain.be

ABSTRACT

Highly deforming domains are a recurring problem in fluid mechanics. In free surface flows, for instance, boundaries are in constant motion. The Particle Finite Element Method, or PFEM, tackles this problem by considering the mesh to be an inherent part of the solution. This Lagrange-based strategy solves the Navier-Stokes equations on a finite element mesh that is constantly adapted to exactly represent the fluid domain. Due to the strong dependence on mesh quality in PFEM, many solvers suffer from poorly shaped elements within the domain. In this work, we propose an approach to adapt the mesh with theoretical guarantees of quality. The approach is based on Delaunay refinement strategies, allowing to efficiently adapt the mesh while maintaining high quality elements. This addresses well-known problems in PFEM of volume variation errors due to undesired element removal and addition. We first present PFEM in general, followed by the mesh adaptation algorithm, and finally some simulation results.

Keywords: finite elements, mesh adaptation, fluid mechanics, free surface flows, PFEM

1. INTRODUCTION

In many computational engineering problems, the domain of interest undergoes strong deformations. This may be due to the relative motion of different parts of the domain or the domain boundaries themselves evolving through time. In such situations, having a suitable fixed mesh can be a challenge. Consider, for example, a seemingly simple fluid mechanics problem of a container partially filled with water. When this container is shaken, the dynamics of the water become relatively complex: interactions with the container walls, mixing of the water with the surrounding air, droplets splashing,... To accurately and efficiently model this, classical Eulerian approaches can become quite expensive.

In such situations, Lagrangian approaches are better suited since the geometry variations are inherent to the model. In most Lagrangian methods, the fluid domain is represented by a set of particles that follow the motion of the fluid. These particles contain all the information: velocity, pressure, temperature, etc. The Particle Finite Element Method, or PFEM [1], is such Lagrangian mesh-based method. This method circumvents issues of mesh distortions just like other particle methods, and relies on an accurate resolution of the conservation equations through the finite element method. The flexibility and accuracy offered by this method make it a good choice for situations where the domain deforms severely.

When PFEM is applied to fluids, the Navier-Stokes equations are solved on a finite element mesh. This mesh is obtained by triangulation of the particles. A velocity field is obtained throughout the domain, which is then used to displace the particles.

The quality of the mesh plays a crucial role in PFEM for the overall accuracy of the solution. Indeed, because the domain can deform, its geometry becomes an unknown of the problem. The mesh therefore needs to be highly accurate in representing this domain. Undesirable effects of volume creation and destruction arise if no attention is paid to the quality of the mesh.

Figure 1: A snapshot of a fluid simulation in PFEM. The geometry is solution dependent.

In this work, we present an approach to triangulate the particles in PFEM using a robust algorithm that constrains the overall quality of the mesh. Indeed, while most PFEM solvers already adapt the mesh through particles insertion and removal, the current techniques are not based on mesh quality criteria [2]. Our algorithm efficiently guarantees the updated mesh respects a given element quality and size throughout the domain.

The approach is based on the Delaunay refinement algorithm proposed by Chew [3]. The philosophy consists of enforcing the mesh to be Delaunay at all times, and refining it based on a user-defined size field. This greatly improves the quality of the method, as it removes the arbitrariness in the detection of the domain. Moreover, the use of a size field is very useful since it allows greater accuracy in regions of interest. Using this method, obtaining accurate and smooth simulations of free surface flows is possible with remarkably few elements.

In the following sections of this research note, the generalities of the particle finite element method are briefly presented first. The central part of this work, the mesh adaptation algorithm, is presented in section 3. Finally, some simulation results are shown, along with future prospects.

2. THE PARTICLE FINITE ELEMENT METHOD

The fluid equations are first briefly described, followed by the development of the PFEM algorithm and a short note on boundary conditions.

2.1 Fluid equations

Although PFEM has been extended to other fields of study[2], we focus here on a fluid domain. The equations of motion are the Navier-Stokes equations, and we consider an incompressible fluid:

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

$\mathbf{u}$ represents the velocity of the fluid, ρ its density, p the pressure, μ the dynamic viscosity, and $\mathbf{g}$ is gravity. A significant advantage of using a Lagrangian approach for fluids is the absence of the convective term in the momentum conservation equation (1), thereby leaving aside the non-linearities in the equation. The boundary conditions are slip-wall conditions along solid walls, and free surfaces elsewhere. Equations (1) and (2) are then solved using a classical first-order finite element formulation. Finally, to address the well-known Ladyzhenskaya-Babuska-Brezzi (LBB) condition, a Pressure-Stabilizing Petrov-Galerkin (PSPG) approach is used to stabilize the method[4]. The general PFEM algorithm is described in the next section.

2.2 PFEM steps

Recall that PFEM is a particle method. Hence, the fluid is initially represented by a set of points, or particles. In these particles, the velocity field of the fluid is stored. Note that no mass is attached to the particles. Additionally, a geometry of the solid walls is necessary to describe the outer boundaries.

At the beginning of each time step, these particles are triangulated to obtain a mesh. At this point, the shape of the fluid is not yet known, since there is no unique definition of the shape of a set of points. Therefore, it is necessary to use an "oracle" function that answers whether or not a point $\mathbf{x}$ lies inside the fluid domain. The oracle function commonly used in PFEM is the α-shape of a set of points[5]. In short, the α-shape uses a quality and size measure based on the circumradius of the elements in the triangulation to determine whether or not it belongs to the fluid. Too large or too poorly shaped triangles will not belong to the fluid, as depicted in figure 2. Hence, the quality of the elements in the mesh plays a key role in determining the shape of the domain.

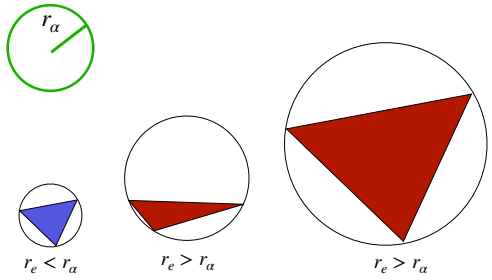

Figure 2: The α-shape of a set of points uses a size and a quality measure to determine whether or not an element belongs to the domain.

Once the fluid domain has been clearly defined, the boundaries can also be characterized. Along solid

walls, a free slip condition is imposed. If the boundary is not along a wall, then it is necessarily a free surface. A few additional comments on the application of the boundary conditions are made in section 2.3.

After detection of the boundaries, the conservation equations (1) and (2) can be solved on the fluid domain. The solution vector $\{\mathbf{U}_i^t, p_i^t\}$ is obtained for each particle. The positions of the particles can then be updated through the velocity vector:

$$\mathbf{X}_i^{t+1} = \mathbf{X}_i^t + \mathbf{U}_i^t \Delta t, \qquad i = 1, ..., n.$$

To avoid mesh distortion, the particles are triangulated again at the next time step. However, this yields the possibility for the α-shape algorithm to remove elements that are, in fact, part of the fluid domain. If nothing is done, volume variations will arise, as presented in [6]. This is the essential motivation for developing the mesh adaptation algorithm presented in section 3. The main steps of the particle finite element method, without mesh adaptation, are summarized in figure 3.

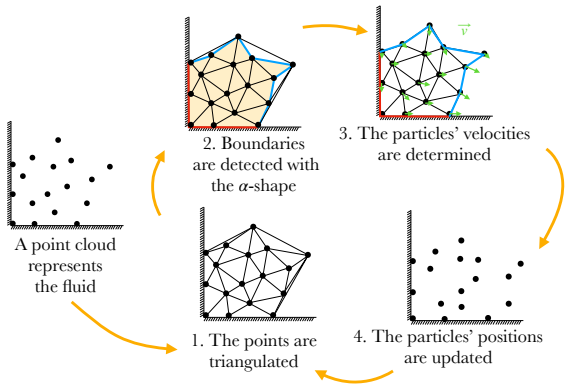

Figure 3: Main steps of the PFEM algorithm.

2.3 Briefly on boundary conditions

The slip-wall boundary condition can be imposed weakly within the finite element formulation for edges along a solid boundary. If a particle crosses a solid boundary from one time step to another due to the explicit nature of the position update, it is projected back onto the wall. Its velocity component normal to the wall is set to zero, recovering the slip wall boundary condition. Once a particle has been tagged a wall boundary particle, it stays on this boundary for the remainder of the simulation.

For the free surface boundary condition, two different situations should be considered. First, atmospheric pressure is imposed for boundaries between the fluid and the outer atmosphere. The second case concerns a cavity within the fluid, for example, a bubble of air surrounded by water. In later work, we will show that imposing incompressibility for every closed cavity is possible, such that volume of each cavity is preserved.

3. THE MESH ADAPTATION ALGORITHM

The α-shape algorithm is a clever geometrical approach to defining the shape of a set of points. The need for an "oracle" function that defines the shape of the domain has already been outlined previously. It is important to remember that the previous mesh is discarded at the beginning of each time step. Only the nodes, or particles, of the previous mesh are kept. The shape of the fluid domain at the next time step will be defined through the application of the α-shape on a new triangulation of the particles. Hence, we recover the correct shape of the domain at the beginning of each time step by adapting the triangulation.

First, a size field is defined to allow heterogeneities in the size of the mesh. For instance, a higher resolution may be desired near the free surface to accurately simulate the strong deformations and topological changes. A KD-tree structure is used to compute the distance to the free surface to derive this size field.

Next, Chew's algorithm for mesh generation is employed [3]. Chew's algorithm ensures that all triangles within the triangulation are well-sized and well-shaped and that internal and external boundaries are maintained. Shewchuck has since demonstrated that using this algorithm, no angles smaller than 26.5° will be generated, confirming the *well-shapedness* of the mesh resulting from this algorithm[7]. The user-defined size field dictates the size of the elements.

The essence of the approach consists of inserting nodes in the triangulation at the circumcenter of poorly shaped or poorly sized triangles. Mesh Delaunayness is ensured at all times through edge swaps whenever the triangulation is modified, i.e., whenever a node is inserted. If a circumcenter falls outside the meshing domain, then the bounding edge that crosses the line-of-sight of the triangle to its circumcenter is split. The algorithm is iterative and stops unconditionally.

The PFEM algorithm is adapted to incorporate the mesh adaptation step; see figure 4. Naturally, only the elements colored as part of the fluid domain should be improved. Hence, the α-shape must be used to define the shape of the fluid domain. This allows determining the bounding edges of the fluid. Chew's algorithm is then used to improve the quality of the fluid elements and respect the size field while constraining the boundary edges, both on solid walls and on the free surface. Next, the solution of the previous time step is projected onto the newly inserted particles. Finally, the

boundary conditions can be applied, and the Navier-Stokes equations are solved on the newly generated mesh.

Using the mesh adaptation algorithm in PFEM greatly improves the quality of the mesh. This step turns out to be crucial, as it removes uncertainties linked to the α-shape's decision of the shape of the fluid. A few preliminary simulation results are shown next.

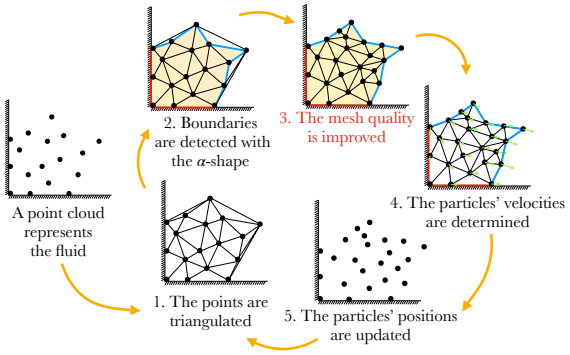

Figure 4: The updated version of the PFEM algorithm, with mesh adaptation.

4. SIMULATION RESULTS

Two preliminary simulation results are presented. Figure 5 is a drop of water falling into a container filled with water. Near the free surface, the mesh is refined to accurately capture the attachement of the drop with the rest of the domain. Effects of surface tension are not considered in this simulation. Figure 6 is a simulation of a water column collapsing in a solid container. The color code refers to the velocity norm. This benchmark simulation, more commonly known as a dam break, highlights again the interest in using a Lagrangian method. Indeed, these simulations show that it is possible to obtain smooth results with a limited amount of nodes.

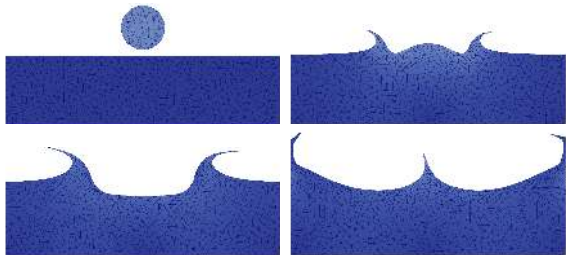

Figure 5: Simulation of a drop of liquid falling in a container.

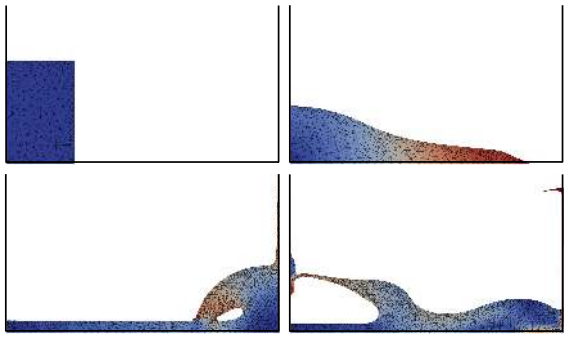

Figure 6: Simulation of a water column collapsing inside a container.

5. CONCLUSION

In this research note, we have presented a novel approach for mesh adaptation in the particle finite element method. As a Lagrangian method, PFEM simulates fluid domains that undergo strong deformations. PFEM relies on the generation of a new mesh at each time step, and the quality of the elements within the triangulation plays a vital role in defining the domain. Indeed, a geometrical function, known as the α-shape of a triangulation, is used to determine the region considered as fluid within the triangulation. To control this function, our mesh adaptation algorithm ensures that the mesh quality remains optimal within the simulation domain and that it respects a pre-defined size field. Regions of interest can thereby be refined. The algorithm has been described, and the capabilities were shown in a few preliminary simulation results. The strong dependence of PFEM on mesh quality deems mesh adaptation a crucial part of the simulation. It removes many uncertainties in the method that are known to cause errors in volume conservation and inaccurate results overall. We have shown that, with a relatively small number of degrees of freedom, using a well-designed distribution of the particles, obtaining a smooth representation of the free surface is possible.

In future works, the goal will be to obtain quantitative results in terms of efficiency, accuracy and consistency of the method. After validation in 2D, the extension to 3D will be studied. Due to the lack of proof in 3D regarding optimality of the mesh, we will focus on local adaptations of the mesh, both geometrically and topologically. Special attention will need to be put on handling slivers.

References

[1] Oñate E., Idelsohn S.R., Del Pin F., Aubry R. "The particle finite element method—an overview." *International Journal of Computational Methods*, vol. 1, no. 02, 267–307, 2004

[2] Cremonesi M., Franci A., Idelsohn S., Oñate E. "A state of the art review of the particle finite element method (PFEM)." *Archives of Computational Methods in Engineering*, vol. 27, no. 5, 1709–1735, 2020

[3] Chew L.P. "Guaranteed-quality mesh generation for curved surfaces." *Proceedings of the ninth annual symposium on Computational geometry*, pp. 274–280. 1993

[4] Tezduyar T.E. "Stabilized finite element formulations for incompressible flow computations." *Advances in applied mechanics*, vol. 28, 1–44, 1991

[5] Edelsbrunner H., Mücke E.P. "Three-dimensional alpha shapes." *ACM Transactions on Graphics (TOG)*, vol. 13, no. 1, 43–72, 1994

[6] Franci A., Cremonesi M. "On the effect of standard PFEM remeshing on volume conservation in free-surface fluid flow problems." *Computational Particle Mechanics*, vol. 4, no. 3, 331–343, 2017

[7] Shewchuk J.R. "Delaunay refinement algorithms for triangular mesh generation." *Computational geometry*, vol. 22, no. 1-3, 21–74, 2002