Moving particle semi-implicit method for fluid simulation with implicitly defined deforming obstacles

Yasutomo Kanetsuki1* and Susumu Nakata2

1Graduate School of Information Science and Engineering, Ritsumeikan University, Japan
2College of Information Science and Engineering, Ritsumeikan University, Japan

∗is006lee@ed.ritsumei.ac.jp

Received: October 6, 2014; Accepted: December 11, 2014; Published: May 31, 2015

Abstract. In this paper, we present a particle-based fluid simulation method with deforming obstacles. For defining the geometric information of time-varying obstacles, we employ the implicit function form that varies along the time axis. Our fluid simulation is based on the Moving Particle Semi-implicit (MPS) method, one of the typical particle-based methods for incompressible flow, and we add a new formulation for implicitly defined deforming obstacles. Although many of the particle-based algorithms require a set of particles on the obstacles, our method works without generating particles on the obstacles. We define the particle motion at the vicinity of the obstacles and generate a new algorithm specific to the implicit deforming obstacles by incorporating approximations in the processes of determining particle forces affected by the obstacles.

Keywords: Fluid simulation, Moving Particle Semi-implicit (MPS) method, Implicit function form, Shape deformation

1. Introduction

In this paper, we present a technique for incompressible fluid simulation with deforming obstacles. We assume that the geometry of deforming obstacles is defined by a time-varying implicit function, i.e., the boundary at time $t \geq 0$ is a set of solutions of an equation $f(x; t) = 0$, where $x \in \mathbb{R}^2$. For incompressible fluid simulation, we employ the Moving Particle Semi-implicit (MPS) method [1, 2] and we modify the algorithm for time-varying implicitly defined obstacles. Note that the spatial dimension can be extended to three without loss of generality.

In most of the particle-based methods, the obstacles are typically represented as a set of boundary particles. The main problem to be addressed for the purpose above is the formulation of particle motion in the vicinity of obstacles. Since implicitly defined obstacles do not possess boundary particles, a new formulation specific to the implicit representation is required. We give a new formulation of MPS-based fluid simulation for implicitly defined obstacles.
deforming obstacles by extending the particle-based method with polygonal boundaries represented without obstacle particles [3, 4]. Since the coordinate of a boundary point is not given explicitly, the corresponding velocity cannot be obtained using a simple partial derivative with respect to time. The idea of our formulation is to estimate the velocity at arbitrary boundary points based on the displacement of boundary in a small time interval.

Particle based fluid simulation is a widely used technique in scientific and engineering fields. The Smoothed Particle Hydrodynamics (SPH) method [5], one of the most widely used particle-based methods, has been applied not only in scientific computing but also in computer graphics [6]. The weakly compressible SPH [7] is an advanced method achieving nearly incompressible flow which is a key requirement for practical applications. Later studies such as the predictive-corrective incompressible SPH [8], the constraint fluids [9] and the position based fluids [10] also enforce incompressibility and has been used mainly in computer graphics. The MPS is another approach to accomplish incompressibility with high accuracy and has been successfully applied to a variety of engineering applications.

The shape modeling based on the implicit function form play an important role in computer graphics and various methods have been proposed in the last two decades. There are different approaches to shape modeling with implicit form such as surface modeling interpolating surface points [11, 12, 13, 14], shape modeling with user manipulation [15, 16, 17, 18] and conversion from polygonal shape models [19, 20]. In the first approach, surfaces are typically generated from a set of given oriented boundary points and current techniques allow automatic generation of accurate and detailed implicit surfaces from millions of input points. The implicit form is well suited not only to shape generation from point clouds acquired by three-dimensional scanning devices but also to the second approach: interactive modeling allowing simple user manipulation. In the third approach, users can obtain approximated implicit form of existing polygonal shape models. Although implicit form typically assumes smoothness of the surfaces, some methods enables implicit expression of sharp edges and corners [21, 22].

One important advantage of implicit representation is the expression of continuously deforming solids. The morphing technique enables users to generate intermediate shapes that interpolate two different implicitly defined shapes [23]. In this method, the intermediate shapes can be automatically generated with a simple expression even if the topology of one original shape is different from that of the other. Time-varying shapes can also be generated from time-varying point clouds [24]. This approach allows modeling of more detailed motion of the shapes. As a result, a field $f(x; t)$ is obtained and the shape at a moment is defined as a zero-level-set of the field by fixing the time variable $t$.

The implicitly defined obstacles can be used for SPH-based fluid simulation [25]. The particle motion around the implicit obstacles is determined by extending the polygon-based method [3] to the implicit form. A remarkable advantage of the use of the implicit representation is that the particle motion can be determined without boundary particles and is free from the boundary particle generation process.

In our method, the fluid simulation is performed based on the MPS which provides incompressible flow and we introduce a new definition of the particle motion acted by implicit obstacles so that the particles behave more naturally around the boundaries. The size of the linear system lead from the incompressibility condition is lower than the original MPS and
this property contributes efficient computation. In addition, assuming that the obstacles deform during the simulation, the viscosity effect by the deforming obstacles are determined depending on the velocity of the deformation.

Note that our method is limited to situations where implicit form of obstacles can be obtained. There are several different approaches to obtain implicit form depending on the data of the original shape model as described above.

2. MPS with boundary particles

The MPS is a method for numerical analysis of incompressible flow with free surfaces. The governing equations consist of the Navier-Stokes equations, the mass conservation law and the incompressibility condition:

\[
\begin{align*}
\frac{Dv}{Dt} &= -\nabla P + \nu \nabla^2 v + f^{\text{grav}}, \\
\frac{D\rho}{Dt} + \rho \nabla \cdot v &= 0, \\
\frac{D\rho}{Dt} &= 0,
\end{align*}
\]

where \(v\) is the velocity, \(\rho\) the density, \(P\) the pressure, \(\nu\) the kinematic viscosity coefficient and \(f^{\text{grav}}\) the gravity. The three terms in the right hand side of the top equation in (1) are the pressure term, the viscosity term and the gravity term, respectively.

In the MPS, the governing equations are discretized based on a particle interaction model. By letting \(N\) be the number of fluid particles and \(r^k_1, r^k_2, \ldots, r^k_N\) the positions of the fluid particles at time \(k\), the \(i\)-th fluid particle \(r^k_i\) moves to a temporal position \(r^*\) and is updated to the final position \(r^{k+1}_i\) in a time interval \(\Delta t\) according to the following formulae:

\[
\begin{align*}
v^* &= v^k + \Delta t (f^{\text{viscous}} + f^{\text{grav}}), \\
r^* &= r^k + \Delta t v^*, \\
v^{k+1} &= v^* + \Delta t f^{\text{press}}, \\
r^{k+1} &= r^* + \Delta t (\Delta t f^{\text{press}}),
\end{align*}
\]

where \(v^k_i\) is the velocity of the \(i\)-th fluid particle at time \(k\). Note that the variables with the superscript * denote the quantities at the temporal position and we assume that their definitions are obtained by replacing the superscript \(k\) with *. The fluid flow is obtained by repeating the above update process using the particle quantities, \(f_i^{\text{press}}\) and \(f_i^{\text{viscous}}\), determined in the following ways.

The pressure term \(f_i^{\text{press}}\) is obtained by the following model:

\[
f_i^{\text{press}} = -\frac{D}{\rho \nu^D} \sum_{j \neq i} \left[ \frac{P^* - P^*}{\|r^*_j - r^*_i\|^2} (r^*_j - r^*_i) \psi(|r^*_j - r^*_i|) \right],
\]

(2)
where \( D \) is the number of spatial dimensions, \( n^0 \) the constant particle number density at the initial stable state, \( w(r) \) a weight function determined by

\[
w(r) = \begin{cases} \frac{r - 1}{r} & (0 \leq r < r_e) \\ 0 & (r_e \leq r) \end{cases},
\]

whose support radius is a constant \( r_e \) and \( \hat{P}_i^* \) the minimum value of the particle pressures \( \{P_i^*\} \) in the support of the \( i \)-th particle. The pressure \( \{P_i^*\} \) is obtained as the solution of the linear system:

\[
-\frac{p}{\Delta t^2} \left( n_i^* - n^0 \right) = \frac{2D}{\Delta t^0} \sum_{j \neq i} \left[ (P_j^* - \hat{P}_i^*) w(\|r_j^* - r_i^*\|) \right],
\]

(3)

where \( \alpha \) is the compressibility of the fluid and \( \lambda \) is the coefficient determined statistically:

\[
\lambda = \frac{\sum_{j \neq i} \left[ \|r_j^0 - r_i^0\|^2 w(\|r_j^0 - r_i^0\|) \right]}{\sum_{j \neq i} w(\|r_j^0 - r_i^0\|)}.
\]

The particle number density \( n_i^k \) is evaluated using the following equations:

\[
n_i^k = \sum_{j \neq i} w(\|r_j^k - r_i^k\|).
\]

(4)

The viscosity term \( f_{i,\text{viscous}} \) is obtained using the following model:

\[
f_{i,\text{viscous}} = \nu \frac{2D}{\Delta t^0} \sum_{j \neq i} \left[ (v_j^k - v_i^k) w(\|r_j^k - r_i^k\|) \right].
\]

3. MPS with implicitly defined obstacles

In our method, the particle motion needs to be determined without using the obstacle particles, although the standard MPS requires a set of particles inside obstacles and along their boundary. Instead of using them, we divide the computational process of each particle quantity into two: the quantity determined by surrounding fluid particles and that determined by the obstacles. If a set of boundary particles is given, the quantities determined by the obstacles are described in the following equations:

\[
\begin{align*}
f_{i,\text{press}, \text{obstacle}} &= -\frac{D}{\rho n^0} \sum_{j \text{obstacle}} \left[ \frac{P_j^* - \hat{P}_i^*}{\|r_j^* - r_i^*\|^2} (r_j^* - r_i^*) w(\|r_j^* - r_i^*\|) \right], \\
f_{i,\text{viscous}, \text{obstacle}} &= \nu \frac{2D}{\Delta t^0} \sum_{j \text{obstacle}} \left[ (v_j^k - v_i^k) w(\|r_j^k - r_i^k\|) \right].
\end{align*}
\]

(5)
In order to replace these equations with new ones without boundary particles, we apply the following three models. The first model is to locate hypothetical particles at the boundaries that determine the quantities at each fluid particle affected by the obstacles. We assume that the hypothetical boundary particles are located along the tangent plane at the boundary point on the implicit obstacles closest to the fluid particle. The second one is to define an artificial pressure term affected by the obstacles. The last one is to define an approximated velocity of a boundary point on the obstacles in a specific way to the implicit representation. We assume throughout this paper that the field \( f(x; t) \) is positive in the area where the fluid particles exist so that the gradient of the field at a boundary point directs to the fluid particles.

In our formulation, we use the boundary point closest to a fluid particle in the vicinity of the boundary and the corresponding normal vector of the boundary. We denote the closest boundary point by \( \mathbf{r}_i^{\text{obstacle}} \) and the corresponding normal vector by \( \mathbf{n}(\mathbf{r}_i^{\text{obstacle}}) \) (see Fig. 1). Unfortunately, the computation of these quantities is not simple because the process of finding the closest point on implicitly defined shapes comes down to a nonlinear optimization problem. We discuss this issue and give approximations in Sec. 4.

### 3.1. Pressure term with hypothetical particles

By applying the first assumption, i.e., by assuming hypothetical boundary particles corresponding to a fluid particle \( r_i^k \), the particle number density (4) can be divided into two: the component determined by the fluid particles and that determined by the hypothetical boundary particles. The latter component comes down in our formulation to a function of the distance from the fluid particle to the closest boundary point, i.e.,

\[
Z(d_i^k) \approx \sum_{j \in \text{obstacle}} u\left(\|\mathbf{r}_j^k - \mathbf{r}_i^k\|\right),
\]

where \( r_j^k \) are the positions of the hypothetical boundary particles and \( d_i^k \equiv \|\mathbf{r}_j^k - \mathbf{r}_i^{\text{obstacle}}\| \). See Sec. 4 for the computation of \( d_i^k \).
One important property of the function $Z(d^k_i)$ is that the value depends only on the distance from a fluid particle to the corresponding closest surface point on an obstacle if the material parameters of the obstacles are constant. From this point of view, the computational process of the function $Z(d^k_i)$ can be divided into two parts: sampling and evaluation, in the same way as the method used for polygonal obstacles [4]. In the process of sampling, we generate sampling points with a small constant interval over the support $(0, r_e)$. At each sampling point, we evaluate the function using (6) with hypothetical boundary particles located at a distance of the sampling point. This process can be performed in advance of the time stepping iteration because the function is independent of the variables other than the distance. In the evaluation process, we obtain an approximation of the function value using the linear interpolation of the sampling values, i.e., the hypothetical particles are not required during the process of particle motion.

In this formulation, the hypothetical particles are assumed to be arranged along the tangent plane of a surface point which may cause a large error if the surface point has a large curvature. In addition to that, the error can be larger if there is a sharp edge around a fluid particle because only one side of two neighboring surfaces sharing the edge is adopted for the computation of $Z(d^k_i)$. From this point of view, our formulation works effectively in the case where the curvature of obstacle surfaces is relatively small compared to the fineness of fluid particles.

For the pressure term in (5), we use the second model, i.e., we define an artificial force that acts on each fluid particle as following:

$$f^\text{press}_{i, \text{obstacle}}(d^*_{i}) \equiv \begin{cases} \beta (r_{\text{init}} - d^*_{i}) \frac{n^*_i Z(d^*_{i})}{d^*_{i}^2} n(r^*_i, \text{obstacle}) & (0 \leq d^*_{i} < r_{\text{init}}) \\ 0 & (r_{\text{init}} \leq d^*_{i}) \end{cases},$$

where $\beta$ is the coefficient determining the force level, $r_{\text{init}}$ the distance between neighboring particles at the stable state (see Fig. 1). The coefficient $\beta$ needs to be chosen properly so that the magnitude of the force is a good approximation of the force level determined by the original MPS with boundary particles. In our tests, we determined an appropriate value on trial by observing the particle motions at the vicinity of the boundaries.

The above artificial force is defined so that the magnitude increases as a fluid particle approaches to an obstacle. The concept of this definition is different from the artificial force for polygonal obstacles [4] in that the magnitude is independent of the time interval $\Delta t$, although the particle motion changes depending in the polygonal approach.

### 3.2. Construction of linear system

In our formulation, the pressure term $f^\text{press}_i$ is determined as the sum of two components: the force acted by surrounding fluid particles calculated by (2) and the force acted by a neighboring obstacle calculated by (7). For the computation of (2), the pressure values $\{P^*_i\}$ at all the fluid particles are required. Note that the superscript $*$ denotes the quantities corresponding to temporal particles positions $\{r^*_i\}$ after applying the viscous term and the gravity as described in Sec. 2. In the standard MPS, the pressure is determined by solving the
linear system (3). The problem of our method is that the quantities, \( P_i^* \) and \( r_j^* \), corresponding to \( j \in \text{obstacle} \) cannot be determined because the corresponding particles do not exist.

To address this problem, we apply the approximation used in the MPS for polygonal obstacles [4], i.e., we approximate the unknowns using the two assumptions, \( r_j^* \approx r_{j, \text{obstacle}} \) and \( P_j^* \approx P_{j, \text{obstacle}}^* \) for \( j \in \text{obstacle} \), where \( r_{j, \text{obstacle}}^* \) is the boundary point closest to the \( i \)-th fluid particle (see Fig. 1) and \( P_{j, \text{obstacle}}^* \) the pressure at the surface point. By applying the second assumptions, the right-hand-side of (3) can be replaced to the following expression:

\[
\begin{align*}
\frac{2D}{\Delta t^0} \sum_{j \neq i, j \in \text{fluid}} \left[ (P_j^* - P_i^*) w(||r_j^* - r_i^*||) \right] + (P_{i, \text{obstacle}}^* - P_i^*) \sum_{j \in \text{obstacle}} w(||r_j^* - r_i^*||) \\
\approx \frac{2D}{\Delta t^0} \sum_{j \neq i, j \in \text{fluid}} \left[ (P_j^* - P_i^*) w(||r_j^* - r_i^*||) \right] + (P_{i, \text{obstacle}}^* - P_i^*) Z(d_i^*). 
\end{align*}
\]

(8)

In this equation, the pressure \( P_{i, \text{obstacle}}^* \) remains unknown. To determine this value, we additionally apply the both assumptions to the top of (5) and obtain the following equation:

\[
\begin{align*}
f_{\text{press}}^*_i, \text{obstacle} &= -\frac{D}{\rho m^0} \cdot \frac{P_i^* \text{,obstacle} - \dot{P}_i^*}{||r_{i, \text{obstacle}}^* - r_i^*||} \left( r_{i, \text{obstacle}}^* - r_i^* \right) \sum_{j \in \text{obstacle}} w(||r_j^* - r_i^*||) \\
&\approx \frac{D}{\rho m^0} \cdot \frac{P_i^* \text{,obstacle} - \dot{P}_i^*}{d_i^*} n(r_{i, \text{obstacle}}^*) Z(d_i^*).
\end{align*}
\]

By comparing this with (7), we obtain the pressure \( P_{i, \text{obstacle}}^* \) by

\[
P_{i, \text{obstacle}}^* \approx P_i^* + \frac{\rho m^0 n_i^* (r_{\text{init}} - d_i^*)}{D d_i^*}.
\]

(9)

Finally, we obtain the following equation by substituting (8) and (9) to (3):

\[
\begin{align*}
-\frac{\rho}{\Delta t^0} \left( \frac{n_i^* - n^0}{n^0} - \alpha P_i^* \right) \\
= \frac{2D}{\Delta t^0} \sum_{j \neq i, j \in \text{fluid}} \left[ (P_j^* - P_i^*) w(||r_j^* - r_i^*||) \right] + \frac{2\rho m^0 n_i^* (r_{\text{init}} - d_i^*) Z(d_i^*)}{\lambda d_i^*}. 
\end{align*}
\]

(10)

This is the linear system used in practical computation to determine the unknown pressure \( P_i^* \) (\( i \in \text{fluid} \)) and all the other variables in this system are known except for the distance \( d_i^* \) which we discuss in Sec. 4.

### 3.3. Viscous term with deforming obstacles

For the computation of \( f_{i, \text{obstacle}}^{\text{viscous}} \) in (5), the velocities of the hypothetical particles \( v_j^k \) (\( j \in \text{obstacle} \)) are required. In order to determine these quantities, we define the velocity at the
boundary point closest to the \(i\)-th fluid particle and assign the velocity to all the hypothetical particles around the \(i\)-th fluid particle. Let \(v^k_{i, \text{obstacle}}\) be the velocity of the surface at \(r^k_{i, \text{obstacle}}\). We estimate the velocity in the following way. From the definition of the implicit representation, the point \(r^k_{i, \text{obstacle}}\) satisfies \(f(r^k_{i, \text{obstacle}}; t) = 0\) and we assume that, after a time interval \(\delta t\), the boundary point moves to \(\hat{r}^k_{i, \text{obstacle}}\) satisfying \(f(\hat{r}^k_{i, \text{obstacle}}; t + \delta t) = 0\).

The boundary point \(\hat{r}^k_{i, \text{obstacle}}\) can be obtained according to the approximation of the closest boundary point described in Sec. 4. Note that \(\delta t\) is independent of the time interval used for the MPS iteration. In order to determine the destination uniquely, we give an assumption that the destination is the boundary point on \(f(x; t + \delta t) = 0\) closest to the particle \(r^k_{i}\). Using these two boundary points, we estimate the velocity by the difference as

\[
v^k_{i, \text{obstacle}} \equiv \frac{\hat{r}^k_{i, \text{obstacle}} - r^k_{i, \text{obstacle}}}{\delta t}\]

and we assign this velocity to the hypothetical particles, i.e., \(v^j_k \equiv v^k_{i, \text{obstacle}}\) for all \(j \in \text{obstacle}\). Then the bottom equation of (5) can be rewritten as:

\[
f_{\text{viscous}}^{i, \text{obstacle}}(d^k_i) = \frac{2D}{\Delta y^0} \left(v^k_i, \text{obstacle} - v^k_j\right) \sum_{j \in \text{obstacle}} w\left(\|r^k_j - r^k_i\|^2\right)
\]

\[
\approx \frac{2D}{\Delta y^0} \left(v^k_i, \text{obstacle} - v^k_i\right) Z(d^k_i).
\]  

As a result, the two quantities, \(f_{\text{press}}^{i, \text{obstacle}}(d^k_i)\) and \(f_{\text{viscous}}^{i, \text{obstacle}}(d^k_i)\), can be estimated using (7) and (12) which we use for practical computations. Note that the function \(Z(d^k_i)\) required for the computation depends only on the distance variable and can be computed in preprocessing which contributes to efficient computation. For the computation of the pressures in (2), we generate the linear system (3) using (10) and solve it at every time step.

### 4. Estimation of closest boundary point

Since the obstacles are represented by implicit function form, the problem of finding the boundary point closest to each fluid particle is not simple as described in the previous section. In this section, we give an approximation specific to the implicit function form.

Let \(r^k_{i, \text{obstacle}}\) be the boundary point closest to the \(i\)-th fluid particle \(r^k_i\). The idea of finding the closest point is to estimate the distance from the \(i\)-th fluid particle to the corresponding closest boundary point and the normal vector of the implicit boundary at the boundary point. If the two quantities, the distance \(d^k_i \equiv \|r^k_i - r^k_{i, \text{obstacle}}\|\) and the normal vector \(n(r^k_{i, \text{obstacle}}) \equiv \nabla f(r^k_{i, \text{obstacle}}; t) / \|\nabla f(r^k_{i, \text{obstacle}}; t)\|\) (see Fig. 1), are given, the closest surface point can be determined by

\[
r^k_{i, \text{obstacle}} = r^k_i - d^k_i n(r^k_{i, \text{obstacle}})
\]  

under the assumption that \(f(x; t)\) is positive in the area where the fluid particle exists.
In order to estimate the distance, we use the Taubin’s distance approximation \[26\], i.e.,
\[ d_k^i \approx f(r_k^i; t) \left\| \nabla f(r_k^i; t) \right\|. \] (14)

In our formulation, the distance is required only when the particle \( r_k^i \) is close to the boundary and it is expected to obtain an accurate distance because the accuracy of the Taubin’s distance tends to increase as the evaluation point approaches to the boundary. For the estimation of the other quantity, the normal vector \( n(r_k^i, \text{obstacle}) \), we give an assumption that the fluid particle is close to the boundary, i.e., \( r_k^i \approx r_k^i, \text{obstacle} \), and then the normal vector can be estimated by
\[ n(r_k^i, \text{obstacle}) \approx \frac{\nabla f(r_k^i; t)}{\left\| \nabla f(r_k^i; t) \right\|}. \] (15)

As a result, the approximate boundary point can be obtained by substituting (14) and (15) to (13). We use the approximate distance (14) for the computation of (6), (7), (10) and (12) and the approximate boundary point (13) for the computation of (11).

5. Results

We tested our algorithm with two different cases: fluid simulation with a static obstacle and that with a deforming obstacle. For the parameter \( \beta \) in (7) and (10), we employed 5.0 which provides appropriate behavior of fluid particles around obstacles. Fig. 2 (left column) illustrates the results of our fluid simulation with the static implicit obstacle the field of which is defined by \( f(x; t) = \left( \sqrt{x^2 + y^2} - 0.5 \right)^2 - 0.168^2 \). For the initial state, we uniformly fill the right half area with 7,900 fluid particles as illustrated in Fig. 2 (a1). The result at \( t = 0.4 \) [s] illustrate that the behavior of the fluid particles are similar to that obtained by the standard MPS with obstacle particles (Fig. 2 (b2)) as a whole, although each particle behaves differently.

The results of our method with a deforming obstacle is illustrated in Fig. 3. In this test problem, we define the field \( f(x; t) \) by interpolating two different static fields, \( f_1(x) \) and \( f_2(x) \) with a parameter \( t \) in the following way:
\[ f(x; t) = T(t) f_1(x) + (1 - T(t)) f_2(x) = 0, \]
\[ f_1(x) = r^4 - x^4 - y^4, \]
\[ f_2(x) = r^2 - x^2 - y^2, \]
\[ T(t) = \cos^2 t. \]

We put 4,659 fluid particles at the right half of the region as the initial state. The results show that an adequate particle motion can be achieved in the vicinity of the boundary by our formulation with deforming obstacles.
6. Conclusion

We developed a particle based fluid simulation method with implicitly defined deforming obstacles. We adopted the MPS method for fluid simulation and implicit form for representing deforming obstacles.

For the problem of the lack of obstacle particles, we incorporated the model that a set of hypothetical boundary particles along the tangent plane at the boundary point closest to every fluid particles. In practical computations, the hypothetical boundary particles are not required because the quantities determined by this model can be replaced by a function of the distance from each fluid particle to the obstacle. For the pressure term, we defined an original artificial force specific to the implicit form so that the fluid particles behaves similarly as the particle motion of the standard MPS. We also defined an approximate velocity of arbitrary boundary point of deforming implicit obstacles by determining a corresponding boundary point after a small time interval.

The results of our algorithm show adequate particle motion in the vicinity of the boundary without obstacle particles. The fluid simulation with deforming obstacles was performed and this approach enabled us to avoid the process of generating boundary particles.

The method is limited to the problems where obstacles are represented in implicit form which can be generated using implicit modeling techniques in computer graphics. In addi-
Figure 3: Results at every 0.3 [s] starting from initial state. The blue circle represents the fluid particle and the black represents the deforming obstacle \( f(x; t) = 0 \).

...tion, the accuracy of particle motion changes depending on the curvature of the boundary and this point is remained as a future work.

**Acknowledgement**

This work was supported by JSPS KAKENHI Grant Numbers 24760066 and 25280044.

**References**


