Sci. Rep. Kanazawa Univ.
Vol. 62, pp. 51–68, 2018

An ODE control system of a rigid body on an ocean wave for a surfer simulation in the SPH method

Reza Rendian SEPTIAWAN*

Graduate School of Natural Science and Technology, Kanazawa University,
Kakuma, Kanazawa, 920-1192, Japan

(Received June 26, 2018 and accepted in revised form August 7, 2018)

Abstract In this work we use a smoothed particle hydrodynamics (SPH) method coupled with a rigid body simulation to simulate a surfing board on top of an ocean wave. External forces are applied to the board to represent a surfer trying to control a surfing board. An ordinary differential equation (ODE) control is used to manipulate the external forces based on a position, velocity, and an inclination angle of the surfing board. The control system successfully helps the surfing board to move to and maintain its desired position.

Keywords. ODE, rigid body simulation, smoothed particle hydrodynamics, surfing board simulation.

1 Introduction

Surfing on ocean waves poses some interesting mathematical and physics problems. One of them is a modeling of a surfer that controls the movement of the surfing board. In what we call a surfing problem, the goal is to maintain the position of the surfing board to be on the upslope part of the ocean wave as long as possible. The surfer maneuvers the surfing board by controlling the force given to the board by pushing or shifting their feet on the board, effectively changing the distribution of forces exerted on the board and modifying its inclination angle.

Since the upslope part of the ocean wave is the domain of interest in the surfing problem, in this work we choose a frame of reference to be located there. This frame moves together with the wave. The goal of this work is to control the position of the surfing board to be at the desired point with respect to the wave. Since the only thing that we can control in an attempt to move the surfing board toward the desired position is its inclination angle, we propose an ODE control of the inclination angle that represents a surfer in a surfing problem. The control takes into account only the position, velocity, and the inclination angle of the surfing board. The inclination angle from the ODE control acts as a "target angle" for the surfing board. To reach the target angle we give two forces at the tips of the board, mimicking the surfer giving forces to the board via their feet. To find suitable parameters for which the control system is stable, we perform a stability analysis of a linearized simplified one-dimensional ODE model of the surfer on an ocean wave.

*Email: za22061991@gmail.com

To verify the capabilities of our ODE control in controlling the dynamics of the rigid body under interactions with the fluid, we simulate the full system using the smoothed particle hydrodynamics (SPH) method. SPH method is one of the most popular Lagrangian solvers of the fluid equations. It was introduced by Lucy [1], and Gingold and Monaghan [2] to solve astrophysical problems. Since then, the SPH method has been used to simulate computational fluid dynamics problems, such as shockwave problems [3, 4], heat transfer problems [5], multi-phase fluids [6], and also water waves [7, 8].

Another interesting problem which can be solved by the SPH method is a fluid-rigid body interaction. A common method to include a rigid body into an SPH simulation is by discretizing the rigid body into a set of boundary points which can interact with other SPH fluid points by exchanging momentum using either a normal SPH interaction scheme for hydrodynamics, or a modified interaction scheme to suit the problem. In [9], the interaction between rigid bodies and fluids is done by using a Lennard-Jones potential repulsive boundary force, called the Monaghan boundary force (MBF), to prevent penetration of fluid points into a rigid body. However, since MBF is designed to prevent penetration of a point with the maximum velocity, the repulsive force is thought to be too strong for slow moving points. In [10] an impulse-based boundary force (IBF) was proposed. Recently, a more robust IBF was introduced in [11] by using a sequential impulse to solve the frictional contact problem with many contact points.

Since the IBF method relies on the normal at the surface of rigid bodies, the calculation for a more complex-shaped rigid body is not easy. Also, IBF is not purely based on hydrodynamic forces. A more versatile interaction between rigid bodies and fluid points was introduced in [12], which also solves a neighbor deficiency problem near the boundary by the inclusion of boundary points in the density calculation.

In this work, a rigid body is also discretized into a set of boundary points. An interaction between fluid points and boundary points is done by using a boundary force based on purely hydrodynamic forces. To prevent particles to penetrate into the rigid body, we give boundary points a constant density function which is equal to a reference density, leaving the repulsive force solely determined by fluid points near the boundary itself.

2 Governing equations

2.1 Fluid dynamics

The motion of the fluid can be observed in two different frames of reference: an Eulerian frame which is fixed in space so that the flow of fluid is considered as a flux of physical quantities, and a Lagrangian frame which treats the fluid as material points, with each point carrying its physical quantities as it moves. As an SPH method uses Lagrangian frame intensively, we will discuss the governing equation of the fluid in a Lagrangian description. The motion of fluid is governed by following conservations laws which are the Euler's equations of the fluid dynamics [13]:

1. Conservation of mass:

$$\frac{D\rho}{Dt} = -\rho \operatorname{div}(u) \tag{2.1}$$

2. Conservation of momentum (for an inviscid fluid):

$$\frac{Du}{Dt} = -\frac{1}{\rho}\nabla p + b \tag{2.2}$$

Here ρ, p, and u are density, pressure, and velocity field, respectively, and b is a body force per unit mass. $\frac{D}{Dt}$ is the so-called *substantial derivative* operator defined as

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + u \cdot \nabla f$$

for any field function $f(x,t)$ and velocity field u.

2.2 Rigid body dynamics

Following [14], let us consider a rigid body with a mass M independent of time specifically given by

$$M = \int_{\mathscr{R}} \rho(x)\, dx,$$

where ρ is a density, and $\mathscr{R}$ is a configuration of the rigid body in a principal frame. The center of mass in the inertial frame is denoted by $X(t)$, and the velocity is $U(t) = \frac{d}{dt}X(t)$. The linear motion of the rigid body is described by its linear momentum

$$G(t) = MU(t). \tag{2.3}$$

Let $F^i(t) = F(x^i,t)$ be a force that acts on a point x^i on the rigid body at a given time t. The total force acting on the rigid body equals to the rate of change of linear momentum

$$F(t) = \frac{d}{dt}(MU(t)) = M\frac{d}{dt}U(t) = MA(t),$$

where $F(t) = \sum_i F^i(t)$ is the resultant force acting on the rigid body, and $A(t)$ is the linear acceleration of the rigid body.

Let $\mathbf{J}$ be the principal moment of inertia tensor. $\mathbf{J}$ is a diagonal matrix whose components are given by

$$\begin{aligned} J_1 := J_{11} &= \int_{\mathscr{R}} \rho(x)(x_2^2 + x_3^2)\, dx, \\ J_2 := J_{22} &= \int_{\mathscr{R}} \rho(x)(x_1^2 + x_3^2)\, dx, \\ J_3 := J_{33} &= \int_{\mathscr{R}} \rho(x)(x_1^2 + x_2^2)\, dx. \end{aligned}$$

Let $\mathbf{R}(t)$ be the rotation matrix of the rigid body. The angular velocity vector $\omega(t)$ of the rigid body is defined in a way that for any fixed vector $v \in \mathbb{R}^3$ we have

$$\frac{d}{dt}(\mathbf{R}(t)v) = \mathbf{W}(t)\mathbf{R}(t)v = \omega(t) \times \mathbf{R}(t)v, \tag{2.4}$$

where $\mathbf{W}(t) = \frac{d\mathbf{R}(t)}{dt}\mathbf{R}^T(t)$. Notice that $\mathbf{R}(t)$ is an orthogonal matrix, that is, $\mathbf{R}(t)\mathbf{R}^T(t) = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Hence,

$$0 = \frac{d}{dt}\left(\mathbf{R}(t)\mathbf{R}^T(t)\right) = \frac{d\mathbf{R}(t)}{dt}\mathbf{R}^T(t) + \mathbf{R}(t)\frac{d\mathbf{R}^T(t)}{dt} = \mathbf{W}(t) + \mathbf{W}^T(t).$$

Therefore $\mathbf{W}(t)$ is a skew-symmetric matrix and can be written as

$$\mathbf{W}(t) = \begin{bmatrix} 0 & -\omega_3(t) & \omega_2(t) \\ \omega_3(t) & 0 & -\omega_1(t) \\ -\omega_2(t) & \omega_1(t) & 0 \end{bmatrix}. \tag{2.5}$$

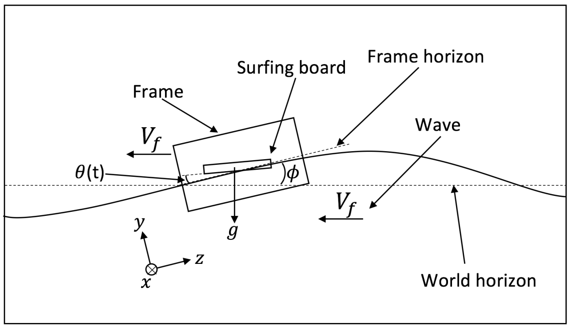


Figure 1: Sideview of the illustration of the frame of system's domain.

The vector $\omega(t) = (\omega_1(t), \omega_2(t), \omega_3(t))$ satisfies (2.4), since $\mathbf{W}(t)w = \omega(t) \times w$ for any vector $w \in \mathbb{R}^3$.

The angular momentum of the rigid body is defined as

$$H(t) = \mathbf{R}(t)\mathbf{J}\mathbf{R}^T(t)\omega(t).$$

The total moment of force acting on the rigid body equals to the rate of change of its angular momentum. Let $M^i(t) = (x^i - X(t)) \times F^i(t)$ be the moment of force F^i at a point x^i on a rigid body with a center of mass $X(t)$ at a given time t. Let $K(t) = \sum_i M^i(t)$ be the total moment of force acting on a rigid body at a given time t. Now we have

$$K(t) = \frac{d}{dt}H(t) = \dot{\mathbf{R}}(t)\mathbf{J}\mathbf{R}^T(t)\omega(t) + \mathbf{R}(t)\mathbf{J}\dot{\mathbf{R}}^T(t)\omega(t) + \mathbf{R}(t)\mathbf{J}\mathbf{R}^T(t)\dot{\omega}(t),$$

where $\dot{\mathbf{R}}(t) = \frac{d}{dt}\mathbf{R}(t)$ and $\dot{\omega}(t) = \frac{d}{dt}\omega(t)$. This can be rewritten as

$$\hat{K}(t) = \hat{\omega}(t) \times \mathbf{J}\hat{\omega}(t) + \mathbf{J}\dot{\hat{\omega}}(t), \tag{2.6}$$

where $\hat{K}(t) = \mathbf{R}^T(t)K(t)$, $\hat{\omega}(t) = \mathbf{R}^T\omega(t)$, and $\dot{\hat{\omega}}(t) = \frac{d}{dt}\left(\mathbf{R}(t)^T\omega(t)\right) = \mathbf{R}^T(t)\dot{\omega}(t)$. (2.6) is the Euler's equation of rigid body dynamics [14].

2.3 ODE model of the surfer

Let us first introduce the geometry of the domain. In this work we use the frame of reference which moves together with the wave and we assume that the wave moves with a constant velocity. The illustration of the domain can be seen in Figure 1. The goal of our surfing problem is to control the position of the surfing board to be at the desired position $\tilde{Z}$. In our surfing problem, the only parameter that we can control is the inclination angle of the board. We introduce the following ODE control for the inclination angle:

$$\dot{\theta}(t) = a(Z(t) - \tilde{Z}) + b(V(t) - \tilde{V}) + c(\theta(t) - \tilde{\theta}), \tag{2.7}$$

where $Z(t)$ and $V(t)$ are the third component of position $X(t)$ and linear velocity of the surfing board, respectively, and $\theta(t)$ is the inclination angle of a surfing board. $\tilde{Z}$, $\tilde{V}$, and $\tilde{\theta}$ are desired position, desired velocity, and desired angle, respectively. Constants a, b, and c are to be fixed below. The desired position $\tilde{Z}$ is given. Because of our choice of frame of reference, in a stable condition, the velocity of the surfing board is zero relative to both the reference frame and the ocean wave, we take $\tilde{V} = 0$. $\tilde{\theta}$ is a desired inclination angle that helps stabilize the board. Up to now we do not have any information about $\tilde{\theta}$. Later we run simulations with various values of $\tilde{\theta}$ to find the value that gives the best result.

To find suitable parameters a, b, and c, we consider a simplified linearized ODE model with the ODE control (2.7). Then we do a stability analysis of this ODE model and choose parameters a, b, and c that can stabilize the system.

From the definition, we have $\dot{Z}(t) = V(t)$. We assume that the acceleration of the rigid body depends on an external body force (in this case, gravity) and a drag force. Since we do not know the drag force in our system, by assuming that the system is close to the stationary point, we can linearize the acceleration of the system as a function of the position, velocity, and the inclination angle of the board. Let $\xi(t) = (Z(t), V(t), \theta(t))$ be the unknown of the system. We consider a simplified linearized ODE model in the following form:

$$\begin{cases} \dot{Z}(t) = V(t), \\ \dot{V}(t) = -\mu\theta(t) - \mu_v V(t) - \mu_z Z(t) - \mu_0, \\ \dot{\theta}(t) = a(Z(t) - \tilde{Z}) + bV(t) + c(\theta(t) - \tilde{\theta}), \end{cases} \tag{2.8}$$

where μ, μ_v, μ_z, and μ_0 are constants related to the drag and gravity forces. Now, let us assume $\mu_v = \mu_z = 0$ for simplicity. Later in the results section we will see that this assumption is not justified, but it does not seem to influence the stability.

We can write (2.8) in a matrix form as

$$\begin{aligned} \dot{\xi}(t) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\mu \\ a & b & c \end{bmatrix} \xi(t) + \begin{bmatrix} 0 \\ -\mu_0 \\ -a\tilde{Z} - c\tilde{\theta} \end{bmatrix} \\ &=: \mathbf{A}\xi(t) + \zeta. \end{aligned}$$

The stationary point is stable if all eigenvalues of matrix $\mathbf{A}$ have negative real parts.

The characteristic equation of $\mathbf{A}$ is

$$\det(\mathbf{A} - \lambda\mathbf{I}) = -\lambda^3 + c\lambda^2 - b\mu\lambda - a\mu = 0. \tag{2.9}$$

By using Vieta's formulas, we see that for all roots of (2.9) to be negative, the value of a and b must be positive, and c must be negative.

Let us set the roots of (2.9) to be $\lambda_1 = -3$, $\lambda_2 = -4$, and $\lambda_3 = -5$. This yields the equation

$$-\lambda^3 - 12\lambda^2 - 47\lambda - 60 = 0. \tag{2.10}$$

Comparing this with (2.9), we get $a = \frac{60}{\mu}$, $b = \frac{47}{\mu}$, and $c = -12$.

The inclination angle from the ODE control is passed to the simulation as a target inclination angle for controlling the movement of the rigid body. We model the controlling action of the surfer on a surfing board by using a two-points model, imitating the locations where the surfer places their feet. The location of those points in our model, each is denoted as r_{c1} and r_{c2}, respectively, can be seen in Figure 2. They are located at the tips of the elongated axis of the rigid body. The total force is constant, but the individual forces are controlled by a ratio $T(t)$ as

$$F_{c1}(t) = T(t)W \tag{2.11}$$

$$F_{c2}(t) = (1 - T(t))W, \tag{2.12}$$

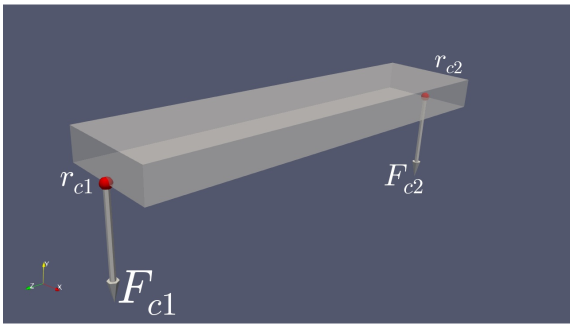


Figure 2: The locations of contact points.

where W is the weight of the surfer, $F_{c1}(t)$ and $F_{c2}(t)$ are the forces given at each point r_{c1} and r_{c2}, respectively, at a given time t, with the total force being constant as they represent the weight of the surfer. The ratio is controlled by the linear function

$$T(t) = 0.5 - \sigma(\hat{\theta}(t) - \min(\theta(t), \theta_m)), \tag{2.13}$$

where $\theta(t)$ is the angle from the ODE control in Section 2.3, θ_m is an allowed maximum inclination angle, and $\hat{\theta}(t)$ is the observed angle at a given time t. σ is a given constant. Here we use a clipped function for the inclination angle since the surfer does not want the front part of their surfing board to be immersed into the water. Note that we choose the inclination angle to be negative when the board is inclined upward, and vice versa. We do not impose $T(t) \in [0,1]$, which is reasonable if the surfer is using straps on their feet.

3 SPH method

3.1 Basic idea of SPH method

The basic idea of SPH comes from a convolution between any field function $f(x)$ with a sufficiently smooth mollifier $\psi(x)$,

$$(f * \psi)(x) := \int_{\mathbb{R}^n} f(y)\psi(x-y)\,dy.$$

If $f \in L^1(\mathbb{R}^n)$ and $\psi \in C_c^k(\mathbb{R}^n)$, then $(f*\psi) \in C_c^k(\mathbb{R}^n)$, which means the convolution product is as smooth as the smoothest of the convolved functions, see [15] for more details.

Let us define

$$\psi_h(x) := \frac{1}{h^n}\psi\left(\frac{x}{h}\right).$$

Clearly $\int_{\mathbb{R}^n} \psi_h(x)\,dx = \int_{\mathbb{R}^n} \psi(x)\,dx$, but the "mass" of ψ_h is more concentrated to the origin as $h \to 0$. If $\int_{\mathbb{R}^n} \psi(x)\,dx = 1$, the family $\{\psi_h\}_{h>0}$ is called an approximate identity. If $f \in C^k(\mathbb{R}^n)$ for some $1 \le k < \infty$, then we have $f * \psi_h \to f$ uniformly as $h \to 0$,

$$f(x) \approx \int_{\mathbb{R}^n} f(y)\psi_h(x-y)\,dy, \tag{3.1}$$

and by following the differentiation of a convolution in [15], the approximation for the derivative of f is

$$\partial^\alpha f(x) \approx \int_{\mathbb{R}^n} f(y)\partial^\alpha \psi_h(x-y)\,dy, \qquad |\alpha| \le k, \tag{3.2}$$

for $\psi \in C_c^k(\mathbb{R}^n)$. Both (3.1) and (3.2) are SPH approximations for a field function and its derivative in an integral form.

We can choose a smoothing function ψ. In this work we are using a compactly-supported piecewise cubic kernel function [16]

$$\psi(x) = \frac{\alpha_n}{6}\begin{cases} (2-|x|)^3 - 4(1-|x|)^3, & 0 \le |x| < 1 \\ (2-|x|)^3, & 1 \le |x| < 2 \\ 0, & 2 \le |x| \end{cases} \tag{3.3}$$

where α_n is 1, $\frac{15}{7\pi}$, or $\frac{3}{2\pi}$ for $n = 1,2,3$ respectively. Note that $\psi \in C_c^2(\mathbb{R}^n)$.

3.2 Discretization of the fluid equations using SPH

To discretize the fluid equations, we keep track of the physical quantities at discrete points that move with the fluid. Let us assume that we have N points with positions $r_i = r_i(t)$ which are reasonably uniformly distributed. We approximate (3.1) and (3.2) by a discrete integration,

$$f(x) \approx \int_{\mathbb{R}^n} f(y)\psi_h(x-y)\,dy \approx \sum_{i=1}^{N} f(r_i)\psi_h(x-r_i)V(E_i), \tag{3.4}$$

and

$$\partial^\alpha f(x) \approx \int_{\mathbb{R}^n} f(y)\partial^\alpha \psi_h(x-y)\,dy \approx \sum_{i=1}^{N} f(r_i)\partial^\alpha \psi_h(x-r_i)V(E_i), \tag{3.5}$$

where $V(E_i)$ is a volume of the set E_i around a point r_i and we assume that the integrand is a constant or close to being linear if E_i is symmetric and r_i is at its center of mass. A common way to approximate $V(E_i)$ is to use a mass and density at point r_i,

$$V_i(E_i) \approx \frac{m_i}{\rho_i}, \tag{3.6}$$

where m_i and ρ_i are mass and density at a point r_i, respectively. By using (3.4) for the density field function and by (3.6), we have

$$\rho(x) \approx \sum_{i=1}^{N} \rho_i \psi_h(x-r_i)V(E_i) \approx \sum_{i=1}^{N} m_i \psi_h(x-r_i).$$

Then we have an approximation for $V(E_i)$ as

$$V(E_i) \approx \frac{m_i}{\sum_{j=1}^{N} m_j \psi_h(r_i - r_j)}. \tag{3.7}$$

Another way to update the density is to use an SPH discretized approximation of (2.1) at a point r_i at a given time t, that is,

$$\frac{d\rho_i}{dt} = -\rho_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} u_j \cdot \nabla \psi_h(r_i - r_j), \tag{3.8}$$

where m_i is a mass of a point r_i, $\rho_i(t)$ and $u_i(t)$ are density and velocity of a point r_i at a given time t, respectively. The more common form for (3.8) is its anti-symmetrized version of (3.8) from [16]

$$\frac{d\rho_i}{dt} = \rho_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} (u_i - u_j) \cdot \nabla \psi_h(r_i - r_j). \tag{3.9}$$

In this work we use (3.9) to update the density of points.

The SPH discretized approximation for (2.2) at a point r_i at a given time t is

$$\frac{du_i}{dt} = -\frac{1}{\rho_i} \sum_{j=1}^{N} \left(m_j \frac{p_j}{\rho_j} \nabla \psi_h(r_i - r_j) \right) + b_i, \tag{3.10}$$

where $p_i(t)$ is a pressure at a point r_i at a given time t, and b_i is a body force per unit mass at a point r_i. Remember that in this work we assume the fluid to be an inviscid fluid. The anti-symmetrized version [16] of (3.10) is

$$\frac{du_i}{dt} = -\sum_{j=1}^{N} \left(m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla \psi_h(r_i - r_j) \right) + b_i. \tag{3.11}$$

In this work we use (3.11) to update the acceleration of points.

The pressure at a point r_i is assumed to be a function of density by using Tait's relation as follows [17, 18, 19]:

$$p_i = \frac{c^2 \rho_0}{\gamma} \left(\left(\frac{\rho_i}{\rho_0} \right)^{\gamma} - 1 \right), \tag{3.12}$$

where ρ_0 is a reference density of fluid, c is a speed of sound in a fluid, and $\gamma = 7$ for water-like fluid.

As is common in the SPH literature, we use the leapfrog time integrator scheme to evolve physical quantities of material points as follows

$$u_i\left(t + \frac{\tau}{2}\right) = u_i\left(t - \frac{\tau}{2}\right) + \frac{du_i}{dt}(t)\tau, \tag{3.13}$$

$$r_i(t+\tau) = r_i(t) + u_i\left(t + \frac{\tau}{2}\right)\tau, \tag{3.14}$$

$$u_i(t+\tau) = u_i\left(t + \frac{\tau}{2}\right) + \frac{du_i}{dt}(t)\frac{\tau}{2}, \tag{3.15}$$

$$u_i(-\frac{\tau}{2}) = u_i(0) - \frac{du_i}{dt}(0)\frac{\tau}{2}, \tag{3.16}$$

where τ is a timestep.

3.3 Rigid body dynamics

Since $\mathbf{J}$ is a diagonal matrix, we can write (2.6) componentwise as

$$J_\alpha \dot{\hat{\omega}}_\alpha(t) = \hat{K}(t)_\alpha + \hat{\omega}_\beta(t)\hat{\omega}_\gamma(t)\left(J_\beta - J_\gamma\right), \tag{3.17}$$

where $(\alpha, \beta, \gamma) = (1,2,3), (2,3,1)$ and $(3,1,2)$. Following [20], in the spirit of the leapfrog time integrator, the angular velocity vector $\hat{\omega}(t)$ can be updated using the fixed point iteration

$$\hat{\omega}_\alpha^{(m+1)}\left(t + \frac{\tau}{2}\right) = \hat{\omega}_\alpha\left(t - \frac{\tau}{2}\right) + \frac{\tau}{J_\alpha}\left(\hat{K}_\alpha(t) + \hat{\omega}_\beta^{(m)}(t)\hat{\omega}_\gamma^{(m)}(t)\left(J_\beta - J_\gamma\right)\right), \tag{3.18}$$

where τ is a timestep, and (m) is an iteration step. Initially, we can set

$$\hat{\omega}_{\alpha}^{(0)}\left(t+\frac{\tau}{2}\right)=\hat{\omega}_{\alpha}\left(t-\frac{\tau}{2}\right).$$

We also choose

$$\hat{\omega}_{\beta}^{(m)}(t)\hat{\omega}_{\gamma}^{(m)}(t)=\frac{1}{2}\left(\hat{\omega}_{\beta}\left(t-\frac{\tau}{2}\right)\hat{\omega}_{\gamma}\left(t-\frac{\tau}{2}\right)\;+\;\hat{\omega}_{\beta}^{(m)}\left(t+\frac{\tau}{2}\right)\hat{\omega}_{\gamma}^{(m)}\left(t+\frac{\tau}{2}\right)\right).$$

Later, we update the rotation matrix by using (2.4) and in the sense of leapfrog integrator as

$$\begin{aligned}\mathbf{R}(t+\tau)&=\mathbf{R}(t)+\tau\frac{d}{dt}\mathbf{R}\left(t+\frac{\tau}{2}\right)\\&=\mathbf{R}(t)+\tau\mathbf{W}\left(t+\frac{\tau}{2}\right)\mathbf{R}\left(t+\frac{\tau}{2}\right),\end{aligned}$$

where $\mathbf{W}$ is a matrix from (2.5) and $\mathbf{R}\left(t+\frac{\tau}{2}\right)=\frac{(\mathbf{R}(t)+\mathbf{R}(t+\tau))}{2}$. After some algebraic operations, we have

$$\begin{aligned}\mathbf{R}(t+\tau)&=\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\left(t+\frac{\tau}{2}\right)\right)^{-1}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\left(t+\frac{\tau}{2}\right)\right)\mathbf{R}(t)\\&=\Theta\left(t+\frac{\tau}{2}\right)\mathbf{R}(t),\end{aligned}\tag{3.19}$$

where $\mathbf{I}$ is the identity matrix, and

$$\Theta\left(t+\frac{\tau}{2}\right):=\left(\mathbf{I}-\frac{\tau}{2}\mathbf{W}\left(t+\frac{\tau}{2}\right)\right)^{-1}\left(\mathbf{I}+\frac{\tau}{2}\mathbf{W}\left(t+\frac{\tau}{2}\right)\right),$$

or more explicitly

$$\Theta\left(t+\frac{\tau}{2}\right)=\frac{\mathbf{I}\left(1-\frac{\tau^2}{4}\omega^2\left(t+\frac{\tau}{2}\right)\right)-\tau\mathbf{W}\left(t+\frac{\tau}{2}\right)+\frac{\tau^2}{2}\left(\omega\otimes\omega\right)\left(t+\frac{\tau}{2}\right)}{1+\frac{\tau^2}{4}\omega^2\left(t+\frac{\tau}{2}\right)},$$

where $\omega^2\left(t+\frac{\tau}{2}\right)=(\omega\cdot\omega)\left(t+\frac{\tau}{2}\right)$.

3.4 Rigid body discretization and coupling with SPH

The rigid body is discretized into N_b points with a regular distance between them. In this work we set the distance between rigid body points to be equal to h. A position of each point is denoted by $r_i(t)=(r_{i,1}(t),r_{i,2}(t),r_{i,3}(t))$ at a given time t. To make the rigid body dynamics calculation easier, we configure the calculation of the rigid body to be in a principal frame. Let $\check{\mathbf{J}}$ be the moment of inertia tensor of the rigid body. By assuming the density of the rigid body is uniform and equals to ρ_b, components of $\check{\mathbf{J}}$ are

$$\begin{aligned}\check{J}_{11}&=\rho_b\sum_{i=1}^{N_b}\left(r_{i,2}^2+r_{i,3}^2\right)\,dx, &\quad \check{J}_{12}=\check{J}_{21}&=-\rho_b\sum_{i=1}^{N_b}r_{i,1}r_{i,2}\,dx,\\ \check{J}_{22}&=\rho_b\sum_{i=1}^{N_b}\left(r_{i,1}^2+r_{i,3}^2\right)\,dx, &\quad \check{J}_{13}=\check{J}_{31}&=-\rho_b\sum_{i=1}^{N_b}r_{i,1}r_{i,3}\,dx,\\ \check{J}_{33}&=\rho_b\sum_{i=1}^{N_b}\left(r_{i,1}^2+r_{i,2}^2\right)\,dx, &\quad \check{J}_{23}=\check{J}_{32}&=-\rho_b\sum_{i=1}^{N_b}r_{i,2}r_{i,3}\,dx.\end{aligned}\tag{3.20}$$

We calculate the *principal* moment of inertia tensor $\mathbf{J}$ of the rigid body by doing an eigendecomposition of the moment of inertia tensor $\check{\mathbf{J}}$. Eigenvalues of $\check{\mathbf{J}}$ are diagonal components of $\mathbf{J}$, while their corresponding eigenvectors construct an orthogonal rotation matrix $\mathbf{R}(0)$.

$$\check{\mathbf{J}} = \Lambda \mathbf{J} \Lambda^{-1}, \qquad \mathbf{R}(0) = \Lambda. \tag{3.21}$$

Similarly to fluid points, we keep track of the physical quantities associated with rigid body points as well. Rigid body points also have a density function which we set to be equal to a reference density ρ_0. Mass and volume of rigid body points are set to match our SPH configuration described in (3.6) and (3.7). Note that the density of rigid body points is a quantity completely different from the density of the rigid body ρ_b.

After the discretization process, interactions between fluid points and rigid body points are handled in the exact same way as with interactions between fluid points. Fluid points see the rigid body points as exactly the *same objects as other fluid points and do not discriminate interactions between them*, on both the density update step and the acceleration update step.

As mentioned before in the introduction, in this work we consider a purely hydrodynamics force for interactions between rigid body points and fluid points. Based on (3.11), since the density of a rigid body point is equal to the density reference yielding zero pressure for a rigid body point, the total force applied to rigid body point r_i is

$$f_i = -m_i c_V \left(\left(\sum_{j=1}^{N} m_j \frac{p_j}{\rho_j^2} \nabla \psi_h(r_i - r_j) \right) + b_i \right), \tag{3.22}$$

where $c_V = \frac{h^3}{V(E_i)}$. The constant c_V is needed since there is a discrepancy on an SPH volume calculation $V(E_i)$ compared to the volume of the cube with length of the edge h occupied by each rigid body point. Note that the sum in (3.22) ranges only over fluid points. That is, we do not consider interactions between rigid body points.

To calculate a linear movement of the rigid body, from (2.3), we have

$$A(t) = \frac{1}{M} \left(\left(\sum_{i=1}^{N_b} f_i(t) \right) + F_{c1} + F_{c2} \right), \tag{3.23}$$

where M is the mass of the rigid body, $A(t)$ is a linear acceleration of the rigid body, $f_i(t)$ is the total force applied to a rigid body point r_i, F_{c1} and F_{c2} are forces from the ODE control given at a contact point 1 and contact point 2, respectively. Updating a linear velocity and a position of the center of mass of the rigid body can be done using the leapfrog time integrator scheme in a similar fashion described in (3.13)–(3.16).

Next, to calculate a rotation movement of the rigid body, we need to sum total moments of force acting on the rigid body,

$$K(t) = \left(\sum_{i=1}^{N_b} (r_i(t) - X(t)) \times f_i(t) \right) + (r_{c1}(t) - X(t)) \times F_{c1} + (r_{c2}(t) - X(t)) \times F_{c2}, \tag{3.24}$$

where $K(t)$ and $X(t)$ are the total moment of force acting on the rigid body and the position of the rigid body at a given time t, respectively, r_{c1} and r_{c2} are a position of a contact point 1 and contact point 2, respectively. We use Euler's equation of rigid body dynamics (2.6) and the iterative scheme (3.18) to update the angular velocity of the rigid body. Then we update the rotation matrix of the rigid body by using (3.19). The position of the rigid body point is updated based on its relative position to the center of mass of the rigid body as

$$r_i(t+\tau) = X(t+\tau) + \mathbf{R}(t+\tau)\mathbf{R}^T(t)\,(r_i(t) - X(t)), \tag{3.25}$$

where τ is the timestep. The velocity of a rigid body point is updated based on a linear velocity and an angular velocity of the rigid body

$$u_i(t+\tau) = U\left(t+\frac{\tau}{2}\right) + \frac{\tau}{2}A(t) + \omega(t+\tau) \times (r_i(t+\tau) - X(t+\tau)), \tag{3.26}$$

where $u_i(t)$ is a velocity of a rigid body point r_i at a given time t, $U(t)$ and $\omega(t)$ are respectively a linear velocity and an angular velocity of the rigid body at a given time t.

3.5 Algorithm of the simulation

The simulation is started by an initialization of all parameters needed for the simulation and initial configuration for all SPH points as discretized representations of fluid and rigid body. The main loop of the simulation is started with updating the density, pressure, and hydrodynamics forces-based acceleration of all SPH points. The forces acting on rigid body points are used to update a linear movement of the rigid body. Then the ODE control updates the contact forces needed to control the rigid body. Then all forces including the contact forces are used to calculate the rotational movement of the rigid body. Next, we update the velocity and position of the rigid body points based on the evolution of the rigid body. Finally, we update the velocity and position of the fluid points, and go back to the starting point of the main loop. The full algorithm of the simulation can be seen in Figure 3.

4 Simulation results and discussions

In this work, we use the frame of reference which moves together with the wave, assuming the wave moves with constant velocity $V_f = 2.5$. The inclination angle of the domain (ϕ in Figure 1) is set to be $\phi = \frac{\pi}{18}$, causing the gravity in our simulation frame to be slanted to $-z$-axis. Gravity acts as an external body force per unit mass is set to be $(0,\ -9.81\ \cos\phi,\ -9.81\ \sin\phi)$. The size of the domain is set to be $1 \times 0.6 \times 1.6$ in x, y, z axis, respectively. The system has a periodic boundary condition in x-axis, non-zero Dirichlet boundary on some parts of the left boundary by using ghost points (rendered with light blue color in Figure 4(a)) for $-0.3 \leq y < -0.12$. The bottom boundary is also set to be a non-zero Dirichlet boundary by using a non-moving boundary points (rendered with dark blue color in Figure 4(a)) for $-0.8 \leq z < 0.56$, and free boundary on the right boundary, top boundary, the left boundary for $-0.12 \leq y < 0.3$, and the bottom boundary for $0.56 \leq z \leq 0.8$. Note that the origin is located at the center of the domain. The depth of the fluid is 0.18. The size of the rigid body is $0.2 \times 0.06 \times 0.8$. The rigid body is represented by red points in Figure 4(a). Initially, the center of mass of rigid body is positioned at $(0, -0.04, -0.27)$.

The density reference ρ_0 is set to be $\rho_0 = 1000$, while the density of the rigid body is $\rho_b = 100$. The fluid is initialized to have an initial velocity $u_i(0) = V_f = 2.5$ toward $+z$-axis, and initial density to be $\rho_i(0) = \rho_0$ for all fluid points r_i. We use the piecewise cubic kernel (3.3) as the mollifier function and initialize the points on a regular grid with a distance h along each axis, $V(E_i)$ and m_i for all points are $V(E_i) = 8 \times 10^{-6}$ and $m_i = 0.008$. We set the parameter of kernel function $h = 0.02$. Time step size is set to be $\tau = 0.0005$ with speed of sound chosen to be $c = 20$. W from (2.11) and (2.12) is set to $W = 10$. We choose σ and θ_m of (2.13) to be $\sigma = 10$ and $\theta_m = -0.05$. The positions of contact point 1 and contact point 2 are $r_{c1} = (0.,\ -0.03,\ -0.4)$ and $r_{c2} = (0.,\ -0.03,\ 0.4)$, respectively, relative to the position of center of mass of the rigid body.

Reynold's number for our configuration, using the real value of the dynamic viscosity of the

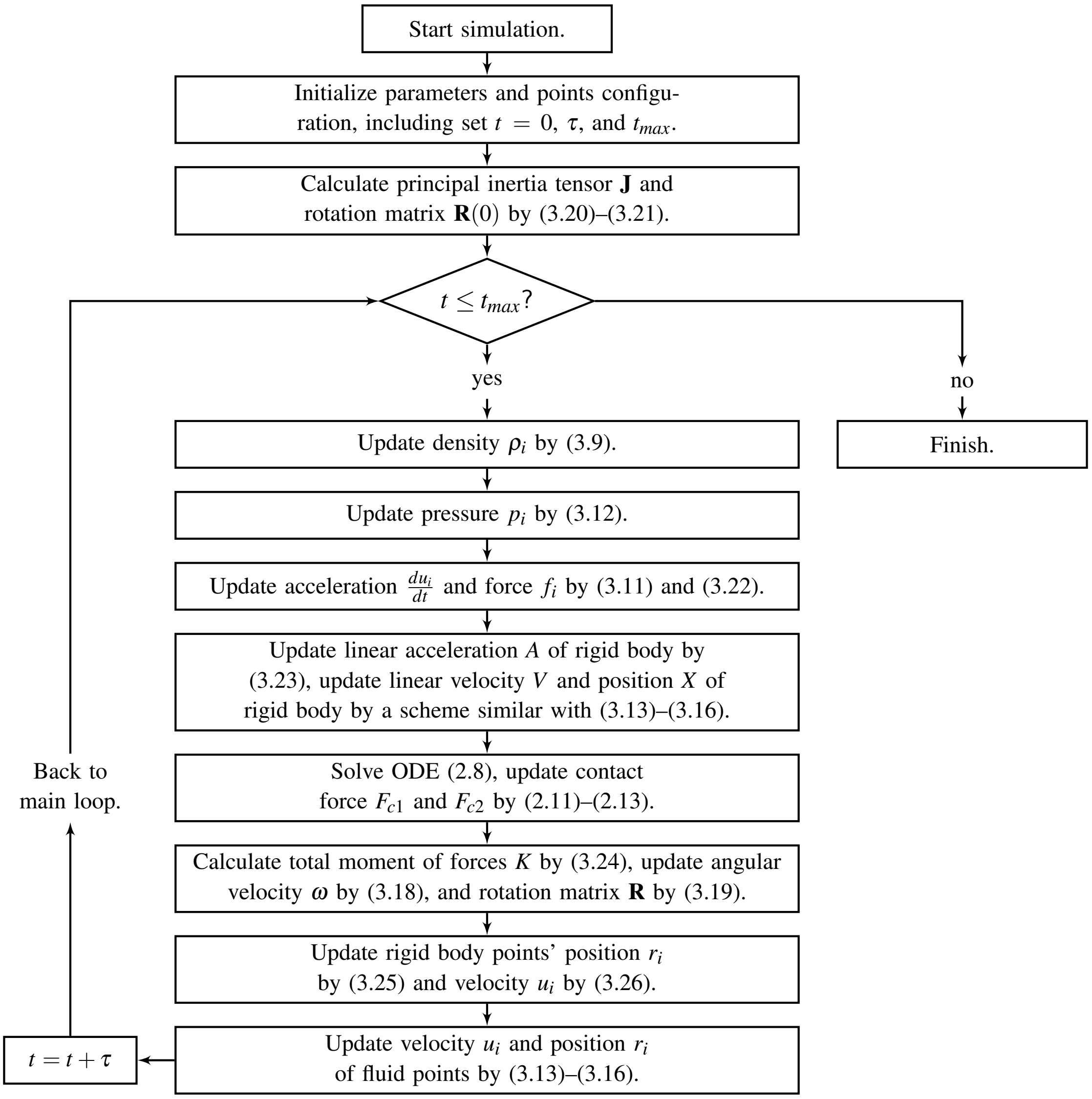

Start simulation.
Initialize parameters and points configuration, including set $t = 0$, τ, and t_{max}.
Calculate principal inertia tensor $\mathbf{J}$ and rotation matrix $\mathbf{R}(0)$ by (3.20)–(3.21).
$t \leq t_{max}$?
yes
no
Finish.
Update density ρ_i by (3.9).
Update pressure p_i by (3.12).
Update acceleration $\frac{du_i}{dt}$ and force f_i by (3.11) and (3.22).
Update linear acceleration A of rigid body by (3.23), update linear velocity V and position X of rigid body by a scheme similar with (3.13)–(3.16).
Solve ODE (2.8), update contact force F_{c1} and F_{c2} by (2.11)–(2.13).
Calculate total moment of forces K by (3.24), update angular velocity ω by (3.18), and rotation matrix $\mathbf{R}$ by (3.19).
Update rigid body points' position r_i by (3.25) and velocity u_i by (3.26).
Update velocity u_i and position r_i of fluid points by (3.13)–(3.16).
$t = t + \tau$
Back to main loop.

Figure 3: Algorithm of the simulation.

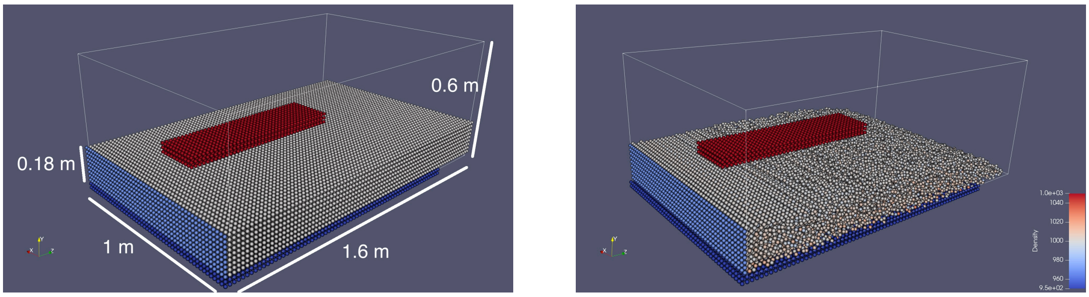
water from [21], is

$$Re = \frac{\rho_0 V_f L}{\nu} = \frac{1000 \times 2.5 \times 0.2}{0.001} = 5 \times 10^5,$$

where ν is the real dynamic viscosity of the water at 20 degree Celsius, and we take the characteristic length L to be the width of the rigid body. It is clear that the viscosity force is negligible compared to the inertial force of the fluid. Hence, in this work we can neglect the viscosity and assume the fluid to be inviscid.

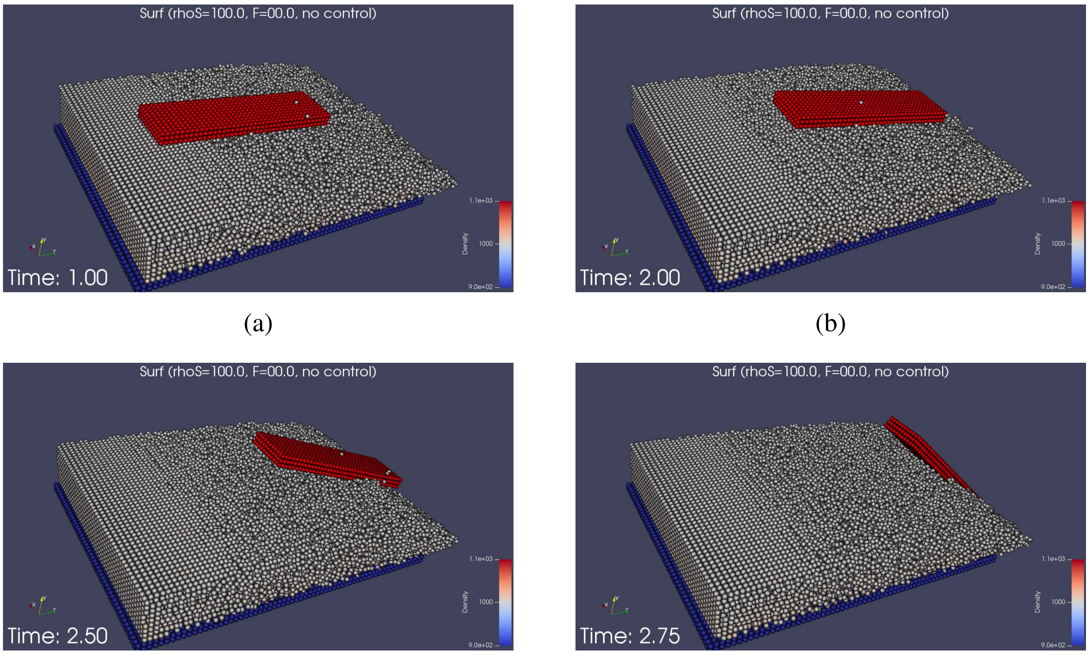
The free boundary condition is implemented by changing the type of any fluid points leaving the domain into a ghost point whose velocity does not change with time and whose density always equals to the reference density ρ_0, but still interacts with other points. If a ghost point reenters the domain, it is marked as a normal fluid point again. But if it leaves the domain farther than h, the point is removed from the simulation.

Before we run the actual simulation, we run the "relaxation" process to stabilize the flow of the water up to $t = 1.5$. The initial condition after relaxation can be seen on Figure 4(b).



(a) (b)

Figure 4: (a) The initial configuration of the system, and (b) the initial condition after relaxation process.


Surf (rhoS=100.0, F=00.0, no control)
Time: 1.00
Surf (rhoS=100.0, F=00.0, no control)
Time: 2.00
Surf (rhoS=100.0, F=00.0, no control)
Time: 2.50
Surf (rhoS=100.0, F=00.0, no control)
Time: 2.75
1.1e+03
1000
9.0e+02
Density

(a) (b)

(c) (d)

Figure 5: Snapshots of the surfing board simulation without an ODE controller at (a) $t = 1.00\ s$, (b) $t = 2.00\ s$, (c) $t = 2.50\ s$, and (d) $t = 2.75\ s$.

First, we try to run a simulation without an ODE control. We can see in Figure 5 that the board cannot maintain its position and drifts away with the flow of the fluid.

To find the best $\tilde{\theta}$ and μ, for each $\tilde{Z} \in \{-0.6, -0.5, -0.4, -0.3, -0.2\}$ we try each combination of $\tilde{\theta} \in \{-0.05, -0.06, -0.07, -0.08, -0.09, -0.1\}$ and $\mu \in \{1, 2, 5, 10, 20, 50\}$. For each simulation, we take the data of the position of the center of mass of the rigid body for each time step to assess the quality of the corresponding parameters configuration. Let $Z(\mu, \tilde{\theta}, \tilde{Z}, m\tau)$ be the third component of the position of the center of mass of the rigid body at a time step $m\tau$ for a simulation with a parameters configuration μ, $\tilde{\theta}$, and $\tilde{Z}$, with $m \in \{0, 1, \ldots, \Upsilon\}$. To assess the chosen parameters configuration, we take an average of the difference between $Z(\mu, \tilde{\theta}, \tilde{Z}, m\tau)$ and

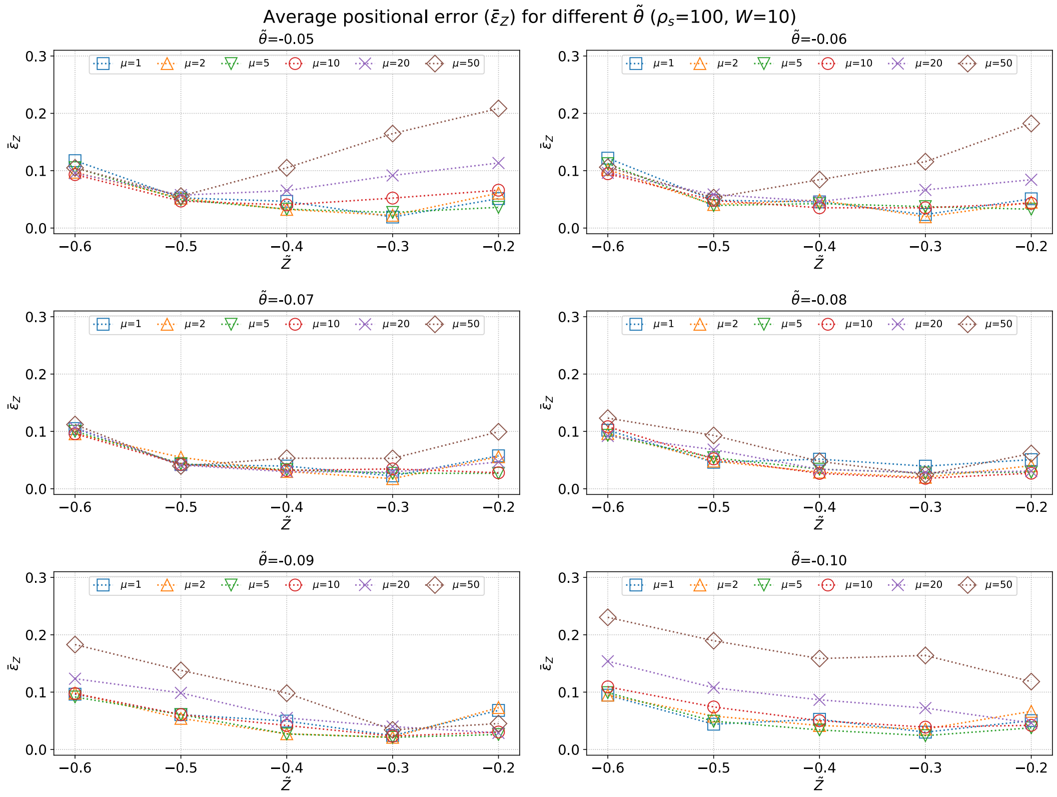


Figure 6: The graphs of the average positional error for different parameters $\tilde{\theta}$, μ, and $\tilde{Z}$.

a corresponding $\tilde{Z}$ for the whole simulation time,

$$\overline{\varepsilon_Z}(\mu,\tilde{\theta},\tilde{Z}) := \frac{\sum_{m=0}^{\Upsilon}\left|Z(\mu,\tilde{\theta},\tilde{Z},m\tau)-\tilde{Z}\right|}{\Upsilon+1}. \tag{4.1}$$

Let us call $\bar{\varepsilon}_Z$ the average positional error. The graphs of $\bar{\varepsilon}_Z$ can be seen on Figure 6. Then we take an average of the average positional error for each $\tilde{Z}$,

$$\overline{\overline{\varepsilon_Z}}(\mu,\tilde{\theta}) := \frac{\sum_{\tilde{Z}\in\tilde{Z}_{\text{set}}}\overline{\varepsilon_Z}(\mu,\tilde{\theta},\tilde{Z})}{\#\tilde{Z}_{\text{set}}}, \qquad \tilde{Z}_{\text{set}} = \{-0.6,-0.5,-0.4,-0.3,-0.2\}, \tag{4.2}$$

and find cumulative errors for each $\tilde{\theta}$ and μ. The average of average positional error and its cumulative errors for each $\tilde{\theta}$ and μ can be seen in Table 1.

From Table 1 we can see that $\tilde{\theta} = -0.07$ and $\mu = 5$ give the smallest cumulative errors for $\tilde{\theta}$ and μ, respectively. Now let us see more in detail the simulation results for $\tilde{\theta} = -0.07$ in Figure 7.

As we can see in Figure 7, smaller values of μ give more oscillations to the position compared to larger μ values. But as the larger value of μ dampens the amplitude and frequency of oscillation to the position, it also shifts the stable position of surfing board. That problem also occurs when we solve the ODE (2.8) directly by using an ODE solver which we can see in Figure 8.

The direct ODE solver is used to observe the behavior of the solution under different parameters choices. The parameters a and b of the ODE control in (2.8) depend on μ. Note again that we do not know the actual value μ of our system. Hence, we carry out trials and errors by varying the value of μ in our simulation, yielding a and b that differ from the correct ones. By using the

Table 1: Table of the average of average positional error $\overline{\overline{\varepsilon_Z}}$ and its cumulative errors for each $\tilde{\theta}$ and μ.

$\mu \backslash \tilde{\theta}$	-0.05	-0.06	-0.07	-0.08	-0.09	-0.10	C.E.
1	5.74e-02	5.81e-02	5.33e-02	5.80e-02	5.98e-02	5.44e-02	3.41e-01
2	5.30e-02	5.13e-02	5.08e-02	4.67e-02	5.48e-02	5.94e-02	3.16e-01
5	5.02e-02	5.29e-02	4.62e-02	4.76e-02	4.53e-02	4.89e-02	2.91e-01
10	5.97e-02	5.13e-02	4.66e-02	4.65e-02	5.08e-02	6.29e-02	3.18e-01
20	8.47e-02	7.04e-02	5.07e-02	5.07e-02	6.92e-02	9.33e-02	4.19e-01
50	1.28e-01	1.08e-01	7.15e-02	7.00e-02	9.97e-02	1.72e-01	6.49e-01
C.E.	4.33e-01	3.92e-01	3.19e-01	3.20e-01	3.80e-01	4.91e-01	

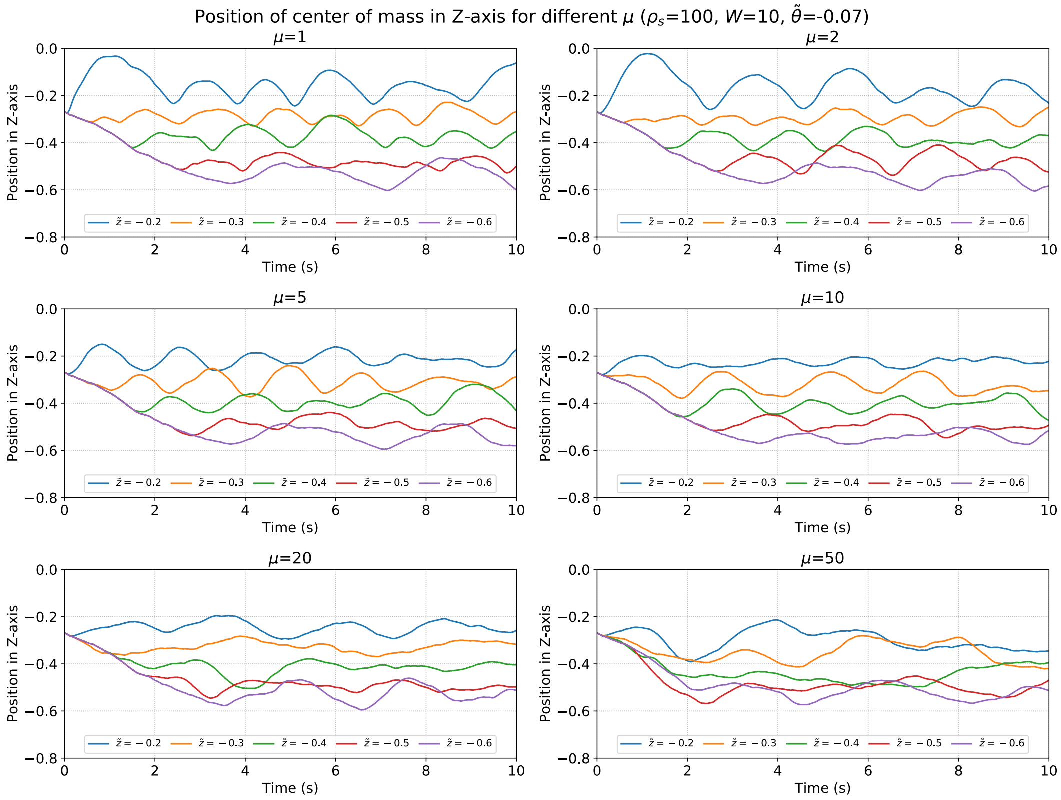


Figure 7: The z-axis-component position of surfing board for $\tilde{\theta} = -0.07$.

direct ODE solver we want to see the effect of the choice of μ in simulation that does not match the actual μ of the system. In the direct ODE solver we choose $\mu_{\text{actual}} = 20$ and use various μ, effectively varying the values of a and b. We choose $\tilde{Z} = -0.5$. From Figure 8 we can see that as the value of μ increases, the oscillation decreases in both amplitude and frequency. But the increase of μ also shifts the stable position farther from the desired position $\tilde{Z} = -0.5$, justifying the same behavior that occurs in our simulation. Adding an additional ODE control for $\tilde{\theta}$ might be the solution for this problem. $\tilde{\theta}$ acts as a "target" inclination angle in our current ODE control. By controlling $\tilde{\theta}$, it is possible to disturb the stability of the system when the board is not located at the desired position, forcing the board to nudge slowly to the desired position.

Notice that in the direct ODE solver, the value of μ affects the oscillation of θ and V instead

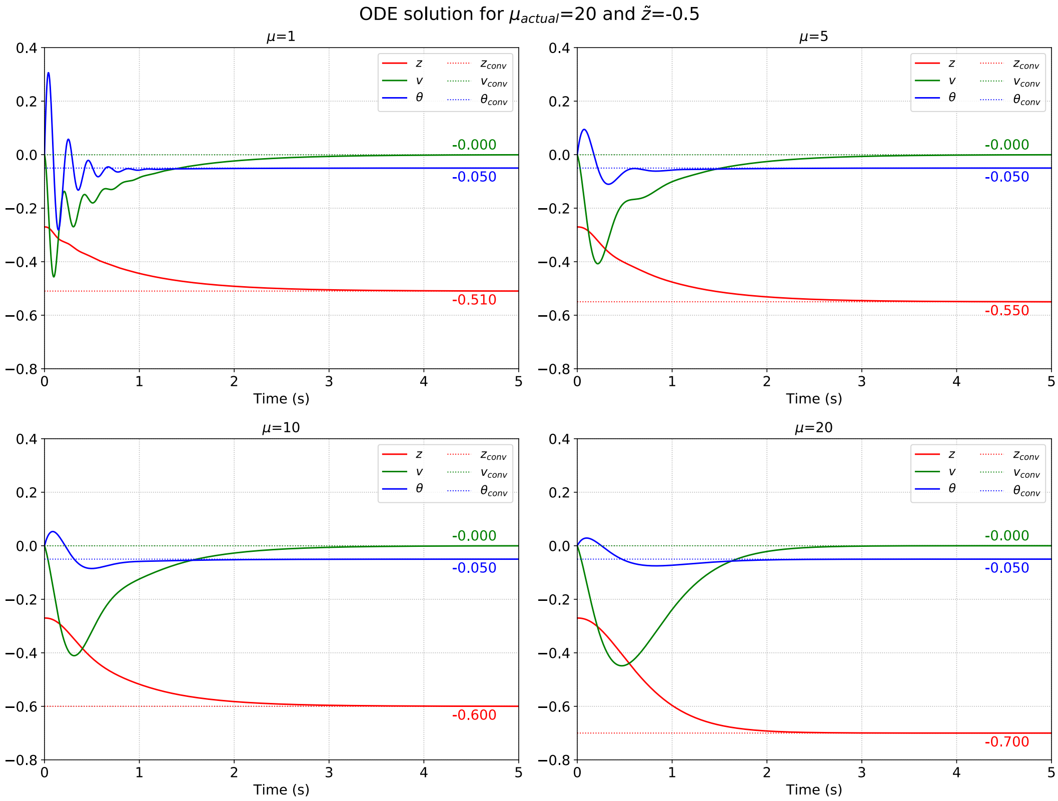


Figure 8: Solution of the ODE system for different μ.

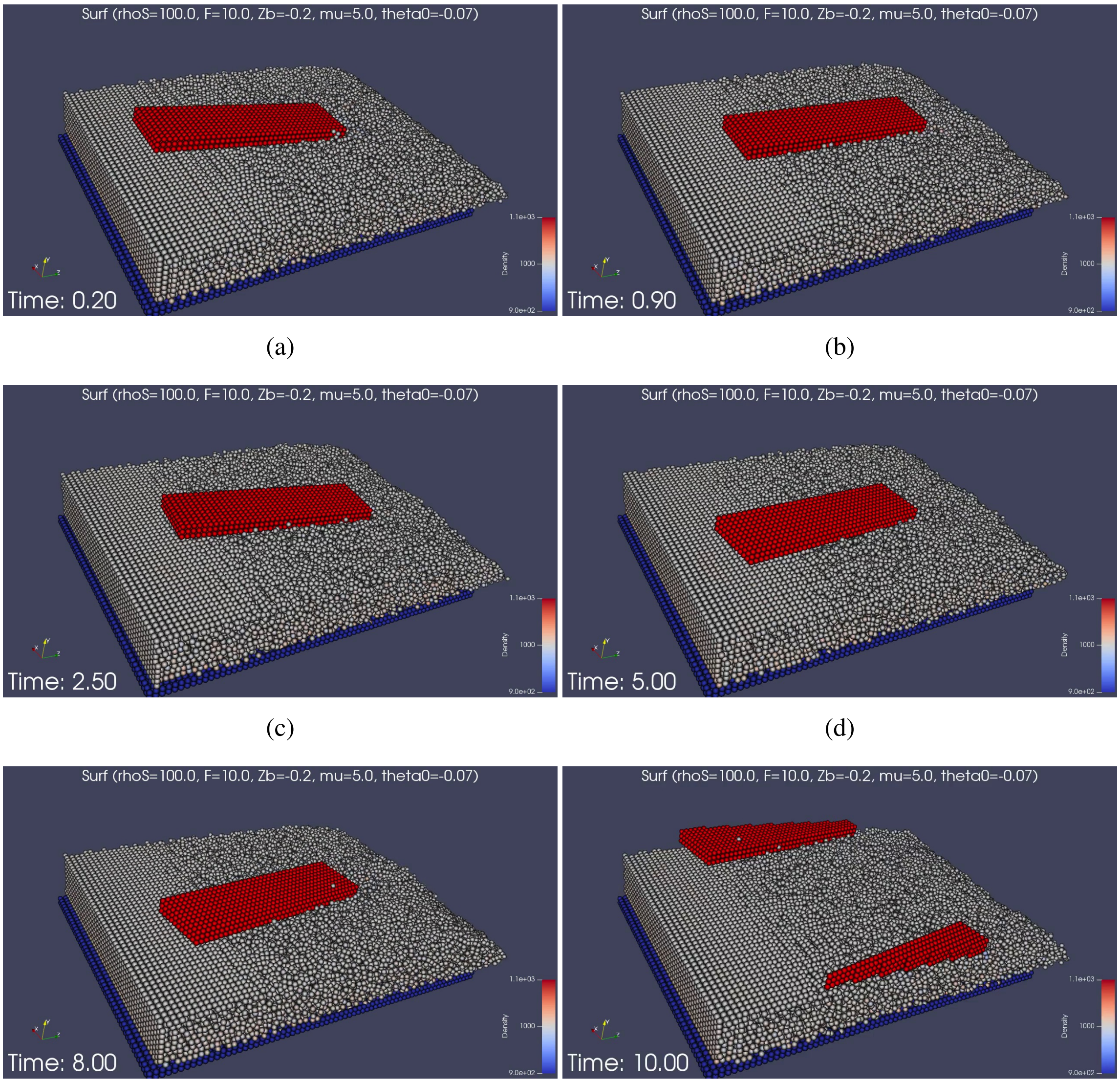
of θ and Z in our SPH simulations. This happens because our SPH simulation cannot translate the change of the inclination angle into the change of the velocity fast enough. The delayed response in velocity propagates the oscillation in the inclination angle to the position in SPH simulations. By tweaking σ from (2.13) it is possible to transfer the change of the inclination angle θ from the ODE control to the inclination angle of the board in the SPH simulation $\hat{\theta}$ faster, which makes a faster change in the velocity and helps the surfing board stabilizes faster.

The oscillation in Figure 7 also occurs since we set μ_v and μ_z from (2.8) to be zero, ignoring the dependency of acceleration to the position and velocity. In fact, this is not correct since the drag force depends not only on the shape of the interface between the fluid and the rigid body, but also on the relative velocity between both of them. Initially we assume that the flow has a constant velocity through the whole domain on z-axis of our frame. But, in reality, the velocity of the flow depends on the position, as the gravity slows the velocity of the flow as a consequence of our choice of the frame.

Snapshots of a simulation with the ODE control can be seen on Figure 9.

5 Summary

An ODE control was successfully implemented into a coupled inviscid fluid-rigid body SPH simulation in an attempt to control the movement of the surfing board. For our system, the best values for $\tilde{\theta}$ and μ are $\tilde{\theta} = -0.07$ and $\mu = 5$. Although $\mu = 5$ gave the best result, it does not mean the actual μ of our system is equal to 5. As shown in the result of the direct ODE solver, as the μ



(a) (b) (c) (d) (e) (f)

Figure 9: Snapshots of the surfing board simulation with an ODE controller with parameters $\tilde{Z} = -0.2$, $\mu = 5$, and $\tilde{\theta} = -0.07$ at (a) $t = 0.20\ s$, (b) $t = 0.90\ s$, (c) $t = 2.50\ s$, (d) $t = 5.00\ s$, (e) $t = 8.00\ s$, and (f) $t = 10.00\ s$.

increases, the stable position shifts farther from the desired position. This problem can be solved by adding an additional ODE control for $\tilde{\theta}$ that can nudge the board toward the desired position. By tweaking values of μ_z, μ_v and σ, it is possible to make the system control the surfing board more accurately and swiftly. In this work we neglected the effect of viscosity since the inertial force is dominating the viscous force.

Acknowledgement

This work is a part of doctoral research of the author. The author would like to thank his Ph.D. supervisor Professor Seiro Omata and his colleague Professor Norbert Pozar for their invaluable supports, advices, and moral supports. This work cannot be done without helps from both of them. The author would like to thank the Ministry of Education, Culture, Sports, Science, and

Technology (MEXT) for their financial support.

References

[1] Lucy, L. B.: A numerical approach to the testing of the fission hypothesis, Astro. J. **82** no. 12 (1977), 1013–1024.

[2] Gingold, R. A. and Monaghan, J. J.: Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. Roy. Astron. Soc. **181** (1977), 375–389.

[3] Monaghan J. J.: Shock simulation by the particle method SPH, J. Compu. Phys. **52** (1983), 374–389.

[4] Monaghan, J. J.: SPH and Riemann solvers, J. Compu. Phys. **136** (1997), 298–307.

[5] Clear, P. W. and Monaghan, J. J.: Conduction modelling using smoothed particle hydrodynamics, J. Compu. Phys. **148** (1999), 227–264.

[6] Colagrossi, A. and Landrini, M.: Numerical simulation of interfacial flows by smoothed particle hydrodynamics, J. Compu. Phys. **191** (2003), 448–475.

[7] Lo, E. Y. M. and Shao S.: Simulation of near-shore solitary wave mechanics by an incompressible SPH method, Appl. Ocean Res. **24** (2002), 275–286.

[8] Dalrymple, R. A. and Rogers, B. D.: Numerical modelling of water waves with SPH method, Coast. Eng. **53** (2006), 141–147.

[9] Monaghan, J. J., Kos, A. and Issa, N.: Fluid motion generated by impact, J. Waterw. Port Coast. **129** (2003), 250–259.

[10] Oh, S. , Kim, Y. and Roh, B.: Impulse-based rigid body interaction in SPH, Comp. Anim. Virtual Worlds **20** (2009), 215–224.

[11] Li, Y. and Asai, M.: Fluid-rigid body interaction simulation based on a stabilized ISPH method incorporated with the impulse-based rigid body dynamics, Trans. Jap. Soc. Comput. Eng. Sci. **2018** Issue 2 (2018), Article No. 20182010.

[12] Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B. and Teschner, M.: Versatile rigid-fluid coupling for incompressible SPH, ACM Trans. Graph. **31** Issue 4 (2012), Article No. 62.

[13] Wendt, John F.: Computational Fluid Dynamics: An Introduction, Springer, 2009.

[14] Rao, Anil V.: Dynamics of Particles and Rigid Bodies: A Systematic Approach, Cambridge, 2006.

[15] Folland, G. B.: Real analysis, 2nd ed., Pure and Applied Mathematics, John Wiley & Sons, Inc., New York, 1999.

[16] Monaghan, J. J. and Lattanzio, J. C.: A refined particle method for astrophysical problem, Astron. Astrophys. **149** (1985), 135–143.

[17] Batchelor, G. K.: An Introduction to Fluid Dynamics, Cambridge, 1967.

[18] Monaghan, J. J.: Simulating free surface flows with SPH, J. Comput. Phys. **110** (1994), 399–406.

[19] Monaghan, J. J.: Smoothed particle hydrodynamics, Rep. Prog. Phys. **68** (2005), 1703–1759.

[20] Omelyan, Igor P. : Algorithm for numerical integration of the rigid-body equations of motion, Phys. Rev. E **58** no.1 (1998), 1169–1172.

[21] Kestin, J., Sokolov, M. and Wakeham, W. A. : Viscosity of liquid water in the range -8^{o}C to 150^{o}C, J. Phys. Chem. Ref. Data **7** no.3 (1978), 941–948.