V International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2013
S. Idelsohn, M. Papadrakakis and B. Schrefler (Eds)

APPLICATIONS OF AN INCOMPRESSIBLE FLUID-RIGID BODY INTERACTION ON PROGRESSIVE MOVING-GRID FINITE-VOLUME METHOD

Shinichi ASAO*[*] and Kenichi MATSUNO[†]

[*]Depertment of Mechanical Engineering,
College of Industrial Technology,
1-27-1, Nishikoya, Amagasaki, Hyogo, 661-0047, Japan
e-mail: asao@cit.sangitan.ac.jp, web page: http://www.sangitan.ac.jp

[†]Department of Mechanical System Engineering,
Kyoto Institute of Technology,
Matsugasaki, Sakyo-ku, Kyoto, 606-8585, Japan
e-mail: matsuno@kit.ac.jp - Web page: http://www.kit.ac.jp

Key words: Incompressible Flow, Moving Mesh, Finite Volume Method, Interaction Simulation

Abstract. The purpose of this paper is to introduce the Trans-mesh method and the Moving computational domain method as a progressive Moving-grid finite-volume method. In the Trans-mesh method, the bodies can move freely in a main mesh that covers the entire flow field. On the other hand, in the Moving Computational Domain method, the whole of the computational domain including bodies inside moves in the physical space without the limit of region size. These methods are constructed based on the four-dimensional control volume in space-time unified domain such that the method assures to be divergence-free in the space-time unified domain and thus satisfies both the physical and geometrical conservation laws simultaneously. The methods are applied to a falling sphere by gravity in an infinite long bending pipe and a trajectory of a flying ball over ground in incompressible fluid. The results indicate that these methods are promising in simulating the interaction of incompressible fluid-rigid body.

1 Introduction

Today, one of the interesting problems in CFD is a moving boundary problem. Especially, in the case of fluid-bodies interaction is interesting on engineering. For this problem, the body-fitted coordinate is usually applied. When multiple bodies move a long distance in the flow field, a body-fitted coordinate system hardly adapt the mesh to the motion of the bodies. This decreases an efficiency and accuracy. To overcome this problem, we have proposed a Trans-mesh Method [1]. In the Trans-mesh method, a body moves freely in a stationary mesh while mesh planes are added, removed, and/or shifted in the main mesh satisfying both physical and geometric conservation laws. Next, we consider a flow around a sphere in the long pipe. It is necessary to make the computational mesh for whole of the pipe. Then, a huge number of computational mesh is needed. As the result, this simulation by using traditional method spends a lot of time. We have proposed the Moving Computational Domain method [2]. This method is kind of moving mesh method. The feature of this method is to move computational domain with the body. The Moving Computational Domain method can consider the region without limit. The only necessary assumption is that the conditions just in front of the computational domain should be known a priori, such as, stationary fluid state or uniform flow and so on. As these flow solvers, we modified the Moving-grid Finite-volume method [3]. The method is constructed based on the four-dimensional control volume in space-time (x, y, z, t) unified domain such that the method satisfies the divergence-free character in the (x, y, z, t) space and both the physical and geometrical conservation laws simultaneously [4]. Due to the use of four-dimensional control volume, the method has a lot of merits or freedom. The purpose of this paper is to introduce applications using the Trans-mesh method and Moving Computational Domain method. The methods are applied to a falling sphere by gravity in an infinite long bending pipe and a trajectory of a flying ball over ground in incompressible fluid.

2 Trans-mesh Method and Moving Computational Domain Method

2.1 Governing equations

The governing equations are the continuity equation and the incompressible Navier-Stokes equations. These are written as follows:

$$\nabla \cdot \boldsymbol{q} = 0, \tag{1}$$

$$\begin{aligned}\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{E}_a}{\partial x} + \frac{\partial \boldsymbol{F}_a}{\partial y} + \frac{\partial \boldsymbol{G}_a}{\partial z} = &-\left(\frac{\partial \boldsymbol{E}_p}{\partial x} + \frac{\partial \boldsymbol{F}_p}{\partial y} + \frac{\partial \boldsymbol{G}_p}{\partial z}\right) \\ &+\left(\frac{\partial \boldsymbol{E}_v}{\partial x} + \frac{\partial \boldsymbol{F}_v}{\partial y} + \frac{\partial \boldsymbol{G}_v}{\partial z}\right).\end{aligned} \tag{2}$$

Here $\boldsymbol{q}$ is the velocity vector, $\boldsymbol{E}_a$, $\boldsymbol{F}_a$, and $\boldsymbol{G}_a$ are advection flux vectors in the x, y, and z directions, respectively, $\boldsymbol{E}_v$, $\boldsymbol{F}_v$, and $\boldsymbol{G}_v$ are viscous flux vectors, and $\boldsymbol{E}_p$, $\boldsymbol{F}_p$, and $\boldsymbol{G}_p$

are pressure flux vectors. The elements of the velocity vector and flux vectors are:

$$
\boldsymbol{q} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \boldsymbol{E}_a = \begin{bmatrix} u^2 \\ uv \\ uw \end{bmatrix}, \quad \boldsymbol{F}_a = \begin{bmatrix} uv \\ v^2 \\ vw \end{bmatrix}, \quad \boldsymbol{G}_a = \begin{bmatrix} uw \\ vw \\ w^2 \end{bmatrix},
$$
$$
\boldsymbol{E}_v = \frac{1}{\mathrm{Re}} \begin{bmatrix} u_x \\ v_x \\ w_x \end{bmatrix}, \quad \boldsymbol{F}_v = \frac{1}{\mathrm{Re}} \begin{bmatrix} u_y \\ v_y \\ w_y \end{bmatrix}, \quad \boldsymbol{G}_v = \frac{1}{\mathrm{Re}} \begin{bmatrix} u_z \\ v_z \\ w_z \end{bmatrix},
$$
$$
\boldsymbol{E}_p = \begin{bmatrix} p \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{F}_p = \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix}, \quad \boldsymbol{G}_p = \begin{bmatrix} 0 \\ 0 \\ p \end{bmatrix}. \tag{3}
$$

Where u, v, and w are the velocity component of the x, y, and z directions respectively, p is the pressure, and Re is the Reynolds number. The subscripts x, y, and z indicate derivatives with respect to x, y, and z, respectively.

As for motion of the body, six degrees of freedom is assumed, and the combined motion of the translation and rotation of the body is considered. The rigid body equations of motion are written as follows:

$$
\frac{\mathrm{d}\boldsymbol{p}_\mathrm{B}}{\mathrm{d}t} = \boldsymbol{f}_\mathrm{B}, \tag{4}
$$
$$
\frac{\mathrm{d}\boldsymbol{L}_\mathrm{B}}{\mathrm{d}t} = \boldsymbol{N}_\mathrm{B}. \tag{5}
$$

Here, $\boldsymbol{p}_\mathrm{B}$ is the momentum vector of the body, $\boldsymbol{f}_\mathrm{B}$ is the external force vector, $\boldsymbol{L}_\mathrm{B}$ is the angular momentum vector, and $\boldsymbol{N}_\mathrm{B}$ is the torque vector, respectively.

2.2 Moving-grid Finite-volume method

To assure the geometric conservation laws, we adopt a control volume in the space-time unified domain (x, y, z, t), which is four-dimensional in the case of three-dimensional flows. Now, Eq. (2) can be written in divergence form as,

$$
\tilde{\nabla} \cdot \tilde{\mathcal{F}} = \boldsymbol{0} \tag{6}
$$

where

$$
\tilde{\nabla} = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} & \frac{\partial}{\partial t} \end{bmatrix}^T, \quad \tilde{\mathcal{F}} = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{F} & \boldsymbol{G} & \boldsymbol{q} \end{bmatrix}^T,
$$
$$
\boldsymbol{E} = \boldsymbol{E}_a - \boldsymbol{E}_v + \boldsymbol{E}_p, \quad \boldsymbol{F} = \boldsymbol{F}_a - \boldsymbol{F}_v + \boldsymbol{F}_p, \quad \boldsymbol{G} = \boldsymbol{G}_a - \boldsymbol{G}_v + \boldsymbol{G}_p. \tag{7}
$$

The present method is based on a cell-centered finite-volume method and, thus, the flow variables are defined at the center of the cell in the (x, y, z) space. The control

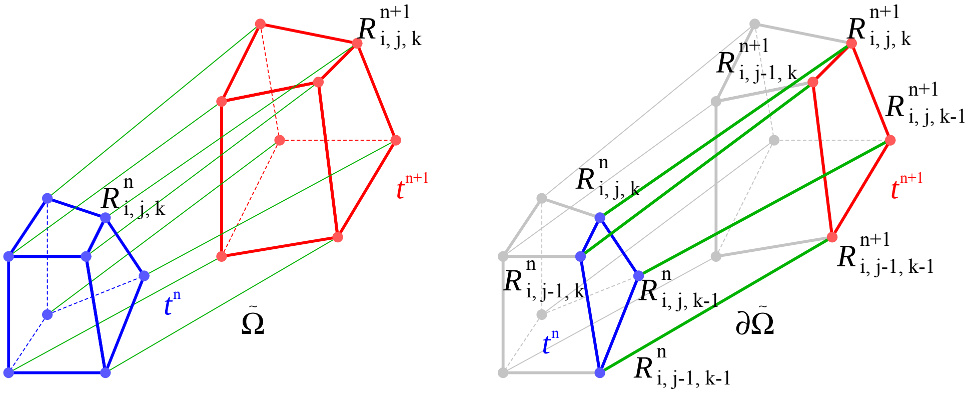


Figure 1: Schematic view of control volume in (x, y, z, t) space-time unified domain.

volume becomes a four-dimensional polyhedron in the (x, y, z, t)-domain, as schematically illustrated in Fig. 1.

We apply volume integration to Eq. (2) with respect to the control volume illustrated in Fig. 1. By using the Gauss theorem, we can write Eq. (6) in surface integral form as,

$$\int_{\tilde{\Omega}} \tilde{\nabla} \cdot \tilde{\mathcal{F}} \mathrm{d}\tilde{\Omega} = \oint_{\partial\tilde{\Omega}} \tilde{\mathcal{F}} \cdot \tilde{\boldsymbol{k}} \mathrm{d}(\partial\tilde{\Omega}) \approx \sum_{l=1}^{8} \left(\tilde{\mathcal{F}} \cdot \tilde{\boldsymbol{n}}\right)_l = \mathbf{0}. \tag{8}$$

Here, $\tilde{\boldsymbol{k}}$ is an outward unit vector normal to the surface, $\partial\tilde{\Omega}$, of the octahedron control volume $\tilde{\Omega}$, in the space-time unified four-dimensional domain, and, $\tilde{\boldsymbol{n}}_l = (\tilde{n}_x, \tilde{n}_y, \tilde{n}_z, \tilde{n}_t)_l$, $(l = 1,2,\ldots\ 8)$ denotes the surface normal vector of the control volume and its length equal to the boundary surface area in four-dimensional (x, y, z, t) space. The upper and lower boundaries of the control volume ($l = 7$ and 8) are perpendicular to the t-axis, and, therefore, they only have an $\tilde{n}_t$ component, and its length corresponds to the volume of the cell in the (x, y, z)-space at times t^n and t^{n+1} respectively. Thus, Eq. (8) can be expressed as,

$$\boldsymbol{q}^{n+1}(\tilde{n}_t)_8 + \boldsymbol{q}^{n}(\tilde{n}_t)_7 + \sum_{l=1}^{6} \left(\tilde{\mathcal{F}}^{n+1/2} \cdot \tilde{\boldsymbol{n}}\right)_l = \mathbf{0}. \tag{9}$$

2.3 Concept of Trans-mesh method

In the Trans-mesh method, a body moves freely in a stationary mesh while mesh planes are added, removed, and/or shifted in the main mesh, as illustrated in Fig. 2. The front mesh plane of the moving body is eliminated from the main mesh in order to prevent the mesh from folding because of the reduced mesh spacing that occurs due to the movement of the body. Moreover, a new mesh plane is added between the rear plane of the moving body and the main mesh in order to maintain the maximum allowable mesh spacing. At the same time, the cells existing and connecting between the main mesh and the side of the moving body are skewed due to the movement of the body. Thus, reconnection or exchange of mesh lines between the moving body and the main mesh is necessary. Hence, the present method essentially includes three inevitable procedures: addition and elimination of mesh planes as well as changes in the connecting relationships of structured mesh lines.

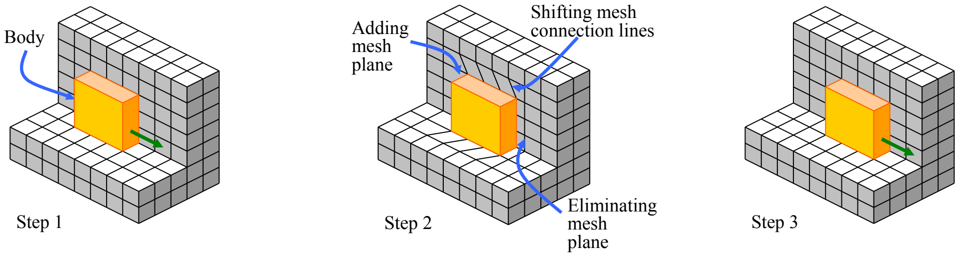


Figure 2: Concept of Trans-mesh method.

2.4 Concept of Moving Computational Domain method

The basic coordinate system of the Moving Computational Domain method is the general, fixed, stationary (x, y, z) Cartesian coordinate system. The computational domain itself, including the body inside, moves in the fixed (x, y, z)-space. The flow around the body is calculated as the moving boundary problem. Unknown flow variables, such as pressure p, x-directional velocity u and so on, are defined at each grid cell center in the computational domain. The motion of the computational domain according to the motion of the body in the physical space is arbitrary, and thus the any kind of the motion of the body can be simulated by the Moving Computational Domain method. The flow field driven by the body is calculated in the computational domain in which the body fitted mesh system is used. Since the computational domain itself moves in the physical (x, y, z) space time-dependently and thus the mesh system of the computational domain also moves in the (x, y, z) space, the flow solver has to be constructed for the moving grid system. In the present Moving Computational Domain method, the Moving-Grid Finite-Volume method [3] is adopted. Only necessary and essential assumption is that the condition in front of the moving computational domain has to be known because it is necessary as a boundary condition of the flow solver. The natural assumption may be the stationary fluid condition in front of the moving computational domain.

2.5 Numerical method

To solve the discretization equation of equation (2) using the Trans-mesh method and the Moving Computational Domain method, we apply the SMAC method [5]. Thus, this equation can be solved in three stages. The equation to be solved at the first stage contains the unknown variables at $(n+1)$-time step in the flux terms. Thus, the equation is iteratively solved using the LU-SGS method [6]. The equation to be solved at the

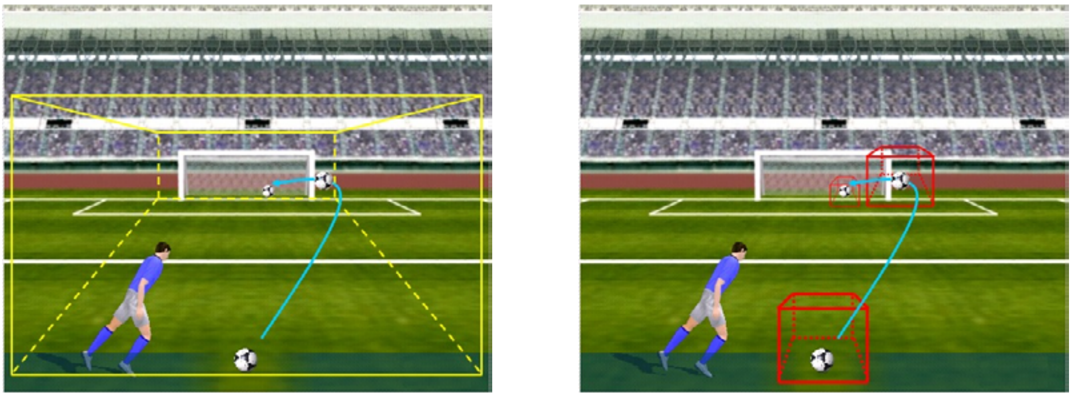


Figure 3: Conventional method (left) and Moving Computational Domain method (right).

second stage is the Poisson equation about the pressure correction. This equation is iteratively solved using the Bi-CGSTAB method [7]. The incompressible fluid-rigid body interaction with collision is calculated in the first step of the SMAC method. Figure 4 shows the flowchart of the fluid-body interaction. First, we calculate the force applied to the body. This force includes the force of collision. The collision force is evaluated based on Glowinski's method[8]. The Transmission Mesh method does not require special treatment when collisions happen. Next, the equation of motion for the body is calculated, and the mesh is moved. Then, the Navier-Stokes equations are calculated. If the physical amounts converge, we proceed to the second step of the SMAC method.

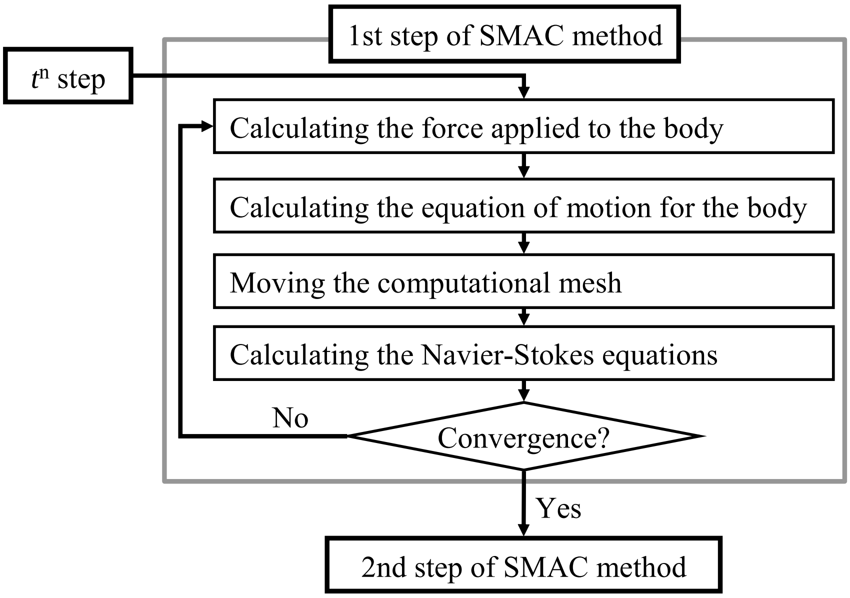

1st step of SMAC method
t^n step
Calculating the force applied to the body
Calculating the equation of motion for the body
Moving the computational mesh
Calculating the Navier-Stokes equations
No
Convergence?
Yes
2nd step of SMAC method

Figure 4: Flowchart for calculation of interaction.

3 Numerical Results

3.1 Falling sphere by gravity in the infinite long bending pipe

As an application of the present methods, the movement of a sphere falling by gravity in the infinite long bending pipe is investigated, as illustrated in Fig. 5.

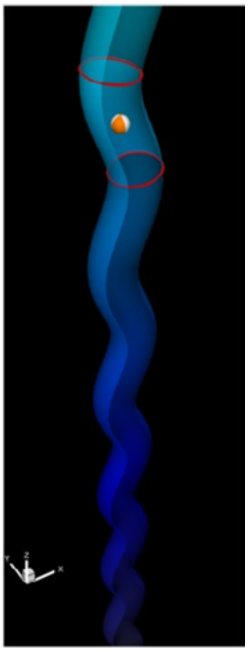


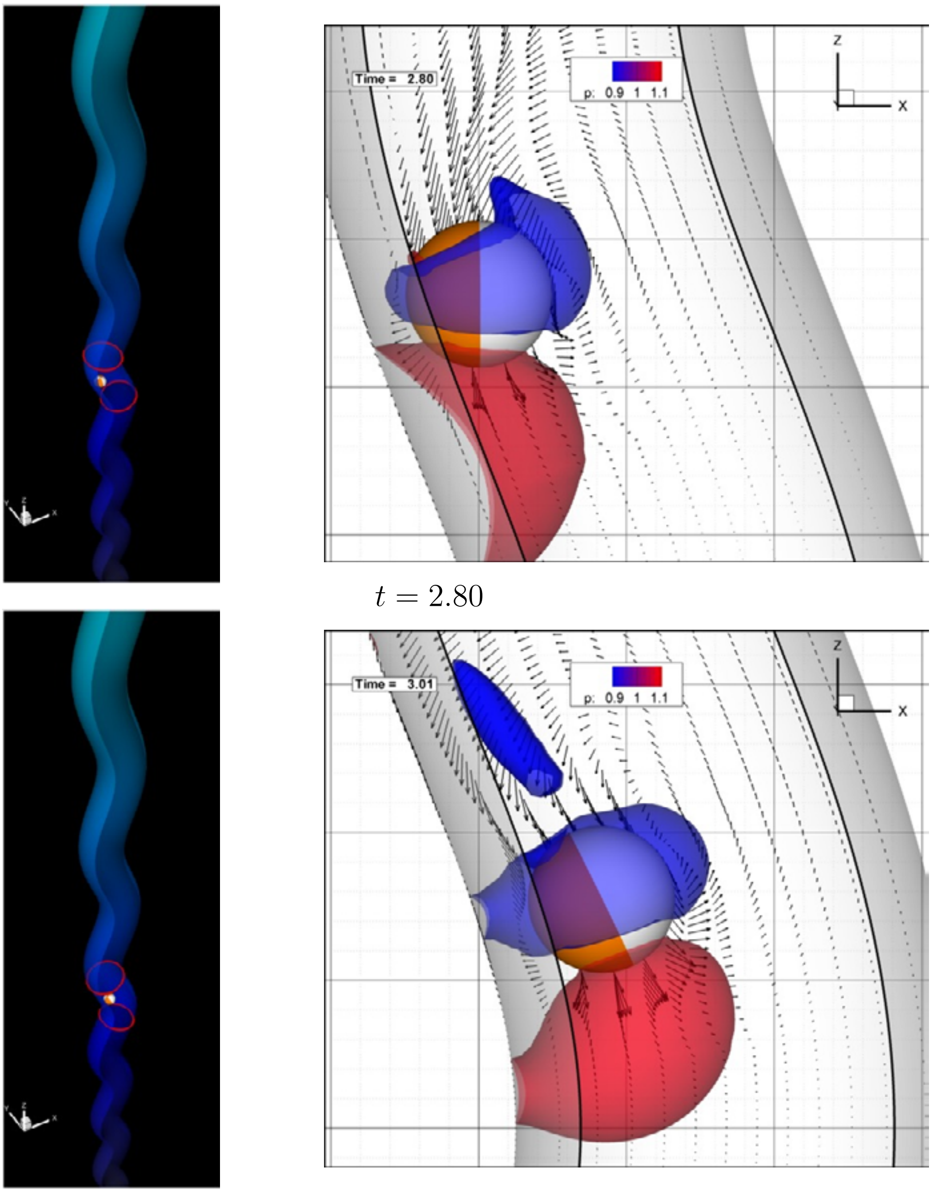
Figure 5: The moving computational domain for falling sphere in a bending pipe.

The diameter of the pipe is $3.0d$, where d denotes the diameter of the sphere. The wavelength and amplitude of the pipe are $12d$ and $0.75d$, respectively. The moving computational domain is red domain in the pipe shown as Fig. 5. The initial position of the sphere is in the center of the moving computational domain. The ratio between body and fluid density (ρ_b/ρ_f) is 1.167. The initial stationary condition of pressure, velocity components in the x, y, z directions are given by $p = 1.0$, $u = v = w = 0.0$. The Reynolds number is 300 based on the diameter of the sphere d, the terminal speed of the sphere w_∞, and the kinematic viscosity. A mesh system of $52 \times 52 \times 104$ is adopted. The time step size is $\Delta t = 0.01$. Table 1 shows setup of physical quantity.

Table 1: Setup of physical quantity

d [mm]	g [m/s^2]	ρ_f [kg/m^3]	μ_f [Ns/m^2]	w_∞ [m/s]	Re
15	9.81	960	10×10^{-3}	0.220	300

As a result, Fig. 6 shows the pressure distribution and velocity vectors on central plane of y as well as the sphere position at the time $t = 2.80$ and 3.01. The sphere moves along the wall of the pipe after the sphere collides against the wall. It is confirmed that the pressure in the front of the sphere is always higher than the pressure in the rear.

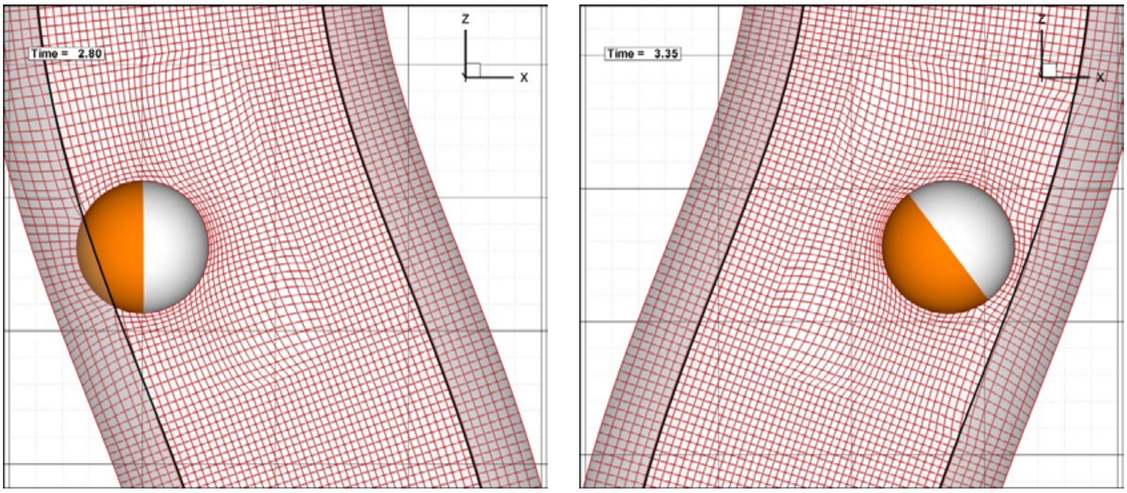


$t = 2.80$

$t = 3.01$

Figure 6: Overall view, pressure distribution, and velocity vectors on central plane of y.

Figure 7 shows the computational mesh near the sphere on central plane of y. We confirmed that the computational mesh holds its shape well even if the computational mesh and the sphere move in any direction.



$t = 2.80$ $t = 3.35$

Figure 7: Computational mesh near the sphere at $t = 2.80$ and 3.35.

3.2 A trajectory of a flying sphere over ground in incompressible fluid

A flying sphere over ground in incompressible fluid, as illustrated in Fig. 8 is presented.

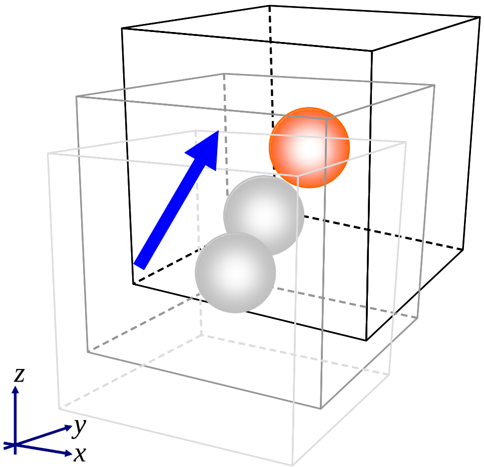


Figure 8: The flying ball over ground in incompressible fluid.

The size of the moving computational domain is $L_x = 21.0d$, $L_y = 21.0d$, $L_z = 21.0d$, where d denotes the diameter of the sphere. The initial stationary condition of pressure, velocity components in the x, y, z directions are given by $p = 1.0$, $u = v = w = 0.0$. Table 2 shows setup of physical quantity. The size and mass of the sphere are the same as a soccer ball. It is assumed that the sphere is a rigid body. The stitching of the soccer ball is not considered. The Reynolds number is 3000 based on the diameter of the sphere d, the initial speed of the sphere, and the kinematic viscosity. A mesh system of $81 \times 81 \times 81$

is adopted. The time step size is $\Delta t = 0.01$.

Table 2: Setup of physical quantity.

Diameter of ball	0.22 m
Mass of ball	0.43 kg
Initial velocity	10.0 m/s
Initial angle	45.0°

As a result, Fig. 9 shows the pressure distribution and velocity vectors on central plane of y as well as the sphere position at the time $t = 0.28\,\mathrm{s}, 0.55\,\mathrm{s}, 0.82\,\mathrm{s}, 1.10\,\mathrm{s}$, and $1.38\,\mathrm{s}$. Figure 10 shows the trajectory of the sphere and angular velocity of the sphere where ω_x, ω_y, and ω_z are the angular velocity component of the x, y, and z axis of rotation, respectively. The sphere is accelerated at $t = 0.22\,\mathrm{s}$. The sphere rises with a negative angular velocity of x. Next, pressure distribution in a rear of the sphere changes from a symmetric shape to dissymmetric shape as time goes by. The movement in the y direction is not confirmed because the Reynolds number is low and a viscosity is high. The sphere begins to fall at $t = 0.8\,\mathrm{s}$ and the magnitude of the angular velocity becomes small. It is confirmed that the sphere collides into the ground at $t = 1.3\,\mathrm{s}$.

4 Concluding Remarks

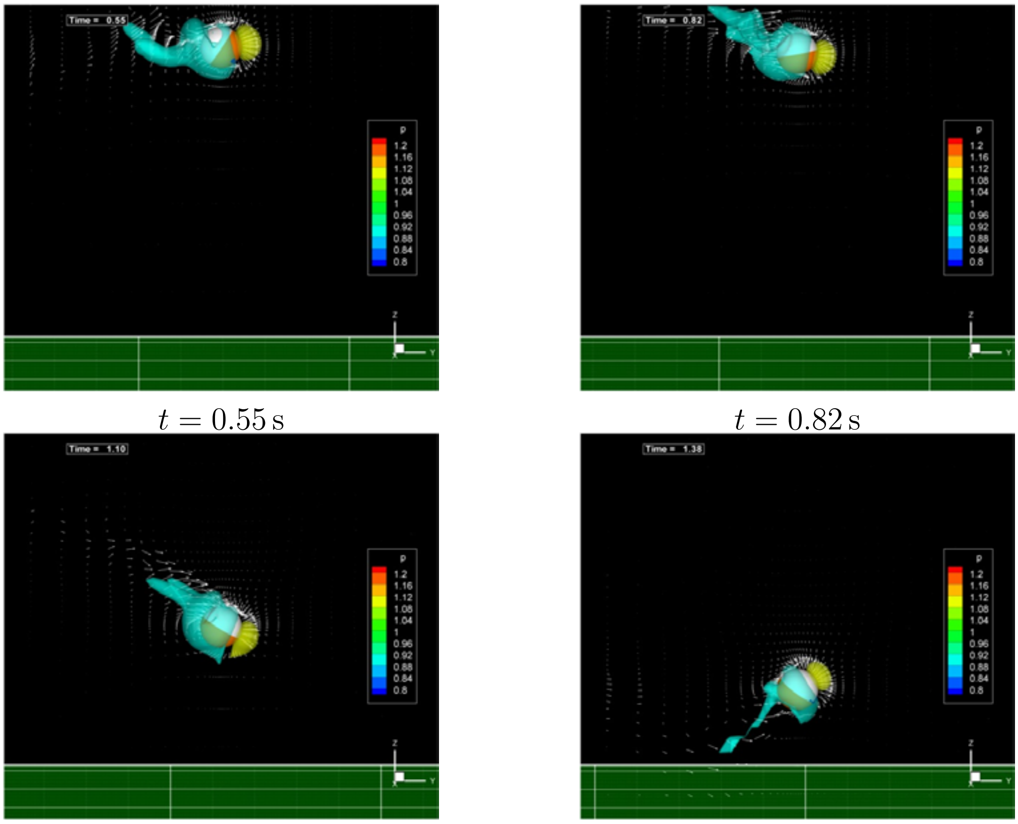
In this paper we introduced the applications using Trans-mesh method and the Moving Computational Domain method and described its application to a falling sphere by gravity in an infinite long bending pipe and a trajectory of a flying ball over ground in incompressible fluid. The results indicated that physically meaningful flows were obtained and the computational mesh holds its shape well even if the computational mesh and the sphere move in any direction. Therefore, we confirmed that the Trans-mesh method and the Moving Computational Domain method are useful for simulating the interaction of incompressible fluid-rigid body.

Acknowledgment

This work was supported by the Science Research Promotion Fund (2010) from the Promotion And Mutual Aid Corporation for Private Schools of Japan and the Grant-in-Aid for Scientific Research (21560175).

REFERENCES

[1] ASAO, S., Matsuno, K., and Yamakawa, M., Trans-Mesh Method and Its Application to Simulations of Incompressible Fluid-Rigid Bodies Interaction, *Journal of Computational Science and Technology*, Vol. 5, pp. 163–174, (2011).



$t = 0.55\,\mathrm{s}$ $t = 0.82\,\mathrm{s}$

$t = 1.10\,\mathrm{s}$ $t = 1.38\,\mathrm{s}$

Figure 9: Pressure distribution and velocity vectors on central plane of y at $t = 0.28$s, 0.55s, 0.82s, 1.10s, and 1.38s.

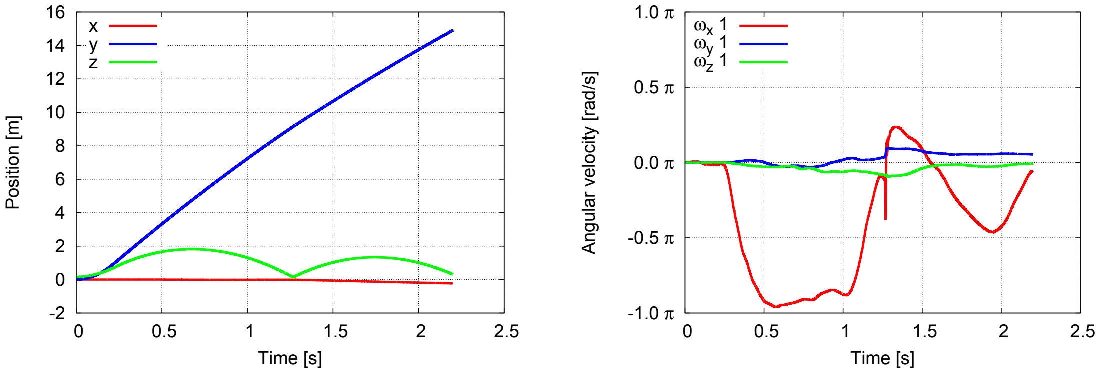


Figure 10: Trajectory of the sphere and angular velocity of the sphere.

[2] Watanabe, K. and Matsuno, K., Moving computational domain method and its application to flow around a high-speed car passing through a hairpin curve, *Journal of Computational Science and Technology*, Vol. 3, pp. 449-459, (2009).

[3] Matsuno, K., Development and Applications of a Moving Grid Finite Volume Method, *Developments and Applications in Engineering Computational Technology*, Saxe-Coburg Publications, Chapter 5, pp. 103–129 (2010).

[4] Zhang, H., Reggio, M., Trepanier, J. and Camarero, R., Discrete Form of the GCL for Moving Meshes and Its Implimentation in CFD Schemes, *Computers & Fluids*, Vol. 22, pp. 9–23 (1993).

[5] Amsden, A. and Harlow, F., A Simplified MAC Technique for Incompressible Fluid Flow Calculations, *Journal of Computational Physics*, Vol. 6, pp. 322–325 (1970).

[6] Yoon, S. and Jameson., A., Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations, *AIAA Journal*, Vol. 26, pp. 1025–1026 (1988).

[7] VAN DER VORST, H. A., Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, *SIAM Journal on Scientific Computing*, Vol. 13, No. 2, pp. 631–644 (1992).

[8] R. Glowinski, T. -W. Pan, T. I. Hesla, D.D.Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, *International Journal of Multiphase Flow*, Vol. 25, pp. 755-794 (1999).