

Moving computational multi-domain method for modelling the flow interaction of multiple moving objects

Momoha Nishimura (✉ d7821007@edu.kit.ac.jp)

Kyoto Kogei Sen'i Daigaku Kogei Kagakubu <https://orcid.org/0000-0002-7727-4566>

Masashi Yamakawa

Kyoto Institute of Technology: Kyoto Kogei Sen'i Daigaku

Shinichi Asao

College of Industrial Technology: Sangyo Gijutsu Tanki Daigaku

Seiichi Takeuchi

College of Industrial Technology: Sangyo Gijutsu Tanki Daigaku

Mehdi Badri Ghomizad

Kyoto Institute of Technology: Kyoto Kogei Sen'i Daigaku

Research

Keywords: Crossing, Moving boundary problem, Compressible flow, Moving mesh method

Posted Date: October 5th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-907888/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Advances in Aerodynamics on January 24th, 2022. See the published version at <https://doi.org/10.1186/s42774-021-00099-2>.

RESEARCH

Moving computational multi-domain method for modelling the flow interaction of multiple moving objects

Momoha Nishimura^{1*}, Masashi Yamakawa¹, Shinichi Asao², Seiichi Takeuchi² and Mehdi Badri Ghomizad¹

*Correspondence:

d7821007@edu.kit.ac.jp

¹Department of Mechanical and System Engineering, Kyoto Institute of Technology, Kyoto, Japan

Full list of author information is available at the end of the article

Abstract

This study proposes a method where the flow field variables are communicated between multiple separate moving computational domains and simulates the flow interaction of multiple moving objects. Instead of using the conventional approach with a single fixed computational domain covering the whole flow field, this method advances the moving computational domain (MCD) method in which the computational domain itself moves in line with the motions of an object inside. The computational domains created around each object move independently, and the flow fields of each domain interact where the flows cross. This eliminates the spatial restriction for simulating multiple moving objects. After the results of the shock tube test verify that the interpolation has been achieved between grids, a validation test is conducted in which two spheres are crossed, and the forces exerted on one object due to the other's crossing at a short distance are calculated. The results verify the reliability of this method and show that it is applicable to the flow interaction of multiple moving objects.

Keywords: Crossing; Moving boundary problem; Compressible flow; Moving mesh method

1 Introduction

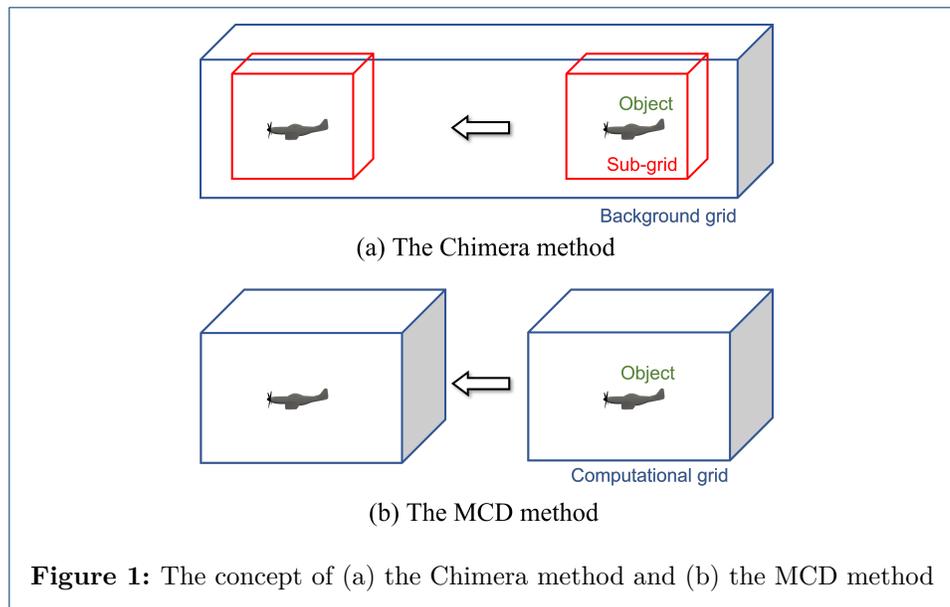
The need for computational fluid dynamics (CFD) has increased along with technological developments and the expansion of the transportation industry. At the same time, CFD itself has also developed. Therefore, the scope of applications in which CFD can be employed has expanded. Accordingly, in addition to modelling static objects through a conventional simulation approach, a numerical representation of moving and deforming objects is also needed. However, simulations for dynamic objects should not only include generally static objects with moving elements (e.g. windmills, turbines in power plants) but also objects that move in a large space (e.g. bicycles, automobiles, aeroplanes). This means that CFD requires novel methods for simulating the movement of objects in a large space. Moreover, since aerial vehicles, such as drones or flying cars, have been a focus of global attention, the interaction of flows between multiple moving objects has become of interest to improve the control accuracy. Other than that, given the current dramatic developments for the space mission, the shock wave interaction between the flows around multiple launch vehicles requires consideration. For future technologies which are developed for use in the sky, simulating only one object is insufficient. CFD must consider the flow on

such a group level, both in terms of aerodynamic efficiency and safety. Thus, a reasonable and practical computational technique that considers the flow interaction associated with multiple moving objects also needs to be developed.

As for the numerical simulation of moving objects, there are several categories of approaches. One category is non-boundary conforming methods, which perform an entire simulation in a Cartesian grid. Since this type of method expresses motions in a Cartesian grid, specific treatments on boundary cells are needed, for example, the immersed boundary method [1, 2]. One of the advantages in using these methods is that they do not require laborious work to generate computational grids: target objects are simply placed in a Cartesian grid. On the other hand, there are disadvantages. These methods are weak in the context of high Reynolds number flows and compressible flows. In addition, since the calculation is conducted in a fixed grid, an extremely large computational domain needs to be prepared when simulating the flow around objects moving in a large space, such as the flight simulations of aeroplanes from take-off to landing.

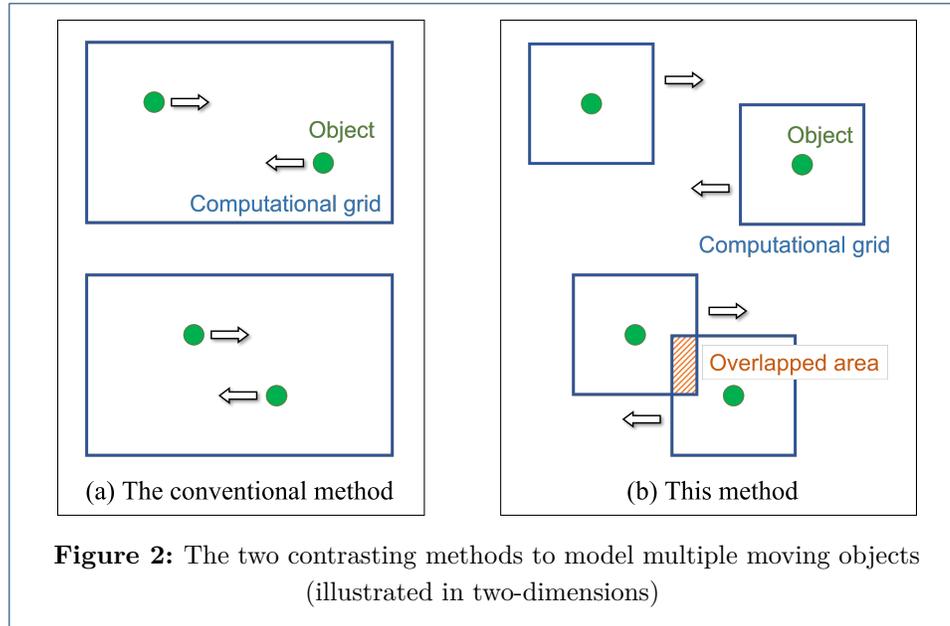
Another approach is the Chimera method [3, 4], which is widely used to represent motions in simulations. This approach makes simulations for moving boundary problems possible by interpolating the flow field variables between one grid and one or more overlapping grids at each time step. With this method, applications such as the separation of a supersonic aeroplane and rocket booster [4] and the manoeuvring of helicopters in a shipborne environment [5] have been reported. However, this method also constrains the size of the computational domain: the overlapping strategy requires a background grid for the interpolation, but the background cannot be set as an infinitely large grid due to the computational cost. Consequently, it is also difficult to simulate objects moving in a large space (Fig. 1a).

To address this difficulty, we previously proposed a unique approach for solving the moving boundary problems, which we named it the moving-grid finite volume (MGFV) method, and its practicality has been demonstrated [6]. The MGFV method automatically meets the geometric conservation law (GCL) condition [7] because it adopts the space and time unified four-dimensional control volume for the discretisation. Since the MGFV method is one of moving mesh methods, objects do not necessarily move in a fixed domain: if the computational domain itself moves in line with the motions of an object inside, the spatial restriction for modelling moving objects can be removed. In other words, it is possible for objects to move in an infinitely large space (Fig. 1b). The method in which a computational domain itself moves was extended to the moving computational domain (MCD) method [8] as an expansion of the MGFV method, and the MCD method has been able to simulate the flow around objects moving in a large space, such as the flight simulation of a tilt-rotor aircraft [9] and a swimmer performing the dolphin kick [10]. However, the MCD method has a downside. Since the computational domain moves, it is difficult to introduce information from outside of it, and currently there is no approach other than substituting values for boundary conditions. For this reason, static fluid is assumed outside the computational domain in the current numerical simulations using the MCD method. Yet for simulations that need to incorporate complex external flow field information (such as turbulent flow, as is often seen in the case of aircrafts, and the flow generated by other objects), the assumption of static surrounding fluid is not valid and leads to unsatisfactory results.



Before discussing the objective of this study, we should mention the current approach and its limitation in simulating the flow interaction of multiple moving objects. Up to now, this has been simulated by placing two objects in a single domain and moving each object (Fig. 2a). Typically, when objects move in a computational grid, cells that are originally almost equilateral become deformed along with the motion and deformation of the objects, which leads to calculation failure. Thus, the crossing of two objects was realised by eliminating the deformed cells and splitting the stretched cells in the grid [11]. However, although this technique was shown to be applicable to simulations with a drastic deformation of unstructured cells, a large domain covering the entire flow field was still necessary. This means that its applications are restricted to only objects moving short distances, while real-life transport vehicles normally travel long distances. In the end, a low-cost simulation with no spatial limit for any movements is required, which is accomplished by preparing computational domains only as big as necessary around individual objects, instead of a large domain containing all the moving objects inside. This can be achieved by using our MCD approach with each domain communicating with each other (Fig 2b).

Therefore, this study proposes a novel method that models multiple moving objects by using computational domains created around individual objects, thus eliminating the redundant areas. When objects perform independent motions, each domain can move independently of the others. When the flow around one object interacts with the flow from the other, the domains are overlapped, and the flow field variables are interpolated between the domains. Once again, this method includes unique features that allow multiple domains to move independently and mutually communicate. This method does not require the entire field where moving objects travel, but each domain created around individual objects is able to move in unlimited directions. The applications are not limited only to objects crossing or overtaking, a full simulation from the beginning to the end of the trajectories



is also feasible. This approach expands the usability of CFD and allows for much more advanced simulations of moving objects.

This paper includes five sections. Section 1 introduced the background and motivation. Section 2 explains the numerical scheme. Section 3 proposes the MCD method and the approach for allowing the communication of multiple domains. The validations are presented in Section 4, which includes a demonstration of crossing two objects as well as a discussion. Finally, Section 5 summarises the study's findings.

2 Numerical scheme

2.1 Governing equations

The three-dimensional Euler equations for compressible flow are applied to the governing equation:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = \mathbf{0},$$

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e + p) \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(e + p) \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \rho w \\ \rho vw \\ \rho w^2 + p \\ w(e + p) \end{bmatrix}, \quad (1)$$

where \mathbf{q} is a vector of conservative variables, and \mathbf{E} , \mathbf{F} , and \mathbf{G} represent the inviscid flux vectors. t , ρ , p , and e represent time, density, pressure, and the total energy per unit mass, respectively, and u , v , w represent the velocity components in the x , y , z coordinates. Eq. 1 is closed with the following equation by considering the ideal gas law:

$$p = (\gamma - 1) \left[e - \frac{1}{2} (u^2 + v^2 + w^2) \right]$$

where γ is the ratio of specific heats. Note that Eq. 2 is used for nondimensionalisation in this study:

$$\begin{aligned} x &= \frac{\hat{x}}{\hat{L}}, y = \frac{\hat{y}}{\hat{L}}, z = \frac{\hat{z}}{\hat{L}}, t = \frac{\hat{t}}{\hat{L}/\hat{c}_\infty}, \rho = \frac{\hat{\rho}}{\hat{\rho}_\infty}, \\ e &= \frac{\hat{e}}{\hat{\rho}_\infty \hat{c}_\infty^2}, p = \frac{\hat{p}}{\hat{\rho}_\infty \hat{c}_\infty^2}, u = \frac{\hat{u}}{\hat{c}_\infty}, v = \frac{\hat{v}}{\hat{c}_\infty}, w = \frac{\hat{w}}{\hat{c}_\infty}, \end{aligned} \quad (2)$$

where, "hat" indicates values with dimensions, and \hat{L} , \hat{c}_∞ , and $\hat{\rho}_\infty$ denote the characteristic length, speed of sound, and characteristic density, respectively.

2.2 Flow solver

Eq. 1 is integrated over the control volume $\tilde{\Omega}$ as follows:

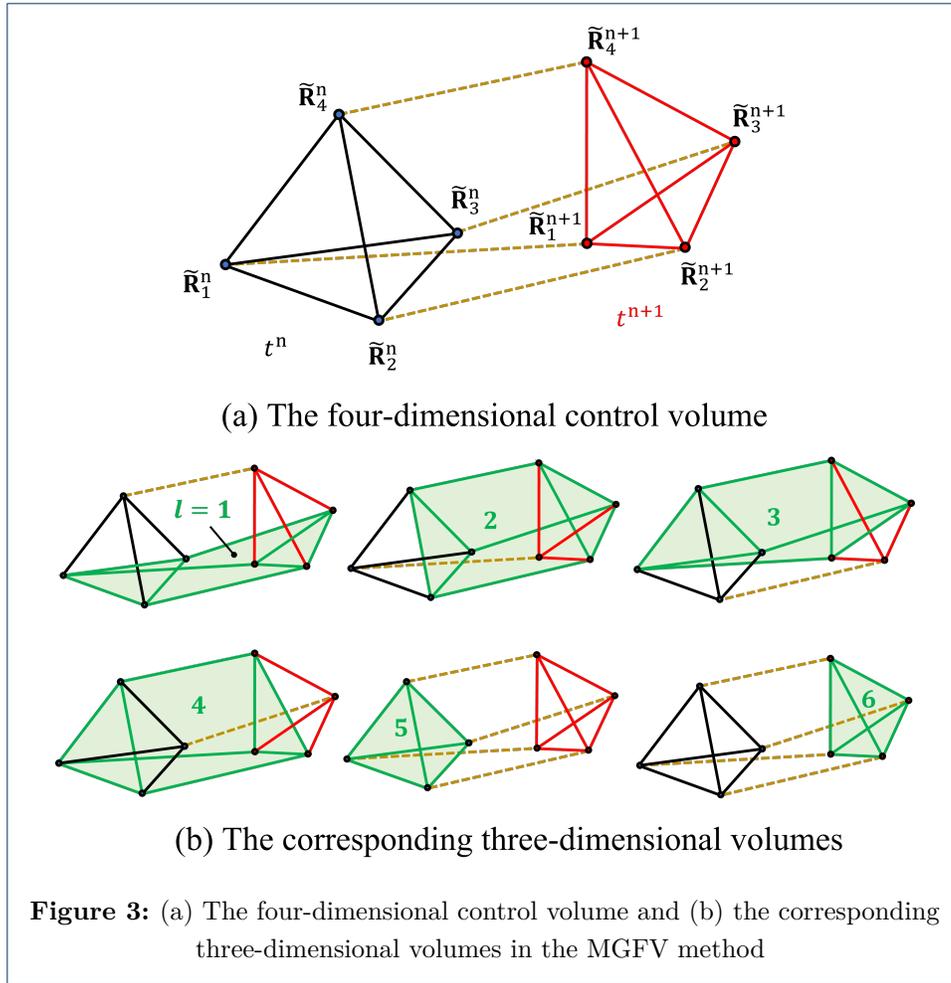
$$\begin{aligned} \int_{\tilde{\Omega}} \tilde{\nabla} \tilde{\mathbf{F}} d\tilde{\Omega} &= 0 \\ \tilde{\nabla} &= \left(\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right), \quad \tilde{\mathbf{F}} = (\mathbf{q}, \mathbf{E}, \mathbf{F}, \mathbf{G}). \end{aligned} \quad (3)$$

In order to solve the governing equations in a moving grid, Eq. 3 is discretised using the unstructured cell-centred approach of the MGFV method. In this method, a four-dimensional space and time unified control volume is used as the control volume. As tetrahedra grids are used in this study, the discrete form of the governing equation is as follows:

$$\int_{\tilde{\Omega}} \tilde{\nabla} \tilde{\mathbf{F}} d\tilde{\Omega} = \int_{\partial \tilde{\Omega}} \tilde{\mathbf{F}} \cdot \tilde{\mathbf{n}} d\tilde{S} = \sum_{l=1}^6 \tilde{\mathbf{F}}_l \cdot \tilde{\mathbf{n}}_l = 0,$$

where $\partial \tilde{\Omega}$ is the three-dimensional volume of the four-dimensional control volume including the time dimension, and $\tilde{\mathbf{n}}$ represents the four-dimensional outward normal vector $\tilde{\mathbf{n}} = [\tilde{n}_t, \tilde{n}_x, \tilde{n}_y, \tilde{n}_z]$. At the three-dimensional volumes $l = 1 - 4$, $\tilde{\mathbf{F}}$ is evaluated by combining the values at the N step and $N + 1$ step, while $\tilde{\mathbf{F}}$ is evaluated only by using the value at the N step at $l = 5$ and by using the value at $N + 1$ the step at $l = 6$, since $l = 5$ represents the tetrahedron at the N step and $l = 6$ represents the tetrahedron at the $N + 1$ step. Each $\tilde{\mathbf{n}}$ of the first to fourth volumes comprises $[\tilde{n}_t, \tilde{n}_x, \tilde{n}_y, \tilde{n}_z]$, while at the fifth and sixth volumes, $\tilde{n}_x, \tilde{n}_y, \tilde{n}_z = 0$, which means $\tilde{\mathbf{n}}_5$ and $\tilde{\mathbf{n}}_6$ only have an \tilde{n}_t component. The details of the MGFV method can be found in [6]. The four-dimensional control volume and the corresponding three-dimensional volumes are depicted in Fig. 3, where $\tilde{\mathbf{R}}$ is a position vector. From the procedure above, Eq. 4 can be obtained:

$$\begin{aligned} \mathbf{q}^{N+1}(\tilde{n}_t)_6 + \mathbf{q}^N(\tilde{n}_t)_5 + \sum_{l=1}^4 \left\{ \mathbf{q}^{N+\frac{1}{2}} \tilde{n}_t + \mathbf{H}^{N+\frac{1}{2}} \right\}_l &= 0 \\ \mathbf{H} &= \mathbf{E}\tilde{n}_x + \mathbf{F}\tilde{n}_y + \mathbf{G}\tilde{n}_z \\ \mathbf{q}^{N+\frac{1}{2}} &= \frac{1}{2} (\mathbf{q}^N + \mathbf{q}^{N+1}), \quad \mathbf{H}^{N+\frac{1}{2}} = \frac{1}{2} (\mathbf{H}^N + \mathbf{H}^{N+1}), \end{aligned} \quad (4)$$



where the superscript N indicates the index of time steps. The inviscid flux vector \mathbf{H}_l is calculated by Roe's flux difference splitting [12]. To provide second-order accuracy, the MUSCL (Monotonic Upstream-centred Scheme for Conservation Laws) scheme is applied. The primitive variables of \mathbf{q} are linearly reconstructed using the gradient computed with the Green-Gauss method and Hishida's limiter [13]. $\mathbf{q}\tilde{n}_t$ in Eq. 4 is estimated by the upwind scheme:

$$\mathbf{q}\tilde{n}_t = \frac{1}{2} [\mathbf{q}^+ \tilde{n}_t + \mathbf{q}^- \tilde{n}_t - |\tilde{n}_t| (\mathbf{q}^+ - \mathbf{q}^-)].$$

In the MGFV method, time stepping is represented by the following equation:

$$\frac{\partial \mathbf{q}}{\partial t} = -\frac{1}{\tilde{\Omega}} \left[\mathbf{q}^{N+1} (\tilde{n}_t)_6 + \mathbf{q}^N (\tilde{n}_t)_5 + \sum_{l=1}^4 \left\{ \mathbf{q}^{N+\frac{1}{2}} \tilde{n}_t + \mathbf{H}^{N+\frac{1}{2}} \right\}_l \right]. \quad (5)$$

For the three-dimensional MGFV method, $\tilde{\Omega}$ represents the four-dimensional control volume, and the value can be approximated with the corresponding three-dimensional sixth volume V_6 :

$$\tilde{\Omega} \approx \Delta t V_6.$$

Here, V_6 has the same value as the absolute value of the normal vector to the sixth volume $\tilde{\mathbf{n}}_6$, but $\tilde{\mathbf{n}}_6$ comprises only the \tilde{n}_t component, then

$$V_6 = (\tilde{n}_t)_6.$$

Therefore, Eq. 5 finally becomes:

$$\frac{\partial \mathbf{q}}{\partial t} = -\frac{1}{\Delta t (\tilde{n}_t)_6} \left[\mathbf{q}^{N+1} (\tilde{n}_t)_6 + \mathbf{q}^N (\tilde{n}_t)_5 + \sum_{l=1}^4 \left\{ \mathbf{q}^{N+\frac{1}{2}} \tilde{n}_t + \mathbf{H}^{N+\frac{1}{2}} \right\}_l \right].$$

A pseudo-time approach based on the Newton-iteration scheme is employed to solve unsteady flows, and the two-stage rational Runge-Kutta (RRK) scheme [14] is applied to pseudo-time stepping. Here, the two-stage RRK scheme can be written with pseudo-time t^* and pseudo-time step ν :

$$\begin{aligned} \mathbf{g}_1 &= -\Delta t^* \mathfrak{L} \left(\mathbf{q}^{N+1(\nu)} \right) \\ \mathbf{g}_2 &= -\Delta t^* \mathfrak{L} \left(\mathbf{q}^{N+1(\nu)} + c_2 \mathbf{g}_1 \right) \\ \mathbf{g}_3 &= b_1 \mathbf{g}_1 + b_2 \mathbf{g}_2 \\ \mathbf{q}^{N+1(\nu+1)} &= \mathbf{q}^{N+1(\nu)} + \frac{2\mathbf{g}_1(\mathbf{g}_1 \cdot \mathbf{g}_3) - \mathbf{g}_3(\mathbf{g}_1 \cdot \mathbf{g}_1)}{(\mathbf{g}_3 \cdot \mathbf{g}_3)}, \end{aligned} \quad (6)$$

where

$$\mathfrak{L}(\mathbf{q}^{N+1}) = \frac{1}{\Delta t (\tilde{n}_t)_6} \left[\mathbf{q}^{N+1} (\tilde{n}_t)_6 + \mathbf{q}^N (\tilde{n}_t)_5 + \sum_{l=1}^4 \left\{ \mathbf{q}^{N+\frac{1}{2}} \tilde{n}_t + \mathbf{H}^{N+\frac{1}{2}} \right\}_l \right]$$

$$\mathbf{q}^{N+1(0)} = \mathbf{q}^N.$$

$\mathbf{g}_i \cdot \mathbf{g}_j$ represents the summation of the scalar product of vectors \mathbf{g}_i and \mathbf{g}_j in all the cells. b_1 , b_2 , and c_2 are coefficients which satisfy the following relations:

$$b_1 + b_2 = 2.0, \quad b_2 c_2 \leq -0.5.$$

This study uses $b_1 = 1.0$, $b_2 = -1.0$, $c_2 = 2.0$ to provide second-order accuracy.

3 Moving computational multi-domain method

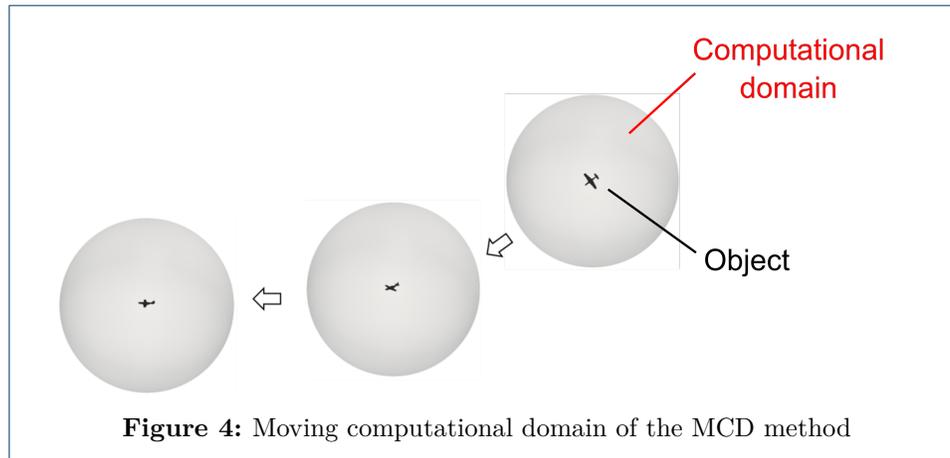
3.1 Moving computational domain (MCD) method

The MCD method [8] is applied when objects move. In order to simulate movement over a large space, an object is located in a computational domain which moves in line with the object inside, as shown in Fig. 4.

3.2 Classification of the cells and their interpolation

3.2.1 Categorising vertices and cells

Inter-grid communication can be achieved by applying the overset approach [4] to implement the interpolation. Each cell is categorised as an active cell, a nonactive cell, or an interpolation cell, based on the positional relationship between one grid



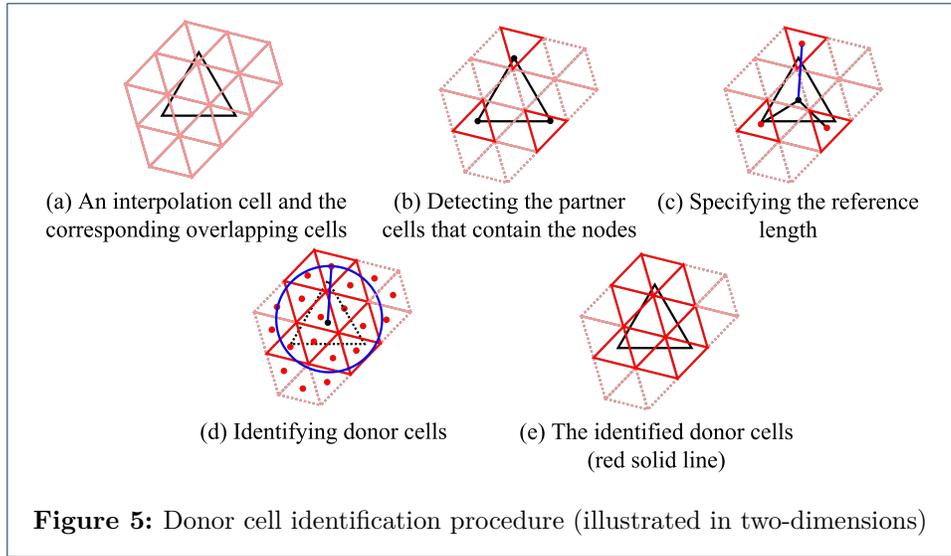
and the other overlapping grid. Active cells are the cells where the flow field variables are calculated; nonactive cells are the cells which are outside the calculating area; and the interpolation cells are the cells where the variables are interpolated.

The cells are classified by following the steps below. First, whether the nodes of each grid are included in the overlapping partner grid is determined. After that, all the node points are designated as either active or nonactive based on the node classification method described in the study by Nakahashi (2000). Since objects are represented as holes in the grids, if a node is situated inside the object in the partner grid, the node is designated as an in-hole node. Then, a tetrahedron cell is assigned as an active cell if all the nodes of the cell are active; as a nonactive cell if all the nodes of the cell are nonactive or if one or more nodes of the cell are in-hole nodes; and the remaining cells are the interpolation cells which are responsible for inter-grid communication. If the layer of the detected interpolation cells is insufficiently thin, the cells around the interpolation cells are also reassigned as interpolation cells, ensuring a sufficient number of interpolated cells.

3.2.2 Finding donor cells

After the cells are classified, the donor cells are identified. The variables of the donor cells are used for the interpolation to obtain those of the interpolation cells. First, the nodes of each tetrahedron interpolation cell are checked to determine which overlapping partner cells contain them; Fig. 5a and b illustrate this in two-dimensions. In the next step, the distances between the centre of an interpolation cell and the centres of its detected partner cells are calculated, and in turn, the partner cell that has the largest distance is found, where the distance is determined as a reference length (Fig. 5c). Then, the distances between the centre of an interpolation cell and the centres of the other cells in the overlapping partner grid are calculated. The partner cells with smaller centre-to-centre distances than the reference length are identified as donor cells (Fig. 5d and e).

It is important to note that if any of the donor cells detected for an interpolation cell is not active at the current or previous step, the interpolation cell can no longer be an interpolation cell and must be changed to an active cell. This process ensures that variables are interpolated only from cells which have reliable values.



3.2.3 Inverse distance weighted interpolation

Inverse distance weighting is employed for the interpolation. \mathbf{q} at the interpolation cells \mathcal{X}^{ip} , is computed from the variables of the donor cells $\mathbf{q}^{\text{donor}}$:

$$\mathbf{q}^{\text{ip}} = \frac{\sum_{i=1}^m w_i \mathbf{q}_i^{\text{donor}}}{\sum_{i=1}^m w_i}$$

$$w_i = \frac{1}{d(\mathcal{X}^{\text{ip}}, \mathcal{X}_i^{\text{donor}})^p}$$

Here, $d(\mathcal{X}^{\text{ip}}, \mathcal{X}_i^{\text{donor}})$ is the distance between the centre of an interpolation cell \mathcal{X}^{ip} and the centre of the corresponding donor cells $\mathcal{X}_i^{\text{donor}}$, and m represents the total number of donor cells found for an interpolation cell. The power parameter $p = 1$ is used in this study.

3.2.4 Modification of the RRK scheme for inter-grid communication

The cells which are outside the flow field must be excluded in the process of calculating flow. Therefore, each cell i has the following value depending on whether the cell is a blanked cell or not:

$$\text{iblack}(i) = \begin{cases} 1, & \text{if a cell is not blanked} \\ 0, & \text{if a cell is blanked.} \end{cases}$$

Multiplying this $\text{iblack}(i)$ to the formula of the RRK scheme ensures that the solutions are not updated in a blanked cell. Since active cells are the cells that calculate the flow field variables, interpolation cells are the cells that only perform interpolation, and nonactive cells are the cells outside the calculating area, $\text{iblack}(i)$ takes the form of the following equation:

$$\text{iblack}(i) = \begin{cases} 1, & \text{if a cell is active} \\ 0, & \text{else.} \end{cases}$$

Therefore, the RRK scheme (Eq. 6) is modified with the preceding iblack:

$$\mathbf{q}_i^{N+1(\nu+1)} = \mathbf{q}_i^{N+1(\nu)} + \text{iblack}(i) \frac{2\mathbf{g}_{1i}(\mathbf{g}_1 \cdot \mathbf{g}_3) - \mathbf{g}_{3i}(\mathbf{g}_1 \cdot \mathbf{g}_1)}{(\mathbf{g}_3 \cdot \mathbf{g}_3)}. \quad (7)$$

4 Validations

Two cases were tested to validate the reliability of this method. One is a shock tube test with a shock tube including a moving grid inside. An empty grid (sub-grid) was prepared in the shock tube (main grid), and the sub-grid was moved. Since the sub-grid does not contain any object inside, the main flow should not be affected even if the sub-grid moves. Therefore, the purpose of this test is to confirm that grid movement does not affect the flow. In addition, the results can be compared with the exact solution to verify that the variables are accurately interpolated between the overlapping grids.

Another validation test involves crossing two spheres. This context is close to the practical application of this method, such as aerial vehicles passing each other. However, the simulation of real-life vehicles with complex shapes can cause problems due to the difficulties in modelling the configurations and conditions. For this reason, two spheres were chosen as a simple shape for the validation test. In addition, a sphere-crossing simulation has been reported so far in [11]. Therefore, the current study uses sphere crossing as a validation test to allow the results to be compared with the values calculated using the method reported by Yamakawa (2017).

4.1 Shock tube test

4.1.1 Computational conditions

A gas at high pressure and a gas at low pressure are separated in a tube by the diaphragm, which serves as a barrier. A sub-grid was placed in the low-pressure region of a shock tube (main grid), and the grid was moved in the z direction. The initial positions of the main grid and sub-grid are depicted in Fig. 6. Since a shock wave passes through the low-pressure region of the main grid, the sub-grid was positioned in the region in order to verify that the interpolation does not change the shape of the shock wave. Although the sub-grid can be moved in either the x , y or z direction to verify this method, the movement in the x direction was fixed. This allowed the flow and shock wave before, during, and after passing through the sub-grid to be examined at a cross section perpendicular to the y -axis or z -axis, since the shock wave propagates in the x direction. The y direction was also fixed to check the flow and movement of the sub-grid in a recognisable manner. Thus, the sub-grid moved in the z direction in Eq. 8 from rest at $t = 0$. The initial conditions are shown in Table 1. The boundary conditions at the wall of the main grid were the slip conditions for an inviscid fluid. All the cells at the wall of the sub-grid were interpolation cells. The variables in the sub-grid were calculated using the values interpolated from the outer main grid. Conversely, the information from the sub-grid was communicated to the main grid through the interpolation cells of the main grid. The test was carried out until the shock wave reached the right end of the main grid.

$$\Delta z = 0.075 + 0.075 \sin(\pi t - 0.5\pi) \quad (8)$$

Table 1: Initial conditions (shock tube test)

High-pressure region (subscript H)		Low-pressure region (subscript L)	
ρ_H	1.0	ρ_L	0.1
u_H	0.0	u_L	0.0
v_H	0.0	v_L	0.0
w_H	0.0	w_L	0.0
p_H	ρ_H/γ	p_L	ρ_L/γ

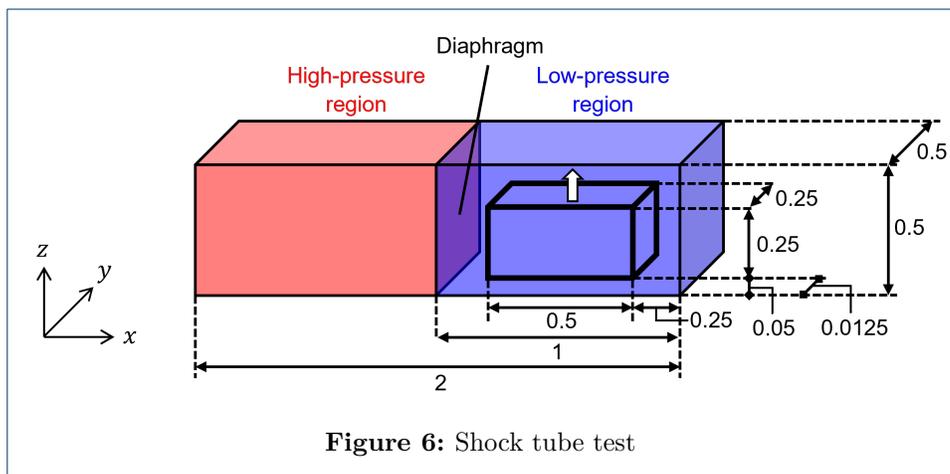


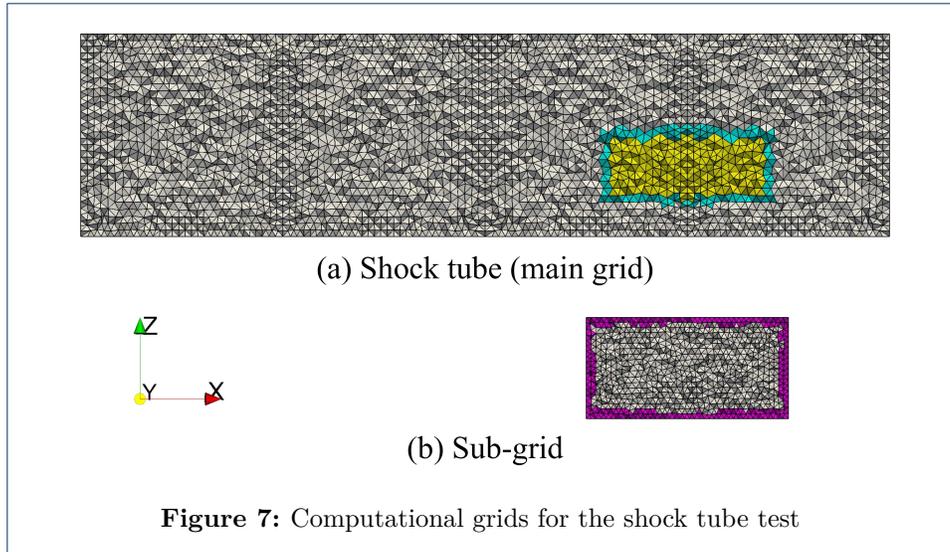
Figure 6: Shock tube test

The two unstructured grids for the main grid with 512152 cells and for the sub-grid with 80928 cells were generated by *MEGG3D* (an unstructured mesh generator [15, 16]). Fig. 7 shows the cross section at the middle of the length in the direction of (a) the main grid and (b) the sub-grid, with colouring according to the cell type. The white cells represent active cells, the cyan cells are the interpolation cells of the main grid, the magenta cells are the interpolation cells of the sub-grid, and the yellow cells are nonactive cells. Fig. 7 implies that the region of the main grid which overlaps the sub-grid was occupied with nonactive cells, and the sub-grid performed the calculation instead.

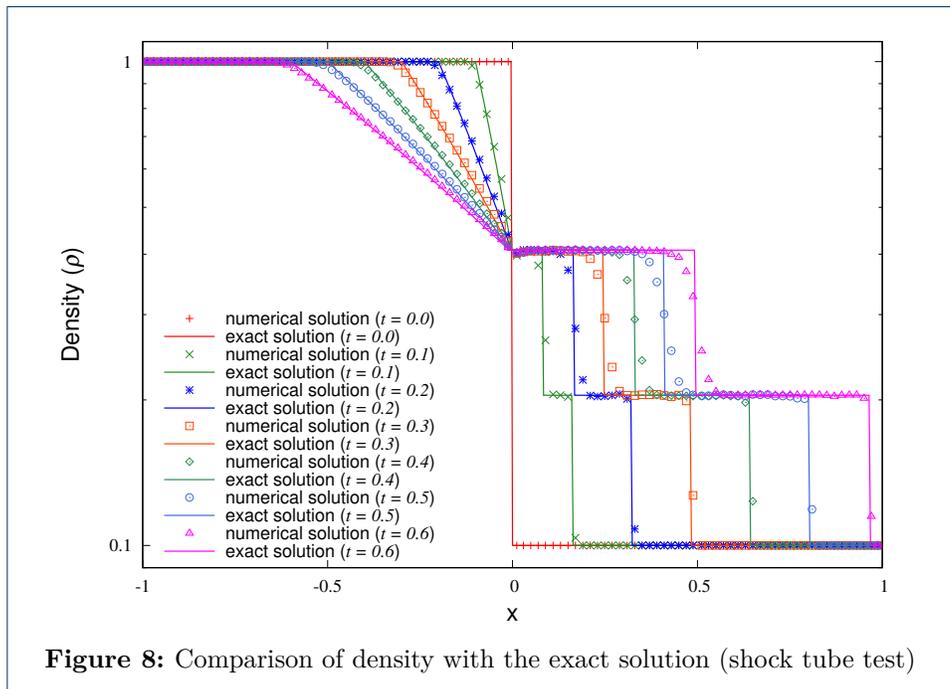
4.1.2 Results and discussion

The graph in Fig. 8 provides the density distribution at the centre of the length in the y and z directions of the grids and compares it with the exact solution. Fig. 9 presents the flow field in the cross section at the centre of the length in the y direction. The upper side illustrates the density distribution while the lower side illustrates the grids. The nonactive cells are omitted, and only the active and interpolation cells are shown. The colouring rule is the same as in Fig. 7, but the active cells of the sub-grid are shown in grey tones to emphasise the fact that they belong to the sub-grid.

The grids in Fig. 9 show that the sub-grid moved upwards. In addition, the interpolation cells of each grid did not overlap but were slightly shifted to ensure that the donor cells were active cells, as mentioned in Section 3.2.2. It can also be seen that the classification of each cell was dynamically updated at each step as the sub-grid moved. The density distribution in Fig. 9 shows that the shock propagated after the diaphragm was removed. The shape of the shock remained after



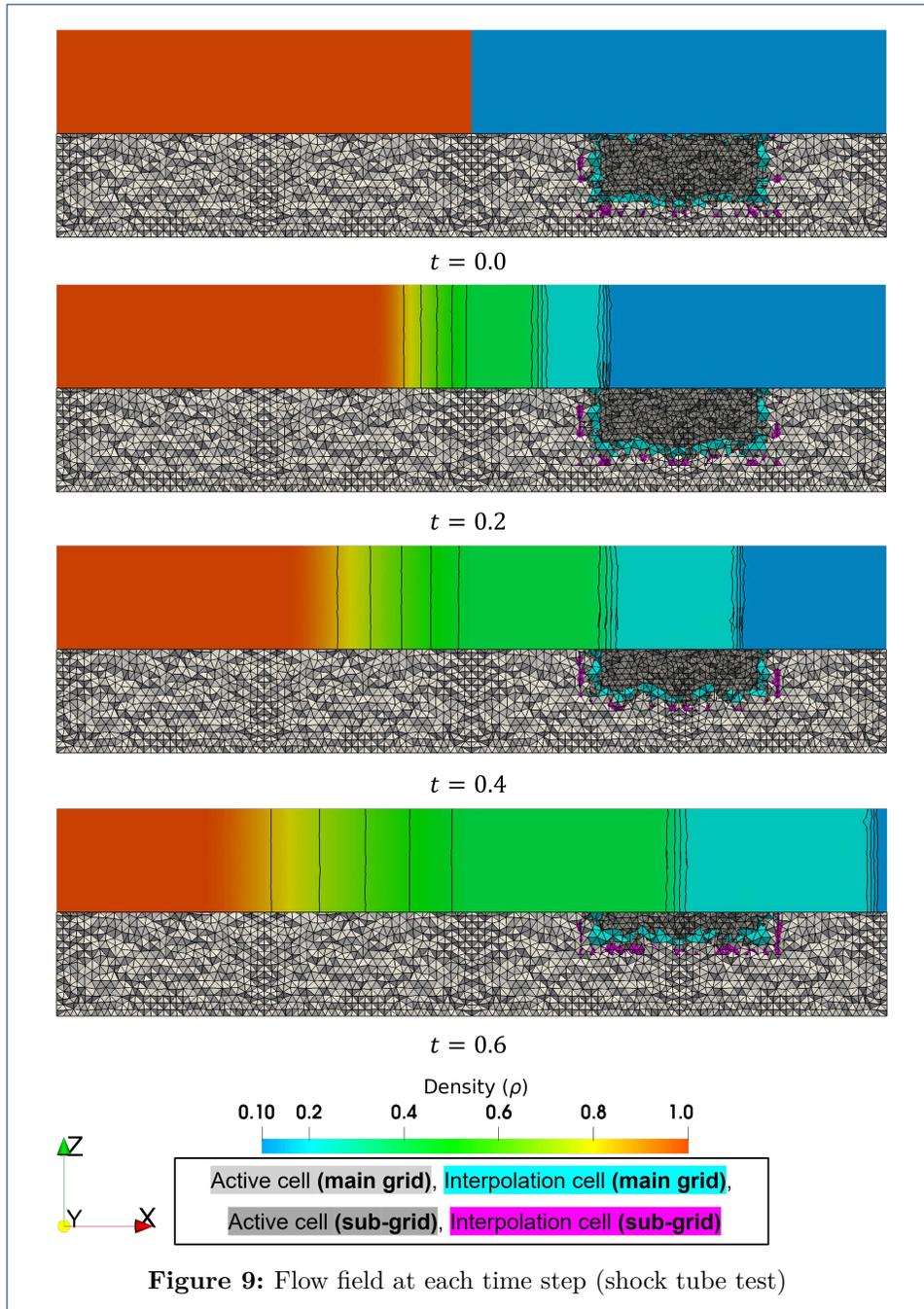
it passed through the sub-grid. Fig. 8 shows the position of the shock that was captured at each time, compared with the exact solution. Therefore, these results verify that grid movement does not affect the flow and that the flow field variables are accurately communicated between the two grids.



4.2 Crossing test

4.2.1 Computational conditions

Two identical grids which include a sphere in the centre were prepared in the configuration depicted in Fig. 10. The grids are not shifted in the y direction. To compare with the values provided using the method described in Section 2 and 3 in [11],



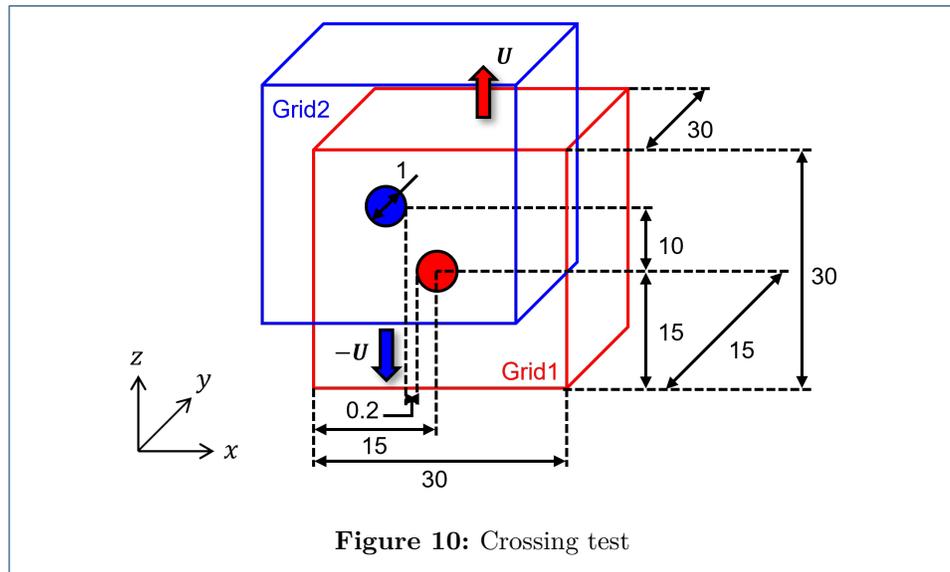
the spheres were placed in exactly the same initial positions and were moved at exactly the same speed as the ones in the literature. The comparative data were calculated using remeshing, adding and eliminating the cells as the spheres moved, with a single grid including the two spheres, as depicted in Fig. 7 of the literature. Please refer to the literature for the detailed methodology for the comparative data.

In the current study, the two spheres started to approach each other at speed U in the z direction from the position where the centres were 10 apart and crossed at a distance of 0.2. The moving speed U is shown in Eq. 9. Since the velocity is nondimensionalised by the speed of sound in this study (Eq. 2), U was equal to the speed of sound when $U = 1.0$. The initial conditions are shown in Table 2. The boundary conditions were the slip conditions at the wall of the spheres and the Riemann invariant boundary conditions at the outer boundaries of each grid.

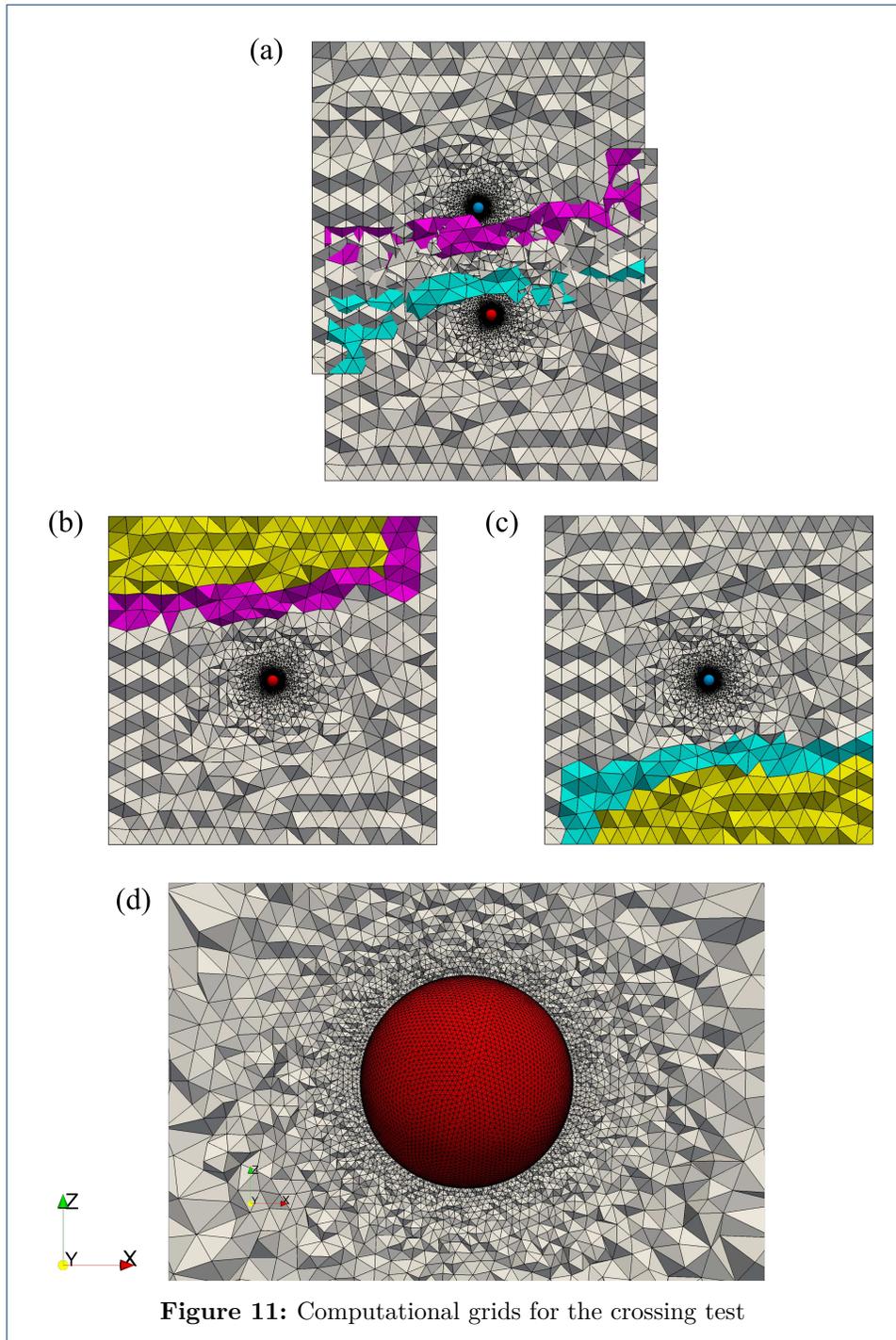
$$U = \begin{cases} 10t, & t < 0.1 \\ 1.0, & t \geq 0.1 \end{cases} \quad (9)$$

Table 2: Initial conditions (crossing test)

Whole region	
ρ	1.0
u	0.0
v	0.0
w	0.0
p	ρ/γ



The two identical unstructured grids with 502506 cells were generated by *MEGG3D*. The fineness of the computational grids was as close as possible to that of the literature. Fig. 11 illustrates the cross section of the grids at the centre of the spheres in the y direction, with (a) showing the overlapping grids without the nonactive cells, with (b) and (c) showing Grid 1 and Grid 2, respectively, and with



(d) showing the cells around the sphere. The cells are coloured according to the cell type. The white cells represent active cells, the magenta cells are the interpolation cells of Grid 1, the cyan cells are the interpolation cells of Grid 2, and the yellow cells are nonactive cells. At the region where Grid 1 and Grid 2 overlap, only one grid or the other has nonactive cells.

4.2.2 Results and discussion

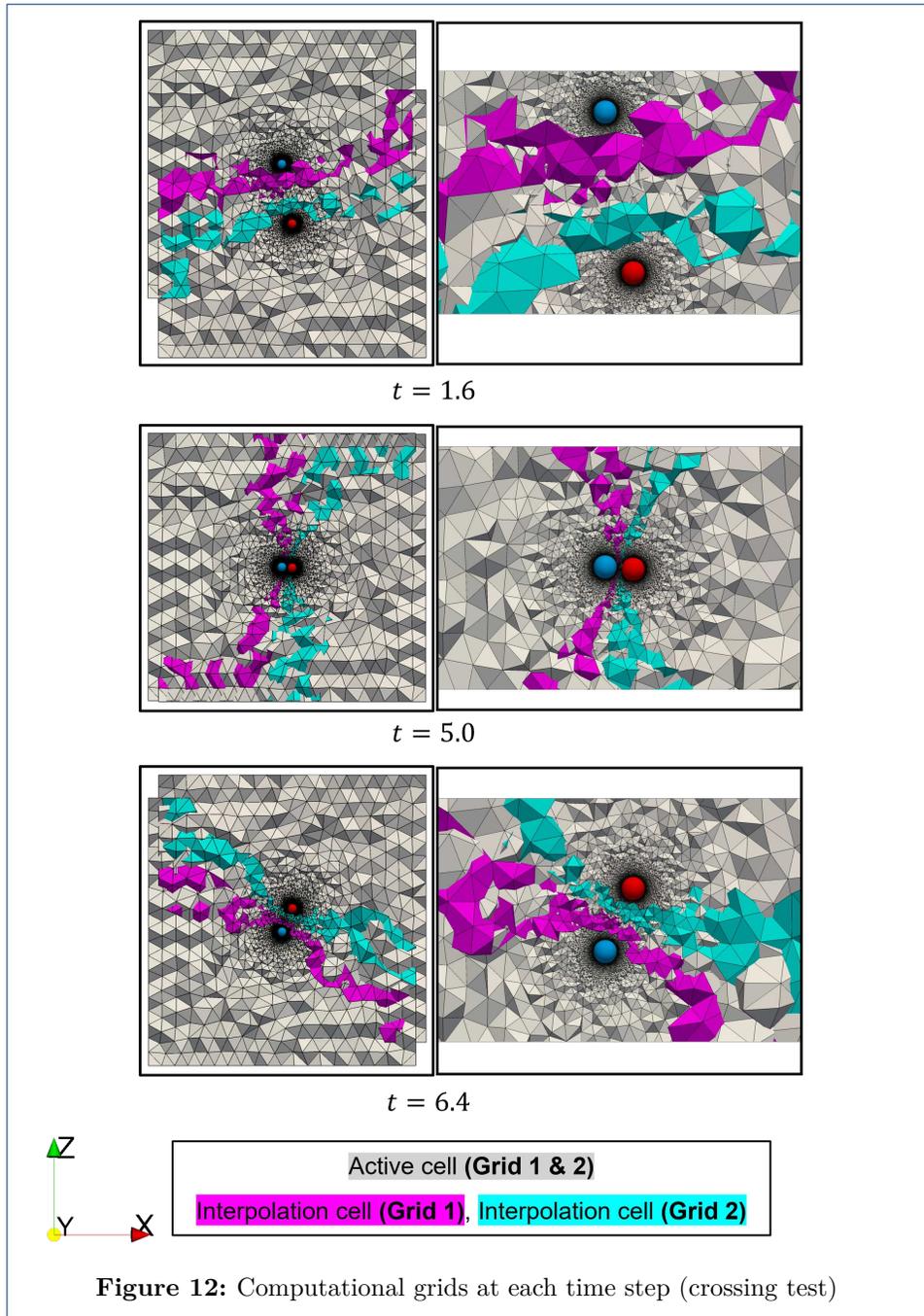
Fig. 12 shows the cross section of the grids at the centre of the spheres in the y direction, with each step of the spheres (a) approaching each other ($t = 1.6$), (b) crossing ($t = 5.0$), and (c) moving away from each other ($t = 6.4$). The left side depicts the whole view, and the right side illustrates the enlarged view around the spheres. The nonactive cells are omitted, and only the active and interpolation cells are shown. The colouring rule is the same as in Fig.11. Fig. 13 shows the pressure distribution on the surface of the spheres and the contour plot of the pressure in the flow field at each time step. The graph in Fig. 14 provides the time histories of the drag coefficient C_D , the lift coefficient C_L , and the side-force coefficient C_S for one sphere and compares them with the values calculated using the method of the literature [11]. These coefficients are calculated by Eq. 10:

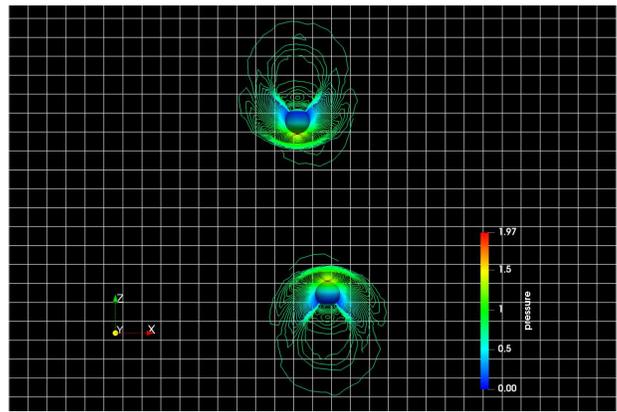
$$C_D = \frac{2F_D}{\rho V^2 S}, \quad C_L = \frac{2F_L}{\rho V^2 S}, \quad C_S = \frac{2F_S}{\rho V^2 S} \quad (10)$$

where F_D , F_L , and F_S represent the drag, lift, and side-force, respectively, which are the forces exerted on the sphere in the z -, y -, and x -axes, respectively. V indicates the maximum speed of the sphere, and S denotes the frontal area of the sphere.

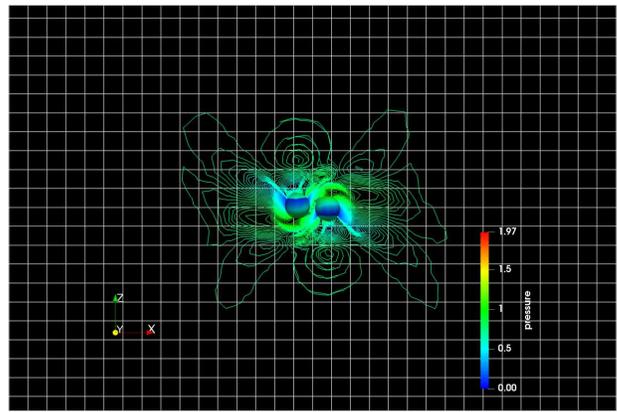
The grids in Fig. 12 show that the computational domains followed the movement of the corresponding spheres as they moved. Each domain moved independently, and the classification of each cell was updated as the steps progress. Fig. 13 implies that a bow shock was generated in front of the spheres, which travel at the speed of sound, and that another shock was generated behind the spheres at $t = 1.6$. When crossing at $t = 5.0$, the shock in front of one sphere and the shock behind the other sphere interacted. After crossing at $t = 6.4$, the shock behind each sphere affected the rear of each other's sphere. Similar phenomena can be observed in the literature by Yamakawa (2017), and therefore it can be said that the flow field variables were successfully communicated between the domains in this study. Fig. 14 shows that C_D made a dramatic increase just before crossing and then fell drastically as the crossing was over. These phenomena were caused by the mutual influence of the high-pressure fields in front of the two spheres and the low-pressure fields behind the spheres. In addition, fluctuation can also be observed in C_S , and this implies that the forces which made the spheres waver were exerted on the spheres when crossing. The comparative data (the dashed lines) in Fig. 14 suggests that the same trend can be seen in all the drag, lift, and the side-force coefficients. In particular, these coefficients were in good agreement with the comparative data when crossing, which is a noticeable feature in this simulation.

These results show that the flow interaction of multiple moving objects can be simulated using computational domains created around each object that communicate with each other. On the other hand, these multiple-object simulations using

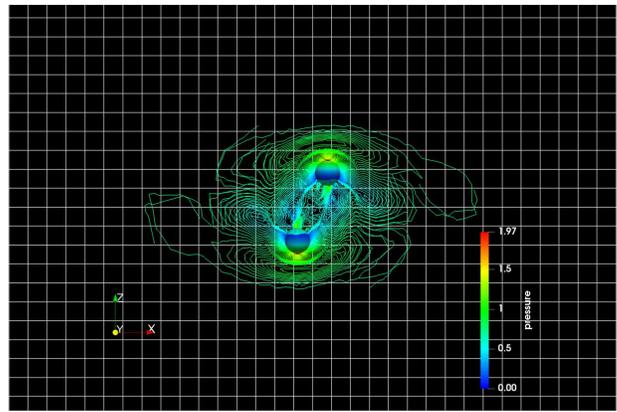




$t = 1.6$



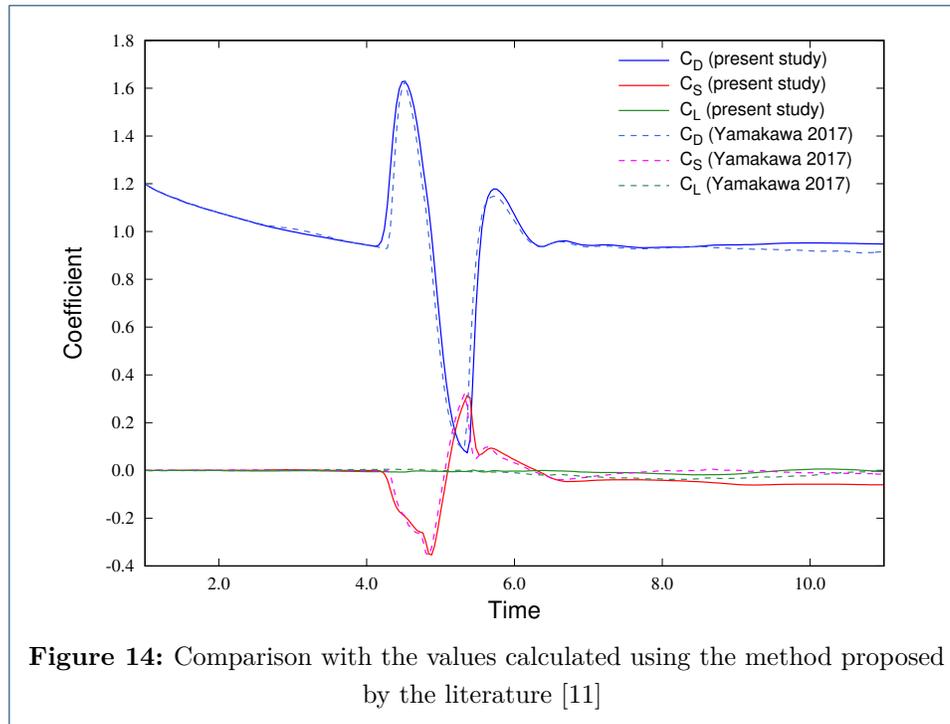
$t = 5.0$



$t = 6.4$



Figure 13: Pressure distribution on the sphere surface and pressure contour at each time step



a single fixed computational domain containing all the objects normally require remeshing the inside of the domain as the objects move. Such remeshing needs to introduce complex techniques, such as adding and eliminating cells as represented in Yamakawa (2017). In addition, the limitation in the size of the computational domain constrains the space in which objects can move. In contrast, remeshing is unnecessary with the method introduced in the current study. Moreover, the computational domains themselves can be moved freely, which makes it possible to move each object along with the corresponding domain, without spatial limitations.

5 Conclusions

This study introduces moving computational domains that communicate with each other as a means of simulating the full trajectory of multiple moving objects with the aim of improving their aerodynamic efficiency and safety. The MCD method is applied to modelling the movement of objects. In this method, computational domains are moved in line with the trajectory of each object, thereby removing the spatial constraints for simulating moving objects. The interaction of each domain's flow with the other's flow is achieved by applying the overset approach, which allows the communication of the flow field variables. A shock tube test is conducted to verify that the basic requirements for this method are satisfied. The comparison of the results to the exact solution confirms that the flow variables are accurately interpolated between the domains. Then, the crossing of two spheres is simulated. This test demonstrates that the shock wave generated around each sphere interacts with the other, and that fluctuation can be observed in the drag coefficient of one sphere. Also, fluctuation observed in the side-force coefficient of the sphere suggests that the forces that cause the spheres to waver are exerted on them when crossing.

These results verify that this method can reliably model the flows around multiple objects and their interaction.

Acknowledgements

N/A.

Funding

This publication was subsidised by JKA through its promotion funds from KEIRIN RACE and by JSPS KAKENHI Grant Number 21K03856.

Abbreviations

CFD: Computational Fluid Dynamics

MCD: Moving Computational Domain

MGFV: Moving-Grid Finite Volume

GCL: Geometric Conservation Law

RRK: Rational Runge-Kutta

MEGG3D: Mixed-Element Grid Generator in 3 Dimensions

Availability of data and materials

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MN is the corresponding author of this paper, who was responsible for conceptualisation, program development, and writing. YM contributed to conducting the validation tests, interpreting data, and writing the manuscript. SA and ST participated in the discussion and contributed to interpreting data and writing the manuscript. MB participated in the discussion and contributed to writing the manuscript. All the authors read and approved the final manuscript.

Author details

¹Department of Mechanical and System Engineering, Kyoto Institute of Technology, Kyoto, Japan. ²College of Industrial Technology, Department of Mechanical Engineering, Hyogo, Japan.

References

1. Peskin, C.S.: Flow patterns around heart valves: A numerical method. *J Comput Phys* **10**(2), 252–271 (1972). doi:10.1016/0021-9991(72)90065-4
2. Mittal, R., Iaccarino, G.: Immersed boundary methods. *Annu Rev Fluid Mech* **37**(1), 239–261 (2005). doi:10.1146/annurev.fluid.37.061903.175743
3. Steger, J.L., Benek, J.A.: On the use of composite grid schemes in computational aerodynamics. *Comput Methods Appl Mech Eng* **64**(1), 301–320 (1987). doi:10.1016/0045-7825(87)90045-4
4. Nakahashi, K., Togashi, F., Sharov, D.: Intergrid-boundary definition method for overset unstructured grid approach. *AIAA J* **38**(11), 2077–2084 (2000). doi:10.2514/2.869
5. Crozon, C., Steijl, R., Barakos, G.N.: Coupled flight dynamics and CFD — demonstration for helicopters in shipborne environment. *Aeronaut J* **122**(1247), 42–82 (2018). doi:10.1017/aer.2017.112
6. Matsuno, K.: In: Topping, B.H.V., Adam, J.M., Pallarés, F.J., Bru, R., Romero, M.L. (eds.) *Development and Applications of a Moving Grid Finite Volume Method*, vol. 5, pp. 103–129 (2010). doi:10.4203/cssets.26.5
7. Thomas, P.D., Lombard, C.K.: Geometric conservation law and its application to flow computations on moving grids. *AIAA J* **17**(10), 1030–1037 (1979). doi:10.2514/3.61273
8. Watanabe, K., Matsuno, K.: Moving computational domain method and its application to flow around a high-speed car passing through a hairpin curve. *J Comput Sci Technol* **3**(2), 449–459 (2009). doi:10.1299/jcst.3.449

9. Takii, A., Yamakawa, M., Asao, S., Tajiri, K.: Six degrees of freedom flight simulation of tilt-rotor aircraft with nacelle conversion. *J Comput Sci* **44**, 101164 (2020). doi:10.1016/j.jocs.2020.101164
10. Yamakawa, M., Mizuno, N., Asao, S., Tanaka, M., Tajiri, K.: Optimization of knee joint maximum angle on dolphin kick. *Phys Fluids* **32**(6), 067105 (2020). doi:10.1063/1.5142422
11. Yamakawa, M., Mitsunari, N., Asao, S.: Numerical simulation of rotation of intermeshing rotors using added and eliminated mesh method. *Procedia Comput Sci* **108**, 1883–1892 (2017). doi:10.1016/j.procs.2017.05.061. International Conference on Computational Science, ICCS 2017, 12–14 June 2017, Zurich, Switzerland
12. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. *J Comput Phys* **43**(2), 357–372 (1981). doi:10.1016/0021-9991(81)90128-5
13. Hishida, M., Hashimoto, A., Murakami, K., Aoyama, T.: A new slope limiter for fast unstructured CFD solver FaSTAR (*in Japanese*). In: JAXA Special Publication: Proceedings of 42nd Fluid Dynamics Conference / Aerospace Numerical Simulation Symposium 2010, vol. JAXA-SP-10-012 (2011). <https://ci.nii.ac.jp/naid/120006828212/>
14. Wambecq, A.: Rational Runge-Kutta methods for solving systems of ordinary differential equations. *Computing* **20**(4), 333–342 (1978). doi:10.1007/BF02252381
15. Ito, Y., Nakahashi, K.: Surface triangulation for polygonal models based on CAD data. *Int J Numer Methods Fluids* **39**, 75–96 (2002). doi:10.1002/fld.281
16. Ito, Y.: Challenges in unstructured mesh generation for practical and efficient computational fluid dynamics simulations. *Comput Fluids* **85**, 47–52 (2013). doi:10.1016/j.compfluid.2012.09.031