# Multi Axes Sliding Mesh Approach for Compressible Viscous Flows

Masashi Yamakawa[1]([✉]), Satoshi Chikaguchi[1], Shinichi Asao[2], and Shotaro Hamato[1]

[1] Kyoto Institute of Technology, Matsugasaki Sakyo-ku, Kyoto 606-8585, Japan
yamakawa@kit.ac.jp
[2] College of Industrial Technology, Nishikoya, Amagasaki, Hyogo 661-0047, Japan

**Abstract.** To compute flows around a body with a rotating or movable part like a tiltrotor aircraft, the multi axes sliding mesh approach has been proposed. This approach is based on the unstructured moving grid finite volume method, which has adopted the space-time unified domain for control volume. Thus, it can accurately express such a moving mesh. However, due to the difficulty of mesh control in viscous flows and the need to maintain the stability of computation, it is restricted to only inviscid flows. In this paper, the multi axes sliding mesh approach was extended to viscous flows to understand detailed flow phenomena around a complicated moving body. The strategies to solve several issues not present in inviscid flow computations are described. To show the validity of the approach in viscous flows, it was applied to the flow field of a sphere in uniform flow. Multiple domains that slide individually were placed around the sphere, and it was confirmed that the sliding mesh did not affect the flow field. The usability of the approach is expected to be applied to practical viscous flow computations.

**Keywords:** Computational fluid dynamics · Unstructured moving mesh · Sliding mesh approach · Viscous flows

## 1 Introduction

Numerical simulations of flows around a body with movable parts like a rotorcraft or sports athlete has a high utility value for various fields. However, handling a moving mesh is challenging in a body-fitted coordinate system. When the movable scope of its parts is small, the moving mesh method using a tension spring [1] can be used. On the other hand, for large motions, the mesh method is restricted. It is almost impossible to express a rotary motion such as the rotor part of a helicopter by using the moving mesh method with spring. To resolve this issue, the sliding mesh approach [2] was proposed. In this approach, the motion of a body is expressed by sliding the boundary of adjacent divided computational domains. This is different from the overset grid method in which one domain is put on another domain. An information exchange of physical values between domains is then conducted by interpolation, which might not satisfy physical conservation laws. On the other hand, by using the sliding mesh approach for the information exchange, the physical value can be conserved. One of the simplest

applications of the sliding mesh approach is the divided cylindrical computational domain for axial direction. Its rotating cylinder has been applied to, for example, the simulation of a flow around a multistage turbine cascade. Also, one domain can be also embedded in another domain. In this case, the embedded sub domain should be cylindrical or spherical. Furthermore, there should not be a gap between two domains during the rotation of the embedded domain.

Although the sliding mesh approach is very useful, it is difficult to express complicated motion. For example, the rotor part of a helicopter is expressed with comparative ease, but to express the rotor blade of a tiltrotor like the Osprey V-22 is impossible. This is because the rotor blade rotates, and moreover, an engine nacelle having a rotor blade also rotates on different axis to change the flight mode. In this case, the flows around a tiltrotor during rotor-blade mode and fixed-wing mode are computed [3] individually. In a simulation focused on changing flight modes, its computations [4] were conducted for fixed degrees of the engine nacelle at 0, 30, 60, and 90° as calculating a moving engine nacelle was quite difficult. For this issue, we proposed the multi axes sliding mesh approach [5], in which the moving engine nacelle is expressed in the middle size computational domain. The small size domain including the rotating blade is then embedded in the middle size domain with both domains embedded in the large size main domain. Furthermore, we succeeded in rotating the small and middle domains individually. However, the approach is conducted under inviscid flows to prioritize reproducibility of complicated motion. Therefore, the turbulent flow transition phenomenon in the wake of rotor could not be calculated.

The objective of this paper is to apply the multi axes sliding mesh approach to viscous flows. The formulation of the approach and its validity when applying a flow around a sphere will be shown.

## 2 Numerical Approach

### 2.1 Governing Equation

For the governing equation, the following three-dimensional (3D) Navier–Stokes equation for compressible flows written in conservation law form is adopted.

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = \frac{1}{\mathrm{Re}}\left(\frac{\partial \mathbf{E_v}}{\partial x} + \frac{\partial \mathbf{F_v}}{\partial y} + \frac{\partial \mathbf{G_v}}{\partial z}\right) \tag{1}$$

Where

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix},\ \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e+p) \end{bmatrix},\ \mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(e+p) \end{bmatrix},\ \mathbf{G} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(e+p) \end{bmatrix},\ \mathbf{E_v} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ f_{E5} \end{bmatrix},\ \mathbf{F_v} = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ f_{F5} \end{bmatrix},\ \mathbf{G_v} = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ f_{G5} \end{bmatrix}$$

$$\tag{2}$$

The unknown variables $\rho$, $u$, $v$, $w$, and $e$ show the gas density, velocity components in the $x$, $y$, and $z$ directions, and total energy per unit volume, respectively. The working fluid is assumed to be a perfect gas, and the pressure $p$ is defined by

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

(3)

$f_{E5}$, $f_{F5}$, and $f_{G5}$ are shown in Eq. (4). Here, $\mu$ and $\mu_t$ are the coefficients of molecular viscosity and eddy viscosity, respectively. $Pr$, $Pr_t$, and Re are the Prandtl number, turbulent Prandtl number, and Reynolds number, respectively. The ratio of specific heats $\gamma$ is typically taken as being 1.4. In this study, $Pr = 0.72$ and $Pr_t = 0.9$ are obtained.

$$f_{E5} = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \frac{1}{\gamma - 1}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial T^2}{\partial x}$$

$$f_{F5} = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + \frac{1}{\gamma - 1}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial T^2}{\partial y}$$

(4)

$$f_{G5} = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + \frac{1}{\gamma - 1}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial T^2}{\partial z}$$

## 2.2   Numerical Schemes

The sliding mesh approach is a type of moving mesh approach. In this study, the unstructured moving grid finite volume method [6] is adopted. The method assures a geometric conservation law [7] as well as a physical conservation law. A control volume in the space-time unified domain $(x, y, z, t)$, which is four-dimensional (4D) for 3D flows, is then used. This approach has been mainly applied to Euler equations for inviscid compressible flows. In this paper, the approach is discretized for compressible viscous $F_v$ flows. For the discretization, Eq. (1), which is written in divergence form, is integrated as

$$\int_{\Omega} \nabla\widetilde{\mathbf{F}}_v d\Omega = 0,$$

(5)

where

$$\widetilde{\mathbf{F}}_v = \left(\mathbf{E} - \frac{1}{Re}\mathbf{E}_v, \mathbf{F} - \frac{1}{Re}\mathbf{F}_v, \mathbf{G} - \frac{1}{Re}\mathbf{G}_v, \mathbf{q}\right).$$

(6)

Since the approach is based on a cell-centered finite volume method, the flow variables are defined at the center of the cell in the $(x, y, z)$ space. Thus, the control volume becomes a 4D polyhedron in the $(x, y, z, t)$-domain. For the control volume, Eq. (4) is rewritten using the Gauss theorem as:

$$\int_\Omega \widetilde{\nabla \widetilde{\mathbf{F}}}_v d\Omega = \int_V \widetilde{\mathbf{F}}_v \cdot \widetilde{\mathbf{n}} \, dV = \sum_{l=1}^{Ns+2} (\mathbf{q}n_t + \mathbf{\Phi})_l = 0, \tag{7}$$

where

$$\mathbf{\Phi} = \mathbf{H} - \mathbf{H}_v,$$

$$\mathbf{H} = \mathbf{E}n_x + \mathbf{F}n_y + \mathbf{G}n_z, \tag{8}$$

$$\mathbf{H}_v = \frac{1}{\text{Re}} \left( \mathbf{E}_v n_x + \mathbf{F}_v n_y + \mathbf{G}_v n_z \right),$$

Here, $Ns$ indicates the number of boundary surfaces of the element. $l$ is the volume of trajectory generated by the moving boundary surface of the element from $t = n$ to $t = n + 1$. Then, Eq. (7) is rewritten as Eq. (9), and by solving Eq. (9), new $\mathbf{q}$ is obtained.

$$\mathbf{q}^{n+1}(n_t)_{N_s+2} + \mathbf{q}^n(n_t)_{N_s+1} + \sum_{l=1}^{N_s} \left[ \mathbf{q}^{n+1/2} n_t + \mathbf{\Phi}^{n+1/2} \right]_l = 0$$

$$\mathbf{q}^{n+1/2} = \frac{1}{2} \left( \mathbf{q}^{n+1} + \mathbf{q}^n \right) \tag{9}$$

$$\mathbf{\Phi}^{n+1/2} = \frac{1}{2} \left( \mathbf{\Phi}^{n+1} + \mathbf{\Phi}^n \right)$$

The inviscid flux vectors are evaluated using the Roe flux difference splitting scheme [8] with the MUSCL scheme as well as the Venkatakrishnan limiter [9]. The vectors are discretized by central difference. To solve the implicit algorithm, the LU-SGS implicit scheme is adopted.

## 2.3 Evaluation on a Boundary

On a boundary, the first derivative of a physical value cannot be evaluated using central difference. For example, discretization of the first derivative for primitive variable $u$ is described. Figure 1 shows a discretization outline of the first derivative.

The first derivative for primitive variable u is obtained by solving the follow equations.

$$\mathbf{A}\dot{\mathbf{u}} = \mathbf{b}, \tag{10}$$

where

$$\mathbf{A} = \begin{pmatrix} x_c - x_w & y_c - y_w & z_c - z_w \\ 0.5(x_{v2} - x_{v3}) - x_{v1} & 0.5(y_{v2} - y_{v3}) - y_{v1} & 0.5(z_{v2} - z_{v3}) - z_{v1} \\ 0.5(x_{v3} - x_{v1}) - x_{v2} & 0.5(y_{v3} - y_{v1}) - y_{v2} & 0.5(z_{v3} - z_{v1}) - z_{v2} \end{pmatrix}, \tag{11}$$
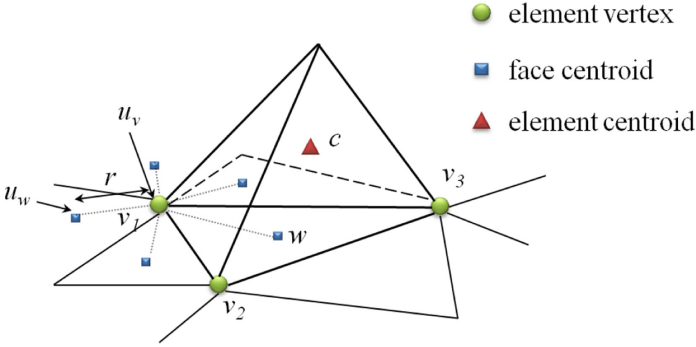
**Fig. 1.** Discretization outline of the first derivative

$$\dot{\mathbf{u}} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} u_c - u_w \\ 0.5(u_{v2} - u_{v3}) - u_{v1} \\ 0.5(u_{v3} - u_{v1}) - u_{v2} \end{pmatrix}.$$

Here, $v$ indicates vertex, $c$ indicates the center of an element, and $w$ indicates the center of a boundary surface for an element. Also, $u_w$ is evaluated as following equation,

$$u_w = \frac{1}{2}\left(u_c + u_{ghost}\right). \tag{12}$$

The vertex of the primitive variable $u_{vi}$ is calculated using the following weighted average method, where $u_{wj}$ is the physical value at the cell center of the triangle constructed by vertex $\psi v_{i\psi}$ and $r_{ij}$ is the distance between the vertex and center point of each cell around it.

$$u_{vi} = \frac{\displaystyle\sum_{j \in i}^{N} u_{wj} \frac{1}{r_{ij}}}{\displaystyle\sum_{j \in i}^{N} \frac{1}{r_{ij}}}, \tag{13}$$

$$r_{ij} = \sqrt{\left(x_{wj} - x_{vi}\right)^2 + \left(y_{wj} - y_{vi}\right)^2 + \left(z_{wj} - z_{vi}\right)^2}. \tag{14}$$

## 3 Sliding Mesh Approach

### 3.1 Multi Axes Sliding Mesh Approach

In the sliding mesh approach, a sliding boundary surface exists. Here, the embedded sub computational domain is rotated in the main domain. In a 3D system, the embedded sub domain should have an almost spherical or cylindrical configuration. Although the

computational cost using the approach is not expensive, the movable range of vertices is limited. In other words, the motions of an object are restricted. Thus, to improve flexibility, the axes of the rotating sub domain are added in the approach. However, to avoid an interaction between sub domains that have individual axes, one sub domain is embedded in the other sub domain, as shown in Fig. 2. In this figure, computational domain 3 is embedded in computational domain 2, which is embedded in computational domain 1. The whole domain can be moved using the moving computational domain (MCD) method [10]. The advantage of this method is that it does not require a spring method to move the object, so it is less likely to create extremely skewed elements. Basically, the multi axes sliding mesh approach has the potential to express any object motion combined with the MCD method without destroying computational mesh.
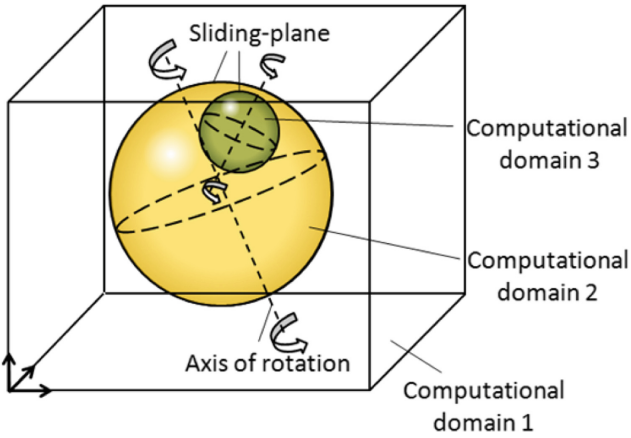


**Fig. 2.** Multi axes sliding mesh approach

The physical values on the sliding plane interpolate with each other through the plane. Interpolation values are determined depending on the area where domain elements overlap. Specifically, the value is calculated in accordance with the area of the overlapping part $S_{ij}$ between the elements of the sliding plane, as shown in Fig. 3. The value of the part is defined with Eq. (15).

$$\mathbf{q}_{\mathbf{b}i} = \frac{\sum\limits_{j \in i} \mathbf{q}_j S_{ij}}{\sum\limits_{j \in i} S_{ij}} \tag{15}$$

Where $\sum_{j \in i}$ shows the sum of cell $j$ adjacent to cell $i$. Then, $\mathbf{q}_j$ is the physical value of cell $j$.
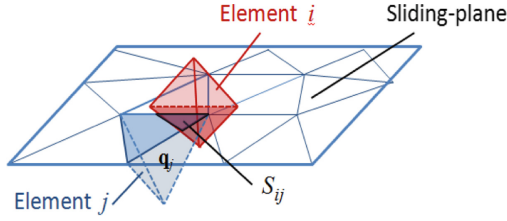
**Fig. 3.** Overlapping part in slide element

## 3.2   Multi Axes Sliding Mesh for Viscous Flow Computation

### 3.2.1   Evaluation of the First Derivative on a Sliding Surface

As the sliding surface is also a boundary, a specific evaluation of the first derivative of the primitive variable on the sliding surface is required along with the evaluation of the boundary. However, unlike boundaries, there is an element on the opposite side of a surface. Thus, the physical value of that element should be used to calculate the first derivative of the primitive variable. First, a ghost cell $j$ adjacent to element $i$ through the boundary surface of element $i$ is generated. Then, $\mathbf{q}_{bj}$ calculated as Eq. (16) is interpolated in element $j$. Here, element $k$ is adjacent to element $i$ across the sliding surface. Figure 4 shows a schematic diagram around the sliding surface.

$$\mathbf{q_{bj}} = \frac{\displaystyle\sum_{k \in j} \mathbf{q}_k S_{jk}}{\displaystyle\sum_{k \in j} S_{jk}}. \tag{16}$$

For example, the first derivative of primitive variable $u$ is calculated using the following central difference Eq. (17) and the partially deformed Gauss-Green's theorem (18).
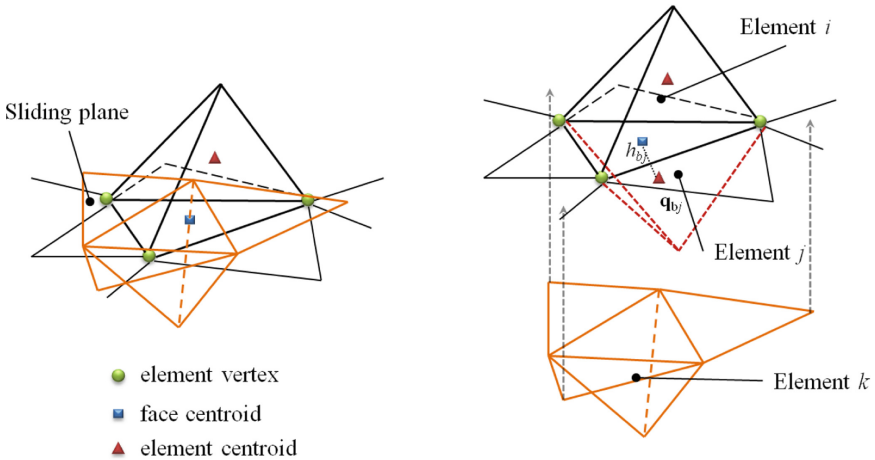


**Fig. 4.** Evaluation of the first derivative on a sliding surface

$$\left(\frac{\partial u}{\partial x}\right)_{ij} = \frac{1}{2}\left\{\left(\frac{\partial u}{\partial x}\right)_i + \left(\frac{\partial u}{\partial x}\right)_j\right\}, \tag{17}$$

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{1}{V_{\Omega i}} \sum_{j\in i}^{N_s} \left(\frac{h_{fi}u_i + h_{fj}u_j\lambda(j) + h_{bj}u_{bj}(1-\lambda(j))}{h_{fi} + h_{fj}\lambda(j) + h_{bj}(1-\lambda(j))}\right)\mathbf{n}_{ij}. \tag{18}$$

Where $h_{bj}$ is calculated using $h_k$, which is the distance between the center point of element $k$ and the center of the adjacent surface of elements $i$ and $j$, as shown in Eq. (19),

$$h_{bj} = \frac{\sum\limits_{k\in j} h_k S_{jk}}{\sum\limits_{k\in j} S_{jk}}. \tag{19}$$

In Eq. (18), if adjacent element $j$ is a ghost cell, $\lambda(j) = 0$, else, $\lambda(j) = 1$.

### 3.2.2 Evaluation of the First Derivative on an Element Having both a Sliding Surface and Boundary

In this subsection, an evaluation of the first derivative on an element that has both a sliding surface and boundary is described. First, the primitive variable for a vertex located on both the sliding surface and boundary is calculated. The first derivative of the primitive variable is then calculated using Eqs. (10) to (14). For example, the calculation procedure of the primitive variable $u_{vi}$ is shown in Eqs. (20) to (22). Its schematic figure of this case is shown in Fig. 5.

$$u_{vi} = \frac{\sum\limits_{j\in i}^{N}\left(u_{wj}\frac{1}{r_{ij}}\lambda_v(j) + u_{bj}\frac{1}{r_{bij}}(1-\lambda_v(j))\right)}{\sum\limits_{j\in i}^{N}\left(\frac{1}{r_{ij}}\lambda_v(j) + \frac{1}{r_{bij}}(1-\lambda_v(j))\right)}, \tag{20}$$

$$r_{bij} = \sqrt{\left(x_{bj}-x_{vi}\right)^2 + \left(y_{bj}-y_{vi}\right)^2 + \left(z_{bj}-z_{vi}\right)^2}, \tag{21}$$

$$x_{bj} = \frac{\sum\limits_{k\in j} x_{ck}S_{jk}}{\sum\limits_{k\in j} S_{jk}}, \quad y_{bj} = \frac{\sum\limits_{k\in j} y_{ck}S_{jk}}{\sum\limits_{k\in j} S_{jk}}, \quad z_{bj} = \frac{\sum\limits_{k\in j} z_{ck}S_{jk}}{\sum\limits_{k\in j} S_{jk}}. \tag{22}$$

Where $u_{bj}$ is the primitive variable in the center of the ghost cell of element $j$ that has vertex $i$. The variable is then calculated from Eq. (15). $(x_{ck}, y_{ck}, z_{ck})$ is the coordinates in the center of element k located adjacent to element $j$ across the sliding surface. Finally, element $j$ has vertex $i$. If element $j$ is a ghost cell, $\lambda_v(j) = 0$, else, $\lambda_v(j) = 1$.
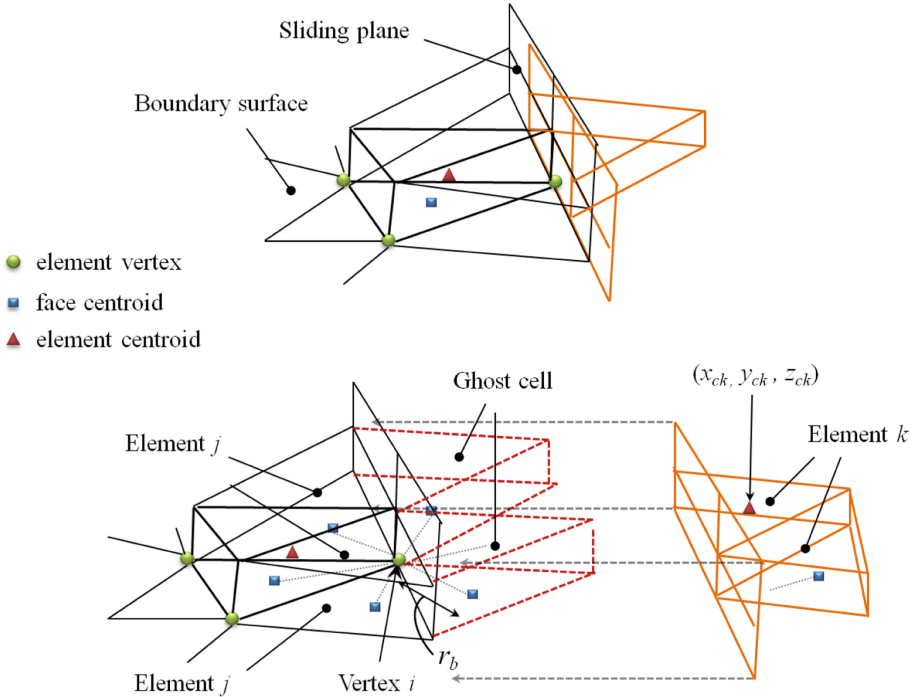
**Fig. 5.** Evaluation of the first derivative on a sliding surface and boundary

### 3.2.3   Prism Element on Sliding Surface

When viscous flows are computed using an unstructured mesh, it is necessary to use quite thin prism elements in the boundary layer. However, if the shape of the body boundary is curved, part of an element might overlap the sliding element and static element as shown in Fig. 6. If there is no overlap between the elements, the physical value cannot be interpolated. Such a problem occurs when the difference between both volumes is not small. Thus, the volume difference should be as small as possible.
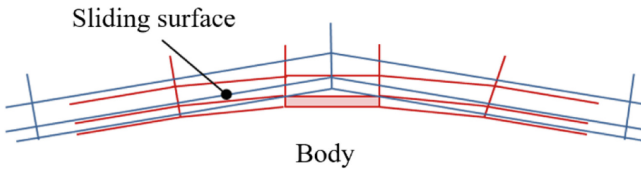


**Fig. 6.** Sliding mesh near a body surface

# 4  Verification of the Multi Axes Sliding Mesh Approach

## 4.1  Application to a Flow Around a Sphere

The multi axes sliding mesh approach is applied to a viscous flow around a sphere. Figure 7 shows a schematic figure of the flow. The sphere is placed in a uniform flow with two sliding cylinders, which have rotation axes in different directions. Each sliding cylindrical mesh rotates around the static sphere, so the sliding mesh must not affect the flow. To confirm the validity of the approach, it is compared with the flow around a sphere in a single mesh. In Fig. 8, case 1 shows a schema of multi axes sliding cylinders around a sphere and case 2 shows its comparison.
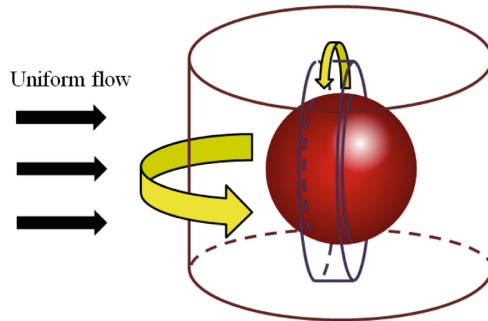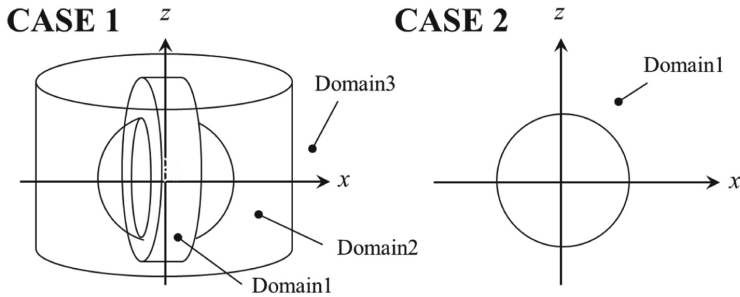


**Fig. 7.** Multi axes sliding mesh around a sphere



**Fig. 8.** Schema of comparative computation

## 4.2  Initial Mesh and Computational Conditions

Figure 9 shows the initial mesh for case 1. The total number of meshes is 4,219,268. Figure 10 shows a single mesh for comparison (case 2). The number of meshes is 4,578,854. Their elements were created by using MEGG3D [11]. The diameter of the computational domain (domain 3 in case 1, whole domain in case 2) is 40 times that of the sphere.
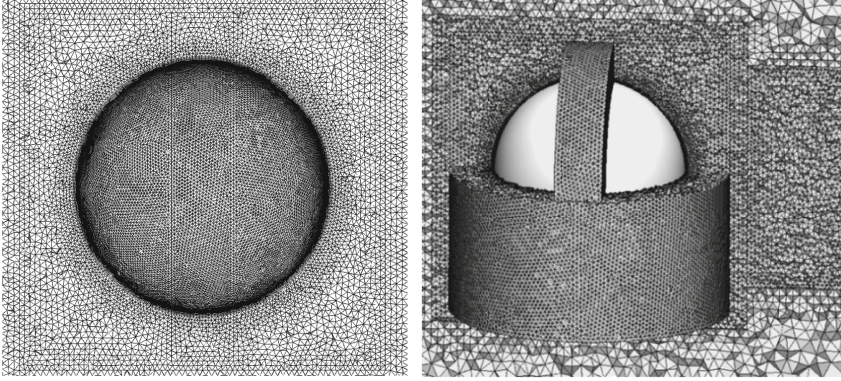
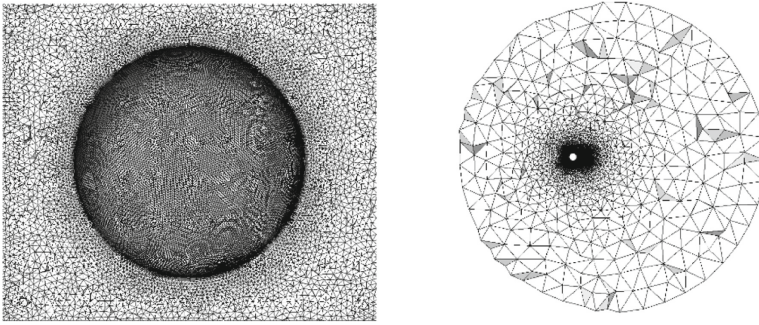**Fig. 9.** Case 1: Multi axes sliding mesh (Left: atmosphere, Right: sliding cylinders)



**Fig. 10.** Case 2: Single mesh (Left: atmosphere, Right: whole mesh)

**Table 1** Conditions to verify the interpolation on the sliding mesh surface in consideration of viscosity

| Name | Symbol | Value |
|---|---|---|
| *Initial conditions* | | |
| Density | $\rho$ | 1.0 |
| Velocity (*x*-direction) | $u$ | 0.1 |
| Velocity (*y*-direction) | $v$ | 0.0 |
| Velocity (*z*-direction) | $w$ | 0.0 |
| Pressure | $p$ | $1.0/\gamma$ |
| *Other conditions* | | |
| Time step size | $\Delta t$ | 0.001 |
| Reynolds number | Re | 10,000 |
| Rotational speed of domains | $\omega_1, \omega_2$ | 0.05, 0.03 |
| Radius (domain 1, domain 2) | $r_1, r_2$ | 0.7, 0.75 |
| Height (domain 1, domain 2) | $h_1, h_2$ | 0.25, 1.5 |

Computational conditions are shown in Table 1. The rotations of domains 1 and 2 in case 1 are dominated by Eqs. (23) and (24), respectively. Therefore, while both domain 1 and its axis rotates, only domain 2 rotates and its axis remains fixed.

$$
\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos \omega t & -\sin \omega t & 0 \\ \sin \omega t & \cos \omega t & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \omega t & -\sin \omega t \\ 0 & \sin \omega t & \cos \omega t \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad (23)
$$

$$
\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \omega t & -\sin \omega t \\ \sin \omega t & \cos \omega t \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (24)
$$

### 4.3 Computational Result

Figure 11 shows the conditions of the sliding mesh as the result. The sliding motion was confirmed to have no skewed and crushed elements. Under the sliding mesh environment, a flow around a sphere was computed. Figure 12 shows the velocity contours. Around the sphere, the flow in case 1 corresponded reasonably well with that in case 2. Thus, highly accurate interpolation was seen on the sliding surface, confirming that the sliding mesh did not affect the flow.

The pressure drag coefficient of the sphere surface in case 1 was compared with that of case 2, other calculation results [12], and experimental results [13] as shown in Table 2. The discrepancy between case 1 and other calculation results is around 1.0%. Furthermore, the deviation from the experimental results is less than 3.0%, which also shows the validity of the sliding approach. The discrepancy between case 2 and the other calculation and experimental results is larger than case 1 despite no moving and sliding mesh around the sphere. This is possibly due to the cylindrical sliding domain potentially generating a regular mesh.

Figure 13 shows the averaged pressure drag coefficient of case 1 and case 2 on a sphere surface. As the flow is unsteady, the time-averaged drag coefficient is used. Both match in front of the sphere, but there is a slight difference in wake. In general, a complicated flow containing vortices occurs behind a sphere. Thus, the mesh behind the sphere should be generated delicately. However, interpolation between the first layer of the static mesh and sliding mesh might affect such a sensitive flow.
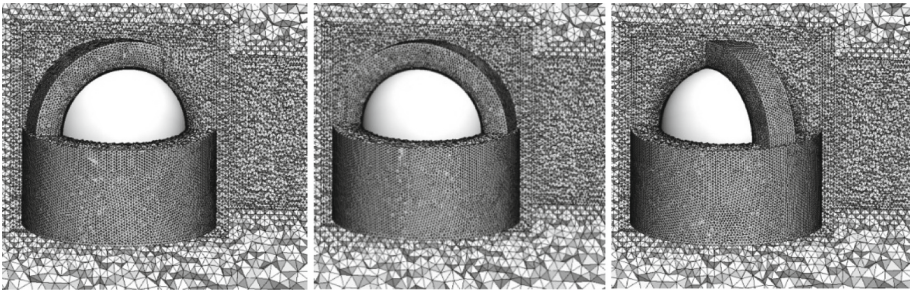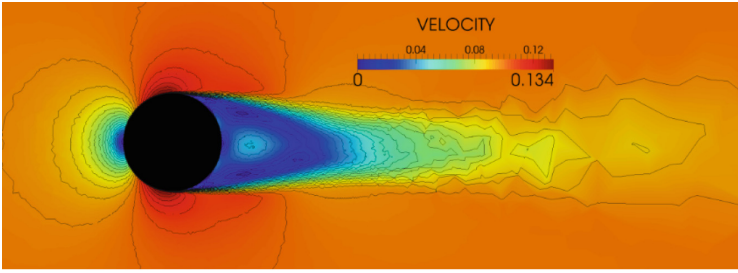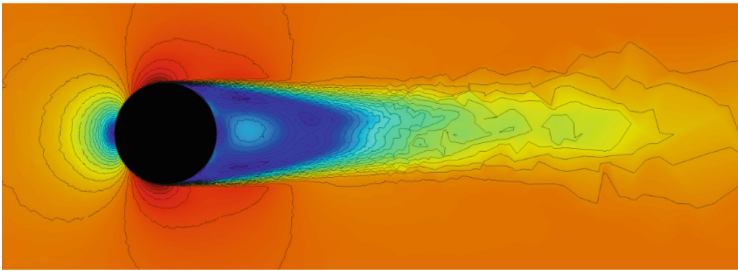


**Fig. 11.** Conditions of the sliding mesh

Case 1

Case 2

**Fig. 12.** Velocity contours around sphere

**Table 2.** Drag coefficient of sphere

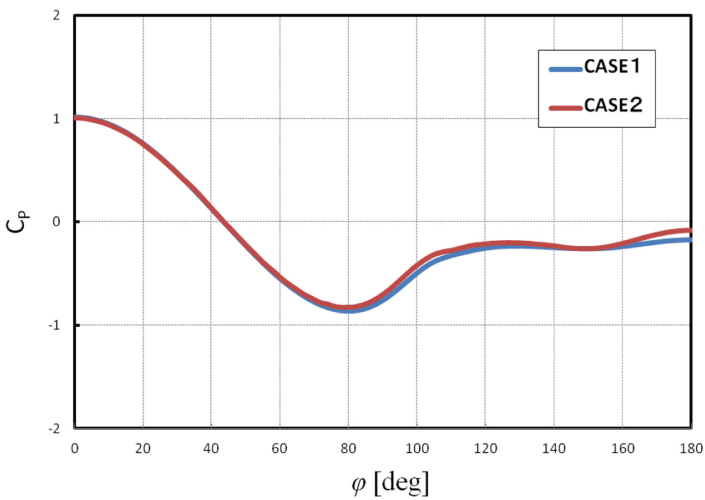|  | Case 1 | Case 2 | Calculated value | Experimental value |
|---|---|---|---|---|
| Drag coefficient | 0.389 | 0.379 | 0.393 | 0.40 |



**Fig. 13.** Averaged pressure drag coefficient on sphere surface

# 5   Conclusion

In this paper, the multi axes sliding mesh approach for compressible viscous flows was formulated. In particular, the interpolation process between prism elements on sliding surfaces was described. As a result of the computation of the flow around a sphere, the sliding motion of multiple cylinders without skewed and crushed elements were confirmed. The results also showed that there the sliding mesh had no affect on flow. A comparison of other experimental and computational results showed the validity of the multi axes sliding mesh approach. This approach could potentially be applied to complicated motions like a bicycle rider is computing.

# References

1. Murayama, M., Nakahashi, K., Matsushima, K.: Unstructured dynamic mesh for large movement and deformation. AIAA Paper 2002-0122 (2002, to appear)
2. Bakker, A., et al.: Sliding mesh simulation of laminar flow in stirred reactors. Chem. Eng. Res. Des. **75**(1), 42–44 (1997)
3. Chaderjian, N.M., Field, M.: Advances in rotor performance and turbulent wake simulation using DES and adaptive mesh refinement. In: 7th International Conference on Computational Fluid Dynamics (2012)
4. Ying, Z., Liang, Y., Shuo, Y.: Numerical study on flow fields and aerodynamics of tilt rotor aircraft in conversion mode based on embedded grid and actuator model. Chin. J. Aeronaut. **28**(1), 93–102 (2015)
5. Takii, A., Yamakawa, M., Asao, S., Tajiri, K.: Six degrees of freedom numerical simulation of tilt-rotor plane. In: Rodrigues, J.M.F., Cardoso, P.J.S., Monteiro, J., Lam, R., Krzhizhanovskaya, V.V., Lees, M.H., Dongarra, J.J., Sloot, P.M.A. (eds.) ICCS 2019. LNCS, vol. 11536, pp. 506–519. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22734-0_37
6. Yamakawa, M., et al.: Unstructured moving-grid finite-volume method for unsteady shocked flows. J. Comput. Fluids Eng. **10**(1), 24–30 (2005)
7. Obayashi, S.: Freestream capturing for moving coordinates in three dimensions. AIAA J. **30**, 1125–1128 (1992)
8. Roe, P.L.: Approximate Riemann solvers parameter vectors and difference schemes. J. Comput. Phys. **43**, 357–372 (1981)
9. Venkatakrishnan, V.: On the accuracy of limiters and convergence to steady state solutions. AIAA Paper, 93-0880 (1993)
10. Asao, S., et al.: Simulations of a falling sphere with concentration in an infinite long pipe using a new moving mesh system. Appl. Thermal Eng. **72**, 29–33 (2014)
11. Ito, Y.: Challenges in unstructured mesh generation for practical and efficient computational fluid dynamics simulations. Comput. Fluids **85**, 47–52 (2013)
12. Constantinescu, G.S., Squires, K.D.: LES and DES investigations of turbulent flow over a sphere at Re = 10,000. Flow Turbul. Combust. **70**, 267–298 (2003)
13. Achenbach, E.: Experiments on the flow past spheres at very high Reynolds numbers. J. Fluid Mech. **54**, 565–575 (1972)