



Analysis of the effects of dynamic mesh update method on simulating indoor airflow induced by moving objects

Huayuan Zhou, Ke Zhong, Hongwei Jia, Yanming Kang*

School of Environmental Science and Engineering, Donghua University, Shanghai, 201620, China



ARTICLE INFO

Keywords:
 CFD analysis
 Moving object
 Dynamic mesh
 Dynamic mesh update method
 Enclosed environment
 Computational performance

ABSTRACT

Local interference flow induced by the movement of human bodies or objects indoors can directly affect breathing air quality. However, very few studies have been carried out on the flow induced by moving objects due to high experimental costs. As an alternative, computational fluid dynamic (CFD) method has been employed in predicting the flow induced by moving objects indoors. During the numerical computations, the mesh needs to be updated continuously for the moving objects behave as part of the boundary. Although the dynamic mesh update (DMU) methods have been used in the CFD method, the performances of these DMU methods vary widely, and it is still lack of analysis on the selection of the DMU methods. In this study, the DMU methods provided by ANSYS FLUENT are used for comparing the numerical performances between computational and experimental results. Five aspects of numerical performance, accuracy, mesh quality, mesh generation complexity, continuity, and computational efficiency, are discussed integrally. The results show the dynamic layering (DL) method is preferred for the case of linear movement due to the high computational efficiency and accuracy though it is difficult and time-consuming in mesh generation. Spring-based smoothing & remeshing (SS&R) method can be used when the complicated-shaped or nonlinearly moving objects are part of the boundary. Both computational efficiency and accuracy of overset mesh (OM) method are low when the moving objects are clinging to the boundary, so this method is not recommended for simulating the flow fields.

1. Introduction

Indoor air quality is significantly affected by the airflow field in interior spaces. Although building ventilation system determines the overall removal efficiency of indoor pollutants, the airflow induced by the movement of human bodies or objects affects the local characteristics of indoor pollutant diffusion, which is related to the air quality for human breathing [1–3]. For example, previous studies, including airline cabin [4], operating room [5], isolation room [6], office [7] and residential kitchen [8], showed that moving human bodies can generate local airflow, and then the induced wake will promote the diffusion and spread of contaminants. This indicates the impact of the local flow produced by moving human bodies on indoor air quality should not be ignored. However, few investigations have been conducted on the movement of human bodies or objects because of the complicated arrangement of measurement points, difficult control of precise experimental conditions, and high cost of experimental facilities.

As an effective method, dynamic mesh technique in commercial CFD software has been employed to simulate a variety of air flows induced by

the movement of human bodies or objects in enclosed environments, i.e., moving boundary conditions need to be considered. Examples include human movements [9–12], upward and downward motions of foot in shoes [13], changing process of sheets on a patient's bed [14], and door opening and closing [15–17]. These previous works showed the numerical results complied with the experimental results from test chambers [18–20] or water tanks [21,22], and therefore dynamic mesh technique can be applied to the simulations of the local flow induced by moving objects or humans, as an alternative or complement to experimental studies.

However, there remains some disadvantages associated with dynamic mesh technique, such as mesh generating difficulty and high computational cost. In the simulation of the flow induced by moving objects, the mesh has to be continuously updated as the boundary of the computational domain moves, which is caused by the motion of the objects. Dynamic mesh update (DMU) methods with different theories and characteristics have been proposed in ANSYS FLUENT program for mesh update during computations. Once an inappropriate DMU method is used for the simulations, it may cause extreme deformation and

* Corresponding author.

E-mail address: ymkang@dhu.edu.cn (Y. Kang).

excessive skewness of cells, and even lead to negative volumes of cells, so the numerical results may be distorted or even the calculation may fail to complete. These computational risks are the main obstacles to the successful application of the dynamic mesh technique. It is still lack of enough attention and investigations to overcome these obstacles, and the differences in numerical performances between different kinds of DMU methods also have not been compared and analyzed.

The DMU methods have been used on the flow induced by moving objects in previous works, but few have addressed the specific reasons for the use of these method. For example, Luo et al. [21], Poussou et al. [22], Wang et al. [23] and Wu et al. [24] investigated the air flow induced by human movement by employing dynamic layering (DL) method, Chow et al. [25] conducted a study of the influence of a surgeon's periodic bending movement by using spring-based smoothing & remeshing (SS&R) method, and Tao et al. [26,27] analyzed human-induced wake flow characteristics by applying remeshing method. Overset mesh (OM) method is also provided in ANSYS FLUENT program, but it has not been used in any study for simulating indoor air flow.

Although more and more numerical studies have conducted on air environment in enclosures with moving boundary conditions, the characteristics and applicability of the DMU methods have not been further investigated so far. For those who intend to study the impact of indoor human walking on air quality by using DMU methods, suggestions on the selection of DMU methods are needed. In the present study, the numerical performances of the three DMU methods, including the computational efficiency, mesh generation complexity, computational accuracy, mesh quality, and continuity, are compared and analyzed by employing the experimental data in the literature [22,28], and some essential information and suggestions are then provided for the use of the dynamic mesh technique in the CFD simulations on enclosed air environments with moving boundaries.

2. Geometry and mathematical model

2.1. Experiment for validation and comparison

The movement of a person or an object in enclosed environments (e.g., office, residential building, industrial plant, and aircraft cabin) causes local air flow. Poussou et al. [22] built a water-based experimental system, including an upside-down cabin mockup, to investigate the flow field with a moving body in an airliner cabin. The detailed description of the experimental system and the test data for the validation and performance evaluation of the DMU methods can be found in Ref. [28]. Therefore, these experiments are employed as case studies in the present study to perform the numerical simulations by using the three DMU methods. Furthermore, to focus on the flow induced by moving body, only the case without ventilation, i.e., baseline case in Ref. [22], is performed in this study. The processes of computation by the three DMU methods are analyzed, and the numerical results are compared with the experimental results to assess the computational performance and

applicability of the three DMU methods in the dynamic mesh model.

The cabin model used by Poussou et al. [22] consisted of an optically-clear, acrylic, semi-cylindrical section of pipe of 244 cm in length, 45.7 cm in outer diameter, and 0.32 cm in thickness, as shown in Fig. 1(a). For simplicity, the exterior of the moving body had the shape of a parallelepiped solid of acrylic measuring 5.54 cm in length (L), 2.23 cm in width and 16.6 cm in height. The geometry of the numerical model is also shown in Fig. 1(a). The initial position of the moving body was 16.6 cm from one end of the model (the yellow box as shown in Fig. 1(b)), point O is the origin point (0, 0, 0), and line AB is the intersection line of $x/L = 18$ and $z/L = 1.5$ (half of the body height). The body was traversed by an automated device placed above the experimental facility along the axis direction (i.e., x direction) of the cabin at a speed of 17.5 cm/s. The temperature of the water was assumed to be isothermal so the buoyancy effects can be neglected, and the interior of the cabin model is free of obstructions to avoid inducing secondary flow effect. The Reynolds number (Re) is 9500.

2.2. Mesh generation and manipulation

To perform the numerical simulations of the flow with the moving boundary, a combined dynamic and static mesh scheme based on the range of movement are usually used in the CFD model to reduce the computing time required for updating solution and mesh at every time level. The interactions between the static and dynamic sections are performed by using interface technique. As the spatial domain (i.e., the dynamic section) occupied by fluid changes its shape with time, remeshing is required. ANSYS FLUENT 2019R1 provides spring-based smoothing method, remeshing method, dynamic layering (DL) method and overset mesh (OM) method. Generally, the spring-based smoothing method and remeshing method are used at the same time and are considered as a combination to cover the shortage of spring-based smoothing method and reduce the times of remeshing. Therefore, the numerical performances of three DMU methods (i.e., spring-based smoothing & remeshing (SS&R) method, DL method and OM method) are analyzed in this study. The geometry and mesh of the present CFD model were built by ANSYS ICEM 2019R1 software.

2.2.1. Model of the spring-based smoothing & remeshing method (Case 1)

The SS&R method, which was used by Chow et al. [25], is a combination of spring-based smoothing method and remeshing method. In the spring-based smoothing method, the edges between any two cell nodes are idealized as a network of interconnected springs, which can be compressed or stretched with the changes of computational domain [29]. The spring-based smoothing method can only adjust slightly the positions of cell nodes instead of the number of cell nodes, resulting that the cell quality can deteriorate, or the cells can become degenerate in the case of large boundary displacement. This will invalidate the cells (e.g., result in negative cell volumes) and consequently, will lead to convergence problems when the solution is updated to the next time level. To circumvent this problem, the remeshing method is used as a complement

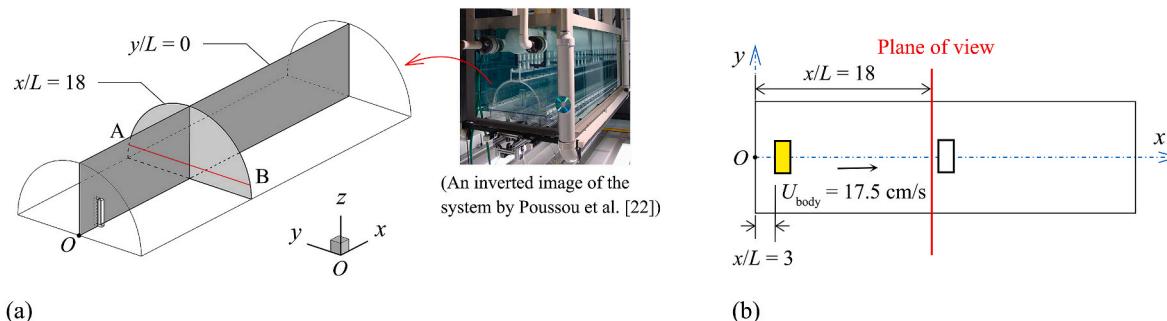


Fig. 1. Model description, (a) CFD geometry, (b) top-view of the computational domain.

to mark cells based on cell skewness and size criteria and update the old mesh with a new set of cells. In this work, the spring-based smoothing method and the remeshing method are also regarded as a combination and the detailed principles of these two methods can be found in Appendix A.1 and Appendix A.2.

In the following the experiments by Poussou et al. [22] and Poussou [28] are numerically simulated by using the methods mentioned above, and it is denoted as Case 1. As shown in Fig. 2, the total meshes of the computational domain (Fig. 2(c)) are split into a dynamic mesh section (Fig. 2(a)), close to the moving body and its trajectory, and a static mesh section (Fig. 2(b)), away from the moving body and its trajectory. The interactions between the static and dynamic sections are done using interface technique. In addition, for the same length scale, the quantity (quality) of structured hexahedral mesh is lower (higher) than the quantity (quality) of unstructured tetrahedral mesh, so tetrahedral and hexahedral meshes are employed in the dynamic and static mesh section, respectively.

2.2.2. Model of the dynamic layering method (Case 2a & Case 2b)

In principle, the DL method is to add or remove layers of cells adjacent to a static boundary based on the height of the layers by setting ideal cell height, layer split factor and layer collapse factor [29]. The equations of the DL method are presented in Appendix A.3.

Two cases shown in Fig. 3 are performed for this method in the present study, i.e., the computational domain only consists of dynamic mesh (Case 2a, Fig. 3(a)), and the domain contains a dynamic mesh section and a static mesh section (Case 2b, Fig. 3(b)). The size of the dynamic mesh section in Case 2b is much smaller than that in Case 2a, and the interface technique is employed in the simulation of Case 2b though it can reduce the accuracy of the flow predictions [30], so the influence of the interface technique on the predicted results can be analyzed by comparing the numerical results of Case 2a and Case 2b.

2.2.3. Model of the overset mesh method (Case 3)

For the OM method, see Fig. 4, the CFD model consists of two types of mesh, i.e., background mesh and component mesh. The background mesh contains the entire computational domain except for the moving body (Fig. 4(a)), whereas the component mesh (Fig. 4(b)) only includes the moving body. Overset interfaces connect the two types of mesh by interpolating cell data in the overlapping regions. In ANSYS FLUENT, the connectivity is established by three main steps: hole cutting, overlapping minimization, and donor search [29]. A detailed description can be found in Appendix A.4. The case by using this method is denoted as Case 3, and hexahedral cells were used for generating both the background and component meshes, as shown in Fig. 4.

2.2.4. Grid independency test

A grid-independency analysis was carried out by constructing a set of

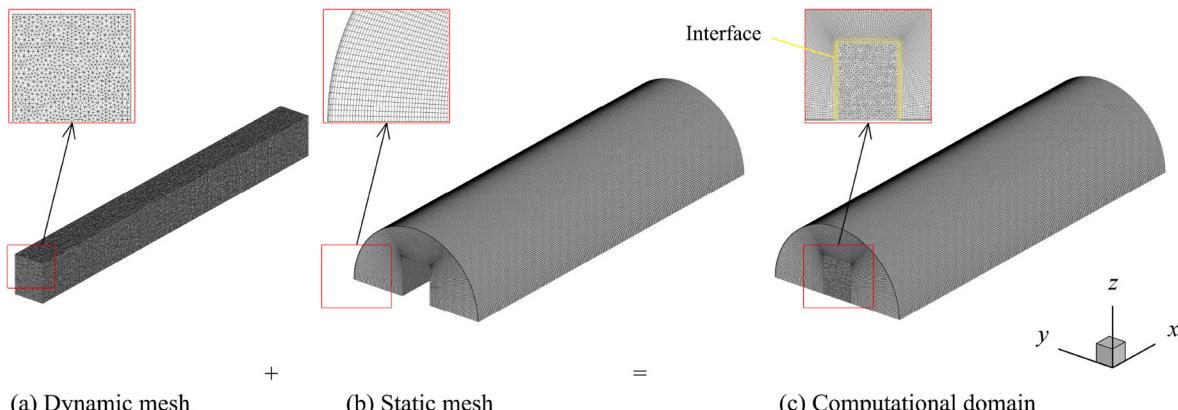


Fig. 2. Mesh generation of the spring-based smoothing & remeshing (SS&R) method.

grids with different grid numbers to ensure the accuracy of the computational results. The grid number is increased until the numerical results will not be affected by the grid size. Considering the high computational cost of the DMU methods, a root-mean-square error in velocity of less than 5% is regarded as the criterion for grid independence in the present study [31], i.e.

$$\sqrt{\frac{1}{N} \sum_1^N \left(\frac{U_{n-1} - U_n}{U_{body}} \right)^2} < 0.05 \quad (1)$$

where U_{n-1} is the velocity in the formal grid number, U_n is the velocity in the current grid number, U_{body} is the body velocity (m/s), and N is the number of the examined sample points. The final numbers of cells for the four cases are listed in Table 1.

2.3. Numerical methodology

2.3.1. Conservation equations of mass, momentum and transport

The flow field in the cabin model is determined by the conservation laws of mass and momentum, the water-based experimental system used by Poussou et al. [22] was isothermal and free of background flow, the flow caused by the moving body is assumed to be three-dimensional, incompressible and turbulent. When the body was moved, local flow was induced around the body, and the fluid far away from the body kept in static state. With the movement of the body, the fluid filled in the entire space was gradually induced by the moving body. The conservation of mass and momentum is referred to Ref. [32].

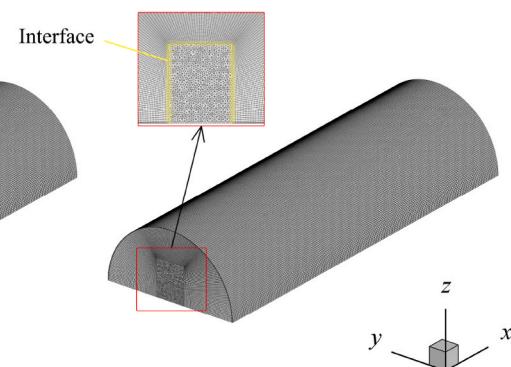
Considering that the computational cost of Large Eddy Simulation (LES) is too high, and the Reynolds-Average Navier-Stokes (RANS) model is used in most of the studies on moving human body [13,17,22, 33,34], the unsteady RANS model is employed to perform the simulations in this study. Compared to other RANS models, the Re-Normalization Group (RNG) $k-\epsilon$ turbulence model was found to deliver the best performance in terms of accuracy, computing efficiency, and robustness for modeling enclosed environments [35]. Therefore, the RNG model is applied in the present study. The transport equation can be found in Ref. [36].

2.3.2. Conservation equations of the dynamic mesh update method

When the dynamic mesh is employed, the integral form of the conservation equation for a general scalar ϕ , on an arbitrary control volume (V) with a moving boundary can be written as

$$\frac{d}{dt} \int_V \rho \phi dV + \int_{\partial V} \rho \phi \left(\vec{u} - \vec{u}_g \right) \cdot d\vec{A} = \int_{\partial V} \Gamma \nabla \phi \cdot d\vec{A} + \int_V S_\phi dV \quad (2)$$

where \vec{u} is the flow velocity, ρ is the fluid density, \vec{u}_g is the mesh velocity



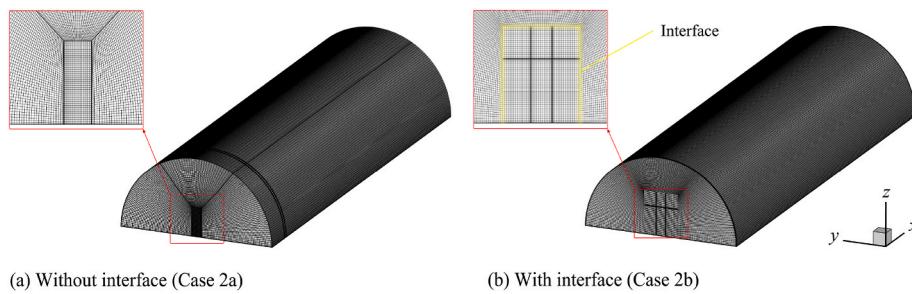


Fig. 3. Mesh generation of the domain by the dynamic layering (DL) method.

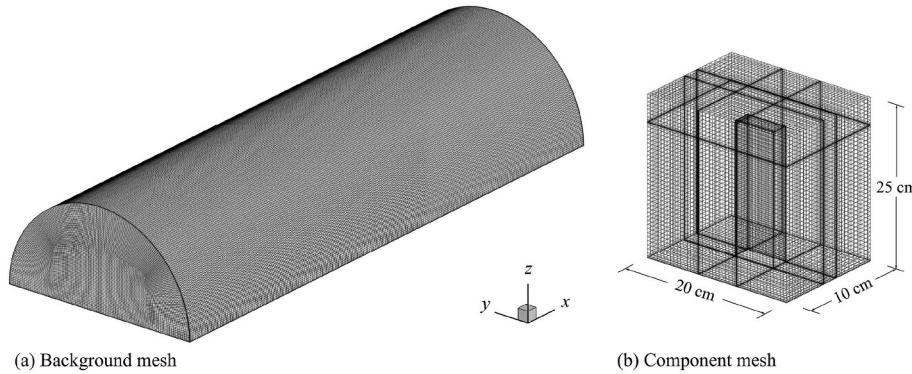


Fig. 4. Mesh generated for the overset mesh (OM) method.

Table 1
Number of cells of the four cases used in the simulations.

Case number	Mesh section	Number of cells
Case 1	Dynamic mesh section	4.15×10^6
	Static mesh section	5.06×10^6
Case 2a	Entire computational domain (Dynamic mesh)	6.77×10^6
Case 2b	Dynamic mesh section	1.83×10^6
	Static mesh section	5.06×10^6
Case 3	Background mesh	6.93×10^6
	Component mesh	0.14×10^6

of the moving mesh, Γ is the diffusion coefficient, S_ϕ is the source term of ϕ . Particularly, ∂V is used to describe the boundary of the control volume V by using a second-order backward difference formula to calculate the first-order derivative with respect to the time of the equation, so the first term on the left-hand side of Eq. (2) can be written as

$$\frac{d}{dt} \int_V \rho \phi dV = \frac{3(\rho \phi V)^{n+1} - 4(\rho \phi V)^n + (\rho \phi V)^{n-1}}{2\Delta t} \quad (3)$$

where $(n+1)$, n , and $(n-1)$ denote the respective quantities from successive time levels with $(n+1)$ denoting the current time level.

The $(n+1)$ th time level volume V_{n+1} can be computed by

$$V^{n+1} = V^n + \frac{dV}{dt} \Delta t \quad (4)$$

where dV/dt is the volume time derivative of the control volume.

Furthermore, to satisfy the mesh conservation law, dV/dt is computed from

$$\frac{dV}{dt} = \int_{\partial V} \vec{u}_g \cdot d\vec{A} = \sum_j^{nf} \vec{u}_{g,j} \cdot \vec{A}_j \quad (5)$$

where nf is the number of faces on the control volume and \vec{A}_j is the j face

area vector.

The dot product $\vec{u}_{g,j} \cdot \vec{A}_j$ on each control volume face is calculated from

$$\left(\vec{u}_{g,j} \cdot \vec{A}_j \right)^{n+1} = \frac{3}{2} \left(\vec{u}_{g,j} \cdot \vec{A}_j \right)^n - \frac{1}{2} \left(\vec{u}_{g,j} \cdot \vec{A}_j \right)^{n-1} = \frac{3}{2} \left(\frac{\delta V_j}{\delta t} \right)^n - \frac{1}{2} \left(\frac{\delta V_j}{\delta t} \right)^{n-1} \quad (6)$$

where $(\delta V_j)^n$ and $(\delta V_j)^{n-1}$ are the volumes swept out by control volume faces at the current and previous time levels over a time step.

2.3.3. Numerical scheme

The computations were performed by using the general purpose CFD code ANSYS FLUENT 2019R1, and RNG $k-\epsilon$ model was used to model the flow field. The discretization schemes used were the second-order upwind scheme for momentum and the convection and diffusion terms, the Coupled algorithm is used for pressure-velocity coupling. The Body Force Weighted algorithm is used as the pressure interpolation method. Enhanced wall function is employed to treat the turbulence flow properties of the near-wall region. Moreover, all simulations are ensured $y^+ < 5$, such that the near wall region was sufficiently resolved to justify the use of RNG $k-\epsilon$ model with enhanced wall function.

The time step (Δt) is related to the Courant number [37] and the minimum time step requirements in dynamic mesh model. Courant number, Co , is a non-dimensional number used in CFD simulations to evaluate the time step requirements of a transient simulation for a given mesh size. It is defined by

$$Co = \frac{U_{body} \cdot \Delta t}{\Delta s} \quad (7)$$

where Δs is the characteristic size of the cells. In this study, $Co < 1$. In dynamic mesh model, the time step must be smaller than the ratio of the moving object's speed to the characteristic size of the cells, which is the minimum time step requirement. The time step was set to 0.001 s in this study, and a total of 5640 time steps were performed. Both case and data were automatically saved every 10 time steps. The iterations continued until the residual errors in the conservation equations were below 10^{-6} .

and the velocity magnitudes became constant at the observation points. All the cases were computed continuously in a 32-node workstation (CentOS 7) to eliminate the potential affecting factors of the computational time and results.

3. Results and discussion

The flow fields in the cabin model are simulated by using different DMU methods, and the velocity and streamline distributions on the mid-plane $y/L = 0$ (Fig. 1) at $t = 2$ s and $t = 5$ s are presented in Fig. 5, where u_x , u_y and u_z are the velocity components in the x , y and z directions and normalized by the velocity of the moving body $U_{\text{body}} = 17.5$ cm/s.

It can be seen from Fig. 5 that during the movement of the body, the fluid in the front of the body was pushed out and drawn around the body entraining it into the wake. The dimensionless velocities in the wake regions were the largest reaching 1.4 times of the velocity of the body. As the body moves, flow separation occurs behind it and a small vortex was generated in this region (see Fig. 5(a)–5(c) at $t = 5$ s), and the results show that the flow fields are similar by using the SS&R method (Case 1) and the DL method (Cases 2a and 2b), while for the OM method (Case 3), as shown in Fig. 5(d), both the flow fields and the streamlines are different from those of Cases 1, 2a, and 2b, i.e., compared with Cases 1, 2a and 2b, the wake region at $t = 2$ s is larger and one more vortex are generated at $t = 5$ s in Case 3.

Therefore, the numerical results of the three DMU methods are compared with the experimental results of Poussou et al. [22] and Poussou [28] in the following sections. Additionally, the three DMU methods may spend different time in computing the cases by using the same workstation, so the numerical performances of the three DMU methods will be analyzed to find out the most appropriate DMU method for predicting the flow with the movement of objects in enclosed spaces.

3.1. Computational performance of different dynamic mesh update methods

3.1.1. Accuracy

The velocity distributions on the vertical plane at $x/L = 18$ (see Fig. 1), for $t = 4.80$ s, 5.22 s, and 5.64 s, were measured by particle image velocimetry (PIV) technique with the measuring error below 0.4% [28]. Fig. 6 shows the comparison between the experimental and computational results (taking Case 2a, i.e., DL method without interface, as an example), in which En ($n = 1, 2, 3$) are the experimental results by Poussou [28], while Cn ($n = 1, 2, 3$) are the computational results by this study, and $n = 1, 2, 3$ represent the time instants, 4.80 s, 5.22 s, and 5.64 s, respectively.

Fig. 6 shows that the present simulations agree well with the test

results by Poussou [28], and strong downwash local flows are formed behind the moving body. Moreover, just like those measured by PIV [28], a pair of counter-rotating vortices (i.e., horseshoe vortices) can be observed in both numerical and experimental results. The vortices develop very rapidly from $t = 4.80$ s to $t = 5.64$ s, and the downwash strikes the ground and both vortices separate from each other in opposite directions. The comparison between group C with group E in Fig. 6 indicates that the changes of the velocity distributions can be well predicted by the numerical simulations. Therefore, the CFD model used in this study is appropriate to capture the transient details of the flows in an enclosed space with moving boundaries.

Fig. 7 compares the velocity components, which are shown as dimensionless forms u_y/U_{body} , u_z/U_{body} and u_{yz}/U_{body} , between the computational results and the experimental results [28,38] along line AB (see Fig. 1(a)) at $t = 5.22$ s and $t = 5.64$ s.

As can be seen from Fig. 7, most of the numerical results agree well with the experimental results. Especially for Cases 2a and 2b (DL method), the numerical results by using this method are in good agreement with the measured data. For Case 1 (SS&R method) and Case 3 (OM method), however, evident discrepancies between the simulations and the measured data can be found in Fig. 7(c), because the values of u_z/U_{body} are underestimated (Fig. 7(b)) by the CFD method. The data interpolation between the component and background mesh in Case 3 and the mesh quality in Case 1 may be the reasons for the notable differences.

Fig. 8(a) shows the absolute errors (σ) of u_{yz}/U_{body} along line AB at $t = 5.64$ s (i.e., including the data in Fig. 7(c)-(ii)). Fig. 8(b) shows the absolute error frequency (f) distribution at $t = 5.22$ s and $t = 5.64$ s on the same location as Fig. 8(a) (i.e., including all the data in Fig. 7).

As shown in Fig. 8(a), except for Case 3 (OM method), the absolute errors of u_{yz}/U_{body} of the other cases stay within ± 0.3 . The absolute errors of Case 1 (SS&R method) are higher than those at other positions from $y/L = 1$ to $y/L = 2$. This is possibly caused by the poor mesh quality (even after remeshing). The absolute errors of Case 2b (DL method) near the position of $y/L = 0$ are about twice higher than those of Case 2a by using the same method, which results from the data interpolation by interface technique.

Fig. 8(a) also shows that absolute errors of Case 3 are the most significant in all the cases. There are two primary reasons causing this error: one reason is the data interpolation during the computation between the component and background mesh; the other is that the moving body clings to the ground (i.e., boundary), leading to the insufficient overlap between the background and component mesh. The results suggest that the OM method may be inappropriate for the moving body clinging to the boundary.

When $\sigma = 0$, the results in Fig. 8(b) show that the frequencies are in

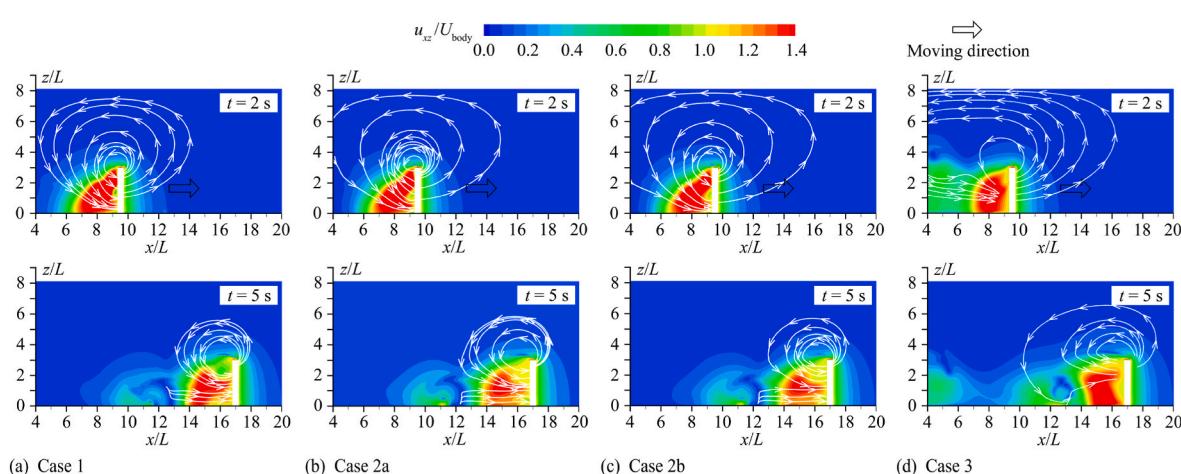


Fig. 5. Normalized flow fields and streamline of the aircraft cabin on the mid plane $y/L = 0$ at $t = 2$ s and $t = 5$ s ($u_{xz}/U_{\text{body}} = (u_x^2 + u_z^2)^{-0.5}/U_{\text{body}}$).

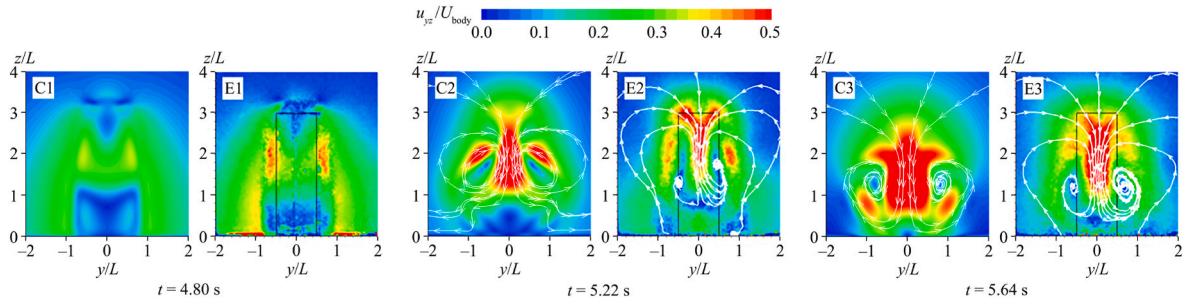


Fig. 6. Comparison of the streamline and dimensionless velocity distributions between the experimental and the numerical results ($u_{yz}/U_{body} = (u_y^2 + u_z^2)^{-0.5}/U_{body}$).

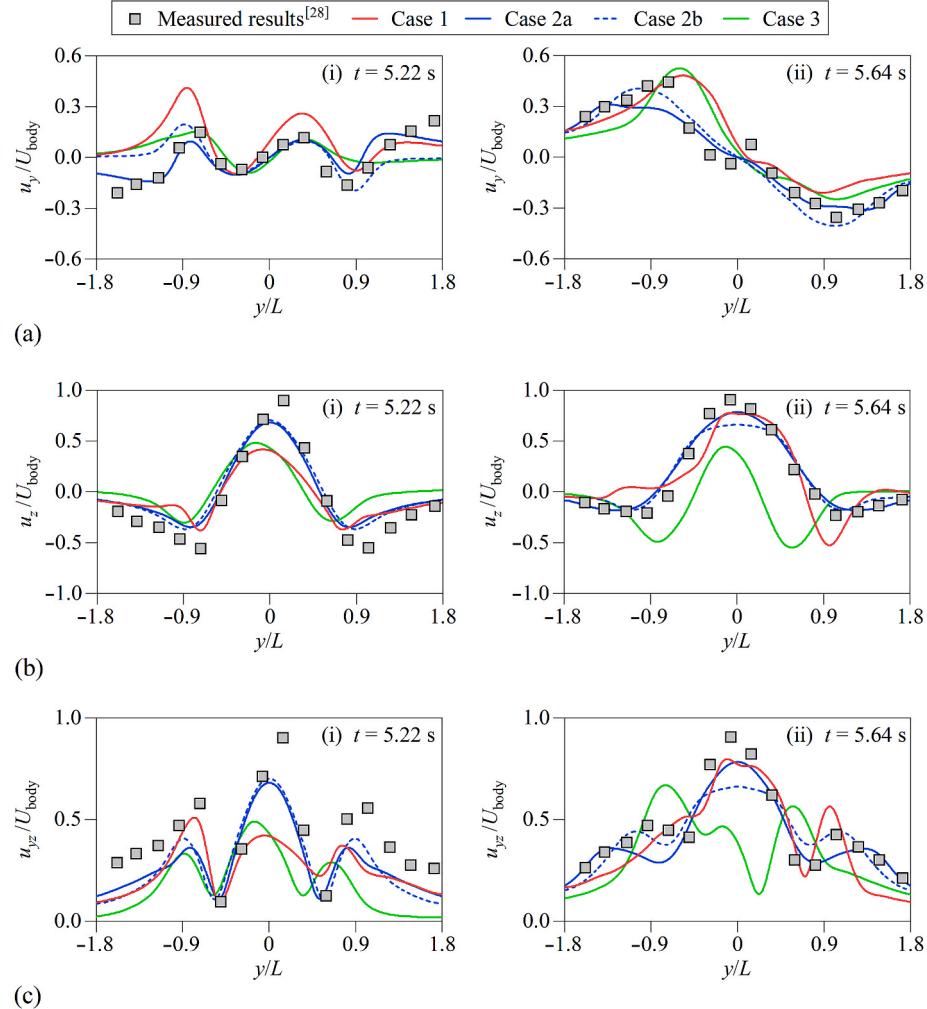


Fig. 7. Comparisons of the velocity components between the computational and experimental results on line AB at $t = 5.22$ s and $t = 5.64$ s: (a) u_y/U_{body} , (b) u_z/U_{body} , and (c) u_{yz}/U_{body} .

descending order, i.e., 36% (Case 2a), 34% (Case 2b), 25% (Case 1), and 21% (Case 3). In addition, the absolute error frequency distribution of Case 3 is symmetrical about $\sigma = -0.1$ instead of $\sigma = 0$, which is quite different from the other three distributions, indicating that the velocity calculated in Case 3 were generally smaller than the experimental data.

3.1.2. Mesh quality

Mesh quality plays a significant role in the accuracy and stability of numerical simulations. One indicator of mesh quality provided in ANSYS Fluent is orthogonal quality (OQ, hereinafter referred to as η_{OQ}). OQ depends on cell type: (i) for hexahedral and polyhedral cells, the OQ

is the same as the orthogonality, and (ii) for tetrahedral, prism, and pyramid cells, the OQ is the minimum of the orthogonality and skewness quality. The skewness quality (η_{sq}) is calculated by

$$\eta_{sq} = 1 - \eta_{cs} \quad (8)$$

where η_{cs} is the value of cell skewness, defined as the difference between the shape of the cell and the shape of an equilateral cell of equivalent volume. Highly skewed cells can decrease accuracy and destabilize the solution.

To determine the orthogonality of a cell, two quantities, η_{q1} and η_{q2} , are calculated for each face, as is given by

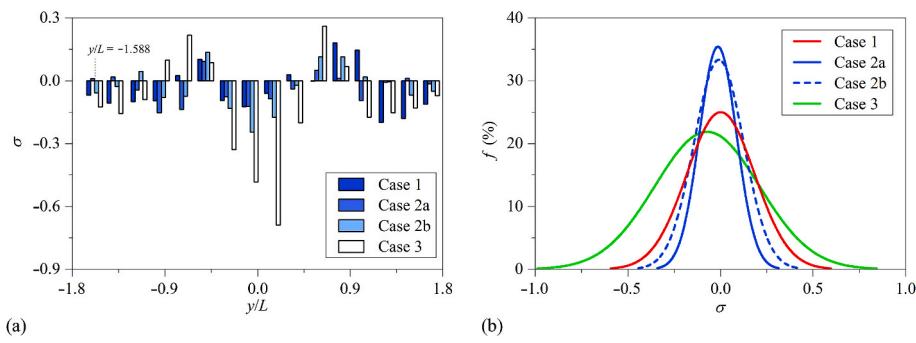


Fig. 8. Absolute error of dimensionless velocity along line AB, (a) absolute errors of dimensionless velocity components at $t = 5.64$ s, (b) absolute error frequency distribution at $t = 5.22$ s and $t = 5.64$ s.

$$\eta_{q1} = \frac{\vec{A}_i \cdot \vec{f}_i}{|\vec{A}_i| \cdot |\vec{f}_i|} \quad (9)$$

where \vec{A}_i is the normalized dot product of the area vector of a face, and \vec{f}_i is a vector from the centroid of the cell to the centroid of that face.

$$\eta_{q2} = \frac{\vec{A}_i \cdot \vec{c}_i}{|\vec{A}_i| \cdot |\vec{c}_i|} \quad (10)$$

where \vec{c}_i is a vector from the centroid of the cell to the centroid of the adjacent cell that shares that face.

The minimum value that results from calculating Eq. (9) and Eq. (10) for all the faces is then defined as the orthogonality of the cell. The worst cells will have an orthogonality closer to zero and the best cells will have an orthogonality closer to unity.

Fig. 9 shows the statistical characteristics of OQ (η_{OQ}) in each mesh section. Neither the background and component meshes in Case 3 nor the static mesh sections of Case 1 and Case 2b vary with time, so the OQ values in these sections are depicted in the subfigure (in gray) on the left of Fig. 9.

It can be found from Fig. 9 that the mesh of Case 3 consists of a background mesh and a component mesh. The mesh involved in the computation at each time step is determined by the overset mesh algorithm, so the mesh quality maintained constant during the computation. The OQ values of the two mesh sections are at a high value (see the subfigure in Fig. 9), and $\eta_{OQ,75\%}$ of the background mesh and component mesh are 0.98 and 1.0, respectively. The mesh of Case 2b contains a dynamic mesh section and a static mesh section. The mesh quality in the static mesh section is stationary and high (see the subfigure). The OQ values of the mesh in the dynamic mesh section are unity (the maximum value) during the computation. Only dynamic mesh exists in Case 2a. $\eta_{OQ,95\%}$ of the dynamic mesh quality is 0.88, slightly worse than the dynamic mesh quality of Case 2b. Similar to Case 2b, the mesh of Case 1 is composed of a dynamic mesh section and a static mesh section. The static mesh quality is the same as that of Case 2b (also see the subfigure). The dynamic mesh quality at the initial state ($t = 0$) is obviously worse

than the other three cases ($\eta_{OQ,95\%} = 0.63$), and the mesh quality dropped sharply once the object moves ($\eta_{OQ,75\%}$ changed from 0.91 to 0.62). Subsequently, it continues to decrease gradually ($\eta_{OQ,25\%}$ changed from 0.88 to 0.80).

To further analyze the deterioration of the OQ in the dynamic mesh section of Case 1, the OQ values of the cells near the moving body (within 5 cm from the front and back of the moving body) in the dynamic mesh section are shown in Fig. 10.

The OQ values of the cells near the moving body decrease sharply at the beginning of the computation and roughly remain constant during the computation process at the low level of $\eta_{OQ} = 0.65 \pm 0.15$ (see Fig. 10). This explains the inaccuracy of Case 1, i.e., low-quality cells result in inaccurate solutions. In addition, the low-quality cells near the moving body can gradually diffuse to the entire dynamic mesh section as the boundary moves, so the OQ values of the cells in the dynamic mesh section of Case 1 (see Fig. 9) decrease continuously with the computation process.

3.1.3. Computational efficiency

Because of the differences in principles between the three DMU methods, the computational time required by the three DMU methods are also different. In this study, the physical time was recorded every ten time-steps to analyze the numerical efficiency of the three DMU

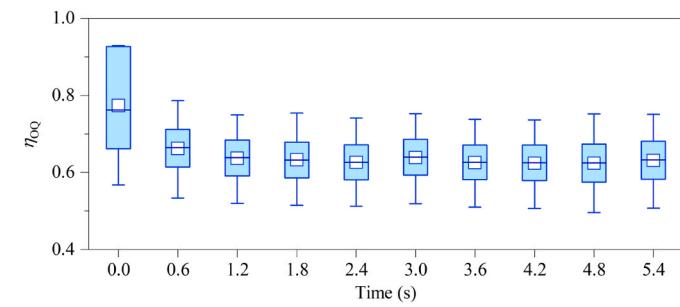


Fig. 10. Temporal variation of orthogonal quality near the moving body in the dynamic mesh section of Case 1.

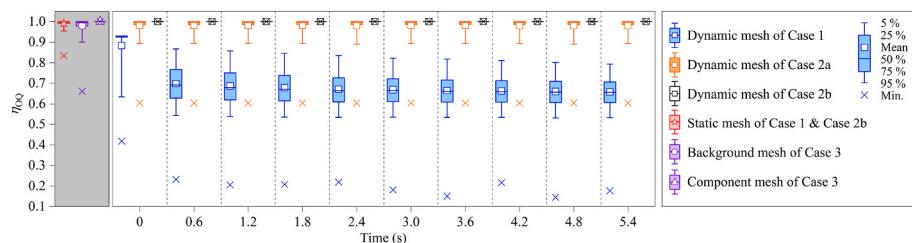


Fig. 9. Statistical characteristics of orthogonal quality in each mesh sections (Min. represents the minimum value).

methods. Fig. 11 shows the relationship between the completion rate (φ) and the computational time cost in the simulation by using the three DMU methods. The sharp and high-frequency fluctuations in Fig. 11(a) are caused by the varying time spent saving data.

It can be seen from Fig. 11 that the computational cost of Case 3 is higher than those of Cases 2a and 2b. In particular, the computational time for each ten time-steps (t_c) of Cases 2a and 2b are similar and time-independent, as shown in Fig. 11(a). It should be noted that the number of cells in dynamic mesh section is unrelated to the computational time by using DL method, because the cells in the dynamic mesh section of Case 2a are about five million more than those of Case 2b (see Table 1).

The reason for the long computational time of Case 3 is that the time-consuming data interpolation between the component and background meshes is required every iteration. The value of t_c in Case 1 is proportional to the time increment, which may be attributed to two reasons: one is the continuous increase in the number of cells due to the remeshing during the computation, the other is the frequent remeshing required for maintaining a good mesh quality near the moving body where the OQ values (η_{OQ}) are the lowest as mentioned in Section 3.1.2. Particularly, the irregular and sudden increase of t_c in Case 1 (see Fig. 11(a)) may result from the long time remeshing.

Fig. 11(b) shows that Cases 2a and 2b take about 12000 core-hour (a unit of work indicating that an application ran for a time t_r on p_c cores, such that computational time $t_{ct} = p_c \cdot t_r$ core-hour) to complete the calculation. It is noted that the accumulative computational time of Case 1 and Case 3 are 1.5 times and over 2 times longer than those of Cases 2a and 2b. Therefore, the numerical efficiencies of DL method (Case 2a and Case 2b) are much higher than those of SS&R method (Case 1) and OM method (Case 3).

3.1.4. Continuity

Previous studies [6,22] have suggested that the computational time can be reduced by dividing the domain mesh into the dynamic and static mesh sections and interacting by using the interface technique. This strategy, however, may lead to the risk of discontinuities in velocity fields. Considering the interface technology is not used in Case 3, in the following this case will not be discussed. Fig. 13 shows the normalized velocity fields of the cabin model at $z/L = 1.5$ in Case 1, Case 2a and Case 2b at $t = 5.64$ s. The computational domains of Case 1 and Case 2b contain dynamic and static mesh sections, whereas the mesh in the entire domain of Case 2a is all dynamic mesh.

As shown in Fig. 12, the flow fields of Case 1 and Case 2b are obviously different from that of Case 2a, which is the most accurate among all the cases (as is mentioned in Section 3.1.1). Because the DL method is employed and the interface technology is not applied in Case 2a, no discontinuity can be found in the flow field (Fig. 12(b)), which suggests that this strategy is the most appropriate method to simulate the flow induced by moving objects. It is evident in Fig. 12(a) and (c) that the discontinuities exist at the interface (see the gray circles) in both Case 1 and Case 2b, because of the inevitable data interpolation for interactions between dynamic mesh section and static mesh section. Furthermore,

the flow field of Case 1 is asymmetry about $y/L = 0$ (see Fig. 12(a)) because the irregular cells (i.e., tetrahedral cells) are used in the dynamic mesh section. Therefore, the SS&R method seems not appropriate for the simulation with moving boundaries.

The dimensionless velocity curves along the spanwise direction (y direction) at $x/L = 18$ in Case 1, Case 2a and Case 2b at $t = 5.64$ s are shown in Fig. 13, where the short-dashed lines are the fitting results by cubic polynomial from the data in the non-discontinuity zone (from $y/L = -2.10$ to $y/L = -1.95$ and from $y/L = -1.65$ to $y/L = -1.50$).

The results in Fig. 13 indicate that the dimensionless velocity curves of the three cases generally show a similar pattern. As shown in Fig. 13 (b), the velocity variation in Case 2a exhibits good continuity in the whole flow field. However, as presented in Fig. 13(a) and (c), evident discontinuities are found near the interfaces (at $y/L = -1.805$) in Case 1 and Case 2b. This provides the direct evidence for the findings that the interface can reduce the accuracy of flow predictions [30].

To quantitatively analyze the discontinuity between the interfaces, the degree of discontinuity, δ , is introduced and defined by

$$\delta = \frac{|V_s - V_f|}{V_f} \times 100\% \quad (11)$$

where V_s is the velocity obtained from numerical simulation, and V_f is the velocity from the fitting curve. The degree of discontinuity along the spanwise direction (y direction) at $x/L = 18$ at $t = 5.64$ s in Case 2b is shown in Fig. 14.

It can be observed from Fig. 14 that the discontinuities of almost all the heights reach the peaks at the interface ($y/L = -1.805$) and the maximum of δ is about 6%, meaning that the sudden change in velocity accounts for 6% of the fitting velocity. Due to the gaps between the cell nodes on both sides of the interface, the data in both regions are interpolated and transferred. This leads to sudden changes of physical quantities (e.g., velocity) at the interface. On the other hand, the influence of discontinuity ranges from $y/L = -1.7$ to -1.9 . The discontinuity is less than 1% where $y/L > -1.7$ and $y/L < -1.9$.

3.1.5. Mesh generation complexity

Because of the specific requirements of mesh types in applications, DMU methods vary in the mesh generation complexity in the order from easy to difficult as follows: OM method (Case 3), SS&R method (Case 1), DL method with interface (Case 2b) and DL method without interface (Case 2a).

OM method (which has the lowest accuracy and computational efficiency) is the easiest method for mesh generation among the three DMU methods, because the component and background meshes can be independently generated. Comparing with the other methods, this method is easier to generate hexahedral mesh, whose mesh generation process is usually simple. In addition, only the connectivity of the component and background meshes is needed during the simulation, instead of smoothing and remeshing.

When SS&R method (Case 1) is employed, only tetrahedral mesh can

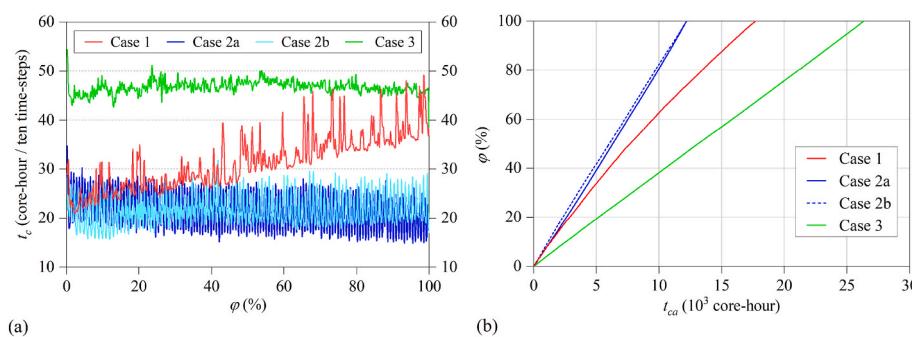


Fig. 11. Relationship between the completion rate (φ) and (a) computational time for each ten time-steps (t_c), (b) accumulative computational time (t_{ct}).

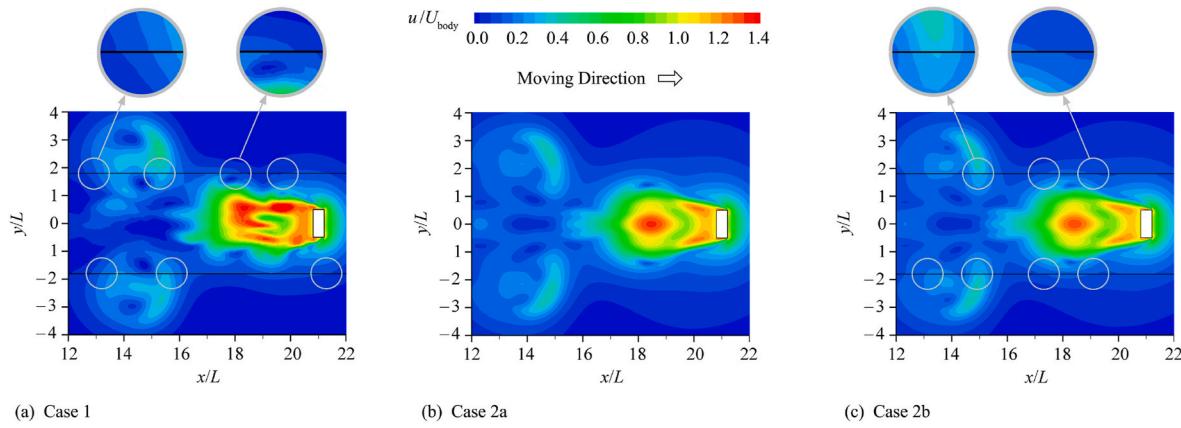


Fig. 12. Normalized velocity fields of the cabin model at $z/L = 1.5$ in (a) Case 1, (b) Case 2a, and (c) Case 2b at $t = 5.64$ s ($u/U_{\text{body}} = (u_x^2 + u_y^2 + u_z^2)^{-0.5}/U_{\text{body}}$).

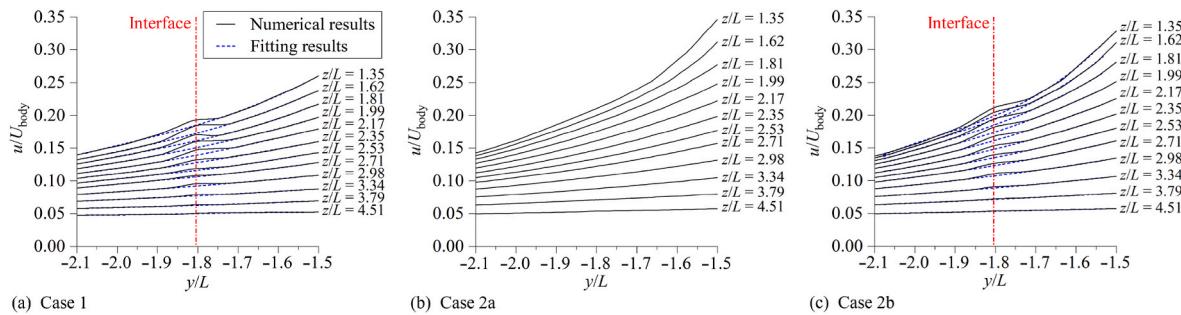


Fig. 13. Dimensionless velocity curves along y direction at $x/L = 18$ at $t = 5.64$ s: (a) Case 1, (b) Case 2a, (c) Case 2b.

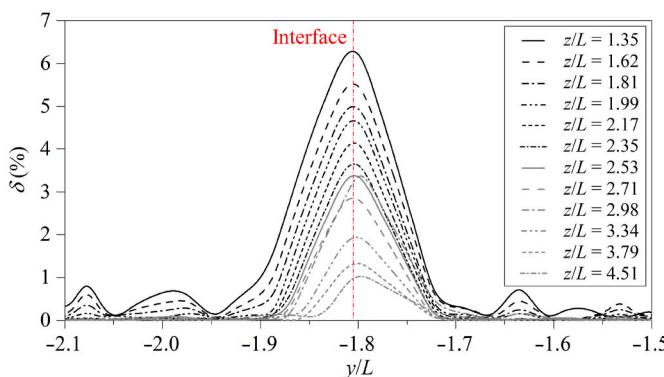


Fig. 14. Discontinuity distribution along y direction at $x/L = 18$ at $t = 5.64$ s in Case 2b.

be used in the dynamic mesh section. If tetrahedral mesh is generated in the entire computational domain, it will result in an extremely large number of cells which cannot be processed by common computers. Accordingly, the domain should be divided into two sections, dynamic mesh section and static mesh section. For the dynamic mesh section, tetrahedral mesh can be easily generated by embedded meshing algorithms. Hexahedral mesh is adopted in the static mesh section to reduce the number of cells to the level which can be operated by common computers. The generation of a block-based hexahedral mesh involves three main steps:

- (1) Define the block model through block splitting and modification, edge or face modification and vertex movements after importing a geometry file.

- (2) Associate the vertexes of the block model with the points of the geometry, the edges with the curves and the faces with the surfaces and assign the meshing parameters.
- (3) Generate the hexahedral mesh and check the mesh quality to ensure specified mesh quality criteria are met.

The process of hexahedral mesh generation is time-consuming and requires proficient skills. Therefore, the mesh generation complexity of this method is significantly higher than that of the OM method.

As for DL method (Cases 2a and 2b), only hexahedral mesh can be used in the dynamic mesh section, and a hexahedral mesh is also needed in the static mesh section to reduce the number of cells, so block models are required to generate the mesh in both static and dynamic mesh sections. If the interface technology is used (Case 2b), i.e., the domain is divided into two sections, the process of mesh generation will be the same as mentioned above (SS&R method), requiring the same proficient skills, but will double the time spent in building the block models. If the interface technology is not used (Case 2a), the boundaries of the computational domain will inevitably contain several concave surfaces, and the mesh generation process become more difficult. This will take investigators a very long time to complete the block model and then generate mesh, which requires more proficient skill and investigators' patience compared to those of the methods mentioned above and risk a failure in generating the mesh. Consequently, the mesh generation complexity of this method is the highest among the three DMU methods.

3.2. Analysis of applicability and suggestions of the dynamic mesh update methods

To compare the computational performances of the four cases in the above aspects (excluding the mesh generation difficulty), performance index, $A_{i,j}$, is introduced (a larger value of $A_{i,j}$ means a better numerical performance in the aspect), and calculated by

$$A_{ij} = \frac{a_{ij}}{a_{\max,j}} \quad (12)$$

where a_{ij} is the performance value of Case i in aspect j of numerical performance. The first subscript $i = 1, 2, 3$, and 4 represent Cases 1, 2a, 2b, and 3, respectively. $j = 1, 2, 3$, and 4 are denoted as continuity, accuracy, computational efficiency, and mesh quality, respectively. $a_{\max,j}$ is the maximum of a_{ij} in the aspect j of numerical performance. The evaluation criteria of a_{ij} are defined as follows:

- (1) $a_{i,1} = 1.0$ for the cases without discontinuity in the flow field and $a_{i,1} = 0.5$ for the cases with discontinuity in the flow field (i.e., using the interface technique).
- (2) $a_{i,2}$ is the absolute error frequency (f) when $\sigma = 0$ in Fig. 8(b).
- (3) $a_{i,3}$ is the completion rate (φ) of the cases at the time instant 10,000 core-h (see Fig. 11(b)).
- (4) $a_{i,4}$ is the worst OQ value (η_{OQ}) of the mesh section in the corresponding case during the simulation (see Fig. 9).

The performance indexes of the four cases ($A_{ij} = 1.0$ is the best and $A_{ij} = 0$ is the worst) are shown in Fig. 15.

From Fig. 15, it can be seen that the three DMU methods perform differently in various aspects, which are summarized as follows:

The performance indexes of DL method (Cases 2a and 2b) are high in most aspects, so the DL method is recommended to simulate the flows induced by the moving objects. It should be noted that this method is only applicable to the simulations with the linearly moving objects. In addition, due to the discontinuity problem, interface technology should be avoided if the investigator's skill is proficient enough, otherwise interface should be placed away from the critical zone in computational domain.

The mesh quality of the SS&R method (Case 1) is the worst among all cases, but the two performance indexes, computational efficiency and accuracy, are of intermediate level, as shown in Fig. 15. Consequently, this method can be used for the simulation of flow induced by complicated-shaped objects or nonlinearly moving objects. However, it should be noted that as the numerical simulation progresses, the calculation time for each ten time-steps of this method will gradually increase due to the increasing number of cells, and the mesh quality along the trajectory of the moving boundary will gradually decrease. This method may be not appropriate for the moving objects with large displacements. During the computations, close attention should be paid to the mesh quality near the moving boundary.

As for the OM method (Case 3), the performance indexes of computational efficiency and accuracy are the lowest among all cases (in particular, the accumulative computational time is twice longer than that of the DL method). Consequently, it is not recommended to apply this method in the case of the moving objects clinging to the boundary. This method might perform well if moving objects are not in contact with the domain boundary, which requires further research.

It should be noted that the accuracy and the computational efficiency of the DMU methods by using the RNG model may be different from those by using other turbulence models, and it requires to be further studied.

4. Conclusions

Indoor air quality is impacted by the flow field indoors to a large extent. Local airflow induced by the movement of humans or objects indoors can directly affect the breathing air quality. The high cost of experiments, however, has so far resulted in limited experimental studies on the flows induced by moving humans or objects. Recently, a growing number of investigations focused on the numerical simulations of the flows induced by moving objects in enclosed environments. During the computations, the mesh needs to be updated continuously, and the performances of the commonly used dynamic mesh update

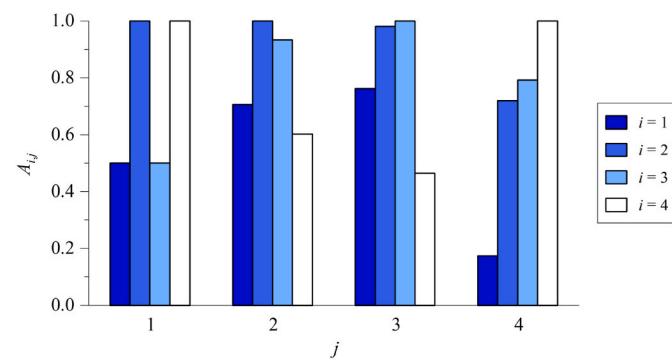


Fig. 15. Performance index of the dynamic mesh update methods.

(DUM) methods vary widely in several aspects, but few studies have discussed the characteristics of the three DMU methods.

Therefore, three DMU methods provided in ANSYS FLUENT are applied for analysis on numerical performance and comparison of computational results with measured results, and five aspects of the performance of the three DMU methods, i.e., mesh generation complexity, accuracy, mesh quality, computational efficiency and continuity, are discussed and analyzed in the present study. Recommendations on the selection of DMU methods are provided to those who intend to investigate the impact of indoor human walking on air quality by using DMU methods in the future. The main conclusions are as follows:

- (1) Dynamic layering (DL) method is the most efficient and accurate method for the computation of air flow caused by moving objects even though it is difficult and time-consuming to generate the mesh. This method has a high performance index in this study, so it is recommended for the objects moves linearly.
- (2) Although the mesh quality of spring-based smoothing & remeshing (SS&R) method near the moving boundary is relatively low, the efficiency and accuracy of computation are still acceptable. Hence, this method can be employed in the case of the complicated-shaped moving object or nonlinear trajectory (i.e., dynamic layering (DL) method is not applicable).
- (3) The computational cost of overset mesh (OM) method is very high (i.e., the calculation time is twice compared with that of the DL method), and the accuracy of this method is the worst in the current model among the three DMU methods. It is not recommended to apply OM method in the case with moving boundaries.

CRediT authorship contribution statement

Huayuan Zhou: Writing – original draft, Visualization, Validation, Methodology, Data curation, Conceptualization. **Ke Zhong:** Writing – review & editing, Resources, Project administration, Methodology. **Hongwei Jia:** Conceptualization. **Yanming Kang:** Writing – review & editing, Visualization, Supervision, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 42075179), the Science and Technology Commission of Shanghai Municipality (Grant No. 19DZ1205005), and the Chinese Universities Scientific Fund (Grant No. CUSF-DH-D-2020075). The

authors would like to thank Dr. Y. Gu for providing valuable comments on the manuscript.

Appendix A. Principles of dynamic mesh update method

A.1 Spring-based smoothing method

For spring-based smoothing method, the edges between any two mesh nodes are idealized as a network of interconnected springs. The initial spacings of the edges before any boundary motion constitute the equilibrium state of the mesh. A displacement at a given boundary node will generate a force proportional to the displacement along all the springs connected to the node. Using Hooke's Law, the force on a mesh node can be written as

$$\vec{F}_i = \sum_j^{n_i} k_{ij} (\Delta\vec{X}_j - \Delta\vec{X}_i) \quad (A.1)$$

where $\Delta\vec{X}_i$ and $\Delta\vec{X}_j$ are the displacements of node i and its neighbor j , n_i is the number of neighboring nodes connected to node i , and k_{ij} is the spring constant (or stiffness) between node i and its neighbor j . The spring constant for the edge connecting nodes i and j is defined as

$$k_{ij} = \frac{k_{\text{fac}}}{\sqrt{|\vec{X}_i - \vec{X}_j|}} \quad (A.2)$$

where k_{fac} is the value of spring constant factor, ranging from 0 to 1. \vec{X}_i and \vec{X}_j are the positions of node i and its adjacent node j . The parameter k_{fac} represents the stiffness of the entire spring system. In the case of $k_{\text{fac}} = 0$, the elastic damping of the entire spring system is 0 (i.e., no energy dissipation exists in the entire deformation when the cells are deformed). On the contrary, in the case of $k_{\text{fac}} = 1$, the mesh deformation will be concentrated at the boundary, which may cause the high mesh density at the boundary.

At equilibrium, the net force on a node due to all the springs connected to the node must be zero. This condition results in an iterative equation, as is given by

$$\Delta\vec{X}_i^{m+1} = \frac{\sum_j^{n_i} k_{ij} \Delta\vec{X}_j^m}{\sum_j^{n_i} k_{ij}} \quad (A.3)$$

where m is the iteration number.

Since displacements are known at the boundaries (after boundary node positions have been updated), Eq. (A.4) is solved using a Jacobi sweep on all interior nodes. At convergence, the positions are calculated and updated by

$$\vec{X}_i^{n+1} = \vec{X}_i^n + \Delta\vec{X}_i^{\text{converged}} \quad (A.4)$$

where $n+1$ and n are denoted as the positions at the next time step and the current time step, respectively.

A.2 Remeshing method

The remeshing method is regarded as a supplement to the spring-based smoothing method in the present study. In this method, cells that violate the skewness or size criteria are agglomerated in ANSYS FLUENT and locally remeshed. Each cell is evaluated in ANSYS FLUENT and marked for remeshing if it meets one or more of the following criteria:

1. It has a skewness that is greater than a specified maximum skewness.
2. It is smaller than a specified minimum length scale.
3. It is larger than a specified maximum length scale.

If the new cells or faces satisfy the skewness criterion, the mesh is locally updated with the new cells (with the solution interpolated from the old cells). Otherwise, the new cells are discarded, and the old cells are retained.

A.3 Dynamic layering method

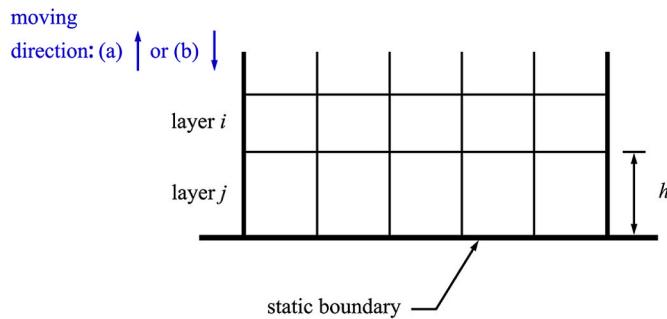


Fig. A.1. Dynamic layering method.

The layer of cells adjacent to the static boundary (layer j in Fig. A.1) is split or merged with the layer of cells next to it (layer i in Fig. A.1) based on the height (h) of the cells in layer j . If the cells in layer j are expanding (Fig. A.1(a)), the cell height is allowed to increase until

$$h > (1 + \alpha_s)h_{ideal} \quad (\text{A.5})$$

where h_{ideal} is the ideal cell height, and α_s is the layer split factor. When the condition in Eq (A.5) is met, the cells are split based on the ratio (α_s) of the new cell heights to old cell heights.

If the cells in layer j are being compressed (Fig. A.1(b)), they can be compressed until

$$h < \alpha_c h_{ideal} \quad (\text{A.6})$$

where α_c is the layer collapse factor. When this condition is met, the compressed layer of cells (layer j in Fig. A.1) is merged into the layer of cells above the compressed layer (layer i in Fig. A.1).

A.4 Overset mesh method

In ANSYS FLUENT, the connectivity is established by three main steps: hole cutting, overlapping minimization, and donor search. Hole cutting is the process by which cells lying outside of the flow region (i.e., inside bodies and outside of the computational domain) are marked as dead cells. This is achieved by marking all the cells that are cut by physical boundary zones (e.g., wall, inlet, outlet, symmetry), and marking seed cells determined to lie outside the flow region for subsequent flood filling of dead cells.

Overlap minimization is used to minimize mesh overlap among different component and background meshes by converting additional solve cells into receptor cells and turning unnecessary receptors into dead cells. During this process, a solve cell is turned into a receptor cell if the cell can find a suitable donor cell with higher donor priority. By default, smaller cells have a higher donor priority. Thus, in mesh overlap areas, without additional user input the solver attempts to obtain the solution on the finest local mesh.

The donor search is the final step in establishing the domain connectivity. FLUENT searches other meshes for valid solve cells for each receptor. The solve cell containing the cell centroid of the receptor cell, along with its connected solve cells, are used as donor candidates for a given receptor. Each receptor must have at least one valid donor cell. There must be four or more cells in the overlap of both meshes to ensure a successful donor search. The receptor cells, which form the fringe layer of a mesh zone, must overlap sufficiently with the opposite mesh, such that they find valid solve cells as donors.

References

- [1] J.M. Villafruela, J.F. San José, F. Castro, A. Zarzuelo, Airflow patterns through a sliding door during opening and foot traffic in operating rooms, *Build. Environ.* 109 (2016) 190–198, <https://doi.org/10.1016/j.buildenv.2016.09.025>.
- [2] H. Matsumoto, Y. Ohba, The influence of a moving object on air distribution in displacement ventilated rooms, *J. Asian Architect. Build Eng.* 3 (2004) 71–75, <https://doi.org/10.3130/jaabe.3.71>.
- [3] W. Wu, Z. Lin, Experimental study of the influence of a moving manikin on temperature profile and carbon dioxide distribution under three air distribution methods, *Build. Environ.* 87 (2015) 142–153, <https://doi.org/10.1016/j.buildenv.2015.01.027>.
- [4] M. Liu, D. Chang, J. Liu, S. Ji, C.H. Lin, D. Wei, Z. Long, T. Zhang, X. Shen, Q. Cao, X. Li, X. Zeng, H. Li, Experimental investigation of air distribution in an airliner cabin mockup with displacement ventilation, *Build. Environ.* 191 (2021) 107577, <https://doi.org/10.1016/j.buildenv.2020.107577>.
- [5] H. Brohus, K.D. Balling, D. Jeppesen, Influence of movements on contaminant transport in an operating room, *Indoor Air* 16 (2006) 356–372, <https://doi.org/10.1111/j.1600-0668.2006.00454.x>.
- [6] J. Eslami, A. Abbassi, M.H. Saidi, Numerical simulation of the effect of visitor's movement on bacteria-carrying particles distribution in hospital isolation room, *Sci. Iran.* 24 (2017) 1160–1170, <https://doi.org/10.24200/sci.2017.4097>.
- [7] W. Zhao, S. Lestinen, S. Kilpeläinen, R. Kosonen, An experimental study of the influence of a moving person on airflow characteristics and thermal conditions with diffuse ceiling ventilation, *Indoor Built Environ.* 29 (2020) 860–880, <https://doi.org/10.1177/1420326X20917202>.
- [8] L. Lv, L. Zeng, Y. Wu, J. Gao, W. Xie, C. Cao, Y. Chen, J. Zhang, Effects of human walking on the capture efficiency of range hood in residential kitchen, *Build. Environ.* 196 (2021) 107821, <https://doi.org/10.1016/j.buildenv.2021.107821>.
- [9] S.-J. Cao, D. Cen, W. Zhang, Z. Feng, Study on the impacts of human walking on indoor particles dispersion using momentum theory method, *Build. Environ.* 126 (2017) 195–206, <https://doi.org/10.1016/j.buildenv.2017.10.001>.
- [10] Z. Han, W. Weng, Q. Huang, Numerical and experimental investigation on the dynamic airflow of human movement in a full-scale cabin, *HVAC R Res.* 20 (2014) 444–457, <https://doi.org/10.1080/10789669.2014.882677>.
- [11] S. Mazumdar, S.B. Poussou, C.-H. Lin, S.S. Isukapalli, M.W. Plesniak, Q. Chen, Impact of scaling and body movement on contaminant transport in airliner cabins, *Atmos. Environ.* 45 (2011) 6019–6028, <https://doi.org/10.1016/j.atmosenv.2011.07.049>.
- [12] J. Wang, T.-T. Chow, Numerical investigation of influence of human walking on dispersion and deposition of exhalation droplets in airborne infection isolation room, *Build. Environ.* 46 (2011) 1993–2002, <https://doi.org/10.1016/j.buildenv.2011.04.008>.
- [13] I. Goldasteh, Y. Tian, G. Ahmadi, A.R. Ferro, Human induced flow field and resultant particle resuspension and transport during gait cycle, *Build. Environ.* 77 (2014) 101–109, <https://doi.org/10.1016/j.buildenv.2014.03.016>.
- [14] S. Mazumdar, Y. Yin, A. Guity, P. Marmion, B. Gulick, Q. Chen, Impact of moving objects on contaminant concentration distributions in an inpatient ward with displacement ventilation, *HVAC R Res.* 16 (2010) 545–563, <https://doi.org/10.1080/10789669.2010.10390921>.

- [15] Y.-C. Shih, C.-C. Chiu, O. Wang, Dynamic airflow simulation within an isolation room, *Build. Environ.* 42 (2007) 3194–3209, <https://doi.org/10.1016/j.buildenv.2006.08.008>.
- [16] J.-I. Choi, J.R. Edwards, Large-eddy simulation of human-induced contaminant transport in room compartments, *Indoor Air* 22 (2012) 77–87, <https://doi.org/10.1111/j.1600-0668.2011.00741.x>.
- [17] J. Hang, Y. Li, W.H. Ching, J. Wei, R. Jin, L. Liu, X. Xie, Potential airborne transmission between two isolation cubicles through a shared anteroom, *Build. Environ.* 89 (2015) 264–278, <https://doi.org/10.1016/j.buildenv.2015.03.004>.
- [18] Y. Yin, W. Xu, J. Gupta, A. Guity, P. Marmion, A. Manning, B. Gulick, X. Zhang, Q. Chen, Experimental study on displacement and mixing ventilation systems for a patient ward, *HVAC R Res.* 15 (2009) 1175–1191, <https://doi.org/10.1080/10789669.2009.10390885>.
- [19] W. Lu, A.T. Howarth, N. Adam, S.B. Riffati, Modelling and measurement of airflow and aerosol particle distribution in a ventilated two-zone chamber, *Build. Environ.* 31 (1996) 417–423, [https://doi.org/10.1016/0360-1323\(96\)00019-4](https://doi.org/10.1016/0360-1323(96)00019-4).
- [20] N. Rostamy, D. Sumner, D.J. Bergstrom, J.D. Bugg, Local flow field of a surface-mounted finite circular cylinder, *J. Fluid Struct.* 34 (2012) 105–122, <https://doi.org/10.1016/j.jfluidstructs.2012.04.014>.
- [21] N. Luo, W. Weng, X. Xu, M. Fu, Human-walking-induced wake flow – PIV experiments and CFD simulations, *Indoor Built Environ.* 27 (2018) 1069–1084, <https://doi.org/10.1177/1420326X17701279>.
- [22] S.B. Poussou, S. Mazumdar, M.W. Plesniak, P.E. Sojka, Q. Chen, Flow and contaminant transport in an airliner cabin induced by a moving body: model experiments and CFD predictions, *Atmos. Environ.* 44 (2010) 2830–2839, <https://doi.org/10.1016/j.atmosenv.2010.04.053>.
- [23] J. Wang, T.T. Chow, Influence of human movement on the transport of airborne infectious particles in hospital, *J. Build. Perform. Simul.* 8 (2015) 205–215, <https://doi.org/10.1080/19401493.2014.905636>.
- [24] Y. Wu, N. Gao, The dynamics of the body motion induced wake flow and its effects on the contaminant dispersion, *Build. Environ.* 82 (2014) 63–74, <https://doi.org/10.1016/j.buildenv.2014.08.003>.
- [25] T.-T. Chow, J. Wang, Dynamic simulation on impact of surgeon bending movement on bacteria-carrying particles distribution in operating theatre, *Build. Environ.* 57 (2012) 68–80, <https://doi.org/10.1016/j.buildenv.2012.04.010>.
- [26] Y. Tao, K. Inthavong, J.Y. Tu, Dynamic meshing modelling for particle resuspension caused by swinging manikin motion, *Build. Environ.* 123 (2017) 529–542, <https://doi.org/10.1016/j.buildenv.2017.07.026>.
- [27] Y. Tao, K. Inthavong, J. Tu, A numerical investigation of wind environment around a walking human body, *J. Wind Eng. Ind. Aerod.* 168 (2017) 9–19, <https://doi.org/10.1016/j.jweia.2017.05.003>.
- [28] S.B. Poussou, *Experimental Investigation of Airborne Contaminant Transport by a Human Wake Moving in a Ventilated Aircraft Cabin*, Purdue University, 2008.
- [29] Fluent, ANSYS Fluent User's Guide v19.3, ANSYS, 2019.
- [30] T.E. Tezduyar, Finite element methods for fluid dynamics with moving boundaries and interfaces, in: *Encyclopedia of Computational Mechanics*, American Cancer Society, 2004, <https://doi.org/10.1002/0470091355.ecm069>.
- [31] Y. Wang, K. Zhong, N. Zhang, Y. Kang, Numerical analysis of solar radiation effects on flow patterns in street canyons, *Eng. Appl. Comput. Fluid Mech.* 8 (2014) 252–262, <https://doi.org/10.1080/19942060.2014.11015511>.
- [32] G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967, <https://doi.org/10.1017/CBO9780511800955>.
- [33] Y. Tao, K. Inthavong, J. Tu, A numerical investigation of wind environment around a walking human body, *J. Wind Eng. Ind. Aerod.* 168 (2017) 9–19, <https://doi.org/10.1016/j.jweia.2017.05.003>.
- [34] S. Mazumdar, Y. Yin, A. Guity, P. Marmion, B. Gulick, Q. Chen, Impact of moving objects on contaminant concentration distributions in an inpatient ward with displacement ventilation, *HVAC R Res.* 16 (2010) 545–563, <https://doi.org/10.1080/10789669.2010.10390921>.
- [35] Z. Zhang, W. Zhang, Z.J. Zhai, Q.Y. Chen, Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: Part 2—comparison with experimental data from literature, *HVAC R Res.* 13 (2007) 871–886, <https://doi.org/10.1080/10789669.2007.10391460>.
- [36] A. Orszag, S. Yakhot, S. Flannery, F.E. Boysan, D. Choudhury, J. Maruzewski, B. Patel, *Renormalization group modeling and turbulence simulations*, 1993.
- [37] R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzengleichungen der mathematischen Physik, *Math. Ann.* 100 (1928) 32–74, <https://doi.org/10.1007/BF01448839>.
- [38] S. Mazumdar, *Transmission of Airborne Contaminants in Airliner Cabins*, Purdue University, 2009.

Nomenclature

- CFD:** Computational fluid dynamics
DMU method: Dynamic mesh update method
DL method: Dynamic layering method
SS&R method: Spring-based smoothing & remeshing method
OM method: Overset mesh method
Co: Courant number
Re: Reynolds number, which is 9500
L: Length (m)
t_r: Time that an application ran for (hour)
p_c: Core(s) that an application ran on (core)
t_{ct}: Computational time (core-hour)
h_{ideal}: Ideal cell height in dynamic layering method (m)
k_{fac}: Spring constant factor in spring-based smoothing & remeshing method
U_{body}: Body velocity, which is 17.5 cm/s
U_{n-1}: Velocity in the formal grid number (m/s)
U_n: Velocity in the current grid number (m/s)
N: Number of the examined sample points
ū: Flow velocity
ū_g: Mesh velocity of the moving mesh
S_{d_j}: Source term of ϕ
n_f: Number of faces on the control volume
A_j: Face j area vector
Ā_i: Normalized dot product of the area vector of a face
f̄_i: A vector from the centroid of the cell to the centroid of that face
c̄_i: A vector from the centroid of the cell to the centroid of the adjacent cell that shares that face
t: Time (s)
Δt: Time step, which is 0.001 s
Δs: Characteristic size of the cells (m)
u_x: Velocity component in the x direction (m/s)
u_y: Velocity component in the y direction (m/s)
u_z: Velocity component in the z direction (m/s)
f: Absolute error frequency
t_c: Computational time for each ten time-steps (s)
t_{ca}: Accumulative computational time (s)
V_s: Velocity obtained from numerical simulation (m/s)
V_f: Velocity from the fitting curve (m/s)
A_{i,j}: Performance index of Case i in aspect j of numerical performance
a_{i,j}: Performance value of Case i in aspect j of numerical performance
a_{max,j}: Maximum of $a_{i,j}$ in the aspect j of numerical performance
- Greek symbols**
 ρ : Fluid density
 μ : Molecular viscosity
 I : Diffusion coefficient
 σ : Absolute error
 η_{OQ} : Value of orthogonal quality
 η_{sq} : Value of skewness quality
 η_{cs} : Value of cell skewness
 φ : Completion rate
 δ : Degree of discontinuity