COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **4**, No. 2, pp. 350-377

Commun. Comput. Phys.
August 2008

An Efficient Parallel/Unstructured-Multigrid Implicit Method for Simulating 3D Fluid-Structure Interaction

X. Lv, Y. Zhao*, X. Y. Huang, G. H. Xia and X. H. Su

School of Mechanical and Aerospace Engineering, College of Engineering, Nanyang Technological University, Singapore 639798.

Received 16 April 2007; Accepted (in revised version) 16 January 2008

Communicated by Boo-Cheong Khoo

Available online 18 March 2008

Abstract. A finite volume (FV) method for simulating 3D Fluid-Structure Interaction (FSI) is presented in this paper. The fluid flow is simulated using a parallel unstructured multigrid preconditioned implicit compressible solver, whist a 3D matrix-free implicit unstructured multigrid finite volume solver is employed for the structural dynamics. The two modules are then coupled using a so-called immersed membrane method (IMM). Large-Eddy Simulation (LES) is employed to predict turbulence. Results from several moving boundary and FSI problems are presented to validate proposed methods and demonstrate their efficiency.

PACS: 47.11-j

Key words: Finite volume method, moving boundary, fluid-structure interaction, unstructured multigrid, computational structural mechanics, immersed membrane method, large-Eddy simulation.

1 Introduction

The simulation of fluid flows with arbitrarily moving solid/elastic bodies is one of the challenges in computational fluid dynamics (CFD). The development of accurate, robust and efficient methods that can tackle this problem would provide a powerful tool to solve many practical engineering problems. In recent years significant research efforts have been devoted to the development of numerical models for studying moving boundary problems based on the finite volume and finite element methods. Luo and Pedley [1–3] performed a time-dependent simulation of a coupled flow-membrane problem, using

*Corresponding author. *Email addresses:* xin.lv@plymouth.ac.uk (X. Lv), myzhao@ntu.edu.sg (Y. Zhao)

the Arbitrary Lagrangian Eulerian (ALE) method together with a spine scheme to treat a compliant wall moving in its wall normal direction in a channel. Zhao *et al.* [4,5] have also proposed a new dynamic mesh scheme to simulate an arbitrarily moving elastic wall in a similar channel based on the ALE. Gaitonde [6,7] developed a moving mesh method for the computation of compressible viscous flow past moving aerofoils. A sequence of body conforming grids and the corresponding grid speeds were required, where the inner and outer boundaries of the grids moved independently. The required grids and their velocities were found by using a transfinite interpolation technique. Heil [8] studied a three-dimensional steady Stokes flow in an elastic tube using the ALE, which was described by non-linear shell equations. Only the final equilibrium state was presented because the flow was steady. Lefrancois *et al.* [9] developed a finite element model for studying fluid-structure interaction and an ALE formulation was used to model the compressible inviscid flow with moving boundaries with large deformation. It is observed that all the works reported have mostly relied on the costly grid regeneration method to capture large movements of the boundary in the flow field, whist the less costly dynamic grid method is believed to be unable to handle large boundary and mesh deformations.

An alternative to the ALE is the Eulerian method, where the computational mesh is fixed without deformation or movement. The group of Eulerian methods includes Immersed Boundary (IB) method and Fictitious Domain (FD) method *etc*. Peskin *et al.* [10] proposed the IB method to simulate the motion of human hearts and heart valves. At its early stage, the IB method could not consider the inertial effect of the structure because the dynamic equation of the structure was not used to calculate its movement. In their recent work [11], Zhu and Peskin did consider the inertial effect by taking the structure's (a soap film) mass into account. The IB method has been applied to a wide range of problems, mostly in bio-fluid dynamics, including blood flow in the human heart [12], platelet aggregation during blood clotting [13, 14] and the motion of flexible pulp fibers [15]. Generally speaking, Eulerian methods are relatively less-complicated techniques by using fixed fluid meshes, which reduce the computational costs for mesh treatment. However, the above Eulerian methods do not allow for 'jump' conditions between two sides of immersed thin structures, because the flow conditions on structural boundaries are smoothed over several mesh cells across or near the immersed structures due to the fact that the structures are considered as internal conditions in the flow field and source terms are distributed to nearby fluid nodes for constraining the flow field. Recently Zhao *et al.* [16] has also developed a so-called Immersed Object Method (IOM) with overlapping unstructured grids for general Fluid-Structure-Interaction (FSI) simulation. The main idea of the method is that the fluid covered by immersed objects is assumed to be frozen and moves like a solid, whose kinematics is enforced by adding source terms to the momentum equations. Overlapping grids are wrapped around the objects and the boundary conditions for the overlapping grids are transferred from the Eulerian grid to the overlapping grids for further computation on the overlapping grids, in order to capture the fine details of boundary lays over the moving objects.

Tai and Zhao [17] parallelized an incompressible Navier-Stokes solver based on the

artificial compressibility approach and higher-order characteristics-based finite volume scheme for unstructured non-moving grids using a multigrid domain decomposition approach (MG-DD) and the single program multiple data (SPMD) parallel strategy with message passing interface (MPI). They developed a communication scheme for an overlapping partition structure in order to obtain continuity of results in the whole domain. Singh and Zhao [18] developed a parallel dynamic unstructured moving grid Direct Monte Carlo Simulation (DMCS) of molecular gas dynamics and the associated thin film deposition. It is envisaged that the combination of the ALE approach and the parallel unstructured MG, as well as the preconditioning, may not be efficient enough and easy to be implemented into computer codes, compared with the combination with an Eulerian method for handling moving objects and the resulting unsteady 3D flows, not to mention the difficulty for the ALE to simulate large mesh deformations.

In this study, we aim to develop a parallel unstructured multi-grid preconditioned compressible Navier-Stokes solver implementing the IMM [19, 20] to calculate 3D unsteady low-Mach-number flows with rigid/elastic moving objects. The developed solver makes it possible to include moving objects in the flow fields with complexities that existing methods can not easily handle because it does not requires complicated interpolation of boundary conditions along surface normal direction. Unlike the aforementioned Eulerian methods, the new method does not smooth fluid forces across the immersed structure and discontinuities in pressure and gradients of flow properties can be accurately accounted for as a result. To further illustrate the capability of proposed method to handle the FSI situations, we also coupled the current flow solver to a newly developed finite volume based computational structural mechanics solver [21]. The Large-Eddy simulation (LES) has been chosen to calculate the possible arising turbulent phenomena. In this work, a compressible dynamic form of Smagorinsky SGS model has been implemented to calculate the SGS stress tensor. This model relies upon the Germano identity, which has been generalized in order to be applied to other subgrid-terms arising in the filtered energy equation. Furthermore, an improved formulation of the dynamic mixed model has been proposed for better representing the backscatter of turbulence energy which has been proven to be important for compressible flows. The unstructured grid filtering was adapted from a new filtering approach based on the least-squares technique. This approach can filter a function to any given level of commutation error on unstructured grids. When used together with IMM, to avoid complex modifications of the filtering operator at the vicinity of the fluid-structure interface, a linear reconstruction procedure similar to that used for the velocity field is also applied for turbulent viscosity μ_t and Prandtl number Pr_t.

The paper is structured as follows. In Section 2 which is the next section, the finite-volume preconditioned compressible flow solver on tetrahedral grids is described. In Section 3, a brief description of our parallel and multigrid implementation is presented. This is followed by a description of the IMM algorithm in Section 4. Some information about the coupling between the flow and structure solvers can be found in Section 5. Brief description of turbulent modeling method employed in current work can be found

in Section 6. In Section 7, a grid convergence study is carried out for flows induced by a sphere oscillating in a box to establish the order of accuracy of the method and to demonstrate the capability of the method for handling moving boundary problems. Next, the result for a classical benchmark problem involving the turbulent flow over a sphere is presented to validate the proposed LES model. The accuracy of the method is then further validated by applying it to predict the aeroelastic flutter of an ONERA M6 wing in the transonic flow.

2 Governing equations and numerical methods

2.1 Governing equations

The Navier-Stokes equations for three-dimensional compressible unsteady flows can be given in vector form explicitly expressing the conservation laws of mass, momentum and energy. We also introduce, in the equations, pseudo-time terms to provide pseudo-time marching for their numerical solutions:

$$\Gamma_1 \frac{\partial W_p}{\partial \tau} + \frac{\partial W_c}{\partial t} + \nabla \cdot \vec{F}_i = \nabla \cdot \vec{F}_v \tag{2.1}$$

where

$$W_c = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_t \end{bmatrix}, \quad W_p = \begin{bmatrix} p \\ u \\ v \\ w \\ T' \end{bmatrix}, \quad \vec{F}_i = \begin{bmatrix} \rho\vec{U} \\ \rho u\vec{U} + p\vec{i} \\ \rho v\vec{U} + p\vec{j} \\ \rho w\vec{U} + p\vec{k} \\ \rho H\vec{U} \end{bmatrix}, \quad \vec{F}_v = \begin{bmatrix} 0 \\ \vec{\tau}_x \\ \vec{\tau}_y \\ \vec{\tau}_z \\ (\bar{\bar{\tau}} \cdot \vec{U} - \vec{q}) \end{bmatrix}. \tag{2.2}$$

τ is the pseudo time and Γ_1 is the preconditioning matrix in the pseudo-time terms for low-Mach-number flows which is defined in the appendix. W_c and W_p are the vectors of conservative and primitive dependent variables respectively; $\vec{F}_i$ and $\vec{F}_v$ are the inviscid convective flux and viscous flux vectors. Furthermore we have the following formulas:

$$\begin{aligned}
&\vec{U} = u\vec{i} + v\vec{j} + w\vec{k}, \qquad \bar{\bar{\tau}} = \vec{\tau}_x\vec{i} + \vec{\tau}_y\vec{j} + \vec{\tau}_z\vec{k}, \\
&\vec{\tau}_i = \tau_{ix}\vec{i} + \tau_{iy}\vec{j} + \tau_{iz}\vec{k}, \quad \tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\nabla \cdot \vec{U}\right), \\
&\vec{q} = q_x\vec{i} + q_y\vec{j} + q_z\vec{k}, \qquad T' = \frac{p}{\rho} = \frac{c^2}{\gamma}, \quad \gamma = \frac{C_p}{C_v}.
\end{aligned}$$

$\vec{i}$, $\vec{j}$ and $\vec{k}$ are the three unit vectors in three Cartesian directions τ_{ix}, τ_{iy} and τ_{iz} the viscous stresses and q the heat transfer flux vector, defined by

$$\vec{q} = -\kappa\nabla T = -\frac{\mu C_p}{Pr}\nabla T,$$

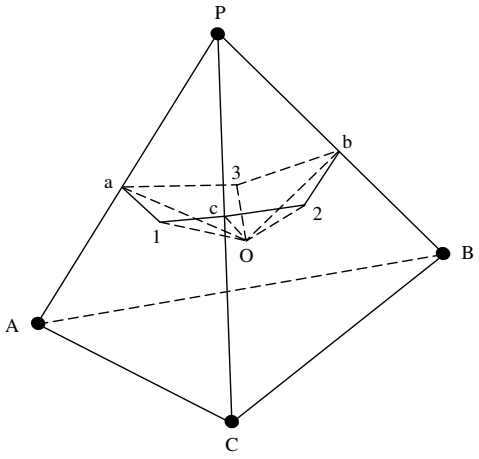

Figure 1: Construction of control volume within a tetrahedron for a node P.

where T the temperature and Pr the Prandtl number:

$$Pr=\frac{\mu C_p}{\kappa}.$$

The above equations are non-dimensionalized and the non-dimensional variables used are defined in [19].

2.2 Numerical methods

The 3D equations (2.1) are transformed into an integral form and discretized on an unstructured grid. A cell-vertex finite volume scheme is adopted here. For every vertex, as shown in Fig. 1, a control volume is constructed using the median duals of the tetrahedral cells. Spatial discretization is performed by using the integral form of the conservation equations over the control volume surrounding node P:

$$\iiint_{cv}\frac{\partial Q_1'}{\partial\tau}dV+\iiint_{cv}\frac{\partial W_c}{\partial t}dV+\iiint_{cv}\nabla\cdot\vec{F}_i dV-\iiint_{cv}\nabla\cdot\vec{F}_v dV=0. \tag{2.3}$$

Noted that a new variable Q_1' has arisen as $\frac{\partial Q_1'}{\partial\tau}=\Gamma_1\frac{\partial W_p}{\partial\tau}$, and the Jacobian $\Gamma_1=\frac{\partial Q_1'}{\partial W_p}$, so that,

$$\frac{\partial Q_1'}{\partial\tau}=\frac{\partial Q_1'}{\partial W_p}\frac{\partial W_p}{\partial\tau}=\Gamma_1\frac{\partial W_p}{\partial\tau}.$$

The convective term is transformed into a summation:

$$\iiint_{cv}\nabla\cdot\vec{F}_i dV=\int_{S_{cv}}\vec{F}_i\cdot\vec{n}dS=\sum_{k=1}^{nbseg}\left[\left(\vec{F}_i\right)_{ij}\cdot\vec{n}\Delta S\right]_k, \tag{2.4}$$

where *nbseg* is the number of the edges associated with node P, $(\vec{F}_i)_{ij}^k$ is the inviscid flux through the part of control volume surface associated with edge k, and $\vec{n}$ is the unit normal vector of the control volume surface. Finally, ΔS_k is a part of the control volume surface associated with edge k. Therefore, all the fluxes are calculated for the edges and then collected at the two end of each edge for updating of flow variables in time marching. The viscous term is calculated using a cell-based method:

$$\iiint_{cv} \nabla\cdot\vec{F}_v dV = \int_{S_{cv}} \vec{F}_v\cdot\vec{n}dS = \sum_{i=1}^{ncell}\left[\vec{F}_v\cdot\vec{n}\Delta S_c\right]_i, \tag{2.5}$$

where *ncell* is the number of elements associated with node P and ΔS_{ci} is the part of control volume surface in cell i. By using the following relation:

$$\int_{S_{cv}} \overrightarrow{dS} = 0,$$

the total vector surface of the control volume in a cell i becomes:

$$\vec{n}\Delta S_{ci} = \frac{1}{3}\left(\vec{n}\Delta S_{pi}\right).$$

Thus, the calculation of viscous terms can be simplified as

$$\sum_{i=1}^{ncell}\left[\vec{F}_v\cdot\vec{n}\Delta S_c\right]_i = \frac{1}{3}\sum_{i=1}^{ncell}\left[\vec{F}_v\cdot\vec{n}\Delta S_p\right]_i, \tag{2.6}$$

where $\vec{n}\Delta S_{pi}$ is the surface vector of the face opposite node P of the tetrahedron under consideration. Here the $(\vec{F}_v)_i$ is calculated at the center of the tetrahedron with a node P, and can be obtained by using the Green's Theorem based on the variables at the four vertices of the tetrahedron. Similar to the Galerkin type of formulation, the gradient of a flow variable ϕ at the center of a tetrahedron is evaluated as follows:

$$grad\phi_c = -\frac{\sum_{i=1}^4 \phi_i 9S_i}{27V} = -\frac{1}{3}\frac{\sum_{i=1}^4 \phi_i S_i}{V}, \tag{2.7}$$

where ϕ_i is the flow variable at a vertex i of the tetrahedron and S_i is the surface area that is opposite to node i, V is the volume of the tetrahedron. Gradients at the vertices are obtained by a volume averaging of the gradients at the center of cells associated with the vertex under consideration.

In this work, a high-order Roe's TVD scheme for compressible flow for arbitrary unstructured 3D grids has been adopted. Because of the preconditioning matrix Γ_1, the inviscid fluxes, $(\vec{F}_i)_{ij}^k$, through the face k is now reformulated as

$$\begin{aligned}(\vec{F}_i)_{ij}^k &\equiv \frac{1}{2}\left((\vec{F}_i)_i + (\vec{F}_i)_j\right)_k - \frac{1}{2}\left|\frac{\partial\vec{F}_i}{\partial Q_1'}\right|_k (\delta Q_1')_k \\ &= \frac{1}{2}\left((\vec{F}_i)_i + (\vec{F}_i)_j\right)_k - \frac{1}{2}\left|\frac{\partial\vec{F}_i}{\partial W_p}\frac{\partial W_p}{\partial Q_1'}\right|_k \left(\frac{\partial Q_1'}{\partial W_p}\right)_k \left((W_p)_j - (W_p)_i\right)_k.\end{aligned} \tag{2.8}$$

Note that we have retained the variable, Q_1', in computing this flux. Define the Jacobian in the normal direction as

$$\left(H_p\right)_k=\left(\frac{\partial\vec{F}_i}{\partial W_p}\right)_k.$$

By using the previously defined Jacobian $\Gamma_1=\frac{\partial Q_1'}{\partial W_p}$, the above expression becomes

$$\left(\vec{F}_i\right)_{ij}^k\equiv\frac{1}{2}\left((\vec{F}_i)_i+(\vec{F}_i)_j\right)_k-\frac{1}{2}\left|H_p\Gamma_1^{-1}\right|_k\Gamma_{1k}\left((W_p)_j-(W_p)_i\right)_k.$$

Drop the subscript k on the flux vector and the Jacobian with the assumption that the fluxes and Jacobians all correspond to conditions in the normal direction on the given control volume surface. After some simple algebraic derivations we have

$$\left(\vec{F}_i\right)_{ij}\equiv\frac{1}{2}\left((\vec{F}_i)_i+(\vec{F}_i)_j\right)-\frac{1}{2}\Gamma_1\left|\Gamma_1^{-1}H_p\right|\left((W_p)_j-(W_p)_i\right). \tag{2.9}$$

Combined with the third-order MUSCL interpolation, it can produce accurate and stable solution on unstructured grids. The left and right state vectors W_L and W_R at a control volume surface are evaluated using a nominally third-order upwind-biased interpolation scheme. If the left and right state vectors are set to W_i and W_j (i and j corresponding to the two end nodes of an edge), it is a first-order upwind scheme, which are shown as follows:

$$W_L=W_i+\frac{1}{4}\left[(1-\kappa)\Delta_i^-+(1+\kappa)\Delta_i^+\right], \tag{2.10a}$$

$$W_R=W_j-\frac{1}{4}\left[(1-\kappa)\Delta_j^++(1+\kappa)\Delta_j^-\right], \tag{2.10b}$$

where

$$\begin{aligned}
&\Delta_i^+=\Delta_j^-=W_j-W_i,\\
&\Delta_i^-=W_i-W_{i-1}=2\overrightarrow{ij}\cdot\nabla W_i-(W_j-W_i)=2\overrightarrow{ij}\cdot\nabla W_i-\Delta_i^+,\\
&\Delta_j^+=W_{j+1}-W_j=2\overrightarrow{ij}\cdot\nabla W_j-(W_j-W_i)=2\overrightarrow{ij}\cdot\nabla W_j-\Delta_j^-.
\end{aligned}$$

Therefore, substituting the above equations into Eqs. (2.10a) and (2.10b), the final equations based on upwind-biased interpolation scheme is shown as follows:

$$W_L=W_i+\frac{1}{2}\left[(1-\kappa)\overrightarrow{ij}\cdot\nabla W_i+\kappa\Delta_i^+\right], \tag{2.11a}$$

$$W_R=W_j-\frac{1}{2}\left[(1-\kappa)\overrightarrow{ij}\cdot\nabla W_j+\kappa\Delta_j^-\right], \tag{2.11b}$$

where κ is set to $1/3$, which corresponds to a nominally third-order accuracy. $\overrightarrow{ij}$ is the vector representing the edge, which points from node P to its neighbouring node under

consideration. The gradients of W at i and j are calculated by volume-averaging the gradients of the cells that surround i and j.

Finally, for a given node P, the spatially discretized equations (2.3) form a system of coupled ordinary differential equations, which can be reformulated as

$$\begin{aligned}&\frac{\partial\overline{Q}_1'}{\partial\tau}\Delta V_{cv}+\frac{\partial\overline{W}_c}{\partial t}\Delta V_{cv}\\&=-\left\{\frac{1}{2}\sum_{k=1}^{nbseg}\left[\left((\vec{F}_i)_i+(\vec{F}_i)_j\right)-\Gamma_1\left|\Gamma_1^{-1}H_p\right|\left((W_p)_j-(W_p)_i\right)\right]_k\cdot\vec{n}\Delta S-\frac{1}{3}\sum_{i=1}^{ncell}\left[\vec{F}_v\cdot\vec{n}\Delta S_P\right]_i\right\}\\&=-R,\end{aligned}\tag{2.12}$$

where R represents the residual which includes the convective and diffusive fluxes and ΔV_{cv} is the control volume of node P. The over-bar in Eq. (2.12) denotes the cell-averaging value. An implicit scheme is adopted for Eq. (2.12) and the time dependent term is discretized using a second-order-accurate backward differencing scheme,

$$\begin{aligned}\frac{\partial\overline{Q}_1'}{\partial\tau}\Delta V_{cv}&=-R^{n+1}-\left(\frac{1.5\Delta V_{cv}^{n+1}W_c^{n+1}-2.0\Delta V_{cv}^{n}W_c^{n}+0.5\Delta V_{cv}^{n-1}W_c^{n-1}}{\Delta t}\right)\\&=\tilde{R}^{n+1},\end{aligned}\tag{2.13}$$

where the superscript $(n+1)$ denotes the physical time level $(n+1)\Delta t$ and all the variables are evaluated at this time level, $\tilde{R}(W_p^{n+1})$ is the new modified residual which contains both the time derivative and flux vectors. The derivative with respect to a pseudo time τ is discretized as

$$\Delta V_{cv}^{n+1}\Gamma_1\frac{W_p^{n+1,m+1}-W_p^{n+1,m}}{d\tau}=\tilde{R}^{n+1},\tag{2.14}$$

whose solution is sought by marching to a pseudo steady state in τ. Here m and $(m+1)$ denote the initial and final pseudo time levels. Once the artificial steady state is reached, the derivative of W_p with respect to τ becomes zero, and the solution satisfies $\tilde{R}^{n+1}=0$. Hence, the original unsteady Navier-Stokes equations are fully recovered. Therefore, instead of solving the equations in each time step in the physical time domain (t), the problem is transformed into a sequence of steady-state computations in the artificial time domain (τ). Eq. (2.14) can be integrated in pseudo time by an explicit five-stage Runge-Kutta scheme. However, the pseudo time step size may be severely restricted if the physical time step size is very small. Hence, a fully implicit dual time stepping is adopted here. A Taylor series expansion is performed for the residual in Eq. (2.14) with respect to the pseudo time for node i,

$$\tilde{R}_i^{m+1}=\tilde{R}_i^{m}+\frac{\partial\tilde{R}_i}{\partial(W_p)_i}\Delta(W_p)_i+\sum_{j=1}^{nbseg}\frac{\partial\tilde{R}_i}{\partial(W_p)_j}\Delta(W_p)_j,\tag{2.15}$$

where *nbseg* is the number of edges connected to i, which is also equal to the number of neighboring points connected to point i through the edges. An approximate flux function is introduced here to simplify the implicit time stepping calculation. The total flux (including both convective and viscous fluxes) across a control volume surface associated with a certain edge ij can be approximated as

$$F_{ij}\approx\frac{1}{2}\left[(\vec{F}_i)_i\cdot\vec{n}+(\vec{F}_i)_j\cdot\vec{n}-|\lambda_{ij}|\left((W_p)_j-(W_p)_i\right)\right],$$

where λ_{ij} is the spectral radius based on the preconditioned system which is associated with edge ij. After combining all the residuals terms at every node in the flow field into a vector and dropping the third term of the right-hand side of Eq. (2.15), we have

$$\begin{aligned}\tilde{R}_i^{n+1,m+1}&=\tilde{R}_i^{n+1,m}-\frac{\partial R_i}{\partial(W_p)_i}\Delta(W_p)_i-1.5\frac{\Delta V_{cv}^{n+1}}{\Delta t}\frac{\partial W_c}{\partial W_p}\Delta(W_p)_i\\&=\tilde{R}_i^{n+1,m}-\sum_{j=1}^{nbseg}(H_{p,j})\Delta(W_p)_i-1.5\frac{\Delta V_{cv}^{n+1}}{\Delta t}M\Delta(W_p)_i,\end{aligned}$$

where

$$H_{p,j}=\frac{1}{2}[\frac{\partial(\vec{F}_i)_{ij}}{\partial(W_p)_i}+|\lambda_{ij}|],\quad M=\frac{\partial W_c}{\partial W_p}.$$

The whole-field equivalent of Eq. (2.13) can then be re-written as

$$\begin{aligned}&\left(I+\frac{1.5\Delta\tau}{\Delta t}\Gamma_1^{-1}M+\frac{\Delta\tau}{\Delta V_{cv}^{n+1}}\sum_{j=1}^{nbseg}\left(\Gamma_1^{-1}H_{p,j}\right)\right)\frac{W_p^{n+1,m+1}-W_p^{n+1,m}}{\Delta\tau}\Delta V_{cv}^{n+1}\\&=\Gamma_1^{-1}\left(-\frac{1.5W_c^{n+1,m}\Delta V_{cv}^{n+1}-2.0W_c^n\Delta V_{cv}^n+0.5W_c^{n-1}\Delta V_{cv}^{n-1}}{\Delta t}-R^{n+1,m}\right).\end{aligned}\tag{2.16}$$

Further approximation can be introduced in order to achieve matrix-free computation. Please check reference [19] for more related details.

2.3 Boundary treatment

The far field calculations are based on characteristic variables (Reimann invariants). Thus at inflow the incoming variables corresponding to positive eigenvalues are specified while the outgoing variables corresponding to negative eigenvalues are extrapolated. Once we change the time dependent equations we also change the characteristics of the system (though not the signs of the eigenvalues). Hence, it is also necessary to modify the boundary conditions for the preconditioned system. Here the flux at a boundary is defined as

$$\vec{F}_i\cdot\vec{n}S_k=(\vec{F}_i^+ +\vec{F}_i^-)\cdot\Delta S_k.$$

Here, $\vec{F}_i^{\pm}$ has been redefined as $\vec{F}_i^{\pm} = X_{H_p,R}\Lambda^{\pm}X_{H_p,L}W_p$, where $\Lambda^{\pm} = (\lambda_i \pm |\lambda_i|)/2$, λ_i represents the eigenvalues of H_p (see [19] for details of λ_i), $X_{H_p,R}$ and $X_{H_p,L}$ are the right and left eigenvectors of H_p. See [19] for the detailed description of how to calculate the boundary flux vectors.

3 The parallel unstructured multigrid method

This work focuses on examining the one-level method of parallelization strategies in the geometric domain decomposition technique, which employs MPI [22] as the communication library. And the multigrid domain decomposition (MG-DD) approach [17] is adopted for the multigrid parallelization. METIS [23] is used to decompose the flow domain into a set of S sub-domains that may be allocated to a set of P processors. The nodes and elements that are allocated uniquely to a processor are referred to as core mesh components in this work and each processor calculates the flow field variables and nodal gradients for it. Nodes and elements are separately renumbered as a result of the use of the SPMD (Single Programme Multiple Data) approach, i.e., each partition is treated as a separate flow domain and copies of the same code are used for all these domains for calculations. Each sub-domain is enclosed by a layer of nodes and elements, which overlap the neighboring sub-domains along the inter-partition boundaries and provide the necessary boundary conditions obtained from its neighbors. These outer most nodes in the layer are called ghost nodes because they lie in the neighboring domains and their flow variables are obtained by transferring the flow conditions from their corresponding images (core nodes in the neighbors) to them. Communication between these core and ghost nodes is based on MPI and proper synchronization between the computations in neighboring partitions ensures that the necessary boundary conditions are correctly exchanged between them. An algorithm developed in [17] is employed to identify the ghost nodes, overlapping elements and to write the individual grid files with local numbering for each partition. The main concept of this algorithm is that those elements along the inter-processor boundaries with nodes having different partition numbers are considered as overlapping elements which are cut through by partition lines. And those nodes that formed these elements are a mixture of *core* and *ghost* nodes with the outer most nodes being ghost nodes and inner ones core nodes. Basically, a *ghost* node of a partition is the mirror image of a corresponding *core* node in a neighboring partition.

The basic idea of the multigrid method is to carry out early iterations on a fine grid and then progressively transfer these flow field variables and residuals to a series of coarser grids. On the coarser grids, the low frequency errors become high frequency ones and they can be easily eliminated by a time stepping scheme. The flow equations are then solved on the coarser grids and the corrections are then interpolated back to the fine grid. The process is repeated over a sufficient number of times until satisfactory convergence on the fine grid is achieved. For ease of implementation, the non-nested mesh method using independently generated non-nested (or overset) coarse meshes is adopted [17]. Two

different cycle strategies have been investigated in the present work, which are V-cycle and W-cycle. The initial solution and residuals on the coarse grid $(h+1)$ are transferred from the fine grid (h) using volume-weighted transfer operators [17]. In order to drive the coarser grid solution using the fine grid residual, a forcing function is calculated at the first stage of the implicit Runge-Kutta time stepping scheme and subsequently added to the residual on the coarse grid. After calculating the variables on the coarsest grid, the corrections are evaluated and interpolated back level-by-level to the finest grid. The correction is the difference between the newly computed value on the coarser grid, and the initial value that was transferred from the finer grid. To improve efficiency for the simulation of viscous flows, the viscous terms are only evaluated on the fine grid but not evaluated on the coarser grids. Since the coarser grids are only used to cancel the dominating low frequency errors, this treatment does not affect the accuracy of the solution. The upwind-biased interpolation scheme is also set to first-order at the coarser levels.

The MG-DD approach is adopted in this study. This means that the non-nested multi-grids are independently generated first. Then domain decomposition of the finest grid is performed, which is followed by decomposition of the various coarse levels of grids guided by the finer grid partitions. This is achieved by using the fine grid partitions to infer the coarse level partitions (i.e., the coarse grid is to inherit its partition from that of its corresponding finest grid) and load balancing in the coarse mesh is reasonably well ensured. A two-level multi-grid and two sub-domains are used to demonstrate the procedure of partitioning the coarse grid using the fine grid. The main idea about this algorithm is that the fine grid is partitioned into two sub-domains according to the algorithm developed for single grids. And both the maximum and minimum values in the x and y-directions (X_{min}, X_{max}, Y_{min} and Y_{max}) of each partition for the fine grid are found. With these dimensions, an imaginary bounding box enclosing the sub-domains is formed. The main purpose of these bounding boxes is to identify the coarse nodes that fall within these boxes according to the fine grid partitions including those nodes beyond the sub-domain boundaries. The algorithms depicted in [17] are used to search for those actual coarse nodes that fall within a fine grid sub-domain and those nodes that fall beyond the sub-domain boundary are ignored. After classifying the respective coarse nodes according to which partition they belong to, then the *ghost* nodes and overlapping elements are identified using the algorithm depicted in Section 4. Individual grid files for the partitioned coarse grids and data structures for communication are then generated.

4 The immersed membrane method (IMM)

When the immersed body in a flow field is a thin structure, it will cause discontinuous fluid conditions across itself. Although velocity is continuous, the gradients of velocity, pressure, gradient of pressure and fluid stresses are quite different on both sides of the thin structure. In this work, the IMM [19, 20] is adopted, which treats thin structures or the boundaries of normal structures (fluid occupies one side of the membrane only) as

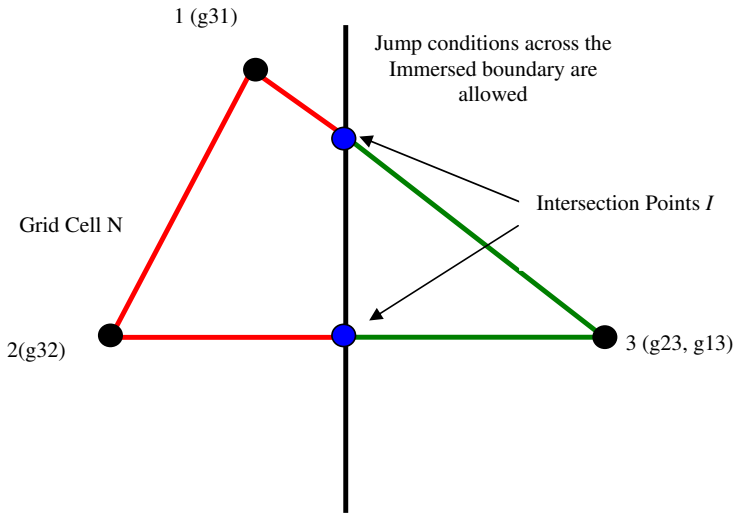

Figure 2: Real nodes and ghost nodes of a 2D grid cell.

membranes. This method uses an Eulerian background mesh for the fluid domain. When the membrane is present in the flow field, it will intersect with the Eulerian mesh. Taking a 2D mesh cell N for example, it is cut by the membrane as illustrated in Fig. 2. Fluid conditions on its nodes 1, 2 and 3 are discontinuous across the membrane. A set of imaginary ghost nodes are introduced here to replace the original nodes when they are on different sides of the membrane, i.e., nodes having discontinuous flow conditions. Considering node 1 on the left side of the membrane, node 3 stores discontinuous flow variables since it is on the other side of the membrane. And a ghost node g13 will be introduced to replace node 3 when the computation for node 1 involves conditions at node 3. In such a case, node 1 is called the real node and node 3 is called its corresponding ghost node g13. The naming convention for the ghost node is that it is prefixed with g for ghost and the first number denotes the real node and the second number the corresponding ghost node. Under this naming convention, all the ghost nodes are listed in Fig. 2. As described, the computations of convective fluxes are based on mesh edges. In the computation of convective flux along edge 23, for example, it involves the flow conditions at node 2 and 3. When the convective flux is computed for node 2, ghost node g23 is introduced to replace node 3 in the computation. Likewise, when the convective flux is calculated for node 3, ghost node g32 is introduced to replace node 2. Computations of viscous fluxes and gradients are based on mesh cells. For example, in cell N, when the viscous flux is computed for node 1, it involves fluid conditions of node 1, node 2 and node 3, and ghost node g13 is introduced to replace node 3 in the computation. The fluid variables at ghost nodes are extrapolated linearly from their corresponding real nodes based on the mesh edges which are intersected by the membrane. The extrapolated values are called ghost-node values. The novel feature of this IMM is that extrapolation of flow variables

to ghost nodes is always along cell edges instead of the traditional membrane-normal direction [24,25]. As a result, every node can hold multiple ghost nodes and thus multiple ghost values since a node can be connected by multiple edges, which is especially true for 3D unstructured grids. The selection of a particular ghost value depends on which edge and node the computation is for. This feature is extremely efficient compared with the wall/membrane-normal approach, because one does not need to: 1) construct wall normal lines; 2) find out what cell faces they intersect with; 3) locate the exact positions of the interaction points; 4) interpolate the flow conditions from nearby nodes to the intersection. These calculations are very complex for a 3D surface/membrane intersecting with a 3D unstructured mesh.

Extrapolations of ghost-node flow variables are illustrated in [19] using a 2^{nd}-order linear formulation. Then, later a higher-order MUSCL interpolation scheme is described in [21]. When the immersed body is an arbitrary object with a finite volume, the given concept and interpolation method still apply, except that only the flow variables of the ghost nodes within the immersed body need to be calculated because physically there are no fluid nodes inside the body. The Eulerian fluid nodes inside the object are just bypassed in flow calculations. These inner nodes can be efficiently searched for and identified by using an internal volume mesh within the object based on a quatree search in every time step, while the surface mesh of the object is used to perform the interpolation and extrapolation.

5 Coupling between flow solver and computational structural dynamics solver

We have already reported the development of a new finite volume based unstructured multi-grid computational structural dynamics solver and its detailed coupling method with our flow solver [19,21]. Please check the relevant references for further details.

6 Turbulence modeling

The computation of complex turbulent flows of practical interest to engineers continues to be a challenge. The Large-Eddy Simulation (LES) has been chosen to calculate the possible arising turbulent phenomena in this study. Despite its inherent superiority compared to other techniques for turbulence modeling, there are several issues which are still impeding the application of LES to practical problems. The present available models tend to break down for complex geometries and do not work well with low-order numerical schemes. Furthermore, such models have not been developed or adequately evaluated for many Subgrid-Scale (SGS) contributions to the compressible form of the filtered energy equation. In this work, a compressible dynamic form of Smagorinsky-Lilly SGS model has been implemented to calculate the SGS stress tensor. This model relies upon

the Germano identity, which has been generalized in order to be applied to other subgrid-terms arising in the filtered energy equation. Furthermore, an improved formulation of the dynamic mixed model has been proposed for the better representing the backscatter of turbulence energy which has been proven to be important for compressible flows. For illustration purpose, The SGS turbulent stress tensor σ_{ij} can be reformulated as

$$\sigma_{ij}=\overline{\rho}\left(\widetilde{u_iu_j}-\tilde{u}_i\tilde{u}_j\right)=L_{ij}+C_{ij}+R_{ij}, \tag{6.1}$$

where

$$L_{ij}=\overline{\rho}\left(\widetilde{\tilde{u}_i\tilde{u}_j}-\tilde{u}_i\tilde{u}_j\right), \tag{6.2}$$

$$C_{ij}=\overline{\rho}\left(\widetilde{\tilde{u}_iu_j''}+\widetilde{\tilde{u}_ju_i''}\right), \tag{6.3}$$

$$R_{ij}=\overline{\rho}\widetilde{u_i''u_j''}, \tag{6.4}$$

are respectively, the SGS Leonard, cross, and Reynolds stresses based on Favre averaging filtering, which implies the decomposition $f=\tilde{f}+f''$, where $\tilde{f}$ is the resolved component, and f'' is the unresolved component.

It is clear that the Leonard stress in Eq. (6.2) can be calculated directly from the filtered variables and does not need to be modeled. The cross stress is modeled with the scale similarity model [26, 27] with a coefficient of unity to ensure Galilean invariance overall model.

$$C_{ij}=\overline{\rho}\left(\tilde{u}_i\tilde{u}_j-\tilde{\tilde{u}}_i\tilde{\tilde{u}}_j\right). \tag{6.5}$$

The SGS Reynolds stress tensor is separated into deviatoric and isotropic parts, respectively, as follows:

$$R=R_D+R_I, \tag{6.6}$$

$$R_D=\overline{\rho}\left(\widetilde{u_i''u_j''}-\frac{1}{3}\widetilde{u_k''u_k''}\delta_{ij}\right), \tag{6.7}$$

$$R_I=\frac{1}{3}\overline{\rho}\widetilde{u_k''u_k''}\delta_{ij}, \tag{6.8}$$

where $\delta_{ij}=1$ if $i=j$ and zero otherwise. The models employed to estimate R_D and R_I can be described by following expressions:

$$R_I=\frac{\overline{\rho}\left\langle\left(\overline{\rho}\tilde{u}_k\tilde{u}_k\right)^{\wedge}-\left(\widehat{\overline{\rho}\tilde{u}_k}\widehat{\overline{\rho}\tilde{u}_k}\right)/\hat{\overline{\rho}}\right\rangle}{\left\langle 3\frac{\hat{\tilde{\Delta}}^2}{\Delta^2}\hat{\overline{\rho}}\left|S(\hat{\tilde{u}})\right|^2-3\left(\overline{\rho}\left|S(\tilde{u})\right|^2\right)^{\wedge}\right\rangle}\left|S(\tilde{u})\right|^2\delta_{ij}, \tag{6.9}$$

$$R_D=-2\overline{\rho}C_d\Delta^2\left|S(\tilde{u})\right|\left(S_{ij}(\tilde{u})-\frac{1}{3}S_{kk}(\tilde{u})\delta_{ij}\right), \tag{6.10}$$

$$C_d=\frac{\left\langle \begin{array}{l}\left[\left(\overline{\rho}\tilde{u}_i\tilde{u}_j\right)^{\wedge}-\left(\widehat{\overline{\rho}\tilde{u}_i}\widehat{\overline{\rho}\tilde{u}_j}\right)/\hat{\overline{\rho}}\right]-\left[\hat{\overline{\rho}}\left(\widehat{\tilde{\hat{u}}_i\tilde{\hat{u}}_j}-\hat{\tilde{\hat{u}}}_i\hat{\tilde{\hat{u}}}_j\right)-\left(\rho\left(\widetilde{\tilde{u}_i\tilde{u}_j}-\tilde{\tilde{u}}_i\tilde{\tilde{u}}_j\right)\right)^{\wedge}\right] \\ +\frac{1}{3}\left[\hat{\overline{\rho}}\left(\widehat{\tilde{\hat{u}}_k\tilde{\hat{u}}_k}-\hat{\tilde{\hat{u}}}_k\hat{\tilde{\hat{u}}}_k\right)-\left(\rho\left(\widetilde{\tilde{u}_k\tilde{u}_k}-\tilde{\tilde{u}}_k\tilde{\tilde{u}}_k\right)\right)^{\wedge}\right]\delta_{ij}-\frac{1}{3}(T_{mm}-\hat{\sigma}_{mm})\delta_{ij}\end{array}\right\rangle}{\left\langle \begin{array}{l}-2\hat{\overline{\rho}}\hat{\Delta}^2|S(\hat{\tilde{u}})|\left(S_{ij}(\hat{\tilde{u}})-\frac{1}{3}S_{kk}(\hat{\tilde{u}})\delta_{ij}\right) \\ +2\Delta^2\left[\left(\overline{\rho}|S(\tilde{u})|S_{ij}(\tilde{u})\right)^{\wedge}-\frac{1}{3}\left(\overline{\rho}|S(\tilde{u})|S_{kk}(\tilde{u})\delta_{ij}\right)^{\wedge}\right]\end{array}\right\rangle}, \tag{6.11}$$

where

$$T_{mm}-\hat{\sigma}_{mm}=(\overline{\rho}\tilde{u}_m\tilde{u}_m)^{\wedge}-(\widehat{\overline{\rho}\tilde{u}_m}\widehat{\overline{\rho}\tilde{u}_m})/\hat{\overline{\rho}}, \tag{6.12}$$

$$S_{ij}(\tilde{u})=\frac{1}{2}\left(\frac{\partial\tilde{u}_i}{\partial x_j}+\frac{\partial\tilde{u}_j}{\partial x_i}\right),\quad |S(\tilde{u})|=\sqrt{2S_{ij}(\tilde{u})S_{ij}(\tilde{u})}. \tag{6.13}$$

Apart from the grid-filter level, denoted by the ($\tilde{\cdot}$) corresponding to the filter width Δ, we introduce a test-filter, which is denoted by the hat ($\hat{\cdot}$) and corresponds to the filter width $\hat{\Delta}=2\Delta$. The consecutive application of these two filters, resulting in e.g., $\hat{\overline{\rho}}$, defines a filter which a filter width $\hat{\tilde{\Delta}}=k\Delta$ can be associated. This dynamic procedure requires an explicit filtering operation on the Favre averaging variables. The unstructured grid filtering was adapted from a new filtering approach basing on the least-squares technique [32]. This approach can filter a function to any given level of commutation error on unstructured grids. In this approach, the expression for the filtered values can be written in the form of a weighted sum,

$$\overline{\Phi}_0=\sum_{i=1}^{d_0}\overline{\omega}_{0i}\Phi_i. \tag{6.14}$$

The weight for linear filtering in three dimensions is given by [33]

$$\begin{aligned}\overline{\omega}_{0i}=\frac{1}{r_{44}^2}\Bigg[1-\left(\frac{r_{44}}{r_{11}}-\frac{r_{12}}{r_{11}}\frac{r_{24}}{r_{22}}-\frac{r_{34}}{r_{33}}\frac{r_{13}r_{22}-r_{12}r_{23}}{r_{11}r_{22}}\right)\Delta x_{0i}\\ -\frac{r_{24}r_{33}-r_{23}r_{34}}{r_{22}r_{33}}\Delta y_{0i}-\frac{r_{34}}{r_{33}}\Delta z_{0i}\Bigg],\end{aligned} \tag{6.15}$$

where the geometric terms r_{ij} are given by the following expressions:

$$r_{11}=\sqrt{\sum_{i=1}^{d_0}\Delta x_{0i}^2},\quad r_{12}=\frac{1}{r_{11}}\sum_{i=1}^{d_0}\Delta x_{0i}\Delta y_{0i},\quad r_{13}=\frac{1}{r_{11}}\sum_{i=1}^{d_0}\Delta x_{0i}\Delta z_{0i}, \tag{6.16}$$

$$r_{14}=\frac{1}{r_{11}}\sum_{i=1}^{d_0}\Delta x_{0i},\quad r_{22}=\sqrt{\sum_{i=1}^{d_0}\Delta y_{0i}^2-r_{12}^2},\quad r_{23}=\frac{1}{r_{22}}\left(\sum_{i=1}^{d_0}\Delta y_{0i}\Delta z_{0i}-r_{12}r_{13}\right), \tag{6.17}$$

$$r_{24}=\frac{1}{r_{22}}\left(\sum_{i=1}^{d_0}\Delta y_{0i}-r_{12}r_{14}\right),\quad r_{33}=\sqrt{\sum_{i=1}^{d_0}\Delta z_{0i}^2-\left(r_{13}^2+r_{23}^2\right)},\tag{6.18}$$

$$r_{34}=\frac{1}{r_{33}}\left(\sum_{i=1}^{d_0}\Delta z_{0i}-\left(r_{13}r_{14}+r_{23}r_{24}\right)\right),\quad r_{44}=\sqrt{d_0-\left(r_{14}^2+r_{24}^2+r_{34}^2\right)}.\tag{6.19}$$

In the interests of improving the spectral behavior of Eq. (6.14), it may be replaced by

$$\overline{\Phi}_0=\overline{\omega}_{00}\Phi_0+(1-\overline{\omega}_{00})\sum_{i=1}^{d_0}\overline{\omega}_{0i}\Phi_i,\tag{6.20}$$

where $1\leq\overline{\omega}_{00}\leq 1$ is a user-defined parameter, normally defined as 0.5.

The first issue that needs to be addressed in constructing an operational filtering using above method is the construction of stencil used for the filtering. For every vertex P, all of its 1th neighboring vertexes are used for constructing the filter stencil. If any of following rules is violated, its 2th neighboring vertexes will be further included to constructing the filter stencil.

Rule 1: The number of its 1th neighbors does not give the required stencil size, which is given by

$$d_{0,\min}=\frac{1}{D!}\prod_{i=1}^{D}(q+i),\tag{6.21}$$

where q denotes the degree of the highest derivative included in x. D the space dimension. Typically, for a second-order filtering operator in three dimensional space, $d_{0,\min}$ is 4.

Rule 2: r_{11}, r_{22}, r_{33}, or r_{44} is zero. When cooperating with the immersed membrane method (IMM), to avoid complex modifications of the filtering operator at the vicinity of the fluid-structure interface, a linear reconstruction procedure similar to that used for the velocity field is also applied on turbulent viscosity μ_t and Prandtl number P_{r_t}.

7 Results and discussion

7.1 Flow induced by an oscillating sphere in a closed cavity

To determine the overall accuracy of the method, we carry out a grid convergence study for a test problem where a solid oscillating sphere is immersed in a fluid enclosed within a cube with solid walls. This is to demonstrate the capability of the method in handling objects moving with large displacements. The sphere of diameter $D=1.0$ unit is placed initially at the center of the cube of dimension $2\times2\times2$ units and oscillates horizontally with a nondimensional time period of 1.0 and an amplitude of 0.25D. The oscillation is effected by moving the sphere as a rigid body with velocity given by

$$u=0.25\pi\sin(2\pi t),\quad v=w=0.$$

Table 1: Rate of convergence γ calculated for different error norms.

	Grids
Norm	40^3, 80^3, 160^3
L_∞	1.82
L_1	2.84
L_2	2.15

The Reynolds number (based on the sphere diameter and maximum velocity) has been set to 20. The following sequence of grid sizes is employed in performing the error analysis: 20^3, 40^3, 80^3 and 160^3. And the result on the 160^3 mesh is taken to be the "exact" solution for this case. A small time step of $\Delta t = 0.005$ is chosen for all these simulations in order to minimize the effect of temporal errors on the solution. The simulations are carried out for one oscillation period and the velocity components at each grid point are recorded for all the meshes under consideration at the end of the period. The instantaneous streamlines at the end of an oscillation cycle ($t = 1.0$T) are shown in Fig. 3.

For all the grids, the simulation is continued for one complete period, at the end of which the L_∞ and L_q norms of the u-velocity errors are calculated as follows:

$$\varepsilon_N^\infty = \max_{i=1,N^3} \left| u_i^{(N)} - u_i^e \right|, \quad \varepsilon_N^q = \left[\frac{1}{N^3} \sum_{i=1}^{N^3} \left| u_i^{(N)} - u_i^e \right|^q \right]^{1/q},$$

where ε_N^∞ and ε_N^q are the infinity and qth error norms, $u_i^{(N)}$ is the u-velocity component at the ith node of the N^3 mesh, and u_i^e is the 'exact' velocity field calculated on the 160^3 grid. The results of the grid convergence study are summarized in Fig. 4, which shows the variation of the L_∞, L_1 and L_2 norms of errors with grid spacing in logarithmic coordinates. The lines with slope one and two are also given as reference. It is evident from Fig. 4 that the method is second order accurate. To further demonstrate the accuracy of our method, we also use the Richardson estimation procedure to study the accuracy of the solver as in [25]. Let f^N denote the numerical solution on the N^3 mesh. Assume that the discrete solution is a γ-order approximation to its value f^{exact}, and the flow field is continuous and has no singularity points, then we have

$$\gamma = \frac{\log\left(\left\| f^N - f^{N/2} \right\| / \left\| f^{N/2} - f^{N/4} \right\| \right)}{\log 2},$$

where $\| \ \|$ denotes an error norm (L_∞, L_1 or L_2). If $\gamma \approx 2$ the solution is second-order accurate. We apply the above procedure for $N = 160$ (using solutions obtained on meshes 40^3, 80^3, and 160^3) to calculate γ for successively refined meshes. We apply the above procedure for $N = 160$ (using solutions obtained on meshes 40^3, 80^3, and 160^3) to calculate γ for successively refined meshes. And we use all three norms to compute the error and the results are summarized in Table 1, which strongly supports our assertion about the second-order accuracy of our method.

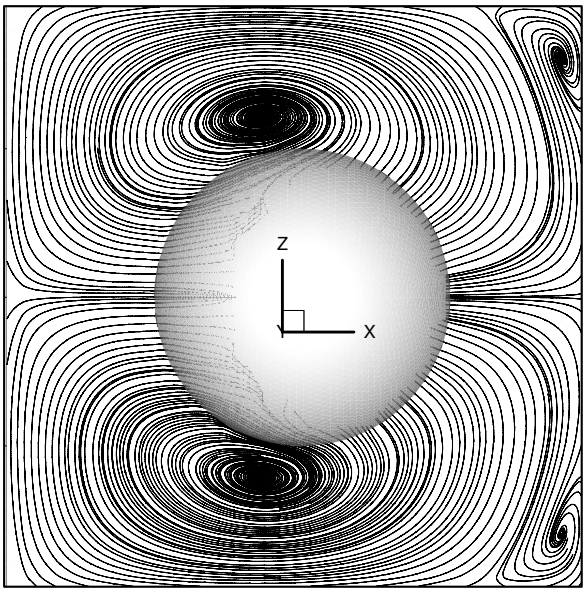

Figure 3: Instantaneous streamlines at $t=1.0$T for flow induced by an oscillating sphere in a closed cube filled with compressible and viscous fluid.

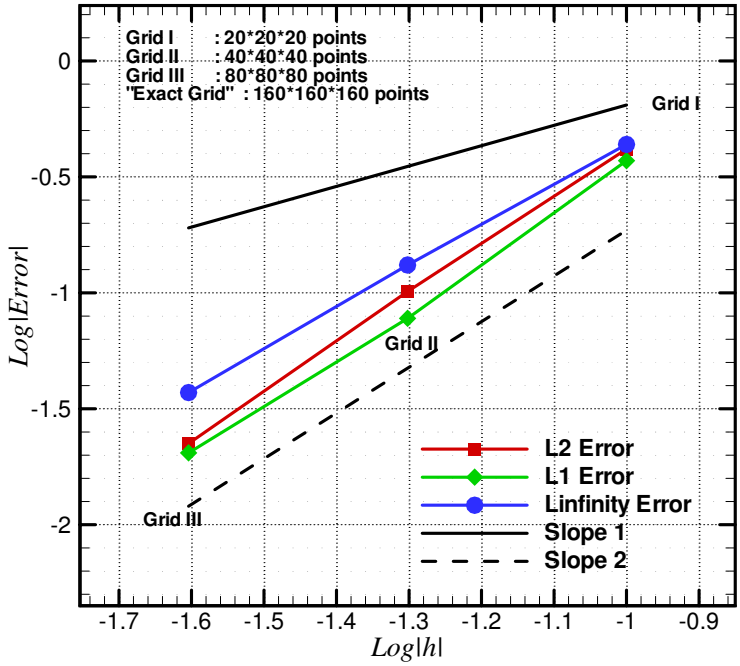

Figure 4: Convergence of the L_∞, L_1 and L_2 error norms for the velocity field of oscillating sphere in a closed cube. Slope 1 and Slope 2 are the reference lines for 1-order and 2-order accuracy respectively.

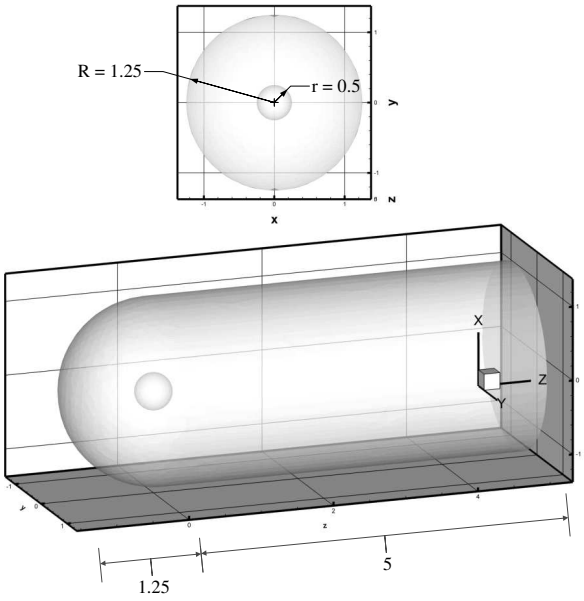

Figure 5: Computational domain for turbulent flow over a sphere.

7.2 Turbulent flow over a sphere at Re=10,000

This classical testing case was selected for the purpose of demonstrating the accuracy of both IMM and LES modeling. The basic structure of the flow past a sphere has been experimentally investigated using a variety of approaches, including flow visualization [34, 35], and hot wire measurements of velocity fluctuations in the wake [36]. Over a wide range of Reynolds numbers ($\mathrm{Re} = 280$ to about $\mathrm{Re}_{\mathrm{crit}} = 3.7\times10^5$ when the 'drag crisis' occurs) unsteady shedding of vortices from the sphere is observed. Previous work shows that some of the main features and parameters determining shedding, such as the way in which the vortices are shed in the wake and the values of the frequencies associated with the wake instabilities, change significantly over this range of Reynolds numbers (e.g., see [35,36]). Numerical simulations of the flow around a sphere offer a strong complement to experimental measurements and are useful for improving fundamental understanding of the flow field itself and the strengths and limitations of turbulence models used to predict flows with massive separation. In this study, the turbulent flow over a sphere at a subcritical Reynolds number (based on free stream velocity, U, and sphere diameter, D) of 10^4 is predicted using the proposed method. The problem definition is shown in Fig. 5. And the computational domain is meshed with 754,735 grid points and 4,443,428 tetrahedral elements. The grid density near the sphere surface and wake region is locally refined to better resolve the smaller-scale structures that arise via shear-layer instabilities. As pointed out by Mittal and Moin in [37] that even if the use of centered schemes for discretization of spatial derivatives is more natural for methods such as LES because these schemes do not contain numerical dissipation, for complex

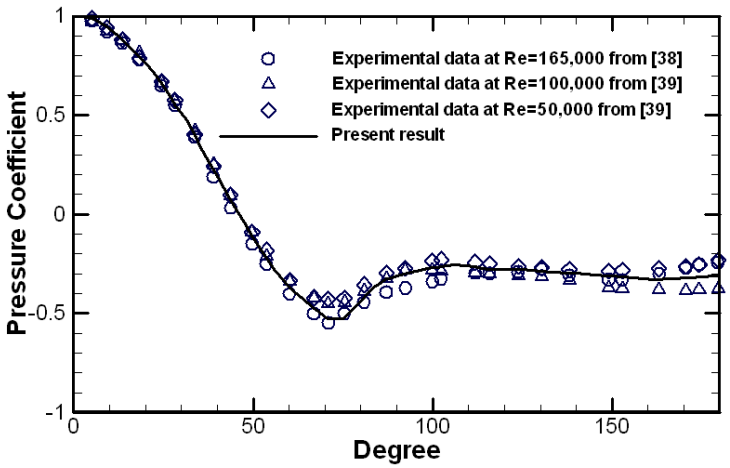

Figure 6: Mean pressure coefficient distribution over the sphere.

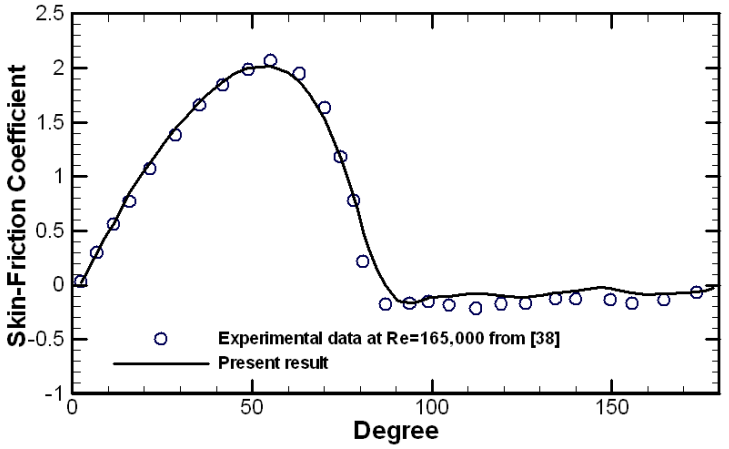

Figure 7: Mean skin-friction coefficient distribution over the sphere.

flows the results are sensitive to grid discontinuities and stretching ratios, formulation of boundary conditions, etc., all of which are critical to successful simulations. Upwind discretizations are less sensitive to these problems, at the cost of introducing numerical dissipation. This is because the inherent dissipation present in these schemes acts as a built-in aliasing control mechanism, which dissipates the energy content of the high frequencies in the flow. This dissipation can affect the accuracy of the computed solutions in regions where it is of the same order of magnitude as the one due to the molecular, or SGS dissipation. Thus the use of upwind schemes near the sphere was found to be necessary to prevent numerical instabilities from blowing up the solution. In current simulation, the third-order upwind scheme was employed to discretize the convection term.

Predictions of the mean pressure coefficient C_p and skin-friction coefficient C_τ are shown in Figs. 6 and 7, respectively. The distributions were obtained by averaging over

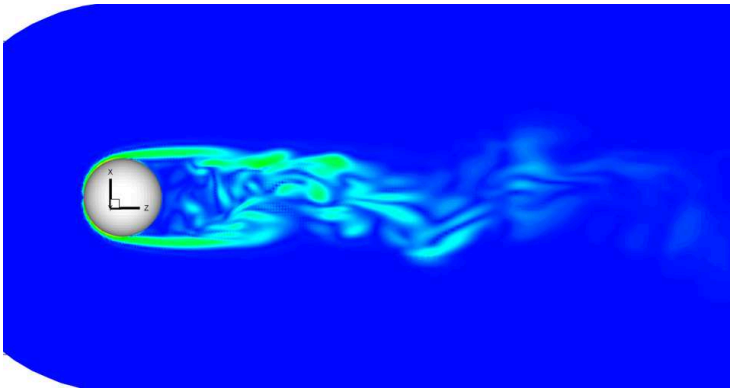

Figure 8: Instantaneous vorticity contours in the centre Y plane.

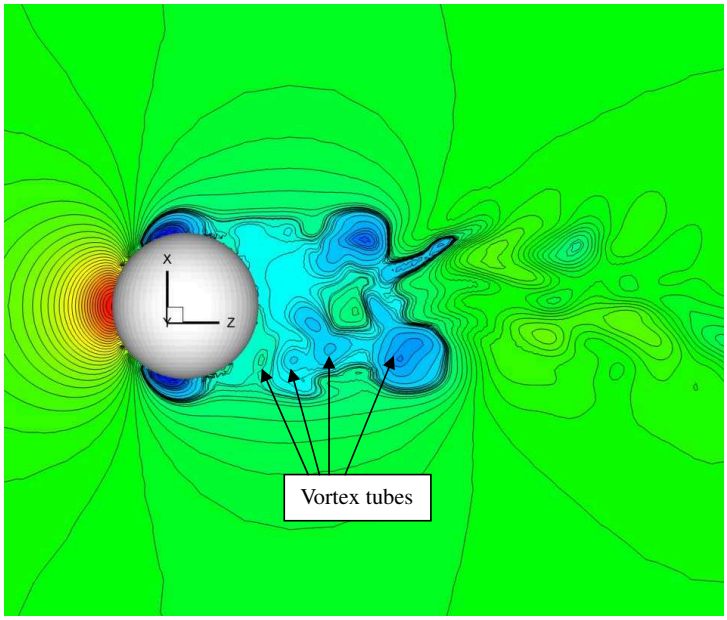

Figure 9: Instantaneous pressure contours in the centre Y plane.

the azimuthal (Y) direction and over at least five shedding cycles in an attempt to ensure adequate statistical convergence (averaging over 10 shedding cycles instead of 6 yielded less than a 5% change in the maximum values in the figures). The skin-friction coefficient C_τ is made dimensionless using $\rho U_\infty^2\sqrt{\mathrm{Re}}$, which is appropriate for the laminar boundary layers on the sphere and compared with the measurements of Achenbach [38] for the subcritical flow at $\mathrm{Re}=1.65\times10^5$. We are also showing the data of Bakic [39] for C_p at $\mathrm{Re}=5\times10^4$ and $\mathrm{Re}=1\times10^5$. Because the drag exhibits little variation between

$\mathrm{Re}=10^4$ and 1.65×10^5 and experimental investigations [36,39] show little change in the wake structure over the same interval, it is reasonable to expect that Reynolds number effects in the pressure and skin-friction coefficients are not important. Predictions of the mean streamwise drag C_d, shedding Strouhal number S_t are 0.389 and 0.178 respectively, and in good agreement with the experimentally measured values summarized in [38,40] ($C_d(\exp)=0.40$, $S_t(\exp)=0.18\sim0.195$). Instantaneous vorticity contours (see Fig. 8) show a wide range of scales in the separated region behind the sphere. The development of K-H instabilities along the periphery of the separated shear layer is clearly visible in the figure. This is also visualized in Fig. 9 as successive patches of high and low instantaneous pressure levels in the detached shear layers. The vortex tubes in the shear layers arise because of the interaction between the generally inviscid outer flow and the wake flow. As could be observed in Figs. 8 and 9, the average dimension of these tubes increases with the distance from the separation point.

7.3 Aero-elastic flutter of an ONERA M6 wing in transonic regime

The ONERA M6 wing is a classic CFD validation case for external flows because of its simple geometry combined with complexities of transonic flow. To evaluate the accuracy of the steady aerodynamic solution of proposed method, calculations for this wing are firstly carried out and compared to available experimental data. In this step, the wing body will be considered as a rigid solid and the structural domain will be ignored during the calculations. In reference [28], the ONERA M6 wing was tested throughoutly, and the pressures at various span-wise locations were recorded, which will be served as a main source of validation for the current study. Next, the former aerodynamic result is employed as the initial state in the full fluid-structure interaction simulation and the wing flutter will be predicted using the proposed method. The transonic flutter of the M6 wing has been extensively studied by many researchers [29–31] and numerous results are available for validation purpose.

The chord-wise time averaged pressure coefficient distributions on the wing surface at seven span-wise locations are presented in Fig. 10. The collected sample time mean pressure coefficients data at the same locations of the wing from referenced experiment [28] are also identified in the figures for ease of comparison. Figs. 11, 12 and 13 show the convergence histories of the lift coefficients C_l and the drag coefficients C_d, as well as the moment coefficient C_m along z-axis, respectively. A mean lift coefficient C_l and a mean drag coefficient C_d are derived from the limited time samples. They are listed in Table 2, and show good agreements with other published results. Results for three different flow domain grids are tabulated for ease of comparison. These results show that the predicted pressure coefficient agrees well with the experimental result, which in turn exhibits the accuracy of proposed method.

The M6 wing flutter computation begins by employing the converged solution from the former computation as the initial solution for the flow domain and the initial wing velocity is zero. The immersed structural domain (represented by the ONERA M6 wing

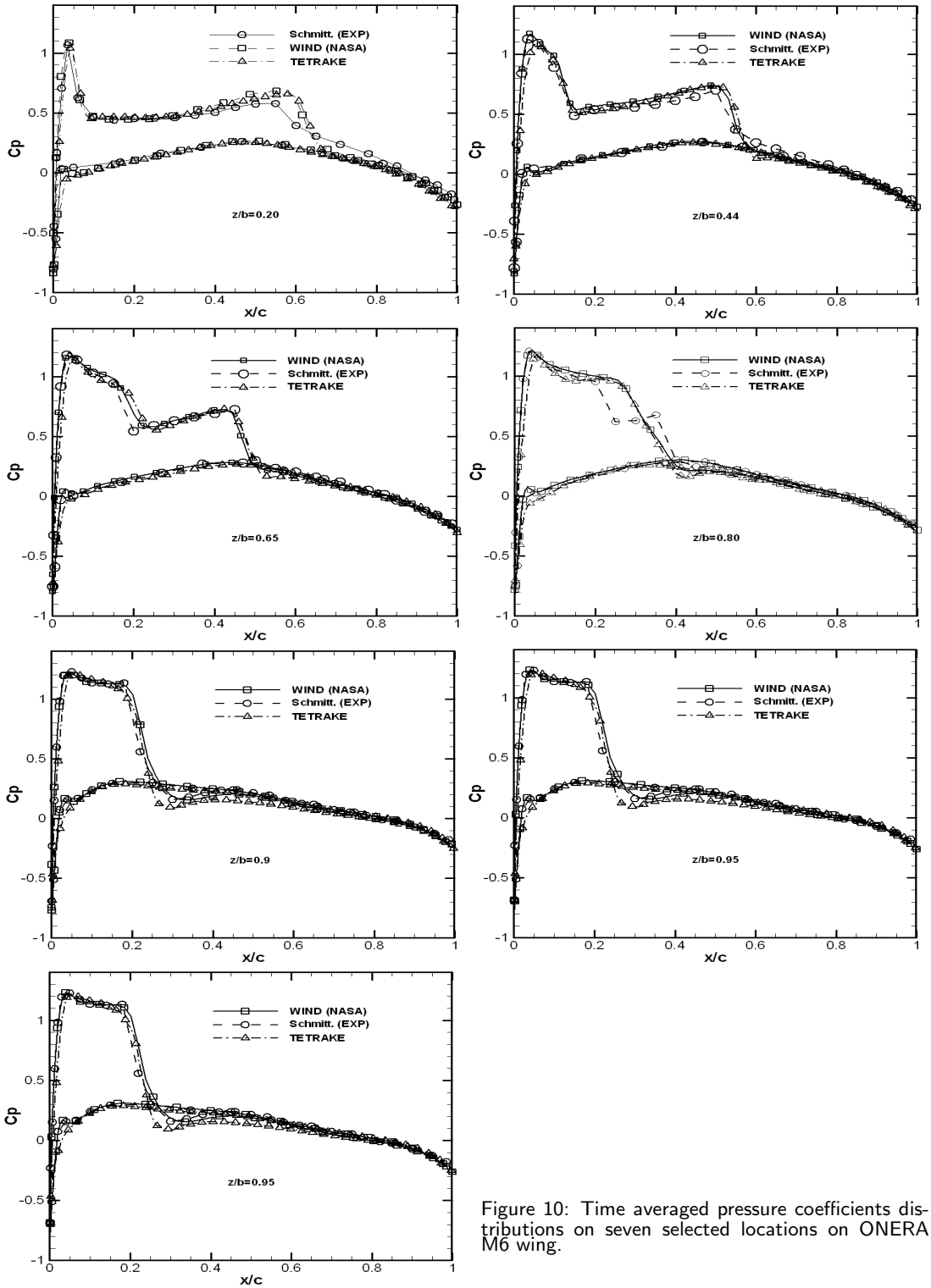

Figure 10: Time averaged pressure coefficients distributions on seven selected locations on ONERA M6 wing.

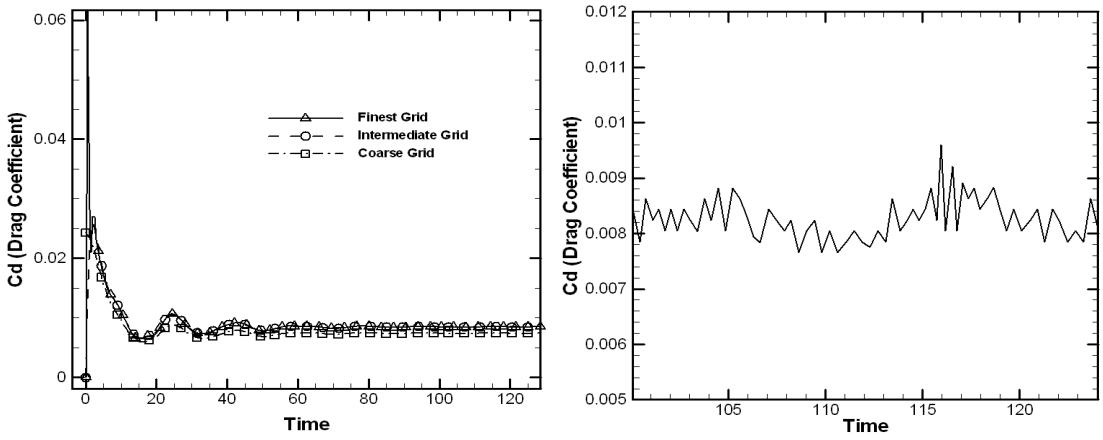

Figure 11: Time history of the drag coefficient of the ONERA M6 wing.

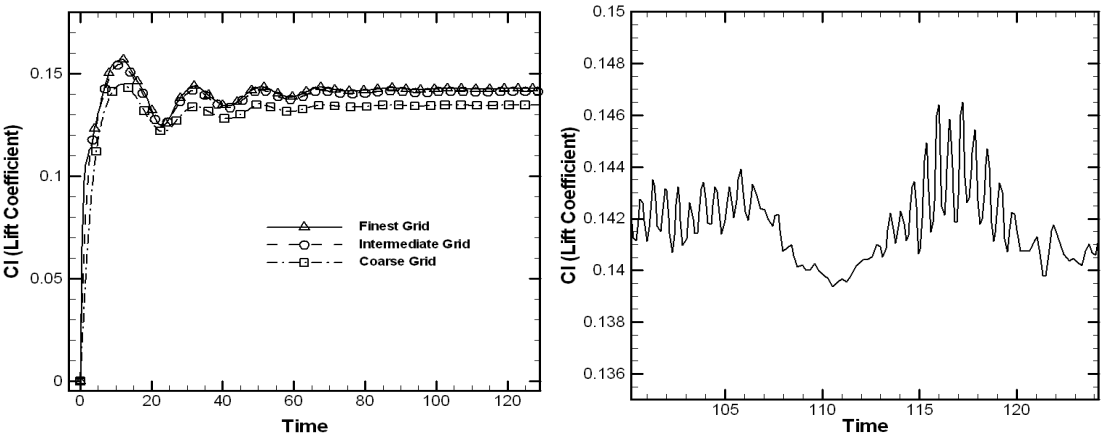

Figure 12: Time history of the lift coefficient of the ONERA M6 wing.

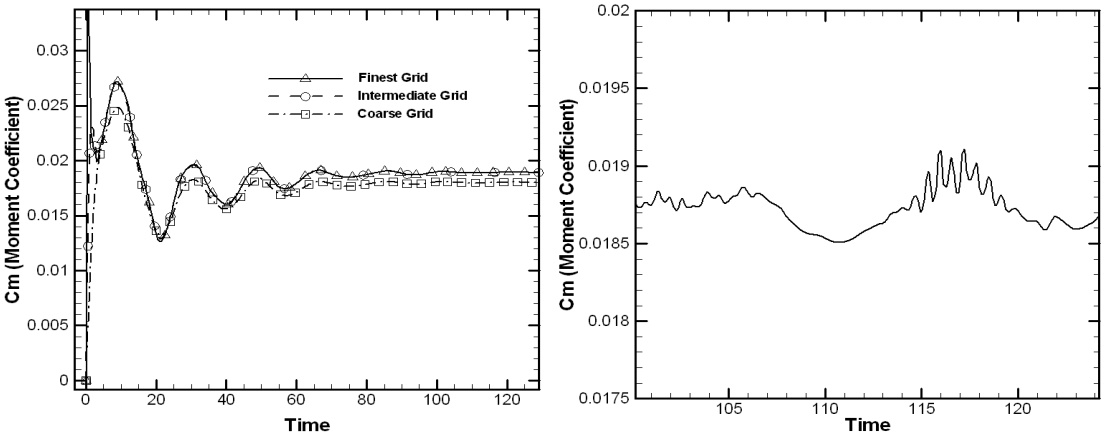

Figure 13: Time history of the moment coefficient of the ONERA M6 wing.

Table 2: Predicted mean Lift and Drag coefficients for M6 wing.

	Mean Lift Coefficient	Mean Drag Coefficient
Finest Grid	0.1428	0.0084704
Intermediate Grid	0.1420	0.0084663
Coarse Grid	0.1347	0.0073866

Note:

Grid one	1,769,472 elements, 306,577 nodes
Grid two	3,566,552 elements, 604,583 nodes
Grid three	5,308,416 elements, 909,521 nodes

body) is meshed by 42,150 tetrahedron elements and 9,094 nodes. The wing surface is meshed with 7,868 boundary triangles. The structure material parameters are as follows:

$$\text{Young's Modulus,} \quad E = 7.102 \times 10^{10}\, Pa,$$
$$\text{Material Density,} \quad \rho = 2770.0 kg/m^3,$$
$$\text{Poisson's Ratio,} \quad \nu = 0.32.$$

A fixed time step for both flow and structural domain is employed in the current simulation. Fig. 14 shows the flutter responses of the ONERA M6 wing at $M_\infty = 0.8395$ and 3.06^o angle of attack. While Fig. 15 shows the time variation of lift and drag coefficients for the ONERA M6 wing during the flutter process. As can be seen from the figures, the constant energy difference $(E_{tot} - W_E)$ is equal to the initial energy. This verifies that the energy exchange between the structure and the fluid satisfies the global conservation law for the total energy. We also find that the maximum tip displacement at leading edge is about 87.6% of that at trailing edge, which means that the flutter of M6 wing is a composition motion of the first bending and first torsion mode while dominated by the former.

8 Conclusion

In this paper, we describe the development of a parallel unstructured multi-grid preconditioned compressible Navier-Stokes solver implementing the IMM to calculate 3D unsteady low-Mach-number flows with rigid/elastic moving objects. The developed solver makes it possible to include moving objects in the flow fields with complexities that existing methods can not easily handle because it does not requires complicated interpolation of boundary conditions along surface normal direction. A dynamic SGS model has been implemented to calculate the SGS contribution. As to our knowledge, LES computation for compressible flow on unstructured grids with FSI capability can be hardly found in the literature and still posts a challenge to researchers. The preliminary results shown in the paper illustrate the capability and accuracy of the proposed method.

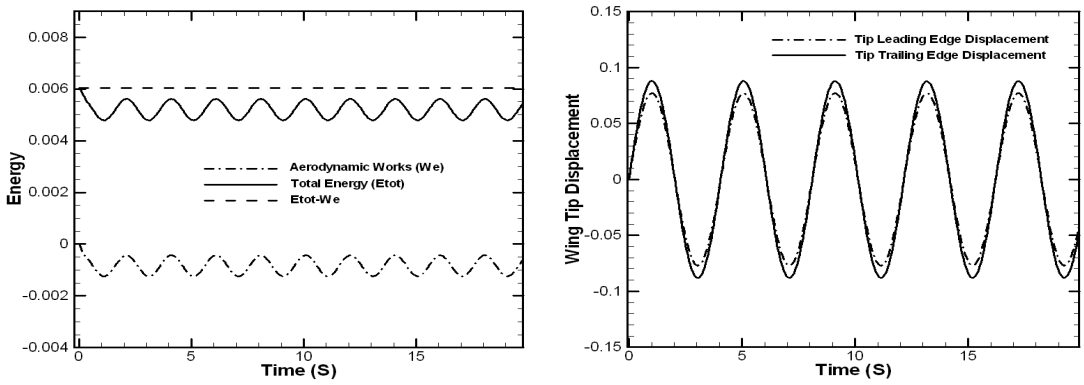

Figure 14: Flutter responses for ONERA M6 wing using intermediate grid at $M_\infty = 0.8395$ and 3.06° angle of attack (Displacement in y direction).

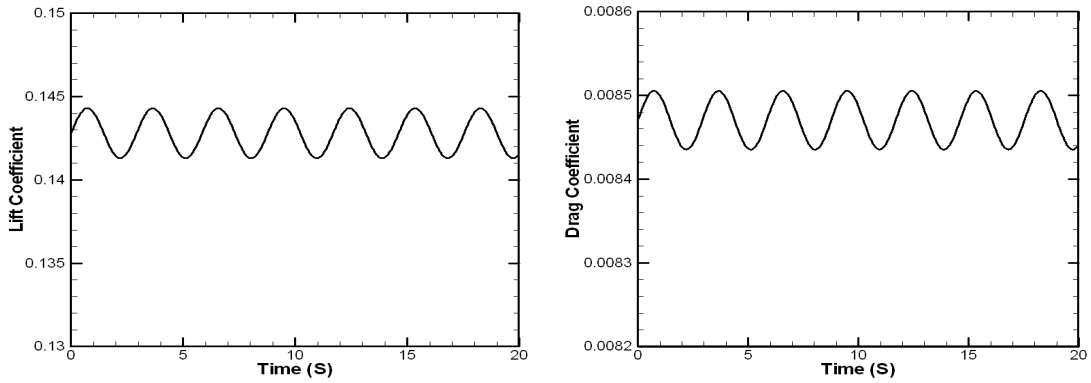

Figure 15: Time history of lift and drag coefficients for M6 wing fluttering using intermediate grid at $M_\infty = 0.8395$ and 3.06° angle of attack.

References

[1] X. Y. Luo and T. J. Pedley, A numerical simulation of unsteady flow in a two-dimensional collapsible channel, J. Fluid Mech. 314 (1996) 191-225.

[2] X. Y. Luo and T. J. Pedley, Numerical simulation of steady flow in a 2-D collapsible channel, J. Fluids Struct. 9 (1995) 149-197.

[3] X. Y. Luo and T. J. Pedley, The effect of wall inertia on flow in a two-dimensional collapsible channel, J. Fluid Mech. 363 (1998) 253-280.

[4] Y. Zhao and F. Ahmed, A general method for simulation of fluid flows with moving and compliant boundaries using unstructured grids, Comp. Methods Appl. Mech. Eng., 192(39-40) (2003), 4439-4466.

[5] Y. Zhao, C. H. Tai and F. Ahmed, Simulation of micro flows with moving boundaries using high-order upwind FV method on unstructured grids, Comput. Mech., 28(1) (2002), 66-75.

[6] A. L. Gaitonde, A dual-time method for two-dimensional unsteady incompressible flow calculations, Int. J. Numer. Meth. Engrg., 41 (1998) 1153-1166.

[7] A. L. Gaitonde, An artificial compressibility method for the solution of the 2D incompress-

ible Navier-Stokes equations, Report no. 715, Aero. Eng. Dept., Bristol University, 1995.
[8] M. Heil, Stokes flow in an elastic tube—A large-displacement fluid-structure interaction problem, Int. J. Numer. Meth. Fluids, 28 (1998) 243-265.
[9] E. Lefrancois, G. Dhatt and D. Vandaromme, Fluid-structure interaction with application to rocket engines, Int. J. Numer. Meth. Fluids, 30 (1999) 865-895.
[10] C. S. Peskin, Flow patterns around heart valves: A digital computer method for solving the equations of motion, PhD thesis, Albert Einstein College of Medicine, Yeshiva University, 1972.
[11] L. Zhu and C. S. Peskin, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, J. Comput. Phys., 179 (2002), 452-468.
[12] M. F. McCracken and C. S. Peskin, A vortex method for blood flow past heart valves, J. Comput. Phys., 35 (1980), 183-205.
[13] A. L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting, J. Comput. Phys., 56 (1984), 111-134.
[14] A. L. Fogelson, Mathematical and computational aspects of blood clotting, in: Proceedings of the 11th IMACS World Congress on System Simulation and Scientific Computation, Volume 3, 5-8, North Holland, New York, 1985.
[15] J. M. Stockie and S. I. Green, Simulating the motion of flexible pulp fibres using the immersed boundary method, J. Comput. Phys., 147 (1998), 147-165.
[16] C. H. Tai, Y. Zhao and K. M. Liew, Parallel computation of unsteady incompressible viscous flows around moving rigid bodies using an immersed object method with overlapping grids, J. Comput. Phys., 207 (2005), 151-172.
[17] C. H. Tai and Y. Zhao, Parallel unsteady incompressible viscous flow simulation using an unstructured multigrid method, J. Comput. Phys., 192(1) (2003), 277-311.
[18] A. Singh and Y. Zhao, Parallel unstructured dynamic grid direct Monte Carlo simulation of molecular gas dynamics and the associated thin film deposition, Comput. Fluid Solid Mech., in: K. J. Bathe (Ed.), The First MIT conference on Computational Fluid and Solid Mechanics, Elsevier Science, June 11-13, 2001, MIT, Cambridge, MA 02139, USA.
[19] X. Lv, Y. Zhao, et al., An efficient parallel/unstructured-multigrid preconditioned implicit method for simulating 3D unsteady compressible flows with moving objects, J. Comput. Phys., 215(2) (2006), 661-690.
[20] G. H. Xia, Y. Zhao and J. H. Yeo, An immersed membrane method for simulation of fluid-structure interaction in bio-fluid flows, in: 1st International BioEngineering Conference (IBEC 2004), 08-10, September, 2004.
[21] X. Lv, Y. Zhao, X. Y. Huang and G. H. Xia, A matrix-free implicit unstructured multigrid finite volume method for simulating structural dynamics and fluid-structure interaction, J. Comput. Phys., 225 (2007), 120-144.
[22] W. Gropp, E. Lusk and A. Skjellum, Using MPI: Portable Parallel Programming With The Message-Passing Interface, Cambridge, Massachusetts: The MIT Press, 1994.
[23] G. Karypis and V. Kumar, Metis: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices, Version 4.0, University of Minnesota, Department of Computer Science, September 1998.
[24] R. Mittal and G. Iaccarino, Immersed boundary method, Annu. Rev. Fluid Mech., 37 (2005), 239-261.
[25] A. Gilmanov and F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, J. Comput. Phys., 207(2) (2005), 457-492.

[26] Y. Zang, R. L. Street and J. R. Koseff, A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows, Phys. Fluids, 5 (1993), 3186-3196.
[27] B. Vreman, B. Geurts and H. Kuerten, On the formulation of the dynamic mixed subgrid-scale model, Phys. Fluids, 6 (1994), 4057-4059.
[28] V. Schmitt and F. Charpin, Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers, Experimental Data Base for Computer Program Assessment, Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138, May 1979.
[29] G. Hwang, Parallel finite element solutions of nonlinear aeroelastic and aero-servoelastic problems in three-dimensional transonic flows, PhD Dissertation, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, CA, 1997.
[30] O. O. Bendiksen and G. Y. Hwang, Transonic flutter suppression using dynamic twist control, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference and Exhibit, 37th, Salt Lake City, UT, Apr. 15-17, 1996, Technical Papers. Pt. 4 (A96-26801 06-39), Reston, VA, American Institute of Aeronautics and Astronautics, pp. 2670-2684, 1996.
[31] G. Seber and O. O. Bendiksen, Nonlinear flutter calculations using finite elements in a large deformation direct eulerian-lagrangian formulation, 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference Austin, Texas, 2005.
[32] A. Haslbacher and O. V. Vasilyev, Commutative discrete filtering on unstructured grids based on least-square techniques, J. Comput. Phys., 187 (2003), 197-211.
[33] A. Haslbacher, Private Communication, 2006.
[34] H. J. Kim and P. A. Durbin, Observations of the frequencies in a sphere wake and of drag increase by acoustic excitation, Phys. Fluids A, 31 (1988), 3260-3265.
[35] S. Taneda, Visual observations of the flow past a sphere at Reynolds numbers between 104 and 106, J. Fluid Mech. 85 (1978), 187-192.
[36] H. Sakamoto and H. Haniu, A study of vortex shedding from spheres in an uniform flow, J. Fluids Eng., 112 (1990), 386-392.
[37] R. Mittal and P. Moin, Suitability of upwind-biased finite difference schemes for large-eddy simulation of turbulent flows, AIAA J., 35 (1997), 1415-1417.
[38] E. Achenbach, Experiments on the flow past spheres at very high Reynolds numbers, J. Fluid Mech., 54 (1972), 565-575.
[39] V. Bakic, Experimental investigation of turbulent flows around a sphere, PhD Thesis, TUHH Hamburg, Germany, 2002.
[40] M. Provansal, From the double vortex street behind a cylinder to the wake of a sphere, in: K. Hourigan, T. Leweke, M. C. Thompson and C. H. Williamson (Eds), Proceedings of the Conference on Bluff Body Wakes and Vortex-Induced Vibration (BBVIV3), Port Douglas, Queensland, December 17-20, 2002.