Time efficient aeroelastic simulations based on radial basis functions

Wen Liu*, ChengDe Huang, Guowei Yang

Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China

A R T I C L E   I N F O

Article history:
Received 8 October 2015
Received in revised form 19 October 2016
Accepted 27 October 2016
Available online 2 November 2016

Keywords:
Aeroelastics
Fluid–structure interpolation
Mesh deformation
Radial basis functions

A B S T R A C T

Aeroelasticity studies the interaction between aerodynamic forces and structural responses, and is one of the fundamental problems to be considered in the design of modern aircraft. The fluid–structure interpolation (FSI) and mesh deformation are two key issues in the CFD–CSD coupling approach (the partitioned approach), which is the mainstream numerical strategy in aeroelastic simulations. In this paper, a time efficient coupling scheme is developed based on the radial basis function interpolations. During the FSI process, the positive definite system of linear equations is constructed with the introduction of pseudo structural forces. The acting forces on the structural nodes can be calculated more efficiently via the solution of the linear system, avoiding the costly computations of the aerodynamic/structural coupling matrix. The multi-layer sequential mesh motion algorithm (MSM) is proposed to improve the efficiency of the volume mesh deformations, which is adequate for large-scale time dependent applications with frequent mesh updates. Two-dimensional mesh motion cases show that the MSM algorithm can reduce the computing cost significantly compared to the standard RBF-based method. The computations of the AGARD 445.6 wing flutter and the static deflections of the three-dimensional high-aspect-ratio aircraft demonstrate that the developed coupling scheme is applicable to both dynamic and static aeroelastic problems.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

As a typical fluid–structure interaction phenomenon, aeroelasticity is one of the fundamental problems to be considered in the design of modern aircraft. The numerical simulations of aeroelastic problems can be classified broadly under two categories: the monolithic approach that solves the aerodynamic forces and structural responses simultaneously using the integrated aero-structural solver; and the partitioned approach (coupling approach) that solves the aerodynamic forces and structural motions in a separate manner with additional interfacing technique to communicate between different solvers. The monolithic approach usually requires the solutions of integrated aero-structural equations in both Eulerian and Lagrangian systems [1–3]. This leads to the matrices being orders of magnitudes stiffer for structure system than fluid system, which makes it virtually impossible to solve the equations for large-scale problems. The partitioned approach, however, allows the use of existing tools for computational fluid dynamics (CFD) and computational structure dynamics (CSD). This flexibility of choosing different solvers maintains the independence of the fluid and structure system, and thereby is favored in current computational aeroelastic researches.

* Corresponding author.
E-mail address: liuwen@imech.ac.cn (W. Liu).

http://dx.doi.org/10.1016/j.jcp.2016.10.063
0021-9991/© 2016 Elsevier Inc. All rights reserved.
One issue for the partitioned approach is the data communication between fluid and structure systems. Because both the aerodynamic and structural domains are discretized in a physically different manner, the two meshes will generally not coincide at the fluid–structure interface. In order for the calculation to proceed, it is necessary to transfer the aerodynamic loads from the aerodynamic surface to the structural nodes so that the deflections can be computed by the CSD method. These deflections then need to be transferred back to deform the aerodynamic surface consistently for the CFD computation. Numerous approaches have been investigated for this CFD–CSD interpolation, including the methods of Infinite Plate Spline (IPS) [4], Inverse Isoparametric Mapping (IIM) [5,6], Constant Volume Tetrahedra (CVT) [7,8], and Boundary Element (BEM) [9,10], etc. The partitioned approach typically involves motions of fluid–structure interface due to the deflection of the structure. To be able to perform the CFD computations accurately with the moving boundaries, it is usually necessary to adapt the fluid volume mesh based on the deformation of flow boundaries. It is a natural choice to regenerate the entire CFD mesh according to changes of the fluid domain. However, the generation of a complex grid is usually nontrivial and computationally expensive, especially in the case that the mesh update needs to be performed each time step in an unsteady flow computation. Furthermore, the grid topology is often not preserved by generating new mesh, which may introduce more uncertainties for the numerical errors. Mesh deformation is another choice to update the CFD grid, which could potentially be a more convenient and efficient approach with mesh topology preserved. Many ways to deform the volume mesh have been investigated, such as Transfinite Interpolation (TFI) [11], Spring Analogy (SA) [12–14], PDE solution method [15,16], etc. However, all the mesh type dependent methods require the knowledge of the grid connectivity. The displacement of interior grids usually needs to be solved by a system of equations including all the points in the domain, and therefore can be computationally expensive. Hence, for large meshes or time-dependent simulations with frequent mesh updates, effective methods requiring no connectivity information are preferable. Liu et al. [17] has developed a fast approach that utilizes the Delaunay mapping to interpolate deformations with a simple meshless interpolation algorithm. Although this scheme is quite effective, it makes no attempt to preserve mesh orthogonality near the deforming boundary, and large boundary movements may cause some points move across each other which results in negative coefficients for the interpolation.

Recently, Rendall and Allen [18] presented a multivariate interpolation scheme using RBF, which leads to a unified formulation for the fluid–structure interpolation (FSI) and mesh motion problems. The proposed RBF-based method was demonstrated to be a generic approach applicable to arbitrary mesh type. The displacement of the targeted point (aero-surface node in the FSI or volume node in the mesh deformation) can be interpolated, with no connectivity information required, by the control points at the boundary (structure node in the FSI or aero-surface node in the mesh deformation). A small system of equations, only involving the boundary points, has to be solved to determine the interpolation function, and no elaborate computations are required to evaluate the movements of targeted points except simple matrix–vector multiplications. The computational effort associated with the volume mesh motion scales with $N_{vp} \times N_{vp}$ ($N_{vp}$ is the total number of volume points, and $N_{vp}$ denotes the number of control points at the boundary). Rendall and Allen [19] improved the performance of the RBF interpolation method by the implementation of the ‘greedy’ algorithm to reduce the size of the control points at the boundaries, sacrificing the accuracy of the deformation at the surface boundary with an acceptable error. In time-dependent simulations with large meshes, the volume mesh update may still consume significant computing resources, since the volume usually contain millions of multiplication and summation operations at each time step where the number of volume grid $N_{vp}$ is massive. Therefore, it is necessary to further reduce the size of volume mesh in the interpolation to improve the efficiency. Michler [20] presented a confinement technique, which locally restricts the mesh deformation to the vicinity of the moving components enclosed by the control surface. However, the confinement technique was initially designed for mesh deformations with prescribed local component deformations, and thereby was highly configuration dependent.

In the present paper, a time efficient coupling scheme is developed based on the RBF interpolations. During the FSI process, the positive definite system of linear equations is constructed with the introduction of pseudo structural forces. The acting forces on the structural node can be calculated more efficiently via the solution of the linear system, avoiding the costly computations of the aerodynamic/structural coupling matrix in the previous study [18]. For mesh motions, the multi-layer sequential mesh motion (MSM) algorithm is proposed to improve the efficiency of the volume mesh deformations in large mesh applications. In MSM, the evaluation of volume mesh motions can be split into sequential sub-layers, each of which has its specific interpolation relation. The total computational effort equals to the summation of the local cost in each sub-layer where both surface control points and volume points can be reduced considerably, and therefore the performance of the mesh deformation can be effectively improved. The MSM algorithm has great advantages in deforming the mesh of multi-element geometry with local component deflections (e.g. the flap deflection in the high-lift wing system). By confining the mesh deformation in the vicinity of the moving component surface, the use of MSM could effectively reduce the overall computational complexities, and more importantly, maintain the grid consistency in the majority part of the domain which makes the deformed mesh more robust for the time-dependent simulations. Moreover, the MSM algorithm can be implemented easily with no requirement of additional boundary conditions, and has the potential to become a generic mesh deformation method.

The organization of present paper is as follows. Section 2 describes the partitioned strategy for the fluid–structure interaction simulations. Section 3 discusses the fluid–structure interpolation using RBF, and a time efficient solution strategy is derived to calculate the structural forces. The MSM algorithm for volume mesh deformations is proposed in Section 4. The
applicability of the developed coupling scheme is verified by both dynamic and static aeroelastic test cases in Section 5, followed by the conclusions in Section 6.

2. Numerical strategy for the partitioned approach

The fluid–structure interaction problem typically involves fluid motion, structure dynamics, and the communication between fluid and structure domains, which can be described by the system of equations as follows,

\[
\frac{\partial}{\partial t} \left( \Phi(x, t) \right) + \frac{\partial}{\partial x} \left( F(\Phi(x, t)) \right) + \frac{\partial}{\partial x} \left( F_v(\Phi(x, t)) \right) = 0; \tag{1}
\]

\[
\mathcal{S}(u) = \mathcal{F}(\Phi, x); \tag{2}
\]

\[
x = \mathcal{H}(u). \tag{3}
\]

Eq. (1) represents the general form of governing equations of fluid motion, where \( \Phi(x, t) \) is the solution vector in the Eulerian coordinates, and \( F, F_v \) denote the inviscid and viscous flux vectors respectively. Eq. (2) represents the general form of structure dynamics equation. Here \( u \) is the vector of displacements at the structural nodes and \( \mathcal{F} \) represents the external forces acting on the structural nodes, which are determined by the integration of the aerodynamic loads in the fluid domain. Eq. (3) describes the continuity condition at the fluid–structure interface, which ensures the consistency of the displacement and deformation between the fluid and structure domain.

The partitioned approach decouples the fluid–structure interaction problem, and solves the aerodynamic forces and structural motions separately. This maintains the independence of the fluid and structure system, and allows the use of existing tools for CFD and CSD computations. In general, the aerodynamic and structural domains are discretized in a physically different manner: the spatial discretization of the fluid field is often based on the finite volume formulation in the Eulerian coordinates, while the structure domain is usually discretized using finite element method in the Lagrangian system. Moreover, the structural discretization is not constrained to conform to the solid boundary compared to the CFD mesh where sufficiently smooth aerodynamic shape is mandatory. This means that the two domains do not generally share the same nodes at the interface, and interpolations need to be implemented to transfer forces and displacements between the two systems following physical laws such as continuity and energy conservation. An illustration of the partitioned approach is shown in Fig. 1.

The procedure of fluid–structure coupling simulations is as follows, sketched in Fig. 2.

1. Compute the aerodynamic loads using CFD methods;
2. Construct the interpolation relation between structural nodes and aerodynamic nodes at the interface;
3. Derive the relation between structural forces and aerodynamic loads based on the equivalence of virtual work, and calculate the structural forces;
4. Obtain the displacements at the structural nodes through the CSD computations;
5. Deform the aerodynamic mesh at the interface using the FSI in (2), and update the CFD volume mesh;
6. Repeat (1) to (5) until static equilibrium is achieved (for static elastic problem) or one physical time step is finished (for dynamic elastic problem).

3. Fluid–structure interpolation

The interpolations with RBF are based on the spatial positions of control points only, and can be performed on arbitrary point clouds with no connectivity information required. The general form of RBF interpolation is

\[
\mathcal{S}(x) = \sum_{j=1}^{N} \alpha_j \phi (\|x - x_j\|) + p(x); \tag{4}
\]
where $\phi(||\cdot||)$ represents a specific RBF, $N$ is the number of control points, and $p$ is a low degree polynomial. The coefficients $\alpha_j$ and the polynomial $p$ are determined by the interpolation condition:

$$S(x_j) = g_j, \quad 1 \leq j \leq N;$$

and the additional condition:

$$\sum_{j=1}^{N} \alpha_j q(x_j) = 0;$$

for all polynomials $q$ with a degree of $\text{deg}(q) \leq \text{deg}(p)$. It can be proved that, given a positive definite function $\phi$, there exists exactly one interpolation of the form (4) that satisfies both (5) and (6) [21]. In the FSI, linear polynomials are usually adopted to recover the rigid body translation and rotation, which means that if the body simply translates or rotates the interpolation relation at the fluid–structure interface stays unchanged. The polynomial term can be written as:

$$p(x) = \gamma_0 + \gamma_1 \cdot x_i;$$

where $x_i$ is the direction component of $x$. It should be noted that the introduced polynomial term not only ensure that the interpolant is uniquely determined, but also constrain the total force and moment to be conserved. This fact is not obvious and will be discussed in detail later in this section.

The FSI starts with the construction of interpolation relation between structural nodes and aerodynamic nodes at the interface. For a three-dimensional problem, the interpolant can be written as,

$$\Delta x_s = \sum_{j=1}^{N} \alpha_j \phi(||x_s - x_j||) + \gamma_0 + \gamma_1 \cdot x_1 + \gamma_2 \cdot x_2 + \gamma_3 \cdot x_3;$$

where subscript $s$ denotes the structural nodes, $N$ is the total number of nodes on the structure domain. The matrix form of Eq. (8) is

$$
\begin{pmatrix}
\Delta x_{s1} \\
\vdots \\
\Delta x_{sN}
\end{pmatrix}
= 
\begin{pmatrix}
1 & x_{s1} & y_{s1} & z_{s1} & \phi_{s1s1} & \cdots & \phi_{s1sN} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_{sN} & y_{sN} & z_{sN} & \phi_{sNs1} & \cdots & \phi_{sNsN}
\end{pmatrix}
\begin{pmatrix}
\gamma_0 \\
\gamma_1 \\
\gamma_2 \\
\gamma_3 \\
\alpha_1 \\
\vdots \\
\alpha_N
\end{pmatrix}.
$$

Using the initial condition (5) and additional condition (6), the interpolation coefficients can be solved by

$$
\begin{pmatrix}
0 \\
\Delta x_s
\end{pmatrix}
= 
\begin{pmatrix}
0 & P^T \\
P & M
\end{pmatrix}
\lambda = \mathbf{C}_{ss} \lambda;
$$

where

$$\mathbf{C}_{ss} = 
\begin{pmatrix}
S(x_1) & \cdots & S(x_N) \\
\vdots & \ddots & \vdots \\
S(x_N) & \cdots & S(x_1)
\end{pmatrix}.$$
where

\[
P = \begin{pmatrix}
1 & \cdots & 1 \\
x_1 & \cdots & x_{2N} \\
y_1 & \cdots & y_{2N} \\
z_1 & \cdots & z_{2N}
\end{pmatrix},
\quad
M = \begin{pmatrix}
\phi_{3s1} & \cdots & \phi_{3sN} \\
\vdots & \ddots & \vdots \\
\phi_{3Ns1} & \cdots & \phi_{3NsN}
\end{pmatrix},
\quad
\lambda = \begin{pmatrix}
\gamma_0 \\
\gamma_1 \\
\gamma_2 \\
\vdots \\
\alpha_1 \\
\vdots \\
\alpha_N
\end{pmatrix},
\quad
\phi_{3s2} = \phi(\|x_1 - x_2\|).
\]

The square matrix \( C_{ss} \) based on a specific radial basis function and the linear polynomials is positive definite \([21]\), which means the inverse matrix exists. Thus the solution of the interpolation coefficients can be written as

\[
\lambda = C_{ss}^{-1} \begin{pmatrix} 0 \\ \Delta \mathbf{x}_s \end{pmatrix}. 
\tag{11}
\]

Since the displacements and deformations are consistent at the interface between fluid and structure domains, the interpolation based on the structural nodes can be applied to the aerodynamic nodes as well. Therefore, the displacements at the aerodynamic surface nodes can be interpolated by

\[
\Delta \mathbf{x}_s = A_{ss} \lambda;
\tag{12}
\]

where

\[
A_{ss} = \begin{pmatrix} \mathbf{Q}^T & \mathbf{K}^T \end{pmatrix},
\quad
\mathbf{Q} = \begin{pmatrix} 1 & \cdots & 1 \\
x_1 & \cdots & x_M \\
y_1 & \cdots & y_M \\
z_1 & \cdots & z_M
\end{pmatrix},
\quad
\mathbf{K} = \begin{pmatrix}
\phi_{a1s1} & \cdots & \phi_{aMs1} \\
\vdots & \ddots & \vdots \\
\phi_{a1Ns1} & \cdots & \phi_{aMsN}
\end{pmatrix}
\]

and \( M \) is the total number of aerodynamic surface nodes. Finally, the mathematical relation between structural nodes and aerodynamic surface nodes can be constructed as follows,

\[
\Delta \mathbf{x}_s = A_{ss} \lambda = A_{ss} C_{ss}^{-1} \begin{pmatrix} 0 \\ \Delta \mathbf{x}_s \end{pmatrix} = A_{ss} \begin{pmatrix} \mathbf{M}_p \mathbf{P}^{-1} \mathbf{M}_p \mathbf{P}^{-1} \end{pmatrix} \Delta \mathbf{x}_s;
\tag{13}
\]

where

\[
\mathbf{M}_p = (\mathbf{P} \mathbf{M}^{-1} \mathbf{P}^T)^{-1}.
\]

Thus, the coupling matrix relates the aerodynamic displacements to the structural ones can be obtained,

\[
\mathbf{H} = A_{ss} \begin{pmatrix} \mathbf{M}_p \mathbf{P}^{-1} \\
\mathbf{M}^{-1} - \mathbf{M}^{-1} \mathbf{P}^T \mathbf{M}_p \mathbf{P}^{-1} \end{pmatrix}.
\tag{14}
\]

The forces acting on the structural nodes then can be obtained given the aerodynamic loads, considering the equivalence of virtual work,

\[
\delta \mathbf{W} = \delta \mathbf{u}_s^T \mathbf{f}_s = \delta \mathbf{u}_a^T \mathbf{f}_a;
\tag{15}
\]

where \( \delta \mathbf{W} \) represents the virtual work, \( \delta \mathbf{u} \) denotes the virtual displacement, and \( \mathbf{f} \) is the force vector. The subscript \( s \) and \( a \) represents structural and aerodynamic nodes respectively. Eq. (13) can be written as

\[
\delta \mathbf{u}_a = \mathbf{H} \delta \mathbf{u}_s;
\tag{16}
\]

together with Eq. (15), we have

\[
\delta \mathbf{u}_a^T \mathbf{f}_s = \delta \mathbf{u}_a^T \mathbf{H}^T \mathbf{f}_a.
\tag{17}
\]

Since the virtual displacement is arbitrary, then the relation between structural forces and aerodynamic loads can be determined as

\[
\mathbf{f}_a = \mathbf{H}^T \mathbf{f}_s.
\tag{18}
\]

In the previous study [18], the structural forces are computed using the coupling matrix \( \mathbf{H} \). The calculation of the coupling matrix will inevitably involve massive operations of matrix multiplication and inversion, which has the time complexity of approximately \( 14N^2/3 + 2MN^2 \) \((N \gg 4)\). From engineering perspective, the direct calculation of \( \mathbf{H} \) turns to be extremely time consuming when the structural system has very large number of \( N \), and would reduce the efficiency of the fluid–structure coupling significantly.
In this paper, a positive definite system of linear equations is constructed by the introduction of pseudo structural forces. The acting forces on the structural nodes then can be calculated more efficiently via the solution of the linear system, avoiding the costly computations of the aerodynamic/structural coupling matrix. According to Eq. (15), we construct a system of linear equations as follows,

$$\begin{pmatrix} 0 \\ \Delta x_s \end{pmatrix}^T \begin{pmatrix} f_i \\ f_5 \end{pmatrix} = \Delta x_s f_5; \tag{19}$$

where $f_i$ is defined as the pseudo structural forces, which are not real forces acting on the structural nodes. Using Eq. (10) and Eq. (12), Eq. (19) can be written as

$$\lambda^T C_{ss}^T \begin{pmatrix} f_i \\ f_5 \end{pmatrix} = \lambda^T A_{ss}^T f_5. \tag{20}$$

Then, the structural forces can be calculated by the solution of the positive definite system of linear equations as follows

$$C_{ss}^T \begin{pmatrix} f_i \\ f_5 \end{pmatrix} = A_{ss}^T f_5. \tag{21}$$

The time complexity of solving Eq. (21) is approximately $2N^3/3 + 2MN (N \gg 4)$, which reduces the computing cost by a factor of $O(n)$. It must be noted that Eq. (20) and Eq. (21) are not equivalent, since the interpolation coefficients $\lambda$ are not arbitrary. However, it can be proved that the solution $f_i$ provided by Eq. (21) is equivalent to the solution given by Eq. (18), which means that the structural forces can be accurately computed using Eq. (21). In fact, according to Eq. (21), we have

$$\begin{pmatrix} f_i \\ f_5 \end{pmatrix} = \begin{pmatrix} 0 \\ P^T \end{pmatrix}^{-1} A_{ss}^T f_5 = \begin{pmatrix} -M_p \\ M^{-1} P^T M_p \end{pmatrix} \begin{pmatrix} -M_p \\ M^{-1} P^T M_p \end{pmatrix}^T A_{ss}^T f_5;$$

hence

$$f_5 = \begin{pmatrix} M_p P^{-1} \\ M^{-1} - M^{-1} P^T M_p P^{-1} \end{pmatrix}^T A_{ss}^T f_5 = H^T f_5. \tag{22}$$

It was mentioned earlier that the total force and moment shall be conserved by the inclusion of the linear polynomials in the interpolation scheme. Now we can prove it mathematically. Multiply the vector of structural forces to the linear polynomial matrix $P$, and we have

$$P f_5 = P \begin{pmatrix} M_p P^{-1} \\ M^{-1} - M^{-1} P^T M_p P^{-1} \end{pmatrix}^T A_{ss}^T f_5 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} A_{ss}^T f_5 = Q f_5; \tag{23}$$

which has the matrix form of

$$\begin{pmatrix} x_1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ x_N & \cdots & 1 \\ y_1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ y_N & \cdots & 1 \\ z_1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ z_N & \cdots & 1 \end{pmatrix} \begin{pmatrix} f_{s1} \\ \vdots \\ f_{sN} \end{pmatrix} = \begin{pmatrix} x_1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ x_M & \cdots & 1 \\ y_1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ y_M & \cdots & 1 \\ z_1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ z_M \end{pmatrix} \begin{pmatrix} f_{s1} \\ \vdots \\ f_{sM} \end{pmatrix}.$$

It can be observed that the first row of the Eq. (22) represents the conservation of total force, and the last three rows show the conservations of moment in three directions respectively. In fact, there are also physical meanings related to the pseudo structural forces. Similarly, the solutions for the pseudo structural forces can be written as

$$f'_i = \begin{pmatrix} -M_p \\ M_p P^{-1} \end{pmatrix} A_{ss}^T f_5; \tag{24}$$

multiplied by $P^T$, then we have

$$P^T f'_i = P^T \begin{pmatrix} -M_p \\ M_p P^{-1} \end{pmatrix} A_{ss}^T f_5 = \begin{pmatrix} -M P^{-1} \\ 1 \end{pmatrix} \begin{pmatrix} Q \\ K \end{pmatrix} f_5;$$

thus

$$P^T f'_i + M f_5 = K f_5. \tag{25}$$

Suppose we compute the structural forces, denoted as $f'_i$, in a different interpolation with the linear polynomials removed. Because the total force will not be conserved by the new interpolation, the imbalanced forces can be defined as $\Delta f_i = f_i - f'_i$. Combining Eq. (22) and Eq. (24), we have

$$\Delta f_i = -M^{-1} P^T f'_i.$$
4. Volume mesh deformation

The deformation of fluid–structure interface, due to the deflection of the structure, usually requires the updates of the volume mesh so that the grid quality (i.e. continuity, smoothness, and orthogonality, etc.) can be preserved. In the same way used by the FSI, a mapping relation can be established between the displacements at surface boundary nodes and displacements at volume interior nodes, so that the motions of volume mesh can be interpolated based on the deflection at the boundary. Similar to Eq. (9), the interpolant can be formed given the movements on the solid surface are already known (where subscript $a$ denotes the points at the aerodynamic surface).

$$\Delta x_a = \begin{pmatrix} \Delta x_{a1} \\ \vdots \\ \Delta x_{aN} \end{pmatrix} = \begin{pmatrix} \phi_{a1a1} & \cdots & \phi_{a1aM} \\ \vdots & \ddots & \vdots \\ \phi_{aNMa1} & \cdots & \phi_{aNMaM} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{pmatrix} = C_{aa} \alpha. \quad (26)$$

It should be noted that the inclusion of the polynomial terms in the interpolant is not required, since the conservation of total force and moment is not necessary for the volume mesh deformations. Moreover, the polynomial terms will cause the mesh to translate or rotate, which is undesirable when a fixed domain boundary is often compulsory for the implementation of boundary conditions. The displacements of volume mesh can be expressed as follows (where subscript $v$ denotes the interior points in the volume mesh).

$$\Delta x_v = \begin{pmatrix} \Delta x_{v1} \\ \vdots \\ \Delta x_{vN} \end{pmatrix} = \begin{pmatrix} \phi_{v1a1} & \cdots & \phi_{v1aM} \\ \vdots & \ddots & \vdots \\ \phi_{vNa1} & \cdots & \phi_{vNaM} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{pmatrix} = C_{va} \alpha; \quad (27)$$

thus

$$\Delta x_v = C_{va} \alpha = C_{va} C_{aa}^{-1} \Delta x_a. \quad (28)$$

In general, it is more efficient to solve the linear system of Eq. (26) to obtain the interpolation coefficients $\alpha$ first, and $\Delta x_v$ can be calculated by simple matrix multiplication subsequently.

The computational effort associated with the volume mesh updates scales with $N_{vp} \times N_{sp}$ ($N_{vp}$ refers to the total number of volume points, and $N_{sp}$ refers to the number of control points at the boundary). For industrial applications with large meshes, e.g. $10^7$ cells mesh with $10^6$ surface points, even the straightforward matrix multiplications are too expensive considering current computing capability. Rendall and Allen [19] improved the performance of the RBF method by the implementation of the ‘greedy’ algorithm to reduce the size of the control points at the boundary, which sacrifices the accuracy of the deformation at the surface with an acceptable error. However, in time-dependent simulations with large meshes, the volume mesh motions may still consume significant computing resources, since the evaluations usually contain tens of millions of multiplication and summation operations at each time step where the number of volume grid $N_{vp}$ is massive. Therefore, it is necessary to further reduce the size of volume mesh in the interpolations to improve the efficiency.

In the present paper, we develop the multi-layer sequential mesh motion (MSM) algorithm to improve the efficiency of the volume mesh deformations in large mesh applications. During a MSM process, the evaluation of volume mesh movements can be split into sequential sub-layers, each of which has its specific interpolation relation. The total computational effort equals to the summation of the local cost in each sub-layer where both the number of control points and volume points may be reduced considerably, and therefore the performance of the mesh deformations can be effectively improved. The principles of the MSM algorithm are described as follows,

1. **Constructing the interpolation functions**: the interpolations are based on C-functions [22], which are compactly supported radial basis functions in the form of

$$\Delta x = \sum_{i=1}^{N} \lambda_i \phi(\|x - x_i\|);$$

where

$$\phi(\|x - x_i\|) = \begin{cases} \phi\left(\frac{d_i}{R}\right) & d_i < R, \\ 0 & d_i \geq R. \end{cases}$$

2. **Sub-layer partition**: without loss of generality, the volume mesh can be divided into three sub-layers: near-wall layer (normally enclose the flow boundary layer) $Z1 = \{D < \delta\}$; intermediate layer (subject to further partitions) $Z2 = \{\delta < D \leq R\}$; exterior layer $Z3 = \{D > R\}$. $D$ refers to the distance between the volume point and the solid surface, $R$ represents the compactly supporting radius of the RBF, and $\delta$ is the thickness of the near-wall layer.

3. **Mesh deformation in the near-wall layer**: A reduced set of surface points will be selected as the control points (total number of points is $N_c$ with $N_c \ll N_{sp}$) using the ‘greedy’ algorithm with surface error tolerance $\varepsilon_1$. The volume
points in Z1 then can be moved by the interpolation based on the control points, and the maximum displacement \( \Delta \) at the layer boundary will be recorded (for the use of the mesh motion in the subsequent layer). Z1 encloses the viscous boundary mesh and is required to conform to the surface shape with good smoothness. Therefore, the surface interpolation error \( \varepsilon_1 \) must be sufficiently small compared to the minimum grid size at the wall boundary. The local position correction step may also be performed within this layer to remove the induced error completely at the interpolated surface [23].

(4) **Mesh motion in the intermediate layer:** The volume points in Z2 are evaluated by a different interpolation based on control points on the outer boundary of the previous layer Z1. In fact, the mesh distribution turns coarser as the distance to the solid wall increases, and is not necessary to conform to the aerodynamic shape. As a result, larger surface interpolation errors can be tolerated \( (\varepsilon_2 \gg \varepsilon_1) \), which means that significantly fewer control points \( (N_d \ll N_c) \) on the layer boundary will be selected for the interpolation. The continuity problem may exist at the Z1/Z2 interface, when cell penetration (or negative volume cell) happens due to the independent interpolations in each layer with large variance of interpolation errors. This problem can be solved by introducing a limiter to the surface error tolerance \( \varepsilon_2 \) according to the largest deformation in the previous layer, i.e. \( \varepsilon_2 \ll \gamma \Delta \), where \( \gamma \) is defined as the continuity coefficient. Numerical experiments show that \( \gamma = 0.001 \) is sufficient to preserve the grid quality at the layer interface.

(5) **The efficiency of MSM:** the volume points in Z3 do not actually move due to the definition of compactly supported RBF, therefore no interpolations are necessary in this layer. The total computational cost for the MSM algorithm is then proportional to \( N_m = N_c \times N_{vp1} + N_d \times N_{vp2} \), and can be considerably economic compared to \( N_c \times N_{vp} \), since \( N_{vp1} \ll N_{vp} \) and \( N_d \ll N_c \).

(6) **Local mesh deformation control:** The MSM algorithm has great advantages in deforming the mesh of multi-element geometry with local component deflections (e.g. the flap deflection in the high-lift wing system). Mesh deformations can be confined in the vicinity of the moving component surfaces, through the sub-layer partitions and additional fixed control points in Z2 to define the stationary boundary. In this way, MSM algorithm could effectively reduce the overall computational complexities, and more importantly, maintain the grid consistency (untouched) in the majority part of the domain which makes the deformed mesh more robust for the time-dependent simulations.

5. Applications

### 5.1. Mesh deformation for block rotations

The efficiency of the MSM algorithm is demonstrated by the mesh motion due to severe rotation of a block in a square domain. The mesh nodes on the block follow its movement, while the nodes on the outer boundary are fixed. The block has dimension of \( 2 \times 1 \) with geometric center located in the center of a square domain which has dimension \( 20 \times 20 \), depicted in Fig. 3. The volume mesh was generated in the hybrid manner with structured quadrilateral grids clustered at the wall and unstructured triangular grids filled in the rest of the domain. The number of grid points on the block surface is 146, while the total point count in the volume is 3515. The block was rotated 45° counterclockwise around the center of the block in a single step, so that large mesh deformation can be produced.

Three different RBF-based algorithms were performed to calculate the mesh deformations of block rotation, including the standard RBF interpolation (SR), RBF interpolation with data reduction using ‘greedy’ algorithms (RR), the MSM algorithm. The compactly supporting radius was fixed of value 10 for the reason of comparison between algorithms. In the MSM algorithm, the computational domain was divided into three sub-layers for the 2D block case, consisting of the near-wall layer \( Z1 = \{ D < 0.1 \} \), the intermediate layer \( Z2 = \{ 0.1 < D \leq 10 \} \), and the exterior layer \( Z3 = \{ D > 10 \} \), shown in Fig. 4.
Fig. 4. Sub-layer partitions in the MSM algorithm.

Fig. 5. Mesh comparison after deformation (RR in red; MSM in blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 5 compares the updated meshes using different mesh deformation algorithms. The deformed meshes coincide, between the RR and MSM algorithms, within the near-wall layer $Z1$, which means there would be no loss of mesh quality using the MSM algorithm in the near wall region where significant boundary flow may exist. In the intermediate layer $Z2$, mesh inconsistency exists due to the different interpolations between two algorithms. The criteria of Equiangle Skew $Q_{EAS}$ can be used to inspect the grid quality, which is defined as follows,

$$Q_{EAS} = \max \left[ \frac{\theta_{\text{max}} - \theta_e}{180 - \theta_e}, \frac{\theta_e - \theta_{\text{min}}}{\theta_e} \right];$$  \hspace{1cm} (29)

where $\theta_{\text{max}}$ is the largest cell angle, $\theta_{\text{min}}$ is the smallest cell angle, $\theta_e = 90^\circ$ for quadrilateral/hexahedral cell and $\theta_e = 60^\circ$ for triangular/tetrahedral cell. The values of $Q_{EAS}$ range from 0 to 1, and $Q_{EAS} = 0$ represents the orthogonal grid, while $Q_{EAS} = 1$ means the highly skewed grid. Fig. 6 compares the grid quality of the deformed meshes using different algorithms, and shows that the grid quality can be well preserved using the MSM algorithm.

The computational efficiency of each algorithm is demonstrated in Table 1. For the 2D test case, the use of RR algorithm reduced the number of control points on the aerodynamic surface from 146 to 100 and therefore improved the computing efficiency by a factor of 1.5 approximately, compared with the standard RBF-based algorithm. Further reductions of surface control points and volume points can be performed by the sub-layer partitions in the MSM algorithm, which leads to efficiency improvement by a factor of 4 compared with SR algorithm. In fact, the reduction factor will increase for larger meshes, since the control points required to represent surface shape is only geometry dependent.
Table 1
The computational cost of mesh deformation algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error tolerance (ε)</th>
<th>No. of control points (N_{sp})</th>
<th>No. of volume points (N_{vp})</th>
<th>Computational cost (N_{sp} × N_{vp})</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>0</td>
<td>146</td>
<td>3,515</td>
<td>513,190</td>
</tr>
<tr>
<td>RR</td>
<td>5 × 10^{-8}</td>
<td>100</td>
<td>3,515</td>
<td>351,500</td>
</tr>
<tr>
<td>MSM: Z1</td>
<td>0.004</td>
<td>928</td>
<td>21</td>
<td>137,320</td>
</tr>
<tr>
<td>MSM: Z2</td>
<td>0.2</td>
<td>21</td>
<td>2120</td>
<td>137,320</td>
</tr>
<tr>
<td>MSM: Z3</td>
<td>0</td>
<td>0</td>
<td>467</td>
<td>137,320</td>
</tr>
</tbody>
</table>

5.2. Mesh deformation for multi-element configurations

In this section, a more realistic configuration of three-element high-lift wing is investigated using the MSM algorithm. The test case started with the initial mesh given in Fig. 7, with 360 nodes on the flap surface and approximately 0.35 million points in the volume. Multi-block structured grids were generated for this case with clustered boundary layer mesh extruded from the solid surface. The volume mesh was deformed driven by small local component motions, i.e. the flap rotation of 2° in counterclockwise direction around the axis at (0.9, 0), as depicted in Fig. 8.

Both the RR algorithm and the MSM algorithm were performed to calculate the mesh deformations due to the flap rotation, with the fixed compactly supporting radius of 0.2. In the MSM algorithm, the computational domain was divided, based on the flap surface, into three sub-layers, consisting of the near-wall layer Z1 = {D < 0.004}, the intermediate layer Z2 = {0.004 < D ≤ 0.2}, and the exterior layer Z3 = {D > 0.2}, shown in Fig. 9.
Fig. 8. The illustration of the small flap motion (deflected position shown in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 9. Sub-layer partitions in the MSM algorithm, a) overall view; b) close up view.

Table 2
The computational cost of mesh deformation algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error tolerance ($\varepsilon$)</th>
<th>No. of control points ($N_{sp}$)</th>
<th>No. of volume points ($N_{vp}$)</th>
<th>Computational cost ($N_{sp} \times N_{vp}$)</th>
<th>CPU time$^a$ (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR</td>
<td>$5 \times 10^{-8}$</td>
<td>200</td>
<td>349,937</td>
<td>69,987,400</td>
<td>1.1232</td>
</tr>
<tr>
<td>MSM: $Z_1 = (D &lt; 0.004)$, $Z_2 = (0.004 &lt; D \leq 0.2)$, $Z_3 = (D &gt; 0.2)$, $\Delta \approx 10^{-2}$;</td>
<td>$5 \times 10^{-8}$</td>
<td>200</td>
<td>15,257</td>
<td>4,953,033</td>
<td>0.2964</td>
</tr>
<tr>
<td>Z1</td>
<td>$1 \times 10^{-5}$</td>
<td>31</td>
<td>61,343</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z2</td>
<td>0</td>
<td>0</td>
<td>273,337</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$^a$ All the CPU time listed in this paper is based on Intel(R) Xeon(R) CPU E5-2690 v3 2.60 GHz.

The computational efficiencies of two algorithms are demonstrated in Table 2. For the multi-element configuration, the mesh deformations can be confined in the vicinity of the moving flap, through the sub-layer partitions and additional fixed control points in $Z_2$ to define the stationary boundaries. Therefore, the computational cost using MSM can be reduced significantly by a factor of 14 compared to that of the RR algorithm. In addition to the memory size reduction, MSM algorithm can be almost 4 times faster than RR method to perform the mesh deformation for this more complicated geometry.

Fig. 10 compares the grid quality of the deformed meshes using different algorithms. The quality of structured meshes can be examined directly by the value of orthogonal quality, which is defined as the normalized dot product of the vector connecting two adjacent cell centers and the interface area vector. It is shown that the grid quality can be well preserved using the MSM algorithm.
5.3. **AGARD 445.6 wing flutter analysis**

The AGARD 445.6 wing is a standard model for the flutter boundary predictions with published measurement data, and therefore selected in this section to verify the present algorithm for aeroelastic simulations. This wing model has a quarter-chord weep angle of 45°, an aspect ratio of 1.65, a taper ratio of 0.66 and a NACA65A004 airfoil section. Structured quadrilateral mesh was generated for the CFD computation, shown in Fig. 11, with 4560 cells distributed on the wing surface and 42 million cells filled in the fluid domain. 242 structural nodes were created for the finite element computation in the structure domain.

Flow Mach numbers range from 0.499 to 1.072 were investigated in the flutter boundary analysis. The aerodynamic forces were calculated by solving the unsteady compressible Navier–Stokes equations, which were based on the finite volume discretizations with one-equation Spalart–Allmaras formula for turbulence modeling. The Green–Gauss integration was adopted to calculate the gradient, while the flux was computed using the Roe-FDS scheme. The governing equations of structure dynamics were solved in the modal approach, with the first four natural vibration modes provided by the measurement [24].

The modal equations of structure motions can be written as

\[
\frac{d^2}{dt^2}(q_i) + \omega_i^2(q_i) = \frac{\Phi_i^T f_c}{m_i}, \quad (i = 1 \ldots 4);
\]

where \(q_i\) is the generalized displacement of \(i\)-th mode, \(\Phi_i\) refers to the mode shape, \(\omega_i\) is the natural frequency of \(i\)-th mode, and \(m_i\) is the modal mass of \(i\)-th mode. The actual displacements of the structure can be recovered from the generalized displacements in the modal space,

\[
\mathbf{u}(t) = \Phi_i q_i(t).
\]

The FSI was performed in a loosely coupled manner with time step size of \(2 \times 10^{-4}\) second, which ensures that each time period covers at least 50 time steps for the highest mode. The simulations started with the stationary rigid body wing model. After the flow around the wing was fully developed, the rigid body wing was switched to the flexible wing. Given a small imposed perturbation on the first mode displacement, the wing was then allowed to deflect in response to the
aerodynamic loads. In Figs. 12–14, the computed time histories of the generalized displacements of the AGARD 445.6 wing at $M_{\infty} = 0.96$ are plotted for different flutter speed index $V^*$ ($V^* = U_{\infty}/(b\omega_\alpha \cdot \mu^{0.5})$, where $b$ is the semi-chord at root, $\omega_\alpha$ is the natural frequency of the first torsion mode, and $\mu$ refers to the mass ratio). For a given Mach number, several simulations with different $V^*$ were performed to determine the location of the flutter boundary. It should be noted that the free-stream Reynolds number and static temperature would be changed accordingly since $U_{\infty}$ was altered. However, the effect on the final solutions due to the small variations in the Re and temperature was negligible according to the numerical experiments. When $V^*$ is smaller than the critical value on the flutter boundary, the amplitudes of all modes are in a decreasing trend corresponding to the damped response as shown in Fig. 12. When the value of $V^*$ is close to the critical value, the balanced response can be observed in Fig. 13. When $V^*$ exceeds the flutter boundary, the amplitudes of all modes are in an increasing trend corresponding to the diverging response as shown in Fig. 14. At $M_{\infty} = 0.96$, the predicted value of the flutter boundary is 0.2887 which is consistent with the measurement result of 0.3076 provided by [24].

Figs. 15–17 depict the wing surface pressure contours with three different $V^*$, in which differences of pressure distributions can be observed due to the variations of free-stream dynamic pressures. The comparison of predicted flutter boundary and the measurement data for AGARD 445.6 wing is illustrated in Fig. 18. In general, the computed results are in good agreement with the experimental data for subsonic flows. However, the flutter boundary is over predicted when the flow exceeds the sound speed. This discrepancy was also noted by other researchers in the previous studies [25], which suggested that it may be caused by the inadequacy of the turbulence modeling for strong shock/wave boundary interaction or may be due to the inaccurate measurement record.
Fig. 14. Time histories of the generalized displacements for $M_\infty = 0.96$, $V^* = 0.2987$.

Fig. 15. Wing surface pressure distributions for $M_\infty = 0.96$, $V^* = 0.2788$.

Fig. 16. Wing surface pressure distributions for $M_\infty = 0.96$, $V^* = 0.2887$. 
Fig. 17. Wing surface pressure distributions for $M_\infty = 0.96$, $V^* = 0.2987$.

Fig. 18. Flutter boundary comparisons for AGARD 445.6 wing in a) flutter speed; b) flutter frequency.

Fig. 19. Mesh distribution on the aerodynamic surface.

5.4. Static aeroelastic simulation with the high-aspect-ratio aircraft

In this section, the static deflection of the high-aspect-ratio aircraft due to fluid–structure interactions was simulated using the developed coupling scheme. It is demonstrated that current scheme is applicable to three-dimensional large meshes with industrial level of complexity. The CFD mesh was generated in hybrid type with clustered boundary layer grids. Approximately 0.5 million cells are distributed on the aerodynamic surface, depicted in Fig. 19, while the total cell count in the computational domain is 15 million.
The initial flow condition was provided with the free-stream Mach number of 0.5, 0° angle of attack and the altitude of 15 km. Three-dimensional compressible Navier–Stokes equations based on the finite volume discretizations were solved in the fluid domain for the steady flow, with the one-equation Spalart–Allmaras formula for turbulence modeling. The Green–Gauss integration was adopted to calculate the gradient, while the flux was computed using the Roe-FDS scheme. In current case, the structural movement was solved using the flexibility method, with the stiff matrix $K$ provided by the finite element calculations. Neglecting the damping term, the structural displacement $u$ can be given as

$$u = K^{-1}f_s;$$  \hspace{1cm} (32)

where $f_s$ refers to the acting forces on the structural nodes after the FSI. The finite element nodes are distributed over the whole aircraft body with total point number of 1077, shown in Fig. 20.

The static aeroelastic simulation started from the CFD computation based on the original aerodynamic shape. Then, the structural movements were determined by the flexibility method using the computed aerodynamic loads, and consequently corrected the surface shape for the next round CFD computation. This process would be iterated until the status of static equilibrium was achieved when the both aerodynamic forces and structural movements converged. As mentioned in section 3, the calculation of the structural forces during FSI can be extremely time-consuming using the traditional coupling matrix method, when the structural system has very large degree of freedom. The acting forces on the structural nodes can be calculated more efficiently via the proposed FSI scheme with pseudo structural forces. The efficiency improvement is significant for the three-dimensional case, as shown in Table 3, that almost 100 times CPU time reduction can be achieved.

The computational costs to calculate the volume mesh deformation are demonstrated in Table 4, with both RR and MSM algorithms listed. For the three-dimensional complicated case, the MSM algorithm can effectively reduce the problem size by approximately 40%, and reduce the CPU time by 35%.

Fig. 21 shows the convergence of the largest displacement over the aircraft surface after several iterations. $\Delta S_{\text{max}}$ represents the maximum displacement (at the wing tip in this case), and b refers to the semi-wingspan. After the first iteration, the wing tip moved a distance of approximately 5.2% of semi-wingspan, which is over deflected compared to the static equilibrium state. The deformation of the aircraft surface converged after 6 iterations, which has the largest displacement of 4.2% with respect to the semi-wingspan. It can be shown in Fig. 22 that significant deflections occur to the wing surface while the deformations on the vertical tail and fuselage are relatively small.

The computed surface pressure distributions at various cross sections are shown in Fig. 23. The location of $y/b = 0.3$ is near the wing root and $y/b = 0.9$ is close to the wing tip. The surface pressures near the root show only small differences
between the original and deformed wing, which is expected since the structural deflections at this location are very small. The discrepancy of surface pressures becomes more significant at the wing tip where larger structural deflections occur. This means that the aerodynamic performance will be affected by the static aeroelastic deformations for the high-aspect-ratio aircraft, which must be considered in advance at the design stage.

6. Conclusions

In the present paper, a time efficient coupling scheme is developed based on the radial basis function interpolations. Two key issues related to the partitioned approach of aeroelasticity are discussed, with novel algorithms proposed to improve
the computational efficiency. During the FSI process, the positive definite system of linear equations is constructed with the introduction of pseudo structural forces. The acting forces on the structural nodes can be calculated more efficiently via the solution of the linear system, avoiding the costly computations of the aerodynamic/structural coupling matrix. For mesh motions, the multi-layer sequential mesh motion (MSM) algorithm is proposed to improve the efficiency of the volume mesh deformations, which will be adequate for large-scale time dependent applications with frequent mesh updates. The 2D mesh deformation cases demonstrate that the MSM algorithm is more computational efficient with grid quality well preserved, compared to the standard-RBF based method. The developed coupling scheme is validated by the AGARD 445.6 wing flutter boundary computations, and the predicted results are in good agreement with the measurement data. The static aeroelastic simulations with the high-aspect-ratio aircraft show that current scheme is applicable to three-dimensional large meshes with industrial level of complexity.

References