AIAA-2003-3514

Application of Spline Matrix for Mesh Deformation with Dynamic Multi-Block Grids

K.L. Lai[¶], H.M. Tsai[‡]

Temasek Laboratories, National University of Singapore
10 Kent Ridge Crescent, SINGAPORE 119260

F. Liu [§]

Department of Mechanical and Aerospace Engineering
University of California, Irvine, CA 92697-3975

ABSTRACT

The paper presents a three-dimensional mesh deformation algorithm for dynamic multiple-block moving mesh configurations. The flow domain is modelled as an elastic solid body where the Boundary Element Method (BEM) is applied to formulate a spline matrix that transforms the displacement vectors at a solid boundary to the interior of the field grid. Using a similar approach, a spline matrix for the interaction between the fluid grid and the structure grid of the flexible body can be generated. The BEM-based approach provides an unified treatment for both the flow mesh and the flexible body. For efficient implementation of deforming mesh, the BEM-based algorithm is augmented with a conventional grid deformation method based on transfinite interpolation (TFI). The BEM-based interpolation determines the boundary deformations of each block of the multiple-block flow mesh, while an arclength based TFI deforms the grid within each flow-block.

1. INTRODUCTION

In calculations of the unsteady flow over flexible structures in problems such as flutter analysis, mesh deformation plays a vital part and has direct bearing on the overall accuracy and efficiency of the numerical scheme. As the fluid-structure system evolves, the boundary between the fluid and structure may undergo deformations. A deforming mesh algorithm is needed to update the flow mesh in response to the boundary deformation at each time step.

In the solution process where a coupled fluid and structure model is involved, deforming the flow mesh is effected in two steps. The first step is determination of the boundary deformation for a specified deformation of the structural model; this is accomplished by interpolation between the structural grids and the boundary grids. In the second step, the boundary perturbations are propagated into the flow domain to update the field mesh.

1.1 BEM-Based Interpolation

In general, the Computational Fluid Dynamics (CFD) model and the Computational Structural Dynamics (CSD) model are developed independently and will not have common grid points. Interpolation or splining methods are required to connect the displacements and forces between the structural and aerodynamic grids. A CFD-CSD interfacing algorithm, initially proposed by Chen and Jadic[2] based on the Boundary Element Method for linear elasticity, has been developed in the present work. A spline matrix [***S***] is generated to relate the displacements [***u***] and forces [***f***] between the aerodynamic grid (denoted with subscript *a*) and the structural grid (subscript *s*) as

$$[u_a]=[S][u_s] \tag{1}$$

and

$$[f_s]=[S]^T[f_a] \quad . \tag{2}$$

[¶] Research Scientist, Temasek Laboratories, National University of Singapore. Member AIAA.

[‡] Principal Research Scientist, Temasek Laboratories, National University of Singapore, Member AIAA.

[§] Associate Professor, Department of Mechanical and Aerospace Engineering, University of California, Irvine. Senior Member AIAA.

The BEM-based interpolation method is extended for the purpose of deforming grids. By considering the field grid as a three-dimensional elastic body with a void where the flexible structure is, a spline matrix [B_f] for the field mesh is obtained to relate the deformations of the field mesh [$\boldsymbol{u_f}$] to the boundary displacements [$\boldsymbol{u_a}$] as

$$[u_f]=[B_f][u_a] \quad . \qquad \textbf{(3)}$$

Treating the field grid as an elastic solid body avoids the potential problem of grid crossing as might happen in spring-analogy method. Inherent in the BEM-based method is that the motions of the grid points are not determined using their neighbourhood points.

The BEM-based approach is applicable to both mesh systems with or without overlapping grids. An example of the former system is the overset grids used by the CFD codes OVERFLOW. For mesh systems without overlapping grid, i.e. each flow block shares common boundaries with adjoining flow-blocks, the BEM model (a multiple block BEM model in this case) of the field mesh coincides with the flow-blocks. Consequently, the flow-block deformation is inherent with the BEM model. For mesh systems with overlapping grids, the BEM model of the field mesh is formed, though not as straightforward as non-overlapping grids, independent of the flow-blocks. The motion of each flow-block is related to the boundary deformation by the spline matrix [B_f], using equation (3). The present method allows overlapping grids to evolve dynamically in an automatic manner and yet maintain the quality of the original grid. It is a significant extension for the use of overset methods in design optimisations and other problems involving moving aerodynamic surfaces.

Conceptually, provided with the sufficient computing resources that required, we may form a spline matrix [B_f] that gives the displacement vectors at all field nodes and thus defines the moving grid in response to the specified boundary deformation [$\boldsymbol{u_a}$]. In real applications, a reasonably discretised field mesh may have degrees of freedom in the order of 10^6. When such a large number of grid points are involved, however, the sole BEM-based approach becomes prohibitive since a huge matrix is required. The BEM-based approach is enhanced by incorporating into it a conventional mesh deformation technique such as Trans-Finite Interpolation (TFI). The spline matrix is applied to a finite number of control points, which are taken as inputs to the TFI procedure that follows. These control points are selected on the boundary of the flow blocks. Based on the deformations at the control points, the TFI method is invoked to deform the grid within each of the flow blocks.

1.2 Mesh Deformation Methods

On the subject of mesh deformation, Batina[3] proposed a spring-analogy method for dynamic grids. The method can adapt to large surface deformations and is applicable to both structured and unstructured grids, as exemplified in the works of Lee-Rausch and Batina[4, 5], and Prananta and Hounjet[6]. The spring-analogy uses an iterative scheme resembling an elliptic grid generator and is inherently time consuming if a large number of grid points are involved. Byun *et al.*[7] and Ji and Liu[8] used simple algebraic methods to form dynamic grids in their flutter simulations. The algebraic algorithm deforms an existing grid by redistributing the nodal points along grid lines that are normal to the structural surface. This method is fast but cannot adapt to large spatial displacements where grids of poor quality or crossover of grid lines may result. Maintaining grid quality such as smoothness and direction normal to the structural surfaces is desirable but not readily possible with simple algebraic grid generation method.

Jones and Samareh-Abolhassani[9], and Reuther *et al.*[10] used a method based on variants of the Trans-Finite-Interpolation (TFI) to form structured grids. TFI is an algebraic algorithm, capable of automatically handling multiple moving surfaces and taking into account of the original interior grid distribution in moving the grid points. TFI methods can be incorporated for multi-block grids; in this case, the grid motion of the block surfaces must be determined first. Tsai *et al.*[11] incorporated the salient features of the spring analogy method by Batina[3] to first determine block corner of each of the blocks before applying a TFI based method for block faces and volume grid of each individual block.

Certain criteria have been noted in the design of dynamic moving grids. It is desirable that the moving grid generator be flexible in handling moving surfaces of arbitrary configuration, capable of handling multiple blocks of structured grids in either a structured or an unstructured arrangement. For numerical stability, the moving grid should retain as best as possible the essential qualities of the initial grid that it evolves from. To facilitate computation on parallel computers, the grid is to be naturally parallelizable in a multi-block parallel flow solver. With these factors borne in mind, we present here a dynamic grid generator for complex geometries, which makes use of a multi-block method for grid representation. A previous attempt to address this need was reported by Tsai *et al.*[11] but it applies only to multi-block grids that have matching block faces.

2. MATHEMATICAL BACKGROUND

The BEM-based approach used here follows the principles proposed by Chen and Jadic[2]. It is based on the formulations for solid mechanics analysis. The interpolation scheme is established using the condition of minimum strain energy. For a body that is modelled with multiple blocks, such as the multiple-block moving grid and the component-based approach of modelling large complex body, the multiple-block BEM solver is used.

Emphasis in this section is given to the underlying principles of BEM and the development of the interpolation scheme using the BEM approach. We will also discuss the issues of corner and edge treatment in the boundary element method and the assembly method for multiple block BEM.

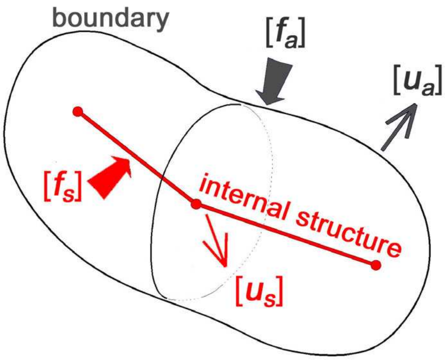


Figure 1 Schematic aeroelastic system where a flexible body contains an internal loading carrying structure.

2.1 Boundary Element Methods

Interfacing between the structural grid and the aerodynamic grid is treated here on the basis of elastostatic problem. Consider an arbitrary three-dimensional homogeneous body Ω whose boundary is defined by a closed surface Γ, schematic shown in Figure 1. The CFD surface grid is defined on Γ and the CSD (internal) grid within Ω. To arrive at the desired interfacing algorithm, we make use of the BEM formulation to relate the displacements and forces vectors between Ω and Γ. The BEM formulations for elastostatic problem (see, for example, Brebbia and Dominguez[1]) are based on the differential equations of displacement. The Betti's reciprocal work theorem and the Somigliana identity for the displacements are used to derive an integral equation for the displacements at an interior point *p* due to tractions *t(Q)* and displacements *u(Q)* at a point *Q* on the boundary Γ. In numerical implementation, the integral equations give a set of linear algebraic equations which, in matrix form, is

$$[u_i]+[H_{bi}][u_b]=[G_{bi}][t_b] \quad . \qquad \textbf{(4)}$$

The subscripts *b* and *i* signify boundary and interior values respectively. By moving the load point *p* to the boundary and taking care of the singularities that arise when *p* coincides with *Q*, we establish a relationship for the boundary variables only. The boundary integral equations with boundary variables only take the form

$$[H_{bb}][u_b]=[G_{bb}][t_b] \quad . \qquad \textbf{(5)}$$

The subscript *bi* refers to boundary-to-interior influence where the subscript *bb* indicates boundary-to-boundary influence. The matrices [*H*] and [*G*], termed as the displacement matrix and the traction matrix, respectively, contain the integrals of the kernels T_{ij} and $U_{ij,}$ respectively.

Note that equation (4) relating the displacements at internal points to the variables at the boundary is meaningful only for points that are within the volume of Ω. Therefore, those coefficients in [H_{bi}] and [G_{bi}] which are associated with external points are nullified.

For three-dimensional problems, each element of [*H*] and [*G*] is a 3×3 sub-matrix. From equation (5), the stiffness matrix [*K*] that relates the boundary traction to boundary displacement as [t_b] = [*K*] [u_b] is defined as

$$[K]=[G_{bb}]^{-1}[H_{bb}] \quad . \qquad \textbf{(6)}$$

Substituting equation (6) and (5) into (4), we can relate the displacements at internal points to those of the boundary grids in the form of

$$[u_i]=[B][u_b] \ , \qquad \textbf{(7)}$$

in which, the influence matrix [*B*] is given by

$$[B]=[G_{bi}]^{-1}[K]-[H_{bi}] \quad . \qquad \textbf{(8)}$$

Equations (6) and (8) are the basic equations for establishing the spline matrix between boundary grids and internal grids.

2.2 Constructing the Spline Matrix

The aim of this section is to develop the spline matrix that relates the boundary displacement to the given internal (structural) displacement.

Substituting equation (1) into (8), one may notice that the spline matrix [*S*] may be interpreted as the 'inverse' of [*B*] and that computing [*S*] may be regarded an inverse BEM problem. To this connection, some considerations of the characteristics of matrix [*B*] are in order.

In most practical applications, there is great disparity between the number of internal grids and the number of boundary grids, resulting in the formation of a non-square and therefore non-invertible matrix [*B*]. Furthermore, by considering rigid body motion, for each row of [*B*], the sum of all coefficients must equal the identity matrix. However, exact satisfaction of the rigid-body-motion condition leads to a singular matrix, [*B*]. Therefore, the [*B*] matrix is not invertible even if the number of points defined in the interior is the same as that of the boundary grids, i.e. [*B*] being a square matrix.

In all cases, inversion of matrix [*B*] is not possible. Clearly, the spline matrix [*S*] cannot be obtained by direct operation on [*B*] alone. An additional condition is required to define [*S*].

The minimum strain energy requirement proposed by Chen and Jadic[2] is used in the present work as the additional condition to determine the spline matrix. For an elastic body acted upon by external forces only, the strain energy stored in the deformed body equals the work done by all external forces. A strain energy function *U* is defined in terms of traction and displacement as

$$U=\int_{\Gamma} \boldsymbol{u}\cdot\boldsymbol{t}\, d\Gamma \ . \qquad \textbf{(9)}$$

Expressed in discretized form for numerical implementation, equation (9) takes the form

$$U=\sum_{m=1}^{M} \int_{-1}^{+1}\int_{-1}^{+1}\left(\sum_{c=1}^{n} N_c(\xi_1,\xi_2)\boldsymbol{u}_{\boldsymbol{c}}^{\boldsymbol{e}}\right) \cdot\left(\sum_{c=1}^{n} N_c(\xi_1,\xi_2)\boldsymbol{t}_{\boldsymbol{c}}^{\boldsymbol{e}}\right) J(\xi_1,\xi_2)d\xi_1 d\xi_2 \qquad \textbf{(10)}$$

where u^{e} and t^{e} denote, respectively, the displacements and tractions at the nodal points of a boundary element, and, N_c is the interpolation (shape) function and *J* the Jacobian of transformation between global and local coordinates. Rewriting (10) in matrix form, we have

$$U=[u_b]^T[R][u_b] \qquad \textbf{(11)}$$

where [*R*] = [*N*][*K*] and [*N*] contains the integrations of the interpolation function $N(\xi_1, \xi_2)$ indicated in (10). Since the strain energy due to rigid body motion must be zero, it follows that the sum of all coefficients in each row of [*R*] must equal zero. This implies singularity of the matrix [*R*].

For a given set of displacement vectors $[u_i]$ at internal points, the strain energy function *U* is to be minimized so as to avoid undue deformation of the structure. This is a constrained quadratic minimization problem that can be solved using the Lagrange multiplier technique. The Lagrangian $\mathscr{L}$ of the problem is given by

$$\mathscr{L}=[u_b]^T[R][u_b]-\lambda^T([B][u_b]-[u_i]) \qquad \textbf{(12)}$$

where λ is a vector containing the Lagrange multipliers. Substituting equation (7) into (12) for $[u_i]$ and differentiating the resulting equation with respect to u_b and λ separately, we have two differential equations:

$$\begin{aligned} \frac{\partial\mathscr{L}}{\partial u_b} &= [R][u_b]+[R]^T[u_b]-[B]^T\lambda \\ \frac{\partial\mathscr{L}}{\partial\lambda} &= [u_i]-[B][u_b] \end{aligned} \qquad \textbf{(13)}$$

Letting the derivatives in equation (13) be zero, and rearranging, we arrive at an equation that relates $[u_b]$ and $[u_i]$. The spline matrix [*S*] is obtained as

$$[S]=[R+R^T]^{-1}B^T[B\,[R+R^T]^{-1}B^T]^{-1} \ . \qquad \textbf{(14)}$$

A sufficient condition for equation (14) to be solvable is that both matrices [*R*] and [*B*] have full rank. The solution process for a spline matrix therefore requires the singularity in these matrices to be eliminated. This is accomplished by separating the displacements into rigid body motion and elastic deformation, resulting in the decomposition of [*R*] and [*B*] matrices. Based on the decomposed matrices [*R′*] and [*B′*] for elastic motion, which are non-singular, a spline matrix [*S′*] for elastic deformation is obtained. The desired spline matrix [*S*] for the free-free body is thus derived by recovering the rigid body motion from [*S′*]. The correctness of [*S*] may be verified by multiplying [*S*] with [*B*], the product should approximate the identity matrix.

2.3 Treatment of Corners and Edges

As a prelude to the multiple block BEM presented in next section, this section presents the solution procedures in the boundary element method for modelling geometries with corners where multiple traction values occur at a node. The treatment of corners is of significant relevance to multi-block BEM where the interfaces between blocks always involve corners or edges.

The boundary integral equation (5) generates a system of equations for solving one traction vector and one displacement vector at a node. When corners are present, tractions at corner points may take up multiple values. As a result, the system of equations given by (5) will have more unknowns than equations. Thus, auxiliary equations are needed to close the system of equations. Consider, for instance, a three-dimensional corner where three surfaces meet, Figure 2, a total of twelve unknown quantities exist at the corner node: nine traction components and three displacement components. By using equation (5) six linear equations can be established. Therefore six auxiliary equations are needed to close the equation set. Similarly, three auxiliary equations are needed for three-dimensional edges where two surfaces meet.

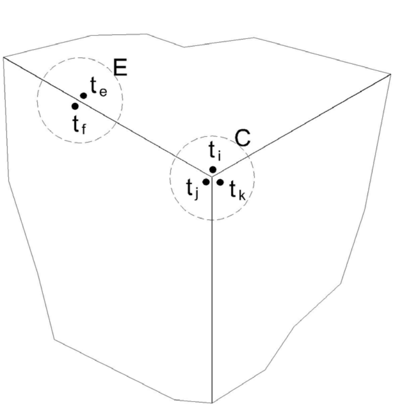


Figure 2. 3D corner node and edge node, labelled C and E, where tractions acquire discontinuous values. Nodal positions are deliberately offset from the corner point for clarity.

A number of schemes for obtaining the auxiliary equations for the treatment of corners and edges have been proposed for the boundary element method, for example, Chaudonneret [12], Rudolphi [13] and Gao and Davies [14]. The scheme used in the present study is based on the differential equations of equilibrium proposed by Gao and Davies[14]. When the equilibrium equation is applied to boundary stresses, it implies that the shear stress approaches a constant value near the corner. On the boundary Γ the differential equation of equilibrium can be written as

$$\frac{\partial t^i_{\zeta_1}}{\partial \zeta_1} = 0 \; ; \quad \frac{\partial t^i_{\zeta_2}}{\partial \zeta_2} = 0 \quad for \; i = 1,2,3. \tag{15}$$

To obtain the derivatives in equation (15), a local orthogonal coordinate system (ζ_1, ζ_2) in the local tangential directions at the corner is defined for each boundary element. The components of the traction on the surface of the boundary element i are differentiated with respect to ζ_1 and ζ_2, yielding two linear equations for each boundary element. Equation (15) can be rewritten in terms of the local intrinsic coordinates (ξ_1, ξ_2), the interpolation function N_c (ξ_1, ξ_2) and the global traction of a boundary element. Application of equation (15) to the three boundary elements connecting to the corner yields six linear equations. There are now the same number of equations as the number of unknown quantities. For three-dimensional edge where three auxiliary equations are needed, equation (15) provides four auxiliary equations. In this case, the redundant equation may be discarded.

In general, the auxiliary equations given by (15) are linearly independent because they are derived from the geometry of different boundary elements. The only linkage between adjoining boundary elements is the common nodes they share. In the cases where neighbouring elements have corners along one common axis (for example, one edge of a prism), the auxiliary equations given by (15) for the nodes involved form a set of linearly dependent equations, of degrees of freedom $(n-1)$ where n is the number of nodes involved. In this situation, another additional equation is needed to close the system of equation.

Another auxiliary equation can be generated on the assumption that the stress tensor is unique at a point. Making use the relationship between tractions t and stresses σ (in tensor notations as $t_i = \sigma_{ij}\, n_j$), we obtain

$$n^p \cdot t^q = n^q \cdot t^p \; , \tag{16}$$

where p and q are the indices of the boundary elements connected to the corner, and n is the local normal vector at the corner node. Equations (15) and (16) may be used in conjunction with each other to form a set of auxiliary equations. Writing the auxiliary equations in matrix form, we have

$$[a] = [A][t] \; , \tag{17}$$

where $[A]$ denotes the auxiliary matrix and $[a]$ the auxiliary array which generally contains zero's .

When corners are present, the number of displacement variables is greater than the number of traction variables, resulting in $[H_{bb}]$ and $[G_{bb}]$ being non-square matrices. Incorporating the boundary integral equation (5) and the auxiliary equation (17), and partitioning the traction vectors into $[t]$ and $[t_a]$ where the latter accounts for the additional traction variables at corners and edges, we have the system of linear equations in the form

$$\begin{bmatrix} H_{bb} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ a \end{bmatrix} = \begin{bmatrix} G_{bb} \\ A \end{bmatrix} \begin{bmatrix} t \\ t_a \end{bmatrix} \; , \tag{18}$$

where $[I]$ is an Identity matrix. Now that the system matrices in equation (18) are square. The traction matrix can be inverted. The stiffness matrix $[K]$ is obtained by eliminating the terms at the degrees of freedom associated with $[a]$. That is, $[K]$ is the left-hand side partition $[K_l]$ of the product of the following multiplication

$$[K_l | K_r] = \begin{bmatrix} G_{bb} \\ A \end{bmatrix}^{-1} \begin{bmatrix} H_{bb} & 0 \\ 0 & I \end{bmatrix} \tag{19}$$

For the internal grids, the matrices $[H_{bi}]$ and $[G_{bi}]$ are evaluated using equation (5). With the knowledge of matrices $[K]$, $[H_{bi}]$ and $[G_{bi}]$, the influence matrix $[B]$ is hence determined.

2.4 Multiple Block BEM

The motivation of developing a multiple block BEM solver in the present work is driven primarily by the need to handle very large and complex structures, and to reduce run time and disk space requirements.

Examining the solution process for the spline matrix, one can note that it requires computations of matrix inverses. These numerical operations require considerable amounts of computing resources when large matrices are involved. For very large and complex structures, the computing resources needed may become prohibitive to allow the structure to be treated in a single BEM model. To avoid this problem, the complex structure is divided into substructures (blocks) and the BEM model of each block is considerably smaller and more manageable than the model of the entire structure treated as a single block. Using the condition of continuity and the condition of equilibrium, the regional matrices are incorporated into the global system matrices. The interactions between regions are thus reflected in the global system matrices.

There are different methods of assembling the system equations in a multiple block BEM. The method used here is based on the sub-matrix approach. The aim is to arrive at a matrix equation of the form $[H][u] = [G]\,[t]$, in which the global traction matrix $[G]$ is a square matrix while the global displacement matrix $[H]$ is a general rectangular matrix.

Consider the i-*th* region in a system consisting of *m* regions, its BEM model is treated separately to obtain the matrices of equation (6), or equation (18) when corner treatment is implemented. The nodes for each region are divided into two sets: *external nodes* $[\boldsymbol{u}_x]$ that lie on the external (wetted) face, and *interface nodes* $[\boldsymbol{u}_c]$ that lie on the interface between blocks. Note that a subset of $[\boldsymbol{u}_x]$ lies on the intersection between the external face and the block-to-block interface. According to this classification, the traction and displacement matrices can be partitioned into sub-matrices that are associated with the external grids and the interface grids of the i-*th* region:

$$\begin{bmatrix} H_{xx}^{i} & H_{xc}^{i} \\ H_{cx}^{i} & H_{cc}^{i} \end{bmatrix} \begin{bmatrix} u_x^{i} \\ u_c^{i} \end{bmatrix} = \begin{bmatrix} G_{xx}^{i} & G_{xc}^{i} \\ G_{cx}^{i} & G_{cc}^{i} \end{bmatrix} \begin{bmatrix} t_x^{i} \\ t_c^{i} \end{bmatrix} \quad . \qquad \textbf{(20)}$$

The suffices *x* refers to the external grids while *c* corresponds to the interface grids. From (20) we obtain two sets of equations for the i-*th* block:

$$[H_{xx}^{i}][u_x^{i}] + [H_{xc}^{i}][u_c^{i}] = [G_{xx}^{i}][t_x^{i}] + [G_{xc}^{i}][t_c^{i}] \quad , \qquad \textbf{(21)}$$

and

$$[H_{cx}^{i}][u_x^{i}] + [H_{cc}^{i}][u_c^{i}] = [G_{cx}^{i}][t_x^{i}] + [G_{cc}^{i}][t_c^{i}] \quad . \qquad \textbf{(22)}$$

The tractions $[\boldsymbol{t}_c]$ at interface nodes can be expressed as

$$[t_c^{i}] = [\tilde{H}_{cx}^{i}][u_x^{i}] + [\tilde{H}_{cc}^{i}][u_c^{i}] - [\tilde{G}_{cx}^{i}][t_x^{i}] \quad , \qquad \textbf{(23)}$$

where

$$\begin{aligned} [\tilde{H}_{cx}] &= [G_{cc}]^{-1}[H_{cx}] \\ [\tilde{H}_{cc}^{i}] &= [G_{cc}]^{-1}[H_{cc}^{i}] \\ [\tilde{G}_{cx}] &= [G_{cc}]^{-1}[G_{cx}] \end{aligned} \qquad \textbf{(24)}$$

Substituting (23) into (21), we have

$$[\hat{H}_{xx}^{i}][u_x^{i}] + [\hat{H}_{xc}^{i}][u_c^{i}] = [\hat{G}_{xx}^{i}][t_x^{i}] \quad . \qquad \textbf{(25)}$$

where

$$\begin{aligned} [\grave{H}_{xx}^{i}] &= [H_{xx}^{i}] - [G_{xc}^{i}][G_{cc}^{i}]^{-1}[H_{cx}^{i}] \\ [\grave{H}_{xc}^{i}] &= [H_{xc}^{i}] - [G_{xc}^{i}][G_{cc}^{i}]^{-1}[H_{cc}^{i}] \\ [\grave{G}_{xx}^{i}] &= [G_{xx}^{i}] - [G_{xc}^{i}][G_{cc}^{i}]^{-1}[G_{cx}^{i}] \quad . \end{aligned} \qquad \textbf{(26)}$$

When treatment of corner and edge is implemented, the auxiliary equations (11) are also partitioned as

$$[A_x^{i}][t_x^{i}] + [A_c^{i}][t_c^{i}] = 0 \quad ; \qquad \textbf{(27)}$$

in which, we have made use of the fact that the elements of array $[a]$ are generally equal to zero. Using the expression of $[\boldsymbol{t}_c]$ in (23), equation (27) can be rewritten as

$$[\tilde{A}_x^{i}][t_x^{i}] + [\tilde{B}_x^{i}][u_x^{i}] + [\tilde{B}_c^{i}][u_c^{i}] = 0 \qquad \textbf{(28)}$$

where

$$\begin{aligned} [\tilde{A}_x^{i}] &= [A_x^{i}] - [A_c^{i}][\tilde{G}_{cx}^{i}] \\ [\tilde{B}_x^{i}] &= [A_c^{i}][\tilde{H}_{cx}^{i}] \\ [\tilde{B}_c^{i}] &= [A_c^{i}][\tilde{H}_{cc}^{i}] \quad . \end{aligned} \qquad \textbf{(29)}$$

To compile the global system equations, we make an assumption that the blocks are in perfect contact and no traction is applied at the interfaces. The equation of equilibrium, in terms of traction, is

$$\sum_{i=1}^{m} [\boldsymbol{t}_c^{i}] = 0 \quad . \qquad \textbf{(30)}$$

Using the expression of $[\boldsymbol{t}_c]$ given in (23), equation (30) becomes

$$\sum_{i=1}^{m} [\tilde{G}_{cx}^{i}][t_x^{i}] + \sum_{i=1}^{m} [\tilde{H}_{cx}^{i}][u_x^{i}] + \sum_{i=1}^{m} [\tilde{H}_{cc}^{i}][u_c^{i}] = 0 \quad . \qquad \textbf{(31)}$$

In general, for connections involving just two zones, the equilibrium equation based on traction can sufficiently define the equilibrium condition at the interface nodes. When there are more than two zones involved at a interface node, however, the condition of equilibrium derived from traction will generate more equations than the unknown variables (both $[\boldsymbol{u}]$ and $[\boldsymbol{t}]$) at the interface nodes. As a result, the system of equations is over determined and cannot be solved. To avoid this problem, we derive the equations of equilibrium at the interface based on the nodal force, that is

$$\sum_{i=1}^{m} [\boldsymbol{F}_c^i] = 0 \ . \tag{32}$$

Equation (32) states that the sum of the nodal forces $[\boldsymbol{F}_c]$ contributed by each block connecting to a node is equal to zero (at equilibrium). To obtain the nodal force vector $[\boldsymbol{F}]$, one can construct a conversion matrix $[C]$ that relates $[\boldsymbol{F}]$ to the tractions $[\boldsymbol{t}_b]$ as

$$[\boldsymbol{F}] = [C][\boldsymbol{t}] \tag{33}$$

The nodal forces are calculated using the principle of virtual work. If we apply a virtual displacement δu to a node, the work done by the tractions must be equal to the work done by the equivalent nodal forces at that node. This can be written as

$$\boldsymbol{F} \cdot \delta \boldsymbol{u} = \int_{\Gamma} \boldsymbol{t} \cdot \boldsymbol{u}(\delta u) \, d\Gamma \ , \tag{34}$$

where $\boldsymbol{u}(\delta u)$ denotes the boundary displacements due to the applied virtual displacement δu. Using equation (11) for the expression of strain energy, and putting $\delta u = 1$ along each axis of the Cartesian coordinate system, and at each boundary node in turn, one can evaluate the elements of the conversion matrix $[C]$.

Follow the procedure given above of partitioning a matrix into *external* and *interface* parts, equation (33) is rewritten as

$$[\boldsymbol{F}] = [C_x][\boldsymbol{t}_x] + [C_c][\boldsymbol{t}_c] \tag{35}$$

Substituting (23) into (35), we have

$$[\boldsymbol{F}] = [\tilde{C}_x][\boldsymbol{t}_x] + [\tilde{K}_x][\boldsymbol{u}_x] + [\tilde{K}_c][\boldsymbol{u}_c] \tag{36}$$

where

$$\begin{aligned} [\tilde{C}_x] &= [C_x] - [C_c][\tilde{G}_{cx}] \\ [\tilde{K}_x] &= [C_c][\tilde{H}_{cx}] \\ [\tilde{K}_c] &= [C_c][\tilde{H}_{cc}] \end{aligned} \tag{37}$$

As a result, the equilibrium equation based on nodal forces becomes

$$\sum_{i=1}^{m} [\tilde{C}_{cx}^{\,i}][t_x^i] + \sum_{i=1}^{m} [\tilde{K}_{cx}^{\,i}][u_x^i] + \sum_{i=1}^{m} [\tilde{K}_{cc}^{\,i}][u_c^i] = 0 \ . \tag{38}$$

Combining equations (31), (34), (37) and (44), we arrive at the global system equations

$$\begin{bmatrix} \grave{H}_{xx}^1 & N^{12} & \cdots & N^{1m} \\ N^{21} & \grave{H}_{xx}^2 & \cdots & N^{2m} \\ N^{31} & \vdots \ddots \vdots & & \\ \vdots \ \vdots & & \cdots & \grave{H}_{xx}^m \\ \tilde{K}_{cx}^1 & \tilde{K}_{cx}^2 & \cdots & \tilde{K}_{cx}^m \end{bmatrix} \begin{bmatrix} u_x^1 \\ u_x^2 \\ \vdots \\ u_x^m \end{bmatrix} = \begin{bmatrix} \grave{G}_{xx}^1 & M^{12} & \cdots & M^{1m} & -\grave{H}_{xc}^1 \\ M^{21} & \grave{G}_{xx}^2 & \cdots & M^{2m} & -\grave{H}_{xc}^2 \\ M^{31} & \vdots \ddots \vdots \ \vdots & & & \\ \vdots \ \vdots & & \cdots & \grave{G}_{xx}^m & -\grave{H}_{xc}^m \\ \tilde{C}_{cx}^1 & \tilde{C}_{cx}^2 & \cdots & \tilde{C}_{cx}^m & -\sum_{i=1}^{m} \tilde{K}_{cc}^i \end{bmatrix} \begin{bmatrix} t_x^1 \\ t_x^2 \\ \vdots \\ t_x^m \\ u_c \end{bmatrix} , \tag{39}$$

For the sake of simplicity, equation (31) has been embedded into (38) to represent the equilibrium equations in the global system matrices, and equation (28) embedded into the sub-matrices $[H_{xx}]$ and $[G_{xx}]$.

It should be noted that the sub-matrices denoted by $[M]$ and $[N]$ in (39) are formed to establish the condition of equilibrium (based on tractions or nodal forces) at the interface nodes that lie on the external surface of the structure (these are the nodes on the intersection where the external surface meets the block-to-block interface). The matrix equation is constructed in this way so that the global array $[u_x]$ represents the displacement vectors of the wetted surface of the global structure. It becomes clear that the array $[\boldsymbol{u}_c]$ in (39) denotes the displacement vectors at the interior of the interfaces between blocks. The global traction matrix $[G]$, the composite matrix on the right-hand size of equation (39), is square while the global displacement matrix $[H]$, the composite matrix on the left-hand side of (39), is rectangular. Furthermore, the sub-matrices $[G_{xx}]$ are square. This property of the traction matrix facilitates the numerical procedure of matrix inversion by partitioning (see, for instance, Ciarlet and Lion [15]).

The global stiffness matrix $[K]$, that relates the tractions on wetted surface $[\boldsymbol{t}_x]$ to the displacements on wetted surface $[\boldsymbol{u}_x]$ as

$$[t_x] = [K][u_x] \quad , \tag{40}$$

and the global influence matrix $[B_c]$, that relates the displacements at block-to-block interface $[\boldsymbol{u}_c]$ to $[\boldsymbol{u}_x]$ as

$$[u_c] = [B_c][u_x] \quad , \tag{41}$$

can be obtained, respectively, as the upper and lower partitions of the matrix product

$$\left[\frac{K}{B_c}\right] = [G]^{-1}[H] \quad . \tag{42}$$

We now find the influence matrix that maps the displacements at internal grid to the displacements at boundary, i.e. $[\boldsymbol{u}_i] = [B]\ [\boldsymbol{u}_x]$. The partitions that divide the structure into regions also distribute the internal grids to their respective regions. Consider the portion of the internal grid u_i^i that is confined by the i-*th* region, equation (4) is applied only to the i-*th* region as u_i^i is considered external points to other regions. Rewriting the influence matrices into partitions associated with the external grid and interface grid of the i-*th* region, we have equation (4) in the following form

$$[u_i^i]+[H_{ix}^i][u_x^i]+[H_{ic}^i][u_c^i] = [G_{ix}^i][t_x^i]+[G_{ic}^i][t_c^i] \tag{43}$$

Substituting equation (29) into (49), we have

$$[u_i^i] = [\grave{G}_{ix}^i][t_x^i]-[\grave{H}_{ix}^i][u_x^i]-[\grave{H}_{ic}^i][u_c^i] \quad , \tag{44}$$

where

$$\begin{aligned}
[\grave{H}_{ix}^i] &= [H_{ix}^i]-[G_{ic}^i][G_{cc}^i]^{-1}[H_{cx}^i] \\
[\grave{H}_{ic}^i] &= [H_{ic}^i]-[G_{ic}^i][G_{cc}^i]^{-1}[H_{cc}^i] \\
[\grave{G}_{ix}^i] &= [G_{ix}^i]-[G_{ic}^i][G_{cc}^i]^{-1}[G_{cx}^i] \quad .
\end{aligned} \tag{45}$$

Incorporating the displacement vectors $[\boldsymbol{u}_i]$ of each block into a global array, we can write equation (44) in terms of the global system matrices as

$$\begin{aligned}
\begin{bmatrix} u_i^1 \\ u_i^2 \\ \vdots \\ u_i^m \end{bmatrix} &= \begin{bmatrix} \grave{G}_{ix}^1 & 0 & \cdots & 0 \\ 0 & \grave{G}_{ix}^2 & \cdots & 0 \\ 0 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \grave{G}_{ix}^m \end{bmatrix} \begin{bmatrix} t_x^1 \\ t_x^2 \\ \vdots \\ t_x^m \end{bmatrix} \\
&- \begin{bmatrix} \grave{H}_{ix}^1 & 0 & \cdots & 0 \\ 0 & \grave{H}_{ix}^2 & \cdots & 0 \\ 0 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \grave{H}_{ix}^m \end{bmatrix} \begin{bmatrix} u_x^1 \\ u_x^2 \\ \vdots \\ u_x^m \end{bmatrix} - \begin{bmatrix} \grave{H}_{ic}^1 \\ \grave{H}_{ic}^2 \\ \vdots \\ \grave{H}_{ic}^m \end{bmatrix} \big[u_c\big] \ .
\end{aligned} \tag{46}$$

It follows that the global influence matrix $[B]$ is given by

$$\begin{aligned}
\big[B\big] &= \begin{bmatrix} \grave{G}_{ix}^1 & 0 & \cdots & 0 \\ 0 & \grave{G}_{ix}^2 & \cdots & 0 \\ 0 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \grave{G}_{ix}^m \end{bmatrix} \big[K\big] \\
&- \begin{bmatrix} \grave{H}_{ix}^1 & 0 & \cdots & 0 \\ 0 & \grave{H}_{ix}^2 & \cdots & 0 \\ 0 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \grave{H}_{ix}^m \end{bmatrix} - \begin{bmatrix} \grave{H}_{ic}^1 \\ \grave{H}_{ic}^2 \\ \vdots \\ \grave{H}_{ic}^m \end{bmatrix} \big[B_c\big] \ .
\end{aligned} \tag{47}$$

Considering the tractions and displacements on the wetted surface only and using the expression of global stiffness matrix $[K]$ given by (42), one can establish an expression for the strain energy of the global structure in the form $[\boldsymbol{u}_x]^{\mathrm{T}}\ [R]\ [\boldsymbol{u}_x]$, referring to the notations given in (17). Substituting the global matrices of $[B]$, and $[R]$ into the spline matrix formula, we obtain the spline matrix $[S]$ of the global structure.

For BEM models with multiple regions, the spline matrix given by equation (14) maps the displacement vectors $[u_x]$ at the external grids to the displacements of the structural (internal) grids, i.e. $[u_x] = [S][u_i]$. Using the expression for $[B_c]$ given by (42), one can relate the displacements at interface nodes $[u_c]$ to the displacements of the structural grids as

$$[u_c] = [B_c][S][u_i] \quad . \tag{48}$$

With the spline matrices $[S]$ and $[B_c][S]$, we can define the boundary deformation of the global body as well as the boundary deformation of each block in association with a given displacements of the structural grid.

2.5 Transfinite Interpolation

Deforming mesh within each flow block is executed independently using the arclength-based TFI method proposed by Jones and Samareh-Abolhassani[9]. In summary, the solution process of the arclength-based TFI method is implemented in the following steps:

1 Parameterize and normalise all grid points in the mesh.

2 Compute deformations at corners and on edges.

3 Compute mesh deformation using TFI.

4 Generate new grid by adding to the original grid the newly computed deformations.

In the arclength-based TFI approach, the grid coordinates are parameterized by the length of the grid lines in the *i*-, *j*-, and *k*-direction. In the *i* direction, for example, the arclength $s_{i,j,k}$ measured at a node with global index (*i, j, k*) is given by

$$s_{i,j,k} = \sum_{i=2}^{i} \left| r_{i,j,k} - r_{i-1,j,k} \right| \qquad \textbf{(49)}$$

where $r_{i,j,k}$ is the grid coordinates of the node at (*i, j, k*), and $s_{1,j,k} = 0$. The normalised arclength parameter $F_{i,j,k}$ for the grid lines in the *i* direction is obtained as

$$F_{i,j,k} = \frac{s_{i,j,k}}{s_{I,j,k}} \qquad \textbf{(50)}$$

for $i = 1, 2, ..., I$. The normalised arclength parameters for the grid lines in the *j*- and *k*-directions, denoted by $G_{i,j,k}$ and $H_{i,j,k}$ respectively, can be obtained in a similar fashion.

To deform a volume grid, the deformations at corner points or edges or faces of the flow block can be used as input to the TFI process. Depending on the types of inputs specified, the one-, two-, and three-dimensional TFI may be called upon in the process of deforming a volume grid. For instance, if only the deformations at corner nodes of a block are defined and provided with the TFI process, then the one-dimensional TFI formula is first applied to determine the deformations of mesh-edges which are then applied on by the two-dimensional TFI formula to determine the deformations on mesh-faces, and finally the three-dimensional TFI is performed to deform the volume grid.

Using the notations given by Tsai *et al.*[11] where ΔP, ΔE, ΔS and ΔV denote deformation of mesh-points, edges, faces, and volume, respectively, the one-dimensional TFI[11] takes the form (for the edge at $i = 1, j = 1$ in the k direction)

$$\Delta E_{1,1,k} = (1 - H_{1,1,k})\Delta P_{1,1,1} + H_{1,1,k}\Delta P_{1,1,K}\ , \qquad \textbf{(51)}$$

while the two-dimensional TFI [11] (for the face in the *i*=1 plane) is given by

$$\begin{aligned} \Delta S_{1,j,k} &= A_{1,j,k}\,\Delta E_{1,j,1} + B_{1,j,k}\,\Delta E_{1,j,K} \\ &+ C_{1,j,k}\,\Delta E_{1,1,k} + D_{1,j,k}\,\Delta E_{1,J,k} \\ &- A_{1,j,k}C_{1,j,k}\,\Delta P_{1,1,1} - B_{1,j,k}C_{1,j,k}\,\Delta P_{1,1,K} \\ &- A_{1,j,k}D_{1,j,k}\,\Delta P_{1,J,1} - B_{1,j,k}D_{1,j,k}\,\Delta P_{1,J,K} \end{aligned} \qquad \textbf{(52)}$$

where the blending functions A, B, C, and D are those proposed by Soni[16] based on the arclength. The expressions for the blending functions are

$$\begin{aligned} A_{i,j,k} &= 1 - \eta_{i,j,k} \\ B_{i,j,k} &= \eta_{i,j,k} \\ C_{i,j,k} &= 1 - \xi_{i,j,k} \\ D_{i,j,k} &= \xi_{i,j,k} \end{aligned} \qquad \textbf{(53)}$$

and

$$\begin{aligned} P_{1,j,k} &= 1 - \left(G_{1,j,K} - G_{1,j,1}\right)\left(H_{1,J,k} - H_{1,1,k}\right) \\ \xi &= \frac{G_{1,j,1} + H_{1,1,k}\left(G_{1,j,K} - G_{1,j,1}\right)}{P_{1,j,k}} \\ \eta &= \frac{H_{1,1,k} + G_{1,j,1}\left(H_{1,J,k} - H_{1,1,k}\right)}{P_{1,j,k}} \end{aligned} \qquad \textbf{(54)}$$

The three-dimensional TFI formula[11] is given by

$$\Delta V_{i,j,k} = V1 + V2 + V3 - V12 - V13 - V23 + V123 \qquad \textbf{(55)}$$

where

$$\begin{aligned} V1 &= (1 - F_{i,j,k})\,\Delta S_{1,j,k} + F_{i,j,k}\Delta S_{I,j,k} \\ V2 &= (1 - G_{i,j,k})\,\Delta S_{i,1,k} + G_{i,j,k}\Delta S_{i,J,k} \\ V3 &= (1 - H_{i,j,k})\,\Delta S_{i,j,1} + H_{i,j,k}\Delta S_{i,j,K} \end{aligned}$$

$$\begin{aligned} V12 &= (1 - F_{i,j,k})(1 - G_{i,j,k})\,\Delta E_{1,1,k} + (1 - F_{i,j,k})\,G_{i,j,k}\Delta E_{1,J,} \\ &+ F_{i,j,k}(1 - G_{i,j,k})\,\Delta E_{I,1,k} + F_{i,j,k}G_{i,j,k}\Delta E_{I,J,k} \\ V13 &= (1 - F_{i,j,k})(1 - H_{i,j,k})\,\Delta E_{1,j,1} + (1 - F_{i,j,k})H_{i,j,k}\Delta E_{1,j,K} \\ &+ F_{i,j,k}(1 - H_{i,j,k})\,\Delta E_{I,j,1} + F_{i,j,k}H_{i,j,k}\Delta E_{I,j,K} \\ V23 &= (1 - G_{i,j,k})(1 - H_{i,j,k})\,\Delta E_{i,1,1} + (1 - G_{i,j,k})H_{i,j,k}\Delta E_{i,1,K} \\ &+ G_{i,j,k}(1 - H_{i,j,k})\,\Delta E_{i,J,1} + G_{i,j,k}H_{i,j,k}\Delta E_{i,J,K} \end{aligned} \qquad \textbf{(56)}$$

$$\begin{aligned} V123 &= (1 - F_{i,j,k})(1 - G_{i,j,k})(1 - H_{i,j,k})\Delta P_{1,1,1} \\ &+ (1 - F_{i,j,k})(1 - G_{i,j,k})H_{i,j,k}\Delta P_{1,1,K} \\ &+ (1 - F_{i,j,k})\,G_{i,j,k}(1 - H_{i,j,k})\Delta P_{1,J,1} \\ &+ (1 - F_{i,j,k})\,G_{i,j,k}H_{i,j,k}\Delta P_{1,J,K} \\ &+ F_{i,j,k}(1 - G_{i,j,k})(1 - H_{i,j,k})\Delta P_{I,1,1} \\ &+ F_{i,j,k}(1 - G_{i,j,k})H_{i,j,k}\Delta P_{I,1,K} \\ &+ F_{i,j,k}G_{i,j,k}(1 - H_{i,j,k})\Delta P_{I,J,1} \\ &+ F_{i,j,k}G_{i,j,k}H_{i,j,k})\Delta P_{I,J,K} \end{aligned}$$

3. SOLUTION PROCEDURES

Specific to aeroelastic analysis using coupled structure and aerodynamic models, the solution procedures of deforming flow mesh are summarised as follows.

For the flexible body:

1 Build the BEM model of the flexible body and compute the spline matrix [S] using equation (14).

2 For a specified deformation of the structure grid [$\boldsymbol{u}_s$], compute the boundary displacements [$\boldsymbol{u}_a$] using equation [$\boldsymbol{u}_a$] = [S][$\boldsymbol{u}_s$].

For the field mesh:

3 Build the BEM model of the field grid and compute the spline matrix [B_f] using equation (47).

4 Compute flow-block motions [$\boldsymbol{u}_f$] using the equation [$\boldsymbol{u}_f$] = [B_f] [$\boldsymbol{u}_a$].

5 Deform flow grid by performing TFI, equations (51), (52) and (55), on each flow block.

Repeat steps (2), (4) and (5) in subsequent mesh deformation processes.

4. RESULTS AND DISCUSSIONS

Numerical examples are given here to demonstrate the capability of the BEM-based interpolation method in deforming flow mesh. For the sake of clarity, two dimensional problems are considered first to demonstrate the solution procedures involved, followed by a three dimensional problem with a wing section.

Consider a two-dimensional flexible body, Figure 3(a), which is modelled by two BEM blocks. A multiple-block flow grid configuration is used, as shown in Figure 3(c). The undeformed flow grids are shown in Figure 3(e). In order to perform flow-mesh deformation using the BEM-based technique, the flow domain is considered as an elastic body, and the multiple-block BEM is used to represent it. Since the flow grids have matching boundary, the BEM model of the flow domain is formed by taking the flow blocks as the BEM blocks. The spline matrix [B_f] is computed to relate the flow-block deformation to the deflection of the solid body.

When the flexible body is deformed, the flow blocks are deformed by using the spline matrix [B_f]. Deformations of flow blocks associated with the deflection of the solid body are shown in Figure 3(d). To deform the flow grid, the TFI procedures are applied to each deformed flow blocks. The deformed flow grids are computed as shown in Figure 3(f).

In the second example, a system containing two flexible bodies is considered, we demonstrate the usage of the method for deforming overset grids. Figure 4(a) shows the configuration of the undeformed bodies while Figure 4(c) depicts the BEM model of the undeformed flow domain. For each body, a body-fitted grid is formed independent of each other, Figure 4(e). The BEM models of the body-fitted grids are also generated. (For clarity, only the overset grids are shown in the figures while the flow mesh covering the rest of the flow domain is omitted.) The spline matrix [B_f] is computed for the flow blocks to map the deformation of the flow-blocks to the deflection of the flexible bodies. By considering the body-fitted grid as internal nodes of the flow-block BEM model, spline matrices [B_{f1}] and [B_{f2}] are computed to relate the deformations of the body-fitted grids to the deformation of the global flow domain.

In response to the deflections of the flexible bodies, Figure 4(b), the flow blocks are deformed using spline matrix [B_f]. Once the deformation of the global flow domain has been computed, the motion of the body-fitted grid around each body is computed using spline matrices [B_{f1}] and [B_{f2}]. The deformed BEM models of the flow blocks and the body-fitted grids are depicted in Figure 4(d). Applying the TFI procedures on the deformed overset grids, we update the flow mesh within each body-fitted grid, Figure 4(f). It is evident from the consistent patterns of overlapping between the two body-fitted grids, Figure 4(e) and Figure 4(f), that the BEM-based mesh deformation method preserves the connectivity of the original grids in the deformed grids.

Application of the BEM-based interpolation method for deforming three-dimensional flow mesh is given in the third example. In this example, the BEM-based mesh deformation method is integrated with the CFD-CSD interaction method to form an unified approach. For a given deformation of the structure grid, the CFD-CSD interaction method is applied to determine the motion of the boundary grid. The boundary deformation is subsequently propagated into the field mesh using the mesh deformation method. Consider an AGARD 445.6 wing section, the fluid model and the structure model of which are shown in Figure 5(a) and Figure 5(c), respectively. The flow domain is modelled with 32 multiple-grid flow blocks. Figure 5(e) shows one half of the flow domain. The near-body flow blocks are shown magnified in Figure 5(f). The spline matrix [S] for the interaction between the CFD grid and the CSD grid is formed using equation (14). As the structure grid is deformed, the associated deformation of the boundary grid is computed by the spline matrix [S]. Figures 5(d) and 5(b) show the deformation of the structure grid and the associated motion of the wing section.

To propagate the boundary perturbation into the field mesh, the BEM model of the flow domain is formed and the spline matrix [B_f] computed to relate the flow-block motions with the deformation of the boundary grid. This is followed by deforming the field mesh within each flow block using the TFI formulation according to equation (55). For clear visualisation of the deforming flow blocks, and without loss of generality of the method, the BEM models of only the near-body flow blocks are formed and a section of which is shown in the cut-away view of Figure 5(g). The flow-block deformations associated with the deflection of the structure grid are shown in Figure 5(h).

5. CONCLUSIONS

The paper presented an interpolation method based on the boundary element method and discussed its applications in deforming flow mesh and in interaction between fluid model and structure model in aeroelastic simulations. In present approach with the BEM-based interpolation method, the spline matrix [*S*] for CFD-CSD interaction and the spline matrix [B_f] for deforming flow mesh are computed at the onset of computation. They are stored in disk and retrieved when they are needed in the solution process. The spline matrix, based on linear elasticity, is invariant with the deformed shape of the flexible body and thus valid for any elastic deformation that the flexible body may assume. The initial investment in terms of computing resources required for computing the BEM-based spline matrices may be higher than conventional interpolation methods. The efficiency, speed and robustness that the BEM-based interpolation method offers make it a feasible approach for aeroelastic simulation of large and complex configurations.

6. REFERENCES

[1] Brebbia, C.A. and Dominguez, J., *Boundary Elements: An Introductory Course*, McGraw-Hill Book Co, New York, 1992.

[2] Chen, P.C. and Jadic, I., "CFD/CSD Interfacing via an Innovative Structural Boundary Element Approach," AIAA Paper 97-1088, 38th SDM Conference, April 7-10, 1997, Kissimmee, FL.

[3] Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," AIAA Journal, Vol. 28, No. 8, August 1990, pp. 1381-1388.

[4] Lee-Rausch, E. M. and Batina, J. T., "Wing Flutter Boundary Prediction Using Unsteady Euler Aerodynamic Method," *Journal of Aircraft*, Vol. 32, No. 2, March-April 1995, pp. 416-422.

[5] Lee-Rausch, E. M. and Batina, J. T., "Wing Flutter Configurations Using an Aerodynamic Model Based on the Navier-Stokes Equations," *Journal of Aircraft*, Vol. 33, No. 6, Nov.-Dec. 1996, pp. 1139-1147.

[6] Prananta, B. B. and Hounjet, M. H. L., "Large Time Step Aero-Structural Coupling Procedures for Aeroelastic Simulation," *In Proceedings of the International forum on Aeroelasticity and Structural Dynamics, Rome, Italy*, Vol. 2, Confederation of European Aerospace Societies, June 1997, pp. 63-70.

[7] Byun, C. and Guruswamy, G. P., "Aeroelastic Computation on Wing-Body-Control Configurations on Parallel Computers," *Journal of Aircraft*, Vol. 35, No. 2, March-April 1998, pp. 288-294.

[8] Ji, S. and Liu, F., "Flutter Computation of Turbomachinery Cascades Using a Parallel Unsteady Navier-Stokes Code," AIAA Journal, Vol. 37, No. 3, March 1999, pp. 320-327.

[9] Jones, W.T. and Samareh-Abolhassani, J., "A Grid Generation System for Multi-Disciplinary Design Optimization," AIAA Paper 95-1689-CP, June 1995, 12th AIAA Computational Fluid Dynamics Conference Proceedings, Part I, pp. 74-482.

[10] Reuther, J., Alonso, J., Rimlinger, M.J., and Jameson, A., "Aerodynamic Shape Optimization of Supersonic Aircraft Configurations via an Adjoint Formulation on Distributed Memory Parallel Computers," *Computers and Fluids*, Vol. 28, 1999, pp. 675-700.

[11] Tsai, H.M., Wong, A.S.W., Cai, J., Zhu, Y. and Liu, F., "Unsteady Flow Calculation with a Parallel Multiblock Moving Mesh Algorithm," *AIAA Journal*, **Vol. 39**, No. 6, June 2001, pp. 1021-1029.

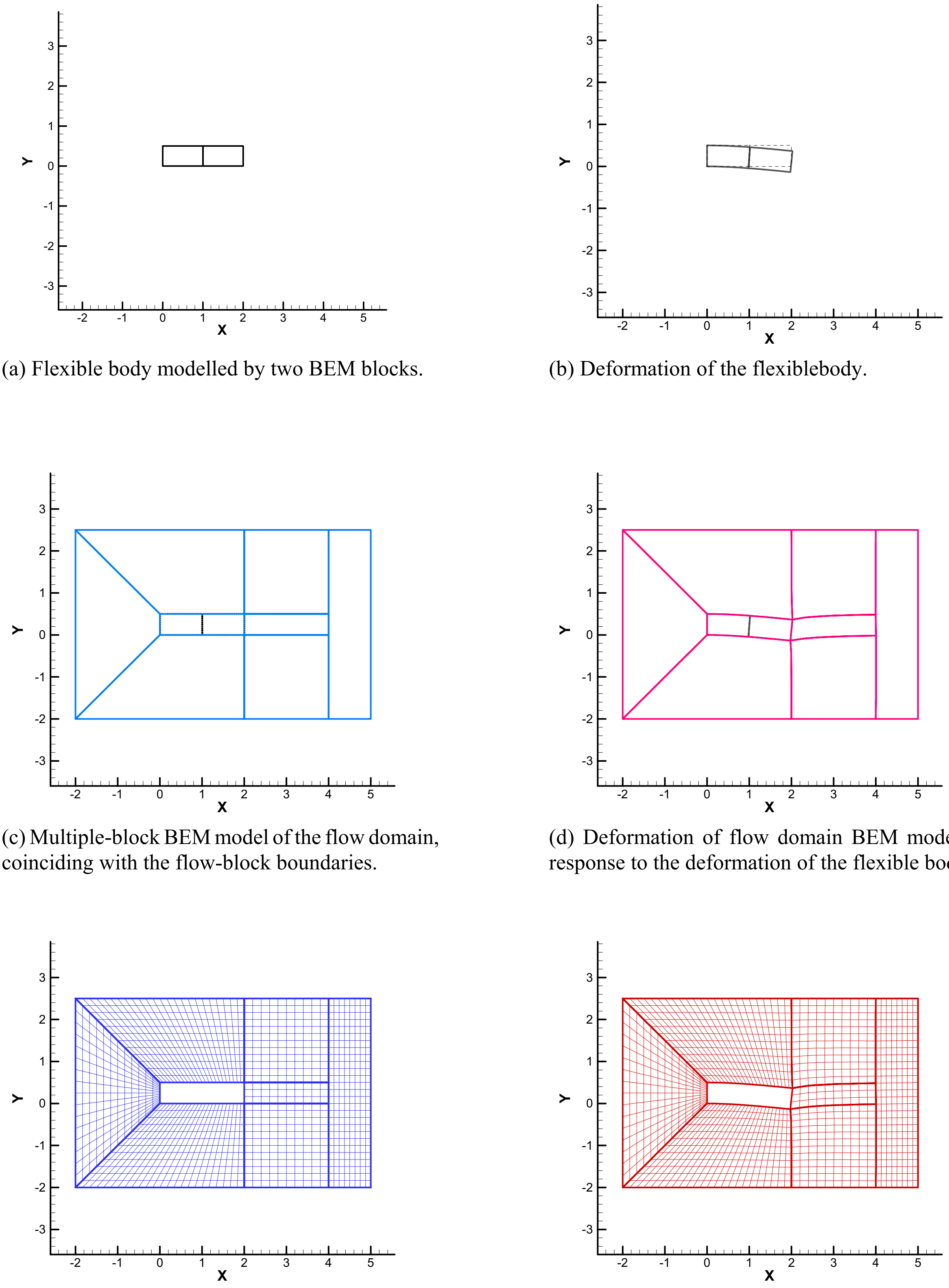
[12] Chaudonneret, M., "On the Discontinuity of the Stress Vector in the Boundary Integral Equation Method for Elastic Analysis," in *Recent Advances in Boundary Element Methods*, ed. C.A. Brebbia, Pentech Press, London, 1978.

[13] Rudolphi, T.J., "An Implementation of the Boundary Element Method for Zoned Media with Stress Discontinuities," *Int. J. Num. Meth. Engng.*, **Vol. 19**, 1983, pp. 1-15.

[14] Gao, X.W. and Davies, T.G. "3D Multi-Region BEM with Corners and Edges," *Int. Journal of Solids and Structures*, **Vol. 37**, 2000, pp. 1549-1560.

[15] P.G. Ciarlet, J. L. Lions. *Handbook of numerical analysis.* Publisher Amsterdam; New York; Oxford, England: North Holland, 1990-

[16] Soni, B. K. "Two- and Three-Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics," AIAA Paper 85-1526, 1985.



(a) Flexible body modelled by two BEM blocks.

(b) Deformation of the flexiblebody.

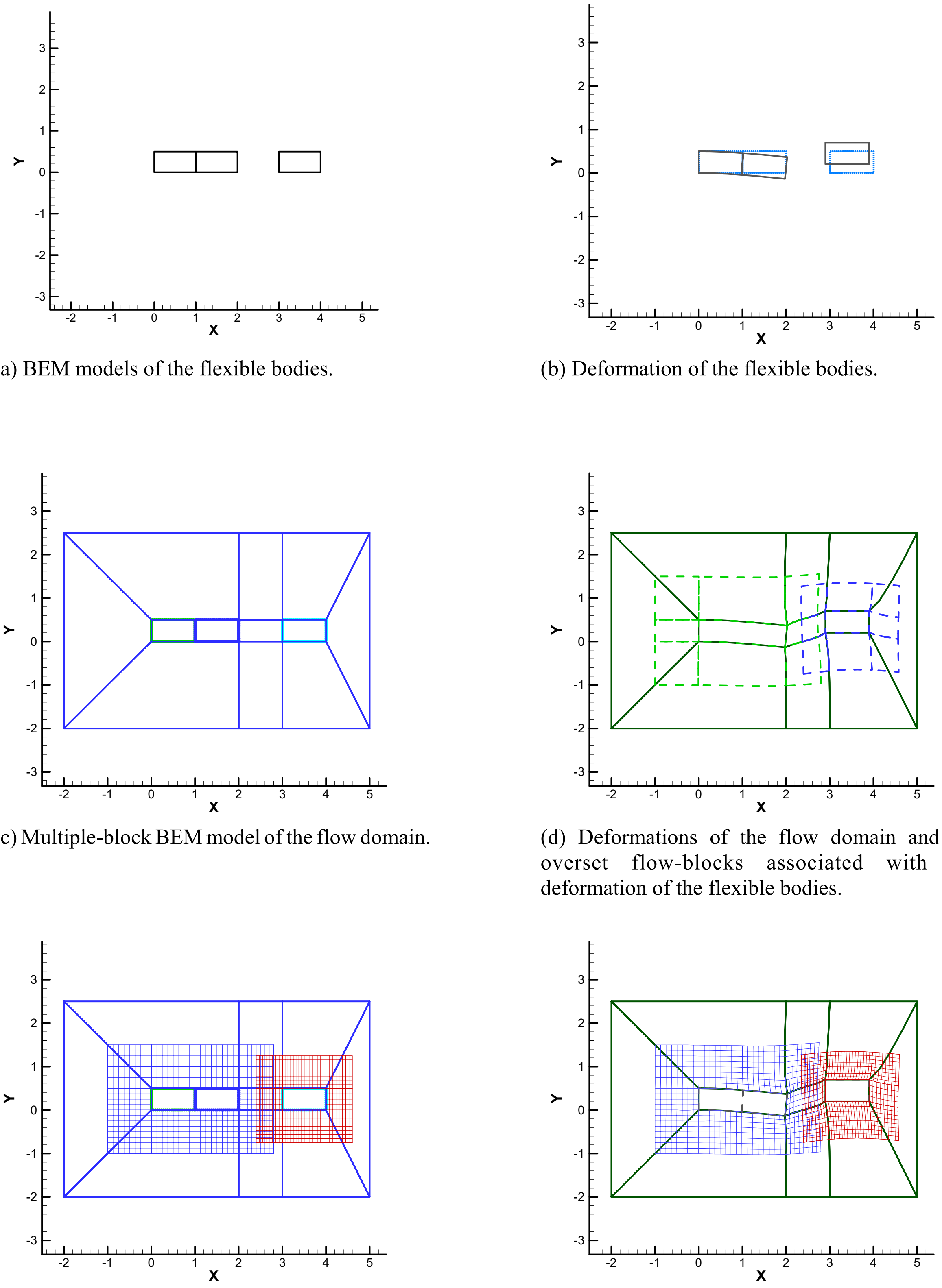
(c) Multiple-block BEM model of the flow domain, coinciding with the flow-block boundaries.

(d) Deformation of flow domain BEM model in response to the deformation of the flexible body.

(e) Configuration of the undeformed flow mesh.

(f) Configuration of the deformed flow mesh.

Figure 3. Mesh deformation using the BEM-based interpolation and Trans-Finite Interpolation; application on a multiple-block moving grid system.



(a) BEM models of the flexible bodies.

(b) Deformation of the flexible bodies.

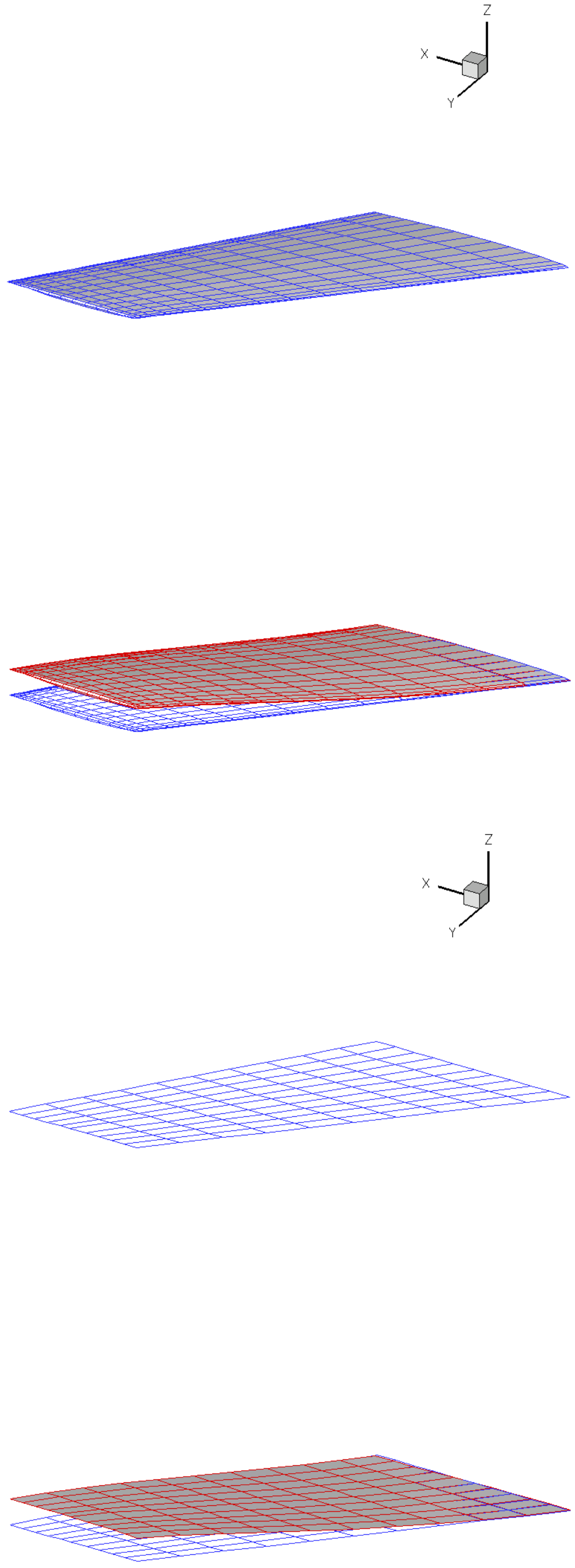
(c) Multiple-block BEM model of the flow domain.

(d) Deformations of the flow domain and the overset flow-blocks associated with the deformation of the flexible bodies.

(e) Body-fitted overset grids of the flexible bodies.

(f) Deformation of overset grids due to deflections of the flexible bodies.

Figure 4. Application of the BEM-based mesh deformation method on overset grids.

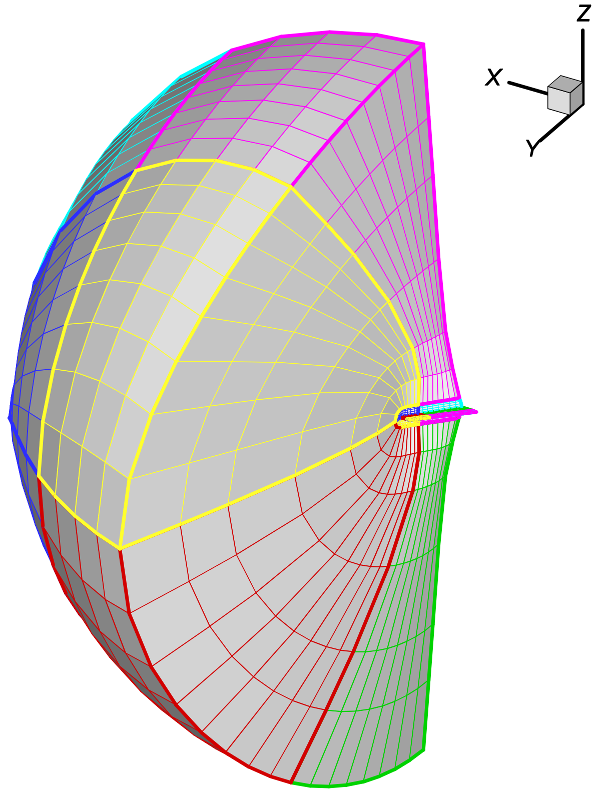


(a)

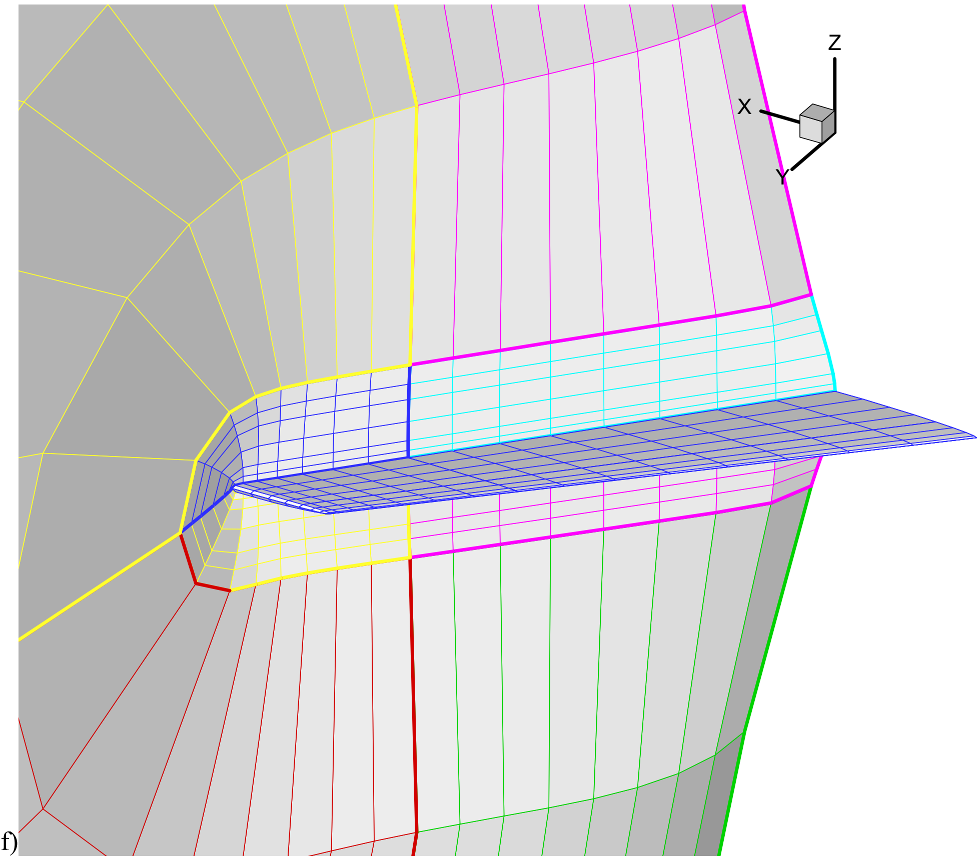
(b)

(c)

(d)



(e)



(f)

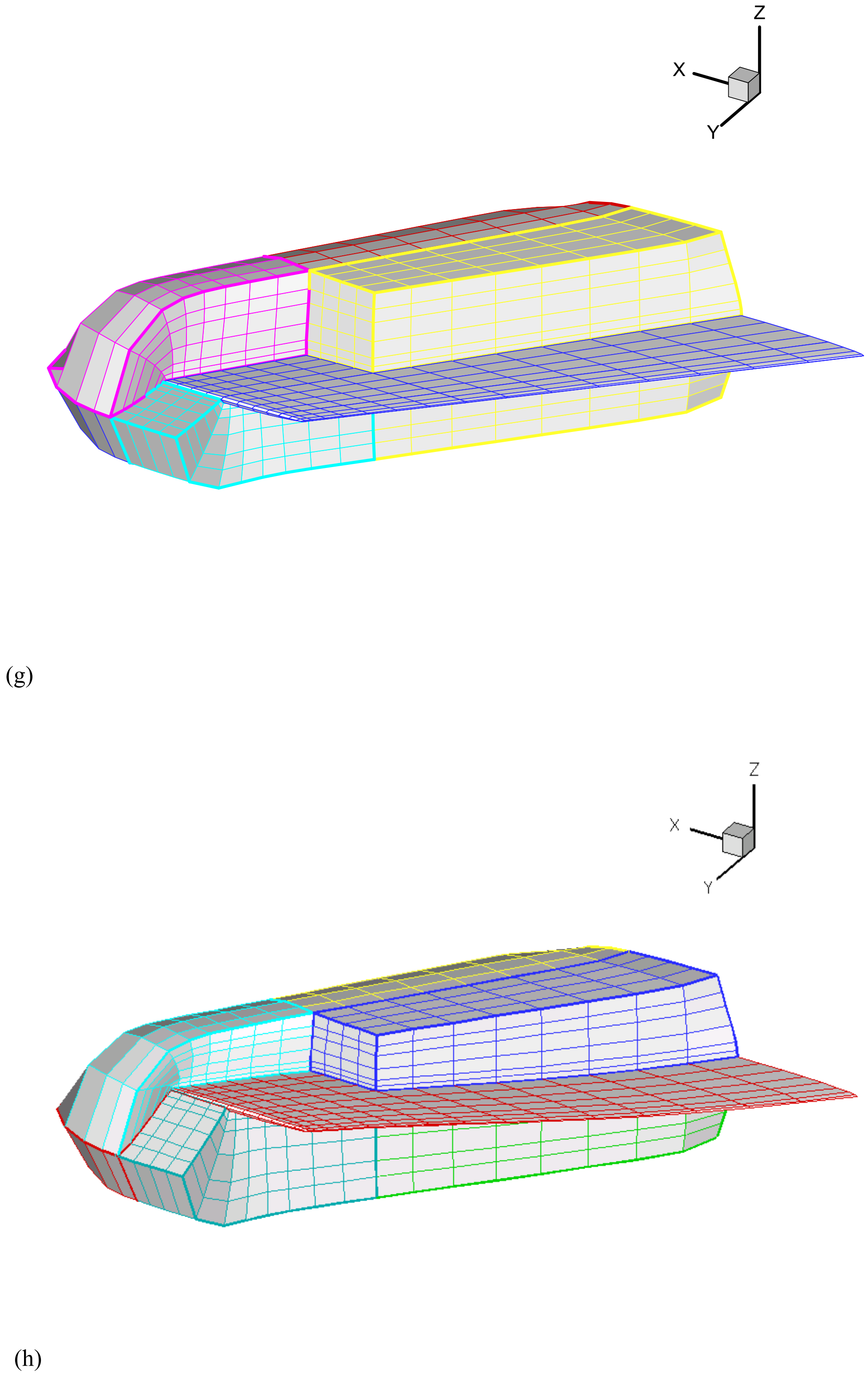


Figure 5. Deforming flow mesh of AGARD 445.6 wing section using BEM-based interpolation and TFI methods. (a) CFD model of wing section; (b) Deflection of wing section; (c) Structural grids; (d) Deformation of structural grids; (e) BEM model of flow domain, shown here is the rear half of the domain. (f) Close-up view of near-body flow domain; (g) Cut-away view of undeformed near-body flow domain; (h) Deformation of flow domain associated with wing deflection.