A Pixelation Method for Loads Transfer on Fluid Structure Interface

Pan Cheng¹,*, Yuchen Sun¹, Yang Zhao¹, Zekai Ding², Wu Zhang²

¹ Shanghai Aircraft Design & Research Institute, Shanghai, P.R. China
² Shanghai Institute of Applied Mathematics and Mechanics, Shanghai University, Shanghai, P.R. China

*Corresponding author: chengpan@comac.cc

Abstract. Fluid Structure Interaction (FSI) is widely applied in aircraft aerodynamic and aeroelastic analysis. Iteration between the CFD and FEA will solve the aerodynamic force, flow field, structure stress and deformation under equilibrium state simultaneously. In aircraft product development, the CFD mesh and Finite element model normally do not match each other because the goal of CFD and FEA are different. To connect between unmatched interface, a pixelation method is developed to transfer the aerodynamic loads onto the finite element model correctly. The pixelation method is also easy to be implemented and parallelized.

1. Introduction
In order to achieve higher aerodynamic efficiency, advanced civil aircrafts are designed to operate in the transonic regime with high aspect ratio wings. Static aeroelasticity, which refers to the interaction between aerodynamic and structural force, leads to significant deformation under flight loads especially for high aspect ratio wings. The wing shape change due to deformation has a strong impact on the aircraft aerodynamic performance and redistributes the load on the wings. As a result, precise prediction of aerodynamic performance and flight loads for elastic aircrafts is a strong need in the aerospace industry. To analyze static aeroelastic problem for large aspect ratio wing under transonic condition, fluid structure interaction is carried out in a procedure integrated with computational fluid dynamics (CFD) code and finite element analysis (FEA) code[1]. The CFD and FE code have been well developed by which the nonlinear behavior of aerodynamic and structure can be predicted. Since the source code are not always available, the coupling process will run CFD and FEA codes by turn and transfer loads and motion between aerodynamic mesh and finite element grid[2]. The loads transfer is of important since the transferred loads on structure should be precisely reflect the loads on aerodynamic mesh. The loads transfer has been implemented with two different mesh strategies. The first strategy focus on the relative position of aerodynamic mesh cells and structure grid so that geometry relation information need to be find out in transfer[3][4]. This class of methods is very accurate in some specific problem, however it is difficult to be used automatically or to be applied on non-match grid. The second strategy based on energy conservative law and the topological relation between aerodynamic mesh and structure grid is unrelated so that it is easy to transfer loads automatically[5][6], nonetheless the loads transfer
might be difficult to control, which means the energy is conservative meanwhile the local distribution could be non-match.

A few methods are developed to solve the FSI loads transfer problem. Early research focus to build a unify connection between the aerodynamic and structure, while the aerodynamic side and structure side are both solved in linearized way. So naturally the spline method are developed to connect two sides, both for displacement and loads. The classic Thine Plate Spline is still used in aeroelastic analysis module in NASTRAN. Then the spline method is extend to RBF interpolation. RBF interpolation is easy to be implemented and suits for unmatched interface, because it only use the information on grid points. Although the RBF method tightly keep the total force and moment before and after transfer, the problem of RBF interpolation method is it cannot keep the distribution of loads consistent through the loads transfer, especially when the grid of structure side is not arranged evenly.

To keep the distribution and summary of loads on the interface consistent simultaneously, some new ideas are introduced and accordingly new methods are developed. Constant Volume Tetrahedral(CVT) is a representative of these new ideas[7]. In FSI problem, CVT will try to find nearest three structure grids for each CFD mesh point. Then a tetrahedral element can be setup by all four point and the tetrahedral element is treated as normal finite element. The finite element will be solved for displacement and loads, and transfer them between CFD and FEA grid. When the CFD mesh is large the CVT finite elements could also be large. The algorithm of CVT is more difficult to be implemented than RBF. C. Farhat develop a momentum and energy conservation method for FSI and it require both side should have surface mesh[8]. It project the fluid surface grid onto the structure surface cell as Gauss point and calculate the loads by shape function interpolation. It uses the surface information of structure side and constrain the interpolation in one cell, not as a global interpolation as RBF interpolation. Same way could be applied on surface mesh movement. To make the projection more precisely, Jiao and Jaiman developed common refinement base data transfer method[9]. After cutting the CFD mesh and FEA mesh with each other, the common refinement method establishes a cell-to-cell projection relationship, which successfully overcomes the shortcomings of point-to-element projection method such as spurious oscillations and overshoots in the load transferred onto the solid mesh. Both Farhat’s method and common refinement based method require the surface mesh on two sides, otherwise the projection cannot be performed.

Back to the problem that engineers facing, the structure model normally only contain the main bearing structure such as spar, rib, stinger and part of the skins, not describe the whole surface of a wing. Sometimes, even a simple beam are used as a wind structure in very preliminary stage of aircraft design. For those special condition, RBF interpolation cannot keep the global conservative, CVT is difficult to be implemented and Farhat’s projection and common refinement based method require too much information that not possible to give in aircraft design. To meet the challenge of the FSI in engineering, pixellation method is develop to meet the requirement.

2. Problem Statement and Summarization

The FSI problem is solved by CFD and CSD alternatively and exchange data on the boundary $\Gamma$ between each other. On structure side, the elastic equation is formulated in Lagrangian coordinates in which grids is move with structure and boundary is denoted as $T_s$. Finite volume method is used to solve Navier-Stokes/Eular equations described in Eulerian form, which means volume cell coordinates do not vary with time. The boundary of fluid side is denoted as $T_f$, on which pressure and viscous stress calculated by CFD are stored as the known information. The total force and moment should be conservative and more than that, the distributive force and moment should matched before and after transferring. Usually, the fluid side is called source and structure is called target.

Mapping method includes many typical algorithms such as mortar element methods[8] and common refinement method[10][11]. This class of methods have strong mathematical foundation and conservation requirement are proved to be strictly satisfied. The quantities on source and target side are treated as function $f$ and $g$ separately. The linear combination of the shape functions and quantities on the nodes on target and $m$ nodes on source is used to define function $f$ and $g$, which are denoted as $f = \}$
\[ \sum_{i=1}^{m} f_i \phi_i \] and \[ g = \sum_{i=1}^{n} g_i \phi_i. \] The error of load should be minimized during transfer so that the \( L_2 \) norm of \( g - f \) should be minimized. The equivalent proposition can be described as the solution to a linear system \( \mathbf{M} \mathbf{x} = \mathbf{b} \), in which \( \mathbf{M} = \int_A \psi_i \psi_j \, dx \) and \( \mathbf{b}_i = \int_A \psi_i f \, dx \). The discretization of \( L_2 \) minimization is the key to calculate every element in the matrix \( \mathbf{M} \) and vector \( \mathbf{b} \)[12].

Different discretization are available, including source-based, target based and common refinement. The first two classes calculate \( \int_A \psi_i f \, dx \) by integrate over source mesh or target mesh. The common refinement is different from above two, it utilizes the overlap of meshes on both side and integrate over the smallest subdomains created by intersection. The accuracy, stability and efficiency are proved to be best among most discretization schemes since the common refinement fully use the geometry information on the surface mesh. However this feature also limit the application of common refinement because the surface mesh on both side are necessary and should be based on the exactly same geometry.

Point wise method, mentioned as interpolation or mesh free method in some literatures, is based on radial basis function (RBF)[13], multivariate splines[14] or any type of multivariate interpolation method[15][16][17]. These methods are used to interpolate displacement of the points on the fluid structure boundary and the concentrated force are transferred by virtual work conservation over the boundary. Discretization of displacement on \( \Gamma \) is formulated as \( \delta \mathbf{u}_f = \mathbf{H} \delta \mathbf{u}_s \), where the \( \mathbf{H} \) is the interpolation matrix. The virtual work performed by the fluid loads and the structure stress should be equal

\[
\delta \mathbf{W} = \delta \mathbf{u}_f^T \cdot \mathbf{F}_f = \delta \mathbf{u}_s^T \cdot \mathbf{F}_s
\]

where \( f_f \) is the discretized concentrated force on fluid points and \( f_s \) are the same one on structure grids. Then the force relation can be deduced as

\[
\mathbf{F}_s = \mathbf{H}^T \cdot \mathbf{F}_s \tag{2}
\]

In different interpolation methods, the \( \mathbf{H} \) matrix can be constructed based on variant basis functions. The earliest interpolation in aeroelastic is proposed in [5], in which thin plate surface(TPS) function are used as the basis function to transfer loads. TPS function in the form of \( \phi(|\mathbf{x}|) = |\mathbf{x}|^{2}\log (|\mathbf{x}|) \) is a special case of RBFs, actually acts as the fundamental solution of infinite plate equations and follows the principle of minimal potential energy. Then many interpolation based on different RBFs are proposed, such as multiquadric, Gaussian or Wendland \( C^n \) functions[18]. Some of them are compact supported, which means the affect range ( circle in 2D and sphere in 3D) of these basis function is adjustable to balance between accuracy and efficiency[19]. Besides RBFs, some other methods use kriging interpolation or BEM to construct \( \mathbf{H} \) matrix.

The interpolation method is easy and flexible to use since the equation is simple and only points are used. However, interpolation also have some drawbacks in practical problems. The first is that the distribution of quantities is not strictly conservative because it use a different basis function instead of the source or target basis function in mapping methods. The basis functions used in mapping implies the boundary geometry information while the basis functions used in point wise method have no relevant to the geometry. It is not big deal when the points are dense and smooth enough but some time it is hard to achieve and big error will come out. Second, a \( m \times m \) linear equations need to be solved to gain \( \mathbf{H} \) matrix where \( m \) is the grids number on structure side. When \( m \) is very large, it is not computationally efficient to solve the equations in a FSI analysis. Although reduce method are introduced[20], the cost is to lower the accuracy. Third, in some interpolation such as RBFs based, three or more nonlinear points must be specified while it is not satisfied when the structure is a beam like model and auxiliary point should be added.

3. Pixelation method
In engineering environment, the mapping and point wise method are not satisfied in some special cases mentioned above. A better method should be design to avoid the drawbacks listed above and a list of requirements are introduced as following.
a) The loads received on structure side should be point wised. This means that the surface mesh on structure side will not be necessary, only points are needed to transfer loads.

b) The conservation should be satisfied, say \( \int_{\Omega} |f - g|^2 d\Omega \) and \( \int_{\Omega} |f - g| d\Omega \) should be minimized or closed to minimization, which means the loads should be conservative before and after transfer.

c) The computation should be robust and efficiency enough to be adaptive to most situation encountered in engineering.

Consider discretized \( \Gamma_s \) is represented by discrete point set \( P_s \), while \( \Gamma_f \) is also discretized as points set \( P_f \) and cells set \( C_f \). On structure side, only points are given while on fluid side, both points and cells are given. The boundary is shown as Figure 1.

![Figure 1](image1)

**Figure 1.** The boundary of structure and fluid side and their discretization

![Figure 2](image2)

**Figure 2.** The Vonoroi diagram of structure points and overlap of two meshes

The basic ideal of the new method is similar to common refinement method, but only surface mesh on fluid side are utilized. There should be a surface mesh on structure side so that to perform intersection, however it is not existed on structure side and it should be reconstructed by existed surface mesh on fluid side. A simple way to reconstruct the surface mesh on structure side is to apply Voronoi diagram[21]. In Voronoi diagram, the space is decomposed to polygon cells and every cell match to one of given points. In other words, any point on the edges of these cells is equally distant to two nearest points in given set. The Voronoi diagram of \( \Gamma_s \) in Figure 1 is shown as (a) in Figure 2. Euclidian distance is referred during the process of cutting surface mesh into smaller pieces.

After the surface mesh reconstruction by Voronoi diagram, the common refinement scheme could be applied by intersecting two meshes theoretically. However, generation of Voronoi diagram on three dimensional surface could be very difficult and integration operation need to be performed after surface mesh generation and intersection. A natural thought is to combine these two steps together, so the pixelation on surface mesh on fluid side is introduced. The edge of Voronoid diagram can be approximated by splitting and the edge will be displayed as zigzag curve along the small cells generated by pixelation method. The implementation of the procedure will be described as following.

Let’s consider a case of only one fluid cell. For an arbitrary cell \( c_i \in C_f \) on fluid mesh, a geometry center of \( c_i \) is defined as \( \hat{c}_i \) and its two nearest neighbor points \( p_{\hat{c}_i}^k \) \( k = 1,2 \) can be found in \( P_s \) if the distance

\[
d(\hat{c}_i, c_i) = \min_k (d(p|p \in P_s, \hat{c}_i))
\]
The distance mentioned is Euclidean distance. Let’s denote \(d(p_i^k, \hat{c}_i)\) as \(d_k^i\). The edge of Voronoi diagram of this cell should be the intersection of the middle plane of \(p_i^k\) for \(k = 1,2\) and the cell. The intersection line on the cell could be approximated by recursively splitting the cell. First, we define a measurement of the cell size as \(s_i\) for each cell \(c_i\), for example the longest edge of triangular cell or longest diagonal of quadrilateral cell. If \(s_i > |d_{1i}^3 - d_{2i}^3|\), the cell should be split into smaller sub cells and classify the pieces by distance. Then, for an arbitrary structure point \(p_j \in P_s\), the nearest cells (including split and non-split cells) will be found through the process above and put them into a subset denoted as \(C_j^{sub}\). The pixelation process is conducted recursively until the cell size \(s_i\) smaller than a predefined value \(\varepsilon\). The Voronoi cell of \(p_j\) is approximately covered by the sub cells in \(C_j^{sub}\).

![Figure 3. A square cell with 3 structure points sample case](image)

A sample case with one square cell with square side \(l\) and three structure points in 2D space is shown in Figure 3. The three structure points \(p_1, p_2\) and \(p_3\) are marked by color blue, green and red, respectively. Their sub cell sets are marked by the same colors. At the beginning, the fluid cell split into 4 pieces and their color shows which structure point they belongs to. But the distance threshold didn’t satisfied, so keep splitting until the threshold reached. After 12 time splitting, the size of smallest cell is less than threshold \(\varepsilon = l/4096\) and the splitting is stopped. The square cell are cut into 3 blocks and every block containing split cells can be regarded as approximation of Voronoi cells. The edge of the Voronoi diagram is not straight lines, but zigzags curve with small steps. Because the splitting termination condition is \(\varepsilon > s_i > |d_{1i}^3 - d_{2i}^3|\) and \(|d_{1i}^3 - d_{2i}^3|\) always greater than 0, \(\lim_{\varepsilon \to 0} |d_{1i}^3 - d_{2i}^3| = 0\). This guarantee the geometry center of the splitting cells on edge will be on the Voronoi edges when \(\varepsilon\) approach 0.

In a full fluid mesh, splitting should be performed on each cells until threshold reached, then the approximated Voronoi diagram for the whole surface will be generated. A more complicated case is presented in Figure 5 which shows the structure points, fluid mesh before and after splitting, and approximated Voronoi diagram. The proposed splitting method is quite friendly for parallel computing as the splitting operation on one cell is irrelevant to other cells. The splitting procedure is also available on other type of mesh or even hybrid mesh. For triangular cells, the splitting cells can be generated by connect the middle point of each edge. For polygon cells, triangular cells will be generated first by connect each vertex and the geometry center then split these triangular cells.

After the splitting procedure, the loads on structure point will be naturally calculated by integration. Suppose we have \(m\) fluid cells in \(G_f\) and \(n\) structure points in \(P_s\). Each cell in \(G_f\) with pressure vector \(\vec{\sigma}_i\) and viscous stress vector \(\vec{\tau}_i\) will be split into \(m_i\) sub cells. The area of the \(j\)th sub cell in these \(m_i\) sub cells is denoted as \(a_{ij}\) and the directional vector from reference point to its geometry center is defined as \(\vec{r}_{ij}\). Normally fluid side is solved by finite volume method, the pressure and viscous stress is averaged, so the values on sub cells are the same to their parent cell. The total loads from the fluid side on boundary, including force and moment is discretized as

\[
F_f = \int_{\Omega} f \, dx = \sum_{i=1}^{m} \sum_{j=1}^{m_i} (\vec{\sigma}_{ij} + \vec{\tau}_{ij}) a_{ij} \\
M_f = \int_{\Omega} f \times r \, dx = \sum_{i=1}^{m} \sum_{j=1}^{m_i} (\vec{\sigma}_{ij} + \vec{\tau}_{ij}) a_{ij} \times \vec{r}_{ij}
\]

(4)
While on the structure side, for each structure point \( p_i \), an approximated Voronoi cell combined by split fluid cell is created and contain \( n_i \) sub cells.

\[
F_s = \int_\Omega f \, dx = \sum_{i=1}^{n} \sum_{j=1}^{m} (\bar{a}_{ij} + \bar{r}_{ij}) a_{ij}
\]

(6)

\[
M_s = \int_\Omega f \times r \, dx = \sum_{i=1}^{n} \sum_{j=1}^{m} (\bar{a}_{ij} + \bar{r}_{ij}) a_{ij} \times r_{ij}
\]

(7)

Because the \( \sum_{i=1}^{n} n_i = \sum_{i=1}^{m} m_i \) and every sub cell are covered in integration, the integral of force and moment on both side are the same. So the conservation of total force and moment \( F_f = F_s \) and \( M_f = M_s \) have been satisfied. The conservation of distribution is not strictly satisfied due to the grain size of the minimum split cells. This will be discussed in next section.

In practice, some finite element do not accept moment loads. So the moment on one structure point received form sub cells of fluid mesh need to be convert to concentrated force. For example in commercial finite element solver such as NASTRAN and ANSYS, multi point constraints (MPC) are used to convert moment to force.

4. Test Cases

The present load transfer method and the loosely coupled CFD/CSD strategy have been applied to the benchmark cases of aeroelastic analysis and the results were discussed as follows.

Agard445.6 is a standard aeroelastic test configuration for flutter prediction. The experimental results quoted in this paper are from tests conducted in NASA Langley Wind Tunnel[22]. However, static aeroelastic analysis were conducted on this configuration in reference[23], which is regarded as a benchmark of numerical aeroelastic analysis. The Agard445.6 is a 0.762m-wide half model with a 45° sweep angle at quarter chord line. The root chord and tip chord are 0.558m and 0.368m respectively. The wing section alone stream-wise direction is made up of NACA 65A004 airfoil. The aspect ratio is 1.65 and the taper ratio is 0.66. The structure used in aeroelastic tests is weaken by drilling holes on wing structure, and the holes are filled with plastic foam to obtain the flutter conditions. A weakened wing structure is modeled by quadrilateral shell elements and the thickness of shell is defined as the thickness of wing configuration. The finite element model of the structure is made of orthotropic material and the parameter of which is shown in Table1. The fluid field is discretized into 22 blocks, totally 200 million elements. The test case description is listed in table 2. The coupling surface on both fluid and structure are shown in Figure 4.

The load transfer results of RBF(Radial Basis Function) interpolation in the same case are also collected for comparison and validation, as is shown in Figure 5. Since the deformation of the wing structure is determined by the force and the position where it exerts, a local coordinate system has been established and the moment along span-wise direction on both fluid mesh and structure has been obtained.

| Table 1. Material Parameter | Parameter | Value |
|----------------------------|-----------|-------|
| Longitudinal Modulus (GPa) | 3.151     |
| Transverse Modulus(GPa)    | 0.4162    |
| Modulus of Rigidity(GPa)   | 0.4392    |
| Poisson Ratio              | 0.31      |
| Density (Kg/m³)            | 381.98    |

| Table 2. AGARD Test Case Description | Parameter | Value |
|--------------------------------------|-----------|-------|
| Mach Number                          | 0.85      |
| AoA                                  | 5°        |
| Reynolds Number                      | 5.6 × 10⁵ |
| q(kPa)                               | 3.073     |
The equilibrium position and pressure coefficient distribution is shown in Figure 6. It can be found that when the structure reaches equilibrium state, the wing appears bending upward along spanwise direction. Figure 7 compares the bending displacement along spanwise direction. As can be seen, the trend of displacement along spanwise direction of both leading and trailing edge are in consistent with the results obtained by Cai et.al[24]. However, the deflection predicted in this paper is slightly larger. It is worth noting that the method used in this article is partitioned approach with loosely-coupled strategy which is based on steady Navier-Stokes Solver. In contrast, Cai uses a strongly-coupled strategy and the flow field is simulated by unsteady Navier-Stokes Solver. The pressure distribution at 34% and 67%, respectively, are obtained on both rigid and elastic configuration, as is shown in Figure 8.
5. Conclusions

A pixelation method is presented and it has been implemented to real world cases as a subroutine in CFD/CSD program. The pixelation method splits the fluid cells by generating an approximation of Voronoi diagram on fluid mesh and then it integrates pressure and viscous force in all fluid cells to accumulate the aerodynamic load. Comparing to RBF interpolation, the pixelation method is more robust and efficient because the RBF interpolation involves solving inverse matrix which is expensive in terms of time consumption in practical problems. Moreover, interpolation-based data transfer algorithm is not strictly conservative which is crucial for repeated data transfer of large spatial gradient. The method presented in this paper is proved to be conservative, robust and efficient for fluid structure interaction problems.

References

[1] E. Livne, “Future of Airplane Aeroelasticity,” *J. Aircr.*, vol. 40, no. 6, pp. 1066–1092, 2003, doi: 10.2514/2.7218.
[2] J. W. Edwards, “Computational aeroelasticity,” 1993.
[3] A. Beckert, “Coupling fluid (CFD) and structural (FE) models using finite interpolation elements,” *Aerosp. Sci. Technol.*, vol. 4, no. 1, pp. 13–22, 2000.
[4] J. R. Cebral and R. Lohner, “Conservative load projection and tracking for fluid-structure problems,” *AIAA J.*, vol. 35, no. 4, pp. 687–692, 1997.
[5] R. L. Harder and R. N. Desmarais, “Interpolation using surface splines,” *J. Aircr.*, vol. 9, no. 2, pp. 189–191, Feb. 1972, doi: 10.2514/3.44330.
[6] M. Smith, C. Cesnik, D. Hodges, and K. Moran, “An evaluation of computational algorithms to interface between CFD and CSD methodologies,” in 37th Structure, Structural Dynamics and Materials Conference, 1996, p. 1400.

[7] G. S. L. Goura, K. J. Badcock, M. A. Woodgate, and B. E. Richards, “Transformation methods for the time marching analysis of flutter,” AIAA Pap., vol. 2457, p. 2001, 2001.

[8] C. Farhat, M. Lesoinne, and P. LeTallec, “Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity,” Comput. Methods Appl. Mech. Eng., vol. 157, no. 1–2, pp. 95–114, 1998, doi: 10.1016/S0045-7825(97)00216-8.

[9] R. K. Jaiman, X. Jiao, P. H. Geubelle, and E. Loth, “Conservative load transfer along curved fluid-solid interface with non-matching meshes,” J. Comput. Phys., vol. 218, no. 1, pp. 372–397, 2006, doi: 10.1016/j.jcp.2006.02.016.

[10] X. Jiao and M. T. Heath, “Overlaying Surface Meshes, Part I: Algorithms,” Int. J. Comput. Geom. Appl., vol. 14, no. 06, pp. 379–402, 2004, doi: 10.1142/S021819590401512.

[11] X. Jiao and M. T. Heath, “Overlaying Surface Meshes, Part II,” Int. J. Comput. Geom. Appl., vol. 14, no. 06, pp. 379–402, 2004, doi: 10.1142/S021819590401512.

[12] X. Jiao and M. T. Heath, “Common-refinement-based data transfer between non-matching meshes in multiphysics simulations,” Int. J. Numer. Methods Eng., vol. 61, no. 14, pp. 2402–2427, 2004, doi: 10.1002/nme.1147.

[13] M. D. Buhmann, “New developments in the theory of radial basis function interpolation,” Multivar. Approx. From CAGD to Wavelets, K. Jetter FI Utreras, eds., World Sci., pp. 35–75, 1993.

[14] M. J. Smith, D. H. Hodges, and C. E. S. Cesnik, “Evaluation of Computational Algorithms Suitable for Fluid-Structure Interactions,” J. Aircr., vol. 37, no. 2, pp. 282–294, Mar. 2000, doi: 10.2514/2.2592.

[15] K. Y. Dai, G. R. Liu, K. M. Lim, and Y. T. Gu, “Comparison between the radial point interpolation and the Kriging interpolation used in meshfree methods,” Comput. Mech., vol. 32, no. 3–4, pp. 39–70, Sep. 2003, doi: 10.1007/s00466-003-0462-z.

[16] X. Wang and Wenbin Song, “A new hybrid interpolation method for information exchange in computational aeroelasticity,” in 2010 3rd International Symposium on Systems and Control in Aeronautics and Astronautics, Jun. 2010, pp. 439–444, doi: 10.1109/ISSCAA.2010.5633103.

[17] P. Chen and I. Jadic, “Interfacing of Fluid and Structural Models via Innovative Structural Boundary Element Method,” AIAA J., vol. 36, no. 2, pp. 282–287, 1998, doi: 10.2514/2.7513.

[18] H. Wendland, “Fast evaluation of radial basis functions: Methods based on partition of unity,” 2002.

[19] A. de Boer, M. S. van der Schoot, and H. Bijl, “Mesh deformation based on radial basis function interpolation,” Comput. Struct., vol. 85, no. 11–14, pp. 784–795, 2007, doi: 10.1016/j.compstruc.2007.01.013.

[20] T. C. S. Rendall and C. B. Allen, “Efficient mesh motion using radial basis functions with data reduction algorithms,” J. Comput. Phys., vol. 228, no. 17, pp. 6231–6249, 2009, doi: 10.1016/j.jcp.2009.05.013.

[21] F. Aurenhammer, “Voronoi diagrams,” ACM Comput. Surv., vol. 23, no. 3, p. 94, 1991, doi: 10.1145/1447-0756.2010.01436.x.

[22] E. LEE-RAUSCH and J. BATINA, “Calculation of AGARD wing 445.6 flutter using Navier-Stokes aerodynamics,” in 11th Applied Aerodynamics Conference, 1993, p. 3476.

[23] E. Baškut and A. Akgül, “Development of a coupling procedure for static aeroelastic analyses,” Sci. Tech. Rev., vol. 61, no. 3–4, pp. 39–48, 2011.

[24] J. Cai, F. Liu, H. Tsai, and A. Wong, “Static aero-elastic computation with a coupled CFD and CSD method,” in 39th Aerospace Sciences Meeting and Exhibit, 2000, p. 717.