Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2005
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

A CONSERVATIVE MESH-FREE APPROACH FOR FLUID-STRUCTURE INTERFACE PROBLEMS

Giuseppe Quaranta†, Pierangelo Masarati† and Paolo Mantegazza†

†Dipartimento di Ingegneria Aerospaziale
Politecnico di Milano, Campus La Masa
via La Masa 34, 20156 Milano, Italy
e-mail: giuseppe.quaranta@aero.polimi.it- Web page: http://www.aero.polimi.it

Key words: Fluid-Structure Interactions, Interface, Moving Least Square, Aircrafts

Abstract. *The problem of building a methodology to couple fluid and structure is addressed in this paper. If the goal of the fluid-structure simulation is the stability assessment, an important issue to be addressed is the conservation of the momentum and energy exchanged between the participating domains. In realistic aerospace applications, the fluid and structural meshes are not compatible, either because discretized with major geometrical and topological discrepancies, or because different resolution requirements are set. As a consequence, different strategies may be required to solve the interface problem in each specific case. A new scheme is presented, based on the Moving Least Square (MLS) patches technique used in the field of surface reconstruction from N irregularly distributed given data points. The MLS method represents the correct framework which can be easily adapted to different interface problems. The goal is to develop an accurate but flexible method, capable of interfacing arbitrary structural or aerodynamic discretization schemes. A complete independence from the formulation of the structural and aerodynamic solvers is sought, while the conservation properties are retained as far as possible. A few applications are shown to highlight the quality of the proposed scheme.*

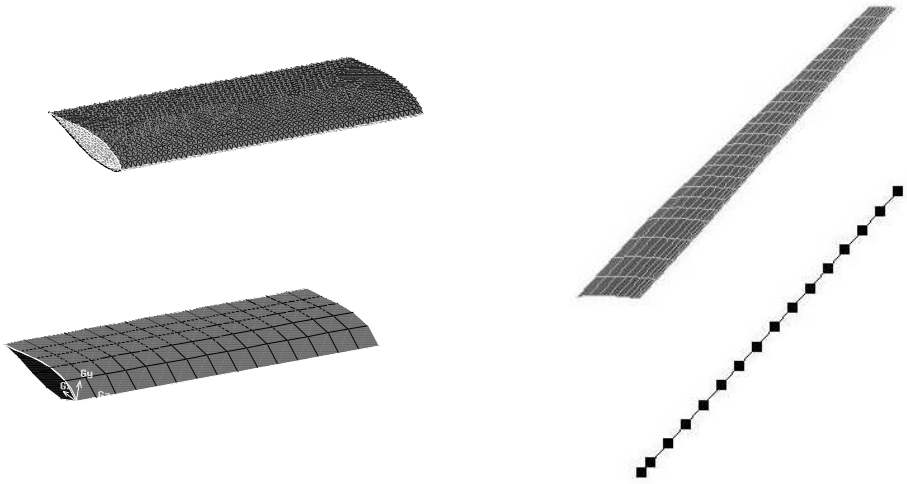
1 INTRODUCTION

The interaction between a deformable structure and the fluid surrounding it gives rise to a rich variety of physical phenomena which may be of primary importance in many fields of engineering. In the aerospace industry, aeroelastic instability occurrences must be carefully analyzed during the design process of a new aircraft. They me be either of a dynamic nature, such as flutter or buffet, or static, as divergence or control surface reversal. However, stability is not the only concern of aeroelastic analyses. The structural deformation of an elastic vehicle under static or dynamic loads may be of primary importance in the evaluation of performances, expecially in the era of ever-lighter, thinner, and more flexible structures. Furthermore, aeroelastic constraints are more and

more accounted for during the preliminary design process, and aeroelastic phenomena are even seen as a resource to improve the performances of the whole system (see for instance Reference[1]). Consequently, the need to have accurate, but efficient, methodologies to assess the aeroelastic behavior is growing. When complex vortical and/or transonic fight regimes need to be analyzed, the classical linearized model used for the investigation of aerodynamic loads must be dropped in favor of higher fidelity equations such as Euler or Navier-Stokes models. Of course, aerospace is not the only field where the investigation of accurate Fluid-Structure Interactions (FSI) may have a great impact: the flow of blood through arteries, or more in general the flow of fluids through pipes, the stability of bridges and tall buildings subjected to winds, the vibration of turbine and compressor blades, the oscillation of heat exchangers, the behavior of MEMS, are all examples of systems that may require this kind of analyses.

Given a fluid and structural system, in principle one can directly write down the system of Partial Differential Equations (PDEs) that governs the coupled FSI problem, and then try to discretize it as a whole. However, this approach, usually denominated "monolithic" is hardly ever used,[2] because each numerical model for each physics has different mathematical and numerical properties, and quite peculiar software implementation requirements. The other approach, called "partitioned" and followed in usual practice, is based on the Domain Decomposition paradigm. The basic idea is to develop specialized methods to solve each field independently, or better to resort to existing, well-established numerical techniques for each discipline. To solve a coupled fluid structure problem it is not sufficient to be able to compute the solution of the structural and of the aerodynamic model separately, it is also necessary to exchange information between the two models: the modification of boundary conditions must be transferred from the deformable structure to the aerodynamic boundary, and conversely, the loads developed in the aerodynamic field must be applied to the discretized structural model. The way of using the information exchange during a time integration or a frequency domain simulation gives rise to further distinctions in the class of partitioned methods among explicit or implicit coupling, "loose" or "strong" coupling.

In addition, to exploiting the great experience gained in the development of solvers for each field, the partitioned approach allows another important freedom: the possibility to use "non-compatible" discretizations, such as, for example, different discretization size and element type, as in Fig. 1(a), or beam elements for structures and flat lifting surfaces for aerodynamics, Fig. 1(b). Of course in all these cases the responsibility of correctly transferring information is left to an "interface scheme", which must interpolate/extrapolate the data in an appropriate manner. The problem of building a methodology to couple fluid and structure has always been a key aspect of aeroelastic methods. Low fidelity approaches, especially for planar configurations, are well established in the aeroelastic community and are still adopted in most of the industrial codes of current use.[3] However, these approaches may not be correct when complex interactions appear, because they can introduce oscillations or discontinuities, especially in regions of high curvature such as



(a) Non-matching surface grids. (b) Beam-lifting surface interface.

Figure 1: Possible fluid-structure interfaces.

wing leading edges or wing-fuselage intersections. In this case a higher fidelity may be necessary to reliably capture the physical mechanism that gives rise to important phenomena. Furthermore, the possibility of large displacements challenges the robustness of classical interface methods. Excellent reviews of methods implemented for the information transfer in FSI problems are published by Hounjet and Meijer,[4] Smith et al.[5] and by Guruswamy.[6] The literature on this subject is made by a large number of contributions. In particular Farhat et al.,[7] Cabral and Löhner[8] and Jiao and Heath[9] raise the important issue of conservation of the exchanged momentum and energy between the two grids. It is worth stressing that when non compatible topologies are used, the exact definition of what strict conservation means may be a difficult job.[9] This incompatibility condition is extremely frequent when aircraft structures are analyzed, since usually only the wing box, with spars ribs and the skin panels, is modeled by means of a hybrid system made by connecting 1-D, 2-D and sometimes 3-D finite elements types. The use of discretizations with a topological incompatibility (Fig. 1(b)) has another drawback, because it often requires extrapolation, which may be quite dangerous in terms of generated errors. In some cases instead of adopting a *pointwise methods*, which use only the data at specified nodes, so they do not need topological information, the interface exchange algorithm requires the definition of a *virtual interface* surface. Projecting the nodes of each grid on this interface allows to exploit interpolation methods which use the fact that all entities now belong to the same topological surface. Of course, especially when large topological discrepancies are present, the automation of the projection phase on the virtual surface may be not a trivial task, and often requires engineering judgment. Beckert[10] for example uses this kind

of techniques. A novel method, different from all the above mentioned ones, is presented by Chen and Jadic;[11] it uses an analogy with elastic deformation problems.

Most of the techniques are advantageous in some cases but may not always be the best choice. The present work tries to exploit another strategy which can be sufficiently flexible to include different advantages of the known methods in a wide range of possible coupling schemes. The definition of the characteristics of a method starts from the indication of the main properties that must be ideally satisfied by an interface algorithm:

- the possibility to interface both non-matching surfaces or non-matching topologies;
- the capability to deal with situations when a control point fall outside the range of the source mesh (extrapolation);
- the computational efficiency of the interface operator computation and use;
- the exact treatment of rigid translations and rotations;
- the capability to deal correctly in situations where there is a wide variation of the node density of the source mesh;
- the independence from the numerical formulation of the Computational Fluid Dynamics (CFD) and Computational Structural Dynamics (CSD) solvers;
- the conservation of the exchanged quantities (in particular momentum and energy);
- the possibility to control the smoothness of the resulting surface;
- easy evaluation of the new interface operator in case of mesh adaption either in the source or target grids.

Using this list as a guide, this work tries to develop an accurate method, capable of interfacing any possible structural and aerodynamic discretization scheme. The basic technique is inferred from the field of construction of surfaces from N irregularly distributed given data, and corresponds to the well known Moving Least Square (MLS). The application of compact weight function allows to define patches which localize the computed interface operator. A complete independence from the formulation of the structural system is achieved, while retaining the conservation properties. A similar approach has been followed by Beckert and Wendland[12] which use an interpolation technique based on the usage of Radial Basis Functions (RBF). The paper presents the details of the technique both in terms of algorithm derivation ad implementation details. Various applicative examples are reported to highlight the high flexibility that may be achieved.

2 PROBLEM STATEMENT

Without any loss in generality, we assume that the fluid dynamics problem is stated using an Arbitrary Lagrangian Eulerian (ALE) formulation to represent also the possible movement of the fluid mesh caused by the structure displacements. The integral form of the compressible unsteady Euler equations for volumes subjected to deformation is

$$\frac{d}{dt}\int_{V(t)} \mathbf{u} dV = -\int_{\partial V(t)} (\mathbf{F} - \mathbf{u}\dot{\mathbf{x}}) \cdot d\mathbf{A}, \tag{1}$$

where $\mathbf{u} = (\rho, \rho u, \rho v, \rho w, E)^T$ are the usual conservative variables, $\mathbf{F}$ are the inviscid fluxes and $\dot{\mathbf{x}}$ is the velocity of the moving boundary of the mesh elements.[13] For the more general Navier–Stokes model the diffusive fluxes must be added. In any case any fluid dynamic model will compute the stresses applied on the wet surfaces, indicated as Γ, given the dynamic state of the boundaries. The discretized structural problem, on turn, is expressed in the most generic format as a system of Differential-Algebraic system of Equations (DAE), which include the exact representation of kinematic constraints,[14] here reported with the effect of the aerodynamic loads, $\mathbf{Q}_a$, highlighted

$$\begin{aligned} &\mathbf{M}(\mathbf{y}_s, t)\dot{\mathbf{y}}_s = \mathbf{z}, \\ &\dot{\mathbf{z}} = \mathbf{K}(\mathbf{y}_s, \dot{\mathbf{y}}_s, t) + \mathbf{Q}_a(\mathbf{y}_s, \dot{\mathbf{y}}_s, t) - \mathbf{G}^T\boldsymbol{\lambda}, \\ &\boldsymbol{\Psi}(\mathbf{y}_s, t) = 0. \end{aligned} \tag{2}$$

where $\mathbf{K}$ represents the structural elastic (or visco-elastic) operator which may, or may not, be linear, and $\boldsymbol{\lambda}$ is the vector of Lagrange's multipliers used to enforce the constraints. To complete the model, the coupling conditions must be added. If $\mathbf{y}_f$ denotes the fluid boundary movement, p its pressure, σ_s and σ_f respectively the structure stress tensor and the fluid viscous stress tensor, and $\mathbf{n}$ the normal vector to Γ, these conditions are

$$\left.\begin{aligned} &\sigma_s \cdot \mathbf{n} = -p\,\mathbf{n} + \sigma_f \cdot \mathbf{n} \\ &\mathbf{y}_s = \mathbf{y}_f \\ &\dot{\mathbf{y}}_s = \dot{\mathbf{y}}_f \end{aligned}\right\} \quad \text{on } \Gamma. \tag{3}$$

The first of Eqs. (3) expresses the dynamic equilibrium between the stresses on the structure with those on the fluid side. The other two are kinematic compatibility conditions. The last one, for inviscid flows, is replaced by the condition on normal velocity

$$\frac{\partial \dot{\mathbf{y}}_s}{\partial n} = \frac{\partial \dot{\mathbf{y}}_f}{\partial n}.$$

These conditions are valid regardless of the formulation used for the description of each field; aerodynamics can be approximated either by the Euler equations or any other CFD model, or by a simpler panel method. If a moving grid is considered, the conditions on the kinematics of the boundary must be reflected also on the moving grid, so $\mathbf{x} = \mathbf{y}_s$, $\dot{\mathbf{x}} = \dot{\mathbf{y}}_s$ on Γ.

2.1 Conservative methodologies

Eqs. (3) are valid for a continuous system. However, the two fields are solved resorting to a discrete approximation based on finite approximations of the functional space under which the solutions fall. As a consequence, the correct expression of Eqs. (3) for the discretized problem must be derived, under the constraint of retaining the conservation properties.[7] The change of energy in the fluid-structure system equals the energy supplied (or absorbed) by an external force. Consequently, at any time t, the energy released/absorbed from the structure, except for the contribution dissipated by the structural damping, must equal the energy absorbed/released by the fluid.

The conservation is trivially retained when the two computational domains of the fluid and structure have matching discrete interfaces and compatible approximation spaces, such as identical spaces for stress at either side of the interface. However, in realistic applications, the fluid and structural meshes are not compatible along the interface, because discretized with major geometric discrepancies (Fig. 1(a)-1(b)). In the following, Γ_f and Γ_s will respectively indicate the discrete fluid/structure non-matching representation of Γ. However, it must be noticed that in some cases the surface Γ_s may not have a clear physical representation (Fig.1(b)).

A classical scheme that can be used for non-matching interfaces (but matching topologies) is based on the use of structural shape functions N_j. In this case the nodal loads on node j induced by the fluid loads on element e will be equal to

$$\mathbf{f}_j^{(e)} = \int_{\Gamma_s^{(e)}} N_j \left(-p\,\mathbf{n} + \sigma_f \cdot \mathbf{n}\right) d\mathbf{A}. \tag{4}$$

The nodal loads $\mathbf{f}_j^{(e)}$ include both forces and moments. To evaluate the integral of Eq. (4), the Gauss quadrature rules may be used, so the pressure and viscous stresses at the Gauss point of each structural element must be computed. This can be done inside each element of the fluid grid using the characteristic shape functions used on that side of the interface. This method, which can be sufficiently accurate in many cases, cannot be labeled as conservative, because it does not guarantee that the sum of the discrete loads $\mathbf{f}_j^{(e)}$ is exactly equal to the fluid loads. In many cases this technique produces numerically consistent results, see for instance the test cases reported by Farhat et al.[15] However, there are cases in which the conservativeness of the interface may be a key issue to correctly assess the stability boundaries. Another problem of correctness of the results may arise when major geometric discrepancies are present, because the structural Gauss points may not belong to the aerodynamic interface, leading to incorrect evaluations. For a review of all the possible failures of this sort of procedure see Reference.[8]

In order to guarantee the conservation between the two models, the correct strategy would be to enforce the coupling conditions only in a weak sense, through the use of simple variation principles such as that of Virtual Work. Let $\delta\mathbf{y}_f$ and $\delta\mathbf{y}_s$ be two admissible virtual displacements for each field. "Admissible" means that the trace of these two fields

on Γ, which can be either a newly defined virtual interface surface or simply the surface of the fluid field Γ_f which is always present, must be equal

$$\mathrm{Tr}\,(\delta\mathbf{y}_f)|_\Gamma = \mathrm{Tr}\,(\delta\mathbf{y}_s)|_\Gamma. \tag{5}$$

For the inviscid case the condition becomes $\mathrm{Tr}\,(\delta\mathbf{y}_f \cdot \mathbf{n})|_\Gamma = \mathrm{Tr}\,(\delta\mathbf{y}_s \cdot \mathbf{n})|_\Gamma$. We used the trace Tr operator to stress the fact that in the most general case the effect of the node displacements on the Γ surface must be inferred.

Regardless of the method chosen to enforce the compatibility of the displacements, the relationship between the admissible virtual displacements can be written as

$$(\delta\mathbf{y}_f)_i = \sum_{j=1}^{j_s} h_{ij}\,(\delta\mathbf{y}_s)_j, \tag{6}$$

where $(\delta\mathbf{y}_f)_i$ are the discrete values of $\delta\mathbf{y}_f$ at the fluid nodes of the grid on the wet surface, and h_{ij} are the coefficients of the displacement interpolation matrix H. The resulting virtual displacement of the boundary surface will be

$$\delta\mathbf{y}_f = \sum_{i=1}^{i_f} N_i \sum_{j=1}^{j_s} h_{ij}\,(\delta\mathbf{y}_s)_j. \tag{7}$$

TheN_i are the base functions that belong to the approximation space of the aerodynamic field discretization, and i_f is the number of nodes that belong to the interface surface Γ_f. The virtual work of the aerodynamic load will be equal to

$$\begin{aligned} \delta W_f &= \int_{\Gamma_f} (-p\,\mathbf{n} + \sigma_f \cdot \mathbf{n}) \cdot \delta\mathbf{y}_f \, d\mathbf{A}, \\ &= \int_{\Gamma_f} (-p\,\mathbf{n} + \sigma_f \cdot \mathbf{n}) \cdot \sum_{i=1}^{i_f} N_i \sum_{j=1}^{j_s} h_{ij}\,(\delta\mathbf{y}_s)_j \, d\mathbf{A}. \end{aligned} \tag{8}$$

The virtual work of the forces and moments acting on Γ_s is equal to

$$\delta W_s = \sum_{j=1}^{j_s} \mathbf{f}_j \cdot (\delta\mathbf{y}_s)_j. \tag{9}$$

After imposing the equality of the virtual works, and by calling $\mathbf{F}_i$ the quantity given by

$$\mathbf{F}_i = \int_{\Gamma_f} (-p\,\mathbf{n} + \sigma_f \cdot \mathbf{n}) N_i \, d\mathbf{A}, \tag{10}$$

the nodal loads applied to the structure are

$$\mathbf{f}_j = \sum_{i=1}^{i_f} \mathbf{F}_i \, h_{ij}. \tag{11}$$

Eq. (11) states that, after computing the nodal loads for the aerodynamic boundary grid points using the correct approximation space, $\mathbf{F}_i$, the loads on the structural nodes, $\mathbf{f}_j$, can be obtained by simply multiplying $\mathbf{F}_i$ times the transpose of the interpolation matrix H that connects the two grid displacements. This is a well known result reported in almost all the works about the implementation of an interface algorithm, that ensure the balance of the energy exchanged between the fluid and the structure. However, this point does not completely resolve the question of conservation, because there is no conservation of the velocity transmitted from the structure to fluid boundaries, so no insurance about the conservation of the momentum transferred to the fluid. The problem of conservation is now shifted on the definition of the correct interpolation matrix H.

3 MOVING LEAST SQUARE

To build a conservative interpolation matrix which enforces the compatibility, Eq. (5), a weak/variational formulation can be used. The idea is to express the problem as a weighted least-square problem

$$\text{Minimize} \int_\Gamma \phi \left(\text{Tr}\left(\delta\mathbf{y}_f\right)|_\Gamma - \text{Tr}\left(\delta\mathbf{y}_s\right)|_\Gamma\right)^2 d\mathbf{A}. \tag{12}$$

In addition to this, additional properties can be sought, like smoothness of the resulting interpolated field, computational efficiency and some control on the interpolation error. A solution which possesses all these qualities can be obtained using the MLS technique. The origin of this approximation is connected to the surface or data reconstruction field (see Lancaster and Salkauskas,[16] and Schaback[17]). Recently, this type of approximation attracted the researchers' attention also for mesh-less methods for the numerical solution of PDEs.[18]

3.1 Mesh-less local reconstruction of surfaces

The problem can be mathematically stated as follows. Given a compact space $\Omega \subseteq \mathbb{R}^n$, the object of the analysis is the reconstruction of a function $f \in C^d(\Omega)$ from its values $f(\bar{\mathbf{x}}_1), f(\bar{\mathbf{x}}_2), \ldots, f(\bar{\mathbf{x}}_N)$ on scattered distinct centers $X = \{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_N\}$. Of course, it is not necessary to derive an analytical expression for f; it is sufficient to have an efficient method to compute the value of f on a different set of centers $Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$. The method should ideally have these properties: a) computational efficiency; b) correct smoothness of the resulting surface; c) quality of reproduction.

The first step is to build a local approximation of f as a sum of monomial basis functions $p_i(\mathbf{x}) \in \mathbb{P}_d$

$$\hat{f} = \sum_{i=1}^{m} p_i(\mathbf{x}) a_i(\mathbf{x}) \equiv \mathbf{p}^T(x)\, \mathbf{a}(x), \tag{13}$$

where m is the number of basis functions, and $a_i(\mathbf{x})$ are their coefficients. $\mathbb{P}_d \subseteq C^d(\Omega)$ is a finite dimensional space of basis functions; usually it is spanned by polynomials, but

other forms may be adopted. If a local singular function needs to be exactly reproduced by MLS, a singular basis function may be used.[19] In this case the adopted basis functions are either linear polynomials

$$\mathbf{p}^T(\mathbf{x}) = (1, x, y, z)^T \qquad \in C^1(\mathbb{R}^3), \tag{14}$$

or quadratic polynomials

$$\mathbf{p}^T(\mathbf{x}) = (1, x, y, z, x^2, xy, y^2, yz, z^2, zx)^T \qquad \in C^2(\mathbb{R}^3). \tag{15}$$

The coefficients $\mathbf{a}_i(\mathbf{x})$ are obtained by performing a weighted least square fit for the approximation

$$\text{Minimize } J(\mathbf{x}) = \int_\Omega \phi(\mathbf{x} - \bar{\mathbf{x}}) \left(\hat{f} - f(\bar{\mathbf{x}})\right)^2 \, d\Omega(\bar{\mathbf{x}}) \tag{16}$$

under the linear constraint

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{m} p_i(\bar{\mathbf{x}}) a_i(\mathbf{x}). \tag{17}$$

This equation is completely equivalent to Eq. (12) which expresses the interface problem. By substituting Eq. (17) in Eq. (16), the minimization gives

$$0 = \frac{\partial J(\mathbf{x})}{\partial a_i(\mathbf{x})} = 2 \int_\Omega \phi(\mathbf{x} - \bar{\mathbf{x}}) \left(\sum_{j=1}^{m} p_j(\bar{\mathbf{x}}) a_j(\mathbf{x}) - f(\bar{\mathbf{x}}) \right) \sum_{i=1}^{m} p_i(\bar{\mathbf{x}}) \delta a_i(\mathbf{x}) \, d\Omega(\bar{\mathbf{x}}), \tag{18}$$

which must vanish for any δa_j. After defining

$$A_{ij} = \int_\Omega \phi(\mathbf{x} - \bar{\mathbf{x}}) p_j(\bar{\mathbf{x}}) p_i(\bar{\mathbf{x}}) \, d\Omega(\bar{\mathbf{x}}), \tag{19}$$

it follows that

$$\sum_{j=1}^{m} A_{ij} a_j(\mathbf{x}) = \int_\Omega \phi(\mathbf{x} - \bar{\mathbf{x}}) p_i(\bar{\mathbf{x}}) f(\bar{\mathbf{x}}) \, d\Omega(\bar{\mathbf{x}}), \tag{20}$$

which gives the coefficients a_i. If we are dealing with a case of non-matching meshes that are distributed over the same topological surface Γ, it is possible to set $\Omega = \Gamma$, call x and y a set of curvilinear coordinates defined on Γ and integrate the expressions Eq. (19) and Eq. (20). However, as stated before, this is hardly ever the case in aeroelastic cases, so we need to solve the case of only a finite set of points X in a compact set Ω which may be the complete space $\mathbb{R}^3$. Consequently the integrals of the previous formulæ are transformed in sums on the set X. Eq. 16 becomes

$$J(\mathbf{x}) = (\mathbf{P}\,\mathbf{a}(\mathbf{x}) - \mathbf{f})^T \Phi(\mathbf{x}) (\mathbf{P}\,\mathbf{a}(\mathbf{x}) - \mathbf{f}), \tag{21}$$

where

$$\mathbf{f} = \left[f(\bar{\mathbf{x}}_1), f(\bar{\mathbf{x}}_2), \ldots, f(\bar{\mathbf{x}}_N) \right]^T,$$

$$\mathbf{P} = \begin{bmatrix} p_1(\bar{\mathbf{x}}_1) & p_2(\bar{\mathbf{x}}_1) & \dots & p_m(\bar{\mathbf{x}}_1) \\ p_1(\bar{\mathbf{x}}_2) & p_2(\bar{\mathbf{x}}_2) & \dots & p_m(\bar{\mathbf{x}}_2) \\ \vdots & \vdots & \ddots & \vdots \\ p_1(\bar{\mathbf{x}}_N) & p_2(\bar{\mathbf{x}}_N) & \dots & p_m(\bar{\mathbf{x}}_N) \end{bmatrix},$$

and

$$\mathbf{\Phi}(\mathbf{x}) = \begin{bmatrix} \phi(\mathbf{x} - \bar{\mathbf{x}}_1) & 0 & \dots & 0 \\ 0 & \phi(\mathbf{x} - \bar{\mathbf{x}}_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \phi(\mathbf{x} - \bar{\mathbf{x}}_N) \end{bmatrix}.$$

After solving the minimization problem, the coefficients result in

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{b}(\mathbf{x})\mathbf{f}, \tag{22}$$

where now $\mathbf{A}(\mathbf{x}) = \mathbf{P}^T\mathbf{\Phi}(\mathbf{x})\mathbf{P}$, and $\mathbf{b}(\mathbf{x}) = \mathbf{P}^T\mathbf{\Phi}(\mathbf{x})$. The resulting approximation function $\hat{f}$ can then be expressed as

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{N} \psi_j^d(\mathbf{x}) f(\bar{\mathbf{x}}_j). \tag{23}$$

$\psi_j^d(\mathbf{x})$ are called shape functions

$$\boldsymbol{\psi}^d(\mathbf{x}) = \mathbf{p}(\mathbf{x})^T \, \mathbf{A}(\mathbf{x})^{-1} \, \mathbf{b}(\mathbf{x}) \tag{24}$$

The superscript expresses the order of the polynomial basis functions used. In this case, in a strictly mathematical sense, the conservation is not satisfied, because is extremely difficult to define the correct set under which the integration must be accomplished. However, the result is obtained through a minimization process which ensure a sort of conservation given the limited amount of information that can be transmitted. It must be noted that Eq. (23) generally is not a simple interpolation of the function values for the given centers. An interpolation can be obtained only if weight functions with a singularity at the diagonal are used, so that $\phi(0) = 0$, a choice which is rather unusual. This is a result of the fact that we started from a conservation principle, so the point values must be viewed as integral quantities that are distributed on a different set of nodal points. Only in case of perfect coincidence of the two node sets the function is transferred maintaining exactly the same values.

The minimization problem has a unique solution if the symmetric matrix A is positive definite. A set

$$\mathcal{X}(\mathbf{x}) := \{\bar{\mathbf{x}}_j \in \mathcal{X} \ : \ \phi(\|\mathbf{x} - \bar{\mathbf{x}}_j\|) > 0\}, \tag{25}$$

is defined; it can be viewed as the "discretized support" of the weight functions. If there is no trivial base function that vanishes on $\mathcal{X}(\mathbf{x})$, if the weight functions are always positive

inside their support, and if the support is large enough to host at least a set of points near $\mathbf{x}$ that are in an arbitrary position with respect to the basis functions, then matrix A has full rank and the problem is solvable, (see Wendland[20] for a rigorous mathematical description of the convergence orders of MLS). Singularities in matrix A correspond to cases for which the interface is not a well-posed problem, such as the case of aligned nodes in the local support. It is also easy to show that the technique is "consistent", meaning that any function which appears in the basis can be reproduced exactly.[19]

The great advantage of the problem expressed in this form is that it can be localized by choosing compact support weight functions such as smooth nonnegative Radial Basis Functions(RBF).

3.2 RBF functions

The necessary weight functions are supposed to satisfy the following requirements:

1. ϕ is a monotonically decreasing function of a scalar parameter that represents the Euclidean distance, $r = \|\mathbf{x}\|_2$, so it has radial symmetry;

2. ϕ has a compact support, so

$$\phi(r) \begin{cases} > 0 & \text{if } r < \hat{r} \\ = 0 & \text{if } r \leq \hat{r} \end{cases};$$

3. a normality property

$$\int_\Omega \phi(\mathbf{x} - \bar{\mathbf{x}}) d\Omega = 1$$

is defined.

Functions of this form are called RBF. A comprehensive presentation of RBF and their properties can be found in Reference,[21] while a review of the numerical techniques based on RBF is in Reference.[22] The most popular RBFs, like Gaussian, thin-plate-spline, multiquadratics, are not compactly supported[‡]. Recently Wu[23] and Wendland[24] constructed smooth compactly supported RBFs. These functions, for a specific space dimension d, possess the lowest possible degree among all piecewise polynomial compactly supported radial functions which are positive definite on $\mathbb{R}^d$ and of a given order of smoothness, which is a useful property in practice. These functions have a dependence on the space dimension d; for three-dimensional cases,[24] proposes (see Fig. 2)

$$\phi^0(r) = (1-r)_+^2 \qquad C^0 \tag{26a}$$

$$\phi^2(r) = (1-r)_+^4 (4r+1) \qquad C^2 \tag{26b}$$

[‡]Those functions are the basic ingredients of classical interface schemes.[5] For those cases, the resulting interface operator is a full matrix, which may not be the best choice in terms of efficiency of the scheme when a large number of nodes are involved.

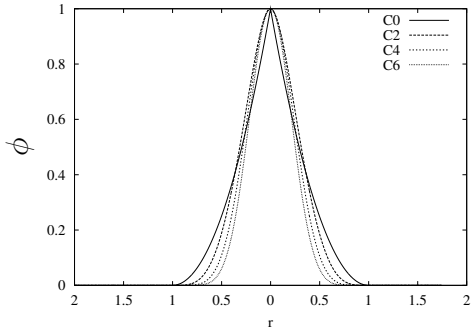


Figure 2: RBF sections with different degrees of smoothness.

$$\phi^4(r) = (1-r)_+^6 \left(\frac{35}{3}r^2 + \frac{18}{3}r + 1\right) \qquad C^4 \qquad (26c)$$

$$\phi^6(r) = (1-r)_+^8 (32r^3 + 25r^2 + 8r + 1) \qquad C^6 \qquad (26d)$$

The subscript $(\cdot)_+$ indicates that the function is not identically zero only if the argument inside the parentheses is greater or equal to zero. Eq. (23) implies that the approximant function $\hat{f}$ is as smooth as the involved weight functions. Indeed, the maximum number of continuous derivatives of $\hat{f}$ is bounded by the number of continuous derivatives of $\mathbf{a}(\mathbf{x})$, whose dependence on $\mathbf{x}$ is related to functions ϕ only.

The last problem, which greatly influences the efficency and quality of the MLS approximation, is the determination of the appropriate local support $\mathcal{X}$ dimensions. Since there may be great variations in the data density, a way to incorporate these variations into the algorithm in a flexible way must be sought. Usually the weight RBF are written as $\phi(r/\delta)$, where delta is a scaling factor that allows to change the function support at different space centers. The parameter δ allows the user to adapt the support radius to the problem, being sure that, on the one hand, enough points are covered, and, on the other hand, far away points have no influence.

3.3 Application to the fluid-structure interface problem

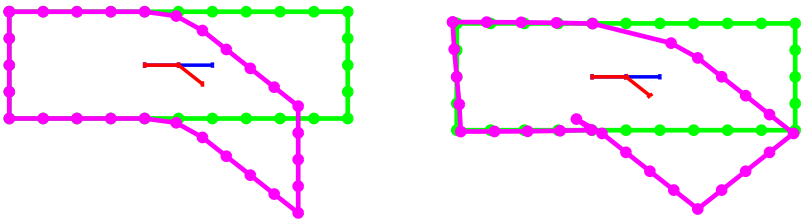
The application of the MLS technique to the fluid structure interface is straightforward. The interface problem, Eq. (12), can be seen as a MLS problem. The matrix H can be built regardless of the function to be approximated, which can be a scalar component of the position, displacement or velocity field. Each coefficient of the interface matrix, H_{ij}, contains the value of the shape function ψ_j^d evaluated at the aerodynamic node $\mathbf{y}_i$. If the aerodynamic node at $\mathbf{y}_i$ is outside the support of the weight function associated with the structural node $\bar{\mathbf{x}}_j$, the coefficient is null.

The first step is the determination of the local support dimensions and of the centers which belongs to each of them, *i.e.* $\mathcal{X}(\mathbf{y})$, which must be done in a preprocessing phase. The implementation proposed here uses the criterium of achieving a constant fill-in for each row of the interpolation matrix, trying to find the appropriate trade-off between

smoothness and efficiency of the interpolation process. To keep a constant fill-in means to set in advance the number of centers that belong to each local support, and to vary the local support radius accordingly. A larger radius/fill-in causes the generation of a smoother surface since it generates larger intersection areas among the different local supports. However, a smaller radius/fill-in reduces the computational burden related to the exchange of information at the interfaces, simultaneously leading to more stable approximation systems. The number of centers has a lower bound related to the polynomial order used for the basis functions. If linear polynomials are used, at least four nodes must be included in the support; for the quadratic case, the minimum is ten. Below this numbers the local system of equations is under-determined. The implemented code allows the user to choose the fill-in value for each case. Different values of the fill-in value can be imposed in different regions of the interface, or the local support function can be deformed to acquire a preferential direction. This allows to exactly enforce structural constraints on the aerodynamic mesh. If, for example, one side of the aerodynamic surface is clamped, the support functions used for the nodes on the clamped surface can be modified in order to be influenced by structural clamped nodes only. The process of local support size definition and of search of the nodes that belong to each $\mathcal{X}(\mathbf{y})$ occurs once only. This type of choice for the local support size leads to ignore sorting and triangularization algorithms for the geometric preprocessing in favor of the usage of *k nearest neighbor algorithms*.[25] The underlying idea is to process the points of X into a data structure, such that, given any query point $\mathbf{y}$, the k nearest points of X to $\mathbf{y}$ can be efficiently looked-up. This allows to perform the geometric preprocessing at $O(N)$ cost when building the structure, and a bare $O(k)$ (in a quasi-uniform case, or $O(k \log N)$ in general) operations to get the nearest neighbor from the data set X.

The resulting interface matrix satisfies all the requirements. It is effective, in the sense that the major computational complexity, which lies in the geometric part of the computation, can be accomplished in an efficient manner; it can produce surfaces with any prescribed smoothness, and it has a high control on the behavior for each local region. The method presented is completely independent from the implementation details of the codes that are interfaced. There is no need to know the shape functions used in each field; as a consequence, the structural elements implementation or the code used to solve the aerodynamic field can be changed without affecting the interface code.

The last problem that requires special attention is related to the treatment of rotations, since we are dealing with structures that can experience large changes in orientation. The rotation information is often necessary to correctly infer the structural deformation behavior (see examples in Sec. 4.1). Rotations need to be accounted for, especially when a beam structural model is interfaced to an aerodynamic surface boundary. Rotations are complex tensorial entities that can be subjected to singularity problems when parameterized. To avoid the need to build a specialized class of interpolation matrices to deal with these entities, since the updated position of an aerodynamic surface node is obtained by combining the effect of linear displacements and rotations, a simple procedure has been



(a) No rotation of the extreme node

(b) Rotation of the extreme node

Figure 3: Large rotational displacement of 2-D structure.

developed. The idea is to add three rigid arms to each structural beam node, directed along the node local reference axes, creating a sort of "fish-bone" structure. The effect of the rotation is accounted for by the displacements of the added fictitious nodes at the end of each arm, and matrix H is computed using only spatial displacements. Of course, the length of the rigid arms needs to be smaller than the distance between two structural beam nodes h_s. As a rule of thumb, it must be lower that $1/5h_s$. A very small distance is in fact perfect because it gives almost the same value of the weight function to each node of the connected arms.

Using the transpose of matrix H, the loads computed at the aerodynamic nodes can be transferred to the structural mesh. The element integral of the pressure and viscous stresses must be computed at each node of the fluid mesh, using the shape functions applied in the aerodynamic discretization, which result in a force vector and eventually in a moment vector. All the forces and moments applied to the fictitious nodes must be transferred to the central beam node, exploiting the property that the fish-bone arms are rigidly connected to the beam node. This makes the interface method rigorously conservative, according to all the conditions expressed in the previous paragraphs.

4 APPLICATION EXAMPLES

In the following, few application examples of the interfacing technique are presented, to highlight the robustness, the flexibility and the high level of control the proposed interface code grants to the user.

4.1 Simple 2D example

The robustness and quality of reproduction of MLS method when a large number of centers is used is well documented in the literature.[17] There is less experience for cases when a small number of source centers are present. Figure 3 shows a schematic two-dimensional case, where a deformation is transmitted from a 9 nodes structural grid to a rectangular 28 nodes grid. The structural grid is made by three nodes plus symmetrically

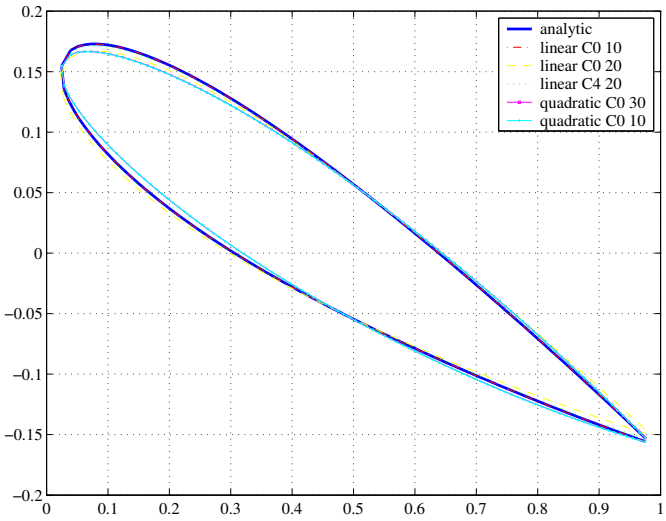


Figure 4: Helicopter blade linear twist: comparison of blade sections obtained with different interface schemes at station 9 m.

added appendices to exchange the rotation information, and to avoid any singularity due to node alignment. In both cases a 45 deg. rotation of the right arm is imposed. The left figure shows the deformed grid without any local rotation of the structural nodes (a pure shear deformation). The right figure shows the the behavior when we apply a rotation to the nodes appendices. Both cases are in principle valid deformations. Differently form what stated by Chen and Jadic,[11] is this information that is needed to discriminate between those two deformations.

4.2 Helicopter blade

In this example the behavior of the interface is analyzed for a simple straight, unswept helicopter blade with a NACA 0012 airfoil and an aspect ratio of 10. The structural model is made of five beam elements, while the aerodynamic surface is represented by a structured grid of 200×21 nodes. The fish-bone structure is obtained by adding four rigid arms at each node, a couple for each direction perpendicular to the beam axis. The two arms in each direction have opposite sign to obtain a final layout which respects the symmetry. In this way all the possible rotation of each node are correctly transmitted to the aerodynamic wet surface.

For the first test case, a linear twist has been applied along the blade, from 0 deg. at root to 20 deg. at tip. After computing the interface matrix, the beam node positions of the twisted blade are used to obtain the deformed surface. The six structural nodes are positioned at stations 0, 2, 4, 6, 8 and 10 m starting from the blade root, at the mid-chord points. Fig. 4 compares the analytical positions of the airfoil nodes on the deformed blade at 9 m from the blade root (90% of the blade span), with the corresponding results obtainable with different interfacing schemes. A numerical comparison is shown

Scheme	**Angle**	**% Error**
Analytic	18	0
Linear C^0 10 pts.	17.9886	0.06
Linear C^0 20 pts.	17.2094	4.40
Linear C^4 20 pts.	17.7743	1.25
Quadratic C^4 20 pts.	17.9880	0.06
Quadratic C^4 30 pts.	17.9905	0.05

Table 1: Helicopter blade linear twist: twist angle at station 9 m.

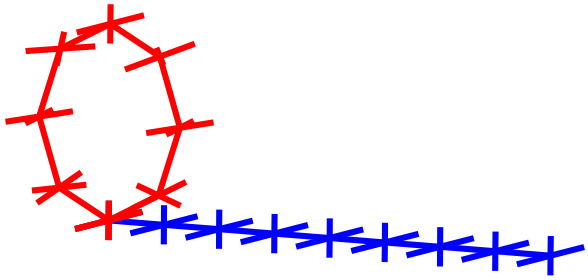


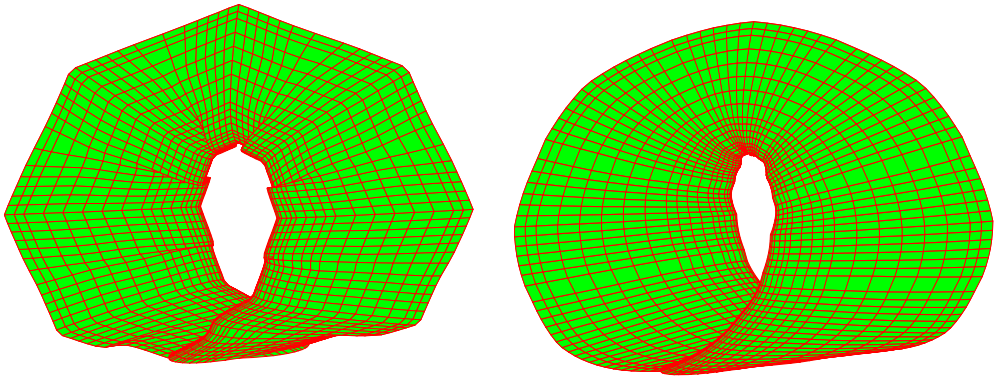
Figure 5: Large circular displacements: beam in the undeformed (blue) and deformed (red) position.

in Tab. 4.2, where the computed twist angle, obtained by using the deformed positions of the trailing and leading edge points, is compared for the different schemes. It is clear that the overall quality of the interfacing schemes is good. Increasing the number of points for the linear basis function causes a slight degradation of the quality, which can be partially recovered by raising the smoothness of the weight functions. The quadratic schemes show a high precision, but the rigidity of the airfoil section it is not perfectly preserved, as appears from Fig. 4, where the deformed quadratic curves cannot be obtained by a simple rotation of the analytic blade section.

To highlight the significance of the possibility to rule the surface smoothness, a case with very large displacements is investigated: the blade beam is bended to create a circle with a 180 deg. twist (Fig. 5). The interfaced displacement on the aerodynamic mesh obtained by using a linear basis with 10 points and C^4 weight functions is shown in Fig. 6(a). The result is extremely discontinuous and entangled, so that an high number of surface elements are invalid. Using a second order basis with 15 points and C^4 weight function a smooth blade surface is recovered.

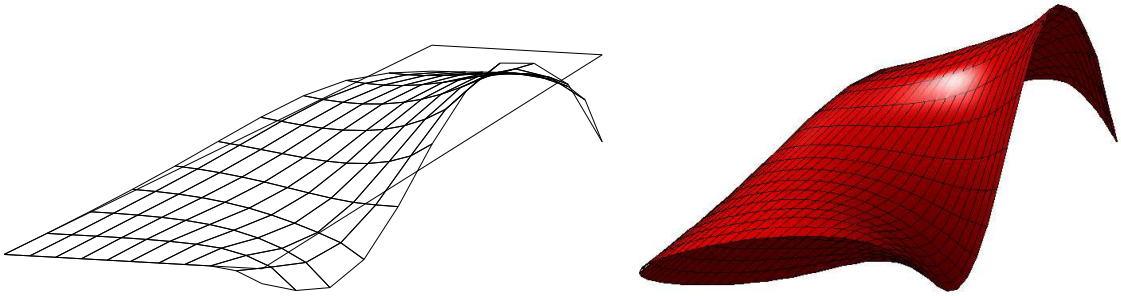
4.3 AGARD wing 445.6

The AGARD 445.6 wing,[26] a typical experimental test case for the evaluation of aeroelastic transonic computations. The same test case has been used by[5] to compare the behavior of different interface methods.



(a) Linear C^4 on 10 points (b) Quadratic C^4 on 15 points

Figure 6: NACA 0012: deformed blade in a circular manner.



(a) Structural grid. (b) Aerodynamic grid, expanded on the z axis.

Figure 7: Mode 5 deformed grid for AGARD 445.6 wing.

In this case the structure is represented by a flat plate that extends from the wing leading to the trailing edge from the wing root to the wing tip. The mesh is evenly spaced, with 11 nodes streamwise and 11 in the spanwise direction. The deformation transfer from the structural to the aerodynamic grid for the fifth mode (Fig. 7) has been analyzed in detail, to highlight the capabilities of the implemented method. Fig. 8 shows the contour of the deformation of the structural grid, which can be seen also in Fig. 7(a). Figure 9(a) shows the results obtained by using linear polynomials as basis functions with the minimum number of nodes into the support of the C^0 RBF function, *i.e.* 4. The overall correlation is quite good; however, some oscillation can be observed near the intersection between the tip and the trailing edge, and near the wing root along the deformation nodal line. After increasing the number of points inside the weight support

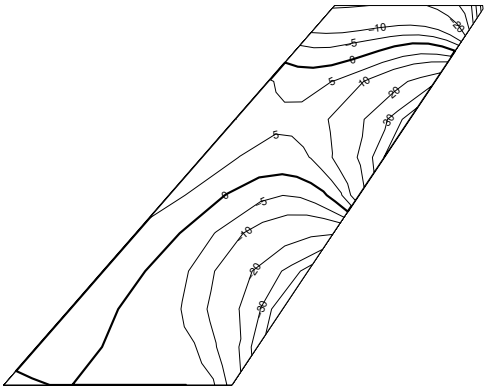


Figure 8: AGARD 445.6: contours of the mode 5 deformation based on original structural grid.

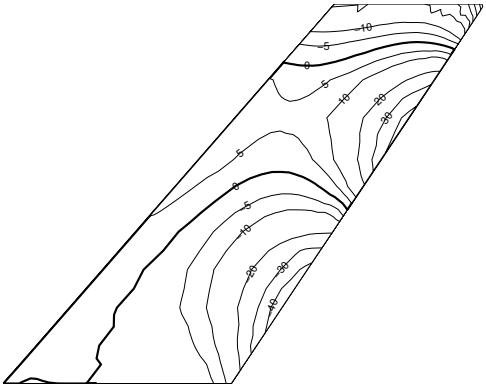
up to 10, the oscillations disappear, Fig. 9(b). A larger number of points causes an excess in smoothness, which reduces the maximum deflections at the trailing edge, and straight too much the nodal line near the wing root, Fig. 9(c). The last two pictures, Figs. 9(d)–9(e), show the effects of using more regular basis functions; with C^2 RBF the deformation blends more smoothly at the root, while C^4 RBF do not improve the results significantly. The excess of regularity required in an area where there are not so many structural nodes causes again the rise of some oscillations. The use of a second order polynomial basis is illustrated in Fig. 10. Using the minimum number of nodes for this interpolation order, *i.e.* 10, yields some oscillations, which are completely wiped out by using 20 point with C^4 basis functions, which is also able to correctly represent the constraint at the root nodes without adding any special treatment. The use of highly regular weight functions together with higher order polynomials gives optimal results.

4.4 Wing and engine nacelle of a commuter aircraft

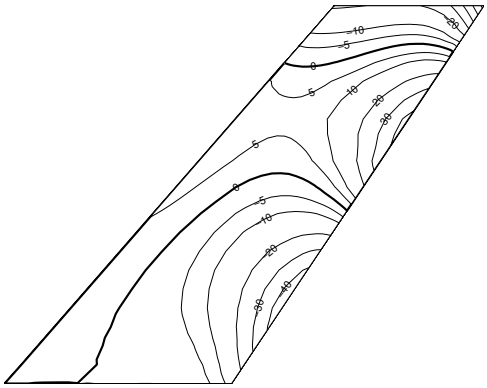
As a final example we show the behavior of the interface method when a complex hybrid wing-box model is used together with a three-dimensional surface grid for CFD computations. The modes of a wing and engine nacelle of a commuter aircraft are transfered to the CFD grid in Fig. 11.

5 CONCLUSIONS

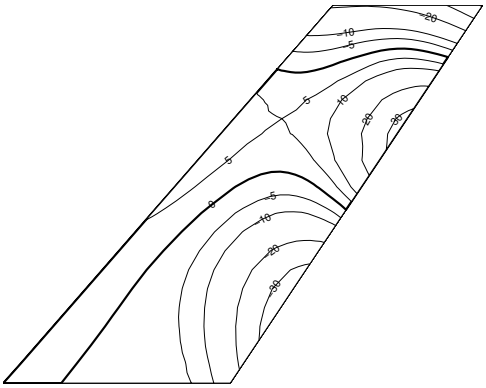
A new interface scheme has been presented. The capability of the method to interface information for partitioned FSI analysis in a wide range of aerospace cases has been shown. The scheme possess a good computational efficiency, given the locality which characterize it, and it leaves the possibility to the user to rule the smoothness easily. Furthermore, the locality allows to update the interface operator in case of mesh adaption without a great computational effort. Application of the presented scheme to fixed or rotary wings FSI cases is currently under way.[27]



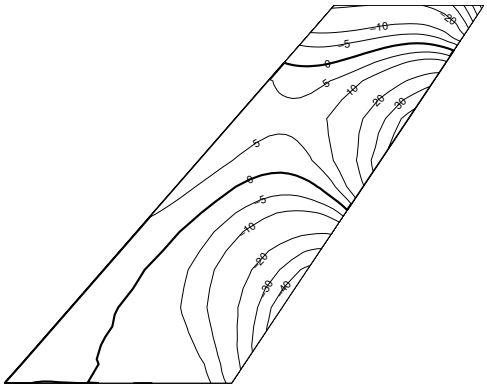
(a) C^0, 4 points.



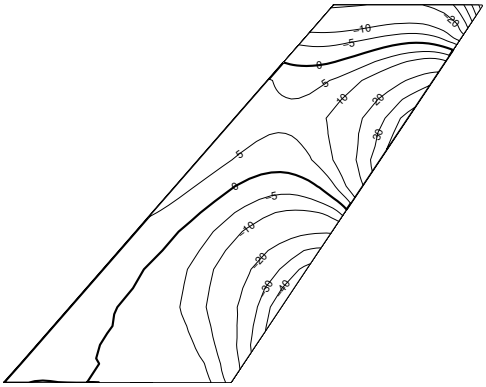
(b) C^0, 10 points.



(c) C^0, 40 points.

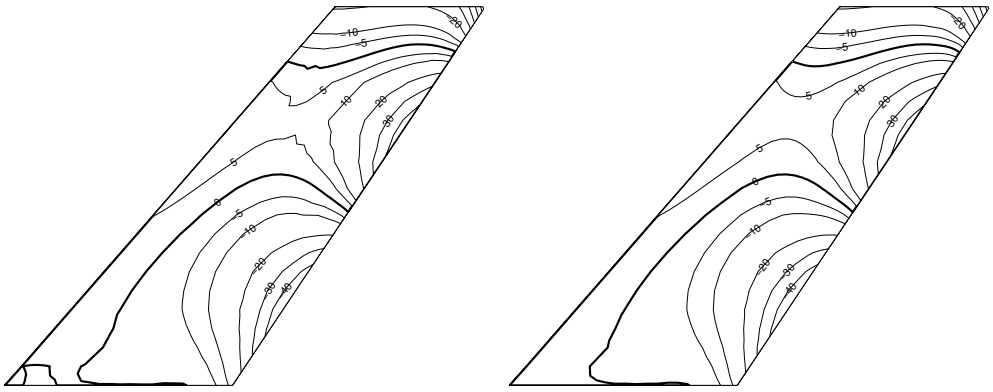


(d) C^2, 10 points



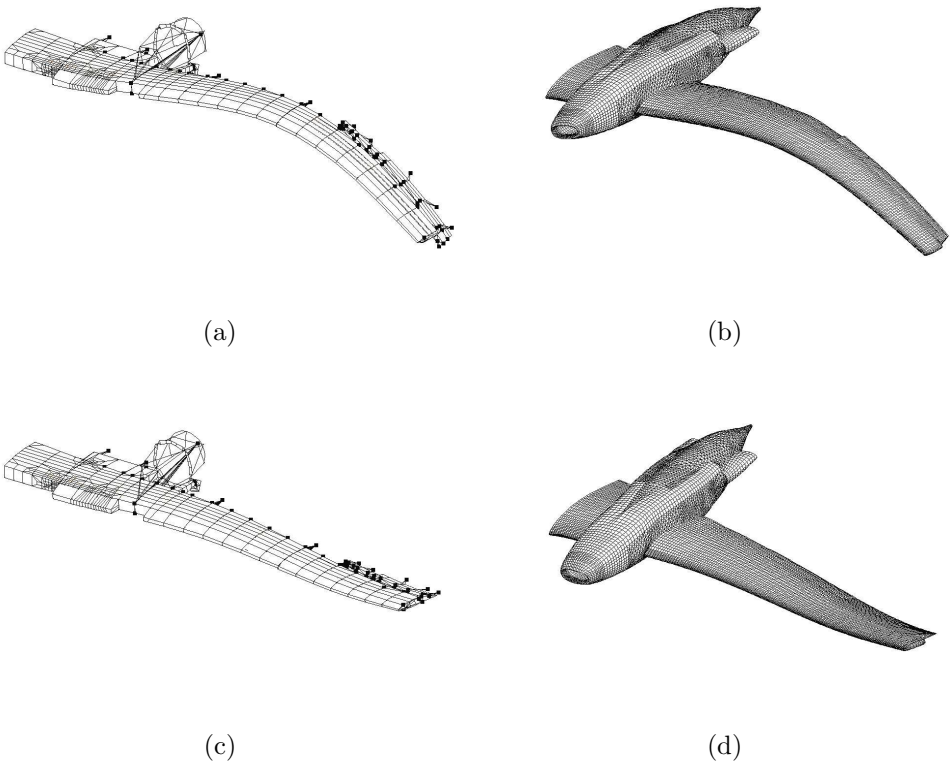
(e) C^4, 10 points.

Figure 9: AGARD 445.6, mode 5: interface results for different schemes with linear polynomial basis.



(a) C^0, 10 points. (b) C^4, 20 points.

Figure 10: AGARD 445.6, mode 5: interface results for different schemes with quadratic polynomial basis.



(a) (b)

(c) (d)

Figure 11: Vibration mode interfacing for a commuter wing-nacelle model.

6 ACKNOWLEDGMENTS

The authors wish to acknowledge the help of Ing. Luca Cavagna for the images of the commuter aircraft modes.

REFERENCES

[1] B. Perry, S. R. Cole, and G. D. Miller. A summary of an active flexible wing program. *Journal of Aircraft*, 32(1P):9–76, 1995.

[2] K.C. Park and C.A. Felippa. Partitioned analysis of coupled systems. In Ted Belytschko and Thomas J.R. Hughes, editors, *Computational Methods for Transient Analysis*, chapter 3, pages 158–219. Elsevier Science Publisher, Amsterdam, The Netherlands, 1983.

[3] *MSC/NASTRAN Handbook for Aeroelastic Analysis I/II*. The MacNeal-Schwendler Corporation, U.S.A., 1987.

[4] M. H. L. Hounjet and J. J. Meijer. Evaluation of elastomechanical and aerodynamic data transfer methods for non-planar configuration in computational aeroelastic analysis. ICAS–Publication, 1994. pp. 10.1–10.25.

[5] Marilyn J. Smith, Dewey H. Hodges, and Carlos E. Cesnik. Evaluation of computational algorithms for suitable fluid-structure interactions. *Journal of Aircraft*, 37(2):282–294, 2000.

[6] Guru P. Guruswamy. A review of numerical fluids/structures interface methods for computations using high-fidelity equations. *Computers & Structures*, 80:31–41, 2002.

[7] Charbel Farhat, Michael Lesoinne, and P. Le Tallec. 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. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.

[8] J. R. Cebral and R. Löhner. Conservative load projection and tracking for fluid-structure problems. *AIAA Journal*, 35(4):687–692, 1997.

[9] Xiangmin Jiao and Michael T. Heath. Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *International Journal for Numerical Methods in Engineering*, 61:2402–2427, 2004.

[10] Armin Beckert. Coupling fluid (CFD) and structural (FE) models using finite interpolation elements. *Aerospace Science Technology*, 4:13–22, 2000.

[11] P. C. Chen and I. Jadic. Interfacing of fluid and structural models via innovative structural boundary element method. *AIAA Journal*, 36(2):282–286, 1998.

[12] Armin Beckert and Holger Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science Technology*, 5:125–134, 2001.

[13] Jean Donea. Arbitrary lagrangian–eulerian finite element methods. In Ted Belytschko and Thomas J.R. Hughes, editors, *Computational Methods for Transient Analysis*, chapter 10, pages 474–516. Elsevier Science Publisher, Amsterdam, The Netherlands, 1983.

[14] Giuseppe Quaranta, Pierangelo Masarati, and Paolo Mantegazza. Multibody analysis of controlled aeroelastic systems on parallel computers. *Multibody System Dynamics*, 8(1):71–102, 2002.

[15] C. Farhat, M. Lesoinne, P. Stern, and S. Lantèri. High performance solution of three-dimensional nonlinear aeroelastic via parallel partitioned algorithms: Methodology and preliminary results. *Advances in Engineering Software*, 28:43–61, 1997.

[16] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37:141–158, 1981.

[17] Robert Schaback. Remarks on meshless local construction of surfaces. In *Proceedings of IMA Mathematics of Surfaces IX Conference*, Cambridge, 2000.

[18] G.R. Liu. *Mesh Free Methods. Moving Beyond Finite Element Method*. CRC Press LLC, July 2002.

[19] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments, 1996.

[20] Holger Wendland. Local polynomial reproduction and moving least square approximation. *IMA Journal of Numerical Analysis*, pages 1–15, 2000.

[21] R. Schaback and H. Wendland. Characterization and construction of radial basis functions. In N. Dyn, D. Leviatan, and D. Levin, editors, *EILAT Proceedings*. Cambridge University Press, 2000.

[22] Robert Schaback and Holger Wendland. Numerical techniques based on radial basis functions. In Albert Cohen, Christophe Rabut, and Larry L. Schumaker, editors, *Curve and Surface Fitting: Saint Malo 1999*, pages 1–16. Vanderbilt University Press, 2000.

[23] Z. Wu. Multivariate compactly supported positive definite radial functions. *Adv. Comput. Math.*, 4:283–292, 1995.

[24] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.*, 4:389–396, 1995.

[25] S. Arya, D. M. Mount, N. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In *Proceedings of the* 5^{rmth} *ACM-SIAM Symposium on Discrete Algorithms*, pages 573–582, 1994.

[26] E. C. Yates. AGARD standard aeroelastic configurations for dynamic response. I wing 445.6. R 765, AGARD, 1985.

[27] G. Quaranta, P. Masarati, G. Bindolino, and P. Mantegazza. Toward a computational framework for rotorcraft multiphysics analysis: Adding computational aerodynamics to multibody rotor models. In *30*$^{\text{th}}$ *European Rotorcraft Forum*, Marseilles, France, September 14-16 2004.