ACADÉMIE
DES SCIENCES
INSTITUT DE FRANCE

Comptes Rendus

Mécanique

Antoine Riols-Fonclare, Yann Vallat, Pierre-Emmanuel Des Boscs, Antoine Placzek, Alain Dugeai, Cédric Liauzun, Christophe Blondeau, Charly Mollet and Mikel Balmaseda

Modular and Interdisciplinary Methods for Aeroelastic Simulations (MIMAS)

Volume 353 (2025), p. 1315-1349

Online since: 1 December 2025

https://doi.org/10.5802/crmeca.337

ACADÉMIE DES SCIENCES
INSTITUT DE FRANCE

Comptes Rendus. Mécanique
2025, Vol. 353, p. 1315-1349
https://doi.org/10.5802/crmeca.337

Research article

Modular and Interdisciplinary Methods for Aeroelastic Simulations (MIMAS)

Antoine Riols-Fonclare [a], Yann Vallat [a], Pierre-Emmanuel Des Boscs [a], Antoine Placzek [a], Alain Dugeai [a], Cédric Liauzun [a], Christophe Blondeau [a], Charly Mollet [a] and Mikel Balmaseda [b]

[a] DAAA, ONERA, Institut Polytechnique de Paris, 92320 Châtillon, France

[b] DAAA, ONERA, Institut Polytechnique de Paris, 92190 Meudon, France

E-mail: antoine.riols-fonclare@onera.fr

Abstract. In the context of high aspect ratio wings or blades, aeroelasticity is becoming increasingly crucial for predicting the safety, efficiency, and performance of modern aircraft. This paper describes the development of MIMAS, a computational framework for simulating complex nonlinear aeroelastic phenomena from incompressible to highly transonic flows, in steady or unsteady configurations. On the one hand, MIMAS features advanced mesh deformation and transfer algorithms that have been renewed and optimised to enable faster computations compared to previous implementations within ONERA legacy codes. On the other hand, it offers ready-to-use coupling scenarios to support loose to strong fluid-structure interactions. This environment provides a high-level end-user abstraction layer that allows to couple a wide range of aerodynamic and non-linear structural simulation codes. In addition, the data structure is sufficiently generic to handle both structured and unstructured discretizations. In this paper, we first demonstrate the ability of MIMAS to reproduce existing representative computations such as harmonic forced motion, static coupling and dynamic coupling, without overhead induced by the modular implementation. Second, we present new capabilities of MIMAS, in particular the improved algorithms for mesh deformation and data transfer and its ability to leverage modern HPC architectures.

Keywords. Aeroelasticity, numerical simulations, coupling, mesh deformation.

Note. Article submitted by invitation.

Manuscript received 15 January 2025, revised 4 August 2025, accepted 24 October 2025.

1. Introduction

Designing efficient aircraft while minimizing environmental impact requires the ability to simulate highly complex unsteady physical phenomena. The latest trend, led by industry-leading companies such as Safran and Airbus, involves the use of high aspect ratio blades or wings often made of composite materials, for which non-linear aeroelastic phenomena need to be modelled accurately. Notable examples include the open rotor and stator engine developed by SAFRAN/GE Aviation in the framework of the RISE project and the X-WING plane developed by Airbus in the framework of the eXtra Performance Wing project. ONERA is also deeply involved in the hydrogen-powered aircraft (Gullhyver research project). These innovative architectures, with enhanced flexibility, must meet a wide range of structural and aeroelastic design constraints to ensure both safety and performances.

For example, in modern aircraft configurations, several technological components such as struts, morphing winglets or folding wing tips are planned to be employed to improve performance and support highly deformable wings. The interplay between all these components introduces a high level of complexity, which must be untangled. During critical flight phases (e.g., take-off and manoeuvres), unsteady aerodynamic forces engage with engine structures, generating periodic stresses that may shorten the lifespan of blade components. Accurate modelling of these forces aids in assessing fatigue-related risks and optimizing blade lifespan. Phenomena like whirl flutter are also crucial in designing the new generation of propellers. This form of aeroelastic instability can induce severe vibration amplitudes on the nacelle, potentially leading to structural failure if not adequately controlled [1–3].

Various approaches exist to address these challenges. Analytical methods and reduced models are commonly used due to the computational cost of high-fidelity coupled aeroelastic analyses. Although the rapid methods are efficient to explore parameter space in early-stage designs, they lack the accuracy to capture relevant physical phenomena. For detailed analyses, high-fidelity simulations are preferred, and there is a growing trend toward integrating multi-fidelity approaches for aircraft design and optimization [4]. In the era of exascale supercomputers, high-fidelity simulations raise the hope of predicting the entire aircraft or engine dynamics with all its detailed technological effects and inherent complexity. However, several factors currently limit these simulations, including code performance, interoperability, and limited physical modelling. To overcome these limitations, High-Performance Computing (HPC) codes designed to solve Fluid-Structure Interactions (FSI) using modern and optimized algorithms have become essential. In this context, ONERA has initiated the development of the SoNICS code (property of ONERA/Safran) for the fluid part [5] with the ambition of running this code on exascale machines with much better performance than existing codes (elsA, property of ONERA/Safran, [6]). The ONERA, DLR and Airbus collaboration is also deeply involved in the implementation of CODA [7], a new Computational Fluid Dynamics (CFD) code. Yet, coupling aerodynamical and structural problems needs additional key operations. These include transferring fields between both grids, deforming the fluid mesh during selected physical iterations, and ultimately coupling the solvers with optimised time-marching strategies. The latter involves in particular the development of new methodologies for solving non-linear coupled problems such as mixed end-point or mid-point temporal schemes, harmonic balance methods for periodic flows [8–11], asynchronous solvers [12], or ultimately the monolithic approach [13].

Historically, high-fidelity aeroelastic simulations at ONERA were performed directly within the elsA code, through a dedicated module [14]. All aeroelastic components are integrated directly within the code, following the approach of software like SU2 [15]. While elsA remains a powerful tool for such simulations, the increasing complexity of modern aerospace configurations is exposing its limitations. The elsA aeroelastic module is inherently restricted to structured grids and is not suited for analyzing non-linear structural behaviors. For static simulations, the structure's representation is via a condensed flexibility matrix while dynamic calculations rely on modal basis projections, which limits the ability to simulate nonlinear structural dynamics. Temporal mesh deformation is generally linearly interpolated due to the high computational cost of the existing techniques. Interpolation techniques for field transfers (e.g., forces, displacements) lack parallelization and are not optimized for large scale simulations. For time-marching, a basic fixed-point method is employed with a static relaxation parameter and coupling frequency, which may not be the most efficient way to couple fluid and structure. Finally, coupling efficiently with external Computational Structural Mechanics (CSM) codes such as NASTRAN is also challenging inside elsA, and adapting the code for that has become increasingly difficult.

To overcome these limitations and upgrade the aeroelastic coupling capabilities, subsequent developments have been performed to externalize and improve the different components outside from elsA's kernel. These components comprise mesh deformation techniques, transfer algorithms, mechanical solver and the driver that controls the time-marching iterations. This modular approach offers greater flexibility in terms of coupling and also makes it possible to work not only with elsA but also with other CFD codes, while reusing the same components for coupling. The modularization also provides a new way to tackle emerging challenges, such as simulating highly flexible structures in conjunction with flight dynamics, and their interaction with control systems and aeroelastic components, all of which are critical for modern aircraft. While simulations of this nature have already been conducted [16–18], they are often constrained by high computational costs. To enable more efficient calculations, it is clear that solvers must be externalized within a modular framework in which algorithms can be easily tested and improved.

Most of the external algorithms and tools have been then regrouped into a unified Python environment called MIMAS for *Modular and Interdisciplinary Methods for Aeroelastic Simulations*. The main goals of this library are to:

(1) take into account more complex physics, particularly structural non-linearities (e.g. contact, large displacements);
(2) improve the global performance of aeroelastic calculations and fit with the current HPC criteria;
(3) handle structured and unstructured mesh discretizations (the latter being crucial for simulating technological components);
(4) ensure the coupling and interchangeability of a large variety of CFD/CSM codes;
(5) extend to more sophisticated coupling (considering flight dynamics, trim, control);
(6) offer high flexibility in the developments of new methodology and the coupling between new components.

Note that the proposed approach is similar to existing coupling libraries such as preCICE [12] dedicated to partitioned multi-physics simulations, including, but not restricted to Fluid-Structure Interaction (FSI), and ParaSiF_CF [19] developed by University of Manchester and dedicated to the resolution of massively parallel partitioned FSI problem. The MDO Lab is also strongly involved in the development of a modular platform integrating aerodynamic (ADflow) and structural (TACS) solvers, and coupling these codes to perform shape optimization (OpenMDAO) [20].

The remainder of this article is structured as follows: in Section 2 a comprehensive overview of the MIMAS library and its alignment with the outlined objectives is given. In particular, the main evolution compared to previous elsA implementations is highlighted and the novel numerical methods employed within the library are detailed, focusing particularly on mesh deformation techniques and transfer algorithms. In Sections 3 and 4, a thorough validation and testing of the library across various industrial and academic configurations, encompassing airplanes, turbomachinery and helicopters, is presented.

2. Capabilities and numerical improvements beyond elsA

In its current version, MIMAS is capable of simulating most of the problems that have historically been addressed by elsA's aeroelastic module (AEL). But more importantly, it features new algorithms and optimized, parallelized versions of the existing ones. It operates with both structured and unstructured meshes (which was not the case with elsA), and has minimal dependence on external libraries, yet taking benefice from the current expertise and HPC libraries available at ONERA. MIMAS is today able to solve complex static or dynamic problems on industrial configurations, some including nonlinear structures in transonic flows. To achieve this, MIMAS has been interfaced with the new generation of CFD codes developed at ONERA such as SoNICS or CODA and finite element codes like NASTRAN, TACS [21] or Code_Aster [22] enabling the achievement of high fidelity aeroelastic simulations. It incorporates advanced parallelized algorithms for mesh deformation and the transfer of displacements and forces, ensuring lower CPU cost at each coupling step. The integration within a PyTree[1] coupling environment also ensures that the entire framework can manage the intricate data structures and workflows required by HPC fluid solvers. This encompasses shared-memory coupling and swift data search capabilities. Finally, the modular and object-oriented approach allows users to assemble components like building blocks in a LEGO set, helping them customize physical problems to their specific needs. In the following sections, we detail each new capability of MIMAS and provide a brief overview of the numerical methods that are used.

2.1. *A modern and modular architecture*

MIMAS is based on a modular architecture for which the fluid (CFD) and structural (CSM) codes are both external and exchange data through a coupling tree. The coupling tree is a Python structure enforcing the CFD General Notation System (CGNS) and sharing the data in memory with both the CFD and CSM codes, whenever possible. These data comprise the fluid and structural geometries (mesh, normals, surfaces, volumes, etc.), and the physical fields of interest (local forces, torques, displacements, structural modes...). Memory-sharing capabilities allow data to be stored efficiently without duplication, enabling quick access for both HPC codes and the coupling library. Dedicated methods have been developed to write data into the tree or access rapidly the fields from the tree in the numpy format (or dictionary format) without additional memory cost. Another advantage of this structure is that it supports parallel processing, enhancing the efficiency of coupled simulations. The transfer and deformation modules of MIMAS, as well as the dedicated module for geometrical calculations, operate independently from the coupling tree but retrieve data through it. Note that to exchange data with software such as NASTRAN, which may be installed on a remote machine, we use the Python module Pyro, which enables remote data communication. For NASTRAN in particular, the coupling is performed in-memory via the OpenFSI module, provided as part of the MSC Software suite. Hopefully, the resulting complex infrastructure is hidden from the user, who interacts only with the high-level coupling tree interface.

Aeroelastic calculations are then built around a central Python driver that executes an iteration loop between external CFD and CSM solvers (depicted as orange boxes in Figure 1). A common interface has been developed for each fluid and structural code, which enables to compute a single iteration or time step, restart the resolution process at a desired iteration, extract relevant data for aeroelastic coupling and update geometry/meshes from the Python framework. We emphasize that with MIMAS, users have the freedom to create their own drivers by constructing dedicated objects (models, coupling trees, mesh deformers, interpolators and solvers). They can manipulate these objects at each step of the aeroelastic loop to customize their calculations. Obviously, this freedom of action would not be possible inside compiled codes like elsA.

2.2. *Dealing with unstructured meshes*

Unstructured grids are increasingly prevalent both in industry and in academia due to their flexibility and ability to accurately represent complex geometries with a reasonable number of cells. Nevertheless, structured grids remain used for specific simulations, so there is today a

[1] A tree-like structure in Python with data stored in the leafs (similar to CGNS format).

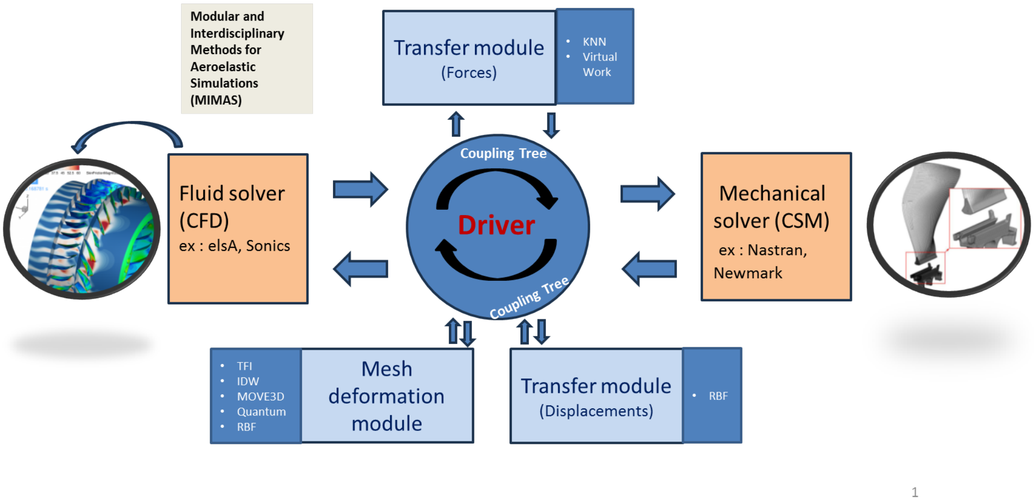


Figure 1. Modular architecture of MIMAS.

need for a coupling environment to handle both structured and unstructured meshes. To address these needs, we designed MIMAS from the very beginning to effectively handle both kind of grid representations.

Unstructured meshes in coupled problems can be challenging due to the multitude of formats available for representing mesh connectivity. We can for instance cite the polyhedral “NGon” format, used by several CFD codes (elsA, SoNICS) or the more classical “elements” format used by CODA and finite-element codes like NASTRAN. To deal with this multitude of formats, a dedicated mesh module has been implemented inside MIMAS to convert unstructured meshes into a common mesh object format. Geometrical quantities can therefore be computed regardless the initial mesh format. We implemented inside MIMAS an entire library in this unstructured framework able to compute geometrical quantities (such as volume, cells areas, face and cells centered quantities) and cell quality criteria (such as shear, aspect ratio, skewness...). Deformation, interpolation and transfer modules are further based on this internal library. At the moment, the library is able to deal with linear elements (TRI3, QUAD4, TETRA4, PYRA5, WEDGE6, HEXA8). In order to be fully generic an interface has been developed with the VTK (Visualization Toolkit) library which provides a wider range of geometric mesh calculation capabilities and many filtering options, including surface extraction. One of the key strengths of VTK is its ability to handle higher order elements. However, for the applications described in this paper, our internal library is largely sufficient. Unitary tests have been performed to check that deformation and transfer algorithms behave similarly in the structured or unstructured configurations (though, not shown in this paper).

2.3. *Improved mesh deformation algorithms*

A key step in aeroelastic calculations involves deforming the fluid mesh to match specified boundary displacements. Research efforts worldwide have been focused on optimizing these methods to reduce CPU cost while preserving the quality of the underlying mesh. Various strategies have been developed: for instance, [23] introduced a fast, accurate method based on Inverse Distance Weighting (IDW) that performs well for large displacements on complex 3D meshes; [24] further enhanced this method by integrating edge-swapping techniques, making it suitable for very large deformations. To maintain mesh orthogonality, some researchers, such as [25], proposed methods that incorporate both translational and rotational components of the deformation, while combining kd-trees traversal and dodecahedron structures to reach high performance. In parallel, alternative strategies have been developed using Radial Basis Functions (RBFs) for mesh deformation. Coulier et al. [26] applied innovative methods such as Fast Multipole Method (FMM) to factorize and solve the RBF interpolation kernel. For structural analogy methods, recent works have been focused into solving linear elasticity equations using modern iterative solvers with better preconditioning. These solvers particularly aim to preserve boundary layer cells by locally increasing stiffness, as described in [27]. Additionally, conditions imposed on boundaries other than the aeroelastic interface often necessitate adjustments to the algorithms governing these methods.

In MIMAS, several mesh deformation techniques have been re-developed or improved compared to existing formulations in elsA. These techniques are:

- the Inverse Distance Weigthing (IDW) [28,29] which has been significantly sped up, extended to unstructured grids and made compatible with turbomachinery boundary conditions;
- a hybrid IDW and transfinite interpolation (TFI) [30–32] for structured grids which takes advantage of IDW acceleration;
- elastic analogy [27,33,34] which has been hybridized with IDW for parallelization.

The primary focus in MIMAS has been to accelerate the IDW deformation method, in particular because this method is well-suited for unstructured meshes and is used in conjunction with almost all other techniques. The cost of IDW increases rapidly with the fluid mesh size since it requires to compute $n \times m$ operations where m is the number of target points in the fluid mesh and n is the number of source points (located at the fluid-structure interface). The deformation vector $\widehat{D}$ is determined by calculating the inverse distance-weighted average of the source points as follows:

$$\widehat{D} = \frac{\sum_{i=1}^{n} w_i D_i}{\sum_{i=1}^{n} w_i}, \tag{1}$$

where D_i is the value of the displacements at the known source location i and w_i is the weight assigned to the known source location i usually defined as $w_i = \frac{s_i}{d_i^\alpha}$. Here, d_i is the distance between the current point location and the known source point i, α is a positive integer power parameter that controls the influence of the distance to the source and s_i is the local cell surface. The larger the value of α, the greater the influence of nearby points on the estimated value. To reduce significantly the cost of the brute force algorithm, two implementations were considered, both preserving the rotational component of the solid displacements. The first is based on a multi-layered clustering of source points, with directional damping to enforce boundary conditions. This method, illustrated in Figure 2, shows typical reduction factor between 5 and 10 compared to the brute force approach, for standard aircraft simulations. The idea is to regroup source points into clusters, and summing the contribution of the cluster barycentres only (instead of all points belonging to the interface). To make this method optimal, the cluster size is increased gradually with the distance to the source. Different layers $\mathscr{L}_k$ in the 3D mesh are defined, depending on their distance to the source, and each of them sees a different set of clusters. For instance, layer $\mathscr{L}_0$ close to the source sees all source points, $\mathscr{L}_1$ sees only half of the original set... The distance criterion that make switch from one layer to another is computed using the above formula:

$$\mathscr{D}_k = \sqrt{\frac{\alpha(\alpha+1)\sigma_k}{2\epsilon_t}}, \tag{2}$$

where α is the IDW exponent, σ_k is the average variance of coordinates and displacements inside the clusters associated with layer $\mathscr{L}_k$ and ϵ_t is a tolerance defined by the user on the error introduced by the IDW clustering approximation. Note that this is obtained by doing a Taylor expansion of the IDW formula assuming that the distances within the cluster are small compared to the distance to the source (see Appendix A). For target points close to the surface, a complementary strategy introduced earlier by Shepard is adopted. The latter selects only a patch in the surface with size proportional to the distance to the source [28].

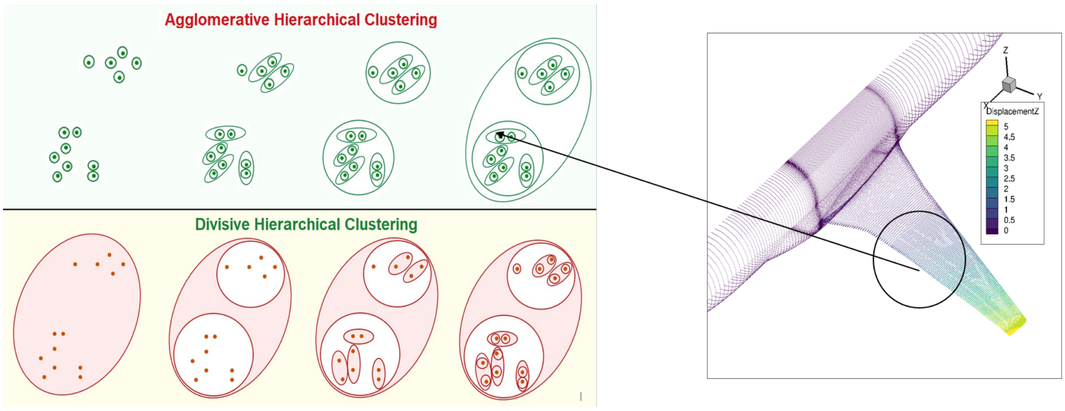

Agglomerative Hierarchical Clustering
Divisive Hierarchical Clustering
Z
Y
X
DisplacementZ
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0

Figure 2. Illustration of the clustering of the source points for IDW cost reduction (adapted from N. Arya 2023, AI, Machine Learning and Deep Learning).

The second implementation of the IDW method in MIMAS is based on quaternion algebra [35] and Fast Multipole like algorithm (FMM) with recursive octree traversal. This algorithm, called *Quantum*, has proven to be efficient and has been tested on multiple configurations including icing simulations on airplane. For external flows in unconfined domains, it is generally faster than the clustering method, since the octree traversal combined with the FMM naturally eliminates interactions between distant points, in a continuous manner. Note however that for turbomachinery cases, involving confined domains, this method is not necessarily the fastest because boundary conditions necessitate several iterations. Moreover, the *Quantum* method is not highly scalable and is particularly hard to linearise due to the algorithm recursion and the fact that rotations are treated incrementally.

Results of these two IDW implementations are presented in Figure 3 with their MPI scalability using multi-cores architecture. Tests have been achieved on a turbomachinery fan, representative of the new generation of Ultra High Bypass Ratio (UHBR) engine. We superimpose the scalability curves of the hybrid TFI/IDW and elastic analogy methods (which have not been accelerated yet in MIMAS). Maximum number of cores used is 256 on the ONERA's development cluster but tests have also been carried out on ONERA's production cluster with up to 512 cores. The original brute force IDW method from elsA aeroelastic module lies above all externalized MIMAS methods. The first IDW algorithm with layered clustering (cyan line) scales quite well and is almost 8 times faster than the IDW from elsA (dashed blue line). The second IDW algorithm *Quantum* with FMM (red line) shows also consequent CPU time reduction (especially for external flows simulations, not shown here) but its scalability seems to be affected for a number of cores larger than 100.

Note that compact RBFs can also be used within MIMAS to deform a mesh and are solved using the MUMPs (MUltifrontal Massively Parallel Sparse direct Solver) library (see Section 2.4) but the choice of the appropriate kernel is not necessarily obvious and additional work is required to clarify this point. Work is also in progress to make the elastic analogy compatible with unstructured meshes and to modernize the linear elastic solver (*move3d*), previously implemented in elsA for structured meshes exclusively. Finally, we stress that mesh deformation differs from adaptive re-meshing (such as AMR) since it preserves the connectivity and topology of the underlying mesh structure, as well as the number of cells in the mesh. When possible, mesh deformation is preferred to adaptive re-meshing because it avoids heavy connectivity computations. However, in the future, strategies combining both methodologies could be considered.

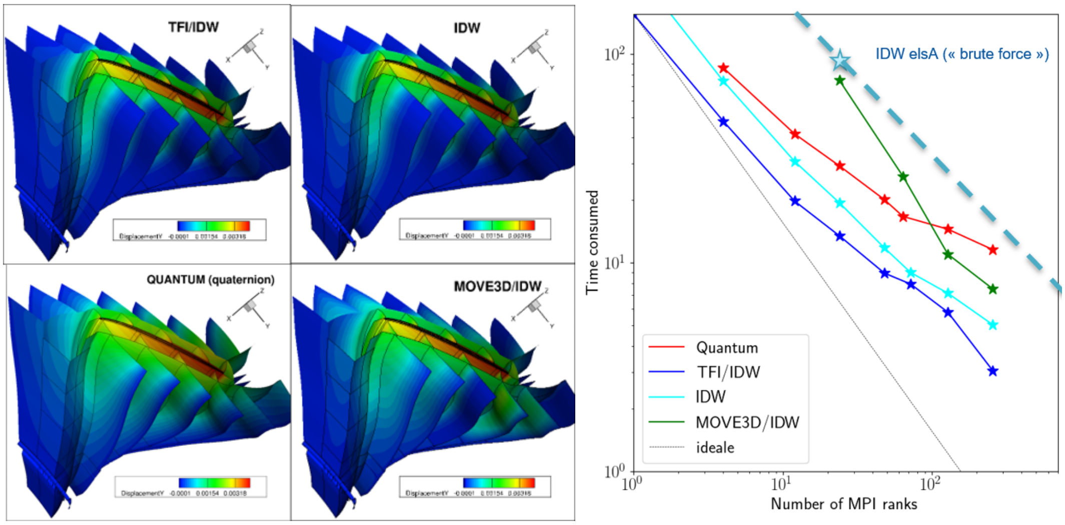


Figure 3. Left: Comparison of the azimuthal displacement field obtained through different mesh deformation methods available in MIMAS. Here they are applied on a single sector mesh of an UHBR turbomachinery fan. Right: Scalability of these methods on ONERA's development cluster for the turbomachinery case (4 million points). The cyan curve accounts for the layered clustering method with 8 layers and a 20 % tolerance on the error made on the IDW formula. The dashed line is the reference calculation using brute force algorithm in elsA.

2.4. *Parallel transfer algorithms*

Field transfer methods play a crucial role in the achievement of aeroelastic coupling. They allow to exchange physical information such as forces and displacement between the aerodynamic and structural grids. The main issue is that the fluid grid's numerical interface rarely matches with the structural elements boundary (see Figure 4). In some cases, the structural model may use simplified elements, like beams, which fail to fully represent the boundary shell, complicating accurate physical computations at the interface. So far, a wide range of algorithms have been proposed in the literature [36–42] and giving an exhaustive review on all these algorithms constitutes a full paper in itself. In most cases, transferring loads necessitates condensing the CFD force field onto a reduced set of points (compared to the original CFD grid) while the transfer of displacements often entails the opposite approach. Consequently, the algorithms ruling these transfers may vary considerably. The primary objective of these algorithms is to maintain high computational speed for large sets of interpolant surfaces while conserving physical quantities relevant to aero-structural interactions.

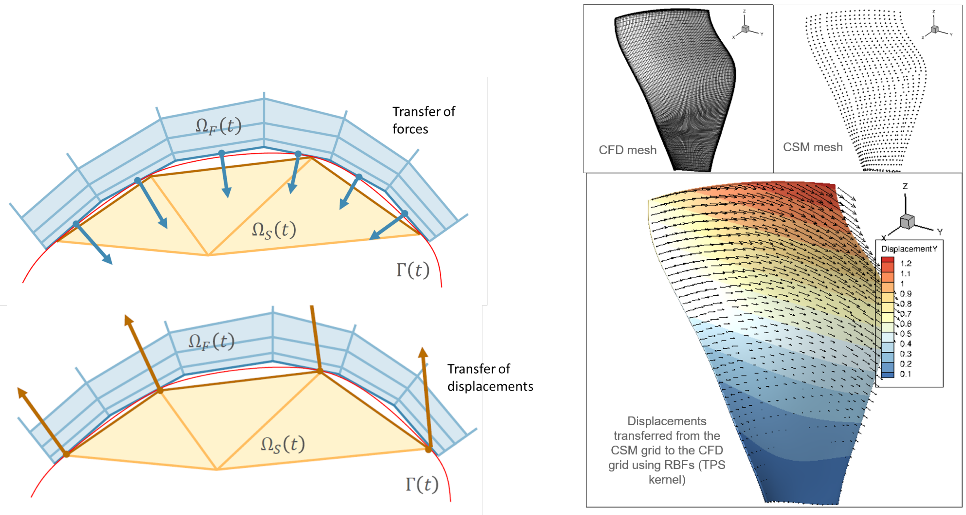


Figure 4. Left: Illustration of data transfer between the fluid and the structural grids. Right: Application of RBFs smoothing algorithm with TPS kernel on an Ultra High Bypass Ratio (UHBR) fan blade.

In MIMAS, we developed and upgraded several transfer algorithms (some of them originally present in elsA's aeroelastic module), with an emphasis on parallelizing them for efficient use in HPC environments. This parallelization was crucial to ensure these algorithms are scalable and capable of handling large datasets.

For the transfer of displacements (CSM grid to CFD grid), a first class of algorithms is the Radial Basis Functions (RBFs) smoothing. This interpolation method is rather convenient since it does not require the knowledge of the mesh connectivities, nor the structural discretization. Given a set of structural data (x_i, u_i) where x_i are the input points and u_i are the corresponding displacement values, the interpolated function $u(x)$ at a given point x in the space is expressed as:

$$u(x) = \sum_{i=1}^{N} w_i \phi\big(\|x - x_i\|\big) + P(x), \tag{3}$$

where: N is the number of radial functions which is chosen to be equal to the number of known structural source points; ϕ represents the chosen radial basis function, which can be "thin plate spline" (TPS), gaussian, multiquadric, etc.; $\|x - x_i\|$ is the distance between the chosen point x and the known data point x_i, and w_i are the weights associated with each data point. Determining the weights involves solving a linear system of equations of size N, typically represented as a dense matrix equation. The weights w_i are adjusted to cancel the error between the interpolated values and the actual data. The decomposition in radial basis functions is often modified to also include polynomial term

$$P(x) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z + \cdots \tag{4}$$

to ensure conditionally positive definite radial functions (e.g. TPS), together with matching constraints on the expansion coefficients [41,43,44]. To sum up, the interpolation problem is equivalent to solve for the system:

$$\begin{bmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\beta}^T \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} \tag{5}$$

with $\mathbf{\Phi}_{ij} = \phi\big(\|x_i - x_j\|\big)$, $\mathbf{w} = (w_1, w_2, \ldots, w_N)$, $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3)$ and

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & z_N \end{bmatrix}. \tag{6}$$

The major drawback of this method is its cost when the structural model becomes large ($\gg 10^4$ points). If used in the context of mesh deformation, this is even worse since the size of the matrix can exceed easily 10^5 by 10^5 points. In the literature, several techniques address this issue while maintaining a smooth global displacement field [40]. In the legacy elsA aeroelastic module, a preprocessing step involved a manual selection of a reduced set of structural points for transferring the displacements. The selection was performed in order to improve the numerical stability of the transfer method (e.g. avoid the excessive point density that could yield ill-conditioning interpolation matrices when the RBF are used) and to conserve the geometric properties of the structural deformations (e.g., displacement gradients, curvature zones). MIMAS offers an automatic technique to sub-sample the structural model using agglomerative clustering methods, similar to those employed for mesh deformation (see Section 2.3). These methods naturally tend to select points near sharp gradients and they give satisfactory results for turbomachinery blades. However, they have limitations, as they rely solely on spatial proximity or field distribution and do not consider the mechanical properties of the points. A dedicated parallel LU solver for dense matrix, using the Message Passing Interface (MPI) protocol, has been developed to solve the linear problem (5). Another solution provided by MIMAS is to use compact support functions, which can result in a sparse RBF matrix. One advantage of this method is that it effectively suppresses the influence of distant points, which is particularly beneficial when dealing with a wing and a horizontal tailplane for example. This localized influence helps in some cases in preserving the physical behaviour of the problem. To provide with an efficient computation of the sparse system, we linked MIMAS with the MUMPs library [45], which is considered today as one of the most efficient direct solver for sparse linear system.

A second class of algorithms is based on beam kinematics. In that case, source data are known along a beam line and the goal is to reconstruct the displacement around it. Given a point P in space, the algorithm first computes its orthogonal projection onto the beam line and reconstructs the displacements locally using hermite spline elements associated with the beam. The novelty in MIMAS lies in the ability to account for significant deformations, automatic detection of corners and broken lines, and the removal of degeneracy in orthogonal projection. Additionally, it lifts the restriction associated with Euler–Bernoulli hypothesis (where it is assumed that sections remain orthogonal to the beam curve), while still maintaining orthogonality in projections. Examples of deformation fields using this technique are illustrated on Figure 5. The red lines account for the beams from which the deformations are known, while the surface meshes around are the target grids where displacements are re-constructed.

Finally, for load transfer, two classes of algorithms have been implemented: the first is based on a nearest neighbour search and conserves the resultant of forces and moment. The structural points for transfer are chosen on the key load-bearing elements, such as beams, wing box, and other primary support components of the structure, so that forces are correctly absorbed by the whole structural model. The main benefit of this method is that it is quite straightforward and cheap in terms of computational time. However when important change in the geometry occurs or when pressure loads encounters huge gradient (typically near the leading or trailing edge of an airfoil, or near the shocks), this approach can lead to a wrong local distribution of forces on the structure. In addition, it does not conserve energy, and methods using virtual work technique can be preferred in that case. The second method, based on virtual work conservation, uses

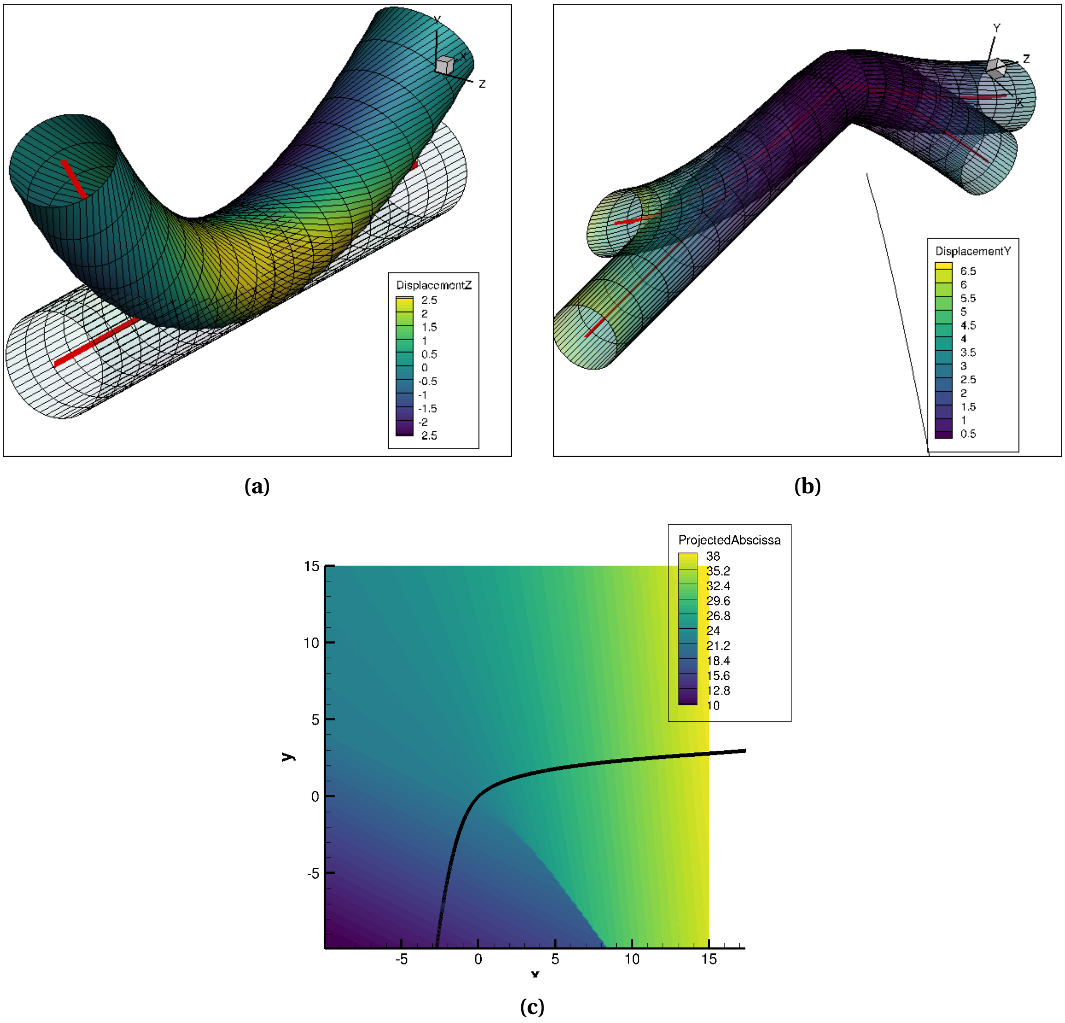


Figure 5. Illustration of kinematics beam transfers. (a) An application of the transfer around a flat beam line with combined torsional and bending motion. (b) The deformation around a curved beam line with critical degenerated surface for orthogonal projection. (c) The intersection of the critical surface with the xy plane (thick black line).

Radial Basis Functions (RBFs) for interpolating forces and displacements. A base of continuous functions $\vec{\Phi}^i$, $i = 1, \ldots, N_b$, is chosen so that an "admissible" displacement field $\overrightarrow{\delta u}$ can be decomposed into:

$$\overrightarrow{\delta u}(x, y, z) = \sum_{i=1}^{N_b} w_i \vec{\Phi}^i(x, y, z). \tag{7}$$

The conservation of virtual work between fluid and structure associated to each individual basis function leads to:

$$\langle \vec{f}_{\text{aero}}, \vec{\Phi}^i \rangle = \sum_{j=1}^{N} \vec{f}_{\text{aero}}[j] \cdot \vec{\Phi}^i[j] = \sum_{k=1}^{M} \vec{f}_s[k] \cdot \vec{\Phi}^i[k] = \langle \vec{f}_s, \vec{\Phi}^i \rangle \tag{8}$$

where j and k are the indices of the fluid and structural nodes respectively, N and M are the number of aerodynamic and structural nodes. Since the aerodynamic nodal forces $\vec{f}_{\text{aero}}$ and the basis functions are known, the first term can be easily evaluated. In the right-hand side, only the terms $\vec{f}_s[k]$ are unknown. The basis functions $\vec{\Phi}^i, i = 1,\ldots,N_b$ is constructed from a sum of Radial Basis Functions (RBFs) and enforce unitary displacement at control point and zero elsewhere. These functions are supplemented with a polynomial, with constant that ensures the conservation of global loads and a linear term to preserve global moments. Efforts are preserved locally through the local support of the RBF functions. The linear system for the unknown $\vec{f}_{s_k}$ is solved using the same methods described previously for displacement transfer. Note that this method can be significantly costly since we need to solve undetermined weights for each basis function $\vec{\Phi}^i$, $i = 1,\ldots,N_b$. It gives generally smoother results than the nearest neighbour (outside discontinuities such as shocks), but can lead to artificial loads concentration when structural elements intersect.

2.5. *Multiple FSI interfaces*

Systems like rotorcraft blades, wings with control surfaces, or compressor stages in turbomachinery involve different structural parts that may be numerically treated separately for several reasons (physical behaviour, performance...). Multiple FSI interfaces allow each structural component to be treated independently in terms of transfer algorithms, while interactions between components are captured exclusively through the fluid dynamics. However, a key challenge of this approach is that transferring individual displacements from each component to its corresponding aerodynamic surface create discontinuities between adjacent surfaces. A well-known example is the interaction between the fuselage and the wings of an aircraft. Suppose that the wing and fuselage are solved using the same structural model, but both components are decoupled during the transfer process. The interpolation methods applied to each component do not process identical data, resulting in discrepancies at the junction between the two. To solve this issue, we implemented in MIMAS an algorithm allowing to match the different components. Consider two non-overlapping sets of points S_1 and S_2 for each structural component and their corresponding aerodynamical surfaces A_1 and A_2. First two transfers are achieved on $A_1 \cup A_2$, the first using the S_1 structural data, the second the S_2 data, giving respectively a displacement field u_1 and u_2. Using efficient kd-trees, we compute for each aero-dynamical point its distance d_I from the intersection between the set A_1 and A_2. This is estimated by computing the distance to the nearest neighbour belonging to the opposite surface. Note that better strategies involving convex hull border or surface fitting with implicit function evaluation may provide more accurate distance to the intersection. When the distance of the aerodynamical points from the intersection is known, we use blending functions (such as sigmoids or polynomials) to match the displacements between the two surfaces. For example, if we use a cubic blending between A_1 and A_2, the transferred displacement is:

$$u(x,y,z) = w u_1(x,y,z) + (1-w) u_2(x,y,z) \tag{9}$$

with

$$w = \begin{cases} \left(\frac{d_I}{\delta}\right)^3 - \frac{3}{2}\left(\frac{d_I}{\delta}\right)^2 + 0.5, & \text{if } d_I \le \delta, \\ 0, & \text{if } d_I > \delta. \end{cases} \tag{10}$$

Here δ represents the depth into the neighbourhood region and is set to the distance of the furthest point identified during the k-nearest neighbour search.

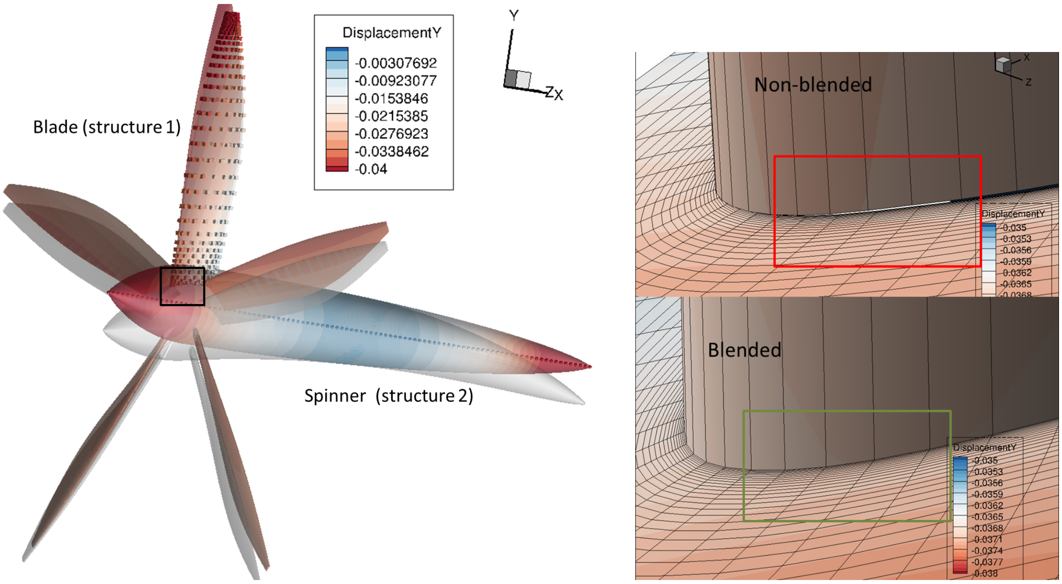


Figure 6. Illustration of the blending method for multiple FSI interfaces when independent transfer algorithms are used for each structural component. On the left is shown the deformation imposed on a 5-bladed propeller with bending of the spinner and bending-torsion of the blades. The cube points on the blade represent the structural nodes used for the transfer on the propeller blades while aligned spheres are the structural nodes for the spinner axis. Right top: Displacement field and fluid mesh after the transfer without blending. Right bottom: Same after blending.

An illustration of the method is provided in Figure 6, where it is applied to a 5-bladed propeller. The structural displacements are modelled using a global analytical mode (though not necessarily physically realistic) that combines the bending of the spinner with the bending-torsion motion of the blades. While the structural components are computed consistently, the displacement transfer between the blade (S_1) and the spinner (S_2) is handled separately. For the blade, we employ a Radial Basis Function (RBF) approach based on a set of 3D points on the blade surface, represented by cube points in Figure 6. For the spinner, only points along the X-axis are considered, with no point taken from the spinner's surface. A beam kinematics transfer method is used to model the displacements on the fluid surface, which is shown in color in the left panel of Figure 6. Once the two independent transfers are combined, we observe that the displacement field is discontinuous between the spinner and the blade, as expected. This discontinuity results in a non-matching intersection, as illustrated in the top right panel of Figure 6. After applying the blending algorithm using kd-tree search, we demonstrate that the displacements become smooth in the intersection regions, with the two meshes coinciding, as shown in the bottom right panel.

Note that there are cases where we would like to decouple FSI interfaces that are not directly connected, such as a wing and a horizontal tailplane. In such cases, we can blend each surface with the shared connecting surface (i.e. the fuselage) or use compact radial basis function (RBF) interpolation. The latter naturally decorrelates the two surfaces once a certain distance condition is satisfied.

2.6. *Coupling strategies and time-marching*

The standard approach in coupling fluid/structure is a sequential execution of fluid and structure solvers. This approach, although known to be quite robust, generally leads to slower convergence and load imbalances in a parallel setting. With the modular and external approach of MIMAS, new ways of coupling fluid and structure can be explored, unlike elsA, which operates within a more rigid and static architecture. First, for static problems, we are not anymore limited by the fixed-point approach with constant relaxation parameter and constant coupling period. We can indeed call the structural code at chosen fluid iterations to optimize convergence. A criterion can be established, such as setting a threshold for the relative change in the fluid residual compared to the previous coupling iteration. The relaxation parameter can also be prescribed by users at any time. Other strategies, like Aitken under-relaxation [46] have been used and tested within our new framework (see Section 4.1). This algorithm adapts the relaxation factor at every iteration based on current and previous state of the structure. If we define the residuals of the structural displacements at previous iteration $k-1$ and current iteration k,

$$\mathbf{r}_{k-1} = \mathbf{u}_{k-1} - \mathbf{u}_{k-2} \qquad \text{and} \qquad \mathbf{r}_k = \mathbf{u}_k - \mathbf{u}_{k-1}$$

and $\Delta\mathbf{r}$ the residual difference:

$$\Delta\mathbf{r} = \mathbf{r}_k - \mathbf{r}_{k-1}.$$

Then the relaxation factor at current step ω_k is calculated as

$$\omega_k = -\omega_{k-1}\frac{\mathbf{r}_{k-1}\cdot\Delta\mathbf{r}}{\|\Delta\mathbf{r}\|^2}$$

where the dot product here represents the sum of the individual product on each structural degree of freedom. To calibrate the initial relaxation parameter ω_0, we use a similar approach inspired by the preCICE library [12]. For dynamical problems, it is almost similar, except that the fixed point approach is applied at each physical time-step. Exchanges between the fluid forces and the structural displacements are performed during sub-iterations (dual loop) of the time integration method (see Figure 7). In elsA, the strategy was slightly different since the aeroelastic solver iterates multiple times on the same time step to converge the fluid and structure. We emphasize though that for dynamical problems, better strategies involving mid-point time schemes, asynchronous coupling schemes or parallel quasi-Newton least-squares solvers [12], mixing both fluid and structure states, are possible. Research on these coupling time-scheme is active at ONERA and could replace progressively the current implementations.

Another scope of research is the externalization of non-linear iterative fluid or structural solvers themselves. Over the past decade, significant efforts have been made to externalize the Jacobian and its matrix-vector products from both CFD and CSM solvers. This is particularly useful if one wants to solve the implicit phase externally. In particular, this offers the capability of pre-conditioning the system with coupling terms and use more advanced or dedicated Newton–Krylov solvers for FSI problems. Ultimately, the monolithic approach could be investigated, for which the whole fluid/structure system is solved externally. Note that other strategies such as the Time Spectral Method (TSM) for periodic systems are also investigated but remain out of the scope of the MIMAS library for now. More details about their current implementation at ONERA is presented in [11].

2.7. *Non-linear structural coupling*

As we mentioned earlier in the introduction, high-aspect-ratio wings and large-diameter fans or propellers exhibit increasingly nonlinear behavior due to large displacements. Another cause of nonlinearity is the friction between structural components (e.g. for blade-disk assemblies)

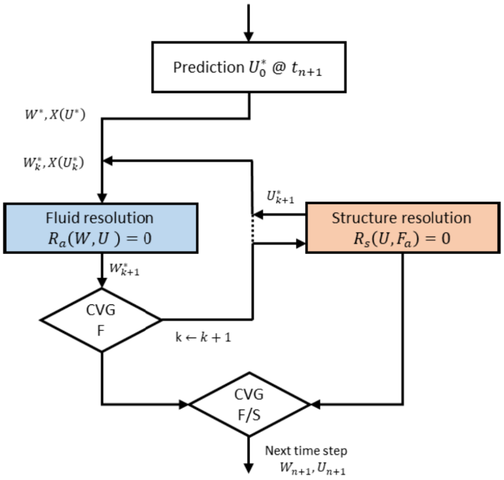


Figure 7. Illustration of the coupling scheme for dynamical problem in MIMAS. The structural displacements and the fluid mesh are updated regularly during the dual loop of the "Gear" or "DTS" (Dual Time Stepping) temporal scheme.

which can influence significantly the aeroelastic response of aeronautical structures. A nonlinear modelling of the structure is therefore required for the resolution of the coupled aeroelastic problem. When structural nonlinearities due to large displacements U are considered, the dynamic equation of motion contains an additional term $f_{\mathrm{nl}}(U)$:

$$M\ddot{U} + C\dot{U} + KU + f_{\mathrm{nl}}(U) = F_a(U, t)$$

where M and C are the mass and viscous damping matrices and $F_a(U, t)$ the aerodynamic forces acting on the structure. In the static case, the equation reduces to

$$Ku_s + f_{\mathrm{nl}}(u_s) = F_{a,s}(W)$$

where u_s is the displacement resulting from a steady aerodynamic force. The treatment of the nonlinear term requires an iterative Newton process. In the dynamic case, the Newton algorithm is combined with the time integration scheme, generally handled with Newmark or more generally HHT methods for structural problems. Dealing with such nonlinear structural models within the CFD solver kernel is not necessarily the most appropriate, as different methods and solvers than those used for the fluid are required. The modular and partitioned approach is then well suited to rely on an external structural solver. In MIMAS, an interface has been implemented with the finite element code NASTRAN and preliminary developments have been carried out to couple MIMAS with Code_ASTER of EDF. Upcoming developments are also intended to couple with the next generation ONERA code A-set. For NASTRAN we use the SOL400 nonlinear solution enabling large displacements, and the coupling is performed either by files or through the C++ OpenFSI interface. This interface offers a way to stop and restart NASTRAN at each coupling iteration, and communicate buffers address directly between NASTRAN and the Python driver. Such a coupling with OpenFSI has already been performed in the past at ONERA, using elsA's legacy aeroelastic module, enabling a partial externalization [47]. Note that user Python interface of MIMAS allows to prescribe several NASTRAN parameters to calibrate the non-linear resolution.

The structural non-linearities could be modelled within a projection based by a Reduced Order Model (ROM). This can be useful for dynamic aeroelastic computations to avoid the coupling with the full finite element model at each physical iteration. In projection based reduced order models, the displacements of the structure are computed by using a representative projection basis that captures the non-linear structural response, so that $U \approx Vq$ with $V = [\Phi, D]$, where Φ represents the linear eigenmodes and D represents additional modes (e.g., modal derivatives or dual modes). The nonlinear dynamic equations of motion are then projected onto this reduced basis, resulting in a simplified set of scalar nonlinear equations:

$$\mu\ddot{q} + \beta\dot{q} + \left(\gamma + \gamma_{\mathrm{nl}}(u_s)\right)q + \widetilde{g}_{\mathrm{nl}}(u) = f_{a,g}(W).$$

With the classical approach, the nonlinear internal forces need to be assessed by means of a FE solver, reducing the time efficiency of the reduced order model. Thus, an autonomous reduced order model could be obtained by considering that the internal non-linear forces are approximated by a third-order polynomial function of the generalised displacements. The k-th coefficient of the nonlinear force vector is thus:

$$\widetilde{g}_{k,\mathrm{nl}} \approx \sum_{i,j} \rho^k_{ij} q_i q_j + \sum_{i,j,m} \theta^k_{ijm} q_i q_j q_m.$$

The polynomial coefficients are identified from a set of precomputed nonlinear static solutions using the Implicit Condensation method with Expansion (ICE) [48]. This provides an explicit expression of the reduced-order model in terms of the generalized coordinates q, which can be efficiently coupled with the fluid solver. The nonlinear reduced order model detailed above has been coupled to the fluid solver elsA on the simplified test case of a flexible beam placed in the wake of a cylinder generating an unsteady forcing [49]. It has also been used to evaluate the dynamic response of a UHBR fan blade subject to an external forcing induced by an inlet distortion [50]. These studies were conducted at ONERA outside the MIMAS framework, but more recently, reduced-order model of the ERATO helicopter blades have been coupled with fluid models within the MIMAS framework (see Section 4.3). For that, an entire interface has been written between MIMAS and an external tool designed to build ROMs from finite element solver Code_ASTER (EDF).

2.8. *Coupling with advanced HPC fluid codes (CODA, SoNICS)*

MIMAS handles a generic interface to HPC CFD solvers which is based on class abstraction. The class provides methods to achieve basic operations, including advancing one or several fluid iterations, extract forces and update geometry. On the ONERA/SAFRAN side, the SoNICS code is a next-generation software designed to enhance performance and usability in turbomachinery applications. It features a new architecture that incorporates operator graph structure for tailored calculations, cache-blocking, automatic linearized and adjoint through the Tapenade library [51], efficient code generation for optimized hardware usage (including GPUs), and advanced capabilities for handling complex physics. SoNICS handles multiple fluid species and dynamic mesh adaptation. We recently carried out developments within the SoNICS code to implement the Arbitrary Lagrangian Eulerian method for enabling grid velocities for aeroelastic studies. We also wrote the triggers classes to achieve basic operations mentioned earlier such as updating the grid and extracting forces. In particular, updating the grid in SoNICS has required to modify the primary execution graph to propagate the new metrics everywhere in the code and to each operators. In terms of data, it utilizes the standard CGNS format, featuring polyhedral connectivity representations (NGon). First simulations have recently been conducted with harmonic forced motions on the M6 wing configuration to validate the implementation of the ALE formulation for deforming grids.

On the ONERA/DLR/Airbus side, CODA is an advanced computational code which includes a large variety of spatial schemes (finite volumes & discontinuous Galerkin). It focuses on optimizing aircraft design through efficient numerical methods, operators linearization, and integrates various physical components, including fluid-structure interactions. It relies on the FlowSimulator DataManager (FSDM) which provides an HPC library for CFD-based simulation workflows, models, data manipulation and multiprocessing. As the core library includes Python interfaces, data is easily accessible through MIMAS. However, since the data format is very different from the standard CGNS format, an important work has been performed to convert efficiently the mesh and the aeroelastic data (forces, displacements, etc.) into our own representation. Section 4.1 shows first aeroelastic calculation using CODA and MIMAS jointly, with comparison with elsA using the same mesh.

2.9. *Coupling with Vortex Particle Method (VPM)*

Finally, MIMAS is able to couple with medium fidelity method such as the Vortex Particle Method (VPM) [52]. The originality of such coupling is that the fluid model is generally mesh-free. In VPM, the flow is represented as discrete particles that carry vorticity. Each vortex particle has properties such as position, circulation strength, and sometimes additional attributes like velocity or density. The method operates in a Lagrangian framework and is particularly efficient to simulate complex wake dynamics. The governing equation is the transport equation of the vorticity, which includes a stretching term and a viscous diffusion term. The viscous diffusion term is used to model the influence of both the molecular viscosity and the turbulence associated to small eddies. In MIMAS, we implement an interface with ONERA's code Vulcains [53]. One of the existing formulations, consists in a lifting-line/VPM coupling where the lifting components (e.g. blades, wings...) are modelled with the Blade Element Theory (at the quarter chord position). At each time step, new particles are generated based on this lifting-line model. The local velocity induced by the particles is then projected onto the normal plane of each lifting-line section, from which the angle of attack, Mach number, and Reynolds number are derived. Using 2D airfoil tables and Kutta–Joukowski theorem, it is then possible to calculate the circulation for each blade section. New particles are generated to account for this circulation and an inner iterative loop is performed at each time step to ensure convergence between the newly generated particles and the blade circulation (as the new particles also contribute to the velocity induced along the lifting line). For aeroelastic coupling, there are two different scenarios:

(i) the structural model is a beam model which coincides with the lifting line; in that case transferring structural forces and displacement between the fluid and the structure is trivial;

(ii) the structural model is based on 3D or 2D finite elements; then the full pressure and force field are reconstructed on the surface using the 2D airfoil tables.

The transfer of the displacements can be done using RBFs on an auxiliary grid (though not optimal and physically questionable) or directly by condensing the structural displacements into a local rigid motion (translation/rotation) for each sections of the lifting line model. An example of simulation coupling VPM and nonlinear Finite Element Model (FEM) is presented in Section 4.3. Note that the VPM can be also coupled with Eulerian grid model or URANS simulations but this is out of the scope of the present article.

3. Validation cases: classical aeroelastic calculations

Before presenting applied cases that showcase new capabilities, we first demonstrate that MIMAS can accurately replicate standard aeroelastic calculations which have been historically computed using elsA's aeroelastic module [14].

3.1. *Static coupling*

To validate the modular approach, we first test the case of a static coupling on the Common Research Model (CRM) plane configuration. To compare MIMAS with elsA's aeroelastic module, a basic fixed-point method, including constant relaxation of the structural displacements, is used. In MIMAS, fluid iterations are solved externally by elsA while the structural problem is solved directly at the Python level using a condensed structural stiffness matrix with smaller dimension compared to the initial NASTRAN model. The model comprises mass loads for the engine and the fuel, while gravity force is added externally by MIMAS using the area/thickness of shell elements. The fluid domain is discretized with a coarse structured grid of about 1 million cells. The CFD resolution is inviscid (Euler model) and uses a Roe scheme with the Van Albada flux limiter. Fifteen fixed point iterations between the fluid and the structural solvers are performed to reach the equilibrium. The coupling is handled with the TFI/IDW hybrid method for deforming the structured mesh, and the nearest neighbour and RBFs respectively for force and displacement transfers. The points for the transfers are represented in the right panel of Figure 8. A Thin Plate Spline (TPS) kernel is used for RBFs, which consists in $r^2 \log(r)$ functions, and the interpolants are computed with our dedicated MPI LU solver. Note that for transferring loads to the structure, we exclude forces acting on the fuselage and tail.

Comparison between the computations with MIMAS and elsA's aeroelastic module are shown in Figure 8 for a Mach 0.85 and an angle of attack of 1.32° at altitude 36000 feet. The pressure distribution for this operating point obtained by MIMAS is also shown in the left panel. The agreement between both computations is excellent in terms of vertical displacement, although the mesh deformation (TFI/IDW method) and transfer methods are not exactly implemented in the same way. We check also that the forces transferred to the structural grid are very similar. Note that with MIMAS, this calculation has been improved to take into account the full finite element model, using the nonlinear NASTRAN solver (SOL 400). For the selected flight condition considered here, no significant differences are however observed with respect to the statically condensed linear model.

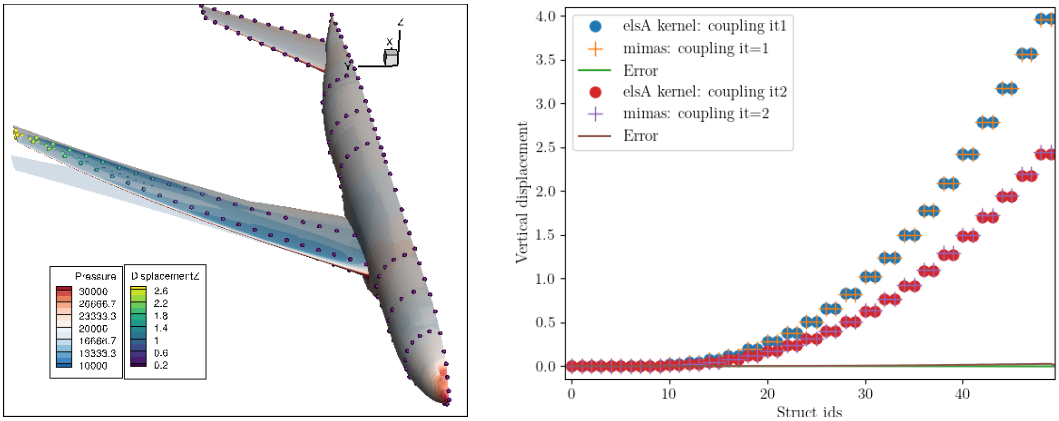


Figure 8. Illustration of the CRM static coupling. Left: Pressure field on the plane with condensed structural points (force and displacements nodes). These points are coloured with the vertical displacement. Right: Comparison of the vertical displacement along the span between elsA's legacy aeroelastic module and MIMAS at first and second coupling iterations.

3.2. *Harmonic forced motion*

Validation of the modular approach has been performed in the unsteady case for forced harmonic motion simulations. The test case is a state-of-the-art fan blade representative of a modern Ultra High Bypass Ratio (UHBR) engine, presented in Figure 9. In these simulations, a natural mode of the structure is excited at given frequency(ies) with a relatively small amplitude to characterize the aerodynamic response and ultimately the generalized aerodynamic forces. The linear unsteady aerodynamic response is computed using CFD codes (such as elsA) and can be re-injected into a structural linear solver to assess the aeroelastic stability (useful for flutter prediction). Mesh deformation is generally computed once, during a preprocessing step stage, and the corresponding mesh deformation associated to the modal shapes is interpolated in time using a sinusoidal law. In the modular approach, the coordinates of the deformed mesh, as well as the grid velocity are calculated outside the CFD code.

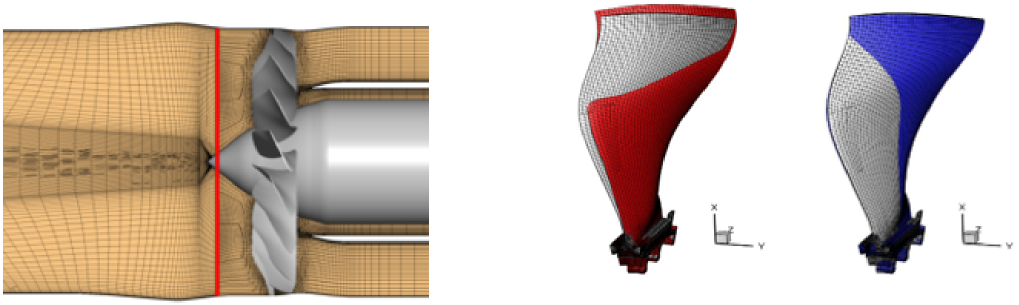


Figure 9. Ultra High Bypass Ratio (UHBR) fan blade test case and real/imaginary mode shapes of interest (deformation artificially amplified for visualization).

For the UHBR fan, a modal basis for the structure is first pre-computed using SAMCEF. Data of the modal shape (displacements) are transferred from the SAMCEF grid to the CFD grid by MIMAS using Radial Basis Functions during the pre-computation phase. We then select one particular mode (here the first bending mode 1F-0D with zero dephasing between the blades) and apply small oscillations to this mode with a given frequency, modal amplitude of 0.3 mm and initial phase of $\pi/2$. Note that with the new mesh deformation algorithms implemented in MIMAS based on IDW method, amplitude up to 4 mm in azimuth of the fan-blade can be reached, which is almost two times the tip gap size (between the shroud and the blade). The slipping condition on the shroud is handled with a directional damping implemented specifically for IDW.

For MIMAS, we use again elsA as an external solver for the CFD calculation. Only one sector is simulated in order to save computational time and periodic conditions are used in azimuth. The mesh is composed of approximately 3.2 million points, divided into 48 blocks. Note that modal shapes with dephasing between the blade can also be simulated in the modular aeroelastic environment, using only one sector. For that, a complex representation of the mode is used, in addition to the so-called phase-lagged condition in elsA. MIMAS mesh deformation module is able to deal with such complex deformation. Figure 10 (left) shows the aerodynamic forces projected onto the structural mode of vibration (GAF), first computed with the aeroelastic module of elsA (blue) and second computed with MIMAS (orange). Clearly, both solutions match perfectly, which validates our modular approach. Note that the same preprocessed deformed mesh for the modal shape has been used in both calculations using the elastic analogy method (*move3d*). With the modular implementation, other mesh deformation methods are available and have been compared in this context. The phase portrait (Lissajous) of the GAF is compared on the right panel of Figure 10 for the last cycle of vibration: a relatively good agreement is obtained when using TFI, IDW or the quaternion approach (Quantum), although it is interesting to notice slight variation of the Lissajous sizes when the deformation algorithm changes. Variations up to 15 % can be measured on the aerodynamic damping of the mode according to the deformation method.

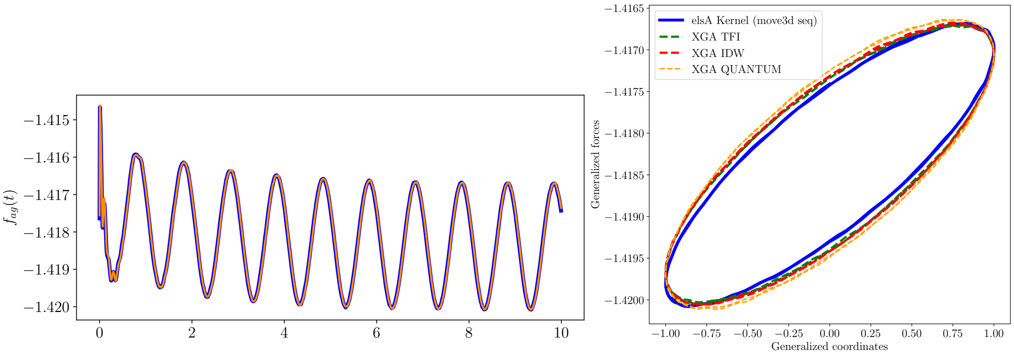


Figure 10. Comparison of elsA's aeroelastic module and MIMAS modular solution for forced motion using the same deformed mesh (left) or alternative external mesh deformation algorithms (right). On the left panel are shown the generalized aerodynamic forces as a function of time. On the right, are shown the Lissajous curves (phase portrait) of the aeroelastic response.

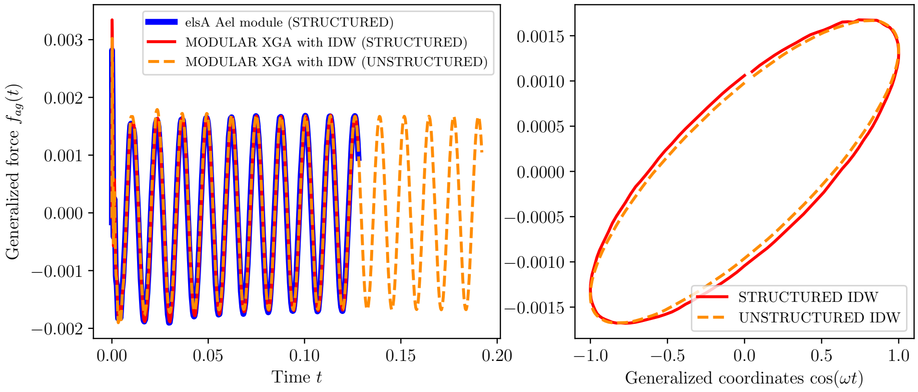


Figure 11. Comparison of forced motion aeroelastic computation with structured or unstructured (destructured) mesh.

Another key advantage of MIMAS compared to elsA's aeroelastic module is the ability to conduct aeroelastic simulations on unstructured grids. In this example, the structured grid was "destructured" for comparison purposes, using the same hexahedral mesh but treated as unstructured. The results match relatively well (see Figure 11) between the two grid representations. Here, the IDW mesh deformation method was applied since the structural analogy approach has not yet been implemented in the modular framework for unstructured meshes.

3.3. *Dynamic coupling with linear structure*

Finally, we demonstrate that MIMAS can reproduce fully coupled dynamical calculations under the assumption of a linear structure, a capability that was previously achievable with elsA aeroelastic module. In this type of simulation, the linear dynamic behavior of the structure is represented by a limited set of eigenmodes $\Phi = [\phi_1, \ldots, \phi_r]$. The displacement is thus approximated as a linear combination of these modes $U = \Phi q$ and the dynamic equation of motion is simplified through projection into r equations:

$$\mu \ddot{q} + \beta \dot{q} + \gamma q - f_{\text{ag}}(W) = 0.$$

Here, q are the unknown generalized coordinates, μ, β, and γ represent the generalized mass, damping, and stiffness matrices, respectively, while $f_{\text{ag}} = \Phi^T f_a$ denotes the vector of generalized aerodynamic forces. This set of equations is solved externally in MIMAS using a Newmark algorithm for the generalized coordinates q, from which the physical displacements can be derived. As mentioned in Section 2.6, the coupling procedure involves mechanical steps and mesh deformation in the implicit dual cfd loop.

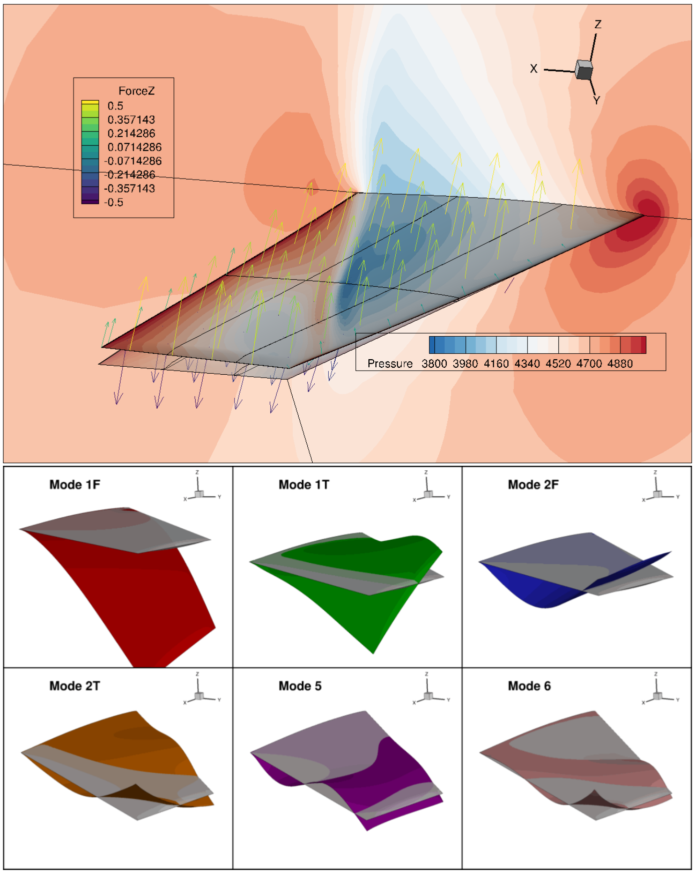


Figure 12. AGARD [54] dynamic coupled simulation — Pressure distribution and deformation of the wing at $t = 3T$ (top) and the six first modes shapes used for projecting the dynamics (bottom).

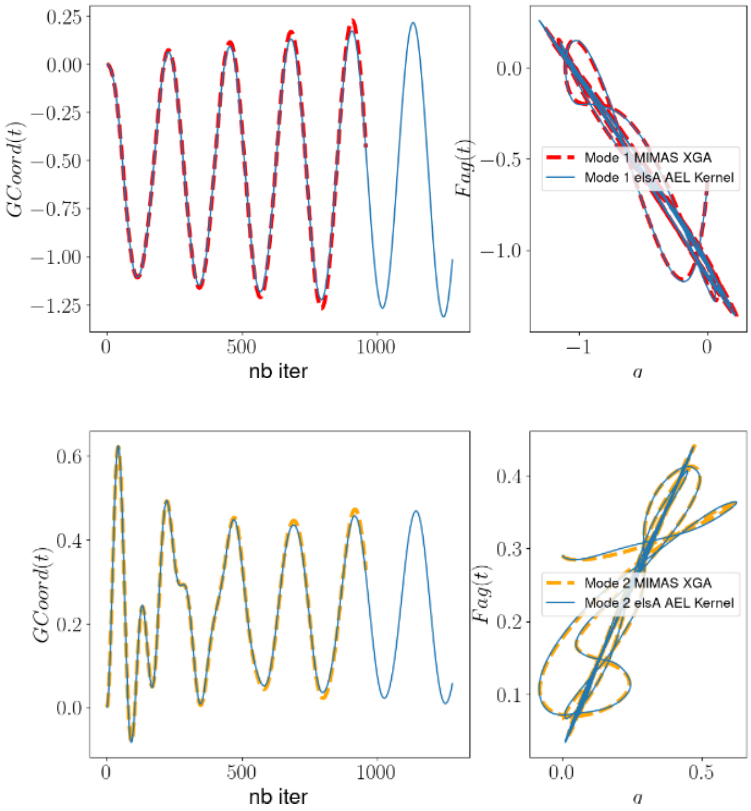


Figure 13. AGARD dynamic coupled simulation — Comparison of the dynamic response of the first two structural modes between MIMAS calculations (dashed lines) and elsA AEL module (plain lines). On the left panel are plotted the generalized coordinates as a function of physical time iterations. On the right panel is shown the phase portrait of the dynamical motion (generalized force vs. generalized coordinates).

For validation, we run a coupled simulation of the AGARD 445.6 wing configuration using either MIMAS or the elsA aeroelastic module. We adjusted the structural model to exhibit an unstable behavior. The flow is inviscid and transonic (Ma = 0.96, AoA = 1°), while the dynamic of the structure is modelled using the first six eigenmodes. The shapes of the modes are illustrated in the bottom panel of Figure 12. The simulation is started from a steady equilibrium state where a perturbation is introduced and run on more than 1000 time steps. On the top of the figure, we show the resulting wing pressure distribution at $t = 3T$ (where T is the pseudo-period of the dynamics) and the wing geometry at initial time and $t = 3T$. Figure 13 shows a comparison of the generalized coordinates between MIMAS (dashed lines) and elsA's legacy reference solution (plain lines). The agreement between the original reference solution and the modular one is very good, although the coupling process is slightly different and the techniques for transferring fields, deforming the mesh are also different. Note that the signal is increasing with time due to the negative damping induced by the aerodynamic loads on the structure.

3.4. *Overheads and difference with elsA Ael module*

To demonstrate that our modular implementation does not add any overhead compared to the original implementation, we show in Table 1 the user time spent on a single node (24 cores) of ONERA's development cluster for elsA aeroelastic module (AEL) and MIMAS. Clearly for all cases, the overhead is not significantly enhanced and remains under the noise of the cluster node performance. We also show differences in terms of displacement and GAFs between the reference AEL module and MIMAS. For the static case, we observe 0.012 % of error in terms of maximum vertical displacement on the plane wing, while for the turbomachinery case, the difference in generalized aerodynamic force is below 1e-3 %. For the dynamic case, errors are larger due to the fact that the numerical temporal coupling scheme is different (see Section 2.6).

Table 1. User times and differences in quantities of interest compared to elsA aeroelastic module implementation.

	User time in elsA AEL (s)	**User time in MIMAS (s)**	**Difference in %**
Static coupling (CRM)	2.156+05	1.726e+05 s	0.012 (z-disp)
Harmonic forced (UHBR)	3.593e+05 s	3.634e+05 s	1e-3 (GAF)
Dynamic coupling (AGARD)	1.35e+04	1.424e+04 s	1e-1 (GAF)

4. Advanced aeroelastic applications with MIMAS

4.1. *The DLR-F25 plane case: CODA/NASTRAN coupling with HPC mesh deformation*

A first demonstration overcoming the conventional capabilities of elsA has been achieved with MIMAS on the DLR-F25 plane. This plane model is widely used as a benchmark in multi-disciplinary optimization and aeroelastic simulations. The goal of the calculation is to evaluate the static deformed shape of the plane during flight condition. The different improvements compared to previous standard calculation are the use of:

- an unstructured grid with tetrahedron, hexahedron and pyramids (see Section 2.2);
- a fast IDW method using clustering and layering (see Section 2.3);
- a compact RBF transfer method for displacements using the MUMPs library for solving the RBF system (see Section 2.4);
- a cleverer time-relaxation scheme using Aitken method to speed up convergence (see Section 2.6);
- a HPC CFD code of last generation (CODA) for the fluid step computation (see Section 2.8);
- a non-linear structural solver (here NASTRAN) directly exchanging data with the CFD code through the coupling tree of MIMAS (see Section 2.7).

Simulations at various angles of attack (from $-1°$ to $2°$) were conducted using both CODA and elsA within the MIMAS modular environment. An identical unstructured mesh of 10 million cells was used for both codes, and a similar numerical setup have been selected in order to be consistent for the comparison. The Roe solver is used in both codes, although in CODA, we employed the *spline quintic* flux limiter with a linear reconstruction, whereas for elsA we adopted a *minmod* limiter due to stability issue. For the structural solver, both linear and non-linear NASTRAN models have been considered, although the aim of this paragraph is not to show the differences between them (a complementary study would be more appropriate for that).

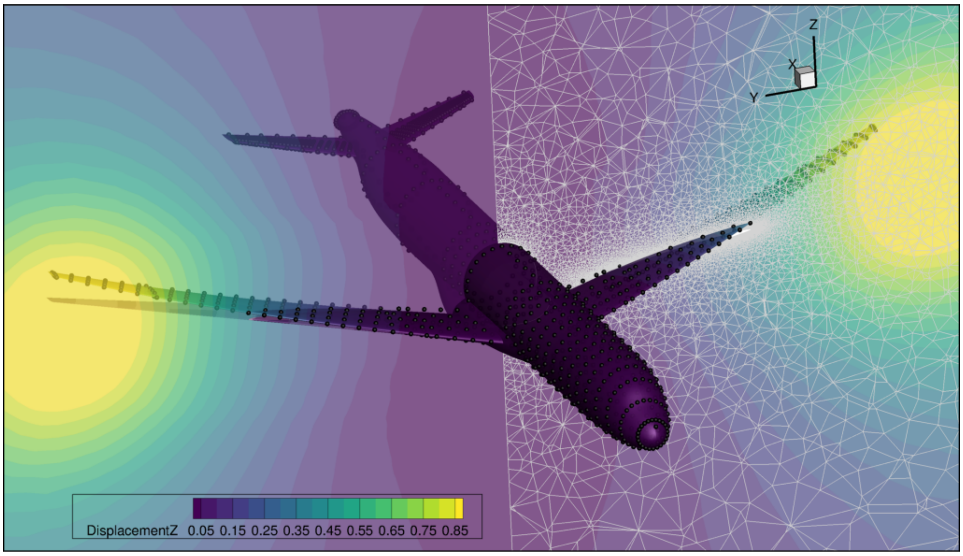


Figure 14. Mesh deformation of the DLR-F25 plane using IDW and clustering method (for angle of attack of 2°). Colours show the vertical displacement in the volume generated by the IDW algorithm on a 10-million-cells unstructured mesh composed of tetrahedrons, wedges, pyramids and hexahedrons.

Before running the aeroelastic calculations, tests have been performed to assess the performance and reliability of the new clustering IDW deformation method for unstructured meshes. Note that the elementary surfaces of the cells have been included in the IDW weights, to improve mesh quality. Rotation of cells are however not enabled here. Figure 14 shows the vertical deformation in the volume generated by the wall surface displacements. We checked that no negative cells are generated and that the mesh conserves a good quality during its deformation. Table 2 summarizes the performance of the method compared to the original brute force algorithm implemented in elsA aeroelastic module.

Table 2. User time for IDW mesh deformation algorithms on a single node (with 96 cores) of new generation ONERA's cluster. The first line corresponds to the time for preparation and computation during the first call of the algorithm. The second line is for the second coupling iteration until the end of the calculation.

	Brute force IDW (elsA)	**Clustered IDW (MIMAS)**
1st deformation including preparation	283 s	132 s
Next deformations	189 s	31 s

Clearly, using the clustering method for static calculations reduces computation costs to one-sixth of those required by the brute force algorithm, while achieving correct deformation. The preparation phase which includes the clustering procedure does not add any significant overhead. The mesh is deformed in almost 30 s at each coupling iteration. For comparison, fluid iteration steps in elsA last 225 s in average between each coupling. In CODA the time between these couplings varies strongly during the convergence history, and depends strongly on the strategy used for convergence. The time spent on deforming the mesh remains thus around 10 % of the total time spent for the fluid resolution during each coupling. With old methods, this would have exceeded 100 % of the CFD time.

We further investigate elsA and CODA runs at an angle of attack of 2°. Different coupling strategies have been tested: the first one relies on a fixed coupling period and relaxation parameter equal to 0.75. Coupling is done after each 50 fluid iterations for CODA and 150 for elsA. In that case, the CODA run required in total 1300 fluid iterations using the GMRES method to reduce the density residual below 10^{-7}. For elsA, 5000 iterations with scalar LU relaxation were needed to reach a residual threshold of 10^{-5}. The second strategy relies on an Aitken under-relaxation scheme. By using this coupling strategy, no real improvement of the convergence has been achieved. The explanation might be that the convergence of the coupled system is primarily driven by the fluid and less affected by the structure because the wing flexibility remains low. Note that on contrary the convergence of the CRM plane (cf. Section 3.1) is strongly boosted by the Aitken under-relaxation scheme, probably because the flexibility is larger and the fluid Euler model converges intrinsically faster, so that the structure has more time to evolve during each coupling iteration.

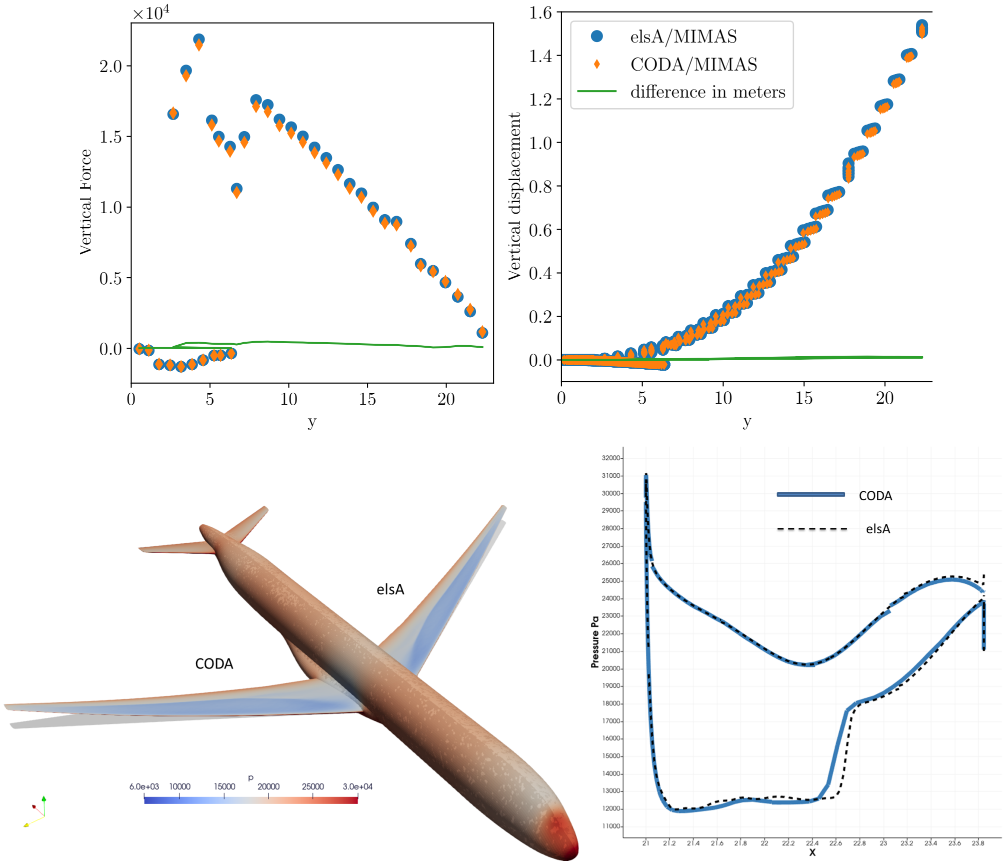


Figure 15. Comparison of the static simulation between elsA and CODA for the CFD part. Left top: Span distribution of vertical force on structural nodes. Right top: Distribution of vertical displacement on structural nodes. Left bottom: Pressure distribution on the plane with deformed shape. Right bottom: Pressure profile along the wall at the wing's half-span.

Figure 15 presents a comparison of the evolution along the span of the vertical displacement and force acting on the structural nodes when convergence is reached. There is a quite good agreement between the two runs, although the force amplitude close to the shock appears to be slightly smaller in CODA. The wing tip deflection is about 1.534 m for elsA and 1.52 m for CODA, which implies a 1.3 cm difference (≤ 1 % of error). For further details on the simulations, we plot in the bottom panel of Figure 15 the pressure field on the plane for both runs (left side of the plane corresponds to CODA, right is elsA) and a comparison of the pressure profile along the wall at the wing's half-span. The shock intensity and position are relatively similar but the shock appears a bit smearer in the CODA run. This is possibly due to the fact that the order of reconstruction of the flux and the type of limiter employed in CODA differ from those used in elsA.

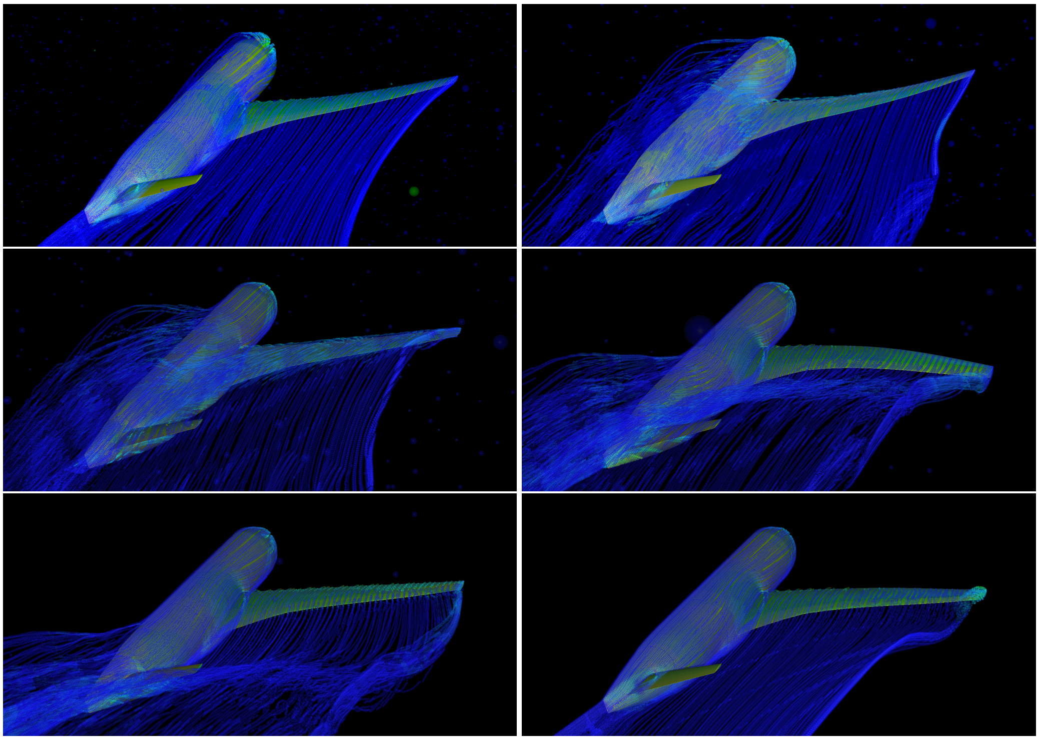


Figure 16. Flexible gust dynamic simulation of the CRM plane that couples the CFD code elsA and the CSM code NASTRAN in real-time, with mesh deformation at every time-step. Each code runs on a different machine, and data are transferred between them remotely using the Pyro Python module. The frames are respectively taken at 0.36, 0.63, 0.75, 0.92, 1.01 and 1.7 seconds after the launch of the gust. The blues lines represents the unsteady streamlines of the flow, computed in real-time during the calculation through a dedicated particle solver, independent of the CFD code.

4.2. *Dynamic aeroelastic response of the CRM aircraft subjected to gust loads*

Another new possibility with MIMAS is to couple directly the CFD code and the CSM code dynamically, while performing a mesh deformation at each time-step (or at regular dual steps of the cfd solver, see Section 2.6) to take into account the current structural displacements.

This is made possible with the new and faster mesh deformation algorithms, which absorb higher deformation. We show in this section a dynamical coupling achieved between elsA and NASTRAN, using the OpenFSI module of NASTRAN fully integrated within MIMAS. As both codes run on distant separate machines, we use the Python module Pyro to enable efficient in-memory data communication between solvers. The simulation presented here shows the response to a vertical negative wind gust characterized by a one-minus-cosine law, with an amplitude of 35 m and a wavelength of 50 m. The gust velocity is set to 100 m/s relative to the aircraft. The gust is imposed to the CFD code by adding to the mesh deformation a positive vertical grid velocity depending on time and space; this setup provides a rather strong gust that is largely above the certification test. However, our goal here is not to replicate actual aircraft tests, but rather to demonstrate MIMAS's capability to perform such coupling and to verify the robustness of the deformation algorithms when large displacements are involved. Both fluid and structural models are detailed in Section 3.1. For the NASTRAN model we restrict the simulation to a linear structure, but nothing prevents to run a nonlinear model, except the CPU time which can be significantly larger. Figure 16 shows a series of simulation snapshots, with the rendering enhanced by a particle tracer illustrating the flow streamlines (in blue). The effect of the gust is clearly visible on the third, fourth and fifth snapshots. The maximum displacement obtained on the wing tip is about 4 m (for comparison the static deformation at equilibrium at the wing reached 2.4 m) and −1.5 m at the gust maximum. To validate our direct coupling methodology, we compared it against a dynamic modal simulation, a coupling scenario which is already well-established and has been validated for the AGARD wing (see Section 3.3). In this reference case, the structural dynamics is restricted to a set of 10 modes, primarily representing the wing's bending and torsional behavior, computed in a pre-process phase. Note that for this reference run, the gravity and aerodynamic loads are directly projected onto the modal basis expressed on the fluid grid, while for the run in Figure 16, the forces and displacements are transferred at each time step using nearest neighbor algorithm and thin plate spline kernel. Figure 17 compares the two simulations in terms of lift coefficient and structural displacement at wing tip. The dynamic response is nearly identical, despite the fact that the underlying numerical algorithms used for coupling are fundamentally different.

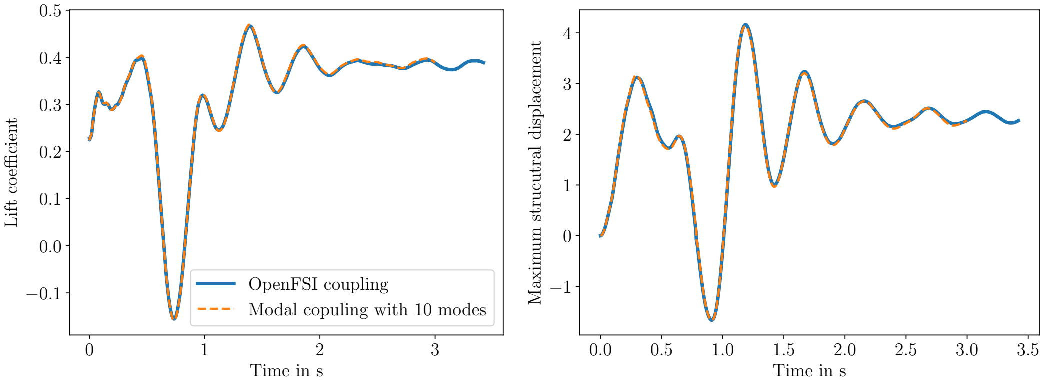


Figure 17. Comparison of the lift coefficient and wing-tip displacement between two gust aero-elastic simulations: one performed using a direct coupling between the CFD and CSM codes (with mesh deformation and data transfer at every time step), and the other using a modal coupling approach, similar to that presented in Section 3.3.

4.3. *The ERATO blade case: non-linear reduced order structural models coupled with CFD and VPM*

The final application presented in this paper focuses on the aero-acoustically optimized helicopter blade ERATO (Étude d'un Rotor Aéroacoustique Technologiquement Optimisé). The ERATO geometry was designed primarily by ONERA and the DLR in the '90s. Experimental studies were carried out in 1998 to aerodynamically characterize the blade for different advance ratio conditions and hover in ONERA's wind tunnel in Modane. The blade has a double sweep configuration to mitigate the effects of compressibility. Traditionally, blade structures are modelled by 1D approaches (rigid elements with stiff connections, FE beam formulation, etc.) and different studies have proven that for non-disruptive panforms the 1D structural modelling is accurate enough to represent the aeroelastic behaviour [55]. However, for ERATO like complex planform, even if the overall beahviour of the blade is captured, detailed studies highlighted the need for more advanced structural models [56] to correctly assess its aeroelastic behaviour. Recently, fully-coupled aeroelastic study with refined Reduced Order Models (ROMs) of the blade based on FEM in hover flight has been achieved with MIMAS [57].

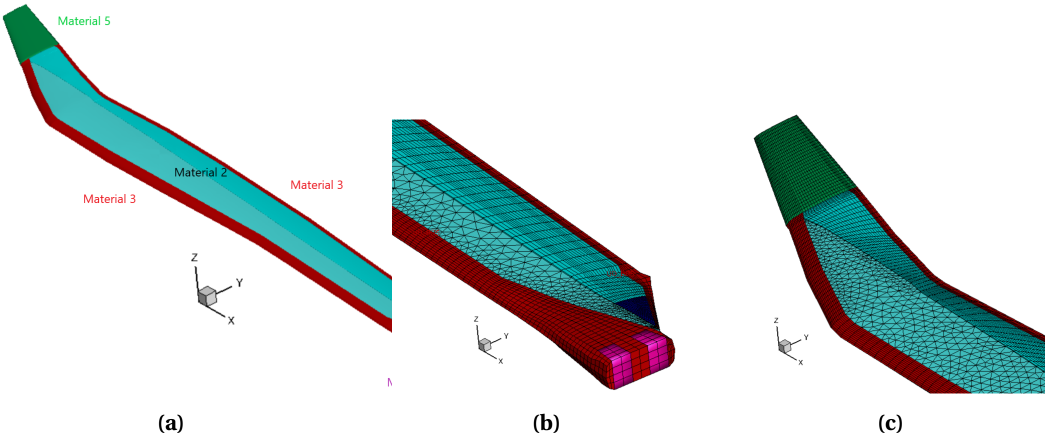


Figure 18. (a) Structural model of the ERATO blade used in this study. (b) Root mesh discretization. (c) Tip mesh discretization.

A detailed view of the root and the tip of the ERATO's structural mesh is presented in Figure 18. The mesh is formed by linear hexahedral, tetrahedral and pyramid FE elements and contains 32 436 nodes. The blade is composed of five different materials (represented by different colors) and is assumed to be clamped at the root. The blade structure is resolved through a geometrically nonlinear reduced model (ROMs) based on ICE projection (see Section 2.7). The construction of the non-linear model is carried out with the finite element software *Code_Aster* during a preprocess stage. To solve the fluid around the blade, several methods including URANS and mid-fidelity methods have been employed, with the aim to compare them. The most expensive computation is obviously the one performed with the elsA code using an overset technique. In that case, a Gear time algorithm is used with a Jameson, Schmidt and Turkel JST centred spatial discretization scheme (including artificial viscosity) and a Wilcox turbulence model for the hover flight. A less expensive method is the Vortex Particle Method (VPM) which is described in Section 2.9.

We summarize below some of the results conducted in this study. Figure 19 shows a snapshot of the VPM simulation during the coupling process. Approximately 30 new particles are launched

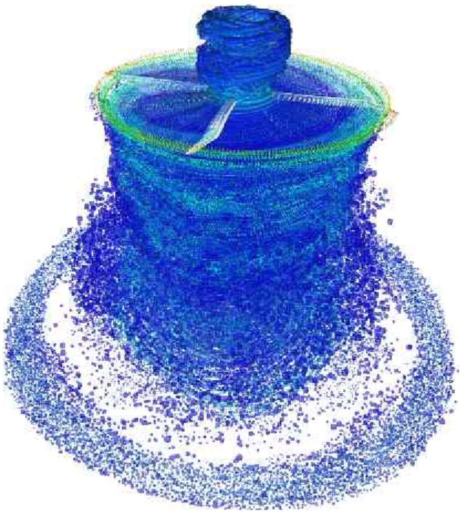
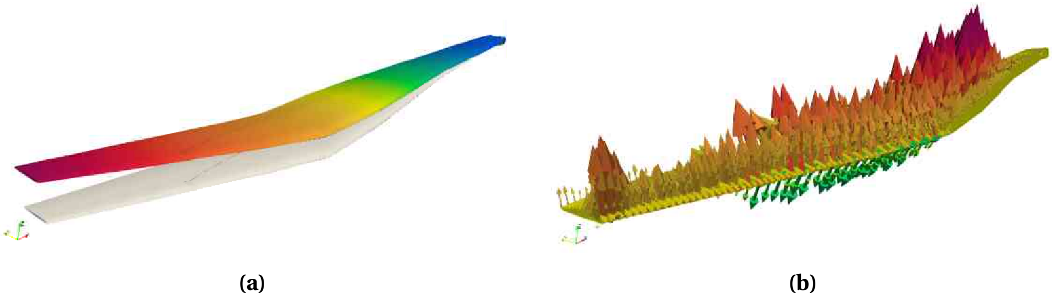


Figure 19. Snapshot of the wake generated by the VPM ONERA's code around ERATO's helicopter blades. The flow is represented by particles shed in the wake during the coupling process. The code has been run on a single GPU of the ONERA's development cluster.



(a) **(b)**

Figure 20. (a) Illustration of displacement transferred on the ERATO blade with the RBF algorithm, resulting from the non-linear reduced structural model. (b) Illustration of forces transferred on the ERATO blade with the nearest neighbour algorithm, resulting from the VPM computed forces.

from the chord quarter at relatively regular times, depending on the spatial and time variations of circulation. As explained in Section 2.9, the velocities of particles simulated in the wake are feeding back a lifting line model, which in turns change the particles distribution. CFD/ROMs and VPM/ROMs analyses show similar trends in terms of provided thrust, torsional deformation (pitch-down), flap displacements, aerodynamic loadings and maximum displacements, which suggest that mid-fidelity aerodynamic method can be employed to assess the aeroelastic behaviour of complex helicopter blades. Figure 20(a) shows that the ERATO blade in hover flight is significantly impacted by the torsional aeroelastic behaviour leading to a pitch-down effect on the blade-tip. In particular, simulations seem to indicate that the blade-vortex interaction is reduced when the flexibility of the blade is fully taken into account. Figure 20(b) illustrates the distribution of vertical loads on the structural nodes during the coupling process. Note that non-linear mechanical effects tend to slightly reduce the final blade deformation, although a more exhaustive study is required to characterise in details the non-linear physics.

5. Perspectives & Conclusions

This paper presents MIMAS, a new tool developed at ONERA designed to perform next-generation of aeroelastic simulations. MIMAS has been developed alongside the new emerging HPC codes such as CODA and SoNICS, with the aim of externalizing various simulation components that were traditionally embedded within elsA's fluid solver kernel. Among these components are the mesh deformation and data transfer algorithms, which have been greatly improved and parallelized to fit with the current HPC standard. Results were presented to demonstrate the reliability of this modular implementation across a wide range of test cases with multiple levels of complexity, including static coupling, forced harmonic motion, and dynamic coupling simulations. This modular framework also currently offers new capabilities, such as compatibility with unstructured meshes, which are essential for next-generation codes like CODA and SoNICS. Additionally, it offers several ways for coupling fluid with non-linear structures, and proposing new implementation of time-marching algorithm dedicated to FSI problems. The library supports now full dynamic coupling with linear or nonlinear structural models and codes (such as NASTRAN), extending its ability to simulate complex unsteady phenomenon with large deformations.

Future development within this library includes several key research areas aimed at enhancing its capabilities and performance. Firstly, efforts will focus on the linearization and adjoint formulation of all functions (or operators), allowing for more efficient optimization and sensitivity analysis. In addition, performance improvements will continue, particularly in the areas of mesh deformation and data transfer, which are critical for handling increasingly complex configurations. For elastic analogy in mesh deformation, an unstructured linear FEM code is under development and could progressively replace the structured version, with better preconditioners to accelerate its convergence. Interpolation methods will also continue to be refined, with a focus on blending techniques that take into account real intersections to improve accuracy. These techniques could also be upgraded to incorporate projection method based on 3D finite elements. Another important direction of research is the improvement of time-coupling schemes, which will allow more accurate and efficient simulations of dynamic systems. As the library evolves, it will become fully compatible with new HPC architectures such as those encountered in SoNICS and for example extend its capability to GPU execution. Extensions to other physics are also not excluded. Current developments, together with future improvements, will hopefully push the boundaries of what is possible in high-fidelity aeroelastic simulations.

Acknowledgments

The DLR-F25 is a conceptual design of an advanced short-to-medium-range airliner provided by DLR. It features an initial shape of a very high-aspect-ratio wing and was developed as a challenge for high-fidelity disciplinary and multi-disciplinary preliminary design. The development was funded by the German Federal Ministry for Economic Affairs and Energy as part of the LuFo project VIRENFREI.

Appendix A. Errors generated by clustering in the IDW formula

The purpose of this appendix is to determine the error bound made on the IDW formula when summing over the inverse distance of clusters barycentre.

Consider a set S of source points M_i and a set C consisting of N_c clusters of these points (see Figure 21). We assume that each point M_i belongs to a unique cluster. Each cluster is denoted by C_k, where k is an integer between 1 and N_c, and its cardinal is denoted n_k.

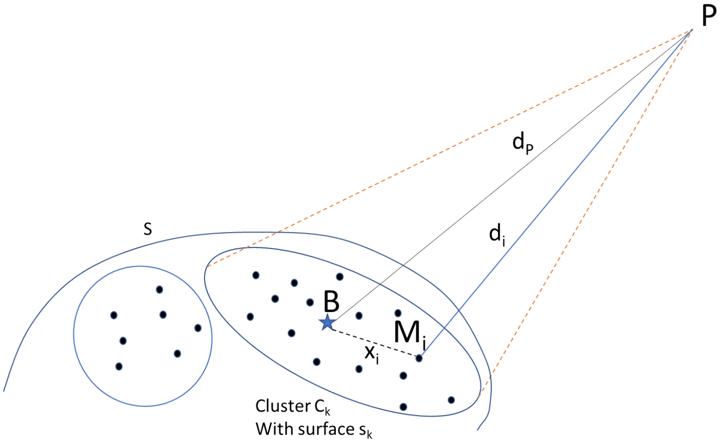


Figure 21. IDW clustering approximation.

For any summation over S, we can rearrange the terms as follows:

$$\sum_{i \in S} w_i = \sum_{k=1}^{N_c} \left(\sum_{j \in C_k} w_j \right). \tag{11}$$

We note with a bar and index k the average of a field over a cluster, then we have for any field w_i:

$$\sum_{i \in S} w_i = \sum_{k=1}^{N_c} n_k \overline{w}_k. \tag{12}$$

For a given cluster C_k, we introduce the dimensionless field dispersion ϵ_w with respect to the cluster barycentre (or average defined above) so that for each point in the cluster we have:

$$w_i = \overline{w}_k (1 + \epsilon_{w\,i}). \tag{13}$$

It comes immediately that:

$$\sum_{j \in C_k} \epsilon_{w\,j} = 0 \qquad \text{and} \qquad \sum_{i \in S} \epsilon_{w\,i} = 0. \tag{14}$$

By using the above relations, the product of two fields w_i and D_i in the summation over the cluster leads to an additional correction term which depends on the product of the two dispersion fields:

$$\sum_{i \in S} w_i D_i = \sum_{k=1}^{N_c} n_k \overline{w}_k \overline{D}_k \left(1 + \sum_{j \in C_k} \epsilon_{w\,j} \epsilon_{D\,j} \right). \tag{15}$$

Now dividing (15) by (12) gives:

$$\frac{\sum_{i \in S} w_i D_i}{\sum_{i \in S} w_i} = \frac{\sum_{k=1}^{N_c} n_k \overline{w}_k \overline{D}_k \left(1 + \sum_{j \in C_k} \epsilon_{w\,j} \epsilon_{D\,j}\right)}{\sum_{k=1}^{N_c} n_k \overline{w}_k}. \tag{16}$$

We recognize on the left the exact IDW formula. On the right the approximated IDW formula computed from cluster barycentre is recovered for zero dispersion ($\epsilon_{w\,j} = \epsilon_{D\,j} = 0$). The relative error $\mathscr{E}$ on the formula is then bounded by:

$$\mathscr{E}_{\max} = \max_{k=1,\ldots,N_c} \left| \sum_{j \in C_k} \epsilon_{w\,j} \epsilon_{D\,j} \right|. \tag{17}$$

The next step would then be to give an estimation of the upper bounds for ϵ_w and ϵ_D. However, in practice, this expansion cannot be used straightforwardly since the average of the IDW weight $\overline{w}_k$

cannot be computed for each point P in the volume without high computational cost. We search therefore for an approximation of the IDW formula that depends only on the distance from P to the clusters barycentre.

For an arbitrary point in space P, the IDW weight is:

$$w_j = \frac{s_j}{d_j^\alpha} \tag{18}$$

where

$$d_j^2 = \bar{\mathbf{x}}_P^2 + \mathbf{x}_j^2 - 2\mathbf{x}_P \cdot \mathbf{x}_j \tag{19}$$

is the distance from P to the point M_i in the cluster, $\bar{\mathbf{x}}_P$ is the position vector of P with respect to the cluster barycentre and s_i is an additional surface element relative to the source. In the following we note $d_P = \sqrt{\bar{\mathbf{x}}_P^2}$ the distance of P from the barycentre. Assuming that the squared distance inside the cluster $\mathbf{x}_j^2$ is small compared to d_P^2, we apply a Taylor expansion to compute the inverse distance:

$$\begin{aligned} \frac{1}{d_j^\alpha} &= \frac{1}{d_P^\alpha}\left(1 + \frac{x_j^2}{d_P^2} - 2\frac{\mathbf{x_j} \cdot \mathbf{x}_P}{d_P^2}\right)^{-\alpha/2} \\ &= \frac{1}{d_P^\alpha}\left(1 + \alpha \mathbf{x_j} \cdot \frac{\mathbf{x}_P}{d_P^2} - \frac{\alpha}{2}\frac{\mathbf{x}_j^2}{d_P^2} + \frac{1}{2}\alpha(\alpha+2)\left(\frac{\mathbf{x}_j \cdot \mathbf{x}_P}{d_P^2}\right)^2\right). \end{aligned} \tag{20}$$

Summing this expression over the points in the cluster C_k and assuming that the elementary surface into the sum is almost uniform gives:

$$\begin{aligned} \overline{w}_k &= \frac{1}{n_k}\sum_{j \in C_k} \frac{s_j}{d_j^\alpha} \\ &= \frac{1}{n_k}\sum_{j \in C_k} \frac{s_j}{d_P^\alpha}\left(1 + \alpha \mathbf{x_j} \cdot \frac{\mathbf{x}_P}{d_P^2} - \frac{\alpha}{2}\frac{\mathbf{x}_j^2}{d_P^2} + \frac{1}{2}\alpha(\alpha+2)\left(\frac{\mathbf{x}_j \cdot \mathbf{x}_P}{d_P^2}\right)^2\right) \\ &= \frac{s_k}{d_P^\alpha}\left(1 + \frac{\alpha}{n_k}\left(\sum_{j \in C_k} \mathbf{x_j}\right) \cdot \frac{\mathbf{x}_P}{d_P^2 n_k} - \frac{\alpha}{2d_P^2}\sum_{j \in C_k} \mathbf{x}_j^2 + \frac{1}{2n_k}\alpha(\alpha+2)\sum_{j \in C_k}\left(\frac{\mathbf{x}_j \cdot \mathbf{x}_P}{d_P^2}\right)^2\right). \end{aligned} \tag{21}$$

Since the sum of dispersion vectors is zero, we end up with

$$\overline{w}_k = \frac{s_k}{d_P^\alpha}\left(1 + \frac{1}{2}\alpha(\alpha+2)\frac{1}{n_k}\sum_{j \in C_k}\left(\frac{\mathbf{x}_j \cdot \mathbf{x}_P}{d_P^2}\right)^2 - \frac{\alpha}{2d_P^2}\frac{1}{n_k}\sum_{j \in C_k} \mathbf{x}_j^2\right). \tag{22}$$

Assuming that $\mathbf{x}_j$ and $\mathbf{x}_P$ make an angle δ_j, we finally obtain:

$$\overline{w}_k = \frac{s_k}{d_P^\alpha}(1 + \epsilon_k), \tag{23}$$

with

$$\epsilon_k = \frac{\alpha}{2d_P^2}\frac{1}{n_k}\sum_{j \in C_k}\left[(\alpha+2)\cos^2(\delta_j) - 1\right]\mathbf{x}_j^2. \tag{24}$$

Since $0 < \left|\cos^2(\delta_j)\right| < 1$, an upper bound for ϵ_k is

$$|\epsilon_k| < \frac{\alpha(\alpha+1)}{2d_P^2}\frac{1}{n_k}\sum_{j \in C_k} \mathbf{x}_j^2. \tag{25}$$

We now use the fact that the clustering procedure ensures that the variance associated with coordinates and displacements inside each cluster is lower than a certain value σ_k. In other words, if we note $\mathbf{x}_j$ the position of a point M_i in the cluster relative to the barycentre, and $\mathbf{D}_j$ the displacement field, we have:

$$\frac{1}{n_k}\sum_{j\in C_k}\mathbf{x}_j^2 \le \sigma_k, \tag{26}$$

$$\frac{1}{n_k}\sum_{j\in C_k}(\bar{\mathbf{D}}_k - \mathbf{D}_j)^2 \le \sigma_k. \tag{27}$$

Using these constraints gives:

$$|\epsilon_k| < \frac{\alpha(\alpha+1)\sigma_k}{2d_W^2} \tag{28}$$

where d_W is the minimal distance from P to the source, which can be easily computed in a pre-process calculation. Finally, going back to eq. 16, we have

$$\frac{\sum_{i\in S} w_i D_i}{\sum_{i\in S} w_i} = \frac{\sum_{k=1}^{N_c} n_k \frac{\bar{s}_k}{d_{P_k}^{\alpha}} \bar{D}_k\big(1+\sum_{j\in C_k}\epsilon_{w\,j}\epsilon_{D\,j}\big)(1+\epsilon_k)}{\sum_{k=1}^{N_c} n_k \frac{\bar{s}_k}{d_{P_k}^{\alpha}}} \tag{29}$$

which leads to the following upper bound for the error

$$\mathscr{E}_{\max} = \max_{k=1,\dots,N_c}\left|\sum_{j\in C_k}\epsilon_{w\,j}\epsilon_{D\,j} + \frac{\alpha(\alpha+1)\sigma_k}{2d_W^2}\right|. \tag{30}$$

Neglecting the variance of the displacement inside each cluster or assuming that the average cross correlation between displacements and IDW weight is negligible, we end up with:

$$\mathscr{E}_{\max}(P) = \frac{\alpha(\alpha+1)\sigma_{\max}}{2d_W^2}. \tag{31}$$

Declaration of interests

The authors do not work for, advise, own shares in, or receive funds from any organization that could benefit from this article, and have declared no affiliations other than their research organizations.

References

[1] V. de Gaudemaris, J.-S. Schotté, A. Placzek, L. Blanc and F. Thouverez, “Unsteady aerodynamic modeling of whirl flutter on a bending wing”, *J. Phys. Conf. Ser.* **2647** (2024), no. 11, article no. 112013 (12 pages).

[2] V. de Gaudemaris, J.-S. Schotté, A. Placzek, L. Blanc and F. Thouverez, “Influence of aerodynamic modeling on the whirl flutter stability of a propeller under axial and non-axial flow conditions”, 2024. Online at https://conf.ifasd2024.nl/proceedings/documents/150.pdf. Conference paper: International Forum on Aeroelasticity and Structural Dynamics (IFASD 2024).

[3] C. Koch and B. Koert, “Including Blade Elasticity into Frequency-Domain Propeller Whirl Flutter Analysis”, *J. Aircraft* **61** (2024), no. 3, pp. 774–784.

[4] A. C. Gray and J. R. R. A. Martins, “A Proposed Benchmark Model for Practical Aeroelastic Optimization of Aircraft Wings”, in *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024.

[5] D. Gueyffier, S. Plot and M. Soismier, “SoNICS: a new generation CFD software for satisfying industrial users needs”, 2022. Online at https://hal.science/hal-04045165/document. Conference paper: OTAN/STO/Workshop AVT-366.

[6] L. Cambier, S. Heib and S. Plot, “The Onera *elsA* CFD software: input from research and feedback from industry”, *Mech. Ind.* **14** (2013), no. 3, pp. 159–174.

[7] S. Görtz, T. Leicht, V. Couaillier, M. Méheut, P. Larrieu and S. Champagneux, “CODA: A European Perspective for a Next-Generation CFD, Analysis and Design Platform”, 2022. Conference paper: NATO AVT-366 Workshop on Use of Computational Fluid Dynamics for Design and Analysis: Bridging the Gap Between Industry and Developers.

[8] A. Gopinath and A. Jameson, "Time Spectral Method for Periodic Unsteady Computations over Two- and Three-Dimensional Bodies", in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, 2005.

[9] N. L. Mundis and D. J. Mavriplis, "Toward an optimal solver for time-spectral fluid-dynamic and aeroelastic solutions on unstructured meshes", *J. Comput. Phys.* **345** (2017), pp. 132–161.

[10] C. Blondeau and C. Liauzun, "A modular implementation of the time spectral method for aeroelastic analysis and optimization on structured meshes", 2019. Online at https://hal.science/hal-02183133/document. Conference paper: International Forum on Aeroelasticity and Structural Dynamics (IFASD 2019).

[11] C. Liauzun and C. Blondeau, "A modular TSM solver for aeroelastic analysis and optimization", 2024. Online at https://conf.ifasd2024.nl/proceedings/documents/34.pdf. Conference paper: International Forum on Aeroelasticity and Structural Dynamics (IFASD 2024).

[12] G. Chourdakis, K. Davis, B. Rodenberg, et al., "preCICE v2: A sustainable and user-friendly coupling library", *Open Res. Eur.* **2** (2022), no. 51, pp. 1–47.

[13] M. Jadoui, *Solveurs de Krylov robustes pour la résolution partitionnée et monolithique du système adjoint couplé aéro-structure*, PhD thesis, Sorbonne Université (France), 2023. Online at https://theses.hal.science/tel-04336964v1/document.

[14] P. Girodroux-Lavigne, "Progress in steady/unsteady fluid-structure coupling with Navier–Stokes equations", *ONERA: Tire a Part* **1** (2005).

[15] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk and J. J. Alonso, "SU2: An Open-Source Suite for Multiphysics Simulation and Design", *AIAA J.* **54** (2016), no. 3, pp. 828–846.

[16] B. Hallissy and C. E. Cesnik, "High-fidelity Aeroelastic Analysis of Very Flexible Aircraft", in *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, American Institute of Aeronautics and Astronautics, 2011.

[17] T. A. Guimarães, C. E. Cesnik and I. V. Kolmanovsky, "An Integrated Low-Speed Aeroelastic-Flight-Dynamics Framework for Modeling Supersonic Aircraft", in *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, 2022.

[18] A. Dugeai and P. Vuillemin, "Highly flexible aircraft flight dynamics simulation using CFD", 2024. Online at https://hal.science/hal-04645957/document. Conference paper: International Forum on Aeroelasticity and Structural Dynamics (IFASD 2024).

[19] W. Liu, A. Skillen and C. Moulinec, *ParaSiF_CF: A Partitioned Fluid-Structure Interaction Framework for Exascale*, techreport, 2022.

[20] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore and B. A. Naylor, "OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization", *Struct. Multidiscip. Optim.* **59** (2019), no. 4, pp. 1075–1104.

[21] K. Boopathy and G. J. Kennedy, "Parallel Finite Element Framework for Rotorcraft Multibody Dynamics and Discrete Adjoint Sensitivities", *AIAA J.* **57** (2019), no. 8, pp. 3159–3172.

[22] J. R. Levesque, "The Code Aster: a product for mechanical engineers", *Epure* **60** (1998), pp. 7–20.

[23] E. Luke, E. Collins and E. Blades, "A fast mesh deformation method using explicit interpolation", *J. Comput. Phys.* **231** (2012), no. 2, pp. 586–601.

[24] N. Barral, E. Luke and F. Alauzet, "Two Mesh Deformation Methods Coupled with a Changing-connectivity Moving Mesh Method for CFD Applications", *Procedia Eng.* **82** (2014), pp. 213–227.

[25] N. R. Secco, G. K. W. Kenway, P. He, C. Mader and J. R. R. A. Martins, "Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization", *AIAA J.* **59** (2021), no. 4, pp. 1151–1168.

[26] P. Coulier and E. Darve, "Efficient mesh deformation based on radial basis function interpolation by means of the inverse fast multipole method", *Comput. Methods Appl. Mech. Eng.* **308** (2016), pp. 286–309.

[27] R. P. Dwight, "Robust Mesh Deformation using the Linear Elasticity Equations", in *Computational Fluid Dynamics 2006* (H. Deconinck and E. Dick, eds.), Springer, 2009, pp. 401–406.

[28] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data", in *ACM '68: Proceedings of the 1968 23rd ACM national conference* (R. B. Blue and A. M. Rosenberg, eds.), ACM Press, 1968, pp. 517–524.

[29] P. A. Burrough, R. A. McDonnell and C. D. Lloyd, *Principles of Geographical Information Systems*, Oxford University Press, 2015.

[30] W. J. Gordon and C. A. Hall, "Construction of curvilinear co-ordinate systems and applications to mesh generation", *Int. J. Numer. Methods Eng.* **7** (1973), no. 4, pp. 461–477.

[31] L. E. Eriksson, "Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation", *AIAA J.* **20** (1982), no. 10, pp. 1313–1320.

[32] S. Sen, G. De Nayer and M. Breuer, "A fast and robust hybrid method for block-structured mesh deformation with emphasis on FSI-LES applications", *Int. J. Numer. Methods Eng.* **111** (2017), no. 3, pp. 273–300.

[33] J. Batina, "Unsteady Euler airfoil solutions using unstructured dynamic meshes", in *27th Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 1990.

[34] A. Dugeai, elsA*: Ael move3D Mesh Deformation Theoretical/User Guide*, techreport, ONERA, no. /ELSA/MU-10004-2014, 2014.

[35] D. Maruyama, D. Bailly and G. Carrier, "High-Quality Mesh Deformation Using Quaternions for Orthogonality Preservation", *AIAA J.* **52** (2014), no. 12, pp. 2712–2729.

[36] R. Franke and G. Nielson, "Smooth interpolation of large sets of scattered data", *Int. J. Numer. Methods Eng.* **15** (1980), no. 11, pp. 1691–1704.

[37] C. Farhat, M. 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", *Comput. Methods Appl. Mech. Eng.* **157** (1998), no. 1, pp. 95–114.

[38] J. Samareh, "Discrete Data Transfer Technique for Fluid-Structure Interaction", in *18th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 2007.

[39] A. De Boer, *Computational fluid-structure interaction: Spatial coupling, coupling shell and mesh deformation*, PhD thesis, Delft University of Technology (The Netherlands), 2008. Online at http://resolver.tudelft.nl/uuid:7e4cfc71-57eb-4a6e-9e90-38854de21ce2.

[40] T. C. S. Rendall and C. B. Allen, "Improved radial basis function fluid-structure coupling via efficient localized implementation", *Int. J. Numer. Methods Eng.* **78** (2009), no. 10, pp. 1188–1208.

[41] T. Achard, *Techniques de calcul de gradient aéro-structure haute-fidélité pour l'optimisation de voilures flexibles*, PhD thesis, CNAM (France), 2017. Online at https://theses.fr/2017CNAM1140.

[42] J. F. Kiviaho and G. J. Kennedy, "Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least Squares", *AIAA J.* **57** (2019), no. 5, pp. 2237–2243.

[43] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB*, Interdisciplinary Mathematical Sciences, World Scientific, 2007.

[44] N. Flyer, G. A. Barnett and L. J. Wicker, "Enhancing finite differences with radial basis functions: Experiments on the Navier–Stokes equations", *J. Comput. Phys.* **316** (2016), pp. 39–62.

[45] P. R. Amestoy, A. Buttari, J.-Y. L'Excellent and T. Mary, "Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures", *ACM Trans. Math. Softw.* **45** (2019), no. 1, article no. 2 (26 pages).

[46] I. Ramière and T. Helfer, "Iterative residual-based vector methods to accelerate fixed point iterations", *Comput. Math. Appl.* **70** (2015), no. 9, pp. 2210–2226.

[47] A. Dugeai, Y. Mauffrey, A. Placzek and S. Verley, "Overview of the Aeroelastic Capabilities of the *elsA* Solver within the Context of Aeronautical Engines", *AerospaceLab J.* (2018), no. 14, article no. AL 14-03 (20 pages).

[48] J. J. Hollkamp and R. W. Gordon, "Reduced-order models for nonlinear response prediction: Implicit condensation and expansion", *J. Sound Vib.* **318** (2008), no. 4, pp. 1139–1153.

[49] T. Flament, J.-F. Deü, A. Placzek, M. Balmaseda and D.-M. Tran, "Reduced-order model of geometrically nonlinear flexible structures for fluid-structure interaction applications", *Int. J. Non-Linear Mech.* **158** (2024), article no. 104587.

[50] T. Flament, J.-F. Deü, A. Placzek, M. Balmaseda and D.-M. Tran, "Reduced Order Model of Nonlinear Structures for Turbomachinery Aeroelasticity", *J. Eng. Gas Turbines Power* **146** (2023), no. 3, article no. 031005 (9 pages).

[51] L. Hascoet and V. Pascual, "The Tapenade Automatic Differentiation tool: principles, model, and specification", *ACM Trans. Math. Softw.* **39** (2013), no. 3, article no. 20 (43 pages).

[52] G. S. Winckelmans and A. Leonard, "Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows", *J. Comput. Phys.* **109** (1993), no. 2, pp. 247–273.

[53] J. Valentin, L. Bernardos, É. Rivoalen and G. Pinon, "Vortex Particle Velocity Diffusion Method Using Dynamic Turbulence Control Based on Enstrophy", 2024. Online at https://hal.science/hal-04703933.

[54] E. C. Yates, N. S. Land and J. T. Foughner, *Measured and Calculated Subsonic and Transonic Flutter Characteristics of a 45 sweptback Wing Planform in Air and in Freon-12 in the Langley Transonic Dynamics Tunnel*, NASA technical note, National Aeronautics and Space Administration, 1963.

[55] R. Moretti, H. Yeo, F. Richez and B. Ortun, "Rotor Loads Prediction on UH-60A Flight Test using Loose Fluid/Structure Coupling", 2023. Online at https://hal.science/hal-04115739/document. Conference paper: Vertical Flight Society 79th Annual Forum & Technology Display.

[56] M. Balmaseda, H. Yeo, B. Jayaraman and B. Ortun, "High-fidelity aerodynamic loads analysis of the double-swept ERATO rotor", 2024. Conference paper: 50th European Rotorcraft Forum (ERF 2024).

[57] M. Aguirre, A. Riols-Fonclare and F. Richez, "Aeroelastic Analysis with Rapid Methods of the Double-swept ERATO Blade with an Homogeneous Structure in Hover Flight", 2024. Conference paper: Vertical Flight Society 80th Annual Forum & Technology Display.