A linear aerodynamics-based preconditioner for high-fidelity aeroelastic analysis and sensitivity analysis

Kristofer Jovanov, Roeland de Breuker, Mostafa M. Abdalla, Christophe Blondeau

► To cite this version:

Kristofer Jovanov, Roeland de Breuker, Mostafa M. Abdalla, Christophe Blondeau. A linear aerodynamics-based preconditioner for high-fidelity aeroelastic analysis and sensitivity analysis. 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Jun 2016, Washington, United States. 19 p., 10.2514/6.2016-4123 . hal-01434890

HAL Id: hal-01434890

<https://hal.science/hal-01434890>

Submitted on 4 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

A linear aerodynamics-based preconditioner for high-fidelity aeroelastic analysis and sensitivity analysis

Kristofer Jovanov, ^{*} Roeland De Breuker, [†] and Mostafa M. Abdalla [‡]

Delft University of Technology, 2629 HS Delft, The Netherlands

Christophe Blondeau, [§]

Onera - The French Aerospace Lab, F-92322 Châtillon, France.

High-fidelity aeroelastic analysis and sensitivity analysis are pivotal components in gradient-based aerostructural optimization. These analyses often give rise to linear sub-problems that require iterative solvers to maintain realistic CPU/Memory consumption. However, the convergence behaviour of iterative solvers is highly influenced by the applied preconditioner. In this paper a linear aerodynamics-based preconditioner is presented with the capability of reducing the solution time of high-fidelity aeroelastic analysis and sensitivity analysis. It is constructed in Schur complement form by a panel method. Hence, it is referred to as a low-fidelity Schur complement. The proposed solution method is demonstrated on the ONERA M6 wing. The structural model of a classical wing box layout has 1812 degrees of freedom and is solved by Nastran. The high-fidelity aerodynamic model is based on the compressible Euler equations and consists of 1.11 million cells. The solver used for the Euler flow analysis and sensitivity analysis is the multi-block structured flow solver elsA. The efficiency of the preconditioner is studied for a subsonic case of Mach 0.60 and a transonic case of Mach 0.85. The application of the low-fidelity Schur complement as a preconditioner shows up to three times faster CPU times for the Mach 0.85 sensitivity analysis case.

I. Introduction

The trend of employing high-fidelity models in multidisciplinary design optimization (MDO) has become more pronounced with increasing computational resources, the advent of code parallelism and improving solution algorithms.^{1–5} State-of-the art solvers tailored for single-disciplinary problems are commonly coupled through well-defined interfaces to facilitate the solution of multidisciplinary analysis problems. If subject to gradient-based optimization, additional design sensitivity analysis is required to compute the gradients of the responses with respect to any design variables. In the context of aerostructural design, these responses typically vary from structural weight and stresses to aerodynamic lift and drag. These quantities, and their associated gradients, are often regarded as inputs for the "black box" optimizer. The bottleneck in high-fidelity MDO is the high computational cost for acquiring these inputs. Much effort is therefore invested in improving the analysis and sensitivity analysis of high-fidelity fluid-structure interaction (FSI).

Maute et al.¹ solve the quasi-static aeroelastic problem using a nonlinear block Gauss-Seidel (NLBGS) method augmented by a stationary under-relaxation technique. This method, which is fundamentally a loosely coupled approach, couples the nonlinear Euler flow solver to a linear finite element model through a spring analogy mesh deformation strategy. The sensitivity problem is solved by applying the same method, modified to a linear problem. This straightforward approach offers a high level of software modularity.

^{*}Ph.D. Student, Faculty of Aerospace Engineering, Aerospace Structures and Computational Mechanics, Kluyverweg 1; K.Jovanov@tudelft.nl

[†]Assistant Professor, Faculty of Aerospace Engineering, Aerospace Structures and Computational Mechanics, Kluyverweg 1; R.deBreuker@tudelft.nl

[‡]Associate Professor, Faculty of Aerospace Engineering, Aerospace Structures and Computational Mechanics, Kluyverweg 1; M.M.Abdalla@tudelft.nl

[§]Research Scientist, Aeroelasticity and Structural Dynamics Department, ONERA; Christophe.Blondeau@onera.fr

As highlighted in a follow-up paper by Barcelos et al.,² the NLBGS method lacks robustness and efficiency. Moreover, the convergence behaviour is highly influenced by the relaxation parameter, whose optimal value is in general unknown. To address this, they propose the Schur-Newton-Krylov (SNK) solver. This approach is based on a Schur complement formulation applied to the interface variables of the aerodynamic mesh. The nonlinear quasi-static FSI problem is then solved by a Newton-Krylov method where the exact Jacobian is used. The same Krylov solver is applied for the subsequent linear sensitivity problem. It is shown that this method is more robust and efficient than the previous method described by Maute.

Kenway et al.⁴ solve the steady-state aeroelastic problem by a matrix-free coupled Newton-Krylov (CNK) method. The aeroelastic Jacobian is approximated by finite differences and the successive linear subproblems are solved by FGMRES.⁶ This approach shows reduced computation times compared to the conventional NLBGS approach, which in this work is augmented by a dynamic Aitken relaxation strategy. The linear sensitivity problem, in contrast to the nonlinear static problem, is solved by the exact aeroelastic Jacobian. The coupled linear system is solved by GMRES,⁷ where the aerodynamic block is preconditioned by ILU-factorization and the structural block is preconditioned by an exact factorization. No preconditioning is applied to the off-diagonal coupling terms. They show that this method converges faster compared to the loosely coupled approach. However, the main shortcoming of an ILU-preconditioner for the aerodynamic block is that the exact flux Jacobian need to be explicitly stored. This can result in vast memory requirements for large-scale problems.⁸

Despite the advancements described above, high-fidelity numerical methods still remain computationally expensive. In particular transonic flow simulations where the Euler or the RANS equations are employed are mainly regarded as computationally expensive. Consequently, these higher order aerodynamic methods are confined to the later stages of the design process where they are used for verification purposes.⁹ The early design stages, in contrast to the later ones, often rely on panel methods that deliver accurate results at a low computational cost. Panel methods, however, are not equally efficient in modelling nonlinear flow phenomena, e.g. shocks and flow separation. This implies that feasible designs might be filtered out in the early design stages due to the seemingly inferior transonic flow modelling capabilities of panel methods. It is therefore imperative that high-fidelity models are adopted early in the design process. This is only viable if these models are computationally inexpensive.

In an attempt to rectify the aerodynamic loads (aeroloads) obtained by panel methods and still maintain a low computational cost, several correction strategies have been developed.^{10–13} The basic principle is to correct the aerodynamic influence coefficient (AIC) matrix or the aeroloads based on data obtained by higher order computational fluid dynamics (CFD) simulations or wind tunnel tests. These correction methods have proven fruitful and are actively used in industry. However, correction methods for steady-state flows are mostly applied to the equilibrium equations. Correcting sensitivities for gradient-based optimization has not received equal amount of attention.

In this paper, we propose the application of a panel method as a preconditioner for solving the high-fidelity static aeroelastic analysis and sensitivity analysis problem. Initially, the high-fidelity equations are expressed in Schur complement form applied to the structural variables. A quasi-Newton method is then used to solve the nonlinear static aeroelastic problem, where the Newton correction is approximately solved by a fixed-point strategy. The fixed-point problem is preconditioned by a low-fidelity Schur complement obtained by the inexpensive panel method. The same principle of solving the fixed-point problem is applied to the subsequent linear sensitivity problem. The main difference is that the linear system in the sensitivity analysis problem is converged to a low tolerance to ensure accurate gradients for the optimizer.

Our proposed method is similar to the work by Heil.¹⁴ However, our approximated Schur complement requires information from the structural model and is therefore interdisciplinary. Furthermore, in this paper we investigate transonic flow problems of moderate size by employing the Euler equations for a 1.11 million cell model. The structural model is linear finite element model with 1812 degrees of freedom. The focus in this paper is purely on analysis. Its application in optimization is envisioned in a follow-up paper.

The remainder of this paper is organized as follows: the computational modules and routines necessary for the aeroelastic analysis and sensitivity analysis are described in Section II. Three different aeroelastic analysis methods, of which one is our proposed method, are described in Section III. The sensitivity analysis with our proposed approach is presented Section IV, followed by the comparison results in Section V. The paper is concluded by a brief discussion and reflection on possible improvements and suggestions for future work in Section VI.

II. Computational modules

Several computational modules must be established to enable aeroelastic analysis and sensitivity analysis, with each module performing individual subtasks and being capable of exchanging information in a systematic fashion. This section provides an overview of the modules and resources necessary for the aeroelastic analysis and the subsequent sensitivity analysis.

Organizing the work in well-defined modules might increase the overall computational effort as opposed to monolithic codes operating in single-language environments. However, this approach allows for a higher level of software modularity which is a coveted feature in industry. It enables engineers to choose among well-established solvers with minimal effort dedicated to adapting the existing code.

In the following subsections the (A) aerodynamic solvers, the (B) structural solver, the (C) coupling module and the (D) low-fidelity Schur complement will be described, respectively.

A. Aerodynamic Solvers

Two aerodynamic solvers are necessary in this work: one solving a set of low-fidelity (LoFi) equations and another solving a set of high-fidelity (HiFi) equations. The former falls into the category of panel methods where vortex ring elements are used to approximate the aerodynamic pressure. The flat two-dimensional panels are distributed on the camber surface of the wing and not on the actual surface (see Figure 1a). It should be mentioned that this particular panel distribution can generate a small offset in the results for thick wing configurations. The discrete linearized potential flow equations can be expressed in residual form as

$$\tilde{\mathcal{R}}_a(\boldsymbol{\gamma}, \mathbf{x}_p) = \mathbf{A}\boldsymbol{\gamma} - \mathbf{b} = \mathbf{0} \quad (1)$$

where $\boldsymbol{\gamma}$ is the unknown vector of vortex strengths, $\mathbf{A}$ is the dense AIC matrix and $\mathbf{b}$ is the Neumann boundary condition vector. Both $\mathbf{A}$ and $\mathbf{b}$ are functions of the panel coordinates $\mathbf{x}_p$. In order to avoid confusion with the HiFi solver later in the text, terms associated with the LoFi solver will be marked with a tilde notation. The AIC matrix is purely a function of geometry and can be assembled through the application of the Biot-Savart law. Once the linear system (1) is solved, the aeroloads can be obtained in a post-treatment step by the Kutta-Joukowski theorem

$$\tilde{\mathbf{f}}_a = \rho_\infty \mathbf{V}_\infty \times \boldsymbol{\gamma} \quad (2)$$

where ρ_∞ and $\mathbf{V}_\infty$ are the free-stream density and free-stream velocity vector, respectively. A Prandtl-Glauert coordinate transformation rule is applied to account for compressibility effects in high-subsonic flows. The computational routines for the LoFi solver are in accordance with, and can be further studied in, Katz and Plotkin.¹⁵ In addition to aeroloads, the LoFi solver must prove capable of generating the LoFi Schur complement, $\tilde{\mathcal{S}}$. The formulation of this term will be described in detail in Subsection D.

The main shortcoming of the LoFi solver is that its application is limited to attached flow conditions in low-subsonic to moderately high-subsonic flows. Panel methods, in general, lack the predictive capability of modelling nonlinearities in transonic flows. To address this shortcoming, a more advanced aerodynamic solver is required.

The HiFi aerodynamic solver is the multi-block structured flow solver elsA.¹⁶ In this work we solve the compressible Euler equations to account for recompression shocks. While the Euler equations might not be acknowledged as "high-fidelity" by many researchers, there are clear differences between the Euler equations and the linearized potential flow equations. It is based upon these relative differences that we have permitted to label the Euler equations in this work as high-fidelity. The discrete steady-state equations of the Euler flow problem can be expressed in residual form as

$$\mathcal{R}_a(\mathbf{w}, \mathbf{x}) = 0 \quad (3)$$

where $\mathbf{w}$ is the unknown vector of conservative variables, with five unknowns per cell, and $\mathbf{x}$ are the mesh coordinates. The aerodynamic residual, or the flux balance, is cancelled out when the solution converges to a steady-state equilibrium. elsA solves the nonlinear steady-state problem by an implicit backward-Euler (BWE) time integration scheme. In addition to solving the steady-state problem for the static aeroelastic analysis, a linearization of these equations are necessary for the subsequent sensitivity analysis. Consequently, four additional components are necessary from the flow solver:

(a) Camber surface panels for the panel method.

(b) Structured Euler mesh for elsA.

Figure 1: Aerodynamic models of the ONERA M6 wing.

1. $\partial \mathcal{R}_a / \partial \mathbf{w}$: This is the exact linearization of the discrete residual (3) with respect to the conservative variables, also known as the exact flux Jacobian. If the underlying spatial discretization scheme for the steady-state equations is of second order, then the flux Jacobian is a 13-banded sparse matrix formulated by an extended neighbours-of-neighbours stencil.
2. $\partial \hat{\mathcal{R}}_a / \partial \mathbf{w}$: This is an accurate approximation of the Jacobian matrix used for the implicitation of the convective fluxes in the BWE scheme. Usually a standard first order upwind approximation of the differential of the flux balance is preferred for better numerical conditioning. The approximate Jacobian matrix can then be expressed in the factorized form: $\partial \hat{\mathcal{R}}_a / \partial \mathbf{w} = (\mathbf{L} + \mathbf{D} + \mathbf{U})$. The resolution of system (4) relies on a method of relaxation which consists in calculating successive approximate solutions of the exact system using forward and backward sweeps through the computational domain.¹⁷
3. $\partial \mathcal{R}_a / \partial \mathbf{x}$: This is referred to as the geometric flux Jacobian. Similar to the exact flux Jacobian, the geometric flux Jacobian maintains a sparse structure. It describes how the flux balance changes with respect to aerodynamic grid perturbations.
4. $\partial \mathbf{f}_a / \partial \mathbf{w}$: This term can be obtained by linearizing the surface load pressure integration scheme. Only cells adjacent to the surface affect this term and as such the term can be decomposed to $(\partial \mathbf{f}_a / \partial \mathbf{w}_b) \times (\partial \mathbf{w}_b / \partial \mathbf{w})$, where $\mathbf{w}_b$ are the wall conservative variables that meet the physical boundary conditions (typically the wallslip condition for Euler equations). They are adequately extrapolated from the numerical scheme variables $\mathbf{w}$ in the adjacent cells.

All the components described above are only formulated as matrix-vector products in order to save memory. This feature is desirable for iterative solvers, such as GMRES, which do not require an explicit matrix formulation but can handle matrix-vector products to solve large sparse linear systems. Moreover, all the components in elsA are computed analytically by hand-differentiating the code.

To obtain the matrix-vector products, modifications to the existing elsA source code were necessary. The opt-module in elsA, dedicated to aerodynamic shape optimization, implements a fixed-point iterative scheme for the aerodynamic sensitivity problem. An extensive overview of the gradient computation capabilities is covered in Blondeau et al.⁵ The preconditioned fixed-point iterative scheme for the direct method gradient computation can be formulated as

$$\underbrace{\left[\frac{\mathbf{V}}{\Delta t} \mathbf{I} + \frac{\partial \hat{\mathcal{R}}_a}{\partial \mathbf{w}} \right]}_{\text{Implicit Operator}} \underbrace{\left(\frac{d\mathbf{w}}{d\mathbf{p}}^{(n+1)} - \frac{d\mathbf{w}}{d\mathbf{p}}^{(n)} \right)}_{\text{Correction}} = \underbrace{\left[-\frac{\partial \mathcal{R}_a}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} - \frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{p}}^{(n)} \right]}_{\text{Right-Hand Side}} \quad (4)$$

where a pseudo-time term Δt in the implicit operator is introduced to stabilize the convergence behaviour by increasing the diagonal dominance of ill-conditioned systems. $\mathbf{V}$ are the cell volumes and $d\mathbf{x}/d\mathbf{p}$ is a geometric gradient term that can be determined by the coupling module in Subsection C. It is apparent that the unknown vector, $d\mathbf{w}/d\mathbf{p}$, converges when the correction, and hence the right-hand side (RHS), approaches zero.

The modification sequence in the elsA/Opt module can be narrowed down to three steps:

1. First, the computation of $(\partial\mathcal{R}_a/\partial\mathbf{x}) \times (d\mathbf{x}/d\mathbf{p})$ is dropped in the RHS of (4). This operation is only required once per gradient computation. Components 3 and 4, which are required for the aeroelastic sensitivity analysis, are obtained in separate routines.
2. The RHS at this point is zero. As it is a restart procedure, the term $d\mathbf{w}/d\mathbf{p}$ is initialized to an external vector of our choice. The RHS construction routine is then called which results in the product of the exact flux Jacobian matrix $\partial\mathcal{R}_a/\partial\mathbf{w}$ by the prescribed vector.
3. At this stage the RHS is equivalent to the flux Jacobian matrix-vector product. We then reset the RHS to a new input vector, $\mathbf{v}$, and (4) now reduces to the following linear system:

$$\left[\frac{\mathbf{V}}{\Delta t} \mathbf{I} + \frac{\partial\hat{\mathcal{R}}_a}{\partial\mathbf{w}} \right] \mathbf{y} = \mathbf{v} \quad (5)$$

where $\mathbf{y}$ is the matrix-vector product of the inverse flux Jacobian approximation. Hence, by relaxing the linear system (5) with multiple sweeps the inverse of the flux Jacobian approximation is multiplied by $\mathbf{v}$. Typically 4-8 cycles, resulting in 8-16 sweeps, are sufficient. An increased number of cycles improves the quality of the preconditioner. However, this comes at the expense of an increased computation time. As an additional control parameter on the diagonal dominance of the preconditioner, it is the users choice to adjust or even cancel the pseudo-time term in the implicit operator.

With the new capabilities at our disposal, the aerodynamic sensitivity problem (4) can alternatively be solved by an external preconditioned GMRES algorithm. The right-hand side is thus solved directly in its original form

$$\left[\frac{\partial\mathcal{R}_a}{\partial\mathbf{w}} \right] \frac{d\mathbf{w}}{d\mathbf{p}} = - \frac{\partial\mathcal{R}_a}{\partial\mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} \quad (6)$$

which is preconditioned by the approximate flux Jacobian. A simple test case is conducted to demonstrate the difference in convergence behaviour between the external GMRES solver and the standard fixed-point LU-SSOR solver. Our test case is identical to the gradient computation test case in Blondeau et al.⁵ The computational Euler model is a 1.11 million cell mesh divided into five blocks (see Figure 1b). An upwind Roe scheme with a MUSCL interpolation associated to a Van Albada limiting function is applied. The Mach number is 0.734 and the incidence angle is 2.8°. The freestream flow condition is set to $\rho_\infty = 1.225\text{m}^3/\text{kg}$ and $p_\infty = 101325\text{N}/\text{m}^2$. The design variable used to construct the right-hand side in Equation (6) controls the section camber at 50% span and affects linearly the region from root to 90% of wing span.

Initially, the steady-state problem is converged to machine precision before the gradient computation is initialized. We implement a preconditioned GMRES solver with a restart subspace of 20 Krylov vectors. This corresponds to 0.82 Gigabytes in our case which is a reasonable memory storage. The number of cycles for the preconditioner, in both methods, is fixed at 8. Figure 2 shows a clear advantage for the GMRES solver which has a convergence rate roughly six times faster compared to the LU-SSOR method. The computation times for the two methods follows the same trend as in Figure 2. The main additional computation time that emanates from the GMRES algorithm is the formulation of the Krylov basis by the Arnoldi iteration which is very small compared to the matrix-vector products. Hence, by implementing non-intrusive modifications to the current source code of elsA we have shown the benefits of implementing a matrix-free GMRES solver. Moreover, the LU-SSOR preconditioner does not require the explicit storage of the flux Jacobian as opposed to ILU-preconditioned GMRES solvers.¹⁸ This implies a significant advantage for large-scale CFD problems.⁸ However, it should be mentioned that the factorization error of LU-SSOR is larger than ILU. An ILU-preconditioner is therefore anticipated to yield higher converge rates than LU-SSOR.

Figure 2: Convergence plot of the direct method gradient computation.

B. Structural Solver

The structural solver in this work is the finite element analysis program Nastran.¹⁹ The main task of Nastran is to assemble and output the structural stiffness matrix, $\mathbf{K}$, to be used externally for the aeroelastic analysis and the sensitivity analysis. If the deformations are of moderate size the structure can be assumed to exhibit a linear-elastic behaviour. Hence, the resulting stiffness matrix need only be generated once for the initial finite element model and stored in memory. In the event of large structural deformations the assumption of linear elasticity no longer holds and the stiffness matrix need to be continuously updated. In this paper only linear elasticity is considered. The discrete structural equations can then be expressed as

$$\mathcal{R}_s(\mathbf{u}) = \mathbf{K}\mathbf{u} - \mathbf{f}_s = 0 \quad (7)$$

where $\mathbf{u}$ are the structural degrees of freedom and $\mathbf{f}_s$ are any externally applied loads. In addition to the stiffness matrix, pseudo-loads are required for the subsequent sensitivity analysis. The pseudo-loads can be derived by linearizing (7) with respect to structural design variables $\mathbf{p}$:

$$\frac{d\mathcal{R}_s}{d\mathbf{p}} = \mathbf{K} \frac{d\mathbf{u}}{d\mathbf{p}} + \underbrace{\frac{\partial \mathbf{K}}{\partial \mathbf{p}} \mathbf{u} - \frac{d\mathbf{f}_s}{d\mathbf{p}}}_{\text{Pseudo-Loads}} = 0. \quad (8)$$

This equation is derived, and its components are explained in detail, in the Nastran Design Sensitivity and Optimization User's Guide.²⁰ What is of interest to us, prior to the sensitivity analysis, is the explicit dependence of the structural residual with respect to structural design variables. The only term that satisfies this requirement in Equation (8) is the stiffness gradient

$$\frac{\partial \mathcal{R}_s}{\partial \mathbf{p}} = - \frac{\partial \mathbf{K}}{\partial \mathbf{p}} \mathbf{u}^* \quad (9)$$

where $\mathbf{u}^*$ represents the structural displacements at static aeroelastic equilibrium. The remaining terms in (8) are implicit functions of the residual and are only available when the sensitivity analysis has converged. The gradient term in Equation (9) is the right-hand side for the sensitivity analysis in Section IV and is of size $n_{fdof} \times n_p$, where n_{fdof} is the number of unconstrained structural degrees of freedom and n_p is the number of structural design variables.

The stiffness matrix and the pseudo-loads can be readily accessed by adding Direct Matrix Abstraction Program (DMAP)²¹ alters in the case control section of the bulk data files. The stiffness matrix requires a SOL 101 solution sequence whereas the pseudo-loads require a SOL 200 static analysis solution sequence.

The DMAP alters can then export the matrices during the execution of Nastran. The exported *op4* files are converted to convenient sparse matrix triplet format using external fortran scripts to be used in the aeroelastic analysis and sensitivity analysis.

C. Coupling module

The coupling module enables the deformation of the aerodynamic mesh and the load interpolation on the structural nodes. As the aerodynamic models and the structural model in this work are non-conforming, there is a necessity for a robust and efficient mesh deformation algorithm. To this end, we apply an algebraic coupling routine based on radial basis functions. The basic principle is to formulate a coupling matrix $\mathbf{H}$ that interpolates the displacements of the aerodynamic grid based on the structural deformations:

$$\mathbf{u}_a = \mathbf{H}\mathbf{u} \quad (10)$$

where $\mathbf{u}_a$ are the aerodynamic grid displacements. The coupling matrix in this work is derived in accordance with the methodology in Beckert and Wendland²² for multivariate data interpolation. We use the globally supported thin-plate spline function for the HiFi aerodynamic model which results in a dense interpolation matrix. To circumvent the need for large amounts of memory, the coupling matrix is only formulated as a matrix-vector product. Hence, it is never stored explicitly. Moreover, the mesh deformation code is executed on multiple processors. As the radial basis function approach does not require connectivity information, parallelizing the code becomes a trivial task.

A simple test case was conducted to assess the parallel scalability. Mesh deformation is performed on the 1.11 Euler model described in Subsection A. These computations, including all others in this work, were performed on Intel Xeon E5-2660 v3 Processors with a 2.60GHz CPU frequency. The mesh deformation wall time was clocked for 2, 4, 8, 16 and 32 processors. The results in Figure 3a show very good scalability up to approximately 16 processors. Hence, the CPU time for one mesh deformation on a 1.11 million cell model can be reduced from 86 seconds on 2 processors to 18 seconds on 16 processors.

(a) Parallel computation scalability. Simulations are performed on 2, 4, 8, 16 and 32 processors. (b) Surface point reduction on the ONERA M6 wing. Only black points are used for the coupling matrix.

Figure 3: Coupling module improvements for accelerated mesh deformation.

Another method, in addition to parallelism, of improving the computation time is to limit the number of surface points in the formulation of the coupling matrix. There are numerous data point reduction algorithms that can be used to reduce the size of the coupling matrix without significantly affecting the quality of the deformation.^{23,24} In this work, due to the structured mesh, we simply apply every 10th surface point to formulate the coupling matrix (see Figure 3b). This has been found to yield a favourable trade-off between mesh deformation accuracy and computation time.

In previous work^{12,13} we used the principle of equivalence of work which allows for the transposed coupling matrix to be used as a load transfer operator

$$\mathbf{f}_s = \mathbf{H}^T \mathbf{f}_a. \quad (11)$$

This, however, is not always a favourable option. Based on our experience this operator can result in unrealistic oscillatory loads in regions where structural nodes are tightly clustered. We therefore resort to a simple, yet highly effective and robust, nearest-neighbour scheme for multivariate interpolation.²⁵ In this work, the load transfer operator is denoted by $\mathbf{T}$, such that

$$\mathbf{f}_s = \mathbf{T} \mathbf{f}_a. \quad (12)$$

D. Low-fidelity Schur complement

As emphasized in Subsection A, the LoFi Schur complement needs to be readily accessible by the panel method. Only its derivation is discussed in this subsection. Its purpose will be covered in Section IV. The LoFi Schur complement can be derived by formulating a static LoFi aeroelastic problem. Consequently, the aerodynamic and structural residual in Equations (1) and (7), respectively, can be concatenated and expressed as an aeroelastic residual

$$\tilde{\mathcal{R}}_{ael} = \begin{bmatrix} \tilde{\mathcal{R}}_a(\boldsymbol{\gamma}, \mathbf{u}) \\ \mathcal{R}_s(\mathbf{u}, \boldsymbol{\gamma}) \end{bmatrix} = 0. \quad (13)$$

If the aeroelastic residual is linearized with respect to the unknown variables the following linear system is obtained:

$$\begin{bmatrix} \frac{\partial \tilde{\mathcal{R}}_a}{\partial \boldsymbol{\gamma}} & \frac{\partial \tilde{\mathcal{R}}_a}{\partial \mathbf{u}} \\ \frac{\partial \mathcal{R}_s}{\partial \boldsymbol{\gamma}} & \frac{\partial \mathcal{R}_s}{\partial \mathbf{u}} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\gamma} \\ \Delta \mathbf{u} \end{bmatrix} = - \begin{bmatrix} \tilde{\mathcal{R}}_a \\ \mathcal{R}_s \end{bmatrix} \quad (14)$$

where the incremental vector of vortex strengths and structural displacements is also known as the Newton correction. This in fact results in Newton's method for solving the static aeroelastic problem where the tangent operator is the aeroelastic Jacobian. The diagonal blocks are inherently single-disciplinary and are therefore determined individually by their respective computational solvers. The off-diagonal blocks include coupling matrices that require information from both computational solvers as well as the coupling module. The Schur complement of the upper-diagonal block in the tangent operator is:

$$\tilde{\mathcal{S}} = \left(\frac{\partial \mathcal{R}_s}{\partial \mathbf{u}} - \frac{\partial \mathcal{R}_s}{\partial \boldsymbol{\gamma}} \left[\frac{\partial \tilde{\mathcal{R}}_a}{\partial \boldsymbol{\gamma}} \right]^{-1} \frac{\partial \tilde{\mathcal{R}}_a}{\partial \mathbf{u}} \right) \quad (15)$$

The four blocks required to construct the LoFi Shur complement are studied in detail:

1. $\partial \mathcal{R}_s / \partial \mathbf{u}$ is nothing but the reduced stiffness matrix $\mathbf{K}$ which is already provided as an output by the structural solver, as described in Subsection B.
2. $\partial \mathcal{R}_s / \partial \boldsymbol{\gamma}$ describes how the structural residual change with respect to vortex strength perturbations. This matrix can be decomposed as: $(\partial \mathcal{R}_s / \partial \tilde{\mathbf{f}}_a) \times (\partial \tilde{\mathbf{f}}_a / \partial \boldsymbol{\gamma})$. The first term is the load transfer operator $\mathbf{T}$ that interpolates loads from the aerodynamic panels on the structural nodes. The second term can be easily obtained by differentiating the Kutta-Joukowski theorem (2).
3. $\partial \tilde{\mathcal{R}}_a / \partial \boldsymbol{\gamma}$ is the AIC matrix, $\mathbf{A}$. This can easily be assembled by the LoFi solver, inverted and stored in memory.
4. $\partial \tilde{\mathcal{R}}_a / \partial \mathbf{u}$ can be decomposed as: $(\partial \tilde{\mathcal{R}}_a / \partial \mathbf{x}_p) \times (\partial \mathbf{x}_p / \partial \mathbf{u})$. The first term requires a linearization of the aerodynamic residual (1) with respect to the panel coordinates. Both the AIC matrix and the Neumann boundary condition vector are affected by the panel coordinates. This term, in particular the AIC derivatives, is cumbersome to obtain and can result in excessive computation times for aerodynamic models with fine panel grids. To overcome this shortcoming, a parallel approach with analytically obtained gradients is implemented. The second term in the decomposition is a displacement interpolation operator $\tilde{\mathbf{H}}$ which is readily available by the coupling module. For the panel method an infinite-plate spline²⁶ method is used to conform the panels with the structural displacements.

III. Aeroelastic Analysis

The main purpose of the static aeroelastic analysis is to determine the state of equilibrium for the FSI problem. This part is crucial in design optimization as it allows for important quantities such as aerodynamic drag and lift, and structural stresses to be evaluated. All these responses can be represented by

$$\mathcal{J} = \mathcal{J}(\mathbf{x}, \mathbf{w}, \mathbf{u}; \mathbf{p}) \quad (16)$$

where $\mathcal{J}$ can be any response. A response is a collective term for both the objective function and other possible constraint functions. The design variables can impose an explicit influence on the response, as well as an implicit influence through the solution of the underlying governing equations. To obtain these responses for a given set of design variables the static aeroelastic problem need to be solved. The static HiFi aeroelastic problem can be defined in a two-field formulation as

$$\mathcal{R}_{ael} = \begin{bmatrix} \mathcal{R}_a(\mathbf{w}, \mathbf{u}; \mathbf{p}) \\ \mathcal{R}_s(\mathbf{u}, \mathbf{w}; \mathbf{p}) \end{bmatrix} = 0. \quad (17)$$

In contrast to the static LoFi problem (13), we now want to satisfy a state of equilibrium between the Euler flow equations and the structural equations. The static aeroelastic problem is solved when the flow and structural subproblem converge within tolerances ϵ_{a1} and ϵ_{s1} , respectively:

$$\left\| \mathcal{R}_a^{(n+1)} \right\|_2 \leq \epsilon_{a1} \left\| \mathcal{R}_a^{(0)} \right\|_2 \quad (18)$$

$$\left\| \mathcal{R}_s^{(n+1)} \right\|_2 \leq \epsilon_{s1} \left\| \mathcal{R}_s^{(0)} \right\|_2. \quad (19)$$

For any iteration with index n , both criteria must be met simultaneously before the problem is considered converged. There are numerous possibilities for solving the nonlinear problem (17).¹⁻⁴ In this work three solution methods are highlighted in the following subsections.

Nonlinear Block Gauss-Seidel (NLBGS) method

The first method is a conventional NLBGS method. It is initiated by approximately solving the steady-state flow problem (3) by performing n_a flow subiterations for the initial shape. The aeroloads are then interpolated on the finite element model (12) and the linear structural equations (7) are solved. Finally, the aerodynamic mesh is updated (10) to align with the deformed structure whereby the routine is restarted and iterated until the aeroelastic problem converges. This method is enhanced by a dynamic relaxation parameter, θ , obtained by Aitken acceleration.²⁷ The NLBGS method is described in Table 1.

Table 1: The NLBGS algorithm

(0) Set initial values for $\mathbf{u}^{(0)}$, $\mathbf{w}^{(0)}$, $\mathbf{x}^{(0)}$ and $\theta^{(0)}$.

Iterate $n = 0, \dots, \infty$

(1) Approximately solve the flow problem, $\mathcal{R}_a(\mathbf{w}^{(n+1)}, \mathbf{x}^{(n)}) \approx 0$, by n_a flow subiterations.

(2) Interpolate the aeroloads on the structural model: $\mathbf{f}_s^{(n+1)} = \mathbf{T} \mathbf{f}_a^{(n+1)}$.

(3) Solve the structural subproblem: $\mathcal{R}_s(\mathbf{u}^{(n+1)}) = 0$.

(4) Evaluate the increment $\Delta \mathbf{u}^{(n+1)} = \mathbf{u}^{(n+1)} - \mathbf{u}^{(n)}$ and update the relaxation parameter by Aitken acceleration $\theta^{(n+1)} = \theta^{(n)} \left(1 - \frac{(\Delta \mathbf{u}^{(n+1)} - \Delta \mathbf{u}^{(n)})^T \Delta \mathbf{u}^{(n+1)}}{(\Delta \mathbf{u}^{(n+1)} - \Delta \mathbf{u}^{(n)})^T (\Delta \mathbf{u}^{(n+1)} - \Delta \mathbf{u}^{(n)})} \right)$.

(5) Relax the structural solution: $\mathbf{u}^{(n+1)} = (1 - \theta^{(n+1)}) \mathbf{u}^{(n)} + \theta^{(n+1)} \mathbf{u}^{(n+1)}$.

(6) Update the aerodynamic mesh by: $\mathbf{x}^{(n+1)} = \mathbf{x}^{(0)} + \mathbf{H} \mathbf{u}^{(n+1)}$.

Schur Quasi-Newton (SQN) method

The second method is a quasi-Newton algorithm. The term "quasi" implies that some form of approximation is applied to the Jacobian. The method can be derived by linearizing the aeroelastic residual (17) with respect to the unknown variables $\mathbf{w}$ and $\mathbf{u}$, respectively

$$\begin{bmatrix} \frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} & \frac{\partial \mathcal{R}_a}{\partial \mathbf{u}} \\ \frac{\partial \mathcal{R}_s}{\partial \mathbf{w}} & \frac{\partial \mathcal{R}_s}{\partial \mathbf{u}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{w} \\ \Delta \mathbf{u} \end{bmatrix} = - \begin{bmatrix} \mathcal{R}_a \\ \mathcal{R}_s \end{bmatrix} \quad (20)$$

where the variable increments represent the Newton correction which is used to update the variables for the subsequent iteration

$$\begin{bmatrix} \mathbf{w}^{(n+1)} \\ \mathbf{u}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{w}^{(n)} \\ \mathbf{u}^{(n)} \end{bmatrix} + \begin{bmatrix} \Delta \mathbf{w} \\ \Delta \mathbf{u} \end{bmatrix}. \quad (21)$$

The design variables are constant throughout the aeroelastic analysis and are therefore not considered in the linearization. A Schur complement formulation applied to the structural variables allows for the elimination of the aerodynamic residual

$$\underbrace{\left(\frac{\partial \mathcal{R}_s}{\partial \mathbf{u}} - \frac{\partial \mathcal{R}_s}{\partial \mathbf{w}} \left[\frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} \right]^{-1} \frac{\partial \mathcal{R}_a}{\partial \mathbf{u}} \right)}_{\mathcal{S}} \Delta \mathbf{u} = -\mathcal{R}_s \quad (22)$$

where $\mathcal{S}$ is the HiFi Schur complement. This method, which is an exact Newton method, exhibits a quadratic rate of convergence in the proximity of the solution. Unfortunately, for large-scale problems the HiFi Schur complement can be difficult and computationally expensive to determine. In addition, matrix inversion or a complete factorization of the flux Jacobian becomes highly impractical. To circumvent this, the HiFi Schur complement can be replaced by its computationally inexpensive LoFi counterpart (15):

$$\tilde{\mathcal{S}} \Delta \mathbf{u} = -\mathcal{R}_s. \quad (23)$$

Consequently, the Schur complement becomes an approximation, hence the name: quasi-Newton. Since it is expressed in Schur complement form, we refer to this method as the Schur quasi-Newton (SQN) method. This method can become a powerful tool if the approximation is close to the exact Schur complement and it is computationally cheap to obtain.

A similar quasi-Newton approach is performed by Gerbeau and Vidrascu²⁸ for transient FSI problems in blood flows. They propose to neglect the expensive mesh motion and the nonlinear terms in the fluid model. In our work, however, the LoFi solver is a stand-alone module that is completely independent from the HiFi solver. The routines for the PQN solver are described in Table 2. Note that in this method the structural residual is only evaluated and not solved, in contrast to the NLBGS method.

Table 2: The SQN algorithm

(0) Set initial values for $\mathbf{u}^{(0)}$, $\mathbf{w}^{(0)}$ and $\mathbf{x}^{(0)}$.

Iterate $n = 0, \dots, \infty$

(1) Approximately solve the flow problem, $\mathcal{R}_a(\mathbf{w}^{(n+1)}, \mathbf{x}^{(n)}) \approx 0$, by n_a flow subiterations.

(2) Interpolate the aeroloads on the structural model: $\mathbf{f}_s^{(n+1)} = \mathbf{T} \mathbf{f}_a^{(n+1)}$.

(3) Evaluate the structural residual: $\mathcal{R}_s = \mathbf{K} \mathbf{u}^{(n)} - \mathbf{f}_s^{(n+1)}$.

(4) Solve the approximated Newton correction: $\tilde{\mathcal{S}} \Delta \mathbf{u} = -\mathcal{R}_s$.

(5) Update the structural solution by the correction: $\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \Delta \mathbf{u}$.

(6) Update the aerodynamic mesh by: $\mathbf{x}^{(n+1)} = \mathbf{x}^{(0)} + \mathbf{H} \mathbf{u}^{(n+1)}$.

Preconditioned Schur Quasi-Newton (PSQN) method

The third method, which is our proposed method, capitalizes on the elsA modifications described in Subsection A to solve (22) by reformulating it into a fixed-point problem

$$\tilde{\mathcal{S}} \left(\Delta \mathbf{u}^{(k+1)} - \Delta \mathbf{u}^{(k)} \right) = - \left(\mathcal{S} \Delta \mathbf{u}^{(k)} + \mathcal{R}_s \right). \quad (24)$$

The fixed-point formulation (24) yields a subiteration sequence with index k . The linear system is solved by a direct method whereas the Schur matrix-vector product, $\mathcal{S} \Delta \mathbf{u}^{(k)}$, is approximated by GMRES. Based on this information two questions emerge: how much do we need to converge the correction and how much do we need to converge the Schur matrix-vector product? An accurate correction is anticipated to result in fewer Newton iterations. This might seem advantageous at first. However, to achieve the quadratic convergence rate of Newton's method, in addition to being close to the solution, the correction must be solved exactly. This implies that the flow problem must converge to a low tolerance by performing sufficient amount of subiterations, n_a . This is not a viable option as the overall computation time needed to perform an exact Newton method would vastly surpass that of an approximated Newton method.

In our approach only one fixed-point iteration is performed for the Newton correction and the residual norm in GMRES is converged to $\epsilon_g = 1e^{-1}$. To distinguish from the exact Schur matrix-vector product, this approximated version can be denoted as $\hat{\mathcal{S}} \Delta \mathbf{u}^{(k)}$, such that Equation (24) can be formulated as

$$\tilde{\mathcal{S}} \left(\Delta \mathbf{u}^{(k+1)} - \Delta \mathbf{u}^{(k)} \right) = - \left(\hat{\mathcal{S}} \Delta \mathbf{u}^{(k)} + \mathcal{R}_s \right). \quad (25)$$

It should be noted that $\hat{\mathcal{S}}$ is based on an exact linearization of the Euler flow model whereas $\tilde{\mathcal{S}}$ is based on the panel method. Since the aeroelastic problem is solved in Schur complement form and $\tilde{\mathcal{S}}$ is used as a preconditioner to approximate the correction, this method is referred to as a preconditioned Schur quasi-Newton (PSQN). The details of the algorithm are explained in Table 3. This method resembles the SQN. The main difference is that the Newton correction is enhanced by the additional steps 5-7. Note that the structural stiffness matrix, the coupling operators and the elsA components described in Subsection A, are used to clarify the construction of $\hat{\mathcal{S}} \Delta \mathbf{u}$.

Table 3: The PSQN algorithm

(0)	Set initial values for $\mathbf{u}^{(0)}$, $\mathbf{w}^{(0)}$ and $\mathbf{x}^{(0)}$.
	Iterate $n = 0, \dots, \infty$
(1)	Approximately solve the flow problem, $\mathcal{R}_a(\mathbf{w}^{(n+1)}, \mathbf{x}^{(n)}) \approx 0$, by n_a flow subiterations.
(2)	Interpolate the aeroloads on the structural model: $\mathbf{f}_s^{(n+1)} = \mathbf{T} \mathbf{f}_a^{(n+1)}$.
(3)	Evaluate the structural residual: $\mathcal{R}_s = \mathbf{K} \mathbf{u}^{(n)} - \mathbf{f}_s^{(n+1)}$.
(4)	Solve $\tilde{\mathcal{S}} \Delta \mathbf{u}^{(0,n)} = -\mathcal{R}_s$ prior to initializing the fixed-point iteration.
	Iterate $k = 0, \dots, \infty$
(5)	Approximately solve $\left[\frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} \right] \Delta \mathbf{w}^{(k+1,n)} \approx \frac{\partial \mathcal{R}_a}{\partial \mathbf{x}} \mathbf{H} \Delta \mathbf{u}^{(k,n)}$ by GMRES where $\epsilon_g = 1e^{-1}$.
(6)	Subsequently compute $\hat{\mathcal{S}} \Delta \mathbf{u}^{(k,n)} = \mathbf{K} \Delta \mathbf{u}^{(k,n)} - \mathbf{T} \frac{\partial \mathbf{f}_a}{\partial \mathbf{w}} \Delta \mathbf{w}^{(k+1,n)}$.
(7)	Determine the fixed-point correction, $\Delta \mathbf{u}^{(k+1,n)}$, by solving Equation (24).
	Terminate for $k = 0$.
(8)	Update the structural solution by the correction: $\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \Delta \mathbf{u}^{(k+1,n)}$.
(9)	Update the aerodynamic mesh by: $\mathbf{x}^{(n+1)} = \mathbf{x}^{(0)} + \mathbf{H} \mathbf{u}^{(n+1)}$.

IV. Sensitivity Analysis

Sensitivity analysis allows for the evaluation of gradients that are used by the optimizer to determine the optimum. A response function, $\mathcal{J}$, can be expressed as a function of one or several design variables. The main purpose of sensitivity analysis is to determine the total gradient of the response function with respect to the design variables, $d\mathcal{J}/d\mathbf{p}$. In aeroelasticity, the response function can have an explicit dependence on the design variables, as well as an implicit dependence by the solution of the underlying governing equations and the mesh deformation problem. The total gradient is therefore obtained by linearizing (16)

$$\frac{d\mathcal{J}}{d\mathbf{p}} = \frac{\partial \mathcal{J}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{p}} + \frac{\partial \mathcal{J}}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{p}} + \frac{\partial \mathcal{J}}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\mathbf{p}}. \quad (26)$$

The partial derivatives of the response function are typically straightforward to evaluate and can be provided directly by the respective solver or by external post-processing tools. The geometric gradient $d\mathbf{x}/d\mathbf{p}$ is managed by the coupling module. To obtain the perturbed variables, $d\mathbf{w}/d\mathbf{p}$ and $d\mathbf{u}/d\mathbf{p}$, it is necessary to solve a linearization of the static aeroelastic problem at equilibrium. Hence, Equation (17) is linearized with respect to the design variables

$$\begin{bmatrix} \frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} & \frac{\partial \mathcal{R}_a}{\partial \mathbf{u}} \\ \frac{\partial \mathcal{R}_s}{\partial \mathbf{w}} & \frac{\partial \mathcal{R}_s}{\partial \mathbf{u}} \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{w}}{d\mathbf{p}} \\ \frac{d\mathbf{u}}{d\mathbf{p}} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathcal{R}_a}{\partial \mathbf{p}} \\ \frac{\partial \mathcal{R}_s}{\partial \mathbf{p}} \end{bmatrix}. \quad (27)$$

The solution of this linear system is equivalent to the direct approach, indicating that the number of linear systems correspond to the number of design variables. This approach is therefore convenient when the optimization task is limited to a few design variables. To solve the linear system by GMRES a quality preconditioner is required. The task of the preconditioner is to improve the spectral properties of the system matrix, such that when applied to an iterative solver the number of iterations decrease. Let us assume that the system matrix in Equation (27) can be expressed as

$$\mathcal{A} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{C} \end{bmatrix} \quad (28)$$

where $\mathbf{A}$, $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{C}$ are blocks that form a generalized saddle point system, $\mathcal{A}$. A block LDU-factorization of this matrix would then yield

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{B}_2 & \mathbf{C} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{B}_2 \mathbf{A}^{-1} & \mathbf{I} \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1} \mathbf{B}_1 \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{U}} \quad (29)$$

where $\mathbf{S}$ is the Schur complement of $\mathbf{A}$ in the system matrix $\mathcal{A}$. The diagonal matrix can then be our candidate preconditioner, $\mathcal{P}_d = \mathbf{D}$. Indeed, it is highlighted in the review paper by Benzi et al.,²⁹ that the preconditioned system matrix $\mathcal{M} = \mathcal{P}_d^{-1} \mathcal{A}$ has three distinct eigenvalues. This implies that a Krylov-based solver, e.g. GMRES, applied to the preconditioned linear system would converge in three iterations or less. Based on this premise, we adopt the block diagonal preconditioner to our linear sensitivity problem (27) in an attempt to replicate the favourable spectral properties. This yields in

$$\mathcal{P}_d = \begin{bmatrix} \frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{bmatrix}. \quad (30)$$

The difficulty of applying this preconditioner is that the inverse of the large sparse multi-banded flux Jacobian, $\partial \mathcal{R}_a / \partial \mathbf{w}$, must be determined. To circumvent this impracticality the following approximated block diagonal preconditioner is constructed:

$$\tilde{\mathcal{P}}_d = \begin{bmatrix} \frac{\partial \hat{\mathcal{R}}_a}{\partial \mathbf{w}} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{S}} \end{bmatrix} \quad (31)$$

where $\partial \hat{\mathcal{R}}_a / \partial \mathbf{w}$ is the aerodynamic preconditioner covered in Subsection A and $\tilde{\mathbf{S}}$ is the low-fidelity Schur complement covered in Subsection D.

Another important issue that emanates for the multidisciplinary linear system is associated with scaling. If the preconditioner is not properly scaled when applied to GMRES, the aerodynamic variables might become accurate while the structural variables are rendered inaccurate, or vice versa.³⁰ We therefore redefine the linear problem (27) by a Schur complement formulation applied to the structural variables

$$\mathcal{S} \frac{d\mathbf{u}}{d\mathbf{p}} = -\frac{\partial \mathcal{R}_s}{\partial \mathbf{p}}. \quad (32)$$

Once the perturbed displacements $d\mathbf{u}/d\mathbf{p}$ are determined, the perturbed conservative variables $d\mathbf{w}/d\mathbf{p}$ may be computed as a pure aerodynamic sensitivity problem by solving

$$\left[\frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} \right] \frac{d\mathbf{w}}{d\mathbf{p}} = -\frac{\partial \mathcal{R}_a}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\mathbf{p}}. \quad (33)$$

In fact, the perturbed conservative variables are implicitly computed during the formulation of the Schur matrix-vector product. Hence, this approach presents an opportunity to solve the coupled sensitivity equations absent the scaling problem. A similar approach to the PSQN solver is employed to solve (32). It is only modified to solve a linear problem, compared to the aeroelastic analysis that is nonlinear. This method is described in the following subsection.

Preconditioned Schur Complement (PSC) solver

We propose to use a preconditioned Schur complement solver by reformulating the linear problem (32) in fixed-point form

$$\tilde{\mathcal{S}} \left(\frac{d\mathbf{u}}{d\mathbf{p}}^{(k+1)} - \frac{d\mathbf{u}}{d\mathbf{p}}^{(k)} \right) = - \left(\hat{\mathcal{S}} \frac{d\mathbf{u}}{d\mathbf{p}}^{(k)} + \frac{\partial \mathcal{R}_s}{\partial \mathbf{p}} \right) \quad (34)$$

which is identical to Equation (25) with the exception of the right-hand side. This is a favourable feature that can reduce the implementation costs. The routines in 4-7 in Table 3 for the static aeroelastic analysis can be reused for the sensitivity analysis by only modifying the right-hand side and performing multiple fixed-point iterations. To ensure that the total gradients for the optimizer are accurate, the convergence is monitored by the criterion:

$$\left\| d\mathcal{R}_a^{(k+1)} \right\|_2 \leq \epsilon_{a2} \left\| d\mathcal{R}_a^{(0)} \right\|_2 \quad (35)$$

$$\left\| d\mathcal{R}_s^{(k+1)} \right\|_2 \leq \epsilon_{s2} \left\| d\mathcal{R}_s^{(0)} \right\|_2 \quad (36)$$

where $d\mathcal{R}_s$ and $d\mathcal{R}_a$ are the residual of Equations (32) and (33), respectively. Both criteria must be satisfied simultaneously before the solution is considered to be converged. This solution method, based on its similarities to the PSQN, is referred to as the preconditioned Schur complement (PSC) solver and is described in Table 4. The iteration index n is constant and is thus omitted for brevity.

Table 4: The PSC algorithm

-
- (0) Determine $\frac{\partial \mathcal{R}_s}{\partial \mathbf{p}}$ in a post-analysis step.
 - Solve $\tilde{\mathcal{S}} \frac{d\mathbf{u}}{d\mathbf{p}}^{(0)} = -\frac{\partial \mathcal{R}_s}{\partial \mathbf{p}}$ prior to initializing the fixed-point iteration.
 - Iterate $k = 0, \dots, \infty$
 - (1) Approximately solve $\left[\frac{\partial \mathcal{R}_a}{\partial \mathbf{w}} \right] \frac{d\mathbf{w}}{d\mathbf{p}}^{(k+1)} \approx \frac{\partial \mathcal{R}_a}{\partial \mathbf{x}} \mathbf{H} \frac{d\mathbf{u}}{d\mathbf{p}}^{(k)}$ by GMRES where $\epsilon_g = 1e^{-1}$.
 - (2) Subsequently compute $\hat{\mathcal{S}} \frac{d\mathbf{u}}{d\mathbf{p}}^{(k)} = \mathbf{K} \frac{d\mathbf{u}}{d\mathbf{p}}^{(k)} - \mathbf{T} \frac{\partial \mathbf{f}_a}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{p}}^{(k+1)}$.
 - (3) Determine the fixed-point correction, $\frac{d\mathbf{u}}{d\mathbf{p}}^{(k+1)}$, by solving Equation (34).
-

V. Results

Two test cases are conducted: one simulating subsonic flight at $Mach = 0.60$ and one simulating transonic flight at $Mach = 0.85$. The purpose is to assess the performance of our proposed method for a flight regime where the aeroloads are in agreement with the HiFi model ($Mach = 0.60$) and a flight regime where the aeroloads are presumed to have a higher offset ($Mach = 0.85$). The aerodynamic model and the solution parameters are identical to the test case in Subsection A. The impact of the number of implicit flow iterations, n_a , on the aeroelastic solution will be investigated by performing two subcases: 1) $n_a = 100$ and 2) $n_a = 300$. A v-cycle multi-grid acceleration technique with two coarse grids and a fixed CFL number of 10 are used to speed up the convergence of the steady-state equations. The number of cycles for the LU-SSOR preconditioner is reduced to 4, whereas the number of cycles is maintained at 8 for the gradient computation in the Schur matrix-vector product. The flow problem is solved on five processors (one for each block) and the mesh deformation and the LoFi Schur complement formulation are split on 16 processors each.

The structural model is a classic wing box layout containing 1812 degrees of freedom. Member thicknesses and sections have been designed in a pre-processing optimization step (see Figure 4a). This model can be easily tuned in order to control flexibility and consequently aeroelastic effects. Small adjustments have been applied to the original model in Blondeau et al.⁵ RBE2 elements are added to the ribs with force nodes at $z=0$ (see Figure 4b). This allows the interpolated loads from the panel method to have a smooth and acceptable distribution. Three force nodes per rib are added for the load transfer. The first one is at 50% of the wing surface leading edge and the wing box leading edge. The second one is located in the centre of the wing box and the third one is at 50% of the wing surface trailing edge and the wing box trailing edge. The governing structural equations (7) in this work are solved by a direct method.

(a) Original model. Shell thickness variation is displayed (b) Modified model. Rigid RBE2 elements are added. Aeroloads are interpolated on the force (red) nodes.

Figure 4: Finite element model.

Comparison of aeroelastic analysis results

The static aeroelastic problem is solved individually by the three methods in Section III. It should be noted that a pure aeroelastic computation is performed with a clamped condition at the root. The equilibrium solution is therefore not a trimmed solution, which means that the aerodynamic lift does not equal the structural weight. The CPU time is monitored for the CFD analysis, the mesh deformation, the formulation of the LoFi Schur complement and the formulation of the approximated HiFi Schur matrix-vector product, respectively. The CPU time for the structural analysis is disregarded in this work as the structural equations are solved in less than 0.01 seconds. The convergence tolerances are set to $\epsilon_{a1} = \epsilon_{s1} = 1e^{-6}$.

The $Mach = 0.60$ test case yields a 7% tip displacement of the semispan. The best performing method for the $n_a = 100$ subcase is the NLBGS (see Table 5). This can be explained by the moderate flexibility at this particular flight condition. No improvement is observed for the remaining methods and they all converge

in 9 iterations. For the $n_a = 300$ subcase, the SQN and the PSQN converge in fewer iterations compared to the NLBGS. The reason is that the aeroloads are evaluated at a higher precision, prompting the quadratic convergence of Newton’s method to become more pronounced. This is illustrated in Figure 5b. However, the best performing method for the $Mach = 0.60$ test case in general is the NLBGS method. The improved robustness of the PSQN method is cancelled out by the increased CFD analysis time.

Table 5: Aeroelastic Analysis

Method	$Mach$	n_a	n	CPU time [s]				
				CFD	Mesh	$\tilde{\mathcal{S}}$	$\hat{\mathcal{S}}\Delta\mathbf{u}$	Total
NLBGS	0.60	100	9	1059	161	—	—	1220
SQN	0.60	100	9	1049	157	82	—	1288
PSQN	0.60	100	9	1053	158	83	1047	2341
NLBGS	0.60	300	9	3170	159	—	—	3329
SQN	0.60	300	8	2808	138	73	—	3019
PSQN	0.60	300	5	1757	88	46	563	2454
NLBGS	0.85	100	20	2334	351	—	—	2685
SQN	0.85	100	12	1400	210	110	—	1720
PSQN	0.85	100	12	1407	209	111	2462	4189
NLBGS	0.85	300	20	7592	367	—	—	7959
SQN	0.85	300	9	3420	164	85	—	3669
PSQN	0.85	300	6	2283	110	57	1185	3635

(a) Residual convergence for $n_a = 100$.

(b) Residual convergence for $n_a = 300$.

Figure 5: Convergence histories for the $Mach = 0.60$ test case.

The equilibrium tip displacement for the $Mach = 0.85$ test case is 13% of the semispan. The best performing method for $n_a = 100$ is the SQN which shows a 36% CPU reduction to the equivalent subcase for NLBGS. Significant reduction in computation times is achieved by accelerating the convergence through the LoFi Schur complement, which is depicted in Figure 6a. The PSQN shows no advantage over the SQN in terms of number of iterations. When the flow subiteration count is increased to $n_a = 300$, as for the $Mach = 0.60$ test case, an improved convergence rate is manifested for the PSQN. This can clearly be observed in Figure 6b. However, the improved convergence rate comes at the expense of an increased CFD analysis time. Hence, the overall top performer for the $Mach = 0.85$ case is the SQN method. The flow

subiteration count could arguably be decreased further ($n_a < 100$) to give the NLBGS the edge over the SQN. However, this is not advisable as it will result in many more coupling iterations, which are illustrated by the peaks in the figures. Moreover, the Mesh-to-CFD CPU ratio might increase up to where one mesh deformation is comparable to one CFD analysis. It is not desirable for a large portion of the CPU time of the aeroelastic problem to be devoted to mesh deformation.

Figure 6: Convergence histories for the $Mach = 0.85$ test case.

Comparison of sensitivity analysis results

The design variable that is used to construct $\partial \mathcal{R}_s / \partial \mathbf{p}$ for the test case is the thickness of the shell element on the upper skin closest to the root and the trailing edge. This term is constructed by multiplying the perturbed stiffness matrix with the equilibrium displacements (9) obtained by the previous analysis. The application for the analysis and sensitivity analysis in this work is structural sizing or aeroelastic tailoring through directional stiffness variation. This implies that the design variables have no explicit influence on the aerodynamic residuals, $\partial \mathcal{R}_a / \partial \mathbf{p} = 0$. For the sensitivity analysis we set $\epsilon_{a2} = \epsilon_{s2} = 1e^{-8}$.

Two methods are compared in the sensitivity analysis. The first one is the PSC method as described in Section IV. The second one is identical with one minor difference: the LoFi Schur complement preconditioner is replaced by the structural stiffness matrix, such that Equation (34) is reduced to

$$\mathbf{K} \left(\frac{d\mathbf{u}^{(k+1)}}{d\mathbf{p}} - \frac{d\mathbf{u}^{(k)}}{d\mathbf{p}} \right) = - \left(\hat{\mathcal{S}} \frac{d\mathbf{u}^{(k)}}{d\mathbf{p}} + \frac{\partial \mathcal{R}_s}{\partial \mathbf{p}} \right). \quad (37)$$

The assumption for this modification is that no panel method is available to precondition the fixed-point linear system. This modification results in a looser coupling and is distinguished with a subscript as $\text{PSC}_{\mathbf{K}}$. The original version (34) is therefore identified for this test case with an added subscript as $\text{PSC}_{\hat{\mathcal{S}}}$. Note that the SQN method from the aeroelastic analysis is not applicable in the sensitivity analysis. The direct application of the LoFi Schur complement would only result in LoFi gradients.

The LoFi Schur complement preconditioner indicates to have an impact on the convergence rate for both test cases, as shown in Table 6. Since we are solving a linear system, only one generation of the LoFi Schur complement is required. The CPU cost of this operation is negligible compared to the overall CPU cost. Moreover, the subcases of setting a fix number of flow iterations, n_a , as in the aeroelastic analysis do not come into play in the sensitivity analysis. Here we are only concerned with gradient computation.

There is only a moderate improvement at $Mach = 0.60$ by using $\text{PSC}_{\hat{\mathcal{S}}}$. A 17% CPU reduction is observed and the convergence behaviour in Figure 7a does not display much difference in the decaying residual between the two methods.

Table 6: Sensitivity Analysis

Method	$Mach$	k	CPU time [s]		
			$\tilde{\mathcal{S}}$	$\hat{\mathcal{S}} \frac{du}{dp}$	Total
PSC $\mathbf{K}$	0.60	11	—	963	963
PSC $\tilde{\mathcal{S}}$	0.60	9	8	788	796
PSC $\mathbf{K}$	0.85	35	—	6222	6222
PSC $\tilde{\mathcal{S}}$	0.85	11	8	1951	1959

For the more flexible $Mach = 0.85$ case, the application of PSC $\tilde{\mathcal{S}}$ results in more than 3 times faster convergence compared to PSC $\mathbf{K}$. The addition of the LoFi Schur complement improves the stability and enables it to converge much faster, as illustrated in Figure (7b). This clearly indicates that $\tilde{\mathcal{S}}$ is more prominent choice of preconditioner. It should be noted that one fixed-point iteration for the $Mach = 0.85$ case is almost twice as costly as the $Mach = 0.60$ case. This might suggest that the $Mach = 0.85$ case experiences a higher degree of ill-conditioning.

The elegance of the PSC $\tilde{\mathcal{S}}$ is that the coupled equations (27) are practically transformed to an aerodynamic sensitivity problem. The mesh deformation operations, that would otherwise be necessary for each iteration, are reduced to a minimum. Only 9 and 11 mesh deformations are required for the $Mach = 0.60$ and $Mach = 0.85$ case, respectively, to converge the perturbed variables to a tolerance of $1e^{-8}$. The bulk of the computation time is devoted to solving Equation (33) which is embedded in the Schur matrix-vector product, $\hat{\mathcal{S}} \frac{du}{dp}$.

Figure 7: Convergence histories for the sensitivity analysis.

VI. Conclusions

We have described a panel code preconditioning method that facilitates the solution of high-fidelity aeroelastic analysis and sensitivity analysis problems. The panel code preconditioner is expressed in Schur complement form applied to the structural degrees of freedom. It is used to improve the spectral properties of the linear subproblems that appear in the aeroelastic analysis, when subject to Newton's method, and the subsequent sensitivity analysis. The linear subproblems are formulated in fixed-point iterative form and are solved approximately for the static aeroelastic problem and exactly for the sensitivity problem. The method applied for the static aeroelastic problem, which we refer to as the PSQN method, is not necessarily the optimal choice. It converges in less iterations compared to the NLBGS and the SQN method, but it also

requires a higher precision on the aeroloads for each Newton iteration. It is the most robust method, in this regard, but not the most efficient in terms of CPU time.

The sensitivity analysis version of the PSQN, the PSC, demonstrates much faster convergence when preconditioned with the LoFi Schur complement. The convergence rate is up to three times faster compared to the case when the PSC is preconditioned with the structural stiffness matrix. Moreover, the reduction of the problem in Schur complement form effectively reduces the number of mesh deformations to a minimum. The number of mesh deformations correspond to the number of fixed-point iterations which varied from 9 to 11 for the two test cases conducted in this paper. These values can be further reduced if the solution is allowed to converge to a higher tolerance. In conclusion, the panel code preconditioner is successfully demonstrated to reduce the computation time of high-fidelity sensitivity analysis for fluid-structure interaction problems.

The PSC method described in this paper for the direct approach gradient computation is directly applicable to the adjoint approach. The full implementation of the method for the adjoint problem is envisioned in a follow-up paper. Furthermore, trim analysis will be performed to allow for response generation that take into account trimmed flight for gradient-based optimization.

VII. Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7-PEOPLE-2012-ITN under grant agreement n 316394, Aerospace Multidisciplinarity-Enabling DEsign Optimization (AMEDEO) Marie Curie Initial Training Network. The authors would like to acknowledge Philippe Girodroux-Lavigne for his support on all actions that allowed us to use the flow solver elsA.

References

- ¹Maute, K., Nikbay, M., and Farhat, C., "Coupled Analytical Sensitivity Analysis and Optimization of Three-Dimensional Nonlinear Aeroelastic Systems," *AIAA Journal*, Vol. 39, No. 11, 2001, pp. 2051-2061.
- ²Barcelos, M., Bavestrello, H., and Maute, K., "A Schur-Newton-Krylov solver for steady state aeroelastic analysis and design sensitivity analysis," *Computer Methods in Applied Mechanics and Engineering*, Vol. 56, No. 6, 2006, pp. 911-933.
- ³Kennedy, G. J., and Martins, J. R. R. A., "Parallel Solution Methods for Aerostructural Analysis and Design Optimization," *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 13 - 15 September 2010, Fort Worth, Texas, AIAA 2010-9308.
- ⁴Kenway, G. K. W., Kennedy G. J., and Martins, J. R. R. A., "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations," *AIAA Journal*, Vol. 52, No. 5, 2014, pp. 935-951.
- ⁵Blondeau, C., Achard, T., Girodroux-Lavigne, P., and Ohayon, R., "Recent Achievements Towards Aero-Structure Gradient Computation using High-Fidelity CFD-CSM in the ONERA elsA Software," *International Forum on Aeroelasticity and Structural Dynamics*, June 28 - July 02, 2015, Saint Petersburg, Russia, IFASD 2015-019.
- ⁶Saad, Y., "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM Journal on Scientific Computing*, Vol. 14, No. 2, 1993, pp. 461-469.
- ⁷Saad, Y., and Schultz, M. H., "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856-869.
- ⁸Nielsen, E. J., and Anderson, W. K., "Recent improvements in aerodynamic design optimization on unstructured meshes," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1155-1163.
- ⁹Jameson, A., "Re-engineering the design process through computation," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 36-50.
- ¹⁰Giesing, J. P., Kalman, T. P., and Rodden, W. P., "Correction Factor Techniques for Improving Aerodynamic Prediction Methods," *Tech. rep. McDonnell-Douglas Corp., Long Beach, Calif.: National Aeronautics and Space Administration*, 1976.
- ¹¹Palacios, R., Climent, H., Karlsson, A., and Winzell, B., "Assessment of strategies for correcting linear unsteady aerodynamics using CFD or experimental results," *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, 2001.
- ¹²Jovanov, K., Breuker, R. D., Abdalla, M. M., and Dillinger, J. K. S., "Accelerated convergence of static aeroelasticity using low-fidelity aerodynamics," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 5 - 9 January, 2015, Kissimmee, Florida, AIAA 2015-0175.
- ¹³Jovanov, K., and Breuker, R. D., "Accelerated convergence of high-fidelity aeroelasticity using low-fidelity aerodynamics," *International Forum on Aeroelasticity and Structural Dynamics*, June 28 - July 02, 2015, Saint Petersburg, Russia, IFASD 2015-214.
- ¹⁴Heil, M., "An efficient solver for the fully coupled solution of large-displacement fluidstructure interaction problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 193, No. 1-2, 2004, pp. 1-23.
- ¹⁵Katz, J., and Plotkin, A., "Low-Speed Aerodynamics," 2nd ed, *Cambridge University Press*, 2001, Cambridge, UK.
- ¹⁶Cambier L., Gazeau M., Heib S., Plot S., Poinot M., Veuillot J.-P., Boussuge J.-F., and Montagnac M., "An Overview of the Multi-Purpose elsA Flow Solver," *Aerospace Lab Journal*, Issue 2 - March 2011.

- ¹⁷Peter, J., and Drullion, F., "Large Stencil Viscous Flux Linearization for the Simulation of 3D Compressible Turbulent Flows with Backward-Euler Schemes," *Computer and Fluids*, Vol. 36, No. 6, 2007, pp. 10051027.
- ¹⁸Luo, H., Baum, J. D., and Lhner, R., "A Fast, Matrix-free Implicit Method for Compressible Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 146, 1998, pp. 664690.
- ¹⁹MacNeal, R.H., "The Nastran theoretical manual," NASA (SP- 221(01)).
- ²⁰Raymond, M., "MSC.Nastran 2012, Design Sensitivity and Optimization Users Guide," 2011, MSC.Software Corporation.
- ²¹Raymond, M., "MSC.Nastran 2013, DMAP Programmer's Guide," 2013, MSC.Software Corporation.
- ²²Beckert, A., and Wendland, H., "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions," *Aerospace Science and Technology* Vol. 5, 2001, pp. 125134.
- ²³Rendall, T. C. S., and Allen, C. B., "Efficient mesh motion using radial basis functions with data reduction algorithms," *Journal of Computational Physics* Vol. 228, No. 17, 2009, pp. 6231-6249.
- ²⁴Rendall, T. C. S., and Allen, C. B., "Reduced surface point selection options for efficient mesh deformation using radial basis functions," *Journal of Computational Physics* Vol. 229, No. 8, 2010, pp. 2810-2820.
- ²⁵Thvenaz, P., Blu, T., and Unser, M., "Interpolation Revisited," *IEEE Transactions on Medical Imaging* Vol. 19, No. 7, 2000, pp. 739-758.
- ²⁶Harder, R., and Desmarais, R., "Interpolation using surface splines," *Journal of Aircraft* Vol. 9, No. 2, 1972, pp. 189-191.
- ²⁷Irons, B. M., and Tuck, R. C., "A Version of the Aitken Accelerator for Computer Iteration," *International Journal for Numerical Methods in Engineering*, Vol. 1, No. 3, 1969, pp. 275-277.
- ²⁸Gerbeau, J. F., and Vidrascu, M., "A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows," *ESAIM: Mathematical Modelling and Numerical Analysis*, Vol. 37 No. 4, 2003, pp. 631-647.
- ²⁹Benzi, M., Golub, G. H., and Liesen, J., "Numerical solution of saddle point problems," *Acta Numerica*, Vol. 14, 2005, pp. 1-137.
- ³⁰Wathen, A., "Preconditioning and convergence in the right norm," *International Journal of Computer Mathematics*, Vol. 84, No. 8, 2007, pp. 1199-1209.