CEAS Aeronautical Journal
https://doi.org/10.1007/s13272-025-00932-0

ORIGINAL PAPER

A multidisciplinary solver study towards efficient aeroelastic analyses using the FlowSimulator ecosystem

Kannan Masilamani[1] · Adam Büchner[1] · Sebastian Gottfried[1] · Lars Reimer[2] · Simon Ehrmanntraut[1] · Arthur Stück[1]

Received: 30 March 2024 / Revised: 9 October 2025 / Accepted: 24 November 2025
© The Author(s) 2025

Abstract
A scalable, multidisciplinary framework approach is presented in this paper that allows for a flexible organization of nested solution algorithms in conjunction with the simulation ecosystem *FlowSimulator*. Special attention is given to a modular software integration of key simulation components called plugins such as the next-generation CFD software by ONERA, DLR and Airbus (*CODA*), the general-purpose finite-element structure-mechanics method *b2000++*, an elasticity analogy CFD-mesh deformation approach together with energy-preserving mesh transfer methods for loads and deformations. A monolithic view on the complete multidisciplinary system is built combining the *FlowSimulator* ecosystem with the *OpenMDAO* framework. The strategy allows for strong quasi-monolithic multidisciplinary solution algorithms operating on the complete set of degrees of freedom in fully MPI-parallel execution. In Newton-type solution schemes, exact and efficient linearizations based on algorithmic differentiation (AD) of the nonlinear CFD software *CODA* are a key ingredient to an efficient and robust numerical solution. A numerical study is presented for an inviscid, static aeroelastic coupling scenario of reduced complexity using the AGARD 445.6 geometry. Different coupling intensities between the disciplines were considered to systematically study the efficiency and the robustness of the multidisciplinary solution approach. For strongly coupled cases, the preconditioned Newton–Krylov solver outperformed a straightforward nonlinear block-Gauss–Seidel algorithm and showed superlinear convergence.

Keywords Multidisciplinary design analysis and optimization (MDAO) · Multiphysics · Static aeroelastic coupling · CFD-CSM coupling · Fluid-structure interaction (FSI) · FlowSimulator

1 Introduction

Aviation, being part of the transport sector, faces ambitious targets on the path to decarbonization [1]. Significant improvements in design capabilities of aircraft and components are required to meet today's environmental challenges in the short term. To make a step change, virtual analyses and optimizations in large scale and high fidelity are key to accelerate design cycles in the context of new or unconventional aircraft designs without extensive (model) testing. Multidisciplinary solver robustness and efficiency for automated design processes are prerequisites to reliably carry out thousands of coupled computations. Strong disciplinary component solvers need to be complemented by strong multidisciplinary solver capabilities to retain robustness and efficiency.

Today, increasingly mature computational methods are available on the disciplinary level. Often, the CFD component is one of the most challenging methods. Due to the

✉ Arthur Stück
arthur.stueck@dlr.de

Kannan Masilamani
kannan.masilamani@dlr.de

Adam Büchner
adam.buechner@dlr.de

Sebastian Gottfried
sebastian.gottfried@dlr.de

Lars Reimer
lars.reimer@dlr.de

Simon Ehrmanntraut
simon.ehrmanntraut@dlr.de

[1] DLR Institute of Software Methods for Product Virtualization, Simulation Frameworks, Nöthnitzer Str. 46b, Dresden, Germany

[2] DLR Institute of Aerodynamics and Flow Technology, Center for Computer Applications in AeroSpace Science and Engineering (C²A²S²E), Lilienthalplatz 7, Braunschweig, Germany

Published online: 29 April 2026

stiffness and the size of the equation systems, solver efficiency and robustness are crucial, in particular for cases where many accurate, high Reynolds-number aerodynamic predictions are required. Advances made in disciplinary methods need to be systematically carried forward to the level of Multidisciplinary Design Analysis and Optimization (MDAO) [2]. In accordance with the notation well-established in the MDAO community, we consider coupled multiphysics simulations—e.g. for the interaction of fluid and structure—as a subset of multidisciplinary design analysis. The coupling intensity between the individual disciplines can vary greatly depending on the multiphysics problem at hand, which requires appropriate solutions in the MDAO software environment.

Coupling software can generally be subdivided into *framework* and *library* approaches. In the framework coupling approach, the disciplinary components are libraries, the methods of which are called from the framework. Among many examples, we name *OpenMDAO* [3] and *GEMSEO* [4] here, since they are closely related to the environment used in this study. The library coupling approach, on the contrary, offers the coupling functionalities in a software library like *preCISE* [5], *MpCCI* [6, 7] or *comana* [8] which, in turn, is called from the different simulation tools to be coupled with some sort of synchronization required by the targeted coupling algorithms.

The aim of this study is to evaluate the potential of advanced, nested solution schemes provided on the multidisciplinary level in combination with mature implicit disciplinary (general purpose) solvers in a modular MDAO framework approach. Chauhan et al. [9] carry out a systematic MDA solver benchmark using the *OpenMDAO* framework. They suggest (non)linear benchmark problems, based on which they demonstrate the added value of advanced nested algorithms of Newton–Krylov type compared to straightforward fixed-point iteration methods of nonlinear block-Jacobi (NLBJ) or block-Gauss–Seidel (NLBGS) type, particularly for cases of strong couplings. Beyond analytical benchmark problems, Chauhan et al. [10] also consider low-fidelity tools for aircraft design analysis, including a vortex-lattice method (VLM) and small structure-mechanics problems. They systematically compare efficiency and robustness of coupled Newton-Krylov and NLBGS algorithms for different coupling strengths confirming the trends observed in [9]. Moreover, hybrid methods are suggested, in which the schemes switch from NLBGS to coupled Newton–Krylov in the course of the iterative solution. Kenway et al. [11] carry out aeroelastic coupled analyses of aircraft using *OpenMDAO* in conjunction with Euler CFD computations. They compare coupled Newton-Krylov and NLBGS for two load cases. The coupling strength was not varied systematically, however, the Newton–Krylov performed particularly well for the aircraft load case with the stronger aeroelastic coupling.

To conduct coupled analyses and optimizations with the above-mentioned MDAO frameworks, integrated simulation packages or ecosystems in the form of software libraries are required. Yildirim et al. [12] present *MPhys*, a modular multiphysics simulation library built with the *OpenMDAO* framework. The latter is augmented by standard programming interfaces for disciplinary simulation components. The use of *MPhys* with high-fidelity CFD and computational structure-mechanics (CSM) software is presented by Gray and Martins [13]. The HPC framework *FlowSimulator* [14] provides a number of high-fidelity simulation components referred to as plugins, which share mesh coupling data in memory in an MPI domain-decomposed way through a central data management called FlowSimulator Data Manager (*FSDM*). Backhaus et al. [15] show the integration of the *FlowSimulator* with *OpenMDAO*. Ehrmanntraut et al. [16] discuss key features of the MDAO framework integration of the new-generation CFD Software by ONERA, DLR and Airbus (*CODA*)—being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners and jointly owned by ONERA, DLR and Airbus.

Note that all of the algorithmic solution ingredients applied here in the context of MDAO frameworks are well-known and established in the context of disciplinary solvers for years. Newton–Krylov methods, for example, are implemented in a memory-efficient, matrix-free way [17, 18] in order to avoid the unnecessary storage of (block)-sparse matrix operators. The matrix-free evaluation of matrix–vector products also frees from the costs of repeated matrix assembly to some extend. Directional derivatives can be computed or approximated on-the-fly. Moreover, code integration is facilitated, e.g. in combination with straightforward finite-difference approximations to the directional derivatives of the residual function. Block-implicit solution methods are key to address strong couplings that are found on the level of different types of blocks embedded in sparse equation systems. Nested solution methods with hierarchies of solvers and (block-)preconditioners are essential in order to solve large, stiff equation systems. Strong point or element coupling can be effectively addressed by means of point-implicit block-Jacobi type preconditioners in the solver stack, e.g. [19]. A pronounced directional coupling perpendicular to numerically resolved wall boundary layers or in outer areas dominated by the convective transport can be found in RANS-type CFD problems at high Reynolds numbers. This calls for line-implicit preconditioners [20, 21], wherein a direct line inversion is applied to line-blocks according to the Thomas algorithm. These block-inversions,

in turn, are embedded into Jacobi or block-Gauss–Seidel type iterations. Moreover, huge (element) blocks can, for example, result from spatial high-order Discontinuous-Galerkin (DG) discretization schemes [22] or from the time-spectral CFD method, wherein large space-time blocks need to be inverted in the solution process [23]. In the MDAO framework approach pursued here, these implicit solution techniques are transferred from the disciplinary level to the level of the overall multidisciplinary problem. The solution capabilities—such as efficient access to disciplinary residual functions, exact directional derivatives or implicit block solvers—of the disciplinary components (or *FlowSimulator,* plugins being libraries in the framework approach) are exploited by systematically turning them into disciplinary block preconditioners. The latter, in turn, contribute their well-established, above-mentioned implicit solution features to robustly solve the multidisciplinary problems.

In a previous study, nested solution algorithms were evaluated in conjunction with the *FlowSimulator* ecosystem and *OpenMDAO* for two 3D static aeroelastic configurations [24], being the LANN wing test case [25] and the NASA SDT engine configuration [26]. Both were considered in fully-turbulent RANS flow conditions. The transfer of loads and displacements between CFD and CSM mesh—a relevant task in the MDAO context [27] to achieve force consistency and energy conservation—was carried out according to the Finite-Interpolation-Elements approach. In this study, the Moving-Least-Squares approach is considered and discussed. Reimer [28] describes and compares both mesh-transfer approaches in detail. According to the cell-centered finite-volume scheme used in *CODA*, fluid loads need to be passed from CFD integration points, i.e. cell centers, to the mesh nodes of the continuous finite-element method *b2000++*.

The remainder of the paper is organized as follows: the MDAO framework infrastructure used in this work is outlined in Sect. 2. An advanced level of framework integration will be described with the focus on the next-generation CFD and CSM methods *CODA* and *b2000++*, respectively, embedded into the *FlowSimulator* HPC framework. The energy conserving mesh transfer method is briefly described. Numerical experiments are carried out for an elementary static aeroelastic example on the AGARD445.6 wing. It is an inviscid wing case of reduced complexity compared to the previous work [24]. However, the underlying static aeroelastic coupling is similar, and the simplified example allows to systematically study the potential of the suggested framework approach in Sect. 3. Conclusions are drawn together with an outlook given in Sect. 4.

2 MDAO technology stack

2.1 Framework infrastructure

The multidisciplinary simulation infrastructure relies on the integration of the *FlowSimulator* HPC ecosystem in the form of an MDAO package, cf. Fig. 1. The infrastructure allows to execute coupled simulation processes in a way that it is HPC-parallel from end to end. MPI-distributed mesh data is shared in memory via the *FSDM*, which is the bottom layer of the sketch. By this means, file I/O is avoided during the coupled simulation for all components that feature a full integration. The individual disciplinary simulation components called plugins are arranged in the middle layer of Fig. 1, among them the CFD software *CODA* and the general-purpose finite-element CSM solver *b2000++*. The plugins will be described in the subsequent section. They offer Python interfaces and are used as libraries in the framework approach pursued here. MDAO algorithms implemented in the NASA-developed open-source software framework *OpenMDAO* [3] are used to solve the coupled multidisciplinary equations. A unified programming interface—represented in Fig. 1 by the light blue interface boxes named Python API that are attached to the individual plugins—is being designed and established in ongoing research projects to make the *FlowSimulator* infrastructure interoperable with state-of-the-art MDAO frameworks like *OpenMDAO* or *GEMSEO* [4]. Note that some plugins decompose into multiple plugin components to arrive at a modular framework integration that enables advanced MDAO algorithms.

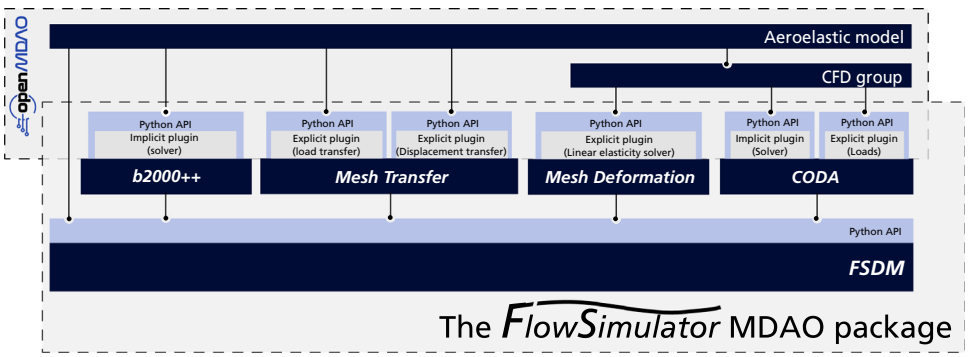


Fig. 1 Three-layered, modular organization of the *FlowSimulator* MDAO package. *FSDM* on the bottom layer provides access to the simulation plugins such as *CODA* on the middle layer. The top layer represents the multidisciplinary framework control driven by *OpenMDAO* in this study

In previous work [29], *OpenMDAO* was integrated into the *FlowSimulator* ecosystem such that it can operate in-place on mesh data arrays provided by the *FSDM*. This is achieved by means of an adapter plugin, *FSOpenMDAO*. It provides a new vector implementation, called `FSDMVector`, for *OpenMDAO* and an extension to the *OpenMDAO* component API to map *FSDM* mesh data sets to *OpenMDAO* system variables. To this end, large-scale vector operations can be executed based on concatenated system block vectors (describing the complete multidisciplinary problem) composed of data sets stored in the *FSDM*. It is illustrated in Fig. 2 for arbitrary plugins called *f* and *g*. The *FSDM* vector blocks, in turn, are MPI-distributed mesh data sets shared between the plugins. Standard coupling objects with unified name schemes are provided so that concatenated vectors holding both values and associated derivatives $\mathrm{d}(.)$ of the disciplinary residuals, states, inputs and outputs – $\underline{R}$, $\underline{\Phi}$, $\underline{\Phi}_I$ and $\underline{\Phi}_O$, respectively – can be accessed systematically by *OpenMDAO*, cf. left-hand side of Fig. 2.

2.2 *FlowSimulator* plugins and their integration

Modularity is key to reach an advanced level of MDAO-framework integration and to benefit from the capabilities offered by frameworks like *OpenMDAO* and *GEMSEO*.

2.2.1 CFD software *CODA*

The CFD software *CODA* [30] is a library by design, which significantly facilitates a modular and granular integration into the suggested MDAO framework approach. It is decomposed into two major plugin components here, which are the implicit CFD equation solver itself (referred to as an implicit component part of the plugin), and a module processing the CFD loads (being forces in the integration points of the CFD surface mesh) acting on the structure model from the CFD solution. The latter component of the *CODA* plugin is called an explicit component since the computation can be considered as an explicit function of the CFD state. In the unified Python API for MDAO, the implicit CFD component offers entry points to compute the (disciplinary) residual vector and to solve the underlying CFD equation system by disciplinary iterative solution methods provided by the *CODA* plugin. The above-mentioned entry-points for generally nonlinear operations are complemented by the corresponding *linear* entry points. By these means, an exact, AD-based residual differentiation is available w.r.t. state and mesh-node coordinates together with the disciplinary *linear* solution capability. The explicit component part of the *CODA* plugin allows to evaluate the vector of integration point forces complemented by its differentiation w.r.t. state and nodal coordinates. The MDAO-framework integration of *CODA* and its performance were in the focus of a previous study by Ehrmanntraut et al. [16].

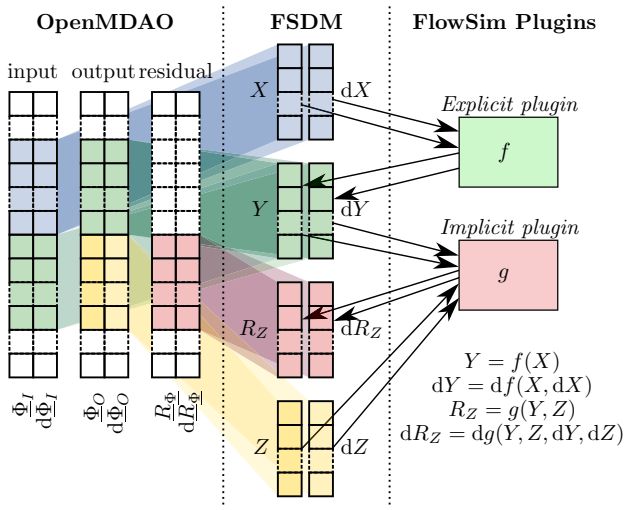


Fig. 2 Full system view by means of concatenated multidisciplinary system vectors for residual R_Φ, input and output vectors, Φ_I and Φ_O, respectively. Directional derivatives are referred to as $\mathrm{d}(.)$. The plugin *FSOpenMDAO* enables *OpenMDAO* to operate in-place on MPI-distributed *FSDM* data sets in conjunction with two arbitrary *FlowSimulator* plugins *f* and *g*

2.2.2 CSM software *b2000++*

The general-purpose finite-element CSM software *b2000++* is the second major simulation package integrated into the MDAO framework approach via the unified *FlowSimulator* MDAO component programming interface. Two *b2000++* plugin components can be accessed from the framework layer: the main *implicit* solver component exposes the degrees of freedom and the residuals of the governing CSM equations to the MDAO framework together with the block solver capabilities; moreover, an explicit component allows to extract the surface displacements from the CSM shell model to be transferred to the CFD surface mesh. As a linear CSM model was used in this study, no additional linearization efforts were required in *b2000++*.

2.2.3 CFD-CSM mesh transfer methods

In addition to the aforementioned plugins *CODA* and *b2000++*, the following explicit *FlowSimulator* mesh-transfer plugins were integrated into the MDAO-framework approach in a way that is integrally force consistent and work conserving:

(a) CFD loads are transferred from the integration points of the cell-centered CFD surface mesh onto the nodes of the CSM surface mesh; and

(b) nodal displacements are transferred from the CSM surface mesh to the CFD mesh.

The transfer of loads and displacements between CFD surface integration points (FI), CFD surface nodes (FN) and CSM surface nodes (SN) is illustrated in Fig. 3. The coupling matrix C contains interpolation coefficients mapping between SN and FI of the underlying meshes. Likewise, the coupling matrix B maps the FI and FN meshes. In the following, the subscripts $\bar{f}$ and s refer to CFD surface integration points and CSM surface nodes, respectively. The displacements v from SN are interpolated to FI by the following matrix–vector product

$$v_{\bar{f}} = C v_{\mathrm{s}} \,. \tag{1}$$

In return, the loads f are interpolated from FI to SN by

$$f_{\mathrm{s}} = C^T f_{\bar{f}} \,. \tag{2}$$

Two mesh transfer approaches are implemented in the *FlowSimulator* environment being the Finite-Interpolation-Elements and the Moving-Least-Squares techniques described and compared in detail by Reimer [28].

The Moving-Least-Squares implementation that was used throughout this study is strictly consistent in integral forces and conservative in work (energy) to machine accuracy, while nodal corrections of moments were neglected in the transfer of local loads. This is deemed to be of higher importance particularly in conjunction with larger disparities in cell spacings between the CFD and the CSM surface meshes. Note that the integral mechanical work computed from the inner product of forces and deformations is consistently evaluated in the integration points FI of the CFD method, whereas it is evaluated node-wise in SN on the CSM surface mesh. By this means, we take account of the natural integration schemes of both the cell-centered finite-volume CFD method and the continuous finite-element CSM approach to ensure discrete conservation of work in the mesh transfer.

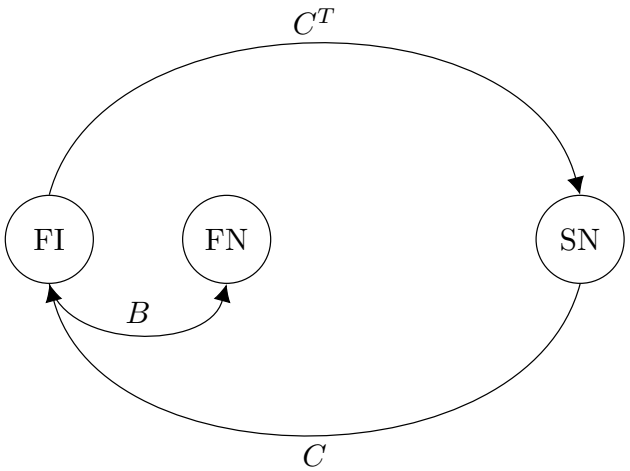


Fig. 3 Mesh transfer of loads (top) and displacements (bottom) between CFD surface integration points (FI), CFD surface nodes (FN) and CSM surface nodes (SN). C and B are coupling or interpolation matrices

2.2.4 CFD mesh deformation

A CFD mesh-deformation plugin was used to propagate the surface displacements imposed by the CSM method into the CFD volume mesh. It is based on the assumption of linear elasticity analogy [31]. Note that the mesh deformation plugin requires the surface displacements in FN, which are reconstructed by interpolation from FI to FN via the operator B, cf. Fig. 3.

All plugins involved, including *CODA* and *b2000++*, support an in-memory communication with the *FSDM* according to Figs. 1, 2. The corresponding disciplinary block vectors for inputs, outputs, residuals and states depicted in Fig. 2 are shared via the *FSDM* subdomain-wise in an MPI-parallel way so that file I/O or additional gather/scatter operations are avoided. It enables *OpenMDAO* to operate inplace on the full multidisciplinary problem representation in an HPC-parallel way. This establishes a monolithic problem view, which is the basis for the nested multidisciplinary solution approaches presented in the numerical experiment in the next Sect. 3.

3 Numerical experiments

3.1 Static aeroelastic setup

The geometry of the AGARD 445.6 configuration [32] was selected here as an elementary benchmark case. It is used to conduct numerical experiments with the multidisciplinary simulation infrastructure described above. A static aeroelastic coupling was carried out at a Mach number of $\mathrm{Ma} = 0.5$. The inviscid computation was performed with the CFD software *CODA* at an angle of attack of 0.2 degree.

3.2 Multidisciplinary equation system

The specific multidisciplinary setup for the simulation ecosystem illustrated in Fig. 1 is described in the following. The CFD solution was computed on an unstructured volume mesh consisting of 305 thousand tetrahedral cells. A standard second-order finite-volume discretization was selected in *CODA* using Roe's convection scheme in conjunction with a linear reconstruction based on Green–Gauss

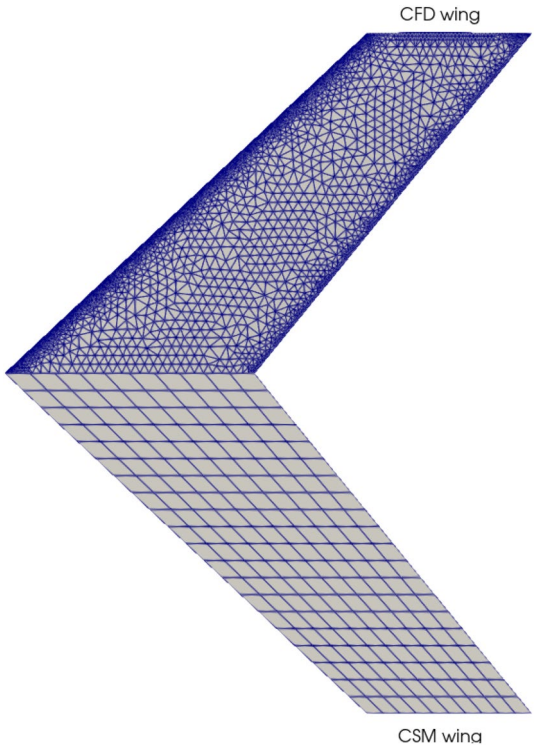


Fig. 4 Surface distributions over the wing for the CFD volume mesh (top) and CSM mesh (bottom)

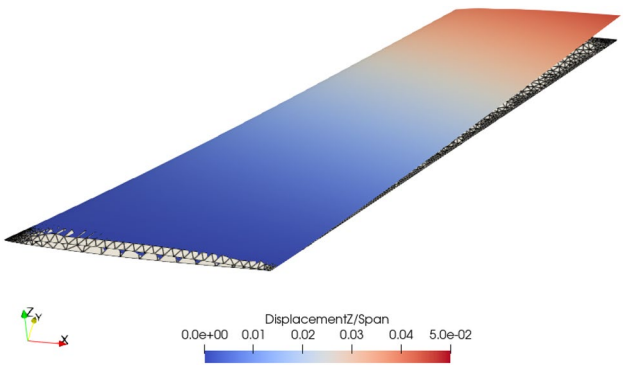


Fig. 5 Computed wing deformation for mid coupling strength

gradients of the flow state U. The aerodynamic loads (forces) $f_{\bar{f}}$ on the CFD surface integration points were post-processed by a dedicated *CODA* plugin (*CODA* loads) and transferred to the CSM surface mesh (subscript s, Fig. 4, bottom) in a conservative way using the Moving-Least-Squares method [28]. The nodal loads on the CSM mesh, in turn, were applied as natural boundary conditions to the finite-element CSM model computed in the *FlowSimulator* plugin *b2000++*. The CSM solution state D contained all degrees of freedom on element nodes. The CSM model was based on 200 quadratic shell elements with mixed interpolation of tensorial components [33]. The stiffness of the CSM model was set to three levels such that a maximum vertical tip deflection divided by the wingspan, $v_{\mathrm{tip}}/L_{\mathrm{span}}$, was $2.64\,\%$ (min), $4.52\,\%$ (mid) and $7.07\,\%$ (max), respectively. The deformation for the maximum coupling strength is shown in Fig. 5. The structural displacements, v_{s}, being the subset of D over the coupling surfaces, were consistently re-transferred to the CFD surface integration points, $v_{\bar{f}}$, and then to the CFD surface mesh v_{f} by the aforementioned Moving-Least-Squares mesh transfer method. Finally, the surface deformations were propagated to the volumetric mesh nodes using the linear elasticity-analogy type finite-element field method implemented in the *FlowSimulator* plugin *FSMeshDeformation*.

The resulting multidisciplinary, nonlinear equation system written in residual form reads

$$R_V(V, v_{\mathrm{f}}) \stackrel{!}{=} 0 \quad \text{CFD mesh deformation} \tag{3a}$$

$$R_U(U, V) \stackrel{!}{=} 0 \quad \text{CODA solver} \tag{3b}$$

$$R_{f_{\bar{f}}}(f_{\bar{f}}, U) \stackrel{!}{=} 0 \quad \text{CODA loads} \tag{3c}$$

$$R_{f_{\mathrm{s}}}(f_{\mathrm{s}}, f_{\bar{f}}) \stackrel{!}{=} 0 \quad \text{load transfer} \tag{3d}$$

$$R_D(D, f_{\mathrm{s}}) \stackrel{!}{=} 0 \quad \text{b2000++pro solver} \tag{3e}$$

$$R_{v_{\mathrm{s}}}(v_{\mathrm{s}}, D) \stackrel{!}{=} 0 \quad \text{b2000++pro displacements} \tag{3f}$$

$$R_{v_{\bar{f}}}(v_{\bar{f}}, v_{\mathrm{s}}) \stackrel{!}{=} 0 \quad \text{displacement transfer (DT)} \tag{3g}$$

$$R_{v_{\mathrm{f}}}(v_{\mathrm{f}}, v_{\bar{f}}) \stackrel{!}{=} 0 \quad \text{displacements to CFD vertices}\,, \tag{3h}$$

with the residual subscripts referring to the associated degree of freedom. Note that the plugins *CODA* and *b2000++* governing the CFD and the CSM disciplines according to Eqs. (3b) and (3e), respectively, are integrated in the form of implicit components. Hence, their algebraic (non)linear residuals and the corresponding directional derivatives are exposed to the multidisciplinary framework layer *OpenMDAO*. The Extended Design Structure Matrix (XDSM) of equation system (3) is depicted in Fig. 6.

The overall multidisciplinary system of equations can be written in block-vector form as $\underline{R}_{\underline{\Phi}} = \underline{0}$, leading to the following equation system for the update

$$\frac{\partial \underline{R}_{\underline{\Phi}}}{\partial \underline{\Phi}} \Delta \underline{\Phi} = -\underline{R}_{\underline{\Phi}}\,. \tag{4}$$

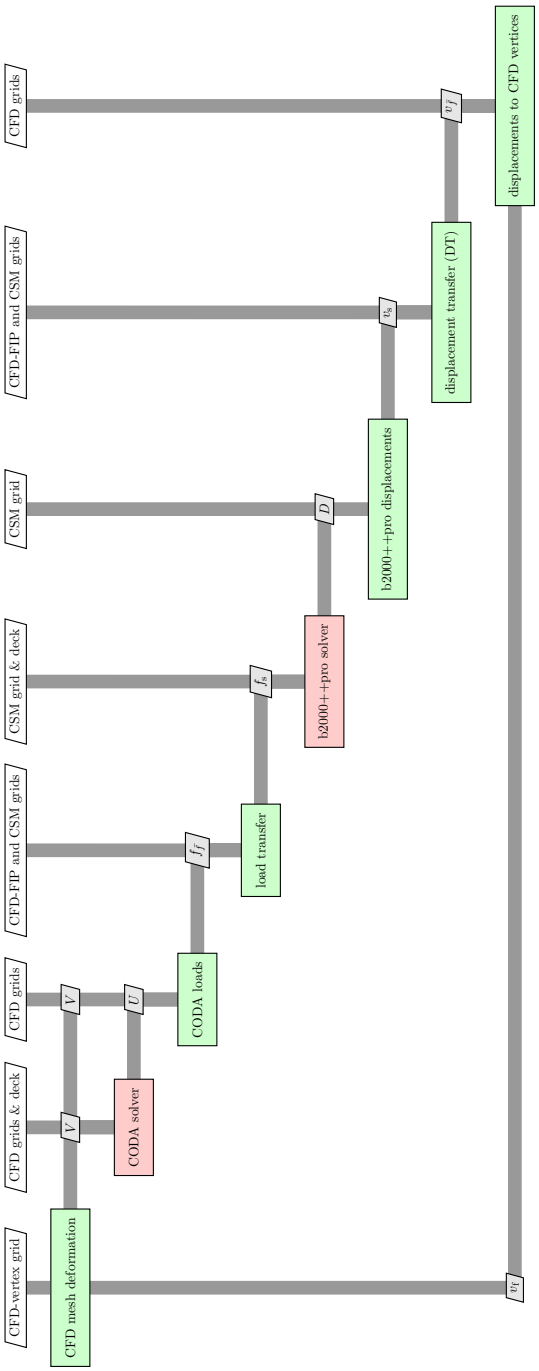


Fig. 6 XDSM of equation system (3)

In order to show the multidisciplinary coupling in the linear operator, the left-hand side of Eq. (4) can be expanded to read

$$\begin{bmatrix} \frac{\partial R_V}{\partial V} & & & & & & & \frac{\partial R_V}{\partial v_f} \\ \frac{\partial R_U}{\partial V} & \frac{\partial R_U}{\partial U} & & & & & & \\ \frac{\partial R_{f_{\tilde{f}}}}{\partial V} & \frac{\partial R_{f_{\tilde{f}}}}{\partial U} & \frac{\partial R_{f_{\tilde{f}}}}{\partial f_{\tilde{f}}} & & & & & \\ & & \frac{\partial R_{f_s}}{\partial f_{\tilde{f}}} & \frac{\partial R_{f_s}}{\partial f_s} & & & & \\ & & & \frac{\partial R_D}{\partial f_s} & \frac{\partial R_D}{\partial D} & & & \\ & & & & \frac{\partial R_{v_s}}{\partial D} & \frac{\partial R_{v_s}}{\partial v_s} & & \\ & & & & & \frac{\partial R_{v_{\tilde{f}}}}{\partial v_s} & \frac{\partial R_{v_{\tilde{f}}}}{\partial v_{\tilde{f}}} & \\ & & & & & & \frac{\partial R_{v_f}}{\partial v_{\tilde{f}}} & \frac{\partial R_{v_f}}{\partial v_f} \end{bmatrix} \cdot \begin{bmatrix} \Delta V \\ \Delta U \\ \Delta f_{\tilde{f}} \\ \Delta f_s \\ \Delta D \\ \Delta v_s \\ \Delta v_{\tilde{f}} \\ \Delta v_f \end{bmatrix}. \tag{5}$$

Mind that in the suggested framework approach, all of the matrix–vector operations according to Eq. (5) are computed in matrix-free mode on the multidisciplinary level. Thus, large sparse-matrix systems do not need to be stored or assembled.

3.3 Multidisciplinary solver study

To showcase the flexibility achieved by the framework approach, different nested iterative numerical solution methods were set up and investigated based on the AGARD 445.6 geometry specified above. With the full system representation accessible on the multidisciplinary framework level, it is straightforward to evaluate the overall system residual vector or its norm to monitor convergence of the coupled system.

3.3.1 Nonlinear block-Gauss–Seidel solver

An elementary solution algorithm applied to the multidisciplinary system of Eq. (3) is a nonlinear block-Gauss–Seidel scheme (MD NLBGS), in which each discipline is solved one after another. It was applied to the multidisciplinary system of *nonlinear* Eq. (3) for the mid coupling strength. Inside the implicit *CODA* plugin, Eq. (3b) was iterated by up to 5 disciplinary backward-Euler steps until a relative tolerance of 10^{-5} was reached. In each backward-Euler step, 50 GMRES sweeps without restarting were conducted, which were right-preconditioned, in turn, by 50 block-Jacobi (BJ) sweeps, applying an LU block-inversion on the mesh-element level. Within the implicit plugin *b2000++*, Eq. (3e), the direct solution method *MUMPS* was applied on the disciplinary level to solve the linear CSM problem. The other disciplines were simply evaluated or processed as explicit functions.

3.3.2 Preconditioned Newton–Krylov solver

The suggested MDAO framework approach is an enabler for advanced, deeply nested multidisciplinary solver

stacks: a multidisciplinary Newton-GMRES (MD Newton→GMRES) method with a linear block-Gauss–Seidel (MD LBGS) preconditioning was applied to the full multidisciplinary problem for the mid coupling strength, viz. MD Newton→GMRES→LBGS. To stabilize the so-called transient phase of the iterative convergence, a single MD NLBGS iteration was performed before entering into Newton's method. For the CFD solver *CODA*, up to five nonlinear disciplinary backward-Euler (BE) sweeps were performed. This approach is referred as MD hybrid Newton→GMRES→LBGS.

At first, the two MD Newton→GMRES→LBGS approaches were investigated for different numbers of GMRES iterations per Newton iteration with and without the hybrid startup option. The convergence of the multidisciplinary residual norm of these two approaches is plotted over the outer nonlinear (NL) iteration count and a metric for accumulated work. To quantify the accumulated work, the innermost linear element block-Jacobi sweeps performed in the *CODA* block-preconditioner were counted. Since the CFD software *CODA* is the most computationally intensive part of the multidisciplinary algorithm, quantifying the accumulated computational effort in terms of innermost CFD iterations is deemed a reasonable measure.

In Fig. 7, the GMRES solver inside Newton's method was executed for a maximum of 2 and 3 iterations, respectively, before the L2-norm of the relative residual normalized with the initial right-hand side was reduced to 10^{-3}. As expected, Newton's method exhibits a linear convergence behavior first, followed by a super-linear convergence after reaching a multidisciplinary relative residual norm of 10^{-6}, once the CFL number relaxation was deactivated in the CFD solver *CODA* after the initial CFL ramp-up. In both MD Newton methods, executing 3 GMRES iterations led to a faster convergence in terms of outer nonlinear iterations and the accumulated work metric. In the legend, the maximum number of iterations defined for the solver is specified in parentheses after the solver name. Especially, the MD hybrid Newton→GMRES(3)→LBGS(1) converged with the least amount of just 8 outer nonlinear iterations and an accumulated work of 797 innermost mesh-element level LU block-inversions. Note that the disciplinary solver *CODA* in the MD Newton method carries out an approximate linear solution of the disciplinary block-problem in the same way as inside the MD NLBGS method described earlier. Accordingly, GMRES(50) and BJ(50) was selected for the CFD-block in combination with an LU-inversion of element/FV blocks. A direct linear solution was carried out inside the *b2000++* block, whereas an iterative linear block solver was used for the CFD block inside MD LBGS. The MD hybrid Newton required slightly more work than the MD Newton scheme. However, it was found that the prior tends to be more reliable in combination with very small numbers of GMRES iterations used here. Moreover, the CFD problems in the multidisciplinary coupling algorithms were consistently started from a solution for which the CFD residual was reduced by four orders of magnitude in rigid mode. Since the multidisciplinary residual is normalized with the free-stream CFD residual, the initial residual values in the convergences histories are below $O(0.1)$.

A further investigation was carried out for the MD Newton→GMRES→LBGS algorithm to find the required

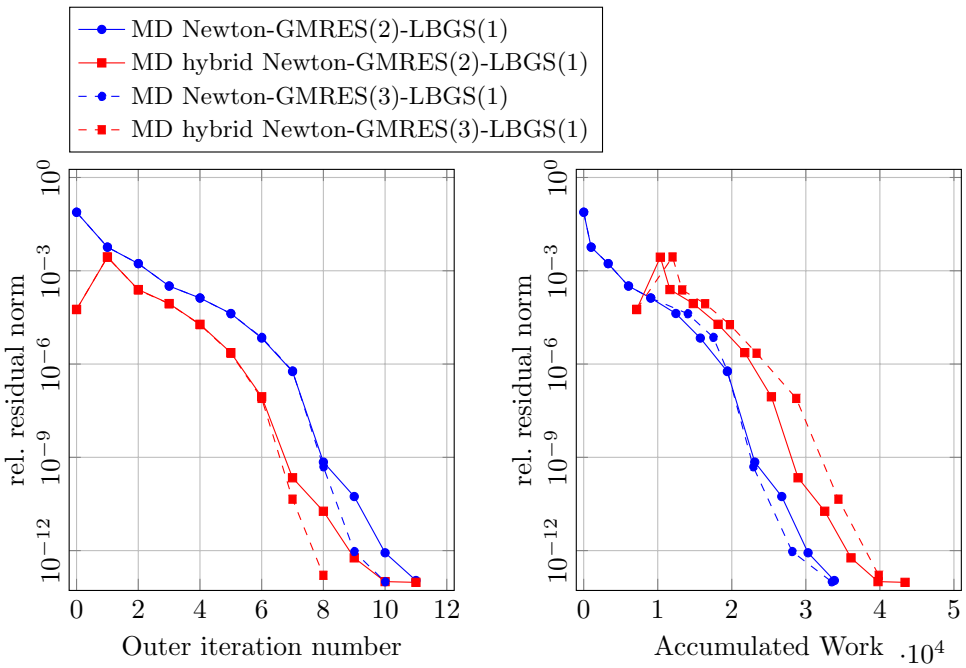


Fig. 7 Convergence of the MD Newton-GMRES-LBGS scheme for different numbers of GMRES iterations per Newton step with and without the hybrid startup option plotted over outer Newton iteration numbers (left) and accumulated work (right)

number of LBGS cycles per GMRES iteration. In Fig. 7, GMRES(3) and GMRES(2) start to differ only from the nonlinear iteration 6. To get a deeper understanding of the reduced convergence rate observed in Fig. 7 (left) from Newton step 7 on, the multidisciplinary problem was initialized with the solution from iteration 6 of MD Newton→GMRES(3)→LBGS(1). The convergences of the multidisciplinary relative residual norm for LBGS with up to 1, 2 and 10 iterations, respectively, are plotted against the outer nonlinear iteration number and accumulated work in Fig. 8. Performing more LBGS iterations improved the convergence of the MD Newton→GMRES→LBGS scheme as more accurate linear solutions led to improved Newton updates. However, the computational work required to obtain such accurate Newton updates was too high for the algorithm to be efficient. This was particularly the case for LBGS(10), as it took 2.6 times more work than LBGS(1) to reach a similar relative residual norm. Performing only 2 LBGS steps seems to be sufficient as it led to a similar convergence as LBGS(10) with relatively little solution effort, i.e. only 1.3 times more than LBGS(1). Even though the effort spent in the LBGS solver is increased, the nonlinear convergence history in conjunction with LBGS(2) does not differ much from its LBGS(1) counterpart. Similarly, performing 2 LBGS iterations from the first MD Newton step on would be relatively expensive for such a minor gain in accuracy. Therefore, considering the overall solution effort, LBGS(1) was chosen as a default setting for the MD Newton→GMRES→LBGS algorithms. Note that we invested relatively high solution efforts into the linear block-preconditioning of the CFD plugin by spending 50 disciplinary GMRES sweeps, in each of which 50 block-Jacobi iterations were carried out. This explains why small numbers of MD GMRES and LBGS sweeps were sufficient to appropriately solve the linear problems. Alternatively, it would be possible to move the solution effort from the disciplinary block preconditioners towards the outer levels of the solution stack being the multidisciplinary linear solvers. However, optimal tuning of the solution effort spent in the individual layers of the solution stack was beyond the scope of this study.

3.3.3 Preconditioned Newton–Krylov versus nonlinear Block-Gauss–Seidel

Previous studies carried out in the context of MDAO frameworks [10] showed that increased coupling strengths between CFD and CSM disciplines can be a significant challenge in terms of solution efficiency and robustness for the multidisciplinary algorithms. Thus, different coupling strengths were considered to systematically evaluate MD Newton→GMRES→LBGS and MD NLBGS algorithms available in the framework approach. Note that only MD *hybrid* Newton→GMRES→LBGS is shown in the plot since it requires fewer nonlinear iterations as presented above. The convergence histories of the multidisciplinary relative residual norm for MD hybrid Newton→GMRES(3)→LBGS(1) and MD NLBGS are plotted over the outer nonlinear iterations and accumulated work in Fig. 9 for the coupling strengths minimum (min), middle (mid) and maximum (max). From one level to the next, the coupling strength was increased by a factor of two, i.e. 1, 2 and 4 for min, mid and max, respectively. As expected, the required number of outer iterations increases with the structural flexibility of the configuration, which intensifies the coupling between the CFD and CSM disciplines. The convergence plotted over nonlinear outer

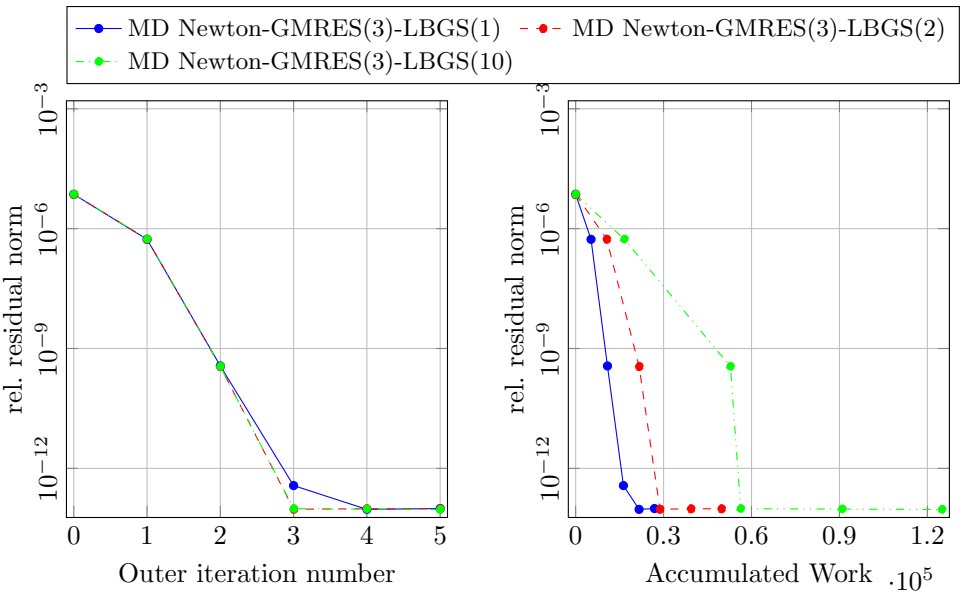


Fig. 8 Convergence of MD Newton-GMRES-LBGS scheme for different numbers of GMRES and LBGS iterations per Newton step against outer nonlinear iteration number (left) and accumulated work (right). The computation was initialized from iteration 6 of the MD Newton-GMRES(3)-LBGS(1) in Fig.7

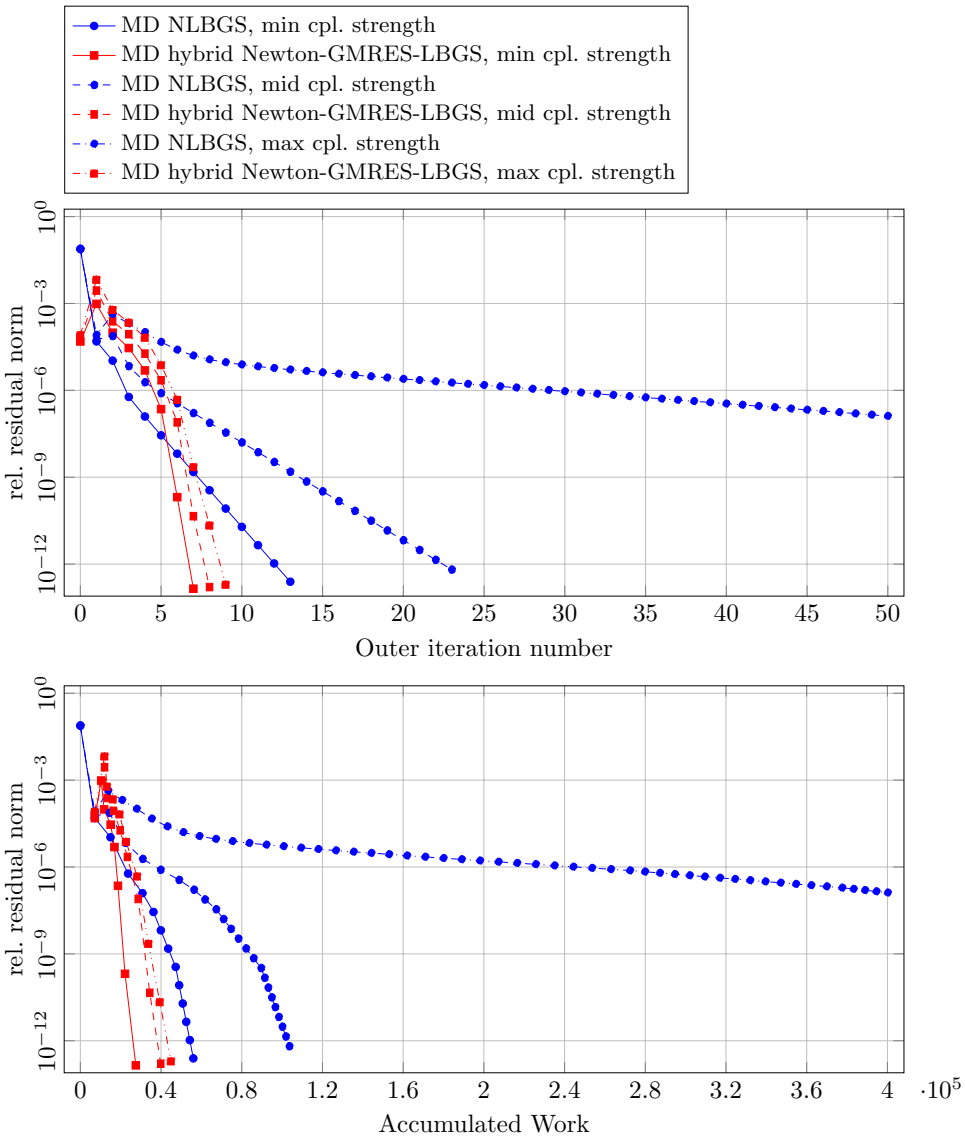


Fig. 9 Convergence of MD NLBGS vs. MD hybrid Newton→GMRES(3)→LBGS(1) schemes for the different coupling strength levels min, mid and max, respectively, plotted over outer nonlinear iteration number (top) and accumulated work (bottom)

iterations of the MD NLBGS algorithm shows a first-order convergence behavior, and the slope of the convergence history gets flatter for the maximum coupling strength, resulting in a slower convergence. In line with the findings of Chauhan et al. [10], the convergence of the MD NLBGS algorithm significantly deteriorates with increased coupling strengths. For the maximum coupling strength, the multidisciplinary relative residual norm of the MD NLBGS algorithm could only be reduced by six orders of magnitude in conjunction with 50 outer nonlinear iterations. On the other hand, the MD hybrid Newton→GMRES(3)→LBGS(1) algorithm converges in a super-linear way for all three coupling strengths, and the required number of outer nonlinear iterations hardly changes when the coupling strength is increased. Even for the maximum coupling strength, the MD hybrid Newton→GMRES→LBGS algorithm was observed to converge faster than the MD NLBGS for the minimum coupling strength. This is a clear indication for a superior robustness of the MD hybrid Newton→GMRES→LBGS algorithm in conjunction with high-fidelity analyses and strong interdisciplinary couplings.

Figure 9 clearly shows the advantages of MD hybrid Newton→GMRES→LBGS over MD NLBGS. The solution effort of MD hybrid Newton→GMRES→LBGS increases only slightly with the coupling strength. The solution effort is only 1.6 times higher when increasing the coupling strength from min to mid. The difference between mid to max coupling strengths for MD hybrid Newton→GMRES→LBGS is only 102 innermost *CODA* preconditioner sweeps, which is very small in comparison to MD NLGBS. This shows that the solution effort of the MD hybrid Newton→GMRES→LBGS algorithm is less dependent on the coupling strength and converges much faster than the MD NLBGS algorithm. It also confirms the multidisciplinary solver efficiency and

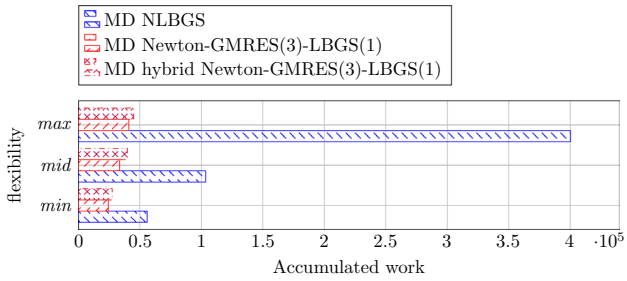


Fig. 10 Comparison of accumulated work of MD NLBGS and MD Newton→GMRES→LBGS with and without the hybrid startup option for different coupling strengths (low, mid and max)

robustness trends observed by Chauhan et al. [9] in the context of MDAO frameworks applied to a tailored numerical benchmark problem.

A comparison of all three algorithms (MD NLBGS, MD Newton and MD hybrid Newton) in terms of accumulated work for different coupling strenghts is given in Fig. 10. With increased coupling strengths, the computational effort of MD Newton→GMRES(3)→LBGS(1) with and without a hybrid startup varies only slightly but for MD NLBGS, the difference is considerable. This confirms the superior algorithmic scalability and robustness of the MD Newton→GMRES→LBGS approach, which is expected to be of higher importance when larger cases, more disciplines and stronger couplings are to be addressed. Figure 10 also reveals that MD hybrid Newton requires a slightly higher solution effort than MD Newton owing to the initial MD NLBGS-type iterations.

The abscissa is scaled in a third way to understand the convergence behavior of MD NLBGS for the maximum coupling strength. The backward-Euler method that was used to solve the disciplinary CFD block within the NLBGS method can also be considered as a CFL-relaxed pseudo-Newton approach. The fact that the CFD contributes the majority of the overall degrees of freedom of the coupled problem motivated a comparison of the required number of (pseudo-)Newton sweeps. The red line and dots in Fig. 11 show the convergence history of MD NLBGS over outer iterations, whereas the blue line and dots depict the convergence of the nonlinear disciplinary CFD block solver within the MD NLBGS sweeps. The spikes are associated with each new MD NLBGS cycle. It also shows that even though the relative residual norm of the CFD discipline is sufficiently reduced, the multidisciplinary relative residual norm is reduced only up to 10^{-6} due to the multidisciplinary coupling.

Clearly, the convergence behavior and performance strongly depend on the choice of disciplinary (internal) solver parameters within the stack of algorithms. However, such parameter studies are expected to be case-specific, and systematic parameter optimizations have not been carried out yet and are beyond the scope of this study. Instead of advocating one specific hierarchy of multidisciplinary algorithms, we want to underline the flexibility benefits of the suggested MDAO framework approach. It allows to conveniently change and adapt the selected multidisciplinary solver stack by a few parameters in the framework.

4 Summary and outlook

A multidisciplinary framework approach based on the *FlowSimulator* HPC ecosystem was presented. It features a modular integration of high-fidelity simulation plugins like the CFD software *CODA*, the general-purpose finite-element CSM method *b2000++* and an energy-conserving mesh transfer approach between CFD and CSM surface meshes. The suggested MDAO framework approach is able to generate a full monolithic view on the multidisciplinary system. *OpenMDAO* was used in this study to control the *FlowSimulator* simulation ecosystem. The suggested approach was

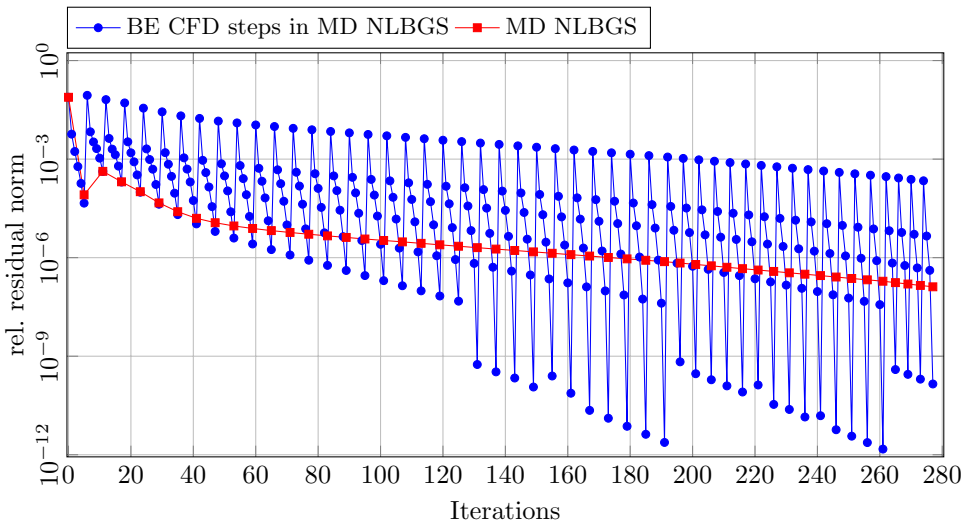


Fig. 11 Residual norm over CFD backward-Euler (or approximate CFL-relaxed Newton) steps in every MD NLBGS for the maximum coupling strength

used to solve a 3D static aeroelastic problem of reduced complexity based on the geometry of the AGARD 445.6 configuration. We have successfully demonstrated that with a seamless *FlowSimulator* MDAO integration of *CODA* and *b2000++* (among other plugins), different multidisciplinary solver stacks like nonlinear block-Gauss–Seidel (MD NLBGS), nested Newton–Krylov preconditioned by linear-block-Gauss–Seidel (MD Newton→GMRES→LBGS) or combinations thereof (hybrid), can be flexibly configured and applied to solve the problem in an MPI-parallel way. The presented MD Newton-type schemes benefit from the systematic derivative accumulation over the disciplines automated by the MDAO framework approach.

As expected and well-known from disciplinary (CFD) solvers, it was confirmed that a reliable and efficient solution of the *multi*disciplinary linear problems inside Newton's method is crucial to leverage the potential of the multidisciplinary Newton approach. Moreover, hybrid multidisciplinary schemes turned out to be valuable for the multidisciplinary convergence process to overcome the transient startup phase of the solution process, before the superlinear convergence of the Newton scheme can be achieved. We have identified how the overall multidisciplinary solver efficiency and robustness are related to key solver parameters on the individual levels of the nested multidisciplinary solver stacks.

For the given problem, we observed that the multidisciplinary preconditioned Newton solver is superior to the straightforward MD NLBGS approach in terms of efficiency and robustness, particularly with growing interdisciplinary coupling strengths. However, the solver parameters involved have not been studied and optimized systematically; for more disciplines, larger meshes and more involved couplings, the availability of advanced multidisciplinary coupling algorithms is deemed to be a key prerequisite to perform multidisciplinary analyses and optimizations of future aircraft configurations in large scale and high fidelity.

In the next steps, we want to demonstrate that – by means of the suggested *FlowSimuator* MDAO API – the HPC ecosystem will be interoperable and able to show its potential in conjunction with different state-of-the-art MDAO frameworks beyond *OpenMDAO*. Moreover, further plugins need to be modularly integrated into the MDAO framework approach to take advantage of the algorithmic capabilities for challenging industrial applications. High-fidelity aeroelastic analyses of engine configurations, which often involve FE-CSM solid models with highly-resolved volume meshes for rotor/stator blades, are expected to likewise benefit from the increased coupling capabilities presented here – however, with a different distribution of the relative disciplinary work load. This is currently being studied in ongoing project work. Like the MDA solution capabilities that were in the focus of this study, gradient-based MDO cases will similarly benefit from the improved infrastructure – for example to automate the multidisciplinary reverse-mode gradient accumulation and to efficiently solve coupled adjoint problems for large-scale MDO. Moreover, we think that the MDAO framework developments are beneficial from a usability perspective, too: in addition to the advanced algorithmic and HPC-parallel capabilities on the multidisciplinary level, the framework approach will enable design engineers to set up and manage MDAO workflows of growing complexity and heterogeneity.

Acknowledgements The funding of parts of these investigations from the LUFO VI-1 project DARWIN ("'Digitaler Faden'-basierte Triebwerksentwicklung mit eingebetteter Künstlicher Intelligenz unter Verwendung hochgenauer Simulationsmethoden", Teilprojekt "Integration von Schlüsseltechnologien aus Simulation und maschinellem Lernen in die Triebwerksentwicklung", FKZ: 20D1911B) by the German Federal Ministry for Economic Affairs and Energy (BMWE) is gratefully acknowledged. The authors gratefully acknowledge the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARO is partially funded by the Ministry of Science and Culture of Lower Saxony and the German Federal Ministry for Economic Affairs and Energy (BMWE).

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. European Commission: European Green Deal (2019)
2. Slotnick, J.P., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., Mavriplis, D.J.: CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences (2014)
3. Gray, J.S., Hwang, J.T., Martins, J.R.R.A., Moore, K.T., Naylor, B.A.: OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. Struct. Multidiscip.

Optim. **59**(4), 1075–1104 (2019). https://doi.org/10.1007/s00158-019-02211-z
4. Gallard, F., Vanaret, C., Guénot, D., Gachelin, V., Lafage, R., Pauwels, B., Barjhoux, P.-J., Gazaix, A.: Gems: A python library for automation of multidisciplinary design optimization process generation. In: 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (2018)
5. Chourdakis, G., Davis, K., Rodenberg, B., Schulte, M., Simonis, F., Uekermann, B., Abrams, G., Bungartz, H., Cheung Yau, L., Desai, I., Eder, K., Hertrich, R., Lindner, F., Rusch, A., Sashko, D., Schneider, D., Totounferoush, A., Volland, D., Vollmer, P., Koseomur, O.: preCICE v2: A sustainable and user-friendly coupling library. Open Research Europe **2**(51) (2022) https://doi.org/10.12688/openreseurope.14445.2
6. Joppich, W., Kürschner, M.: MpCCI - a tool for the simulation of coupled applications. Concurr. Comput. Pract. Exp. **18**, 183–192 (2006). https://doi.org/10.1002/cpe.913
7. Wolf, K., Bayrasy, P., Brodbeck, C., Kalmykov, I., Oeckerath, A., Wirth, N.: MpCCI: Neutral Interfaces for Multiphysics Simulations (2017). https://doi.org/10.1007/978-3-319-62458-7_7
8. Radtke, L., Lampe, T., Abdel-Maksoud, M., Düster, A.: Partitioned simulation of multi-field problems – efficient and robust coupling of fluids and structures. PAMM **18**(1), 201800424 (2018). https://doi.org/10.1002/pamm.201800424
9. Chauhan, S.S., Hwang, J.T., Martins, J.R.R.A.: Benchmarking approaches for the multidisciplinary analysis of complex systems using a taylor series-based scalable problem. In: Schumacher, A., Vietor, T., Fiebig, S., Bletzinger, K.-U., Maute, K. (eds.) Advances in Structural and Multidisciplinary Optimization, pp. 98–116. Springer, Cham (2018)
10. Chauhan, S.S., Hwang, J.T., Martins, J.R.R.A.: An automated selection algorithm for nonlinear solvers in mdo. Struct. Multidiscip. Optim. **58**, 349–377 (2018). https://doi.org/10.1007/s00158-018-2004-5
11. Kenway, G.K.W., Kennedy, G.J., Martins, J.R.R.A.: Mmphys: a modular multiphysics library for coupled simulation and adjoint derivative computation. AIAA J **52**(5) (2014) https://doi.org/10.2514/1.J052255
12. Yildirim, A., Jacobson, K.E., Anibal, J.L.: Mmphys: a modular multiphysics library for coupled simulation and adjoint derivative computation. Struct. Multidisc. Optim. **68**(15), 183–192 (2025). https://doi.org/10.1007/s00158-024-03900-0
13. Gray, A.C., Martins, J.R.: Aerostructural Optimization of the Simple Transonic Wing Using MPhys, ADflow, and TACS. AIAA 2025-2813 (2025)
14. Reimer, L.: The FlowSimulator—A Software Framework for CFD-related Multidisciplinary Simulations. European NAFEMS Conference Computational Fluid Dynamics (CFD) – Beyond the Solve (2015). elib.dlr.de/100536/
15. Backhaus, T., Gottfried, S., Merle, A., Hwang, J.T., Stueck, A.: Modularization of High-Fidelity Static Aeroelastic MDO Enabling a Framework-based Optimization Approach for HPC. AIAA 2021-1236 (2021)
16. Ehrmanntraut, S., Büchner, A., Gottfried, S., Stück, A.: Scalable framework integration of CODA for a multidisciplinary preconditioned matrix-free Newton-Krylov method. In: Dillmann, A., Heller, G., Krämer, E., Wagner, C. (eds.) New Results in Numerical and Experimental Fluid Mechanics XIV, vol. 154. Springer, Cham (2024)
17. Luo, H., Baum, J.D., Löhner, R.: An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids. Comput. Fluid. **30**(2), 137–159 (2001). https://doi.org/10.1016/S0045-7930(00)00011-6
18. Bassi, F., Botti, L., Colombo, A., Crivellini, A., Franciolini, M., Ghidoni, A., Noventa, G.: A p-adaptive matrix-free discontinuous galerkin method for the implicit les of incompressible transitional flows. Flow Turbul. Combust. **105**(8), 437–470 (2020). https://doi.org/10.1007/s10494-020-00178-2
19. Pierce, N.A., Giles, M.B.: Preconditioned multigrid methods for compressible flow calculations on stretched meshes. J. Comput. Phys. **136**(2), 425–445 (1997). https://doi.org/10.1006/jcph.1997.5772
20. Eliasson, P., Weinerfelt, P., Nordström, J.: Application of a Line-Implicit Scheme on Stretched Unstructured Grids. AIAA 2009-163 (2009)
21. Langer, S., Schwöppe, A., Kroll, N.: Investigation and comparison of implicit smoothers applied in agglomeration multigrid. AIAA J **53**(8) (2015) https://doi.org/10.2514/1.J053367
22. Ahrabi, B.R., Mavriplis, D.J.: An implicit block ilu smoother for preconditioning of newton-krylov solvers with application in high-order stabilized finite-element methods. Comput. Methods Appl. Mech. Eng. **358**, 112637 (2020). https://doi.org/10.1016/j.cma.2019.112637
23. Mundis, N.L., Mavriplis, D.J.: Toward an optimal solver for time-spectral fluid-dynamic and aeroelastic solutions on unstructured meshes. J. Comput. Phys. **345**, 132–161 (2017). https://doi.org/10.1016/j.jcp.2017.04.067
24. Masilamani, K., Büchner, A., Ehrmanntraut, S., Gottfried, S., Reimer, L., Stueck, A.: A Scalable MDAO Framework Approach Enabling Nested Solution Algorithms for Static Aeroelastic Coupling Scenarios. AIAA 2024-4403
25. Zwann, R.J.: Lann wing, pitching oscillation. in: Advisory group for aerospace research & development. AGARD Report No.702. Compendium of Unsteady Aerodynamic Measurements. Addendum No.1. (1985)
26. Envia, E.: Fan noise source diagnostics test - vane unsteady pressure results. In: 8th AIAA/CEAS Aeroacoustics Conference & Exhibit (2002). https://doi.org/10.2514/6.2002-2430
27. Schie, S.P., Warner, M., Fletcher, A., Hwang, J.T.: Enforcing Work Conservation in Modular Aeroelastic Coupling For Multidisciplinary Design Optimization. AIAA 2024-2411 (2024)
28. Reimer, L.: Simulation environment for CFD-based aeroelastic analysis of aircraft. PhD thesis, RWTH Aachen University (2020)
29. Backhaus, T., Gottfried, S., Merle, A., Hwang, J.T., Stück, A.: Modularization of High-Fidelity Static Aeroelastic MDO Enabling a Framework-based Optimization Approach for HPC (2021). https://doi.org/10.2514/6.2021-1236
30. Leicht, T., Vollmer, D., Jägersküpper, J., A. Schwöppe, R.H., Fiedler, J., Schlauch, T.: DLR-Project DIGITAL-X – Next Generation CFD Solver FLUCS. Deutscher Luft- und Raumfahrtkongress (DLRK), Braunschweig, September (2016). https://www.dglr.de/publikationen/2017/420027.pdf
31. Rempke, A.: Netzdeformation mit Elastizitätsanalogie in multidisziplinärer FlowSimulator-Umgebung. 20th STAB DGLR Conference, Brunswick, Germany (2016). elib.dlr.de/111205/
32. Yates, E.C.: AGARD standard aeroelastic configurations for dynamic response (1987)
33. Bathe, K.J.: Finite Element Procedures, 2nd edn. Watertown, MA (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.