Performance Study for High-Fidelity Static Aeroelastic Analyses using the OpenMDAO Framework and the FlowSimulator HPC Ecosystem

Kannan Masilamani*, Adam Büchner†, Sebastian Gottfried‡ and Arthur Stück§
German Aerospace Center (DLR)
Institute of Software Methods for Product Virtualization
Nöthnitzer Straße 46 b, 01187 Dresden, Germany

In this paper, we evaluate the multidisciplinary solver performance of an MDAO framework approach integrating the FlowSimulator HPC simulation ecosystem, the CFD Software by ONERA, DLR, Airbus (CODA) and OpenMDAO for large-scale aerodynamic and static aeroelastic analyses. For a robust and efficient implicit solution of stiff multidisciplinary problems, nested solver stacks are applied and evaluated with a tight/monolithic coupling between the disciplines. Two numerical studies are presented in this paper. First, preconditioned Newton–Krylov methods implemented in the MDAO framework approach are applied to solve the RANS equations for the DPW-5/Common Research Model (CRM) wing-body configuration on a sequence of hybrid meshes L2–L5, resulting in 61 million to 1.2 billion degrees of freedom handled by the MDAO framework. The performance of the framework-based solver integrating the CFD software CODA as a single implicit component is compared against the corresponding solution algorithms directly implemented in the performance-optimized CFD solver CODA. In a second MDA solver study, we consider a simplified version of the aeroelastic NASA R4/SDT engine test case in combination with a CSM finite-element (FE) solid model of the flexible rotor blade in b2000++ and a fully-turbulent RANS model in CODA implicitly coupled in the MDAO framework environment. We investigate both algorithmic and parallel performances for multidisciplinary, Newton-GMRES solvers in combination with both parallel linear block-Jacobi and sequential linear block-Gauss–Seidel preconditioning. The parallel efficiency of the concurrent linear block-Jacobi preconditioner outperformed the algorithmic advantage of the linear block-Gauss–Seidel preconditioner for this case.

I. Introduction

Software solutions for multidisciplinary design analysis and optimization (MDAO) need to keep step with state-of-the-art (disciplinary) high-fidelity simulation capabilities to be successful and applicable in high-fidelity multidisciplinary design processes of aircraft. Gradient-enabled MDAO frameworks like OpenMDAO [1] and GEMSEO [2] have become increasingly mature. The framework developments need to be accompanied by the development and integration of (high-fidelity) simulation packages such as MPhys [3, 4] or the FlowSimulator [5, 6] to be ready for large-scale MDAO applications. Monolithic framework solution algorithms are systematically benchmarked in [7] based on generic problems. Further algorithmic solver studies are presented by [8]. Different mesh resolutions L1 to L3 are considered for the Common Research Model (CRM) [9] to compare different multidisciplinary solution algorithms [10].

This paper is a continuation of previous solver studies using a monolithic MDAO framework approach outlined in Section II integrating OpenMDAO and the FlowSimulator ecosystem [6, 11]. Attention is given to the evaluation of the multidisciplinary framework solver capabilities with larger multidisciplinary problems, in which most of the degrees of freedom often stem from the CFD discipline. To this end, the (monolithic) solution algorithms provided by the MDAO framework are benchmarked against the corresponding algorithms implemented in the performance-optimized CFD Software by ONERA, DLR and Airbus (CODA) [12] in terms of algorithmic and parallel efficiency. It is a challenge to retain the disciplinary solver scalability (of the CFD method) in advanced multidisciplinary workflows driven by MDAO

*Researcher, DLR Institute of Software Methods for Product Virtualization, Department Simulation Frameworks.
†Researcher, DLR Institute of Software Methods for Product Virtualization, Department Simulation Frameworks
‡Team Lead, DLR Institute of Software Methods for Product Virtualization, Department Simulation Frameworks.
§Head of Department, DLR Institute of Software Methods for Product Virtualization, Department Simulation Frameworks.

frameworks. The finer mesh levels of the DPW-5/CRM configurations (hybrid mesh levels L2-L5), which result in more than 10^9 degrees of freedom, are considered to evaluate the implicit solver capabilities of the integrated MDAO framework approach. The CFD-centered framework scalability study is presented in Section III.A. The above-mentioned, single-disciplinary solver scalability study is complemented by an investigation of the multidisciplinary coupling capabilities of the framework approach involving monolithic, preconditioned Newton–Krylov solvers applied to a multidisciplinary system. We consider the NASA R4/SDT configuration [13] in Section III.B since it involves a solid CSM/FE model (which contributes more degrees of freedom than beam and shell FE models often used for external aircraft) motivating a parallel processing of the CFD and CSM disciplines via parallel groups. Moreover, different solver/preconditioner nestings and solver parameters (on all levels of the solver stack) are investigated for the monolithic multidisciplinary solution approach, before conclusions are drawn in Section IV.

II. MDAO Framework Approach

In this study, key plugins of the FlowSimulator HPC ecosystem are combined with the state-of-the-art MDAO framework OpenMDAO with the help of the integration libraries FSMDAO and FSOpenMDAO to enable flexible, framework-driven multidisciplinary problems. In this section, these software components are shortly described and their integration is briefly explained.

The (single-)disciplinary solvers used in the numerical experiments presented in this study are part of the FlowSimulator ecosystem – a comprehensive, HPC-enabled simulation environment for multiphysics, CFD-coupled workflows. Individual simulation software components responsible for a single discipline are provided as so-called *plugins*. Mesh data sets are shared in memory between the plugins via the FlowSimulator Data Manager (FSDM) in an end-to-end MPI-parallel fashion.

Among the simulation plugins used in this study is the CFD Software CODA. It is being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR and Airbus. CODA is a CFD *library* by design which provides APIs on different levels of abstraction. We use CODA's spatial discretization API, granting the MDAO framework access to evaluate the underlying residuals and its partial forward/reverse-mode derivatives with respect to its input variables. A solver API allows to solve the (linearized) CFD block-equation system. In the framework-driven, multidisciplinary monolithic solution process pursued in this study, the overall coupled problems are solved by the framework – involving an implicit-Euler pseudo-time stepping towards steady state (pseudo-transient continuation), Krylov-GMRES methods and multidiscipinary block-Gauss–Seidel/Jacobi type preconditioners. The linear block preconditioning of the disciplinary block systems embedded in the nested framework solver are delegated to the disciplinary plugins like the CFD.

For the multidisciplinary study, the CFD simulation is coupled with the general-purpose finite-element CSM solver b2000++ which is also available as a FlowSimulator plugin. It offers linear/nonlinear, static/dynamic analysis capabilities for a wide range of engineering problems. It supports many types of finite elements – including beams, shells and solids.

CFD loads are transferred from CODA's boundary-face integration points to the CSM surface mesh via the method of moving least-square (MLS) [14]. The transposed operators are used to re-transfer CSM deformations to the CFD mesh in the reverse direction in a consistent and conservative way. Finally, the surface-level CSM deformations on CFD vertices are propagated into the volume grid by the plugin FSMeshDeformation according to an elastic analogy (EA) approach.

For this multidisciplinary solver performance study, we used and evaluated key technology bricks developed earlier in conjunction with larger numerical cases and concurrent (block-Jacobi) execution of disciplines, among them are:

- The MDAO framework ecosystem composed of the FlowSimulator HPC ecosystem and OpenMDAO was described in [5]. By this means, OpenMDAO was enabled to operate in-place on domain-decomposed FSDM vector fields for mesh data. The study was based on the CFD solver TAU considered a predecessor of the CFD software CODA. With TAU, the level of framework integration was limited, giving access only to nonlinear residuals, nonlinear solver calls and the adjoint-based procedure computing sensitivity derivatives for the presented static aeroelastic shape optimization problem.
- The modular MDAO framework integration of CODA was presented in [15]. In this paper, we focused on the framework integration of the CFD software CODA as an implicit component to support pseudo-monolithic multidisciplinary solvers offered by the framework. A multidisciplinary Newton–GMRES method was implemented in the framework and demonstrated to couple an ONERA M6 wing to a torsional spring.
- A multidisciplinary solver study was performed in [6]. The general purpose CSM software b2000++ was

integrated as an implicit component in the framework to solve coupled problems by means of preconditioned Newton–GMRES solvers. This multidisciplinary framework solver was systematically studied for the 3D LANN wing and the NASA R4/SDT engine configuration indicating the algorithmic advantages of multidisciplinary Newton–GMRES solvers over nonlinear block-Gauss–Seidel (NLBGS) schemes for CFD coupled problems.
- In a recent study [11], we examined the algorithmic efficiency and the robustness advantages of multidisciplinary pseudo-monolithic preconditioned Newton–GMRES solvers. The multidisciplinary preconditioned Newton–GMRES solvers outperformed elementary nonlinear block-Gauss–Seidel solvers, particularly in conjunction with flexible wings. This investigation was performed on a 3D static aero-elastic problem based on the AGARD 445.6 geometry in Euler flow.

III. Multidisciplinary Solver Performance Studies

The aim of the multidisciplinary solver studies is to evaluate the efficiency of the MDAO framework approach. The latter takes advantage of the FlowSimulator HPC ecosystem integrating CODA and b2000++ as implicit component plugins. Two different computational scenarios were considered in this paper:

The *first scenario* is motivated by the assumption that for many CFD-coupled workflows the bulk of the computational work – or the number of degrees of freedom – is associated with the CFD component. In the context of coupled static-aeroelastic analyses of aircraft, we need to converge large, nonlinear steady-state RANS equation systems for high Reynolds numbers in transonic flow. In the disciplinary CFD solvers, the equation systems are usually converged by means of implicit solution procedures. This gave rise to the first performance study, in which we challenged the MDAO framework approach by applying its nested multidisciplinary solver stack based on a block-preconditioned Newton–Krylov method to a large-scale CFD-only problem in conjunction with the CFD software CODA. The computational efficiency was compared against the performance of the same solver stacks implemented in the performance-optimized CFD software. The target was to evaluate the overhead resulting from the multiphysics environment – though it does not add value for the analysis of a single-disciplinary CFD-only case.

In the first part of the computational study, attention was drawn to the efficiency of the MDAO framework approach without considering any multiphysics coupling effects or challenges. Therefore, we considered a static-aeroelastic problem consisting of the NASA R4/SDT rotor configuration in the *second part of the study*, in which the flexible rotor was modelled as a solid using a finite-element structure-mechanics approach coupled to the CFD. The same framework-driven preconditioned Newton–Krylov solver stack was used in an MPI-parallel execution. In this scenario, we investigated the parallel computation capabilities of the framework for an MPI-distributed execution in combination with both sequential block-wise linear-block-Gauss–Seidel (LBGS-SG) and parallel linear-block-Jacobi (LBJ-PG) type preconditioning. LBGS-GS and LBJ-PG were implemented by means of sequential and parallel groups in OpenMDAO.

All computations were performed on the DLR HPC cluster CARO. It has 2 AMD EPYC 7702 (64 cores; 2,0 GHz) per node with maximum aggregated bandwidth 557 TB/s. Computations for the first scenario were carried out using up to 2048 cores and for the static-aeroelastic problem 256 cores were used.

A. Scalability Study for DPW-5/CRM Wing-Body Configuration

We considered a sequence of unstructured/hybrid meshes, Figure 7, of levels L2–L5, cf. Table 1, for the DPW-5/CRM wing-body configuration [16] at a Mach number of $Ma = 0.85$, a Reynolds number $Re = 5 \times 10^6$ and $\alpha = 2.209°$ angle of attack using the negative extension of the Spalart–Allmaras turbulence model (SA-neg).

Grid	No. of Elems	No. of DoFs
L2	10.1 M	60.6 M
L3	24.1 M	144.6 M
L4	81.0 M	486.0 M
L5	192.5 M	1155.0 M

Table 1 Number of elements and degrees of freedom on each grid level of the DPW-5/CRM wing-body configuration.

The solver stack illustrated on the left-hand side of Figure 2 was used to converge the CFD-only problem in CODA. The performance-optimized implicit solver implemented in CODA was considered as a reference to compare it against

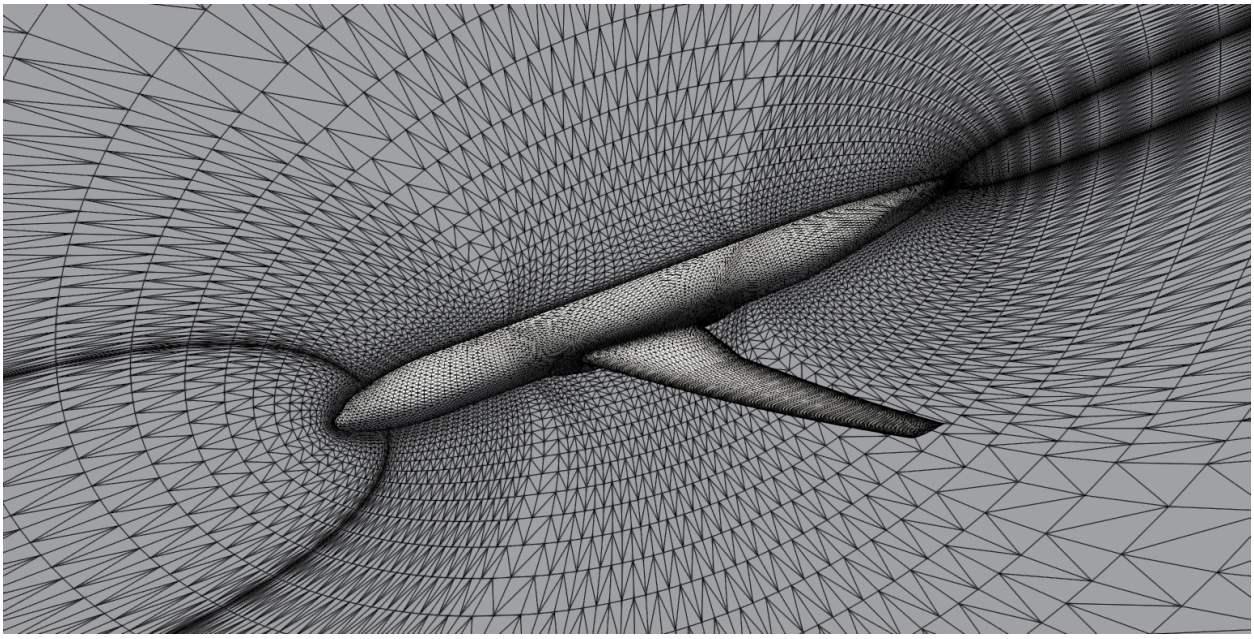


Fig. 1 L1 hybrid mesh of the DPW-5/CRM wing-body configuration.

the framework-based implementation depicted on the right-hand side. We reproduced this algorithm in the presented MDAO framework environment by removing any other disciplinary simulation components from the MDAO framework approach, illustrated by the red X in Figure 2. The counterpart of the CFD-implemented implicit-Euler method is a pseudo Newton-solver in combination with a pseudo-transient continuation approach on the right-hand side. Each implicit pseudo-time step a preconditioned flexible GMRES solver (FGMRES) was applied. The CFD-implemented FGMRES (left) was replaced by its MDAO-framework counterpart implemented by the PETSc scientific computation toolkit [17]. Note that the preconditioning by one block-Gauss–Seidel or block-Jacobi sweep (run once) is without any algorithmic effect for a single-disciplinary case and is only relevant for multidisciplinary couplings. To reproduce the solver stack of the standalone CFD solver in the MDAO framework, the linear element-block-Jacobi preconditioner of the CFD software CODA – highlighted by the dashed frame – was integrated into the framework to perform the block-preconditioning of the implicit CFD component.

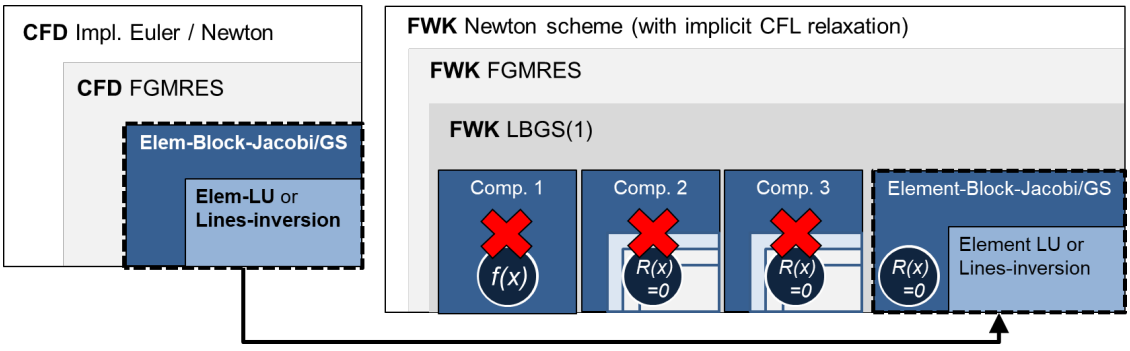
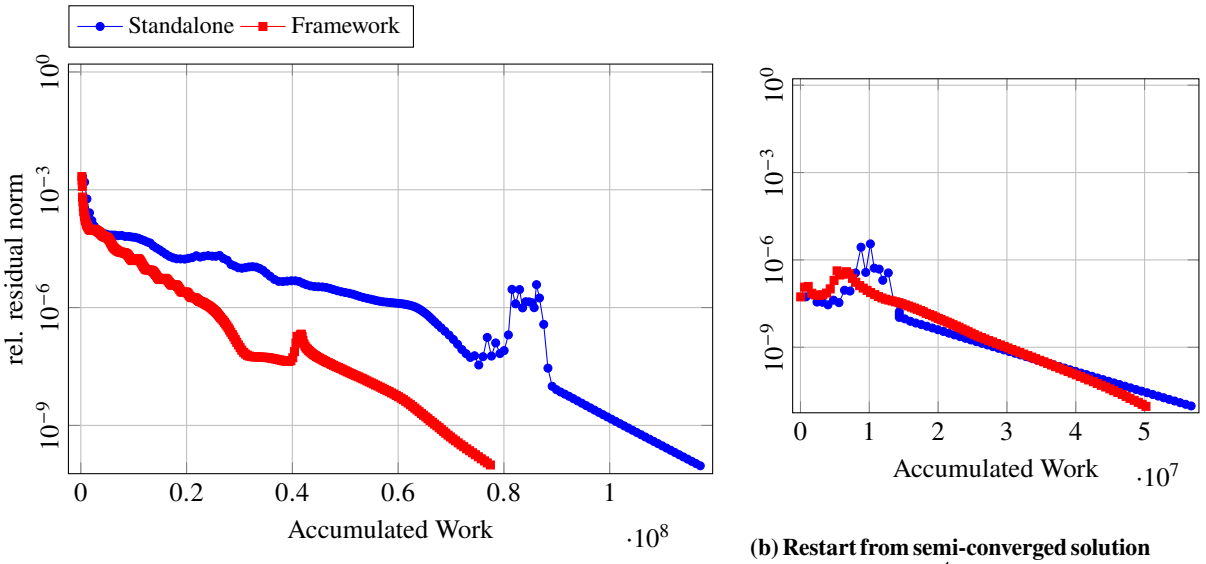


Fig. 2 Sketch of CFD solver-stack Implicit-Euler→FGMRES→Elem-BJ→Elem-LU/Lines-Inversion (left) reproduced inside FlowSimulator-OpenMDAO framework approach (right). The linear preconditioner part, cf. dashed frame, was integrated into the multidisciplinary framework solver stack. Other framework components were eliminated for the purpose of a one-to-one efficiency comparison between CFD-standalone and framework-integrated implementations.

The nonlinear convergence histories for both standalone and framework-based computations on grid level L3 of the DPW-5/CRM configuration are depicted in Figure 3a. The convergence is plotted over an accumulated-work metric

measured in innermost element-block-Jacobi iterations of the CFD solver. Both standalone and framework GMRES were solved for a maximum of 20 iterations (with a Krylov-subspace size of 20), or to a relative tolerance of 10^{-3}. Due to differences in the CFL-relaxation ramp, the nonlinear convergence behavior is different. In the subsequent solver study, the nonlinear solvers were restarted and continued from the same solution at a normalized L2-norm of the nonlinear residual vector of approximately 10^{-7} with a constant CFL number of 10^4, cf. Figure 3b. Note that the framework-integrated solver shows a different, faster convergence history than the CODA standalone solver. This can be attributed to a number of influences, two of which are different residual normalizations and GMRES solver implementations applied in both cases. To enforce comparable linear systems for the subsequent scalability study of the linear solvers, the residual normalizations were unified to give the same nonlinear residual for the same restart point.



(a) Different CFL ramp functions lead to different solver trajectories.

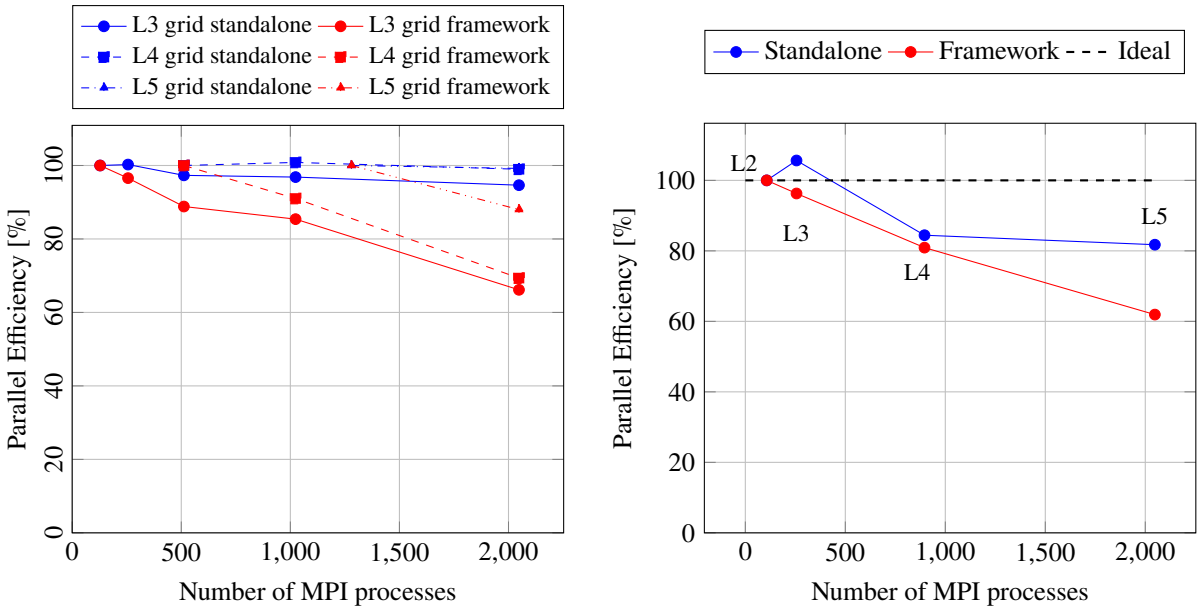
(b) Restart from semi-converged solution with fixed CFL=10^4.

Fig. 3 Comparison of nonlinear solver convergence for standalone vs. framework implementation for DPW-5/CRM, grid level L3. Different solver trajectories were obtained due to different CFL-ramps, residual normalizations and linear solver implementations between framework-based and standalone solvers.

To assess the solver efficiency, we carried out a parallel efficiency study focusing on the CPU-performance of the linear solver part, being the preconditioned flexible GMRES algorithm. 20 GMRES iterations involving 20 Krylov vectors with a defined number of preconditioning iterations (200 for L3, 280 for L4 and 360 for L5) were likewise conducted by the standalone and the framework-driven solver, respectively, on the DLR HPC-cluster CARO. The grid levels L3 to L5 were considered for this study with the number of CPU processes ranging from 128 up to 2048, cf. Figure 4a. In this study, we restricted ourselves to MPI alone in order to evaluate the MPI-parallel performance of the framework approach. Whereas a parallel efficiency of above 95 percent was observed running CODA in standalone mode, the framework approach clearly shows an efficiency decrease with increasing MPI-processes numbers on all mesh levels. For the large computational case (L5), the framework retains a parallel efficiency of almost 65 percent, indicating a fair strong scalability.

The results of a weak scalability study in combination with grid-levels L2–L5 of the DPW-5/CRM wing-body configuration can be seen in Figure 4b. Roughly 565 thousands degrees of freedom were handled by each MPI process. The number of innermost preconditioning element-block-Jacobi iterations in CODA was fixed to 360 for all grids. The standalone approach achieved a parallel efficiency of above 80 percent. The parallel efficiency of the framework approach, on the other hand, decreases constantly almost down to 60 percent for the mesh level L5. Further investigations are required to understand the reason for this behavior. Please note that – due to a specified number of roughly 565 thousand degrees of freedom per MPI process in the weak scalability study – a fully uniform work load distribution over the processors per node could not be guaranteed. This deficiency may have influenced the parallel efficiency measurements for the standalone CODA solver. Moreover, it must be kept in mind that we did not take advantage of CODA's hybrid parallel computation capabilities in this study in order to compare it to the framework approach. Nevertheless, this study demonstrated that the presented MDAO framework approach – consisting of the CFD software

CODA embedded in the FlowSimulator HPC ecosystem in combination with OpenMDAO – is capable of implicitly solving large problems for more than a billion degrees of freedom.



(a) Strong scaling **(b) Weak scaling**

Fig. 4 Parallel efficiency of CODA in standalone and framework for different grids.

According to Figure 5, the wall-clock time ratio of framework over standalone observed with 2048 MPI processes is very similar and clearly below 2 for L4 and L5. The memory requirements of standalone and framework solvers are very similar. For grid level L3, for example, we observed a memory ratio of framework/standalone of approximately 1.3. Note that the FlowSimulator Data Manager (FSDM) was used for both the framework and standalone implementations. The observed memory overhead of the framework approach is very small. However, it has to be kept in mind that in this study, the standalone CFD was not optimized for memory.

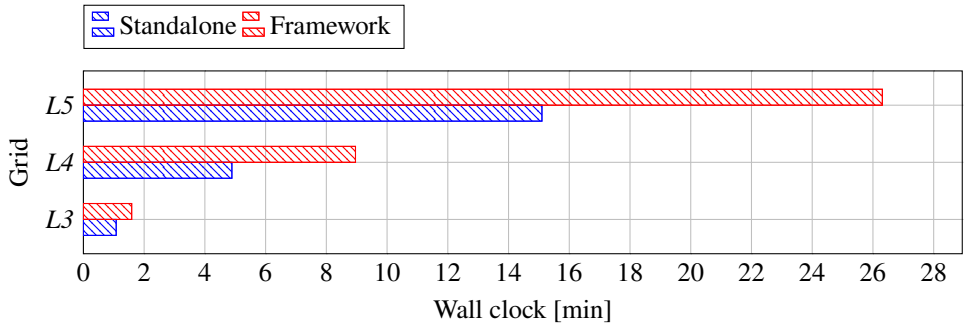


Fig. 5 Comparison of wall clock time for CODA standalone and framework-integrated for 1 pseudo-time step with fixed 20 GMRES iterations accumulated for different grids using 2048 MPI processes (i.e. 16 CPU nodes).

B. Multidisciplinary Solver Study for NASA R4/SDT Configuration

The parallel preconditioning capabilities of monolithic multidisciplinary solution algorithms are in the focus of this numerical study based on a simplified version of the NASA R4/SDT configuration [13], in which only the rotor blade is considered. It is a continuation of a previous study [6] based on the same test case. In this work, attention is drawn on the monolithic framework solver efficiency for different nested solver/preconditioner settings. The solver stack for the

framework-based multidisciplinary preconditioned Newton-GMRES is illustrated in Figure 6. The parts of the algorithm depicted in grey are conducted by the MDAO framework, whereas the disciplinary linear (preconditioning) operations in blue are delegated to the individual FlowSimulator plugins. The plugins are grouped into a *CFD Group* and a *CSM Group* depending on their associations and requirements. The groups can be executed in sequential or parallel. In the sequential group (SG) execution, the plugins in the CSM and the CFD groups are executed one after another, whereas in a parallel group (PG) execution, the two groups are executed concurrently. The parallel groups can be individually associated with different number of MPI resources/processes in a flexible way. The parallel linear block-Jacobi (LBJ)

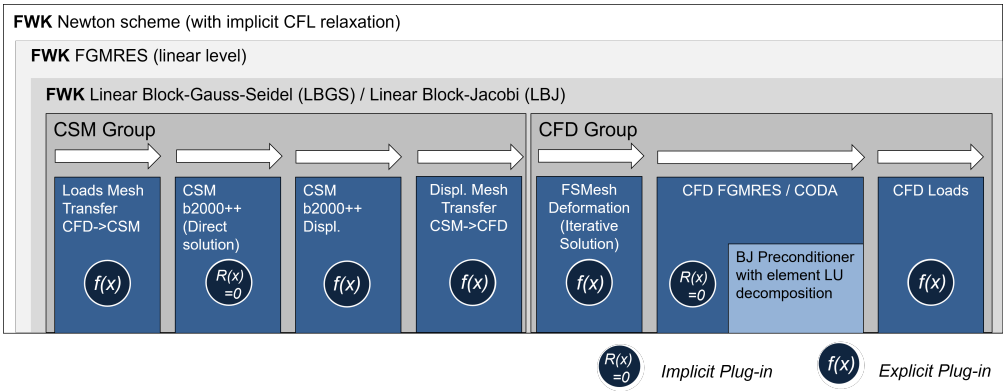


Fig. 6 Sketch of the framework-based multidisciplinary preconditioned Newton–GMRES solver stack. The parts of the algorithm depicted in grey are conducted by the MDAO framework, whereas the disciplinary linear (preconditioning) operations in blue are delegated to the individual FlowSimulator plugins.

approach is expected to provide a parallel-efficiency advantage compared to linear block-Gauss–Seidel (LBGS) type of methods, which require a sequential processing.

The Reynolds number at a blade was $Re = 10^6$ and the inflow Mach number was $Ma = 0.2$. The setup was simulated in fully turbulent flow conditions using the negative extension of the Spalart–Allmaras turbulence model. The numerical CFD solver CODA provides an exact algorithmic differentiation (AD) including the turbulent state equations and periodic boundary conditions. Figure 7 shows the simulation configuration with a rotor blade in blue, stator in yellow and mixing plane in beige and purple. As previously stated, only the rotor blade was considered, neglecting the mixing plane and the stator. The rotor blade was modeled as a flexible structure with a static linear-elastic finite-element model computed in the CSM software b2000++. The root of the rotor blade was mounted/fixed at the bottom. The gap between the rotor blade and the shroud was resolved in the CFD mesh.

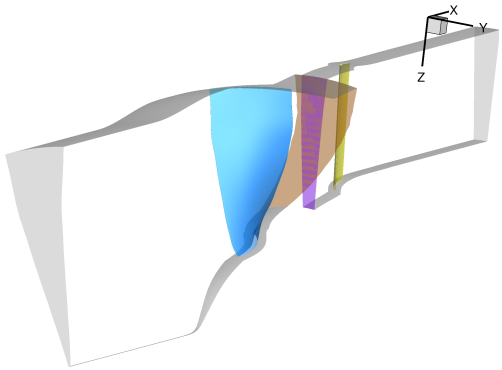


Fig. 7 NASA R4/SDT CFD configuration. The rotor's solid surface is depicted in blue, followed by a mixing plane (shown in beige and purple) and the stator in yellow. Periodic boundary conditions were imposed on the CFD method on both sides of the wedge-shaped domain. The inflow is directed in positive x-direction and the rotor spins counter-clockwise from the point of view of the observer.

The structured CFD mesh contained 3.32×10^5 elements. The solid CSM mesh used for the static elastic model

consisted of 135,930 volume elements and 75,068 surface elements with a linear ansatz and 172,769 nodes. The CFD and CSM meshes for the rotor blade are shown in Figure 8 on the left- and right-hand side, respectively. Unlike in many CFD-CSM surface mesh configurations, the CSM surface mesh is much finer than the CFD surface mesh here. To evaluate the multidisciplinary solver performance, the stiffness of the CSM model was varied, resulting in the flexibility levels referred to as *min*, *mid* and *max*. The relatively small axial deflections of the blade tip divided by the blade height computed for the flexibility levels were 1.14×10^{-5}, 2.02×10^{-3} and 1.24×10^{-2}, respectively.

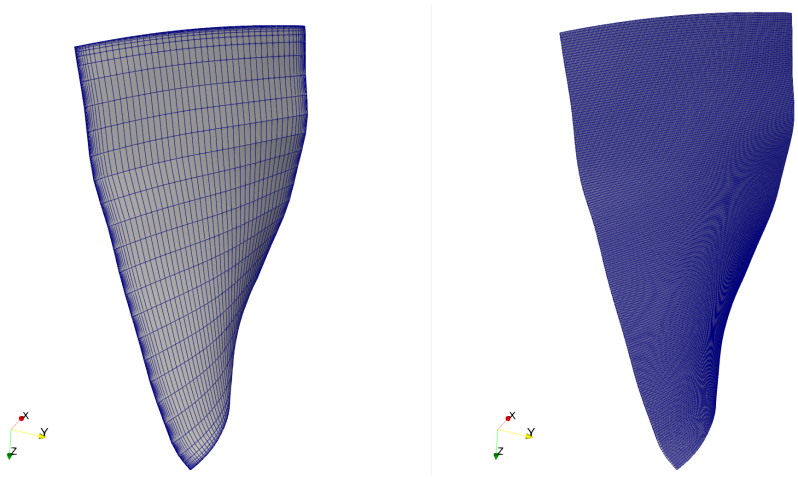


Fig. 8 CFD (left) and CSM (right) surface meshes of NASA R4/SDT rotor blades

In the following, *Hybrid Newton* is defined as a strategy in which each discipline in the multidisciplinary problem is solved once before the first Newton iteration to provide a valid initial guess for the Newton solver. In this initial run, the CFD discipline was solved by the Newton as a sub-solver converging the solution to the relative tolerance of 10^{-4}. Because most of the computational effort in the multidisciplinary algorithm lies in the innermost element-block-Jacobi iterations of the CFD solver, the accumulated work performed in those iterations was chosen to quantify the nonlinear solution effort. Throughout this case study, the innermost element-block-Jacobi iterations were fixed to 50. Furthermore, all multidisciplinary computations were executed using 256 MPI processes, i.e. 2 nodes on CARO HPC cluster. Unless otherwise specified, only one MPI process was associated with the CSM group being a parallel group to the CFD group, both of which were executed concurrently.

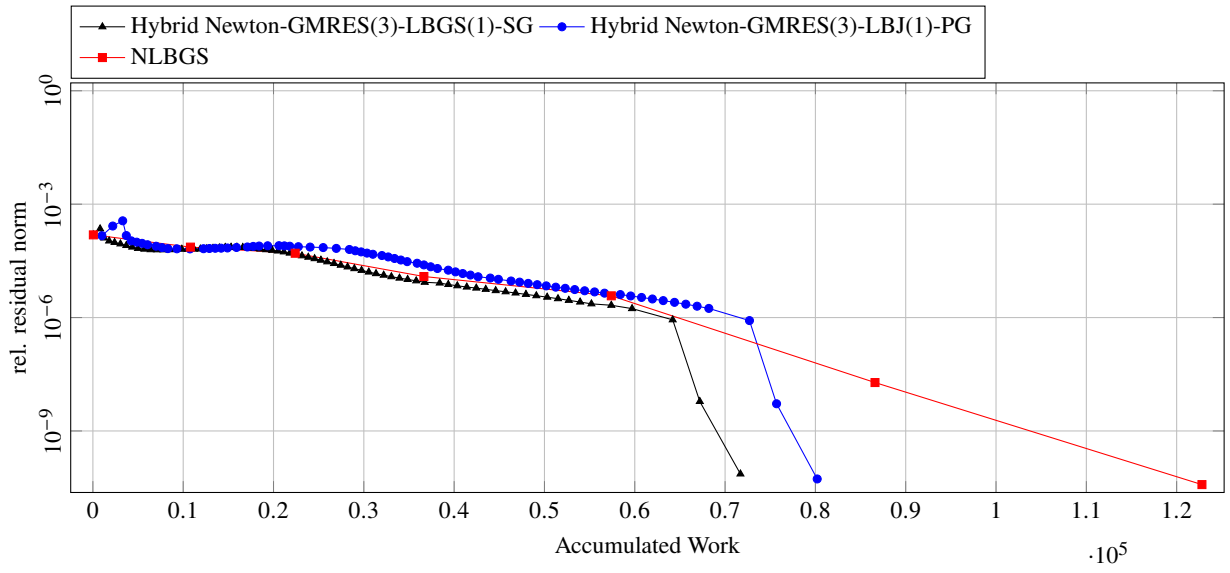


Fig. 9 Convergences of NASA R4/SDT coupled problem for different solver configurations (mid flexibility level).

The nonlinear convergences histories of the nonlinear block-Gauss–Seidel and the Hybrid Newton solver with LBGS organized in a sequential group (LBGS-SG) and LBJ arranged in a parallel group (LBJ-PG) are plotted over the accumulated work in CODA's innermost element block-Jacobi iterations in Figure 9. In both cases, LBGS-SG and LBJ-PG, the framework GMRES solver was iterated for at most 3 steps or until a relative tolerance of 10^{-2} was reached. The preconditioners LBGS-SG and LBJ-PG were applied once in each framework GMRES iteration. The CODA-GMRES solver inside the CFD discipline was iterated for at most 30 or up to the relative tolerance of 10^{-2} was reached. We specify the maximum iteration count behind the solver name, cf. the legend in Figure 9. All three algorithms followed similar convergence paths up to multidisciplinary relative residual norm of 10^{-6}. The trajectories differ after the CFL-ramp has reached infinity in the Newton-type solvers leading to superlinear convergence. As expected, the Newton-GMRES-LBGS-SG algorithm performed best in terms of the accumulated work measure due to the algorithmic superiority of the LBGS algorithm over LBJ. Unlike the Newton solvers, NLBGS converges only linearly.

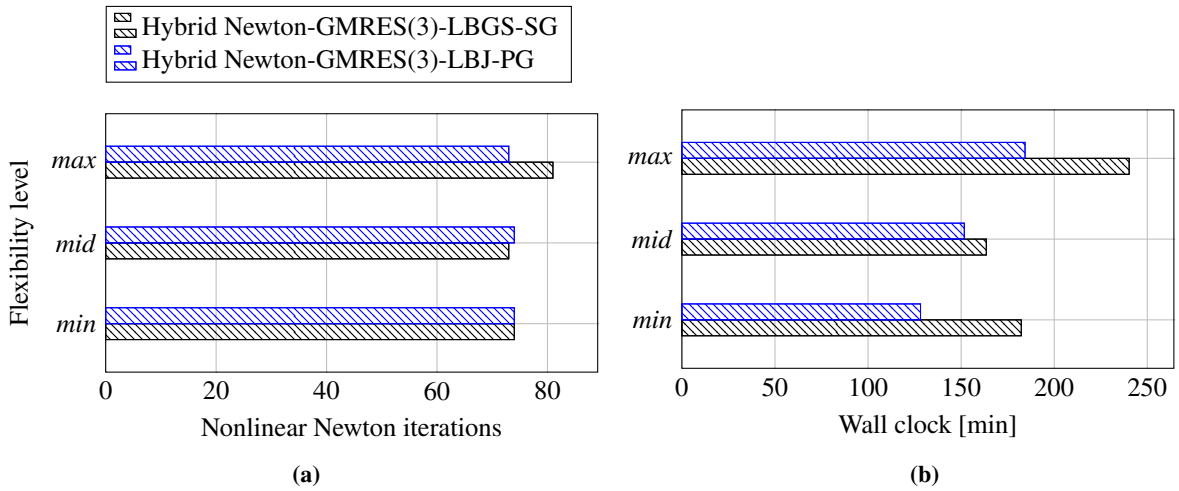


Fig. 10 Nonlinear iteration number (left) and wall-clock time (right) of the nonlinear Hybrid Newton solver for different flexibility levels with FWK-GMRES(3), LBGS(1)/LBJ(1) and CODA-GMRES(30) to a relative multidisciplinary solver tolerance of 10^{-10}. The relative tolerance of framework and CODA-GMRES are 10^{-2}. To stabilize the convergences of the max flexibility level, the maximum iterations of LBGS/LBJ and CODA-GMRES are increased to 2 and 40, respectively.

The number of nonlinear Newton iterations required to converge the nonlinear problem and the corresponding wall-clock times are plotted in Figure 10a and Figure 10b, respectively, for different flexibility levels of the rotor blade. Note that the Newton algorithm is insensitive to the coupling intensities of the problem for different flexibility levels. However, the big jump in wall-clock time for the max flexibility level is due to an increase in CODA GMRES iterations up to 40 and the preconditioner LBGS/LBJ up to 2 for stability reasons. Note, both solvers were solved for the relative tolerance of 10^{-2}. A similar study on the flexibility level was conducted also on the convergences of framework GMRES linear solver. The effort of the linear solver increased with the flexibility of the rotor blade, cf. Figure 11a. As expected, the weaker LBJ preconditioner led to more GMRES solver iterations for the maximum flexibility. However, according to Figure 11b, we observed a huge parallel efficiency advantage for the LBJ approach to solve the linear problem inside the Newton scheme. LBJ outperformed the algorithmic efficiency advantage of LBGS in conjunction with 6 multidisciplinary GMRES sweeps (or a relative tolerance of 10^{-10}) with a maximum of 6 LBGS/LBJ iterations (or a relative residual tolerance of 10^{-5}).

Figure 12a shows that a stronger effort for CFD block-inversion reduces the required number of multidisciplinary GMRES iterations to reach a relative tolerance of 10^{-10}. Reducing the number of multidisciplinary GMRES iterations decreases the wall-clock time, as shown in Figure 12b. Consequently, the overall time to solution is strongly dependent on the multidisciplinary GMRES iteration count, and it can be reduced further by performing additional CODA-GMRES iterations.

In the previous results, we assigned only one out of 256 MPI processes with the CSM solver. According to Figure 13, it turned out that increasing the number of MPI processes further did not increase the computational efficiency. This is

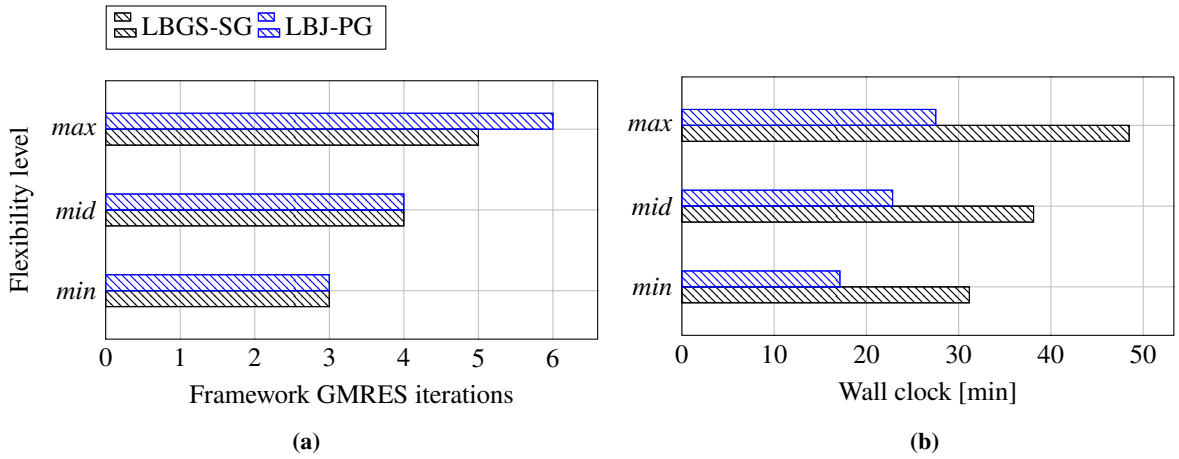


Fig. 11 Number of linear solver iterations (left) and runtime (right) of single Newton step for different flexibility level with fixed 10 CODA-GMRES and 6 framework GMRES sweeps (or relative residual tolerance of 10^{-10}) with maximum of 6 LBGS/LBJ iterations (or relative residual tolerance of 10^{-5})

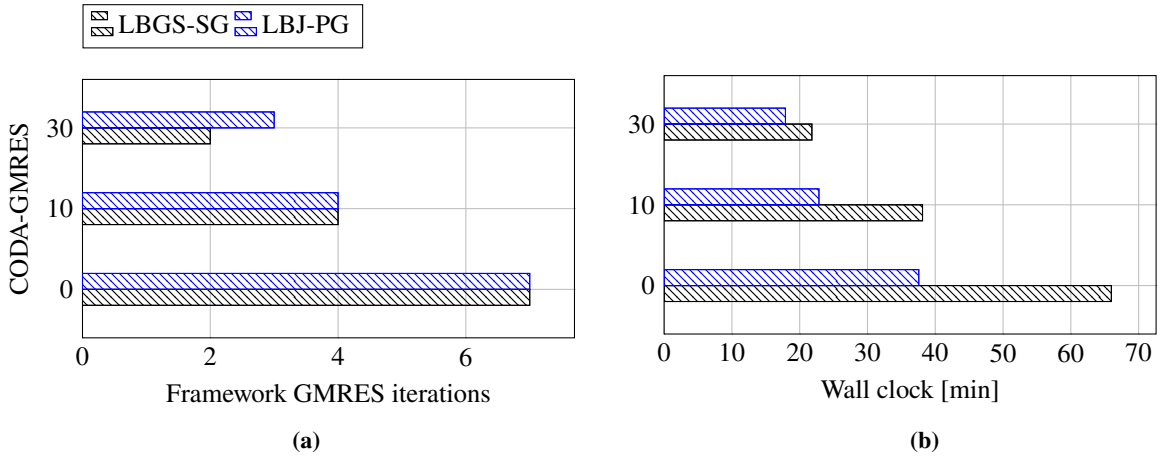


Fig. 12 Number of linear solver iterations (left) and runtime (right) in a single Newton step for different combinations of CODA-GMRES and framework GMRES with linear solver solved for maximum of 6 LBGS/LBJ iterations or up to the relative tolerance of 10^{-5}. Framework GMRES was solved up to the relative tolerance of 10^{-10}.

attributed to the limited domain size of the CSM model.

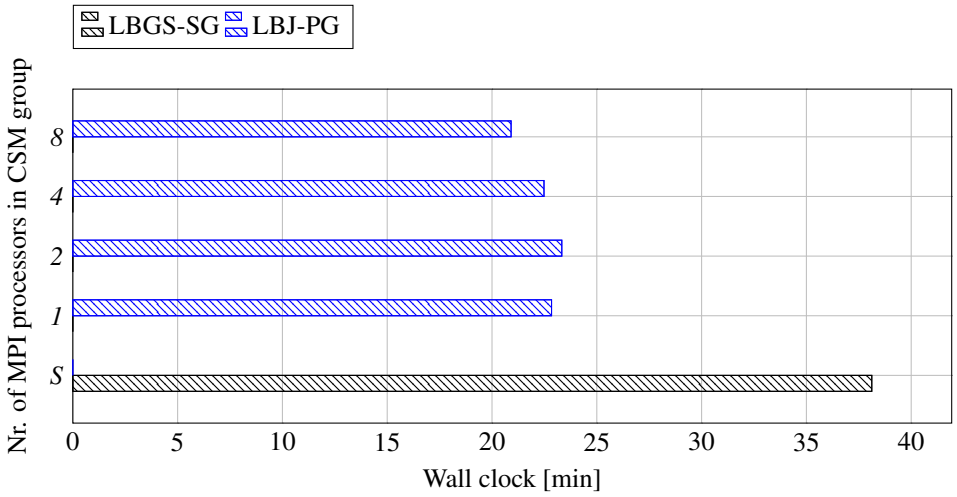


Fig. 13 Runtime of 1 Newton step for different numbers of MPI-Processes in the CSM group with FWK-GMRES(4), LBGS(6)/LBJ(6) and CODA-GMRES(10). Sequential execution is plotted as a reference and denoted by 'S' in the vertical axis.

IV. Conclusion

An MDAO framework approach was presented in this paper with the focus on implicit solution capabilities for large computational cases in parallel execution. Thanks to the modular component integration of implicit simulation components like the CFD solver CODA or the CSM solver b2000++, the framework can carry out monolithic solver operations on the overall-system level of the multidisciplinary problem such as the presented preconditioned Newton-GMRES methods.

Two different computational scenarios were considered in this paper to evaluate nested, multidisciplinary solver stacks implemented in the MDAO-framework: First, we focused on the efficiency of the framework-based solution process, in which the CFD discipline – being by far the largest component – is embedded as a single implicit-component with linear block-preconditioning capabilities. We challenged the MDAO framework approach in combination with a mesh sequence of the DPW-5/CRM configuration L2–L5 resulting in 61 million to 1.2 billion degrees of freedom. In the scalability study, we carried out a prescribed number of preconditioned GMRES sweeps over 5 Newton steps. We observed that the framework is able to solve the large cases and retained a fair scalability of more than 60 percent in both weak and strong scaling. In the second numerical study, we investigated the combination of parallel and algorithmic computational performances for a simplified version of the NASA R4/SDT rotor test case in static aeroelastic conditions. The flexible rotor configuration was computed in combination with a fully-turbulent RANS CFD model and a solid FE/CSM model. Nested multidisciplinary, preconditioned Newton-GMRES solver stacks were evaluated. The framework-based workflows were executed in an MPI-parallel way from end to end. Superlinear convergence was observed for preconditioned Newton–GMRES solvers applied to the nonlinear multidisciplinary problem. It was found that a parallel linear block-Jacobi preconditioning led to a significant parallel efficiency gain outweighing the algorithmic advantage of the linear block-Gauss–Seidel preconditioner.

The solver efficiency studies presented in this paper have successfully demonstrated that the MDAO framework approach can handle large-scale MDAO cases in parallel HPC workflows. The current study was limited to the solution of nonlinear (MDA) problems. We think that the scalability observations presented in this paper are likewise relevant for the solution of large, multidisciplinary *adjoint* problems, which is a key ingredient for efficient multidisciplinary design optimization of aircraft.

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).

References

[1] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, 2019, pp. 1075–1104. https://doi.org/10.1007/s00158-019-02211-z.

[2] Gallard, F., Vanaret, C., Guénot, D., Gachelin, V., Lafage, R., Pauwels, B., Barjhoux, P.-J., and Gazaix, A., "GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation," *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018.

[3] Gray, A. C., and Martins, J. R., "Aerostructural Optimization of the Simple Transonic Wing Using MPhys, ADflow, and TACS," AIAA 2025-2813, 2025.

[4] Yildirim, A., Jacobson, K., and Anibal, J., "MMPhys: a modular multiphysics library for coupled simulation and adjoint derivative computation," *Struct Multidisc Optim*, Vol. 68, 15, 2025, pp. 183 – 192. https://doi.org/10.1007/s00158-024-03900-0.

[5] Backhaus, T., Gottfried, S., Merle, A., Hwang, J. T., and 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.

[6] Masilamani, K., Büchner, A., Ehrmanntraut, S., Gottfried, S., Reimer, L., and Stueck, A., *A Scalable MDAO Framework Approach Enabling Nested Solution Algorithms for Static Aeroelastic Coupling Scenarios*, AIAA AVIATION FORUM AND ASCEND 2024, ???? https://doi.org/10.2514/6.2024-4403, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-4403.

[7] Chauhan, S. S., Hwang, J. T., and Martins, J. R. R. A., "Benchmarking Approaches for the Multidisciplinary Analysis of Complex Systems Using a Taylor Series-Based Scalable Problem," *Advances in Structural and Multidisciplinary Optimization*, edited by A. Schumacher, T. Vietor, S. Fiebig, K.-U. Bletzinger, and K. Maute, Springer International Publishing, Cham, 2018, pp. 98–116.

[8] Chauhan, S. S., Hwang, J. T., and Martins, J. R. R. A., "An automated selection algorithm for nonlinear solvers in MDO," *Structural and Multidisciplinary Optimization*, Vol. 58, 2018, pp. 349–377. https://doi.org/10.1007/s00158-018-2004-5.

[9] Vassberg, J., Dehaan, M. A., Rivers, M., and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," AIAA 2008-6919, 2008.

[10] Kenway, 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 J*, Vol. 52, No. 5, 2014. https://doi.org/10.2514/1.J052255.

[11] Masilamani, K., Büchner, A., Gottfried, S., Reimer, L., Ehrmanntraut, S., and Stueck, A., "A multidisciplinary solver study towards efficient aeroelastic analyses using the FlowSimulator ecosystem," *To appear in CEAS Aeronatical Journal*, 2025. Accepted for publication.

[12] Leicht, T., Vollmer, D., Jägersküpper, J., A. Schwöppe, R. H., Fiedler, J., and Schlauch, T., "DLR-Project DIGITAL-X – Next Generation CFD Solver FLUCS," DLRK, 2016. URL https://elib.dlr.de/111205/.

[13] Envia, E., "Fan Noise Source Diagnostics Test - Vane Unsteady Pressure Results," *8th AIAA/CEAS Aeroacoustics Conference & Exhibit*, 2002. https://doi.org/10.2514/6.2002-2430, URL https://arc.aiaa.org/doi/abs/10.2514/6.2002-2430.

[14] Quaranta, G., Masarati, P., Mantegazza, P., et al., "A conservative mesh-free approach for fluid-structure interface problems," *International conference for coupled problems in science and engineering, Greece*, 2005.

[15] Ehrmanntraut, S., Büchner, A., Gottfried, S., and Stück, A., "Scalable framework integration of coda for a multidisciplinary preconditioned matrix-free newton-krylov method," *STAB/DGLR Symposium*, Springer, 2022, pp. 219–230.

[16] Morrison, J. H., "5th AIAA CFD Drag Prediction Workshop," online, June 2015. URL https://aiaa-dpw.larc.nasa.gov/Workshop5/workshop5.html.

[17] Falgout, R., Malaspinas, J., Heroux, M. J., and Lathrop, J., "PETSc: A Portable, Extensible Toolkit for Scientific Computation," *ACM Transactions on Mathematical Software*, Vol. 18, No. 3, 2002, pp. 123–137. https://doi.org/10.1145/509795.509801.