The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

A CAD-ENABLED MDAO FRAMEWORK APPROACH FOR GRADIENT-BASED AERODYNAMIC SHAPE OPTIMIZATION

T. HAFEMANN[1], M. BANOVIĆ[1], A. BÜCHNER[1], S. EHRMANNTRAUT[1], C. HÖING[1], S. GOTTFRIED[1] AND A. STÜCK[1]

[1] German Aerospace Center (DLR)
Institute of Software Methods for Product Virtualization
Zwickauer Str. 46, 01069 Dresden, Germany
e-mail: thomas.hafemann@dlr.de, arthur.stueck@dlr.de

Key words: CAD integration, gradient-based optimization, CFD, MDAO Frameworks, algorithmic differentiation (AD)

Summary. We aim at a seamless CAD-integration into a framework-based approach for multidisciplinary design analysis and optimization (MDAO) that allows an automated forward and reverse accumulation of AD-based gradients throughout complex high-fidelity workflows. The suggested framework approach relies on the FlowSimulator HPC ecosystem, in which a number of high-fidelity simulation tools, called plug-ins, share large mesh-based coupling data sets in memory via the FlowSimulator Data Manager (FSDM) in workflows that are MPI-parallel from end-to-end. Based on the FlowSimulator infrastructure and a granular plug-in integration, the MDAO framework OpenMDAO [1] was used to drive the CAD-enabled, gradient-based optimization in conjunction with the CFD software CODA. A fully-resolved system representation is generated for the MDAO problem at hand based on a systematic registration of plug-in input/output dependencies. In this work, the algorithmically differentiated CAD kernel OpenCascade Technology (OCCT) was integrated into the FlowSimulator ecosystem to centrally provide a gradient-enabled CAD link for all the simulation plug-ins involved in the problem. The individual FSDM surface mesh objects—that can be (MPI) domain-decomposed—are linked to the corresponding CAD surfaces using a reliable, meta-data enabled mesh-to-CAD association to robustly deal with fine meshes in the context of high Reynolds-number flows. Selected aerodynamic wing configurations in 2D and 3D are considered in this study to verify and demonstrate the AD-enabled sensitivity analysis with the focus on the framework integration of the CAD kernel OCCT in the context of shape optimization.

1 INTRODUCTION

Aircraft design is a complex multidisciplinary problem in which many different coupled analyses are required to meet the targets and constraints regarding aerodynamic efficiency, structural integrity and fuel consumption, to name a few. Most of the disciplines involved require access to the geometry model in different levels of fidelity. In industrial Computer Aided Engineering (CAE), a CAD representation is usually considered as the *true geometry* of the targeted design. A unique and consistent CAD model integrated in a seamless manner into multidisciplinary, simulation-based product design is deemed a key ingredient in the digital design chain.

However, in the context of simulation-based design, there are a number of technical challenges that can impede an advanced level of CAD-integration: numerical (surface) meshes, some of which are highly resolved, need a systematic and robust association with the corresponding CAD patches, the CAD software needs to be ready for HPC in the case of Hi-Fi design analyses, proprietary CAD software – for which the source code is not available – may limit the options for MDAO integration or it may lead to restrictions in supported operating systems and availability of software licenses for large-scale optimizations.

The so-called CAD-free approach does not use the CAD model or its design parameters in the analyses or optimizations so that CAD-integration is reduced to a minimum. Instead, it directly operates on the nodes of the numerical mesh. Since the use of the complete surface mesh can quickly result in $O(10^5)$ degrees of freedom for 3D CFD optimization problems, regularization techniques like Sobolev smoothing or filtering of sensitivity derivatives are required to enforce desirable, smooth shapes. Moreover, the resulting shapes often obtained in huge design spaces can – in the general case – hardly be represented in standard CAD models. Furthermore, it can be cumbersome to retain geometrical constraints or features on the basis of a surface mesh without a link to the CAD patches. Alternatively, modifications can be applied to existing (dead) geometries by superimposing external deformations such as Hicks–Henne bump functions [10] or free-form deformations (FFD) [4]. The prior approach is surface-based whereas the latter is volume-oriented. Such modifications are smooth by construction, however, the parametrization is independent of the original (parametric) geometry generation like in the node-based approach and the handling of constraints and features remains difficult for complex 3D geometries in engineering.

Alternatively, a surrogate of the CAD model can be built, referred to as CAD-ROM approach. The CAD-ROM can help to circumvent a number of CAD-integration issues associated with the use of (proprietary) CAD software in advanced multiphysics workflows, e.g. [2]. However, it has to be kept in mind that the surrogate is one more approximation in the overall system model. The effort for the generation of the CAD-ROM quickly increases with the dimensionality of the design space.

A seamless integration of the true CAD-system into advanced multidisciplinary processes/ frameworks certainly comes at the highest development costs; in some cases, it may be impossible, e.g. due to organizational constraints. However, it has the potential to conduct large-scale simulation-based analyses and optimizations on the basis of the real geometry that was defined by the design engineer. A key ingredient in CAD-enabled, simulation-based processes, which is also discussed in this paper, is a reliable and robust link between the discrete surface mesh representation used in the simulation and the underlying CAD patches of the geometry model. This is required – for instance – when new mesh nodes are to be located on the true geometry in the context of mesh adaptation, or when CAD-based deformations are to be applied to the nodes of the computational mesh in order to perform shape optimizations. To overcome a number of problems associated with the use of general-purpose CAD software packages on HPC systems, Haimes and Dannenhoffer [12] present a simplified but HPC-ready, light-weight CAD kernel based on EGADS. Another approach, presented by [3], is the use of the open-source, Linux-friendly, CAD kernel of OpenCascade Technology (OCCT) for the optimization process.

Optimization processes are usually driven by optimization algorithms that are typically divided into gradient-free and gradient-based. Here, the gradient-based methods are considered due to their efficiency. This is beneficial in shape optimization workflows associated with High-

Fidelity (Hi-Fi) coupled simulations that typically impose large computational costs. Nevertheless, this approach requires gradient computations from each component of the optimization workflow. For instance, the adjoint approach has been considered as state-of-the-art for computing gradients in CFD [11]. The computation of sensitivity derivatives is a challenge in conjunction with CAD-based shape design. When it comes to commercial CAD systems, one typically approximates the gradient information using finite differences [6]. However, gradient inaccuracies of the finite differences caused by approximation errors (perturbation size too large) or cancellation errors (perturbation size too small) can be both a robustness and an efficiency issue in complex optimization workflows, in which a systematic fine-tuning of perturbation sizes can quickly become tedious. On the contrary, to compute the exact gradient information, one can apply algorithmic differentiation (AD) to the CAD sources, provided that these are available. This was successfully demonstrated on the general-purpose, open-source CAD kernel OCCT [3].

This work presents an extension to the multidisciplinary optimization approach presented in [1], systematically integrating an additional CAD component based on the differentiated OCCT kernel. While also the level of framework integration was improved for some framework components like volume mesh deformation [7] and CFD [8], the focus of this paper is on the gradient-enabled integration of the CAD-component OCCT into the process chain in order to systematically provide a direct link-to-CAD for the FlowSimulator HPC ecosystem.

2 INGREDIENTS FOR CAD-BASED ANALYSES AND OPTIMIZATIONS

A number of selected aspects are described below regarding the setup and the use of parametric geometry models by means of the OCCT CAD kernel. The subsequent capabilities and techniques are required to provide a direct mesh-to-CAD link in conjunction with high-fidelity CFD simulations and gradient-based optimizations in the FlowSimulator HPC ecosystem.

2.1 AD-enable CAD kernel OCCT

To achieve a fully CAD-consistent framework integration for MDAO, the parametric modelling, the mesh-to-CAD association and the computation of the sensitivities, are all centrally enabled using the algorithmically differentiated version of the OCCT CAD kernel (v.7.6.2). The CAD model is comprised of several basic topological objects, which can be sorted into faces, edges or vertices. These objects can also be reduced to their geometric representation, which normally does not contain the topological information of the object, but are used for transformations and the calculation of the geometric sensitivities.. Metadata, such as labels, colours, hash-codes, among others, can be associated with each topological element and are stored in the IGES and STEP common CAD formats using the OpenCascade Application Framework (OCAF). The parametrized CAD model consists of algorithms used to create the desired geometries where certain construction parameters are externalized as inputs of the design. Changes in the value of input parameters require a reconstruction of the CAD model resulting in new objects, while keeping the same metadata association, as long as there are no changes in topology. Metadata can be groups, labels, and any kind of colour or characteristic associated with specific CAD objects. In an optimization workflow, where the geometry is recomputed for changes in the design parameters, the metadata is responsible for maintaining the association from newly created CAD objects and mesh nodes.

An important feature of the CAD software is the capability of projecting nodes to it surfaces

and curves. This allows a verification of the position of mesh nodes, as well as their correction based on the latest state of the geometry. The CAD parametric coordinates associated with a face or a curve obtained from a projection are associated with the given node, allowing the direct evaluation of the Cartesian coordinates and associated sensitivities. Based on the original position of the mesh node, the necessary displacement of the surface node for a given change in the geometry can be estimated. Additionally, the AD-enabled OCCT provides the geometric sensitivities for a given parametrization, and is a key component for the gradient-based optimization.

2.2 Mesh-CAD association

A new plugin, FSOCCT, was implemented to serve as an interface for the OCCT kernel and the FlowSimulator HPC framework. This includes a unified Python plugin API giving access to (re)evaluate the CAD model and provide its derivatives and interfaces to the FlowSimulator data manager (FSDM) to share large mesh-data between the plugins in an MPI-distributed way. The interfaces allow the communication between FSDM datasets and OCCT classes and gives access to geometric calculations, such as the evaluation of Cartesian points on the surface of the CAD objects and the assessment of the associated sensitivities, among others. While OCCT itself is not parallelized, the plugin works locally on an MPI-subdomain mesh. One OCCT instance is executed per MPI-rank, allowing a parallel execution of MPI-distributed FlowSimulator workflows, in which MPI-distributed mesh datasets are shared between the plugins in memory via the FSDM.

The correct displacement of mesh nodes based on CAD parameters requires a robust mapping between CAD structures and mesh. This mapping is done as a pre-processing step, when there is only a small deviation between surface mesh nodes and their projection on the respective CAD structures. Three different mapping scenarios are considered, based on how metadata was implemented and passed to the meshing tool.

The first scenario considers a full mapping. Therefore, each surface or CAD object was labelled accordingly, so that the mesh presents the same labelling scheme. As a result, each CAD object has a direct link to the associated mesh cells.

The second scenario considers that structures were grouped according to common boundary conditions or regions. Although there is a direct link from a group of CAD objects to a group of mesh cells, individual objects within the groups still need to be associated to each other. This is accomplished using a search algorithm restricted to the matched groups.

The third scenario considers that no metadata is available to link geometry and mesh objects. For this case, a search algorithm has to be applied for all the existing mesh nodes. This is a common issue when working with benchmark models, where pre-existing geometries and meshes are used.

The search algorithm evaluates the projection distance of surface cell nodes to the CAD objects, and associates them to objects where this distance is minimal. Additionally, the boundaries of the cells, represented by a box, and surface and cell normals are also considered and evaluated at an initial stage, to exclude unrelated CAD objects of the projection verifications. Instead of associating each node separately, where a robust search-and-match algorithm for each node is necessary [5], the nodes are mapped based on the associated mesh cells or given group IDs, thus using the available metadata in the form of boundary conditions or labelled groups. This

reduces the number of necessary verifications to match a given group of cells to the respective CAD object, since not all nodes are verified. It also allows the identification of edges or vertices by referencing nodes that have different group IDs associated depending on the analysed cell. With this approach, specific CAD structures can also be identified by the intersection of parent structures, while also allowing the association of mesh nodes to specific geometrical features.

2.3 Mesh deformation

Displacement applied to the surface mesh need to be propagated into the interior volume mesh, which is particularly challenging for highly-stretched cells to numerically resolve the boundary layer flow at high Reynolds numbers. The mesh deformation is performed by the FlowSimulator plugin FSMeshDeformation [7]. The method uses the linear-elasticity approach, where Dirichlet boundary conditions for the nodes are enforced. Each volumetric cell has an associated stiffness, which is primarily defined by the cell size and its stiffness factor. The boundary conditions consist of the movabilities of the mesh nodes, which are defined as follows: (i) fixed at a prescribed position, (ii) only allowed to move in the direction normal to the surface cell, (iii) only allowed to move parallel to the surface cell (no-normal movement), and (iv) free to move. Mesh cells located in the bulk of the fluid are free to move, while cells on the surface of the geometry are mostly fixed. For the case where nodes at the symmetry plane were not mapped, to avoid folded cells, the movement parallel to the surface is allowed. Both the displacement of the surface nodes and the definition of the boundary conditions are obtained from the CAD. The sensitivities, or mesh velocities, are analytically provided by the method, which solves the linear elasticity problem.

2.4 CFD Solver

The present computations use the CFD software CODA, developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA supports both 2nd-order finite-volume discretizations and higher-order DG schemes on fully unstructured meshes. Special attention is given to HPC capabilities [9]. It was algorithmically differentiated, so that it can provide the necessary aerodynamic sensitivities. In this work, the cell-centered finite volume discretization is used with the Roe-upwinding convection scheme. CODA is a CFD library by design with Python plugin interfaces enabling a seamless and modular MDAO framework integration. Calculations are done using the compressible RANS (Reynolds-averaged Navier-Stokes) equations with the negative Spalart-Allmaras turbulence model. Density and viscosity are computed using the ideal gas law and the Sutherland's law, respectively.

2.5 Multidisciplinary optimization framework

The multidisciplinary analyses and optimizations were achieved by integrating the previously described plugins of the FlowSimulator HPC ecosystem with the framework OpenMDAO for multidisciplinary design analyses and optimizations, cf. [8]. Each component requires the definition of its function and related partial derivatives. Therefore, the derivatives relative to the input parameters have to be provided.

The CAD model is the first discipline in the chain. Its inputs are the positions of control points, whereas its outputs are the displacements of the surface mesh nodes. The derivatives of

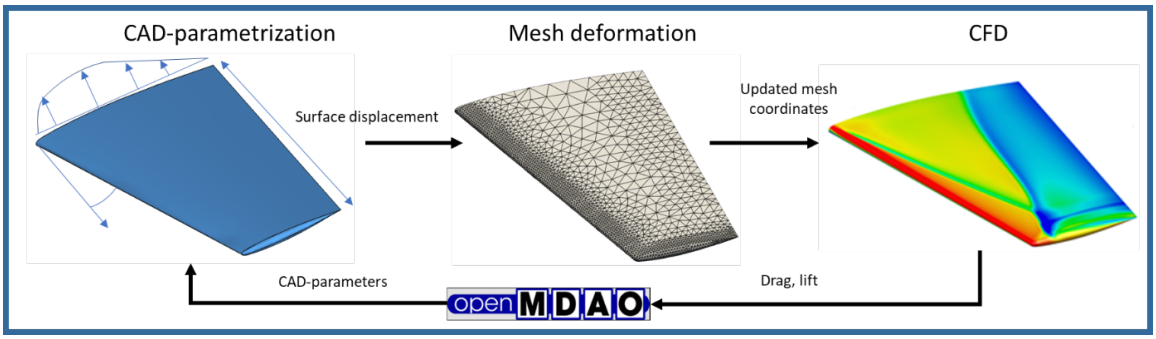

Figure 1: Aerodynamic shape optimization framework with main simulation components provided as FlowSimulator plugins. Inputs and outputs are mentioned component-wise.

the surface mesh nodes w.r.t. the inputs can be computed using the forward or reverse mode of AD. The partial derivatives are provided to the OpenMDAO framework in a matrix-free manner. Furthermore, ordered control points and knots are required to maintain a consistent CAD model. They are implemented as constraints of the CAD subsystem .

The next discipline, the volume mesh deformation, utilizes the displacements as input parameters while it provides the new mesh-node coordinates as output parameters. The control of the displacement behaviour of nodes is given by the defined movabilities, specified in the CAD component.

To conclude the optimization workflow, the CFD solver and the CFD post-processing components are grouped in a discipline. Inputs of the discipline are the coordinates of the mesh, Mach number and angle of attack, while outputs are given by the lift, drag and momentum coefficients.

An overview of all components is shown in Figure 1. Eq. 1 presents the related derivatives (in the forward and reverse mode) of the objective function $\mathbf{J}$ with respect to the geometric parameters $\mathbf{P}$, with $\mathbf{X}$ the node coordinates of the whole mesh and $\mathbf{S}$ the coordinates of the surface nodes

$$\left[\frac{d\mathbf{J}}{d\mathbf{X}}\frac{d\mathbf{X}}{d\mathbf{S}}\frac{d\mathbf{S}}{d\mathbf{P}}d\mathbf{P}\right]^T = d\mathbf{P}^T\left(\frac{d\mathbf{S}}{d\mathbf{P}}\right)^T\left(\frac{d\mathbf{X}}{d\mathbf{S}}\right)^T\left(\frac{d\mathbf{J}}{d\mathbf{X}}\right)^T . \tag{1}$$

3 TOWARDS A CAD-ENABLED AIRFOIL OPTIMIZATION WORKFLOW

3.1 Shape parametrization

A simplified geometry of a 2D profile was parametrized using two B-splines, where the positions of their control points and knots are considered as the design parameters of the optimization. This parametrization differs from more common approaches, such as the NACA 5-digit profiles, where airfoil parameters are used instead. It is important to note that this simplified configuration is defined in order to demonstrate and validate the concept for a gradient-enabled CAD-integration in the MDAO framework approach. For verification purposes, a limited number of control points and knots was used, to maintain a low number of degrees of freedom. The degree of the B-splines is equal to three, while the multiplicities are equal either to four for the extremes (the leading and the trailing edge) or one for the inner knots. The B-spline weights were constant and equal to one.

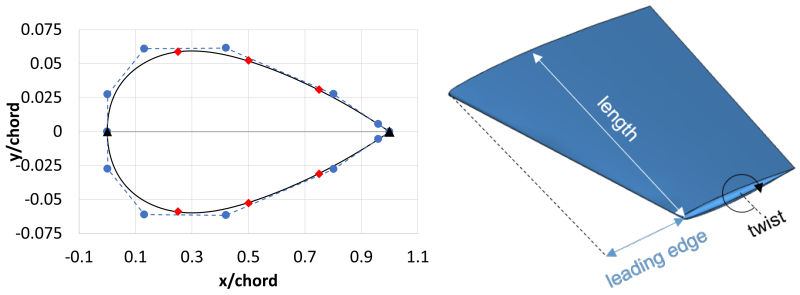

Figure 2: Parametrization of airfoils. Left: 2D NACA0012 profile with fixed (black) and movable control points (blue) and knots (red), right: 3D ONERA wing with taper ratio, leading edge (x, length) and twist.

The baseline configuration of design parameters is set such that the geometry approximates the shape of a NACA0012 profile. To allow a robust design with constant chord length, the profile was represented by two B-splines with fixed extremes. The number of movable control points per B-spline was five, with only three movable knots, as shown in Fig. 2 (left).

The 3D wing design expands the 2D parametrization by adding a taper ratio, a twist angle and the tip position itself, allowing the scaling of the wing tip, the rotation of the tip, and forward and backward swept wing designs, respectively. The parametrization is illustrated in Fig. 2 (right). The verification and optimization run make use of known cases, i.e. the NACA0012 and RAE 2822 airfoil profiles and the 3D ONERA M6 wing. The M6 wing is defined by a taper ratio 0.56, without twist, and a 30° sweep angle, resulting in the leading-edge tip at x_t=(tan(π/6)l, l, 0.0), where l is a half wing span.

With the modelling of the 3D case the following observations were made with regard to robustness. Instead of a single rotation of the profile tip, a lofting operation using both upper-half and lower-half profiles and a blend between them at the horizontal plane was used. The rotation would strongly restrict the possible geometries for the tip and would result in a non-smooth transition from tip to wing. This influences the following: (i) the degrees of freedom of a specific design, and (ii) the robustness of the parametrization, such that non-feasible geometries are avoided. A verification of AD sensitivities is required prior to optimization and it is described in the following section.

3.2 Comparison of CAD sensitivity derivatives

The OpenMDAO framework allows the verification of the implemented components by comparing the provided derivatives with finite differences. A comparison between the sensitivities of the mesh node displacements w.r.t. control points, obtained with the algorithmic differentiation and finite differences, was done for the CAD component. The AD/FD sensitivities are calculated for a chosen number of surface mesh nodes evaluated on the CAD model. In the first test, the sensitivities of the surface mesh nodes were calculated relative to the input parameters individually, with a FD step size of $\epsilon = 10^{-6}$. To allow a single direct comparison, the Euclidean norm of the sensitivities along the control points and knots was calculated. The result for both the algorithmic differentiation and the finite central differences for a NACA0012 wing is shown in Fig. 3 and Fig. 4.

Strong sensitivities are visible at the center of the airfoil, where the sensitivities of more

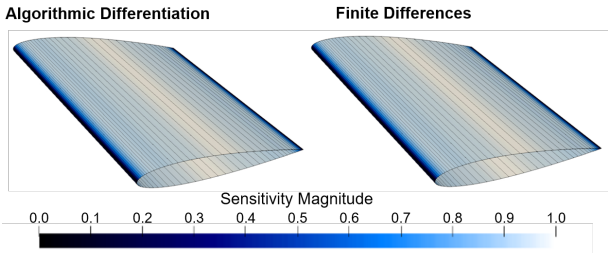

Figure 3: Euclidean norm of the sensitivities relative to the control points of the geometry (NACA0012). Left: AD sensitivity, right: central finite differences for a step size of $\epsilon = 10^{-6}$.

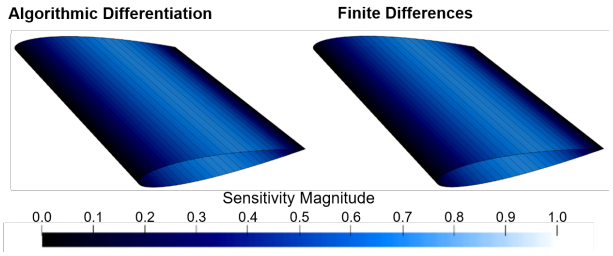

Figure 4: Euclidean norm of the sensitivities relative to the knots of the geometry (NACA0012). Left: AD sensitivity, right: central finite differences for a step size of $\epsilon = 10^{-6}$.

control points overlap. The lowest sensitivities are located at leading and trailing edges, due to fixed control points. Similarly, the sensitivities relative to the knots also present their maximum at the center. The magnitude, however, is smaller than as seen for the control points. This can be traced back to the low number of knots when compared to control points, and the fact that each control point can be moved both in x and y directions, while the knot is a function of the parametric coordinate.

The same analysis was also performed for the 3D geometry of the ONERA M6 wing (Fig. 5). The strongest sensitivities are at the tip, which combines the influence of the profile, already seen for the NACA0012 wing, the scaling of the profile through the taper ratio, the positioning of the leading edge of the tip and the twist angle.

3.3 Finite-difference convergence study for CAD sensitivity derivatives

OpenMDAO provides the necessary functionality to verify the implemented derivatives by calculating the partial and total derivatives of the used components. The AD sensitivities obtained from unitary perturbation vectors are used to construct the Jacobian matrix for both forward and reverse modes. The Jacobian matrix of the implemented method, in this case the AD, is then compared against the Jacobian matrix obtained with finite differences (FD). For the convergence study, the Euclidean norm of the difference between AD and FD Jacobian matrices was calculated for all directions as given by Eq. 2. The same check was repeated for different FD step sizes ϵ. In the forward AD, the sensitivities for each design parameter is calculated separately. Similarly, the verification of the reverse, or adjoint, approach calculates the influence of a perturbation of each sensitivity on the design parameters.

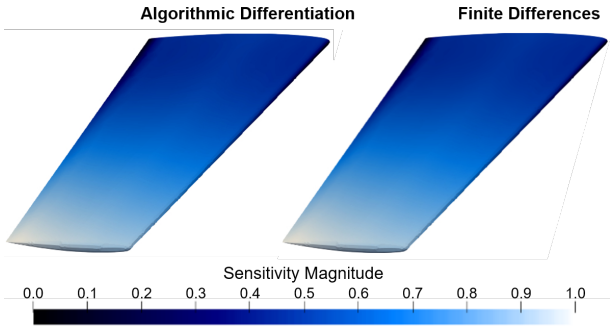

Figure 5: Euclidean norm of the sensitivities relative to the input parameters of the geometry (ONERA M6). Left: AD sensitivity, right: central finite differences for a step size of $\epsilon = 10^{-6}$.

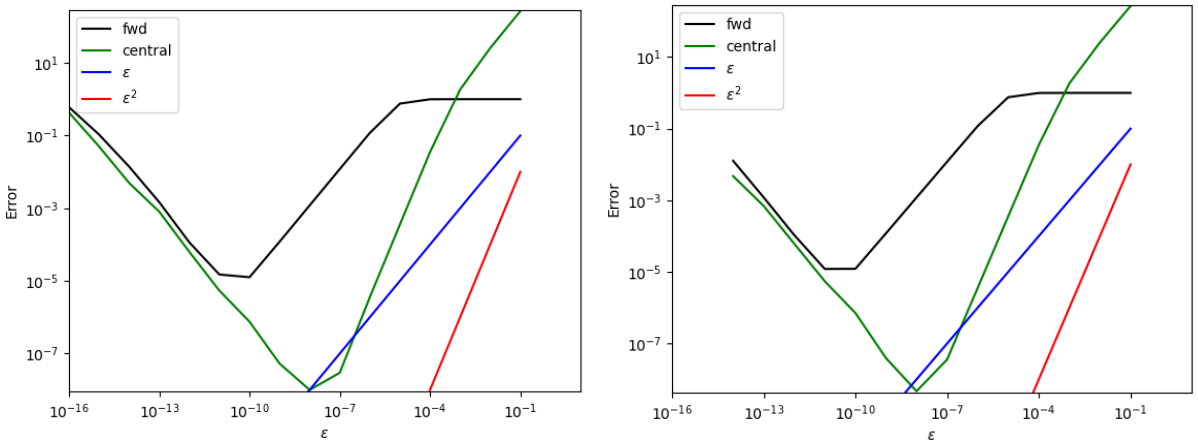

Figure 6: Convergence of CAD component with norm of displacements when comparing AD and FD approaches (NACA0012). Left: forward AD, right: reverse AD (adjoint).

$$Error = \left|\left|\left|\left|\frac{d\Delta\mathbf{X}}{d\mathbf{P}}\right|\right|\right|\right|_{AD}^{T} - \left|\left|\left|\frac{d\Delta\mathbf{X}}{d\mathbf{P}}\right|\right|\right|_{FD}^{T}\Bigg|\Bigg| \tag{2}$$

Figure 6 presents the resulting convergence for the total derivative of the CAD component, for both the forward and reversed AD approaches, for the NACA0012 airfoil. The step size ϵ and ϵ^2 are plotted along the results to illustrate first and second order convergence. Step sizes are kept below $\epsilon = 10^{-1}$ to avoid construction errors within the CAD tool. According to Figure 6, the central FD presents a second-order convergence down to a step size of $\epsilon = 10^{-7}$, while the forward FD is only first order, reaching the minimum error at a step size of $\epsilon = 10^{-10}$. When the minimum error is achieved, further decreasing the step size increases the error. This behaviour identifies when machine precision was reached, since the calculated difference caused by the perturbation is of the order of the error multiplied by the step size for the minimum error, e.g. $10^{-8} \cdot 10^{-8} = 10^{-16}$. Similar results can be observed both for the forward AD and the reverse AD.

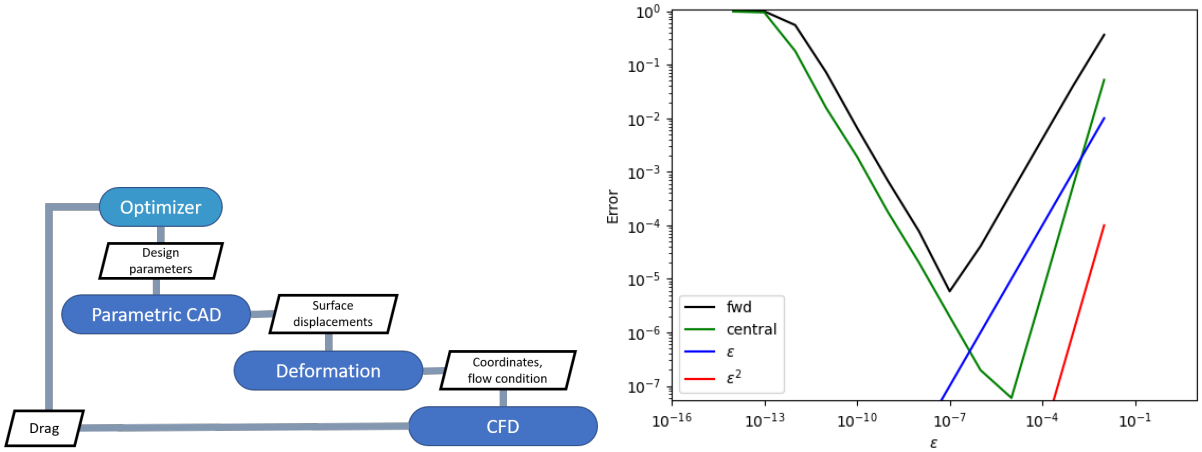

Figure 7: Convergence of the complete optimization system with CAD, deformation and CFD disciplines (NACA0012). (Left) XDSM diagramm, (right) calculated error when comparing forward AD and FD approaches.

Figure 7 shows the corresponding finite-difference convergence study considering the forward AD for the whole system (left); with CAD, deformation and CFD components in the chain, a similar convergence behavior is obtained as in the disciplinary CAD-case. It is important to note that the same level of finite-difference convergence against the exact AD (right) is retained for the full dependency chain.

3.4 Feasibility study towards a CAD-enabled optimization workflow

A first optimization test run was carried out in conjunction with an RAE 2822 airfoil at $Ma = 0.734$, $Re = 6.5 \cdot 10^6$, angle of attack $\alpha = 2.6°$ in order to demonstrate the feasibility of the CAD-integrated optimization procedure to minimize the drag without constraining a minimum lift. The optimization driver used was SciPy's SLSQP algorithm. The test case showed a considerable reduction of the drag coefficient, C_d, of 26%, from 0.0172 to 0.0127, with a reduction of the lift coefficient, C_l, of 11%, from 0.7406 to 0.6584. This result was obtained after the optimization was stopped, after 8 iterations, before the mesh deformation resulted in negative volumes due to a large displacement of the nodes at the leading and trailing edges. The variation in drag, in shape and in the pressure coefficient are summarized in Figure 8. The shown optimization history presents the variation of the drag and lift coefficients . As one can notice, a large reduction of the drag is accompanied by a decrease of the lift. The last four iterations show a smaller variation of the drag, with some recovery of the lift. This was not a result of constraining the lift, but only different attempts of the optimizer to find a new solution. In the depicted profile, with unscaled axes, to improve the visibility of the changes. The thickest point at the lower side was slightly shifted backwards, while the upper side presents a decrease in thickness. The pressure coefficient distribution was calculated at the mesh cell centers. For this profile it is common knowledge, that for a considerable reduction of the drag the shock, visible as a bump on the upper side, has to be reduced. The optimized profile slightly reduces the bump, thus confirming this assumption. Due to the existing constraints and the number of control points the defined problem is too rigid to allow a larger reduction of the bump without

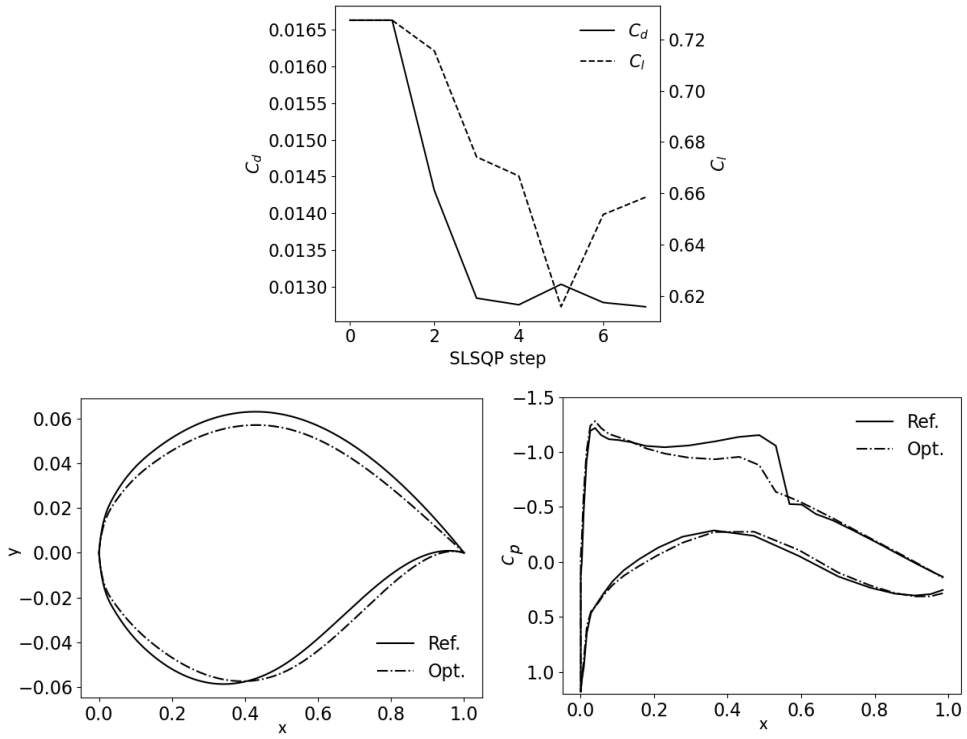

Figure 8: Preliminary optimization of RAE 2822 airfoil. (Top) optimization history, (bottom-left) comparison between original and optimized profile, (bottom-right) pressure coefficient along the chord length.

forcing the mesh deformation component beyond the linear regime.

4 CONCLUSIONS AND OUTLOOK

In this work, we presented an integration of the AD-enabled OCCT kernel into an optimization framework consisting of the FlowSimulator HPC environment and OpenMDAO for CAD-based shape optimization. The implementation and integration of the individual disciplines were verified component-wise with a focus on the CAD component. A systematic verification study was carried out in order to verify the AD-computed sensitivity derivatives. Simple 2D and 3D configurations were used to test the framework approach with a small number of parameters.

In future work, more complex and industrial MDAO cases will be considered using the CAD-enabled framework approach. Further disciplines such as a general-purpose finite-element CSM method will be considered together with different geometrical constraints formulated via the CAD in both implicit and explicit ways. The differentiated OCCT is capable of providing such derivatives for the CAD-based constraints. Another goal is the introduction of the differentiated PythonOCC in the framework approach. This is expected to significantly increase the flexibility of the Python-based MDAO framework since it allows a direct modelling of the geometry with access to exact sensitivity derivatives.

REFERENCES

[1] Backhaus, T., Gottfried, S., Merle, A., Hwang, J. T. and Stück, A. (2021). Modularization of High-Fidelity Static Aeroelastic MDO Enabling a Framework-based Optimization Approach for HPC. AIAA Scitech 2021 Forum. AIAA 2021-1236.

[2] Merle, A., Bekemeyer, P., Görtz, S., Keye, S. and Reimer, L. (2023). Adjoint high-dimensional aircraft shape optimization using a CAD-ROM parameterization. CEAS Aeronaut. J. 14, 729–738.

[3] Banović, M., Mykhaskiv, O., Auriemma, S., Walther, A., Legrand, H., Müller, J.D. (2018). Algorithmic differentiation of the Open CASCADE technology CAD kernel and its coupling with an adjoint CFD solver. Optim. Methods Softw. 33(4–6), 813–828.

[4] Kenway, G., Kennedy, G. and Martins, J. R. R. A. (2010). A CAD-Free Approach to High-Fidelity Aerostructural Optimization. 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference. AIAA 2010-9231.

[5] Zhao Z., Zhang Y., He L., Chang X., Zhang L. (2020). A large-scale parallel hybrid grid generation technique for realistic complex geometry. Int. J. Numer. Meth. Fluids 92, 1235–1255.

[6] Agarwal, D., Marques, S., Robinson, T., Armstrong, C. and Hewitt, P. (2017). Aerodynamic Shape Optimization Using Feature based CAD Systems and Adjoint Methods. 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.

[7] Cristofaro, M., Fenske, J. A., Huismann, I., Rempke, A. and Reimer, L. (2023). Accelerating the FlowSimulator: Improvements in FSI simulations for the HPC exploitation at industrial level. COUPLED 2023.

[8] Ehrmanntraut, S., Büchner, A., Gottfried, S. and Stück, A. (2024). Scalable Framework Integration of CODA for a Multidisciplinary Preconditioned Matrix-Free Newton-Krylov Method. STAB/DGLR Symposium 2022.

[9] Jaegerskuepper, J. and Vollmer, D. (2022). On highly scalable 2-level-parallel unstructured CFD. ECCOMAS 2022.

[10] Hicks, R. M., and Henne, P. A (1978). Wing design by numerical optimization. Journal of Aircraft, 15(7), 407–412.

[11] Giles, M.B., Duta, M.C., Müller, J.D., Pierce, N.A. (2003) Algorithm developments for discrete adjoint methods. AIAA journal, 41(2), 198–205.

[12] Robert Haimes and John F. Dannenhoffer (2018). EGADSlite: A Lightweight Geometry Kernel for HPC. AIAA 2018-1401. AIAA Aerospace Sciences Meeting.