ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

TOWARDS MULTIDISCIPLINARY ADJOINT OPTIMIZATION OF TURBOMACHINERY COMPONENTS

Marc Schwalbach[1] **and Tom Verstraete**[2]

[1] von Karman Institute for Fluid Dynamics
72, Chaussee de Waterloo, B1640 Rhode Saint Genese, Belgium
e-mail: marc.schwalbach@vki.ac.be

[2] Queen Mary University Of London
E1 4NS Mile End Road, London, UK
e-mail: t.verstraete@qmul.ac.uk

Keywords: adjoints, algorithmic differentiation, turbomachinery, CAD, multidisciplinary optimization, fluid-structure interaction

Abstract. *The current state-of-the-art adjoint design optimizations for turbomachinery components focus solely on aerodynamic cost functions and constraints, yet disregard structural feasibility during the optimization procedure. This paper presents the first steps taken towards including structural constraints in a multidisiplinary adjoint optimization design chain for turbomachinery components. Particularly in turbomachinery, deformations arise due to centrifugal and pressure load during running conditions, which lead to a coupled fluid-structure interaction problem. While most optimization methods treat the fluid and structure domains separately in a single-disciplinary fashion, we seek to directly include the coupled fluid-structure interaction within the adjoint optimization. To this end, a cold-to-hot transformation tool that deforms a CAD geometry based on FEM displacements is implemented and differentiated using adjoint algorithmic differentiation to compute the required transformation sensitivities.*

1 INTRODUCTION

Current adjoint optimization methods for turbomachinery components focus mainly on aerodynamic cost functions and constraints , e.g. [3], [10], [18]. The structural feasibility of the resulting optimized shape is usually tested a posteriori by conducting a stress analysis. As a result, an aerodynamically optimized shape may exceed defined stress tolerance levels and a new shape will have to be designed. This can lead to several costly design iterations. Multidisciplinary adjoint optimization seeks to take structural constraints into consideration during the optimization process by coupling the fluid and structural disciplines. This will avoid unnecessary design iterations and directly compute an aerodynamically optimized shape, which is also structurally feasible.

Turbomachinery components pose an additional challenge due to the deformations which occur at running conditions. These are caused by rotation-induced centrifugal forces, gas pressure loads, as well as thermal expansions due to temperature changes. The deformation process is known as the *cold-to-hot* transformation [5], which can be seen as a fluid-structure interaction problem. The *cold* state defines the at-rest state at which a component is manufactured. The *hot* state defines the deformed state of a component during running conditions. Typically, these deformations are not taken into account during the optimization process itself, but rather a shape is optimized in its hot state and transformed a posteriori to its cold shape for manufacturing. In this case, the parametrized CAD geometry used in the optimization, also named the master CAD geometry, is defined in the hot state [6]. However, this method has the downside that the master CAD geometry used in the optimization was generated for one specific design point. Different design points, each with their own centrifugal and pressure loads, would result in different deformed hot geometries. Nevertheless, often the same master CAD geometry is used to compute other design points. Additionally, the cold geometry that is generated after the optimization may no longer fulfill manufacturing constraints, e.g. flank milling of radial machines requires ruled surfaces.

As opposed to a hot state optimization and an a posteriori hot-to-cold transformation, we propose setting the cold state CAD parameters as the design parameters for the optimization with the cold-to-hot deformation being computed during the simulation. As a result, the optimization would directly compute a manufacturable cold state geometry of a component optimized for a chosen design point. Since the final goal is an adjoint multidisiplinary optimization, the differentiation of an entire cold-to-hot chain is required. In this paper, we present first steps towards this alternative by implementing an adjoint version of a cold-to-hot transformation tool to compute the required sensitivities.

First, we will motivate our goal of adjoint multidisciplinary optimization in section 2. Sections 3 and 4 outline the cold-to-hot algorithm and its differentiation with AD, respectively. The results are presented in section 5 with concluding remarks in section 6.

2 ADJOINT MULTIDISCIPLINARY OPTIMIZATION

This section introduces the motivation of adjoint multidisciplinary optimization in turbomachinery. With the goal of optimizing the shape of a turbomachinery component, a metric referred to as the *cost function* $J(\boldsymbol{x}) \in \mathbb{R}^c$, e.g. efficiency, has to be optimized, i.e. maximized. The parameters which can be varied to minimize the cost function are referred to as the *design parameters* $\boldsymbol{x} \in \mathbb{R}^d$. The cost function is minimized using an optimization method, which mainly falls into one of the two categories of *gradient-free* or *gradient-based* optimization methods.

Gradient-free optimization methods can be used to compute improved designs using only cost function evaluations, but finding the true optimal design is not guaranteed. Additionally, converging towards an optimum can require a large number of evaluations and the cost of the optimization is dependent on the number of design parameters d. Gradient-based optimization methods, on the other hand, use sensitivity information to compute a minimum with less iterations than gradient-free methods. However, these methods compute a local minimum and cannot guarantee that the computed minimum is indeed the global minimum, unless the optimization problem is convex. Furthermore, the sensitivities of the cost function with respect to the design variables

$$\frac{\partial J}{\partial \boldsymbol{x}} \in \mathbb{R}^{c \times d} \tag{1}$$

have to be computed. Typical approximation methods, such as finite differences (FD), compute the gradient (1) at a cost proportional to the number of design parameters d, which can lead to high computational costs.

Using the *adjoint* approach, first introduced by Pironneau [15] and later in the application of aerodynamic design optimization by Jameson [8], [16], the gradient (1) can be computed at a cost proportional to the size c of the cost function J. In the context of aerospace engineering, one is typically interested in few parameters such as efficiency, mass flow rate, pressure ratio, and maximum stresses. Thus, the size of the cost function is usually much smaller than the size of the design space, such that $c \ll d$. Thus, the adjoint method offers a significant performance advantage for gradient-based optimization methods over gradient-free methods, given a smooth design space. Notably, the cost of computing the gradient is independent of the number of design variables d, allowing a much greater design space. The adjoint approach requires the evaluation of the *adjoint model*

$$\boldsymbol{x}_{(1)} = \frac{\partial J(\boldsymbol{x})}{\partial \boldsymbol{x}}^T J_{(1)} \tag{2}$$

of the cost function. First deriving a continuous form of the adjoint model (2), then discretizing it, is referred to as the *continuous adjoint* [9] approach. However, deriving a continuous adjoint model of a complex system of equations can be extremely difficult and error prone. Alternatively, the *discrete adjoint* [11] approach discretizes the system of equations first, then derives an adjoint model of the discretized system of equations. By using the code transformation method *algorithmic differentiation* (AD) [7], [13], also known as *automatic differentiation*, an adjoint model of a computer program code can be easily generated. In total, this makes the discrete adjoint optimization method an efficient and precise alternative to gradient-free optimization methods.

Gradient-free optimization methods have been applied in the context of CAD-based multidisciplinary optimization of turbomachinery components e.g. by [12], [17]. In this project, we seek to apply the discrete adjoint approach. We especially want to focus on the fluid-structure interaction (FSI) problem portrayed in the introduction (section 1). An essential aspect of FSI within a CAD-based optimization is the interface between the fluid, solid, and CAD disciplines. The cold-to-hot transformation (section 3) provides one such interface. Within an adjoint optimization, a discrete adjoint model of this interface is required, which is achieved by differentiating the cold-to-hot transformation using AD (section 4).

3 COLD-TO-HOT TRANSFORMATION

The cold-to-hot transformation implemented in this project is based on the method detailed in [5]. Given an FEM mesh that is generated from a cold state CAD geometry, an FEM linear elastic computation is carried out to compute the displacements generated by given centrifugal forces. For now, we are focusing on deformations caused only by centrifugal forces. By perturbing the B-spline surface control points, the computed FEM displacements are matched and the CAD geometry is transformed to its hot state. In this section, the key steps of the transformation will be outlined. The interested reader is referred to [5] for a more detailed explanation of the algorithm and to [4], [14] for an in-depth discussion of computational geometry.

A B-spline surface

$$S(u,v) = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} N_i^p(u) N_j^q(v) \boldsymbol{p}_{i,j}^0, \tag{3}$$

is given with basis functions $N_i^p(u)$ and $N_j^q(v)$ of order p and q, respectively, foot points $0 \leq u \leq 1$ and $0 \leq v \leq 1$, and control points $\boldsymbol{p}_{i,j}^0 \in \mathbb{R}^3$. The position of FEM nodes $\boldsymbol{d}_k^0 \in \mathbb{R}^3$ and their respective displacements $\Delta \boldsymbol{d}_k \in \mathbb{R}^3$ are given as well. $k = 0, ..., s-1$, where s defines the number of FEM nodes on the surface.

3.1 FEM node projection

The FEM nodes $\boldsymbol{d}_k^0$ are projected onto surface foot points (u_k, v_k) by solving a point inversion problem, which can be solved using a suitable algorithm from [14]. As depicted in figure 1, the displacements $\Delta \boldsymbol{d}_k$ are imposed onto the foot points (u_k, v_k), to compute the displaced node points

$$\boldsymbol{d}_k = S(u_k, v_k) + \Delta \boldsymbol{d}_k. \tag{4}$$

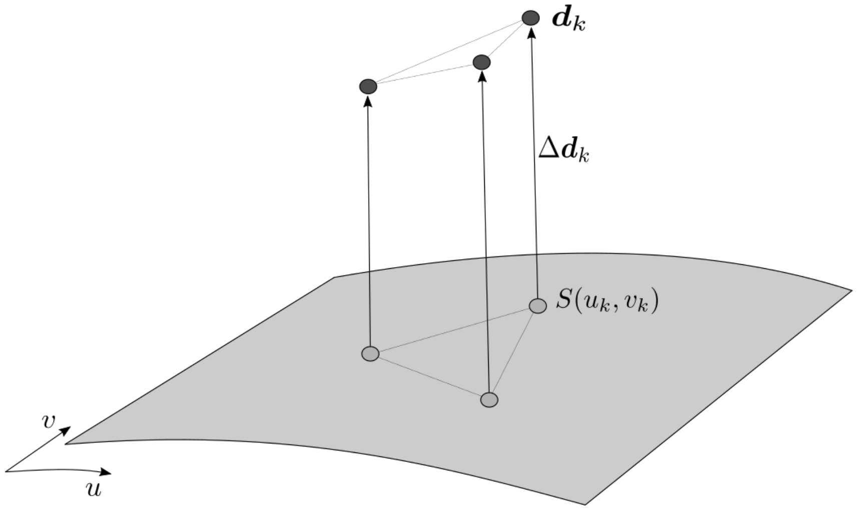


Figure 1: Displacing surface point $S(u_k, v_k)$ with FEM displacement $\Delta \boldsymbol{d}_k$

Using the assumption that the displaced node points $\boldsymbol{d}_k$ must pass through the deformed B-spline surface $\widehat{S}(u_k, v_k)$, the following system of equations can be derived:

$$\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} N_i^p(u_k)N_j^q(v_k)\Delta\boldsymbol{p}_{i,j} = \Delta\boldsymbol{d}_k, \quad k = 0, ..., s-1, \tag{5}$$

where $\Delta\boldsymbol{p}_{i,j}$ represents the change to control point $\boldsymbol{p}_{i,j}$. Typically, this system is overdetermined since the number of FEM nodes s is usually greater than the number of control points $n \cdot m$. This system can be solved using a least-squares method.

3.2 Solve for outer control point displacements

While the solution of the system (5) can deliver the required control point displacements $\Delta\boldsymbol{p}_{i,j}$ for a single surface $\widehat{S}(u, v)$, multiple surfaces with adjacent edges are mostly involved in a complex geometry. An explicit treatment of the surface edges is required to avoid different solutions of $\Delta\boldsymbol{p}_{i,j}$ along adjacent edges, which can lead to undesirable kinks. The control points along the surface edges are thus solved first by treating the edges as B-spline curves. This results in the overdetermined system of equations

$$\sum_{i=0}^{n-1} N_i^p(u_k)\Delta\boldsymbol{p}_i = \Delta\boldsymbol{d}_k, \quad k \in \mathbf{K}_{edge}, \tag{6}$$

with $\mathbf{K}_{edge}$ defining the set of FEM nodes along the edge.

3.3 Solve for inner control point displacements

Once the control point displacements along the edges $\Delta\boldsymbol{p}_{i,0}, \Delta\boldsymbol{p}_{i,m-1}, \Delta\boldsymbol{p}_{0,j}, \Delta\boldsymbol{p}_{n-1,j}$ are determined, the known values of (5) can be moved to the right-hand side to solve for the remaining inner control point displacements

$$\begin{aligned}\sum_{i=1}^{n-2}\sum_{j=1}^{m-2} N_i^p(u_k)N_j^q(v_k)\Delta\boldsymbol{p}_{i,j} = \Delta\boldsymbol{d}_k &- \sum_{i=0}^{n-1} N_i^p(u_k)[N_0^q(v_k)\Delta\boldsymbol{p}_{i,0} + N_{m-1}^q(v_k)\Delta\boldsymbol{p}_{i,m-1}] \\ &- \sum_{j=1}^{m-2} N_j^q(v_k)[N_0^q(u_k)\Delta\boldsymbol{p}_{0,j} + N_{n-1}^p(u_k)\Delta\boldsymbol{p}_{n-1,j}], \\ k \in \mathbf{K}_{inner},\end{aligned} \tag{7}$$

with $\mathbf{K}_{inner}$ defining the set of remaining inner FEM nodes. Afterwards, the control points can be updated by

$$\boldsymbol{p} = \boldsymbol{p}^0 + \Delta\boldsymbol{p} \tag{8}$$

and continuity correction methods can be used to ensure continuity at the common edges of deformed surfaces. Results of the implemented algorithm are presented in section 5.1.

4 SENSITIVITY COMPUTATION

For the computation of the sensitivities of the structural quantity of interest, e.g. the maximum von Mises stress $\sigma_{max} \in \mathbb{R}$, with respect to the design parameters $\boldsymbol{\alpha} \in \mathbb{R}^d$, the sensitivities of an entire chain of operations have to be considered. While the inclusion of pressure loads from the CFD solution is required to fully couple the fluid and structural disciplines, for now we regard only centrifugal loads. Consider figure 2 for the chain of computations required to compute the structural quantity σ_{max}.

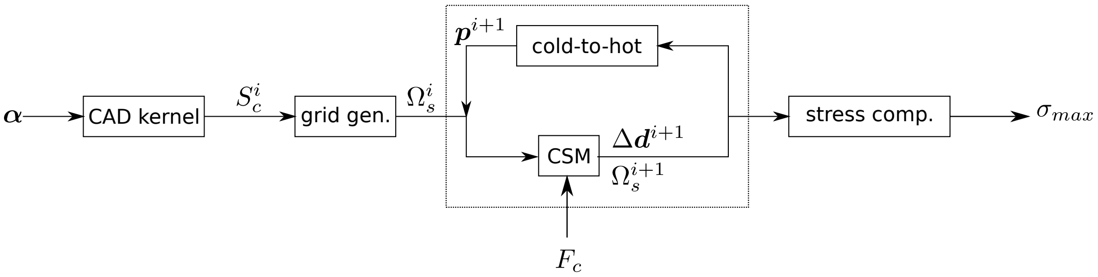


Figure 2: cold-to-hot flow chart

One starts off with design parameters $\boldsymbol{\alpha}$, which can define initial control points of the cold geometry, as well as application specific parameters such as inlet- and outlet radius. $\boldsymbol{\alpha}$ is then plugged into the CAD kernel which generates a cold B-spline surface S_c^i, which is used to generate the initial solid mesh Ω_s^i. The solid mesh, along with the known centrifugal forces F_c, is then used in the computational structural mechanics (CSM) solver to compute the FEM displacements $\Delta \boldsymbol{d}^{i+1}$. The solution of the CSM solver automatically gives us an updated solid mesh Ω_s^{i+1}. The FEM displacements $\Delta \boldsymbol{d}^{i+1}$ are then plugged into the cold-to-hot transformation algorithm to compute the updated control points $\boldsymbol{p}^{i+1}$, which can be used to generate an updated geometry S^{i+1}. Once this deformation loop, which is boxed within a dashed line in figure 2, has converged, the maximum von Mises stress σ_{max} can be computed.

Performing a structurally constrained gradient-based optimization would require differentiating the entire described chain to compute the gradient

$$\frac{\partial \sigma_{max}}{\partial \boldsymbol{\alpha}} \in \mathbb{R}^{1 \times d}. \tag{9}$$

Typically, it would hold that $d \gg 1$, which gives the opportunity of a cheap gradient computation using adjoints. However, this would involve computing the adjoints of the entire cold-to-hot chain. A first step towards computing the adjoints of this entire chain is made by computing the adjoints of the cold-to-hot transformation

$$\boldsymbol{p}(\Delta \boldsymbol{d}) = \boldsymbol{p}^0 + \Delta \boldsymbol{p}(\Delta \boldsymbol{d}). \tag{10}$$

The corresponding 1st-order adjoint model is given by

$$\Delta \boldsymbol{d}_{(1)} = \frac{\partial \Delta \boldsymbol{p}}{\partial \Delta \boldsymbol{d}}^T \boldsymbol{p}_{(1)}, \tag{11}$$

which will yield the gradient

$$\frac{\partial \Delta \boldsymbol{p}(\Delta \boldsymbol{d})}{\partial \Delta \boldsymbol{d}} = \frac{\partial \boldsymbol{p}(\Delta \boldsymbol{d})}{\partial \Delta \boldsymbol{d}} \in \mathbb{R}^{n \cdot m \times s} \tag{12}$$

by performing $n \cdot m$ runs of the adjoint model (11). Each adjoint run i is seeded with the i-th unit vector $\vec{\boldsymbol{i}} \in \mathbb{R}^{n \cdot m}$, where $\vec{\boldsymbol{i}}$ is a vector of 0s, except for the i-th variable, which is a 1 :

$$\boldsymbol{p}_{(1)} = \vec{\boldsymbol{i}} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{13}$$

Analogously, the 1st order tanget model is given by

$$\boldsymbol{p}^{(1)} = \frac{\partial \Delta \boldsymbol{p}}{\partial \Delta \boldsymbol{d}} \Delta \boldsymbol{d}^{(1)}, \tag{14}$$

which allows the computation of the gradient (12) by performing s runs with each run j seeded with a unit vector $\vec{\boldsymbol{j}} \in \mathbb{R}^{s}$:

$$\Delta \boldsymbol{d}^{(1)} = \vec{\boldsymbol{j}}. \tag{15}$$

For the differentiation with AD, the cold-to-hot algorithm is not treated as a black box. Notably, the first step of the cold-to-hot transformation, the projection of FEM nodes onto the surface (section 3.1), does not need to be differentiated. This is because the projection is performed only to determine the foot points (u_k, v_k) of a corresponding FEM node $\boldsymbol{d}_k$. Differentiating equations (6) and (7) is sufficient to compute the gradient (12). This is achieved using the open-source AD tool CoDiPack [1] developed by the Chair for Scientific Computing at TU Kaiserslautern, Germany. The results of the adjoint sensitivity computation compared against finite difference (FD) approximations are presented in section 5.2.

5 RESULTS

In this section, test results of the implemented cold-to-hot transformation and its sensitivity computation are presented. The selected test case is an axial fan blade geometry, which has been used as a baseline geometry for an optimization in [2].

5.1 Cold-to-hot transformation results

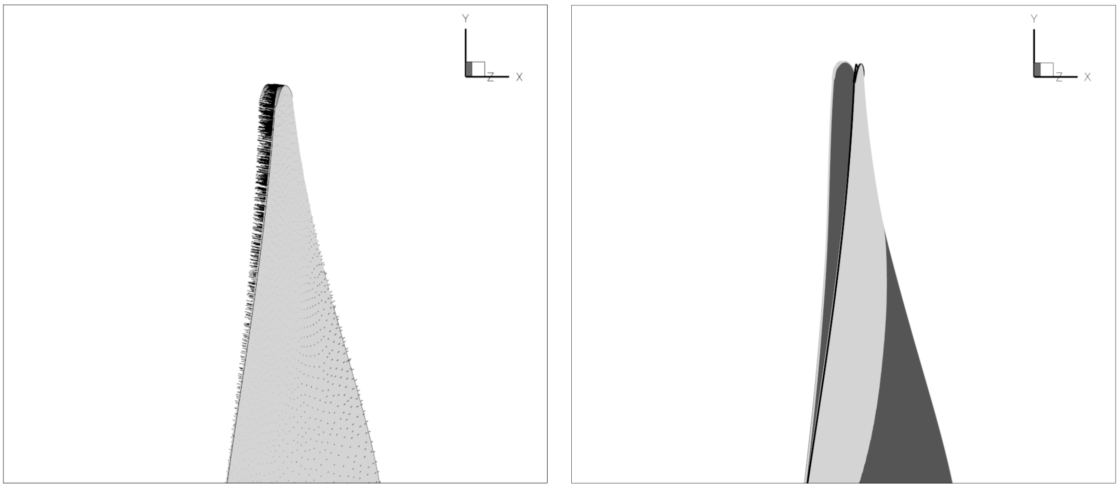
The results of the cold-to-hot transformation of the axial fan are quantified by computing the distance error between the deformed B-spline surface $\widehat{S}(u,v)$ and the displaced points $\boldsymbol{d}$. The error is computed in the same manner as in [5]:

$$\Delta_k = \min_{u,v \in [0,1]} ||\boldsymbol{d}_k - \widehat{S}(u,v)||^2, \quad k = 0, ..., s-1 \tag{16}$$

The maximum and mean distance errors are computed as

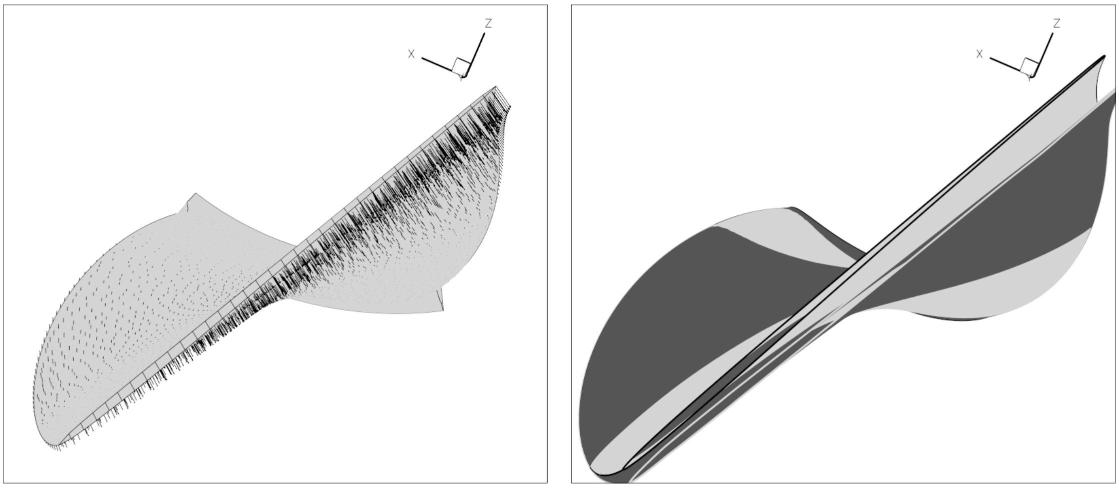
$$\Delta_{max} = \max_k \Delta_k, \quad \Delta_{mean} = \frac{1}{s} \sum_{k=0}^{s-1} \Delta_k. \tag{17}$$

The deformations are visualized in figures 3 and 4, which show that the deformed blade geometry in figures 3(b) and 4(b) follows the displacements given by the vectors in figures 3(a) and 4(a). Distance error results are summarized in table 1.



(a) displacements vectors of FEM results (b) deformed blade: dark. original blade: light

Figure 3: axial fan cold-to-hot transformation



(a) displacements vectors of FEM results (b) deformed blade: dark. original blade: light

Figure 4: axial fan cold-to-hot transformation

	Suction Side	Pressure Side
Δ_{max}[m]	2.83×10^{-3}	2.49×10^{-3}
Δ_{mean}[m]	7.36×10^{-5}	7.53×10^{-5}

Table 1: Maximum distance error Δ_{max} and mean distance error Δ_{mean} of the axial fan blade

5.2 Sensitivity computation results

The cold-to-hot transformation was differentiated using CoDiPack in both forward and reverse mode to compute the 1st order sensitivities (12). The input variables are the FEM node displacements $\Delta \boldsymbol{d} \in \mathbb{R}^s$ and the output variables are the updated control points $\boldsymbol{p} \in \mathbb{R}^{n \cdot m}$. The number of control points $n \cdot m$ is less than the number of FEM nodes s, with

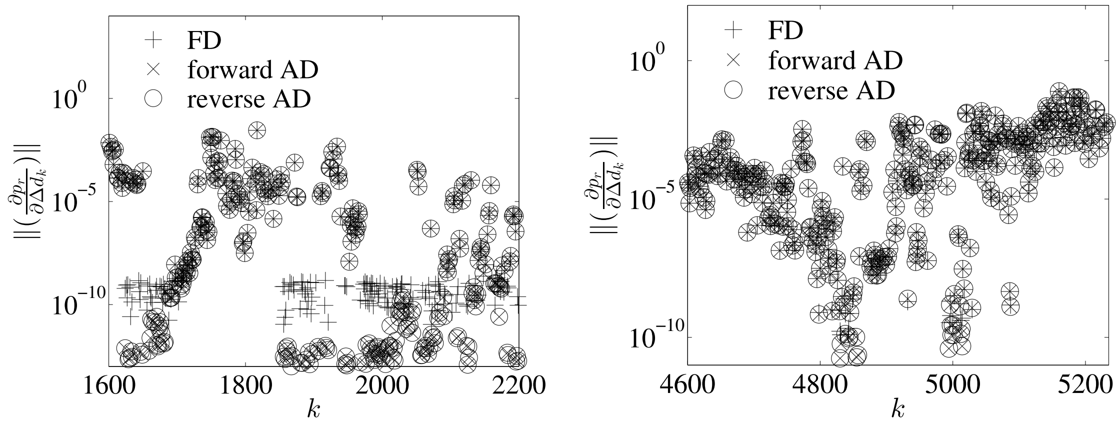
$$n \cdot m = 1053, \quad s = 8149. \tag{18}$$

A finite-difference (FD) approximation of (12) would require $2 \cdot s$ runs of (10) and a forward AD evaluation of the gradient requires s runs of the tangent model (14). On the other hand, computing (12) using reverse AD would require only $n \cdot m$ runs of the adjoint model (11). Thus, a faster run time is expected and also observed for the adjoint AD computation compared to the FD approximations and tangent AD runs (see table 2). The relative run time results are compared to the measured cost of the primal computation of the control point displacements, excluding the FEM node projection, $cost(P) \approx 4.52$ seconds. Note that in the final application of the adjoint cold-to-hot chain, only a single adjoint computation of the cold-to-hot transformation is necessary for each output parameter of the entire chain.

	FD	forward AD	reverse AD
absolute run time [min]	83.57	98.06	11.94
relative run time	$1110.42 \cdot cost(P)$	$1303.04 \cdot cost(P)$	$158.61 \cdot cost(P)$

Table 2: run time comparison of FD vs forward AD vs reverse AD for the 1st order sensitivities (12), with $cost(P) \approx 4.52$ seconds

Figures 5(a) and 5(b) present a comparison between gradients computed using FD, forward AD, and reverse AD for selected rows r and columns k of $\frac{\partial \boldsymbol{p}_r}{\partial \Delta \boldsymbol{d}_k}$. The plots show a good agreement between the FD and AD computed values with discrepancies between FD and AD at values of around $\mathcal{O}(10^{-10})$ in figure 5(a). A maximum discrepancy of $\mathcal{O}(10^{-6})$ and a mean discrepancy of $\mathcal{O}(10^{-13})$ between FD and AD was computed for the entire matrix (12). A visualization of the 1st-order sensitivities of a single control point with respect to the FEM displacements $\Delta \boldsymbol{d}$ is shown in figure 6.



(a) row $r = 100$

(b) row $r = 200$

Figure 5: Comparison of FD vs forward AD vs reverse AD on a semi-logarithmic plot row of $\frac{\partial \boldsymbol{p}_r}{\partial \Delta \boldsymbol{d}}$

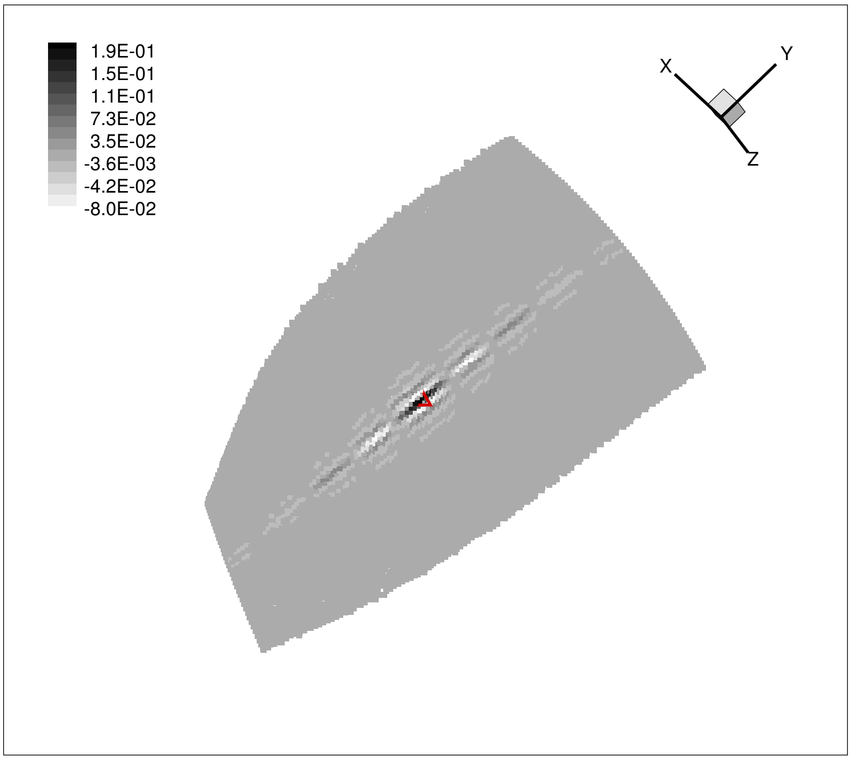


Figure 6: 1st-order sensitivities with respect to $\Delta \boldsymbol{d}$ of center control point, denoted by Δ

6 CONCLUSIONS

With the implementation and adjoint differentiation of the cold-to-hot transformation, an important component of the cold-to-hot chain was addressed. The AD computed sensitivities agree well with the FD computed sensitivities, paving the way towards an adjoint multidisiplinary optimization for turbomachinery components. Further work could involve a local error reduction in the cold-to-hot transformation, as well as the inclusion of pressure forces in the cold-to-hot transformation to couple the fluid and solid computations. An adjoint differentiation of a CSM solver is the next key component of the cold-to-hot chain that is to be implemented.

7 ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support of the IODA (*Industrial Optimal Design using Adjoint CFD*) project, which led to the work presented in this paper. The IODA project is funded by the European Commission within the Horizon 2020 program.

8 NOMENCLATURE

$J \in \mathbb{R}^c$	cost function
$J_{(1)} \in \mathbb{R}^c$	adjoint cost function
$\boldsymbol{x},\ \boldsymbol{\alpha} \in \mathbb{R}^d$	design parameters
$\boldsymbol{x}_{(1)} \in \mathbb{R}^d$	adjoint design parameters
$S,\ S_c,\ \widehat{S} \in \mathbb{R}$	B-spline surface (initial, cold state, deformed)
$N^p,\ N^q \in \mathbb{R}$	B-spline basis functions
$p,\ q \in \mathbb{N}$	B-spline basis function orders
$u,\ v \in \mathbb{R}$	B-spline foot points
$\boldsymbol{p}^0_{i,j},\ \boldsymbol{p}_{i,j} \in \mathbb{R}^3$	control points (initial, displaced)
$\boldsymbol{p}^{(1),i,j} \in \mathbb{R}^3$	tangent control points
$\boldsymbol{p}_{(1),i,j} \in \mathbb{R}^3$	adjoint control points
$\Delta\boldsymbol{p}_{i,j} \in \mathbb{R}^3$	control point displacements
$n \cdot m \in \mathbb{N}$	number of control points
$\boldsymbol{d}^0_k,\ \boldsymbol{d}_k \in \mathbb{R}^3$	FEM nodes (initial, displaced)
$\Delta\boldsymbol{d}_k \in \mathbb{R}^3$	FEM node displacements
$\Delta\boldsymbol{d}^{(1),k} \in \mathbb{R}^3$	tangent FEM node displacements
$\Delta\boldsymbol{d}_{(1),k} \in \mathbb{R}^3$	adjoint FEM node displacements
$s \in \mathbb{N}$	number of FEM nodes
$\mathbf{K}_{edge}$	set of FEM nodes along surface edge
$\mathbf{K}_{inner}$	set of inner FEM nodes
$\sigma_{max} \in \mathbb{R}$	maximum von Mises stres
Ω_s	solid mesh
$F_c \in \mathbb{R}^3$	centrifugal forces
$\vec{\boldsymbol{i}} \in \mathbb{R}^{n \cdot m}$	i-th unit vector
$\vec{\boldsymbol{j}} \in \mathbb{R}^s$	j-th unit vector
$\Delta_k,\ \Delta_{max},\ \Delta_{mean} \in \mathbb{R}$	distance error (k-th, maximum, mean)

REFERENCES

[1] T. Albring, M. Sagebaum, and N.R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. *AIAA 2015-3240*, 2015.

[2] Christopher Chahine, Tom Verstraete, and Li He. Multidisciplinary design optimization of an aero-engine fan blade with consideration of bypass and core performance.

[3] Mihai C Duta, Shahrokh Shahpar, and Michael B Giles. Turbomachinery design optimization using automatic differentiated adjoint code. In *ASME Turbo Expo 2007: Power for Land, Sea, and Air*, pages 1435–1444. American Society of Mechanical Engineers, 2007.

[4] Gerald Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.

[5] Lilia Gaun, Dieter Bestle, and André Huppertz. Hot-to-cold cad geometry transformation of aero engine parts based on b-spline morphing. In *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*, pages V02BT45A020–V02BT45A020. American Society of Mechanical Engineers, 2014.

[6] MB Giles. Some thoughts on exploiting cfd for turbomachinery design. *University of Oxford Technical Report*, 1998.

[7] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Siam, 2008.

[8] Antony Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3(3):233–260, 1988.

[9] Antony Jameson, Sriram Shankaran, and Luigi Martinelli. Continuous adjoint method for unstructured grids. *AIAA journal*, 46(5):1226–1239, 2008.

[10] Jiaqi Luo, Chao Zhou, and Feng Liu. Multipoint design optimization of a transonic compressor blade by using an adjoint method. *Journal of Turbomachinery*, 136(5):051005, 2014.

[11] Dimitri J Mavriplis. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA journal*, 45(4):741–750, 2007.

[12] Lasse Mueller, Zuheyr Alsalihi, and Tom Verstraete. Multidisciplinary optimization of a turbocharger radial turbine. *Journal of Turbomachinery*, 135(2):021022, 2013.

[13] Uwe Naumann. *The art of differentiating computer programs: an introduction to algorithmic differentiation*, volume 24. Siam, 2012.

[14] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012.

[15] Olivier Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(01):97–110, 1974.

[16] James Reuther and A Jameson. Control theory based airfoil design using the euler equations. In *5th Symposium on Multidisciplinary Analysis and Optimization, American Institute of Aeronautics and Astronautics, Reston*, pages 206–222, 1994.

[17] Tom Verstraete. Multidisciplinary turbomachinery component optimization considering performance, stress, and internal heat transfer. *University of Ghent, Ph. D. Thesis*, 2008.

[18] Benjamin Walther and Siva Nadarajah. Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor. *Journal of turbomachinery*, 135(2):021017, 2013.