Adjoint-based Shape Sensitivities for Turbomachinery Design Optimizations

Anna Engels-Putzka and Jan Backhaus

Abstract An adjoint preprocess for an adjoint-based turbomachinery design process is described. The resulting process is compared to a previously established adjoint process based on three-dimensional adjoint solutions and deformed meshes. Within the new process, the calculation of sensitivities is performed using shape sensitivities and surface displacements, where the design parameters are computer-aided design (CAD) parameters describing the geometry of a turbomachinery component. In particular, an automatized process for the generation of surface displacements from variations of the CAD parameters is described. Shape sensitivities are calculated using an adjoint elliptic mesh deformation tool. The method is applied to a counter-rotating fan. Integrating the adjoint preprocess into the gradient evaluation results in a significantly reduced memory usage, since no perturbed three-dimensional meshes have to be generated. It also has the potential to reduce the dependency of the computation time on the number of design parameters. Moreover, the shape sensitivities provide interesting insights into the design problem.

1 Introduction

In the field of turbomachinery a substantial amount of time in the design process is spent in improving the aerodynamical properties of fans, compressors and turbines. These components are already very sophisticated and can only be improved by using highly accurate flow simulations and exploring a large variety of shapes for the blades and the duct. Optimization techniques could help to efficiently search for

Anna Engels-Putzka
Institute of Propulsion Technology, German Aerospace Center (DLR), Linder Hoehe, 51147 Cologne, Germany, e-mail: anna.engels-putzka@dlr.de

Jan Backhaus
Institute of Propulsion Technology, German Aerospace Center (DLR), Linder Hoehe, 51147 Cologne, Germany, e-mail: jan.backhaus@dlr.de

improved designs in large parameter spaces. These techniques may be divided into gradient-free and gradient-based algorithms. Gradient-free techniques only use the simulation process in a black-box manner, which makes them quite easy to apply to simulation-based performance-evaluation processes. However, the computational costs are considered too high for the use in the regular design process, due to the high number of objective function evaluations needed. Gradient-based methods usually require less objective function evaluations, therefore it would be desirable to apply these methods in the turbomachinery design process. However, a substantial reduction in computational costs for the complete optimization will only be realized if the additional calculation of gradients of the cost function is efficient, in terms of runtime and memory consumption. The adjoint method [1, 2] is a means for calculating gradients of simulation-based objective functions which is efficient for large numbers of design parameters. In theory, the cost of one adjoint gradient evaluation is independent of the number of design parameters and in the same order of magnitude as a flow simulation. In practice, most design processes can not be completely adjoined. The problem is that practical objective function evaluations are carried out by running a sequence of programs of different origin and complexity. For a completely adjoined process all of these programs would have to be adjoined (compare for example [3]), which is not always possible and at least a very tedious task. An acceleration of the sensitivity evaluation is already achieved when one adjoins only the end of the process. One may start by adjoining the flow solver and its post-processing, which are also the computationally most expensive steps. The sensitivities of the first parts of the process chain are then approximated using finite differences. While this strategy is successfully applied to turbomachinery design optimizations [4, 5], it can become inefficient for larger numbers of design parameters, due to the creation, storage and processing of perturbed three-dimensional meshes. This effort could be avoided by adjoining the mesh generating tool. Since this is often not possible, we propose adjoining an elliptic mesh deformation tool instead. The resulting process is described and discussed in the following text.

While the theory and implementation of the adjoint mesh deformation have already been presented in a previous publication [6], the focus here is on its integration into an evaluation process which is suitable for the use in an optimization framework. This is discussed in Sect. 2, together with a short review of the previously introduced adjoint preprocess. The second main objective of this paper is the demonstration of the whole process on a realistic turbomachinery design. This application is described in detail in Sect. 3, while in Sect. 4 we present and discuss the obtained results. Section 5 gives a short summary of our findings.

2 Methods

A typical process for an aerodynamic performance evaluation of a turbomachinery component starts by building a CAD model of the blades and the duct from a set of design parameters. Next, the computational mesh is generated, first on the surfaces and then for the whole (three-dimensional) computational domain. After that, the flow simulation is carried out and in a postprocessing step the desired objective functions are evaluated from the obtained flow solution. This process chain is illustrated in Fig. 1.

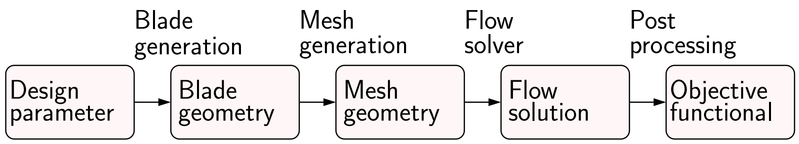

Fig. 1 Schematic representation of a process chain for sensitivity evaluation (reproduced from [6])

In the following we want to give a short mathematical description of this process and its adjoint. For details we refer to [6]. In the last subsection (2.3) we discuss implementational aspects of the sensitivity evaluation based on this surface-based adjoint process.

2.1 Adjoint Method and Mesh Sensitivities

We denote the vector of design parameters by α and assume that an objective functional I is given as $I(q(\alpha))$, i.e. it depends on α only through the flow field q. If we denote further by x the coordinates of the vertices in the (three-dimensional) computational mesh and by y the coordinates of the vertices on the blade surfaces, the process described above is represented by the functional dependence

$$\alpha \mapsto y \mapsto x \mapsto q \mapsto I \tag{1}$$

and the derivative of I with respect to α can be computed using the chain rule. As a first step, we have

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial q}\frac{dq}{d\alpha}. \tag{2}$$

Now q is given implicitly by the condition $R(q,x) = 0$, where R is the residual of the discretized flow equations, so $\frac{dq}{d\alpha}$ can be determined from the equation

$$0 = \frac{d}{d\alpha}R(q(\alpha),x(\alpha)) = \frac{\partial R}{\partial q}\frac{dq}{d\alpha} + \frac{\partial R}{\partial x}\frac{dx}{d\alpha}. \tag{3}$$

If the adjoint method is used, we have

$$\frac{dI}{d\alpha} = -\psi^t \frac{\partial R}{\partial x}\frac{dx}{d\alpha}, \tag{4}$$

where the adjoint solution ψ is given by

$$\left(\frac{\partial R}{\partial q}\right)^t \psi = \left(\frac{\partial I}{\partial q}\right)^t. \tag{5}$$

This adjoint solution can be interpreted as the sensitivity of the functional I with respect to source terms for each primary variable in each cell in the flow field.

The term $\frac{\partial R}{\partial x}\frac{dx}{d\alpha}$ on the right hand side of Eq. 4 can be either approximated as a whole by a finite difference, or the two factors can be considered separately. The first approach is used in the adjoint process which we use for comparison. In this case, the evaluation of Eq. 4 can be considered as a scalar product of two vector fields over the three-dimensional computational mesh.

Now we discuss the explicit computation of $\frac{\partial R}{\partial x}$. Technically, it is convenient to further split up this computation into two steps, since the coordinates of the mesh vertices do not appear in the residual R explicitly. Instead, geometrical properties of the mesh cells and their faces (e.g. cell volumes, coordinates of cell centers, face normal vectors) are derived from these coordinates. While these quantities are given by simple algebraic expressions in terms of the vertex coordinates (see e.g. [7]), which can be differentiated analytically, the derivatives of the residual with respect to the intermediate quantities are essentially computed as finite differences.

We have implemented the computation of $\frac{\partial R}{\partial x}$ as a postprocessing step for the adjoint solution, and the computed derivative is directly multiplied by the adjoint solution. The result can be interpreted as the sensitivity of the functional with respect to the coordinates of the individual mesh vertices, therefore we refer to it in the following as mesh sensitivities.

2.2 *Adjoint Mesh Deformation*

The approach to sensitivity computation described in the previous section requires that for each parameter variation a new mesh is generated. Instead of generating a new mesh from scratch for a (small) parameter variation, one can also take a deformation of the blade surface and propagate this to the three-dimensional mesh using an elliptic mesh deformation algorithm [8, 9]. Such an algorithm is implemented inside the preprocessing tool PREP [10]. The basic idea is that the deformation vector δx is given as the solution of the Poisson equation

$$\nabla \cdot (E(x)\nabla(\delta x)) = 0, \tag{6}$$

where the modulus E is proportional to the inverse cell volume. The discretization of Eq. 6 leads to a linear equation system

$$A\delta x = B\delta y, \tag{7}$$

where δy denotes the prescribed surface deformation, and B is a boundary operator. The system in Eq. 7 is solved in a block-parallel manner, where the deformation is also transported across several blocks.

With the adjoint approach, we determine the variation of a functional I depending on a parameter α under the assumption that a variation in α is first transformed into a variation of surface coordinates δy, i.e.

$$\delta I = \frac{dI}{dy}\delta y. \tag{8}$$

The derivative $\frac{dI}{dy}$ – which can be interpreted as a field of surface sensitivities – is then given as

$$\frac{dI}{dy} = -\xi^t B + \frac{dI}{dx}B', \tag{9}$$

where ξ denotes the adjoint mesh deformation, B is the boundary operator from Eq. 7 and B' another boundary operator which is applied to the mesh sensitivities. The adjoint mesh deformation ξ is the solution of the linear equation

$$A^t\xi = C\left(\frac{dI}{dx}\right)^t, \tag{10}$$

where A is the system matrix of Eq. 7 and C stands for the adjoint boundary operator. For the solution of Eq. 10 the same solver as for Eq. 7 can be used.

The evaluation of Eq. 8 consists of a scalar product of two vector fields on the blade surfaces. This means that the computational cost for this step is negligible compared to the rest of the process.

We assume here that both the surface displacements δy and the surface sensitivities $\frac{dI}{dy}$ are given in the same discretization of the surface. If this is not the case, an additional interpolation step is needed, i.e. we have (for each mesh vertex i on a surface)

$$\delta y_i = \sum_j w_{ij}\delta\tilde{y}_j. \tag{11}$$

This step can of course also been adjoined, then Eq. 8 takes the form

$$\delta I = \left(W^t\frac{dI}{dy}\right)\delta\tilde{y}. \tag{12}$$

2.3 Surface Displacements and Sensitivity Evaluation

Now we discuss the practical realization of the evaluation process, in particular the automatized generation of surface displacements from design parameters. The data flow in this process is illustrated in Fig. 2, where the different shadings indicate how often a step in the process has to be carried out. The sequence of steps from the design parameters to the forward flow solution, on which the adjoint computations are based, has to be performed only once (light gray). The adjoint solution and the corresponding surface sensitivities have to be computed for each, of typically one

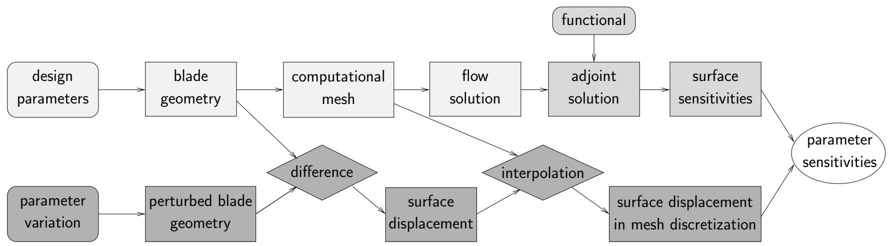

Fig. 2 Schematic representation of data flow in adjoint process

to ten different, objective functionals (gray). A field of surface variation vectors, called surface displacement, has to be generated for each, of the typically hundreds to thousands, of design parameters (dark gray). This part of the process starts with a variation in one of the design parameters, from which a perturbed blade surface is generated. Surface meshes for both the original and the perturbed blade are generated by applying the same discretization to the respective surface splines. By calculating the difference between the perturbed and the original blade surface mesh one obtains the surface displacement. The discretization used here is in general not the same discretization as that used by the mesh generation tool for the computational mesh. Therefore, the result has to be interpolated to the discretization of the surface in the computational mesh (cf. Eq. 11). For this task, a surface mapping and interpolation algorithm implemented in the preprocessing tool PREP [10] is used. Since the interpolation matrix W is constant for all parameters, it is efficient to calculate it only once and apply it to each surface displacement, or even better, to apply its adjoint to the surface sensitivities and then perform the evaluation in the originally chosen discretization (cf. Eq. 12). However, for simplicity of the prototype process, we have not yet implemented this.

The block structure employed by the flow solver as well as the adjoint mesh deformation is exploited also in the sensitivity evaluation process. The surface displacements are only calculated for those blocks which are adjacent to a blade surface that has been deformed, and for each such block a separate file is created. While each functional has to be combined with each surface displacements, the results for the different blocks are independent of each other and only have to be added up. Consequently, we take the iteration over the blocks as the outermost loop, where only the relevant blocks are included. Moreover, in the computation of the scalar product, only those parts of the blade surfaces which have actually been deformed are taken into account, so that the evaluation step can be performed very efficiently.

3 Application

The described methods are demonstrated on a turbomachinery application in the following section.

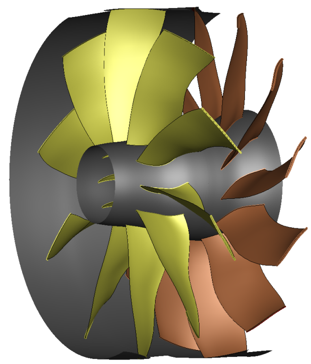

Fig. 3 CAD model of CRISP 2

The computational setup is taken from a recent study in which a counter-rotating integrated shrouded propfan, in short CRISP, has been re-designed using aerodynamic and structural optimization techniques without the aid of sensitivity information [11]. The result, called CRISP 2 (cf. Fig. 3), is used in here. This numerical setup has already been used to validate the adjoint process based on deformed meshes which were created by re-meshing perturbed blade shapes [12]. In the present study, the adjoint surface-based process, using the adjoint mesh deformation algorithm, is applied to this design. The geometry of the model is described by means of a CAD parameterization. Each fan blade is determined by airfoil cross sections which are described through their stagger angle, leading and trailing edge angles, an asymmetry factor at the leading edge, a set of spline control points along the suction side of the cross section and thickness parameters that relate the pressure side curve to the suction side curve in order to generate the spanwise thickness distribution. To keep the number of parameters manageable, not every parameter on every cross section is directly used as a free parameter, instead their values are derived from radial distributions. These radial distributions are chosen to be piecewise linear functions with four supporting points on fixed radial positions which are used as the optimization parameters. Once the airfoil cross sections are created, they are positioned in space along a three-dimensional curve, which is described by streamwise and circumferential positions on supporting points at different radial positions. This results in about one hundred parameters being used for the design of the two blades.

From the blade shapes a block-structured computational mesh is created. This may be either a coarse mesh using wall functions with $y^+ > 35$ and comprising about 700,000 cells, or a finer mesh which resolves the boundary layer, consisting of about 2.5 Million cells. The first mesh is intended for the use in the first stages of the optimization to keep computational times as low as possible. The second mesh can be used to re-evaluate intermediate results with higher precision and to perform detail optimizations in subsequent optimization cycles.

The aerodynamic properties of the designs are evaluated using DLR's flow solver for turbomachinery flows TRACE [13, 14] solving the steady Reynolds-averaged

Navier–Stokes (RANS) equations using the Wilcox k-ω turbulence model. The operating point is described by means of a radial equilibrium equation for the static pressure at the exit of the domain. A mixing plane approach serves for the coupling of the counter-rotating blade rows, non-reflecting boundary conditions are used at the entry and the exit of the domain and the no-slip condition is prescribed on solid walls. The adjoint solver employed here is the discrete adjoint of TRACE, which constructs the adjoint linear system of equations for the same discretization as in the forward solver, by using finite differences for the flux discretization and manually implemented adjoint boundary conditions. For details cf. [15].

4 Results and Discussion

For the validation of our process we compare the surface-based sensitivities to those obtained from the adjoint solutions using deformed meshes (according to Eq. 4), and to sensitivities computed as finite differences using the nonlinear flow solver. The results for some functionals are shown in Fig. 4. For the presentation we use a reduced set of 48 parameters to make the plots clearer.

We see a very good agreement of all three curves for the functionals mass flow and total pressure ratio. In Fig. 5 the deviations of the sensitivities obtained by the two adjoint processes from the reference (finite differences) are plotted for the mass flow. The results of both processes are of similar quality and the deviations are at least one order of magnitude smaller than the sensitivities. In order to judge these deviations, one has to keep in mind that the benchmark results are finite differences of a complete design evaluation chain. It is a quite difficult task to find variation sizes for the design parameters that on the one hand are large enough to be not cancelled out by rounding effects or convergence inaccuracies and on the other hand are small enough to not cause nonlinear behavior in one of the process steps. While special care has been taken to maintain the highest possible accuracy in all programs and interfaces involved, the assessment whether the variation is still in the linear range can only be assured by careful adjustment for each parameter.

For the radial outflow angle, we see in Fig. 4**c** some larger deviations for parameters which affect the geometry of the second rotor. The deviation of the adjoint sensitivities based on the newly generated three-dimensional meshes from the other results is due to the fact that the value of this functional depends strongly on the mesh vertex distribution at the outlet. This dependency is not reflected in the adjoint sensitivities, since the adjoint solver uses the assumption that there is no direct dependency of I on α through the mesh vertex coordinates (cf. Sect. 2.1). This is a valid assumption for the functionals and parameters used in the the context of turbomachinery design. However, since it is not possible to apply restrictions on how the mesh generation tool distributes vertex points in the computational domain, we can not avoid that the mesh changes substantially at e.g. the outlet when the blade shape changes. This effect is illustrated for one parameter in Fig. 6**a** showing large changes in the mesh vertices after re-staggering the airfoil in the tip cross-section area and

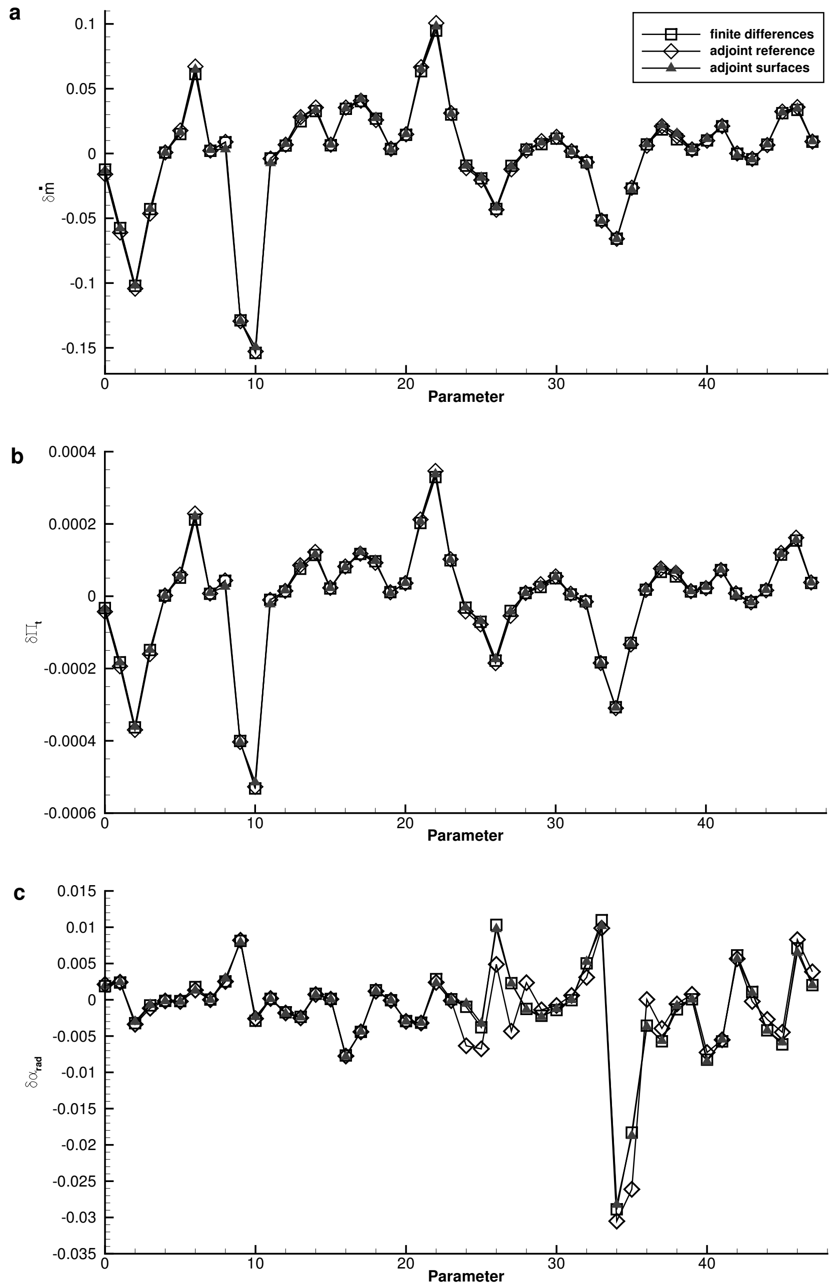

Fig. 4 Comparison of the sensitivities obtained with three different processes (forward process computing finite differences of nonlinear results, adjoint process based on deformed three-dimensional meshes, and adjoint process based on surface displacements) for three functionals: **a** mass flow, **b** total pressure ratio, **c** radial outflow angle

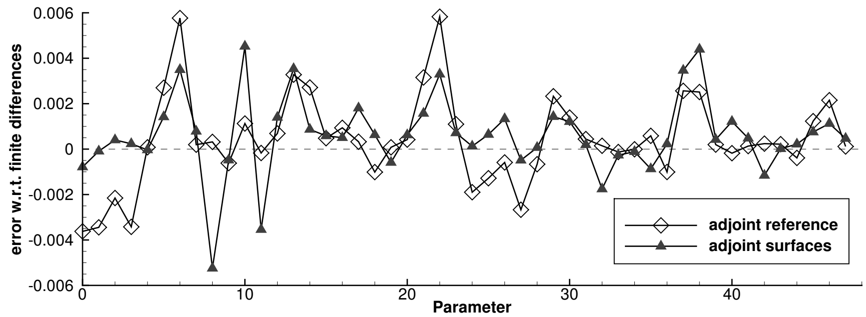

Fig. 5 Errors of adjoint mass flow sensitivities with respect to finite differences

re-meshing this blade. The mesh deformation tool allows to prescribe surface vertex displacements as boundary conditions and therefore allows to keep the surface vertices on inlet and outlet panels constant. Figure 6**b** shows that the deformed mesh from the elliptic mesh deformation tool keeps the vertices on the outlet constant. When the adjoint sensitivity is computed using this mesh, it agrees much better with the finite difference. Figure 4**c** also shows a good agreement of the surface-

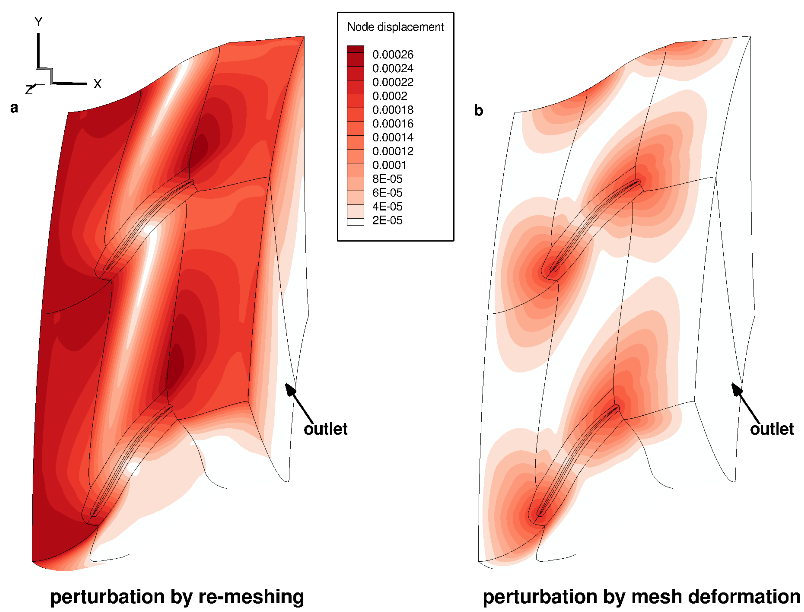

Fig. 6 Comparison of the mesh deformations for re-staggering the airfoil in the tip cross section in rotor 2: **a** deformed mesh created by re-meshing the deformed blade, **b** deformed mesh created by elliptic mesh deformation based on the deformed blade

based sensitivities – where the adjoint of this mesh deformation has been used – with this reference.

An algorithm calculating sensitivities from two-dimensional meshes can be expected to be faster than an algorithm producing the same results from three-dimensional meshes. That being said, we do not show timing comparisons of the two different processes here. The main reason is, that our proposed surface-based sensitivity process is not yet optimized for speed, while the three-dimensional mesh based process has undergone years of speed tuning. Most calculation time in the surface-based process is spent in not yet optimized two-dimensional input/output routines and the recalculation of the interpolation matrix for each perturbed surface. Switching to more efficient data formats and only applying the adjoint interpolation matrix (cf. Sect. 2.3) to the surface sensitivities would be necessary in order to compare the processes.

In contrast, the advantages of the new process in terms of memory usage can already be clearly seen. The mesh which we used for our computations needs roughly 18 MB of disk space. In the case of the finer mesh each deformed mesh is already 64 MB large. For one hundred free parameters this means to write, store and read about 6 GB of deformed meshes per gradient evaluation. In contrast, the surface displacements for each parameter need only 1.3 MB (for about 11400 elements). This means that for this test case we gain at least a factor of ten regarding memory consumption, which becomes particularly important when employing the adjoint process on high performance computing hardware where memory bandwidth, especially for storage, is a limiting factor.

Exemplary results of the adjoint mesh deformation are shown in Figs. 7 and 8. These are called sensitivity maps, since they display the influence of a variation of each surface node on the selected functional. Hereby these sensitivity maps help identifying surface regions with a greater and regions with less influence on the design criteria. Such plots can be used in selecting effective parameters for the optimization or to gain insight into the relation between changes in the shape and changes of the objective function. This makes the adjoint process advantageous even for design problems where optimization techniques are not employed.

5 Summary and Conclusion

We proposed a process for efficiently evaluating sensitivities of aerodynamic objective functions in order to perform gradient-based optimizations in the field of turbomachinery design based on flow simulations. The main focus in this work is on the case of CAD parameterized processes in which the mesh generator can not be adjoined. Therefore we suggested the use of an elliptic mesh deformation tool and its adjoint for calculating surface sensitivities. The interface to the geometry generators are arbitrarily discretized surfaces, i.e. meshes on blade and duct surfaces. While the assumption of arbitrary discretization requires an additional interpolation in the sensitivity evaluation step, it decouples the blade generator and the mesh gen-

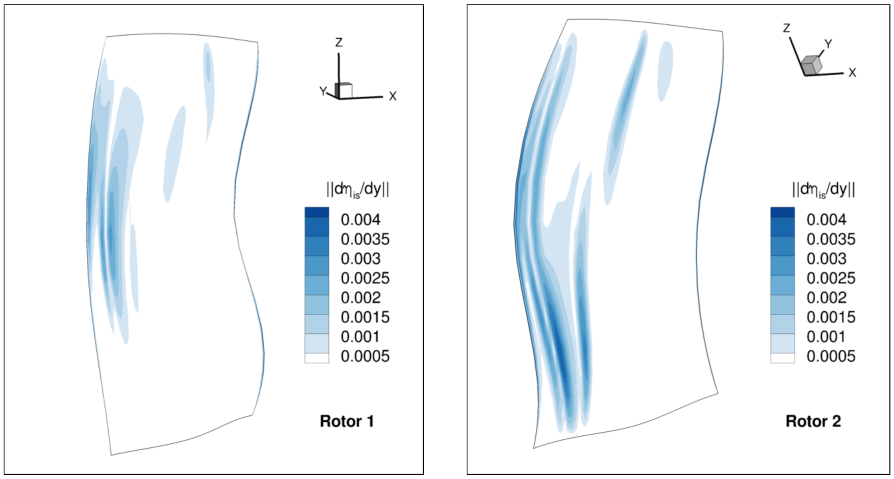

Fig. 7 Norm of the sensitivity vector of isentropic efficiency with respect to surface mesh vertex coordinates for the suction side of the rotor blades

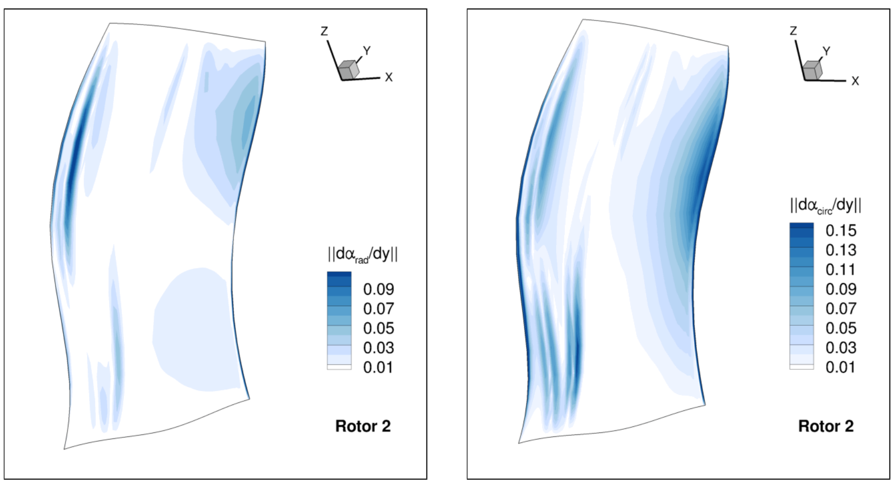

Fig. 8 Norm of the sensitivity vector of radial and circumferential outflow angles with respect to surface mesh coordinates for the suction side of the rotor blades of rotor 2

erator and therefore leads to a more universally applicable process. The main result however is a means to avoid the creation, storage and processing of a large number of three-dimensional mesh files for each sensitivity evaluation, since this is the main obstacle in further enlarging the design spaces from hundreds to thousands of parameters with the previously employed process.

By applying both the established and the newly proposed sensitivity calculation methods to a realistic turbomachinery configuration and comparing against nonlinear solutions, we demonstrated first the applicability of such a process, and second the agreement of sensitivities obtained with both processes and their finite difference approximations. We showed that differences in the calculated sensitivities, observed for the radial outflow angle, are due to difficulties in performing re-meshing of perturbed blade shapes and found that these differences can be avoided when using the elliptic mesh deformation process instead.

An adjoint mesh deformation process has the additional advantage of producing surface sensitivity maps as byproduct. In contrast to adjoint solutions, sensitivity maps can be directly interpreted to deliver additional and novel insight into the design problem.

Acknowledgements The authors would like to thank C. Frey and C. Voigt for carefully reading and commenting on the manuscript. Financial support by MTU Aero Engines (co-sponsorsip of the first author) and the German Ministry of Economy (Project R&E TURB, project number 20T1104B) is gratefully acknowledged.

References

1. Jameson, A.: Aerodynamic design via control theory. Journal of Scientific Computing **3**(3), 233–260 (1988)
2. Giles, M.B., Pierce, N.A.: An introduction to the adjoint approach to design. Flow, Turbulence and Combustion **65**, 393–415 (2000)
3. Gauger, N.R., Walther, A., Moldenhauer, C., Widhalm, M.: Automatic differentiation of an entire design chain for aerodynamic shape optimization. In: N. Kroll, J.K. Fassbender (eds.) New Results in Numerical and Experimental Fluid Mechanics VI, *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, vol. 96, chap. 13, pp. 181–193. Springer Berlin / Heidelberg (2008). DOI 10.1007/978-3-540-74460-3_56
4. Wang, D.X., He, L.: Adjoint aerodynamic design optimization for blades in multistage turbomachines—part I: Methodology and verification. Journal of Turbomachinery **132**(2), 021011 (2010). DOI 10.1115/1.3072498
5. Wang, D.X., He, L., Li, Y.S., Wells, R.G.: Adjoint aerodynamic design optimization for blades in multistage turbomachines—part II: Validation and application. Journal of Turbomachinery **132**(2), 021012 (2010). DOI 10.1115/1.3103928
6. Engels-Putzka, A., Frey, C.: Adjoint mesh deformation and adjoint-based sensitvitities with respect to boundary values. In: ECCOMAS 2012 - European Congress on Computational Methods in Applied Sciences and Engineering, e-Book Full Papers, pp. 1789–1808 (2012)
7. Blazek, J.: Computational fluid dynamics: principles and applications. Elsevier Science (2001)
8. Yang, Z., Mavripilis, D.J.: Unstructured dynamic meshes with higher-order time integration schemes for the unsteady navier-stokes equations. In: Proceedings of the 43th AIAA Aerospace Sciences Meeting (2005)
9. Dwight, R.: Robust mesh deformation using the linear elasticity equations. In: Computational Fluid Dynamics 2006, pp. 401–406. Springer (2006)
10. Voigt, C., Frey, C., Kersken, H.P.: Development of a generic surface mapping algorithm for fluid-structure-interaction simulations in turbomachinery. In: V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010 (2010)
11. Görke, D., Le Denmat, A.L., Schmidt, T., Kocian, F., Nicke, E.: Aerodynamic and mechanical optimization of CF/PEEK blades of a counter rotating fan. In: Proceedings of the ASME Turbo Expo 2012 (2012)
12. Backhaus, J., Aulich, M., Frey, C., Lengyel, T., Voß, C.: Gradient enhanced surrogate models based on adjoint CFD methods for the design of a counter rotating turbofan. In: Proceedings of the ASME Turbo Expo 2012 (2012)

13. Nürnberger, D., Eulitz, F., Schmitt, S., Zachcial, A.: Recent progress in the numerical simulation of unsteady viscous multistage turbomachinery flow. In: ISABE 2001-1081 (2001)
14. Becker, K., Heitkamp, K., Kügeler, E.: Recent progress in a hybrid-grid CFD solver for turbomachinery flows. In: V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010 (2010)
15. Frey, C., Nürnberger, D., Kersken, H.: The discrete adjoint of a turbomachinery RANS solver. In: Proceedings of the ASME Turbo Expo 2009 (2009)