A Comparative Study of Two Different CAD-Based Mesh Deformation Methods for Structural Shape Optimization

Marc Schwalbach*
von Karman Institute for Fluid Dynamics
Waterloosesteenweg 72, 1640 Sint-Genesius-Rode, Belgium
Email: marc.schwalbach@vki.ac.be

Tom Verstraete
Queen Mary University of London
Mile End Road, E1 4NS, London, UK
Email: t.verstraete@qmul.ac.uk

Jens-Dominik Müller
Queen Mary University of London
Mile End Road, E1 4NS, London, UK
Email: j.mueller@qmul.ac.uk

Nicolas R. Gauger
Kaiserslautern University of Technology
Paul-Ehrlich-Strasse 34, 67663 Kaiserslautern, Germany
Email: nicolas.gauger@scicomp.uni-kl.de

Summary

This work introduces and compares two different CAD-based mesh deformation methods. The methods are used within an adjoint structural shape optimization, which is part of an evolving CAD-based adjoint multidisciplinary optimization framework for turbomachinery components.

During an optimization, the CAD geometry is updated at each design iteration, such that the structural mesh has to be deformed appropriately. The mesh is deformed in three stages. First, the nodes along the edges of the outer mesh are displaced to match the shape of the CAD edges, which are given by B-spline curves. Next, the remaining outer mesh nodes are displaced to match the shape of the CAD faces, which are given by B-spline surfaces. Finally, the outer mesh node deformations are used to solve for the inner node deformations using either an inverse distance interpolation or the linear elasticity analogy.

Coupling the mesh deformation with an adjoint structural solver enables gradient computations of structural constraints with respect to CAD design parameters. To compare the robustness of the two mesh deformation methods, a CAD-based structural shape optimization using each method was performed.

Keywords: adjoint optimization, algorithmic differentiation, mesh deformation, computational structural mechanics, linear elasticity, CAD, FEM, turbomachinery

1 Introduction

Multidisciplinary optimizations (MDO’s) have been extensively used to optimize turbomachinery components using gradient-free methods.1–3 Gradient-free methods are non-intrusive and do not require source code access to implement. On the other hand, they require a high number of iterations to converge and are limited by the curse of dimensionality. This means that the computational effort grows exponentially with the number of design parameters. Gradient-based methods use gradients to converge towards a local optimum using less iterations, but require the gradient of the cost function with respect to design parameters.
Computing the gradient using a non-invasive approach, such as second order finite differencing (FD), comes at a high computational cost that is proportional to the number of design parameters. Adjoint methods\textsuperscript{4,5} allow the computation of gradients at a cost proportional to the number of costs and constraints, which is typically far less than the number of design parameters. Algorithmic differentiation (AD)\textsuperscript{6,7} can be used to derive a discrete adjoint model from source code.

State of the art adjoint optimizations in turbomachinery focus on aerodynamic cost functions and constraints. Structural constraints are of high importance because the resulting shape should not only be aerodynamically optimal, but also structurally feasible. As a result, an adjoint structural solver has to be integrated within an MDO framework. In this work, the von Karman Institute’s optimization framework CADO\textsuperscript{8} is used. A CAD-based optimization is strived for, which would also allow for geometrical constraints to be imposed. For example, one may want to impose a constraint on the shape’s curvature for manufacturing purposes. The sensitivities of structural constraints with respect to mesh node coordinates can be computed within this framework,\textsuperscript{9} but the sensitivities with respect to CAD design parameters are missing to perform a CAD-based optimization.

An essential component required for CAD-based gradient optimization methods is a mesh morphing tool which adapts the mesh to the modifications of the CAD model, while maintaining the same mesh connectivity and node count. A re-meshing strategy would alter the mesh topology, causing the objective function to become discontinuous.

Two CAD-based mesh deformation methods are presented in section 2. Algorithmically differentiated versions of the mesh deformations can be used to perform an adjoint shape optimization, which will be discussed in section 3. The optimization results are used to compare the robustness of the two methods in section 4.

2 Mesh deformation method

While iterating through a CAD-based optimization process, the CAD design parameters are updated, morphing the CAD geometry. Based on the updated geometry, the structural mesh should be deformed to compute the cost function and the sensitivities of the updated design. Since the CAD geometry defines the outer skin of the structural mesh, it is used to compute an accurate deformation of the outer mesh nodes.

The CAD-based mesh deformation algorithm can be broken down into three hierarchical steps:

1. morph edge nodes
2. morph face nodes
3. morph inner nodes

The mesh deformations can be expressed as displacements \( u \in \mathbb{R}^{3m} \), where \( m \) is the number of structural mesh nodes. The first two steps are identical in both methods and are used to compute the outer mesh node displacements \( u_{outer} \in \mathbb{R}^{3m_o} \), where \( m_o \) denotes the number of FEM mesh nodes on the external surface of the mesh. The mesh deformation methods differ in the last step of morphing the inner mesh nodes. The three steps are briefly outlined in this section.

2.1 Morph edge nodes

First, the edges of the mesh are morphed based on the deformation of the CAD geometry edges, which are represented as B-spline curves \( C(u) \). The displacements of the first and last vertex nodes of the mesh edge are identical to the first and last points of the B-spline curve. Using this constraint, the displacements of the points along the curve can be solved for. Solving for the displaced foot points \( u \) in parametric CAD space, rather than fitting the mesh coordinates \( x \) to the B-spline curve \( C(u) \), reduces the degrees of freedom to one. The constraint of requiring these mesh points to be on the edge is then implicitly applied.

To visualize this procedure, consider the example presented in figure 1. After the CAD geometry is updated, the B-spline curve \( C \), which describes the edge, turns into the curve \( C^M \). As a result, the begin and end vertices (\( V_B \) and \( V_E \)) are morphed into \( V_B^M \) and \( V_E^M \). A mesh node \( P \) now has to be morphed into \( P^M \) by morphing its parametric coordinate \( u \) into \( u^M \). This is done using the parametric coordinates \( u_B, u_E \) of the begin and end vertices before morphing and their morphed parametric coordinates \( u_B^M, u_E^M \),

\[
    u^M = u_B^M + \frac{u_E^M - u_B^M}{u_E - u_B} (u - u_B)
\]

Figure 1: Morphing of edge mesh nodes using parametric CAD space
A linear spring analogy is used to relax the points along the curve. Performing this step for each edge of the CAD faces results in the displacements of the structural mesh edge nodes.

2.2 Morph face nodes
After having displaced the mesh nodes along the edges, the next step is to displace the remaining outer mesh nodes according to the CAD faces. Each CAD face is represented by a B-spline surface $S(u, v)$, which morphs into $S^M$. Using the computed edges from step 1 as boundary conditions, the inner $(u, v)$ foot points of the displaced CAD face are computed using an inverse distance interpolation. As with step 1, the displaced nodes are solved in parametric $(u, v)$ space to reduce the degrees of freedom to two, which automatically satisfies the constraint that the displaced mesh nodes have to remain on the CAD face. An illustration of this procedure is shown in figure 2. This is done for each face of the geometry.

2.3 Morph inner nodes

2.3.1 Method 1: using the inverse distance interpolation
The first mesh deformation method computes the inner node displacements $u_{\text{inner}} \in \mathbb{R}^{3m_i}$ using an inverse distance interpolation, where $m_i$ denotes the number of inner structural mesh nodes. The inverse distance interpolation is based on the displacements of the outer nodes $u_{\text{outer}} \in \mathbb{R}^{3m_o}$, i.e. the skin of the structural mesh, which are determined by the first two steps. An example of the resulting deformed mesh is shown in figure 3.

2.3.2 Method 2: using the linear elasticity analogy
The second mesh deformation method uses a linear elasticity analogy to solve for the inner node displacements $u_{\text{inner}} \in \mathbb{R}^{3m_i}$. The outer node displacements $u_{\text{outer}}$ are used as boundary conditions to the linear elastic problem

$$Au = b,$$

where $A$ is the stiffness matrix and $b$ is the load vector. A structural solver based on the finite element method (FEM) is used to solve for the mesh displacements $u$. A visualization of the resulting mesh deformation is presented in figure 4. For now, global material properties are used for the entire mesh, i.e. the Young’s modulus $E$ and Poisson’s ratio $\nu$ are constant throughout. These properties could also be defined locally to adjust the stiffness of certain parts of the mesh. Furthermore, the adjoint CSM solver can be recycled for the adjoint implementation of the mesh deformation.

3 CAD-based structural shape optimization
Previous work within this framework has enabled the computation of structural sensitivities with respect to FEM mesh nodes. The adjoint structural solver was differentiated using CoDiPack. While these gradients could be used to perform node-based optimizations, an
optimization using CAD design parameters is aspired. There are several reasons that motivate this approach, one being that CAD design parameters provide a more intuitive design space for engineers compared to computational meshes. Additionally, important geometric constraints can be imposed directly on an optimization, e.g. constraining the blade’s curvature for manufacturing purposes, minimum thickness requirements, etc. CAD-free parametrizations, such as free-form deformation, have a greater difficulty fulfilling such constraints. For gradient-based optimizations, the structural sensitivities with respect to the CAD parameters are required. This is achieved by closing the gap between the CAD-based mesh deformation and the structural solver.

For a structural optimization, a typical cost function would be the maximum von Mises stress $\sigma_{\text{max}}$. As a design space, consider the CAD parameters $\alpha \in \mathbb{R}^n$, which are used as inputs into the CAD kernel to generate the CAD geometry. In CADO, these could e.g. include the blade angle (fig. 5) and thickness distributions (fig. 6). Thus, for a gradient-based optimization, the gradients

$$\frac{\partial \sigma_{\text{max}}}{\partial \alpha} \in \mathbb{R}^n$$

are required. The respective adjoint model

$$\dot{x} = \dot{\sigma}_{\text{max}} \frac{\partial \sigma_{\text{max}}}{\partial x},$$

(4)

can be used to compute the gradients by seeding the model with $\dot{\sigma}_{\text{max}} = 1$.

Previous work has enabled the calculation of the gradient

$$\frac{\partial \sigma_{\text{max}}}{\partial x} \in \mathbb{R}^m$$

with respect to the FEM mesh nodes $x \in \mathbb{R}^m$. Knowing that the FEM mesh $x$ is dependent on the CAD geometry, which is generated based on the CAD parameters $\alpha$, it can be established that

$$\frac{\partial \sigma_{\text{max}}}{\partial \alpha} = \frac{\partial \sigma_{\text{max}}}{\partial x} \frac{\partial x}{\partial \alpha}.$$  

(6)

The gradient (5) can be calculated using the adjoint structural solver, while the gradient $\frac{\partial x}{\partial \alpha}$ can be computed by differentiating the mesh deformation in either forward or reverse mode AD. Using reverse AD, the structural sensitivities (5) could be used to seed the adjoint model of the mesh deformation

$$\dot{\alpha} = \frac{\partial x^T}{\partial \alpha} \dot{x} = \frac{\partial x^T}{\partial \alpha} \frac{\partial \sigma_{\text{max}}}{\partial x},$$

(7)

computing the gradient (6) with a single adjoint evaluation.

4 Optimization Results and Comparison

A radial turbine mesh, discretized using 10-node tetrahedral elements, was used to perform a structural optimization. The initial geometry is shown in figure 7, which contains approximately 30,000 elements. The objective of the optimization was to minimize the maximum von Mises stress $\sigma_{\text{max}}$, which is approximated using the $p$-norm

$$\sigma_{\text{max}} = \left( \sum_{i=0}^{m-1} \sigma_i \right)^{\frac{1}{p}},$$

(8)

using CAD parameters $\alpha$ as design variables. A steepest descent algorithm with a constant step size of $\Delta = 10^{-8}$ was used.

A convergence comparison of two optimizations using the two different mesh deformation methods is shown in figure 11. Overall, the inverse distance mesh deformation method is able to reach a better optimum (12.54% reduction in $\sigma_{\text{max}}$) as opposed to the linear elastic mesh deformation method (11.07% reduction). Using the linear elastic mesh deformation method, an optimum was obtained after 67 iterations and is shown in figure 9. After 24 iterations, the optimal geometry using the inverse distance mesh deformation was obtained and shown in figure 8. At 24 iterations, the mesh deformation method has also achieved a considerable cost reduction. of 10%.

However, using the inverse distance method already requires remeshing at iteration 8, while the linear elastic method requires its first remeshing at iteration 19. The linear elastic method also exhibits a greater cost function reduction until iteration 8. Here, the criterion to remesh is if the value of the new cost function $\sigma_{\text{max}}^{i+1}$ is greater than 5% of the current optimum $\sigma_{\text{max}}^i$, or if the cost function reduction is less than 0.01%.
Figure 7: Initial radial turbine geometry

Figure 8: Optimized radial turbine geometry using inverse distance mesh deformation at iteration 24

Figure 9: Optimized radial turbine geometry using linear elastic mesh deformation at iteration 67

Figure 10: Radial turbine geometry using linear elastic mesh deformation at iteration 24
5 Conclusion

This work introduced two unstructured mesh deformation methods for CAD-based adjoint optimizations. The mesh deformations take a CAD-based approach, especially for the deformation of the outer mesh nodes. This ensures an accurate conformity with the updated CAD geometry. The inner mesh deformations are computed using either an inverse distance interpolation or the linear elastic analogy with the help of a structural solver.

The structural solver, which has adjoint capabilities, additionally enables the computation of sensitivities of the structural cost function, e.g. the maximum von Mises stress, with respect to CAD design parameters. The sensitivities can be used to perform a structural shape optimization based on CAD design parameters. A structural optimization of a radial turbine has been performed using the different mesh deformation methods introduced in section 2. The comparison between the different methods shows that the inverse distance method requires remeshing at an earlier stage in the optimization compared to the linear elastic deformation method. However, using the inverse distance method, the optimizer has achieved a greater cost function reduction of 12.54%.

Future work would involve coupling the structural and fluid disciplines, as well as adding a vibration analysis. Specifically, coupling the adjoint chain of operations from CAD design parameters to structural constraints with an adjoint computational fluid dynamics (CFD) code. A CAD-based adjoint multidisciplinary optimization of a turbomachinery component can then be carried out.

6 Acknowledgments

The work presented in this paper has received funding from the European Commission through the IODA project under grant agreement number 642959.

7 Nomenclature

| Symbol | Description |
|--------|-------------|
| $m \in \mathbb{N}$ | number of FEM mesh nodes |
| $m_i \in \mathbb{N}$ | number of inner FEM mesh nodes |
| $m_o \in \mathbb{N}$ | number of outer FEM mesh nodes |
| $n \in \mathbb{N}$ | number of CAD design parameters |
| $u, v \in \mathbb{R}$ | B-Spline foot points |
| $u_B \in \mathbb{R}$ | foot point of begin vertex |
| $u^M_B \in \mathbb{R}$ | morphed foot point of begin vertex |
| $u_E \in \mathbb{R}$ | foot point of end vertex |
| $u^M_E \in \mathbb{R}$ | morphed foot point of end vertex |
| $b \in \mathbb{R}^{3m}$ | load vector |
| $u \in \mathbb{R}^{3m}$ | FEM mesh displacements |
| $u_{\text{inner}} \in \mathbb{R}^{3m_i}$ | inner FEM mesh displacements |
| $u_{\text{outer}} \in \mathbb{R}^{3m_o}$ | outer FEM mesh displacements |
| $x \in \mathbb{R}^{3m}$ | FEM mesh coordinates |
| $\tilde{x} \in \mathbb{R}^{3m}$ | adjoint FEM mesh coordinates |
| $A \in \mathbb{R}^{3m \times 3m}$ | stiffness matrix |
| $C \in \mathbb{R}^3$ | B-spline curve |
| $C^M \in \mathbb{R}^3$ | morphed B-spline curve |
| $E \in \mathbb{R}$ | Young’s modulus |
| $P \in \mathbb{R}^3$ | mesh point |
| $P^M \in \mathbb{R}^3$ | morphed mesh point |
| $S \in \mathbb{R}^3$ | B-spline surface |
| $S^M \in \mathbb{R}^3$ | morphed B-spline surface |
| $V_B \in \mathbb{R}^3$ | begin vertex |
| $V^M_B \in \mathbb{R}^3$ | morphed begin vertex |
| $V_E \in \mathbb{R}^3$ | end vertex |
| $V^M_E \in \mathbb{R}^3$ | morphed end vertex |
| $v \in \mathbb{R}$ | Poisson’s ratio |
| $\sigma_{\text{max}} \in \mathbb{R}$ | maximum von Mises stress |
| $\sigma_{\text{max}} \in \mathbb{R}$ | adjoint maximum von Mises stress |
| $\alpha \in \mathbb{R}^n$ | CAD design parameters |
| $\tilde{\alpha} \in \mathbb{R}^n$ | adjoint CAD design parameters |
| $\Delta \in \mathbb{R}$ | steepest descent step size |

References

[1] Willeke, S. and Verstraete, T. Adjoint optimization of an internal cooling channel u-bend. In ASME Turbo Expo 2015: Turbine Technical Conference and Exposition, American Society of Mechanical Engineers, (2015).

[2] Luo, J., Zhou, C., and Liu, F. Multipoint design optimization of a transonic compressor blade by using an adjoint method. Journal of Turbomachinery 136(5), 051005 (2014).

[3] Walther, B. and Nadarajah, S. Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor. Journal of turbomachinery 135(2), 021017 (2013).
[4] Pironneau, O. On optimum design in fluid mechanics. *Journal of Fluid Mechanics* **64**(01), 97–110 (1974).

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

[6] Griewank, A. and Walther, A. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Siam, (2008).

[7] Naumann, U. *The art of differentiating computer programs: an introduction to algorithmic differentiation*, volume 24. Siam, (2012).

[8] Verstraete, T. Cado: a computer aided design and optimization tool for turbomachinery applications. In *2nd Int. Conf. on Engineering Optimization, Lisbon, Portugal, September, 6–9*, (2010).

[9] Schwalbach, M., Verstraete, T., and Gauger, N. R. Developments of a discrete adjoint structural solver for shape and composite material optimization. In *The 7th International Conference on Algorithmic Differentiation*, (2016).

[10] Verstraete, T., Müller, L., and Müller, J.-D. Cad based adjoint optimization of the stresses in a radial turbine. In *Proceedings of ASME Turbo Expo 2017: Turbine Technical Conference and Exposition*, (2017).

[11] Albring, T., Sagebaum, M., and Gauger, N. R. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 3240, (2015).

[12] Schwalbach, M. and Verstraete, T. Towards multidisciplinary adjoint optimization of turbomachinery components. In *Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering, M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. P., editor, 3999 – 4010*, (2016).