Proceedings of the 7th GACM Colloquium on Computational Mechanics for Young Scientists from Academia and Industry October 11-13, 2017 in Stuttgart, Germany

Regenerating CAD Models with OpenCASCADE and pythonOCC from Numerical Models with Application to Shape Optimization

Altug Emiroglu[1]*, Andreas Apostolatos[1], Roland Wüchner[1] and Kai-Uwe Bletzinger[1]

Micro Abstract
Sensitivity filtering methods are unavoidable when numerical shape optimization is considered. A mortar based sensitivity filtering method which incorporates underlying CAD parametrization of the numerical models is proposed. The method is combined with a software environment which utilizes the capabilities of the opensource library OpenCASCADE and pythonOCC module respectively for the re-generation of the CAD models directly from the optimized numerical models as a result of the procedure.

[1]Chair of Structural Analysis, Technical University of Munich, München, Germany

***Corresponding author**: altug.emiroglu@tum.de

Introduction

In practice, it is often necessary to bring multi-disciplinary, multi-objective optimization problems or their combinations to a common "platform", in order to advance the design towards the optimum in a consistent and collaborative manner. The problem at this stage stems from the possible non-matching discretizations of the numerical models and of the sensitivity fields. It was initially proposed in [4] that the Mortar operators could be utilized to bridge this gap and that they exhibit certain sensitivity filtering properties similar to the Vertex-Morphing method [3]. The method was demonstrated on the shape optimization of a fluid-structure interaction problem. This study concentrates on the possibility of using the underlying CAD models as a basis for optimization when the link between the discrete analysis models and the smooth CAD geometries parametrized by trimmed multi-patch NURBS shape functions are constructed via a Mortar type mapping operator. The discussed type of linking not only enables directly optimizing the CAD model at hand but it also acts as a sensitivity filtering which is evidently necessary when free form shape optimization problems with rich design spaces are of interest. The method belongs to the so called explicit filtering family of methods which do not require embedding the sensitivity filtering into the sensitivity analysis phase. Thus, it is modular and easy to adopt in existing optimization workflows. Furthermore, the proposed method makes it possible to reconstruct the CAD models either as a by-product of the optimization or could be applied as a final step for the CAD reconstruction of the optimum discrete geometry. Moreover, the proposed method is combined with the capabilities of the OpenCASCADE Community Edition [6, 7] for handling commercial CAD file formats such as STEP, IGES as well as newly emerging JSON type formats. In addition, the pythonOCC project [8] is utilized to enable the preprocessing for the generation of the mentioned linking between CAD and discrete models.

1 The Optimization Problem

Consider the following generic optimization problem:

$$\min_{\mathbf{x}} \quad J\left(\mathbf{x}, \mathbf{u}\right) \ , \tag{1}$$

where J denotes the objective function, whereas $\mathbf{x}$, $\mathbf{u}$ refer to the physical surface coordinates, i.e. the design variables, and the state variables, respectively. Explicit sensitivity filtering methods such as [3] transfer design variables of the above optimization problem to a shape control field without impairing the optimality condition. This field is referred as $\mathbf{s}$ throughout this paper. While it is possible to define various variable transformation methods, in this study it is favoured to employ a Mortar based transformation. Moreover, parametrization of the shape control field is realised by the NURBS shape functions of the underlying trimmed multi-patch CAD geometry. Therefore, the shape control parameters $\mathbf{s}$are the physical coordinates of the NURBS control points.

1.1 Variable Transformation Through Mortar Mapping

For instance, the underlying parametrization of the CAD model could be utilized as a basis for the variable transformation and as a shape control field. In this study, the NURBS shape functions of the CAD model are favoured for their wide applicability in CAGD programs as well as in numerical computational fields such as isogeometric analysis. The Mortar method can be defined as an additional minimization problem as in [2] to be solved besides the optimization problem. Thus, the objective function is augmented with the minimization of distance between the increments of the design and the shape control fields by modifying the optimization problem as follows;

$$\min_{\mathbf{s}} \quad \tilde{J} := J + \left\langle \Delta\mathbf{x} - \Delta\mathbf{s}, \Delta\mathbf{x} - \Delta\mathbf{s} \right\rangle_{0,\Omega_i} \ , \tag{2}$$

where the employed inner product is the norm of the $L^2\left(\Omega\right)$ space:

$$\left\langle \mathbf{u}, \mathbf{v} \right\rangle_{0,\Omega} = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \ \mathrm{d}\Omega \ . \tag{3}$$

The sensitivity analysis in this context requires the computation of the objective function's variation:

$$\delta J = \cancelto{0}{\frac{\partial J}{\partial \mathbf{s}}} \delta\mathbf{s} + \frac{\partial J}{\partial \mathbf{x}} \delta\mathbf{x} + 2 \int_{\Omega_i} \left(\delta\Delta\mathbf{x} - \delta\Delta\mathbf{s}\right) \cdot \left(\Delta\mathbf{x} - \Delta\mathbf{s}\right) \mathrm{d}\Omega \ , \tag{4}$$

$$\delta J = \frac{\partial J}{\partial \mathbf{x}} \delta\mathbf{x} + 2 \int_{\Omega_i} \left(\delta\Delta\mathbf{x} - \delta\Delta\mathbf{s}\right) \cdot \left(\Delta\mathbf{x} - \Delta\mathbf{s}\right) \mathrm{d}\Omega \ . \tag{5}$$

In Equation 5, it is necessary to set the second term to zero, in order to ensure the equality of the sensitivity derivatives of the original problem and the augmented problem. Discretizing and setting the second term to zero gives:

$$\int_{\Omega_i} \left(\delta\Delta\mathbf{s} \cdot \Delta\mathbf{s} + \delta\Delta\mathbf{x} \cdot \Delta\mathbf{x} - \delta\Delta\mathbf{s} \cdot \Delta\mathbf{x} - \delta\Delta\mathbf{x} \cdot \Delta\mathbf{s}\right) \mathrm{d}\Omega = 0 \ , \tag{6}$$

$$\begin{aligned} &\int_{\Omega_i} \left(\delta\Delta\hat{\mathbf{s}}^T \mathbf{N}_s^T \mathbf{N}_s \Delta\hat{\mathbf{s}} - \delta\Delta\hat{\mathbf{s}}^T \mathbf{N}_s^T \mathbf{N}_x \Delta\hat{\mathbf{x}}\right) \mathrm{d}\Omega + \\ &\int_{\Omega_i} \left(\delta\Delta\hat{\mathbf{x}}^T \mathbf{N}_x^T \mathbf{N}_x \Delta\hat{\mathbf{x}} - \delta\Delta\hat{\mathbf{x}}^T \mathbf{N}_x^T \mathbf{N}_s \Delta\hat{\mathbf{s}}\right) \mathrm{d}\Omega = 0 \ , \end{aligned} \tag{7}$$

where $\mathbf{N}_s$ and $\mathbf{N}_x$ refer to the shape function matrices that are used for discretizing the shape control and the design fields respectively. By enforcing the integrals in Equation 7 individually to be zero and using the abbreviation $\mathbf{C} = \int_\Omega \mathbf{N}^T\mathbf{N}\mathrm{d}\Omega$:

$$\delta\Delta\hat{\mathbf{s}}^T\mathbf{C}_{ss}\Delta\hat{\mathbf{s}} = \delta\Delta\hat{\mathbf{s}}^T\mathbf{C}_{sx}\Delta\hat{\mathbf{x}}\ , \tag{8}$$

$$\Delta\hat{\mathbf{s}} = \mathbf{C}_{ss}^{-1}\mathbf{C}_{sx}\Delta\hat{\mathbf{x}}\ , \tag{9}$$

$$\Delta\hat{\mathbf{s}} = \mathbf{A}_{sx}\Delta\hat{\mathbf{x}}\ . \tag{10}$$

$$\delta\Delta\hat{\mathbf{x}}^T\mathbf{C}_{xx}\Delta\hat{\mathbf{x}} = \delta\Delta\hat{\mathbf{x}}^T\mathbf{C}_{xs}\Delta\hat{\mathbf{s}}\ , \tag{11}$$

$$\Delta\hat{\mathbf{x}} = \mathbf{C}_{xx}^{-1}\mathbf{C}_{xs}\Delta\hat{\mathbf{s}}\ , \tag{12}$$

$$\Delta\hat{\mathbf{x}} = \mathbf{A}_{xs}\Delta\hat{\mathbf{s}}\ . \tag{13}$$

Equations 10 and 13 define mapping operators from the discrete design field to the discrete control field and vice versa, respectively.

Having the additional minimization problem solved, one can transfer the sensitivity derivatives of the objective function with respect to the design, to the shape control field and back:

$$\delta J = \underbrace{\mathbf{A}_{xs}\,\underbrace{\mathbf{A}_{sx}\frac{\partial J}{\partial\hat{\mathbf{x}}}}_{\mathrm{d}J/\mathrm{d}\hat{\mathbf{s}}}}_{\partial J/\partial\hat{\mathbf{x}}^*}\,\delta\hat{\mathbf{x}}\ . \tag{14}$$

Above operations map the rugged sensitivity field $\partial J/\partial\hat{\mathbf{x}}$ onto the shape control field and back to the design field to form the filtered sensitivity field $\partial J/\partial\hat{\mathbf{x}}^*$. This paves the way for two possible uses:

1. One can apply the mapping operators individually by firstly mapping the design field sensitivities onto the shape control field sensitivities and have the possibility to drive the optimization process directly on the CAD model. Then, the forward mapping operator is used to regenerate the discrete numerical analysis model. This enables the instantaneous updates of the CAD model throughout the optimization process.

2. Alternatively, the mapping operators can be used consecutively to retrieve the filtered sensitivity field and the optimization process can be driven on the discrete numerical model.

1.2 Patch Coupling Constraints and Geometrical Constraints of the Optimization

Unlike the finite element discretizations where the nodal degrees of freedom could be coupled strongly, multi-patch NURBS geometries often have non-matching discretizations along their coupling interfaces. In practice, it is observed that this aspect causes discontinuous control field deformations and their derivatives -namely the rotations- across the patch interfaces. In order to overcome this, a quadratic penalty based method is adopted in the following form similar to [2]. Here only the continuity of the shape control field is considered for brevity. One can refer to [1, 2] for further details on the rotational coupling. Now the augmented objective function with Mortar method and the patch coupling constraints reads:

$$\begin{aligned}\min_{\mathbf{s}} \quad \tilde{J} &:= J + \langle\Delta\mathbf{x}-\Delta\mathbf{s},\Delta\mathbf{x}-\Delta\mathbf{s}\rangle_{0,\Omega_i} \\ &\quad + \alpha_{\Gamma_{ij}}\langle\Delta\mathbf{s}_i-\Delta\mathbf{s}_j,\Delta\mathbf{s}_i-\Delta\mathbf{s}_j\rangle_{0,\Gamma_{ij}}\ ,\end{aligned} \tag{15}$$

where Γ_{ij} refers to the common interface that bounds the domains Ω_i and Ω_j, whereas $\Delta\mathbf{s}_i$ and $\Delta\mathbf{s}_j$ refer to the shape control increments at the domain boundaries along this interface. In addition, the penalty based formulation can be extended to incorporate the defined geometrical equality constraints on the parts of the domain regardless of being a portion of a domain boundary $\Gamma_k \subseteq \Gamma_i$ or a portion of a domain $\Omega_k \subseteq \Omega_i$:

$$\begin{aligned}\min_{\mathbf{s}} \quad \tilde{J} := J &+ \langle \Delta\mathbf{x} - \Delta\mathbf{s}, \Delta\mathbf{x} - \Delta\mathbf{s} \rangle_{0,\Omega_i} \\ &+ \alpha_{\Gamma_{ij}} \langle \Delta\mathbf{s}_i - \Delta\mathbf{s}_j, \Delta\mathbf{s}_i - \Delta\mathbf{s}_j \rangle_{0,\Gamma_{ij}} \\ &+ \alpha_{\Gamma_k \subseteq \Gamma_i} \langle \Delta\mathbf{s}_k, \Delta\mathbf{s}_k \rangle_{0,\Gamma_k \subseteq \Gamma_i} \\ &+ \alpha_{\Omega_k \subseteq \Omega_i} \langle \Delta\mathbf{s}_k, \Delta\mathbf{s}_k \rangle_{0,\Omega_k \subseteq \Omega_i} \quad . \end{aligned} \tag{16}$$

Following the procedure in equations 4-8, one can find the contributions of the coupling conditions as well as the geometrical constraints to the mapping operators:

$$\delta\Delta\hat{\mathbf{s}}^T \underbrace{\left[\mathbf{C}_{ss} + \mathbf{C}_{ss}^{\Gamma_{ij}} + \mathbf{C}_{ss}^{\Gamma_k \subseteq \Gamma_i} + \mathbf{C}_{ss}^{\Omega_k \subseteq \Omega_i}\right]}_{\mathbf{C}_{ss}^*} \Delta\hat{\mathbf{s}} = \delta\Delta\hat{\mathbf{s}}^T \mathbf{C}_{sx} \Delta\hat{\mathbf{x}} \ , \tag{17}$$

$$\Delta\hat{\mathbf{s}} = \mathbf{C}_{ss}^{*-1} \mathbf{C}_{sx} \Delta\hat{\mathbf{x}} \ , \tag{18}$$

$$\Delta\hat{\mathbf{s}} = \mathbf{A}_{sx}^* \Delta\hat{\mathbf{x}} \ . \tag{19}$$

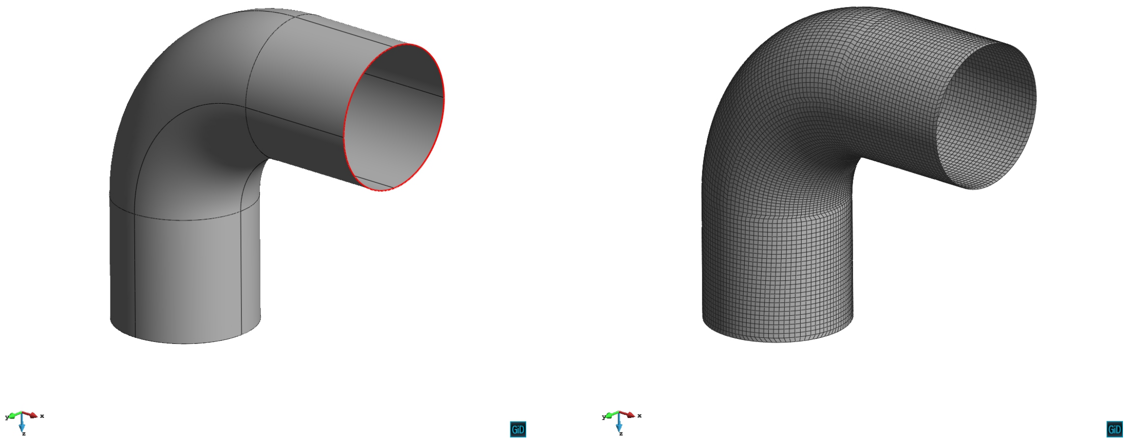
In Equation 17 $\mathbf{C}_{ss}^{\Gamma_{ij}}$, $\mathbf{C}_{ss}^{\Gamma_k \subseteq \Gamma_i}$ and $\mathbf{C}_{ss}^{\Omega_k \subseteq \Omega_i}$ collect the contributions related to the patch coupling conditions and the geometrical constraints of the optimization problem, respectively. By doing so, the equality constraints of the optimization problem are also embedded into the sensitivity filtering operator. Finally, the enhanced variable transformation matrix $\mathbf{A}_{sx}^*$ is computed. It is important to note that neither patch coupling conditions nor the equality constraints of the optimization problem need to be applied on the design field, since they are already considered in the shape control field. Thus, the shape updates follow the constrained shape control field.

2 OpenCASCADE and pythonOCC

Processing commercial CAGD formats and generating computational models require a robust CAD Kernel. OpenCASCADE and pythonOCC offer a great opportunity for this purpose. While OpenCASCADE is a reliable open-source software library, pythonOCC makes the usage, code handling and visualization straightforward. Thus, the implemented method was made available to external usage via a software library and combined with the capabilities of these tools. This combination allows working with commercial formats, preprocessing generic CAD geometries for optimization tasks e.g. defining constraints, and visualizing the results as well as exporting optimized CAD models.

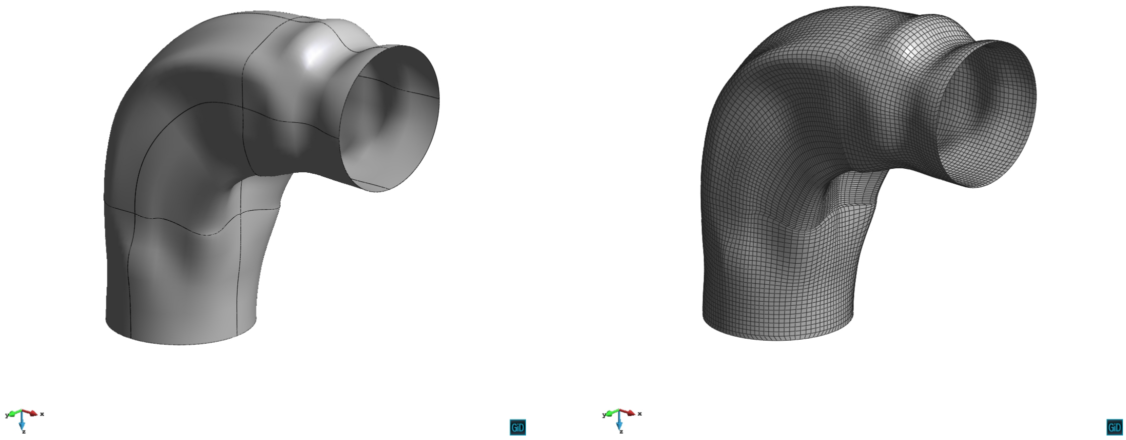
3 Results: Optimization and Regeneration of the CAD Models

As a demonstration case, the strain energy minimization of an elbow pipe was considered. The initial CAD geometry and the computational mesh for the structural as well as the sensitivity analysis can be seen in Figure 1. The structure is loaded in the global z-direction (vertically with respect to the figure) at its red marked end as in Figure 1a and simply supported at its other end. The optimization problem consists of equality constraints that restrict deformations of the supported and the loaded edges. The mapping operator between the computational mesh and the CAD geometry was constructed via the Mortar method. The mapping operator was enhanced with the patch coupling constraints and the equality constraints of the optimization problem. The control points of the CAD geometry were defined as optimization variables and the steepest descent algorithm was adopted for the solution of the optimization problem. During the optimization iterations, it could be observed that the nonsmooth sensitivity fields were smoothened by the variable transformation through the Mortar Mapping operator. As a result, the sensitivity filtering was achieved and the premature failure of the optimization iterations due to infeasible shape updates was avoided. In addition, employed method facilitated the reconstruction of the CAD geometry as a very beneficial by-product of the optimization.



(a) CAD geometry **(b)** Computational mesh

Figure 1. Initial configuration



(a) CAD geometry **(b)** Computational mesh

Figure 2. Optimized configuration

Conclusions

The method to construct a sensitivity filtering operator through Mortar Mapping method is extended to bring CAD and discrete numerical analysis models together. The method facilitates generation of CAD models as a by-product of the optimization iterations. Moreover, due to its explicit nature, it could be utilized to robustly regenerate CAD models making use of the geometrical changes such as deformations or shape updates of the final discrete numerical models. The implementation of the method is accomplished through the open-source project EMPIRE [5]. In order to extend the applicability of the method in existing optimization workflows, the functionalities of the mapping technique are made available as a software library. In order to enhance the versatility and applicability in industrial optimization problems, OpenCASCADE and pythonOCC are utilized. This enables flexible and automated definition of the patch coupling conditions and optimization constraints.

Acknowledgements

The author Altuğ Emiroğlu would like to thank the German Federal Ministry for Economic Affairs and Energy for funding this research.

References

[1] A. Apostolatos, M. Breitenberger, R. Wüchner, and K.-U. Bletzinger. *Isogeometric Analysis and Applications 2014*, chapter Domain Decomposition Methods and Kirchhoff-Love Shell Multipatch Coupling in Isogeometric Analysis, pages 73–101. Springer International Publishing, Cham, 2015.

[2] A. Apostolatos, A. Emiroglu, F. Pean, R. Wüchner, and K.-U. Bletzinger. An isogeometric mortar surface coupling method for trimmmed multipatch cad geometries with application to fsi. In *7th GACM Colloquium on Computational Mechanics*. Universität Stuttgart, Forthcoming 2017.

[3] K.-U. Bletzinger. A consistent frame for sensitivity filtering and the vertex assigned morphing of optimal shape. *Structural and Multidisciplinary Optimization*, Vol. 49(6):p. 873–895, 2014.

[4] A. Emiroğlu, R. Wüchner, and K.-U. Bletzinger. *AeroStruct - Enable and Learn How to Integrate Flexibility in Design*, chapter Treating Non-Conforming Sensitivity Fields by Mortar Mapping and Vertex Morphing for Multi-Disciplinary Shape Optimization. Springer International Publishing, Forthcoming 2017.

[5] Lehrstuhl für Statik, TUM. Enhanced Multi Physics Interface Research Engine - EMPIRE. http://www.empire-multiphysics.com/.

[6] OpenCASCADE. OpenCASCADE Technology. http://www.opencascade.com/.

[7] T. Paviot. OpenCASCADE Community Edition. https://github.com/tpaviot/oce/.

[8] T. Paviot and J. Peringa. pythonOCC - 3D CAD for python. http://www.pythonocc.org/.