



Morphing boxes for the integration of shape optimization in the product design process

Alexander Brune , Thiago Weber Martins  and Reiner Anderl 

Technische Universität Darmstadt, Germany

ABSTRACT

This paper proposes an approach to integrate sensitivity-based shape optimization into the process of product design. The approach is based on B-Spline volume morphing boxes as an interface between the CAD model and the FE model used for optimization. Existing tools within a CAD system are used to emulate B-Spline morphing boxes. In order to use morphing boxes for sensitivity-based shape optimization, a Python program is implemented producing morphing box data that is passed to the optimization software. A consistent workflow is devised that starts with an initial CAD model and ends with a CAD model that has been updated based on optimization results. The workflow is validated on an industrial example.

KEYWORDS

Shape optimization;
parameterization;
computer-aided design

1. Introduction

Shape optimization software provides valuable input during the design of mechanical product parts. Employing mathematical algorithms, it methodically and efficiently improves an existing product's design geometry with respect to user-defined criteria. Typical examples include weight minimization or the maximization of a stiffness measure under weight restrictions. The main obstacle in utilizing shape optimization results often lies in the fact that these are mesh-based, as the algorithms commonly manipulate a given Finite Element (FE) model. Ultimately, they should be applied to a Computer-Aided Design (CAD) model. One basic strategy to achieve this, which is employed in the industry, is a conversion of the optimized FE mesh surface to a new CAD model, for example, by approximating the former by an IGES wire grid model or an STL model. By replacing the original CAD model, however, the underlying construction logic is lost, impeding further design modeling following the optimization.

A more promising approach is to bridge the gap between FE mesh-based optimization and CAD by applying the same method for parameterizing shape changes to both. Various methods of geometry parameterization for shape optimization have been proposed [9]. The most notable is the use of Isogeometric Analysis

(IGA) in which the same functions that define the geometry in the CAD model – most commonly NURBS – are used as basis functions in the FE formulation. IGA was introduced in [3] and has been extensively studied in recent years regarding its application in shape optimization, see, e.g., [14]. While IGA seems promising for this and other purposes, it has yet to be established in the commercial and industrial setting. With a focus on practical applicability in the industry, so-called morphing boxes are employed in this paper, constituting an interface between commercial CAD software and optimization tools. A consistent workflow is devised, starting with a CAD model, and concluding with the application of shape changes to this model. These changes are based on the results of sensitivity-based shape optimization, performed on the FE discretization of the model and incorporating morphing boxes. While the workflow also requires the use of morphing boxes within the CAD environment, current software lacks this functionality so that it is emulated with existing tools. The main goal is not to provide a full implementation of new software tools for CAD and optimization systems but rather to give a proof of concept, making a case for morphing boxes as a step towards the full integration of shape optimization into the industrial product design process in the near future.

CONTACT Alexander Brune  alexander.brune@gmail.com; Thiago Weber Martins  weber@dik.tu-darmstadt.de;
Reiner Anderl  anderl@dik.tu-darmstadt.de

2. Morphing boxes for shape optimization

2.1. Background

2.1.1. Shape optimization

A shape optimization problem is a special type of structural optimization problem in which only the surface of the product design is optimized [5]. It is most commonly applied to improve pre-existing part geometries rather than to obtain a completely new geometry from scratch. Moreover, the optimization is quite often limited to a sub-region of the surface, as certain parts of the product design should not change due to manufacturing or design constraints. A shape optimization problem is typically defined mathematically as

$$\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x}), \text{ such that } g_i(\vec{x}) \leq 0, \quad i = 1, \dots, m. \quad (1)$$

Here, f is called the objective function and g_i are the constraint functions. The components of \vec{x} are the design variables. These variables parameterize the product design. Since effective shape optimization software commonly works with an FE model, the design variables are defined so as to determine the displacement of mesh nodes. Simulia Tosca Structure, which is the software used for optimization in this paper, defines a design variable as the displacement value of a “design node” which is a mesh surface node allowed to move during the optimization. To iteratively find a solution to (1), the sensitivity-based Method of Moving Asymptotes (MMA) is employed [12]. The term “sensitivity-based” indicates that the sensitivities, i.e., the derivatives of f and g_i with respect to \vec{x} are used in an algorithm, which, for the general case, gives it better convergence properties than, e.g., a heuristic algorithm. The sensitivities are computed internally by the software. The termination condition for the optimization is given by a predefined maximum number of iterations.

As mentioned in the introduction, a central issue of shape optimization, when it comes to incorporating it into the product development process, is that the results should be available in the form of a CAD model. In the chain of subsequent processes needed for optimizing a pre-existing CAD model, there are two occurrences of model conversion. The first is the setup of an FE model based on the CAD model, which is a standard procedure. This FE model is passed to the optimization tool. The second model conversion is that of the optimized FE model back to a CAD model. This last task is the most challenging and the one that is addressed in this paper. Preferably, the optimization results should be integrated into the existing CAD model instead of replacing it by a completely new one.

2.1.2. B-Spline morphing boxes

Morphing methods have their origin in computer graphics but have applications in various fields which require the modeling of geometric changes [10]. In the context of shape optimization, they either serve the purpose of preserving FE mesh quality, see for example [11], or that of shape parameterization [2]. It is this second application which is of interest in this paper. In particular, morphing boxes, the concept of which was first proposed by Perry et al. in [6], are used to define design changes in shape optimization. Various other approaches are listed in [9]. The main idea of morphing boxes has been explained in Perry et al. [6] based on [8]:

“Consider a cube of clear, flexible plastic, in which several objects have been embedded. The embedded objects have the same degree of flexibility as the cube. As the plastic cube is deformed, the embedded objects are also deformed in an intuitive manner.”

The main contribution of this work is to use morphing boxes to parameterize the part of the product design geometry that is to be modified within the optimization – in the FE model as well as in the CAD model. In this way, they provide an interface by which the FE mesh-based optimization results are applied to the CAD model. In the optimization step, the morphing box parameterization is coupled to a sensitivity-based optimization algorithm, which is crucial when considering efficiency.

In this work, morphing boxes have been implemented as B-Spline volumes, as has previously been done in [4]. More specifically, they are defined by

$$\begin{aligned} [0, 1]^3 \ni (r, s, t) &\mapsto \vec{V}(r, s, t) \\ &:= \sum_{i=0}^I \sum_{j=0}^J \sum_{h=0}^H \vec{c}_{i,j,h} B_{i,k}(r) B_{j,l}(s) B_{h,m}(t), \\ I, J, H &\geq 1, \end{aligned} \quad (2)$$

where $B_{i,k}$ is the i -th B-Spline basis function of order k for a uniform, open knot-vector spanning the interval $[0, 1]$ with no repeating inner knots. For details on B-Splines, see [7]. The 3-dimensional grid of control points $\vec{c}_{i,j,h}$, herein called control polygon, defines the morphing box volume and thus the shape of the embedded object – which, in this case, is the design to be optimized. The morphing box’s initial geometry is not restricted to a cube, but is flexible, and is in the following merely assumed not to be self-intersecting or singular. Continuity constraints at the box’s side faces are imposed by fixing the corresponding layers of control points. An illustrative example of a morphing box is shown in Fig. 1.

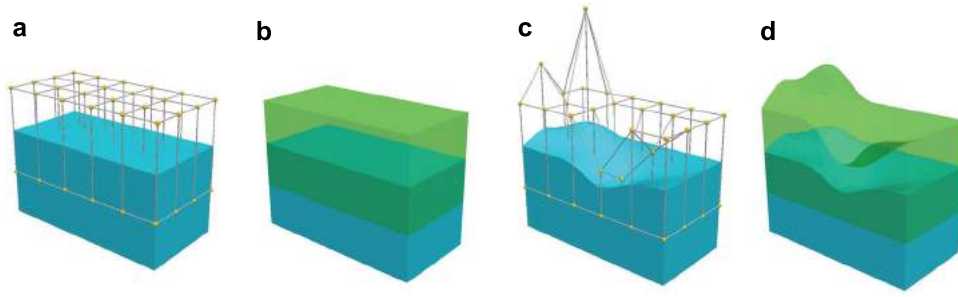


Figure 1. Simple morphing box (green), defined by a control polygon (gray), for manipulating a simple design geometry (blue): (a) and (b): Initial morphing box and design; (c) and (d): Changed control polygon, resulting morphing box and morphed design.

2.2. Morphing boxes

2.2.1. Morphing boxes for shape optimization

To realize the embedding of the FE model in a B-Spline volume morphing box for shape optimization, the control points $\vec{c}_{i,j,h}$ are now defined as the design variables. Design nodes are re-defined as all nodes within a morphing box, including inner nodes of the mesh, and their dependency on $\vec{c}_{i,j,h}$ has to be determined. Let \vec{X}_e denote the position vector of the e -th design node. This node is then defined as a point within the morphing box and local coordinates $(r_e, s_e, t_e) \in [0, 1]^3$ need to be determined which satisfy

$$\begin{aligned} \vec{X}_e &= \vec{V}(r_e, s_e, t_e) \\ &= \sum_{i=0}^I \sum_{j=0}^J \sum_{h=0}^H \vec{c}_{i,j,h} B_{i,k}(r_e) B_{j,l}(s_e) B_{h,m}(t_e). \end{aligned} \quad (3)$$

A modified Newton's method, with the solution space restricted to $[0, 1]^3$, determines the local coordinates (r_e, s_e, t_e) for each FE node, if they exist, and thus locates all design nodes, i.e., all FE nodes lying inside a morphing box. The algorithm has been implemented in Python. In the case of multiple morphing boxes, this locating algorithm is applied once for each morphing box. Eqn. (3) shows that the position of each design node depends linearly on the control points $\vec{c}_{i,j,h}$. Hence, it collects all design node position vectors into one vector \vec{x} and the position vectors of all control points into another vector \vec{p} . The former is defined as a function of the latter in the form of a matrix-vector-multiplication $\vec{x} = \vec{x}(\vec{p}) = M\vec{p}$, with M being a $3N_{dn} \times 3N_{cp}$ matrix and in all practical cases sparse. N_{dn} and N_{cp} denote the number of design nodes and control points, respectively. The previous considerations also apply to the case of several disjoint morphing boxes.

Since the goal is to optimize the control point positions instead of the design nodes directly, the optimization problem in Eqn. (1) is now replaced by the transformed

optimization problem

$$\begin{aligned} \min_{\vec{p}} \tilde{f}(\vec{p}) &= f(\vec{x}(\vec{p})), \text{ such that } \tilde{g}_i(\vec{p}) = g_i(\vec{x}(\vec{p})) \leq 0, \\ &i = 1, \dots, m. \end{aligned} \quad (4)$$

As mentioned earlier, efficient optimization software, such as Tosca Structure, employs sensitivity-based optimization algorithms. The sensitivities of the objective and constraint functions with respect to the new design variables in \vec{p} are found using the chain rule of differentiation:

$$\frac{df}{d\vec{p}} = \frac{df}{d\vec{x}} \frac{d\vec{x}}{d\vec{p}} = \frac{df}{d\vec{x}} M.$$

While the sensitivities with respect to the FE-mesh design nodes \vec{x} are computed by the optimization tool, the transformation matrix M is calculated by the same Python program that handles the location of the design nodes. For a given FE-mesh and a set of morphing boxes, this program writes out M as an ASCII text file and inserts additional information into the FE input file, both of which are read in by optimization tool.

Instead of operating directly on the FE design nodes, the optimization algorithm optimizes the positions of control points $\vec{c}_{i,j,h}$. After each iteration, the FE mesh is updated by computing the new design node positions using Eqn. (3). The final optimization solution is defined solely by the final control point positions, since the design node positions, and thus the geometry of the optimized FE model, follow directly from Eqn. (3).

2.2.2. Morphing boxes in the CAD system

State of the art CAD systems do not offer a morphing box tool as described above. Therefore, it has to be emulated by using a series of existing tools. A consistent workflow has been derived which closely mimics the functionalities of a B-Spline volume morphing box, but which imposes certain restrictions on these functionalities. The utilized tools and the restrictions they entail are laid out below.

The CAD software used in this paper is CATIA V6. It is not possible for the user to create a B-Spline volume

by specifying a three-dimensional control point grid in CATIA V6. B-Spline surfaces, if not B-Spline volumes, are widely used for the internal representation of part models in CAD systems. However, they cannot be drawn upon either, since the user cannot create them directly by specifying the control polygon. Therefore, subdivision surfaces [1] are used instead to model the top and bottom faces of the morphing box volume. These surfaces lend themselves to this application because they can be defined by the user through the specification of structured, two-dimensional grids of control points, similarly to B-Spline surfaces. Furthermore, they are easily set up from data created outside of and imported into the CAD system. Note that the use of these surfaces is not at all mandatory but is one possible means of emulating morphing boxes for a proof of concept.

To clearly differentiate, let V denote the B-Spline volume which defines the actual – but theoretical – morphing box, while $S = \{S_{top}, S_{bottom}\}$ is the set of subdivision surfaces which are used to emulate the morphing box in the CAD environment, and V_S is the volume created by a projecting translation from S_{top} to S_{bottom} . S and V_S are illustrated in Fig. 2 (a)-(b). Subdivision surfaces are defined by 2-dimensional grids of control points. While not mathematically identical, a subdivision surface closely approximates a bi-quadratic B-Spline surface defined for the same set of control points, given a reasonable control point density. Taking two such bi-quadratic B-Spline surfaces, a top and bottom face, as it were, one can define a B-Spline volume as a linear interpolation of these two surfaces. Accordingly, the polynomial orders of the basis functions for this B-Spline volume are $k = l = 2$, $m = 1$ (cf. Eqn. (2)). The morphing box volume V is now chosen to be just such a B-Spline volume. As a result, it is defined by the same set of control points as S . Note that $V = V_S$ holds if the volumes are rectangular parallelepipeds. In general, V_S is an approximation of V .

The next task is to demarcate $\vec{c}_{i,j,h}$ the subdomain of the product which is embedded in the morphing box. In the following, this subdomain is called *design domain*, denoted by D . Let P be the domain of the

whole product, then D is determined by the intersection $D = P \cap V = P \cap V_S$. The second equality holds because the initial morphing box is always a rectangular parallelepiped, which will be motivated momentarily. Now the actual morphing needs to be carried out, i.e., the design domain needs to be deformed along with the morphing box, which results in a morphed design domain \tilde{D} . For this purpose, assume some of the control points $\vec{c}_{i,j,h}$ have been moved to create the new set of control points $\vec{\tilde{c}}_{i,j,h}$, defining a deformed morphing box denoted by \tilde{V} , a new set of subdivision surfaces \tilde{S} , and a new \tilde{V}_S .

Morphing the design domain is done in two steps: extracting deformation data representing the map $V \rightarrow \tilde{V}$ and using this data to morph D . The “Deviation analysis” tool computes a displacement field for two surfaces, which are taken to be the same object in different states of deformation. The input used for this tool are the two sets of subdivision surfaces, S and \tilde{S} – not the boundary surfaces of \tilde{V} and \tilde{V}_S , as one might assume, since this may lead to a misinterpretation of the deformation. This is exemplified in Fig. 2 (c)-(d): A dent is created in the volume V by moving two control points of the top surface downwards, orthogonally to said surface. Accordingly, the displacement field for this deformation should be a field of vectors extending from the original top surface S_{top} to the deformed top surface \tilde{S}_{top} , each vector orthogonal to S_{top} . Contrarily, the “deviation analysis” misinterprets the deformation, attributing it in part to a denting of the right side surface.

The displacement field is used as input for the “digitized morphing” tool which applies the deformation to D . This yields a morphed design domain \tilde{D}_S . The subscript S indicates that it is not mathematically identical to \tilde{D} , which would be the result if using an actual B-Spline volume morphing box, but an approximation of it. When used in the described sequence, the CAD tools constitute a map $D \rightarrow \tilde{D}_S$. While not identical to $D \rightarrow \tilde{D}$, it needs to be geometrically consistent with it. To guarantee that the deformation $V \rightarrow \tilde{V}_S$ is appropriately interpreted and applied to D , further restrictions must be adhered to: Firstly, the initial morphing box must be a

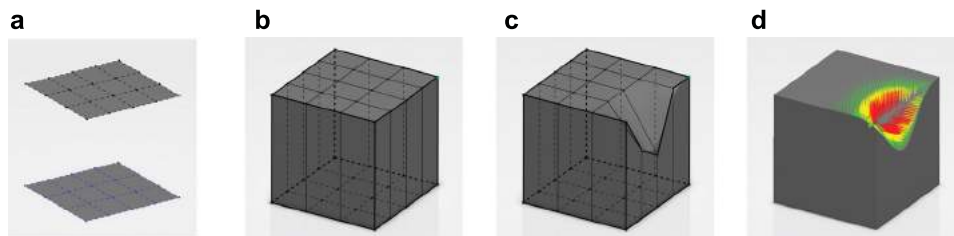


Figure 2. The subdivision surfaces and volume used in the morphing box workflow; a): $S = \{S_{top}, S_{bottom}\}$, b): V_S , c): dented V_S , d): inconsistent “deviation analysis” result.

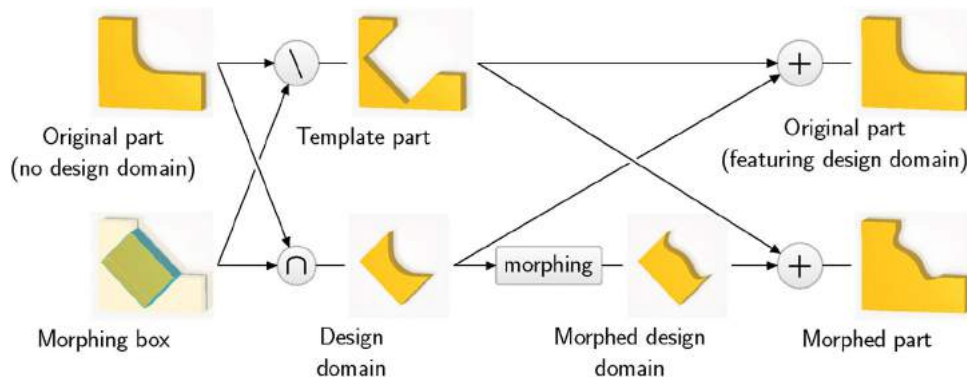


Figure 3. Defining the design domain as a morphable feature of the part by the “cut-and-replace” approach.

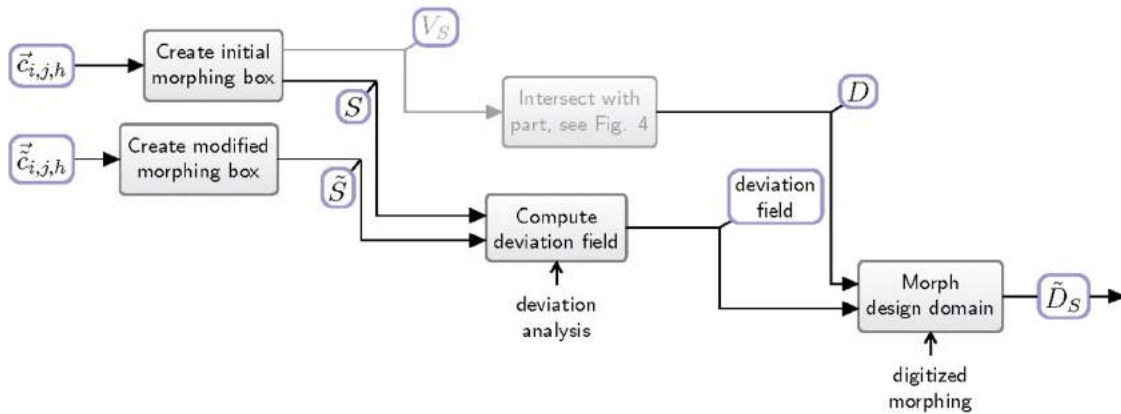


Figure 4. Structured Analysis and Design Technique (SADT) diagram of the morphing box workflow in the CAD environment.

rectangular parallelepiped. Secondly, control points may only be moved orthogonally to their initial plane. This is to avoid ambiguity of the input passed to the “deviation analysis” tool.

Before the thus-implemented morphing operation is performed, D is cut out of the product by Boolean intersection. Morphing then produces \tilde{D}_S which is inserted in its stead. This “cut-and-replace” procedure, illustrated in Fig. 3, ensures that the design domain defined by the morphing box is just one of many features of the product design, which can be morphed, replaced or removed without otherwise interfering with the part’s construction logic. Recall the hierarchical structure of a CAD part’s construction logic, which resembles a logical tree structure. The succession of binary topological and morphing operations of the cut-and-replace approach constitute a sub-tree at the lowest level of this tree. Any changes to this sub-tree do not affect the rest of it – that is, any region of the part model outside of the design domain. If the sub-tree is deleted, the original part is restored.

Now that the cut-and-replace approach has been addressed, the overall procedure that constitutes the morphing operation can be summarized. It is modeled as a process chain using the Structured Analysis and Design

Technique (SADT) and illustrated in Fig. 4., which shows the succession of morphing box definition, deviation analysis, and digitized morphing, in parallel to the cut-and-replace approach. The definition of process chains in the context of computer-aided tools and the product development process is the structured depiction of the flow of information, including its generation, processing and exchange. They are described in more detail in [13]. The diagram in Fig. 4 underlines that two sets of control points are ultimately all the input that is required for the workflow, one set defining the initial morphing box, the other the deformed morphing box.

In addition to specifying the control points interactively within in the CAD environment, morphing box definitions can be imported into the CAD system in the Wavefront OBJ format [3]. For this, the surfaces S_{top} and S_{bottom} are each defined as a two-dimensional polygonal grid in the OBJ file and are imported as subdivision surfaces in CATIA.

2.3. Process chain CAD-FE-SO

So far, morphing boxes have been discussed with regard to their use within the individual processes of CAD

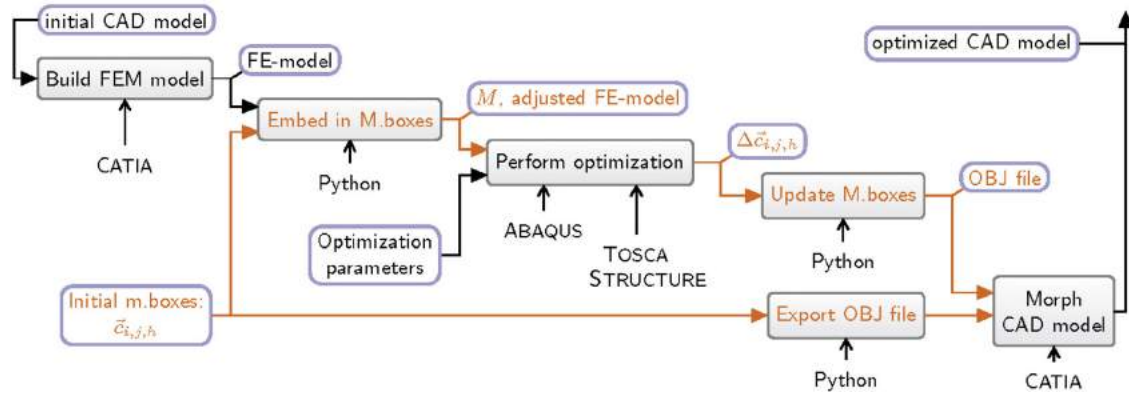


Figure 5. SADT diagram for the process chain CAD-FEA-SO.

and shape optimization (SO). However, for their effective application within the product design process, it is not enough to consider them as single tools. Moreover, product design process is a complex and challenging task that requires an entire treatment of each design process step. Each of them is characterized by decision-making processes that highly influence the final product design regarding its design and functional properties. Therefore, it is essential to set up a consistent workflow by linking the CAD and SO processes that describes the transition from one to the next in the form of a process chain, illustrated as an SADT diagram in Fig. 5.

Initial product design requirements such as dimensions and material properties are captured within the CAD model. Further structural requirements such as loads and boundary constraints are considered while building the FE model. In addition, discretization details are defined and the geometry discretization is performed. Morphing boxes are defined in Python. They are used by the morphing box Python program, firstly to export them as an OBJ file to be read in by the CAD system, secondly to locate all design nodes in the FE model. The program writes out the transformation matrix M and inserts design node definitions into the FE model file because these are required by the shape optimization module of Tosca Structure in the next step. A file containing the definitions of the objective and constraint functions, as well as algorithm and other optimization parameters, is read in by Tosca Structure along with the FE model and M . The optimization is performed iteratively, employing an FE solver in each iteration. The solver used here is SIMULIA Abaqus. A text file is put out containing the optimization displacements $\Delta \vec{c}_{i,j,h}$ of the morphing box control points after the last iteration, such that the optimized morphing boxes' control points are given by $\vec{c}_{i,j,h} = \vec{c}_{i,j,h} + \Delta \vec{c}_{i,j,h}$. This file is read in by the morphing box Python program in order to update the morphing box definitions and write an OBJ file for these optimized

morphing boxes. Both this OBJ file and the one representing the initial morphing boxes are now imported into the CAD environment to perform the morphing workflow described in section 2.2.2. This yields the optimized CAD model of the product solution.

For the sake of completeness, it should be mentioned that the MMA optimization algorithm requires box constraints to be defined for each design variable. Hence, upper and lower limits for the optimization displacement of each control point are defined in ASCII files and parsed by Tosca Structure. This is omitted in Fig. 5.

It should be noted that some assumptions are made when claiming the digital integrity of this approach: If any modeling work is done after morphing has been performed on the design domain, this modeling should not interfere with any geometric details of the initial design domain. Changes that affect the design domain's original geometry prior to morphing will in general lead the morphing results to become inconsistent. The morphed design domain may be reshaped at will. This will generally invalidate the optimality of the morphed design domain. It might be less clear that this also applies to changes of the CAD model outside of the design domain. Because local shape changes affect the mechanical response of the overall part, the morphed design domain can in general no longer be assumed to be optimal if such changes are applied.

Recall that the purpose of the morphing is to change the shape of the design domain so that it is optimal with respect to criteria defined by the user and under the condition that the rest of the CAD model, the non-design domain, as it were, remains fixed. On the one hand, any changes to the CAD model after this optimization can therefore defeat its initial purpose. On the other hand, new insights gained by the user from the optimization results are a practical starting point for further modelling to improve the design or to better accommodate the optimization results. The proposed approach allows for this

case, since morphing data can be removed or updated. A new optimization should be performed for the most recent CAD model, either reusing the initial morphing boxes of the previous optimization if still applicable, defining new morphing boxes if not, or even defining a completely new second set of morphing boxes and using it on the already morphed CAD model. Note that a new FE model will need to be built for the optimization in either case, see Fig. 5.

2.4. Test case

In the following, the workflow as introduced in Fig. 5 is validated on a piston-engine connecting rod as depicted in Fig. 6(a). In this example, the volume v of the connecting rod has to be minimized. Quantities relating to material failure and stiffness are constrained. The FE model is geometrically linear and its material behavior is defined as isotropic, linearly elastic. Four load cases are defined for the rod, in which it is stretched along, compressed along, twisted around, and bent orthogonal to its longitudinal axis, respectively. Four critical system responses are constrained in the optimization: the maximum von-Mises stresses of the first and second load case, $\max \sigma_M^{(1)}$ and $\max \sigma_M^{(2)}$, respectively, the angle w of rotation around the longitudinal axis in the third load case, and the displacement u of the load point in the fourth load case.

The morphing box setup is illustrated in Fig. 6(b). Control points are fixed where necessary to maintain point-continuity or tangency in the product design. Maximum displacement values are defined for the control points in the optimization, but the results remain well within the allowed ranges.

For the optimization problem, the constraint functions are defined so as not to allow the above-mentioned system responses to increase by more than 20% of their value in the initial model. In an actual industrial

Table 1. Change in volume and critical system response values, after the shape optimization and after approximating the shape changes in CATIA V6. Values are relative to the values of the initial model.

	Design response	Tosca Structure morphing result	CATIA morphing result
Objective function:	v	-4.18%	-3.28%
Constraints:	$\max \sigma_M^{(1)}$	4.78%	-12.21%
	$\max \sigma_M^{(2)}$	20.00%	12.38%
	w	14.51%	13.02%
	u	20.00%	18.25%

optimization, a 20% increase of stress peak values or displacements would be considered unacceptable. However, geometric restrictions apply to the morphing boxes that are modeled in the CATIA V6 workflow, as mentioned in the previous subsection. Hence, the optimization needs to be granted a certain amount of leeway to produce a sufficient product design change to validate the overall approach by.

The morphing boxes and mesh after the optimization in SIMULIA Tosca Structure are shown in Figs. 6(c) and 6(d). The optimal morphing boxes are imported into CATIA V6 where they are used to morph the CAD model. For the thus optimized CAD model, an FE analysis is performed for validation. When applied to the CAD model by the proposed workflow, the design changes obtained by the optimization are only approximated. It is therefore necessary to compare the change in volume and system response values of the optimized CAD model with the Tosca optimization results. This is done in Table 1.

3. Conclusions and outlook

Using morphing boxes, FE mesh-based shape optimization results have successfully been applied to a CAD model. The feature- and history-based construction logic of the model is not overwritten and the original model

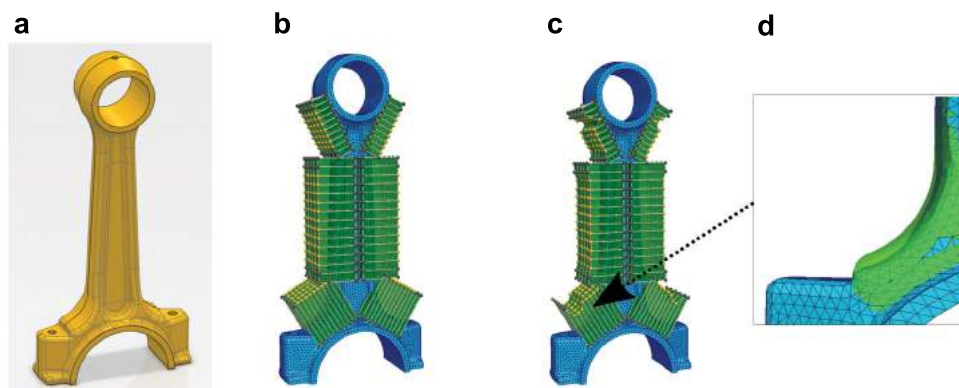


Figure 6. Connecting rod. (a): Initial CAD model; (b): Initial FE model and morphing boxes; (c): Optimized FE model and morphing boxes; (d) Close-up of the optimized lower left leg, with the initial shape super-imposed in green.

is easily restored, as the morphing box operation adds just another feature to the model that can be changed or removed. Furthermore, morphing boxes offer a quite intuitive way of marking out a design area. Also, continuity constraints are easily defined. The consistent workflow linking the CAD and SO processes integrates the resulting information into a product model in a way that improves the product design process. In order to reconstruct the optimized shape more accurately in the CAD model and to exploit the full potential of morphing boxes as a parameterization method, proper B-Spline volume morphing boxes should be implemented in the CAD environment. This would allow further studies to sound out the limits and possibilities of this approach for an integration of shape optimization into the design process. Additionally, an interactive morphing box tool would give the user a feeling for how the number and placement of control points determine the set of possible design solutions. This would address a practical issue in shape optimization, where solutions always depend, among other things, on the adeptness of the user in choosing the proper design variables.

Acknowledgement

The authors would like to present their special thanks to Peter Clausen.

ORCID

Alexander Brune  <http://orcid.org/0000-0001-6857-6557>

Thiago Weber Martins  <http://orcid.org/0000-0002-7763-1156>

Reiner Anderl  <http://orcid.org/0000-0001-7681-2118>

References

- [1] Catmull, E.; Clark, J.: Recursively Generated B-Spline Surfaces On Arbitrary Topological Meshes *Computer-Aided Design*, 10(6), 1978, 350–355. [https://doi.org/10.1016/0010-4485\(78\)90110-0](https://doi.org/10.1016/0010-4485(78)90110-0)
- [2] Gerzen, N.: Analysis and Applications of Variational Sensitivity Information in Structural Optimization, PhD

- Thesis, Dortmund University of Technology, 2014. <https://doi.org/10.17877/DE290R-919>
- [3] Hughes, T.J.R.; Cottrell, J.A.; Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194, 2005, 4135–4195. <https://doi.org/10.1016/j.cma.2004.10.008>
 - [4] Murray, J.; Van Ryper, W.: *Encyclopedia of graphics file formats*. O'Reilly & Associates, 1996. <https://doi.org/10.1109/MM.1996.526933>
 - [5] Olhoff, N.; Bendsøe, M. P.; Rasmussen, J.: On CAD-Integrated Structural Topology and Design Optimization, *Computer Methods in Applied Mechanics and Engineering*, 89(1–3), 1991, 259–279. [https://doi.org/10.1016/0045-7825\(91\)90044-7](https://doi.org/10.1016/0045-7825(91)90044-7)
 - [6] Perry, E.; Balling, R.; Landon, M.: A new morphing method for shape optimization, *American Institute of Aeronautics, Inc.*, 4907, 1998, 1510–1519. <http://doi.org/10.2514/6.1998-4907>
 - [7] Piegl, L.; Tiller, W.: *The NURBS Book*, Springer-Verlag Berlin Heidelberg, 1997. <https://doi.org/10.1007/978-3-642-59223-2>
 - [8] Sederberg, T. W.; Parry, S. R.: Free-form deformation of solid geometric models. *SIGGRAPH Computer Graphics*, 20(4), 1986, 151–160. <https://doi.org/10.1145/15886.15903>
 - [9] Samareh, J.: A survey of shape parameterization techniques, *CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, 1999, 333–343.
 - [10] Sompagdee, P.: Survey of morphing. Department of Computer Science, Thammasat University, 2000.
 - [11] Staten, M.; Owen, S.; Shontz, S.; Salinger, A.; Coffey, T.: A comparison of mesh morphing methods for 3d shape optimization. In *Proceedings of the 20th International Meshing Roundtable*, 2012, 293–311. <https://doi.org/10.1007/978-3-642-24734-7>
 - [12] Svanberg K.: The method of moving asymptotes – a new method for structural optimization. *International Journal For Numerical Methods In Engineering*, 24, 1987, 359–373. <https://doi.org/10.1002/nme.1620240207>
 - [13] Verein Deutscher Ingenieure: *Information technology in product development – Calculation in design, VDI-Guideline 2211*, Berlin, 2003.
 - [14] Wall, W.A.; Frenzel, M.A.; Cyron, C.: Isogeometric structural shape optimization, *Computer Methods in Applied Mechanical Engineering*, 197, 2008, 2976–2988. <https://doi.org/10.1016/j.cma.2008.01.025>