Struct Multidisc Optim (2016) 53:881–892
DOI 10.1007/s00158-015-1359-0

RESEARCH PAPER

On the formulation and implementation of geometric and manufacturing constraints in node–based shape optimization

Oliver Schmitt[1] · **Jan Friederich**[1] · **Stefan Riehl**[1] · **Paul Steinmann**[1]

Received: 1 September 2015 / Revised: 15 October 2015 / Accepted: 22 October 2015 / Published online: 23 November 2015

Abstract We introduce a novel method to handle geometrical and manufacturing constraints in parameter–free shape optimization. Therefore the design node coordinates are split in two sets where one set is declared as new design variables and the other set is coupled to the new design variables such that the geometrical constraint is fulfilled. Thereby no additional equations are appended to the optimization problem. In contrast the implementation of a demolding constraint is presented by formulating inequality constraints which indeed have to be attached to the optimization problem. In the context of a sensitivity–based shape optimization approach all manufacturing constraints have to be formulated in terms of the finite element node coordinates such that first order gradients with respect to the design node coordinates can be derived.

Keywords Shape optimization · Manufacturing constraints · Geometrical constraints · Symmetry · Parameter–free · Sensitivity analysis

1 Introduction

Recent developments within structural shape and topology optimization allow for the efficient improvement of mechanical parts with respect to a large variety of objective functions and constraints. However, in many cases the optimized shapes do not fulfill specific requirements for industrial manufacturing. If at all possible, the modification of the production process would result in high extra costs for the development and fabrication of new machines. For economical reasons manufacturers so instead post–process the optimized shape in order to satisfy the prerequisites for their existing tools. However, these modifications of the part might be sophisticated and are typically done without consideration of the objective function, which leads to a loss of the quality of the part and takes it outside the set of admissible designs. Therefore it is necessary to incorporate manufacturing constraints into the optimization process in order to obtain the optimized, and at the same time, producible shape.

Several investigations have been accomplished to consider geometrical manufacturing constraints in structural optimization. A common example is a demolding constraint for casted parts: the optimized shape must not contain indentations such that the part can not be pulled out of the casting mold after the liquid material has solidified (Ahn et al. 1997). The computer program TopShape (Harzheim and Graf 2002, 2005a) considers a demolding constraint for topology optimization problems (Harzheim and Graf 2005b). Allaire et al. (2013) use the signed–distance function and Xia et al. (2009) modify the design velocity vector to formulate a casting constraint, both using the level–set method. Further constraints regarding fabrication are a limitation of the minimal thickness (Allaire et al. 2014; Guest 2008; Sigmund 2009) to ensure a certain stability of the part. A curvature constraint (Hsu et al. 1995; Wu 2007; Schmitt and Steinmann 2015) for a minimum radius is of interest for features fabricated or reworked by a milling process.

Furthermore, various kinds of symmetrics have to be imposed as geometrical constraints in areas where the

Extended version of a paper presented at the 1st OPT–*i* conference, Kos Island, Greece, 2014

✉ Oliver Schmitt
oliver.schmitt@ltm.uni-erlangen.de

[1] Egerlandstraße 5, 91058 Erlangen, Germany

geometric model can not be reduced, for instance with non–symmetric boundary conditions and loadings or non–symmetric meshes in the context of node–based methods. In the presented manuscript, we extend the idea sketched in Schmitt et al. (2014) and show numerical results from industrial applications.

In the next chapter we establish the boundary value problem and the optimization problem as preliminaries. After this we introduce a novel approach to handle geometrical manufacturing constraints for node–based shape optimization and apply it to the common symmetry types. Furthermore we formulate equations for a molding constraint. Finally, we evaluate the formulation of the manufacturing constraints and their influence on the optimal design trials arrived at by use of several numerical examples.

2 Preliminaries

This chapter is split in two parts. First the domain considered for shape optimization is introduced from the mechanical perspective and the boundary value problem is described (Hughes 1987). In the second part the optimization problem is presented which we use to incorporate the desired manufacturing constraints.

2.1 Mechanical problem

We consider a continuum body $\mathcal{B} \subset \mathbb{R}^3$, with material points $\boldsymbol{x} \in \mathcal{B}$, in the three dimensional Euclidean space (Fig. 1). The boundary of the body $\partial\mathcal{B}$ is split into the Neumann boundary $\partial\mathcal{B}_\mathcal{N}$ and the Dirichlet boundary $\partial\mathcal{B}_\mathcal{D}$ such that $\partial\mathcal{B}_\mathcal{N} \cup \partial\mathcal{B}_\mathcal{D} = \partial\mathcal{B}$ and $\partial\mathcal{B}_\mathcal{N} \cap \partial\mathcal{B}_\mathcal{D} = \emptyset$. On the Dirichlet part $\partial\mathcal{B}_\mathcal{D}$, the displacements $\bar{\boldsymbol{u}}$ are prescribed, whereas on the Neumann part $\partial\mathcal{B}_\mathcal{N}$ the body is loaded by external tractions $\bar{\boldsymbol{t}}$. We consider the case of linear elasticity at small strains where the equilibrium equations take the format

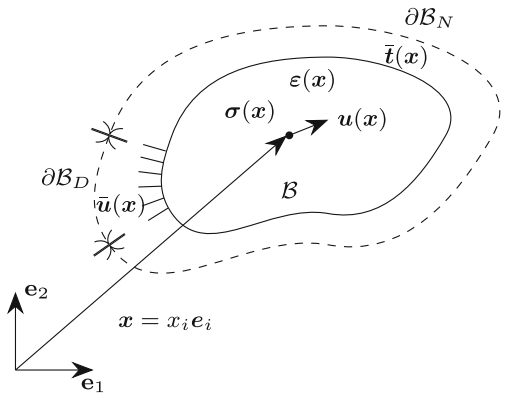


Fig. 1 A continuum body with Dirichlet and Neumann boundary and corresponding boundary conditions

$$(\mathcal{S}) \begin{cases} \nabla \cdot \boldsymbol{\sigma} = \boldsymbol{b} \text{ in } \mathcal{B} \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{\boldsymbol{t}} \text{ on } \partial\mathcal{B}_\mathcal{N} \\ \boldsymbol{u} = \bar{\boldsymbol{u}} \text{ on } \partial\mathcal{B}_\mathcal{D}, \end{cases} \tag{1}$$

with the Cauchy–stress tensor $\boldsymbol{\sigma}$, the outward unit normal $\boldsymbol{n}$ onto $\partial\mathcal{B}_\mathcal{N}$ and the body forces $\boldsymbol{b}$. The symmetric strain tensor $\boldsymbol{\varepsilon}$ is defined as

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left[\nabla\boldsymbol{u} + [\nabla\boldsymbol{u}]^T\right],$$

and the Cauchy stress tensor $\boldsymbol{\sigma}$ can be related to the strain as

$$\boldsymbol{\sigma} = \mathbb{E} : \boldsymbol{\varepsilon}.$$

The material of $\mathcal{B}$ is considered to be isotropic and homogeneous. Hence the elasticity tensor reads

$$\mathbb{E} = \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu\mathbb{I}, \tag{2}$$

where $\mathbf{I}$ and $\mathbb{I}$ are the second and fourth–order unit tensors and λ and μ are the Lamé parameters defined as

$$\lambda = \frac{\nu E}{[1+\nu][1-2\nu]}, \quad \mu = \frac{E}{2[1+\nu]},$$

with Young's modulus E and Poisson's ratio ν. Equation (1) is called the strong form $(\mathcal{S})$ of the boundary value problem. From $(\mathcal{S})$, the weak form $(\mathcal{W})$ of the boundary value problem is derived (Hughes 1987), for simplicity assuming a prescribed zero–displacement ($\bar{\boldsymbol{u}} = \mathbf{0}$) and neglecting the body forces ($\boldsymbol{b} = \mathbf{0}$):

$$(\mathcal{W}) \begin{cases} \text{Find} \\ \boldsymbol{u} \in \mathcal{H}^1_{\partial\mathcal{B}_D}(\mathcal{B}) = \left\{ \boldsymbol{u} \in \mathcal{H}^1(\mathcal{B}) \mid \boldsymbol{u} = \mathbf{0} \text{ on } \partial\mathcal{B}_\mathcal{D} \right) \\ \text{such that} \\ \int_\mathcal{B} \nabla\boldsymbol{v} \cdot \boldsymbol{\sigma}\, \mathrm{d}v = \int_{\partial\mathcal{B}_\mathcal{N}} \boldsymbol{v} \cdot \bar{\boldsymbol{t}}\, \mathrm{d}a \\ \forall\, \boldsymbol{v} \in \mathcal{H}^1_{\partial\mathcal{B}_\mathcal{D}}(\mathcal{B}) = \left\{ \boldsymbol{v} \in \mathcal{H}^1(\mathcal{B}) \mid \boldsymbol{v} = \mathbf{0} \text{ on } \partial\mathcal{B}_\mathcal{D} \right) \end{cases}$$

After discretization of the domain into n_e finite elements

$$\mathcal{B} \approx \mathcal{B}^h = \sum_{e=1}^{n_e} \mathcal{B}^e, \tag{3}$$

the geometry and the displacement are both approximated on each element according to the isoparametric concept such that

$$\boldsymbol{x} \approx \boldsymbol{x}^h = \sum_{i=1}^{n_{en}} \boldsymbol{x}_i N_i \qquad \text{and} \qquad \boldsymbol{u} \approx \boldsymbol{u}^h = \sum_{i=1}^{n_{en}} \boldsymbol{u}_i N_i, \tag{4}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{u}_i$ denote the corresponding nodal values, which are subsumed in global vectors $\mathbf{x}$ and $\mathbf{u}$, N_i are the shape functions on the reference element, and n_{en} is the number of nodes per elements. When incorporating (3) and (4) into the weak form $(\mathcal{W})$ of (1), the problem reduces to solving a linear system

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \tag{5}$$

with the stiffness matrix $\mathbf{K}$, the unknown global solution vector $\mathbf{u}$, and the right hand side, i.e. the global force vector $\mathbf{f}$. Equation (5) can also be formulated as

$$\mathbf{0} = \mathbf{r} = \mathbf{r}^{int} - \mathbf{r}^{ext} = \mathbf{K}\mathbf{u} - \mathbf{f}, \tag{6}$$

where $\mathbf{r}$, $\mathbf{r}^{int}$ and $\mathbf{r}^{ext}$ are the residuum, and the internal and external global force vectors, respectively.

2.2 Optimization problem

In view of the shape optimization problem we consider shape changes of $\partial\mathcal{B}$ given through design updates after each iteration. These shape changes underly boundary conditions for the optimization problem. In the context of node–based shape optimization, the FE–node coordinates act as design variables. For a maximum design freedom all boundary node coordinates, which do not have to stay fixed are defined as design variables. We declare three different types of nodal coordinates:

1. *design node coordinates* $\mathbf{x}_d$:
   Node coordinates which may be displaced by the design update.
2. *controlled node coordinates* $\mathbf{x}_c$:
   Node coordinates which are not updated by the design update but may be displaced in a controlled manner in order to avoid distorted elements.
3. *fixed node coordinates* $\mathbf{x}_f$:
   Node coordinates which are completely fixed during the optimization procedure.

In the optimization problem we consider an objective function f to be minimized with respect to a set of inequality constraints $g_i, i = 1, \ldots, n_{ieq}$:

$$\begin{aligned} \min \quad & f(\mathbf{u}(\mathbf{x}_d), \mathbf{x}_d) \\ \text{subject to} \quad & g_i(\mathbf{u}(\mathbf{x}_d), \mathbf{x}_d) \leq 0, i = 1, \ldots, n_{ieq} \\ \text{and} \quad & \mathbf{K}\mathbf{u} = \mathbf{f}. \end{aligned}$$

The objective function and all constraint functions have to be formulated in terms of the FE–node coordinates (more precisely the design node coordinates), where the inequality constraints are defined as:

$$g_i = g_i^{current} - g_i^{max} \leq 0, \quad i = 1, \ldots, n_{ieq}$$

with the respective function evaluated using the design of the current iteration $g_i^{current}$ and a user specified value g_i^{max}.

A scalar quantity which is often used as objective or constraint function in shape optimization problems is the von–Mises stress, which is computed using the entries of the Cauchy stress tensor:

$$\sigma_{vM}(\mathbf{x}) := \left[-\frac{1}{2} tr(\boldsymbol{\sigma})^2 + \frac{3}{2} tr(\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}) \right]^{\frac{1}{2}} \tag{7}$$

Moreover, we introduce the compliance C as the work of the external forces defined as

$$C := \int_{\partial\mathcal{B}_N} \boldsymbol{u} \cdot \bar{\boldsymbol{t}} \, \mathrm{d}A = \mathbf{u} \cdot \mathbf{f}, \tag{8}$$

and the volume V of $\mathcal{B}^h$ which is given by

$$V := \int_{\mathcal{B}^h} 1 \, \mathrm{d}V = \sum_{e=1}^{n_e} \int_{\mathcal{B}^e} 1 \, \mathrm{d}V. \tag{9}$$

2.2.1 Mesh movement and shape regularization

In a node–based shape optimization method the iterative design update $\Delta\mathbf{x}_d$ returned by the optimization algorithm is directly used to modify the nodal coordinates of all design nodes $\mathbf{x}_d$. In order to avoid highly irregular elements close to the boundary in areas of large shape changes a mesh update has to be taken into account after each design update. Here, this update results from an auxiliary boundary value problem with the design update as prescribed displacements (Yao and Choi 1989):

$$\begin{aligned} \nabla \cdot \tilde{\boldsymbol{\sigma}} &= \mathbf{0} \quad \text{in } \mathcal{B} \\ \tilde{\boldsymbol{u}} &= \boldsymbol{u}_d \quad \text{on } \widetilde{\partial\mathcal{B}_{\mathcal{D}}}, \end{aligned} \tag{10}$$

where $\boldsymbol{u}_d$ describes the shape change resulting of the design update $\Delta\mathbf{x}_d$. The material used for the auxiliary boundary value problem is also considered to be homogeneous and isotropic, but the (linear) elastic behavior is chosen to be very soft (i.e. $E = 2.25$ MPa, $\nu = 0.125$). Thus the actual update $\Delta\mathbf{x}$ consists of the design update $\Delta\mathbf{x}_d$ in the present case computed by SNOPT and the mesh update $\tilde{\mathbf{u}}$, which results from (10):

$$\Delta\mathbf{x} = \Delta\mathbf{x}_d + \tilde{\mathbf{u}}$$

Although the node–based method is one of the most popular approaches in the field of structural shape optimization it suffers from known drawbacks such as jagged design boundaries especially in areas of large shape changes. To gain control of that issue a shape regularization technique based on Scherer et al. (2009) is applied. In this approach the actual objective function is penalized by a fictitious strain energy $\mathcal{I}$,

$$\mathcal{J} = f + \alpha\mathcal{I},$$

where $\alpha \in \mathbb{R}_0^+$ is a parameter controlling the influence of the regularization. Using the same auxiliary boundary value problem as for the mesh movement (10), the fictitious strain energy is given by the quadratic energy function

$$\mathcal{I} := \frac{1}{2} \int_{\mathcal{B}^h} \tilde{\boldsymbol{\varepsilon}} : \widetilde{\mathbb{E}} : \tilde{\boldsymbol{\varepsilon}} \, \mathrm{d}v,$$

with

$$\tilde{\boldsymbol{\varepsilon}} = \nabla^{\mathrm{sym}} \tilde{\boldsymbol{u}},$$

and $\widetilde{\mathbb{E}}$ according to (2) and the material parameters of the auxiliary problem. Then the optimization problem with penalized objective function reads:

$$\begin{aligned} \min \quad & \mathcal{J}\left(\mathbf{u}(\mathbf{x}_d), \mathbf{x}_d, \alpha\right) = f\left(\mathbf{u}(\mathbf{x}_d), \mathbf{x}_d\right) + \alpha \mathcal{I}^*\left(\mathbf{x}_c^*(\mathbf{x}_d), \mathbf{x}_d\right) \\ \text{s. t.} \quad & g_i\left(\mathbf{u}(\mathbf{x}_d), \mathbf{x}_d\right) \leq 0, \quad i = 1, \ldots, n_{ieq} \\ \text{and} \quad & \mathbf{K}\mathbf{u} = \mathbf{f} \end{aligned}$$

The penalizing strain energy $\mathcal{I}$ can be understood as a measure of shape change resulting from the design update. As a consequence, for large values of α the relative weight of the actual objective f decreases and the regularized objective $\mathcal{J}$ increasingly gets influenced by the fictitious strain energy $\mathcal{I}$. However, if α is chosen too large, the decrease of f might be dominated by the increase of the penalization $\alpha\mathcal{I}$, therefore obstructing effective shape changes. On the other hand, for α too small the influence of the fictitious strain energy vanishes, resulting in unregularized shapes.

In order to propose a reasonable choice of the control parameter α in our numerical applications, we will normalize the actual objective w.r.t. its initial value f^0 by setting $f \rightarrow f/f^0$ and rescale the fictitious energy by $\mathcal{I} \rightarrow \mathcal{I}/V^0$, where V^0 is the volume of the initial geometry. Then, values $\alpha \in [1.0, 5.0]$ turn out to be good choices for effective but regular shape changes in our experience. In addition, one could restart the optimization with the optimized shape as initial geometry in order to allow for even greater shape changes. For a more detailed discussion the reader referred to Scherer et al. (2009).

2.2.2 Sensitivity analysis

As a prerequisite for the solution of the optimization problem, the gradients of all objective and constraint functions with respect to the design variables have to be derived. Therefore, for each implicit dependency the chain rule of differentiation has to be applied, cf. Scherer et al. (2009). The functions (7), (8) and (9) generally depend on the displacements, the controlled node coordinates and the design node coordinates.

Due to the dependencies $\mathbf{u}(\mathbf{x}_d, \mathbf{x}_c(\mathbf{x}_d))$ in the mechanical problem with $\mathbf{x}_c(\mathbf{x}_d)$ due to the auxiliary problem, they can be computed solely in terms of the design node coordinates. Thus the sensitivity analysis can be split in two steps:

chain rule step: $\mathbf{u}(\mathbf{x})$

$$\begin{aligned} \frac{\partial f^*(\mathbf{x})}{\partial \mathbf{x}} &= \frac{\partial f^*(\mathbf{u}^*(\mathbf{x}), \mathbf{x})}{\partial \mathbf{x}} \\ &= \frac{\partial f(\mathbf{u}^*(\mathbf{x}), \mathbf{x})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial f(\mathbf{u}^*(\mathbf{x}), \mathbf{x})}{\partial \mathbf{x}} \end{aligned}$$

From the mechanical problem (6)

$$\begin{aligned} \mathbf{0} = \frac{\partial \mathbf{r}^*(\mathbf{x})}{\partial \mathbf{x}} &= \frac{\partial \mathbf{r}^*(\mathbf{u}^*(\mathbf{x}), \mathbf{x})}{\partial \mathbf{x}} \\ &= \frac{\partial \mathbf{r}^{*,int}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}^{*,ext}}{\partial \mathbf{x}} \\ &= \frac{\partial \mathbf{r}^{int}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}^{int}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}^{ext}}{\partial \mathbf{x}} \\ &= \mathbf{K} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}}{\partial \mathbf{x}}, \end{aligned}$$

it follows that

$$\frac{\partial f^*}{\partial \mathbf{x}} = -\frac{\partial f}{\partial \mathbf{u}} \mathbf{K}^{-1} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{x}}. \tag{11}$$

Introducing the adjoint solution $\boldsymbol{\lambda}^*$ by

$$\mathbf{K}^T \boldsymbol{\lambda}^* = -\frac{\partial f}{\partial \mathbf{u}},$$

(11) can be reformulated as

$$\frac{\partial f^*}{\partial \mathbf{x}} = \boldsymbol{\lambda}^* \frac{\partial \mathbf{r}}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{x}}.$$

chain rule step: $\mathbf{x}_c(\mathbf{x}_d)$

$$\begin{aligned} \frac{\partial f^{**}(\mathbf{x}_d)}{\partial \mathbf{x}_d} &= \frac{\partial f^{**}(\mathbf{x}_c(\mathbf{x}_d), \mathbf{x}_d)}{\partial \mathbf{x}_d} \\ &= \frac{\partial f^*(\mathbf{x}_c(\mathbf{x}_d), \mathbf{x}_d)}{\partial \mathbf{x}_c} \frac{\partial \mathbf{x}_c}{\partial \mathbf{x}_d} + \frac{\partial f^*(\mathbf{x}_c(\mathbf{x}_d), \mathbf{x}_d)}{\partial \mathbf{x}_d} \end{aligned}$$

From the auxiliary problem

$$\begin{aligned} \mathbf{0} = \frac{\partial \mathbf{R}^*(\mathbf{x}_d)}{\partial \mathbf{x}_d} &= \frac{\partial \mathbf{R}^*(\mathbf{x}_c(\mathbf{x}_d), \mathbf{x}_d))}{\partial \mathbf{x}_d} \\ &= \frac{\partial \mathbf{R}}{\partial \mathbf{x}_c} \frac{\partial \mathbf{x}_c}{\partial \mathbf{x}_d} + \frac{\partial \mathbf{R}}{\partial \mathbf{x}_d} \\ \Rightarrow \quad \frac{\partial \mathbf{x}_c}{\partial \mathbf{x}_d} &= -\left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}_c}\right]^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{x}_d}, \end{aligned}$$

it follows that

$$\frac{\partial f^{**}}{\partial \mathbf{x}_d} = -\frac{\partial f^*}{\partial \mathbf{x}_c} \left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}_c}\right]^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{x}_d} + \frac{\partial f^*}{\partial \mathbf{x}_d}. \tag{12}$$

Introducing the adjoint solution $\boldsymbol{\lambda}^{**}$ by

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}_c}\right]^T \boldsymbol{\lambda}^{**} = -\frac{\partial f^*}{\partial \mathbf{x}_c},$$

(12) can be formulated as

$$\frac{\partial f^{**}}{\partial \mathbf{x}_d} = \boldsymbol{\lambda}^{**} \frac{\partial \mathbf{R}}{\partial \mathbf{x}_d} + \frac{\partial f^*}{\partial \mathbf{x}_d}. \tag{13}$$

For the sensitivity of the fictitious energy term we get

$$\frac{\partial \mathcal{I}^*(\mathbf{x}_c^*(\mathbf{x}_d), \mathbf{x}_d)}{\partial \mathbf{x}_d} = \frac{\partial \mathcal{I}}{\partial \mathbf{x}_c} \frac{\partial \mathbf{x}_c^*}{\partial \mathbf{x}_d} + \frac{\partial \mathcal{I}}{\partial \mathbf{x}_d} = \frac{\partial \mathcal{I}}{\partial \mathbf{x}_d},$$

where $\frac{\partial \mathcal{I}}{\partial \mathbf{x}_c} = \mathbf{0}$, since the auxiliary problem is solved by minimizing the fictitious energy

$$\mathbf{x}_c^*(\mathbf{x}_d) = \arg\min_{\mathbf{x}_c} \mathcal{I}(\mathbf{x}_c, \mathbf{x}_d).$$

Summarizing, the gradient of the objective function thus reads

$$\frac{\partial \mathcal{J}}{\partial \mathbf{x}_d} = \frac{\partial f^{**}}{\partial \mathbf{x}_d} + \alpha \frac{\partial \mathcal{I}}{\partial \mathbf{x}_d}.$$

2.2.3 SNOPT

For the numerical examples in this contribution the design update is computed using the gradient–based general purpose optimizer *Sparse Nonlinear OPTimizer* (SNOPT) (Gill et al. 2005) which is known for its robust handling of highly constrained, large–scale, nonlinear optimization problems. The algorithm has to be provided with the objective and the constraints evaluated at the current design state as well as their gradients with respect to the design variables. We incorporated SNOPT into an in–house Finite–Element program in Python using the open source package pyOpt (Perez et al. 2011).

3 Manufacturing constraints

We present constraints which have the purpose of ensuring that the optimized shape can still be produced using a specific manufacturing process. We classify two different approaches (Schmitt et al. 2014): In the first approach, we formulate the constraints by defining two disjoint subsets of the design nodes

$$\{\mathbf{x}_d\} \longrightarrow \{\mathbf{x}_d^{opt}, \mathbf{x}_d^{dep}\}$$

and a mapping

$$\mathbf{x}_d^{dep} = \mathcal{F}(\mathbf{x}_d^{opt}), \tag{14}$$

that relates one subset to the other. In the following we call the two subsets optimization nodes $\mathbf{x}_d^{opt}$ and dependent nodes $\mathbf{x}_d^{dep}$ and $\mathcal{F}$ is denoted as implicit constraint function. The optimization node coordinates from now on serve as design variables and the mapping determines the design update for the dependent node coordinates. This yields an equivalent optimization problem with a reduced number of design variables and no additional constraints. However, through an additional chain rule step the sensitivity information of the dependent nodes is considered during the sensitivity analysis and therefore contributes to the computation of the design update. Hence, the selection of the optimization nodes does not influence the final design as long as it meets the desired geometric constraint. This approach is only applicable for certain types of geometrical constraints but implicates two advantages, namely a reduction of the number of design variables and no additional constraints to be appended to the optimization problem. By contrast, in a second approach we derive equality and inequality constraints in terms of the design variables, so–called explicit constraints, and append them to the optimization problem.

3.1 Implicit constraints

Due to the advantages mentioned above, we always aim in implementing manufacturing constraints using the implicit approach. In order to ensure the functionality of a part, several types of symmetrics, such as the herein presented reflection, rotational and cyclic symmetry may be of interest. In the proposed approaches we restrict ourselves to employ tetrahedron elements.

3.1.1 Reflection symmetry

An object is reflection symmetric if there is a plane going through the object dividing it into two pieces which are mirror images of each other. Therefore we define a symmetry plane $\mathcal{S}$ which splits the design nodes into two sets according to their position regarding $\mathcal{S}$ (see Fig. 2). Thereby it is not important which of the two sets is chosen as optimization nodes as the sensitivities of both enter the sensitivity analysis eventually. To couple the two sets we compute the intersection $\boldsymbol{y}_j$ of a line through the dependent node $\boldsymbol{x}_j^{dep}$ and perpendicular to the symmetry plane $\mathcal{S}$ with the surface of the body on the other side of $\mathcal{S}$ (see Fig. 2). The intersection $\boldsymbol{y}_j$ can be expressed in terms of the adjacent FE–nodes of the intersected surface element using barycentric coordinates. We declare surface elements as those facets of finite elements at the boundary which intersect with the boundary of our body $\mathcal{B}$. Thus the function for the update of the dependent nodes reads

$$\begin{aligned}\boldsymbol{x}_j^{dep} &= \boldsymbol{y}_j - 2\mathrm{d}\left(\mathcal{S}, \boldsymbol{y}_j\right) \boldsymbol{n}_{\mathcal{S}} \\ &= \sum_{k=1}^{3} \lambda_k \boldsymbol{x}_{i_k}^{opt} - 2\left[\left[\sum_{k=1}^{3} \lambda_k \boldsymbol{x}_{i_k}^{opt} - \boldsymbol{p}_{\mathcal{S}}\right] \cdot \boldsymbol{n}_{\mathcal{S}}\right] \boldsymbol{n}_{\mathcal{S}}.\end{aligned}$$

where λ_k are the barycentric coordinates, $\mathrm{d}(\mathcal{S}, \boldsymbol{y}_j)$ is the distance between the intersection $\boldsymbol{y}_j$ and the symmetry plane $\mathcal{S}$, $\boldsymbol{p}_{\mathcal{S}}$ is an arbitrary, fixed point on $\mathcal{S}$ and $\boldsymbol{n}_{\mathcal{S}}$ is the unit normal to $\mathcal{S}$ towards the optimization nodes' side.

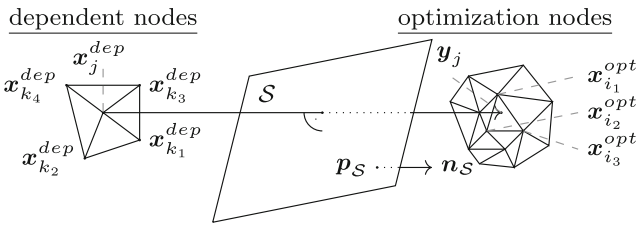


Fig. 2 Symmetry plane with optimization nodes and dependent nodes

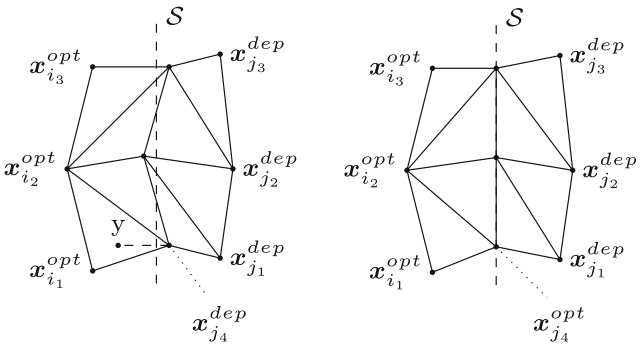


Fig. 3 Conflicts are solved by moving intersected surface element edges on the symmetry plane

Surface elements which are cut by the symmetry plane may lead to conflicts when it comes to the computation of the Jacobian matrix of the implicit constraint function $\mathcal{F}$, since the intersection $\boldsymbol{y}_j$ would be defined by both optimization and dependent nodes. In this case, we move vertices of edges intersected by $\mathcal{S}$ onto the symmetry plane, whereby nodes on the symmetry plane are declared as optimization nodes as illustrated in Fig. 3. This minor modification regularly happens in directions close to the tangential space of the boundary of our domain $\partial\mathcal{B}$ and thus just slightly changes the geometry of our body.

3.1.2 Rotational symmetry

An object is rotational symmetric if a rotation around a fixed axis does not change the object's shape. Therefore, the vertices along a polygonal chain of edges on the lateral surface are chosen as optimization nodes (see Fig. 4), whereas all other nodes are declared as dependent nodes. For the design update of the dependent nodes, we consider all design nodes in cylindrical coordinates (r, h, φ) with respect to the rotation axis $\boldsymbol{a}$ and an arbitrary plane normal to $\boldsymbol{a}$, here defined

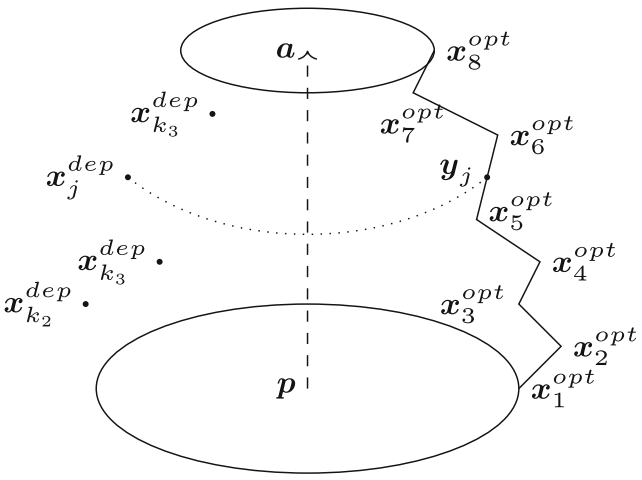


Fig. 4 Rotational symmetry line with choice of optimization nodes

using the point $\boldsymbol{p}$. First, for each dependent node $\boldsymbol{x}_j^{dep}$ a reference point $\boldsymbol{y}_j$ on an edge between two optimization nodes $\boldsymbol{x}_{i_1}^{opt}$ and $\boldsymbol{x}_{i_2}^{opt}$ is found with equal height and radius (depicted in Fig. 4). Then, we determine two coordinates λ_{i_1} and λ_{i_2} in terms of the heights of the vertices $\boldsymbol{x}_{i_1}^{opt}$ and $\boldsymbol{x}_{i_2}^{opt}$ based on

$$h\left(\boldsymbol{x}_j^{dep}\right) = \lambda_{i_1} h\left(\boldsymbol{x}_{i_1}^{opt}\right) + \lambda_{i_2} h\left(\boldsymbol{x}_{i_2}^{opt}\right), \text{ with } \lambda_{i_1} + \lambda_{i_2} = 1,$$

Obviously the angle φ of the design nodes does not influence the geometry of a rotational symmetric body. Thus, using the previously computed coordinates, the height and the radius of the dependent nodes are updated as

$$\begin{aligned} h\left(\boldsymbol{x}_j^{dep}\right) &= \lambda_{i_1} h\left(\boldsymbol{x}_{i_1}^{opt}\right) + \lambda_{i_2} h\left(\boldsymbol{x}_{i_2}^{opt}\right) \\ r\left(\boldsymbol{x}_j^{dep}\right) &= \lambda_{i_1} r\left(\boldsymbol{x}_{i_1}^{opt}\right) + \lambda_{i_2} r\left(\boldsymbol{x}_{i_2}^{opt}\right). \end{aligned}$$

This update may be formulated in one function as

$$\begin{aligned} \boldsymbol{x}_j^{dep} = \boldsymbol{p} &+ \left[\lambda_{i_1} h\left(\boldsymbol{x}_{i_1}^{opt}\right) + \lambda_{i_2} h\left(\boldsymbol{x}_{i_2}^{opt}\right)\right] \frac{\boldsymbol{a}}{||\boldsymbol{a}||_2} \\ &+ \left[\lambda_{i_1} r\left(\boldsymbol{x}_{i_1}^{opt}\right) + \lambda_{i_2} r\left(\boldsymbol{x}_{i_2}^{opt}\right)\right] \frac{\mathbf{v}_j}{||\mathbf{v}_j||_2}, \end{aligned}$$

where

$$\mathbf{v}_j = \boldsymbol{x}_j^{dep} - \left[\lambda_{i_1} h(\boldsymbol{x}_{i_1}^{opt}) + \lambda_{i_2} h\left(\boldsymbol{x}_{i_2}^{opt}\right)\right] \frac{\boldsymbol{a}}{||\boldsymbol{a}||_2}.$$

Note that for the choice of the set of optimization nodes it is crucial to prevent a wriggled chain of edges around the rotation axis. Therefore when picking the optimization nodes we follow the below procedure:

1. Choose the design node with the lowest height as start node: $s_0 = (r_0, h_0, \varphi_0)$.
2. $s_i \rightarrow s_{i+1}$:
   Choose the following node $s_{i+1} = (r_{i+1}, h_{i+1}, \varphi_{i+1})$ such that

   $$|\varphi_{i+1} - \varphi_0| \leq |\varphi_j - \varphi_0|, \quad \forall j \in \mathcal{N}_i$$

   where

   $$\mathcal{N}_i = \{j \mid \mathbf{x}_j \text{ is adjacent to } \mathbf{x}_i \ , \ h(\mathbf{x}_j) > h(\mathbf{x}_i)\}.$$

3.1.3 Cyclic symmetry

An object is cyclic symmetric if a rotation around a fixed axis by a multiple of a fixed angle does not change the object's shape. Thus we define a plane by its normal $\boldsymbol{n}$ which also holds as circular axis intersecting the plane through the center point $\boldsymbol{c}$. We suppose that the initial geometry is cyclic symmetric and divide the plane into $[2\pi/\gamma] \in \mathbb{N}$ equally sized circular sectors, where γ is the angle of the circular sector. Now, the design nodes in one sector are declared as optimization nodes whereas all other nodes are dependent nodes. For the coupling of the dependent nodes to the optimization nodes, a reference point $\boldsymbol{y}_j$ is determined

by rotating each dependent node around the axis $\boldsymbol{n}$ by the respective multiple $s \in \mathbb{N}$ of the sector angle γ into the sector containing the optimization nodes (illustrated in Fig. 5). So with the design update of the optimization nodes the dependent nodes can be updated by

$$\boldsymbol{x}_j^{dep} = \mathbf{c} + \mathcal{R}^{-s}(\gamma, \boldsymbol{n}) \cdot [\boldsymbol{y}_j - \mathbf{c}],$$

where $\boldsymbol{y}_j$ is expressed in terms of the vertices of the corresponding surface element uniquely computed in the initial design:

$$\boldsymbol{y}_j = \lambda_{i_1} \boldsymbol{x}_1^{opt} + \lambda_{i_2} \boldsymbol{x}_2^{opt} + \lambda_{i_3} \boldsymbol{x}_3^{opt},$$

with the rotation matrix $\mathcal{R}(\boldsymbol{n}, \gamma)$ around the axis $\boldsymbol{n}$ by the angle γ.

3.1.4 Additional step in the sensitivity analysis for implicit constraints

The implicitly handled constraints in this chapter split the set of design nodes into two new sets where one set is declared as new design nodes and the other set is updated according to the implicit constraint function $\mathcal{F}$ in terms of the new design variables. This procedure introduces an additional dependency $\mathbf{x}_d^{dep}(\mathbf{x}_d^{opt})$ which has to be considered during the sensitivity analysis. More precisely, let

$$f^{***}\left(\mathbf{x}_d^{opt}\right) = f^{***}\left(\mathbf{x}_d^{dep}\left(\mathbf{x}_d^{opt}\right), \mathbf{x}_d^{opt}\right).$$

chain rule step: $\mathbf{x}_d^{dep}\left(\mathbf{x}_d^{opt}\right)$

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{x}_d^{opt}} f^{***}\left(\mathbf{x}_d^{opt}\right) &= \frac{\partial}{\partial \mathbf{x}_d^{opt}} f^{***}\left(\mathbf{x}_d^{dep}\left(\mathbf{x}_d^{opt}\right), \mathbf{x}_d^{opt}\right) \\
&= \frac{\partial f^{**}}{\partial \mathbf{x}_d^{dep}} \frac{\partial \mathbf{x}_d^{dep}}{\partial \mathbf{x}_d^{opt}} + \frac{\partial f^{**}}{\partial \mathbf{x}_d^{opt}} \\
&\overset{(14)}{=} \left[\frac{\partial \mathcal{F}}{\partial \mathbf{x}_d^{opt}}\right]^T \frac{\partial f^{**}}{\partial \mathbf{x}_d^{dep}} + \frac{\partial f^{**}}{\partial \mathbf{x}_d^{opt}},
\end{aligned}$$

where $\left[\frac{\partial \mathcal{F}}{\partial \mathbf{x}_d^{opt}}\right]^T$ is the transposed Jacobian of the function $\mathcal{F}$ for the dependent nodes and $\frac{\partial f^{**}}{\partial \mathbf{x}_d^{dep}}$, $\frac{\partial f^{**}}{\partial \mathbf{x}_d^{opt}}$ are computed according to (13).

Note that multiple implicit constraints may be combined where for each constraint the respective additional sensitivity analysis step has to be applied.

3.2 Explicit constraints

In cases, where the aforementioned approach cannot be applied, we use the standard procedure and derive equality or inequality constraints restricting the set of admissible domains to fulfill the constraint.

We exemplarily consider a demolding constraint, which has to be fulfilled for parts which are produced in a casting process to ensures that the optimized geometry can be pulled out of the casting mould after the liquid material has been solidified. Therefore a demolding direction $\boldsymbol{d}$ is defined as the direction in which the part is supposed to

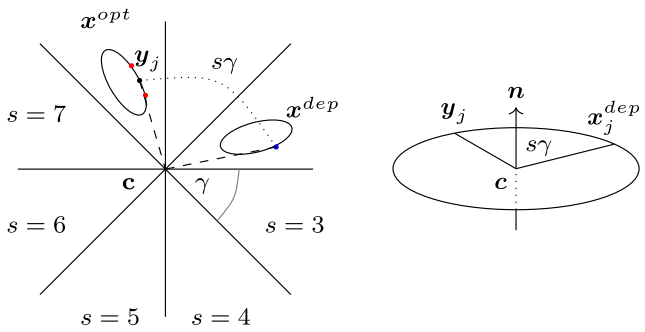


Fig. 5 Sketch of a cyclic symmetric model with eight sectors and an angle of 45°

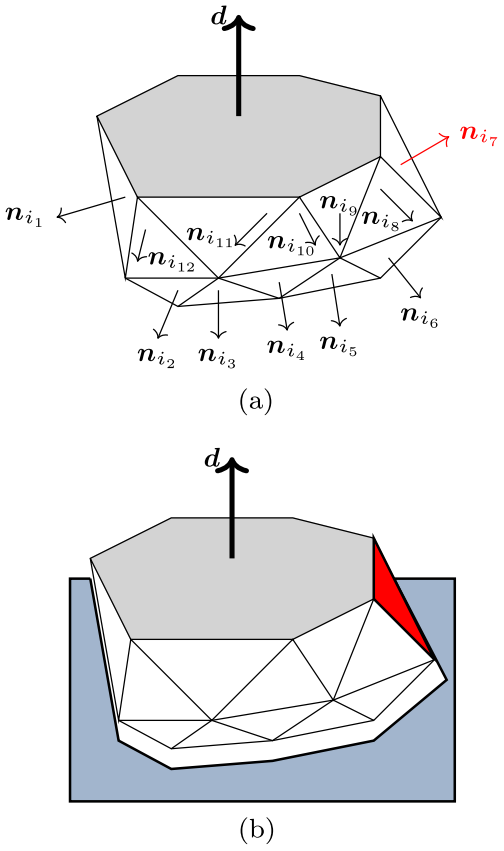


Fig. 6 Cast part **a** with demolding direction and outwards pointing normals and **b** in a casting mould

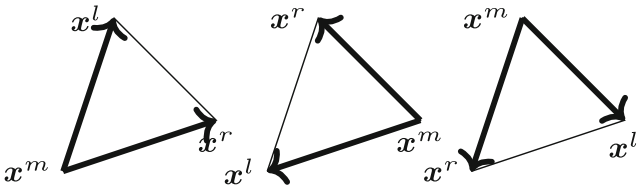


Fig. 7 Three combinations of a cross product of element edges leading to the same surface element normal

be pulled out of the mould. Moreover the outward normals $\boldsymbol{n}_i$ of a declared set of surface elements are computed. Indentations are avoided by requiring that

$$\measuredangle(\boldsymbol{n}_i, \boldsymbol{d}) \geq 90° \quad i = 1, \ldots, n_{\text{demold}},$$

where $\boldsymbol{n}_i$ are the outwards pointing normals to a subset of n_{demold} surface elements (see Fig. 6). This can be formulated as an inequality constraint:

$$g_i := \boldsymbol{n}_i \cdot \boldsymbol{d} \leq 0 \quad i = 1, \ldots, n_{\text{demold}} \tag{15}$$

Hence this manufacturing constraint results in n_{demold} additional inequality constraints to be appended to the optimization problem. The SNOPT we use in our numerical examples is very robust in handling highly restricted problems which is why we add each additional inequality constraint separately to the optimization problem. We would like to mention that one could also aggregate all demolding constraints to just one inequality constraint using e.g. the Kreisselmeier–Steinhauser function (Poon and Martins 2006).

For a sensitivity–based approach the first order gradient of each constraint function is required. In contrast to a general derivation in a continuous setting, see for instance Michailidis (2014), we directly rely on the discrete representation of the design boundary. The outer normal is computed using the cross product of the vectors spanning the respective surface element, i.e.

$$\boldsymbol{n} = \frac{\tilde{\boldsymbol{n}}}{||\tilde{\boldsymbol{n}}||_2}, \quad \text{where} \quad \tilde{\boldsymbol{n}} = [\boldsymbol{x}^l - \boldsymbol{x}^m] \times [\boldsymbol{x}^r - \boldsymbol{x}^m],$$

where $\boldsymbol{x}^l$, $\boldsymbol{x}^r$ and $\boldsymbol{x}^m$ are the vertices of the surface element, see Fig. 7. On noting that any labeling of the vertices as depicted in Fig. 7 results in the same vector $\tilde{\boldsymbol{n}}$, it lasts to compute the derivative with respect to the vertex $\boldsymbol{x}^l$ without loss of generality. Then the derivative of the scalar product in (15) reads:

$$\begin{aligned}
\frac{\partial[\boldsymbol{n} \cdot \boldsymbol{d}]}{\partial \boldsymbol{x}^l} &= \sum_{i=1}^{3} d_i \frac{\partial n_i}{\partial \boldsymbol{x}^l}, \\
\text{with } \frac{\partial n_i}{\partial \boldsymbol{x}^l} &= \left[\frac{\partial \tilde{n}_i}{\partial \boldsymbol{x}^l}||\tilde{\boldsymbol{n}}||_2 - \tilde{n}_i \frac{\partial ||\tilde{\boldsymbol{n}}||_2}{\partial \boldsymbol{x}^l}\right] ||\tilde{\boldsymbol{n}}||_2^{-2}, \\
\frac{\partial ||\tilde{\boldsymbol{n}}||_2}{\partial \boldsymbol{x}^l} &= \frac{1}{||\tilde{\boldsymbol{n}}||_2} \sum_{i=1}^{3} \left[\tilde{n}_i \frac{\partial \tilde{n}_i}{\partial \boldsymbol{x}^l}\right] \\
\text{and } \frac{\partial \tilde{n}_i}{\partial x_b^l} &= \frac{\partial e_{ijk}\left[x_j^l - x_j^m\right]\left[x_k^r - x_k^m\right]}{\partial x_b^l} = e_{ibk}\left[x_k^r - x_k^m\right],
\end{aligned}$$

where e_{ijk} is the Levi–Civita symbol.

4 Numerical examples

The introduced manufacturing constraints are illustrated exemplarily for one implicitly and one explicitly handled case. Both models are discretized using linear tetrahedron elements. SNOPT is adjusted to terminate when the decrease of the objective function for the design update with

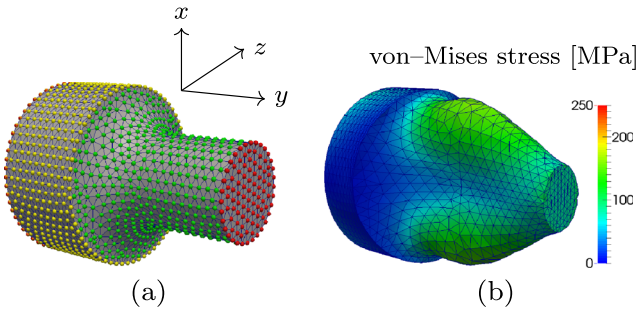


Fig. 9 **a** node declaration for the optimization problem and **b** optimization result without rotational symmetry constraint

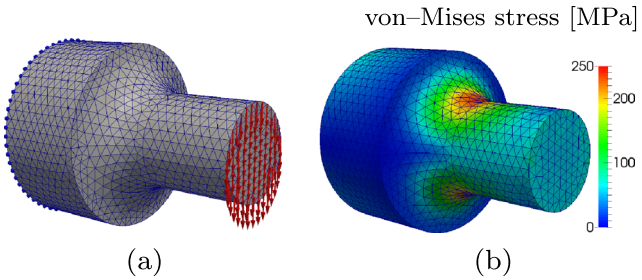


Fig. 8 **a** load case and **b** initial stress distribution of a rotational symmetric part of a shaft

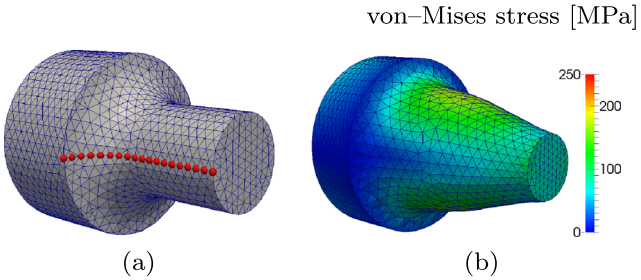
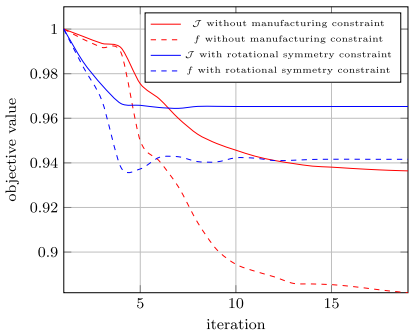
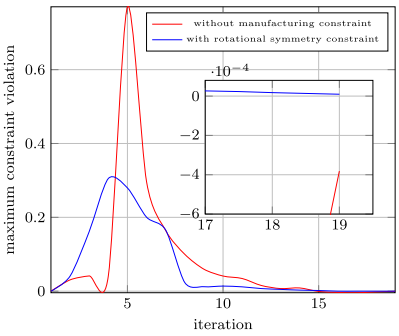


Fig. 10 **a** optimization nodes and **b** optimization result with rotational symmetry constraint



(a)



(b)

# elements	35717	
# nodes	6837	
rotat. sym.	no	yes
# design nodes	743	19
# iterations	19	19
decrease $\mathcal{J}$	$\sim 6.3\%$	$\sim 3.5\%$
decrease f	$\sim 11.8\%$	$\sim 5.8\%$
run time [min]	113	131

(c)

Fig. 11 **a** convergence plots and **b** maximum constraint violation for the optimization of a shaft without and with rotational symmetry constraint and **c** a summarizing chart

respect to the previous iteration is lower than a predefined tolerance c

$$\mathcal{J}(\mathbf{x}_d^{i-1}) - \mathcal{J}(\mathbf{x}_d^{i}) < c = 10^{-4},$$

where i is the current iteration and

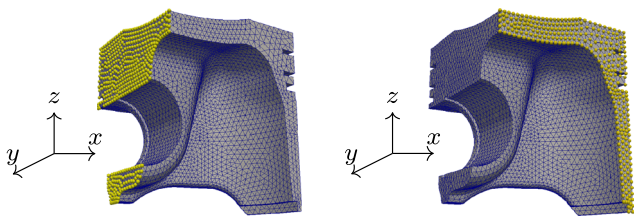
$$\mathcal{J} = \frac{f}{f^0} + \alpha \frac{\mathcal{I}}{V^0}.$$

Here, the objective function f (i.e. the compliance, volume or von–Mises stress) is scaled by it's initial value such that the objective value for every optimization starts at one whereas the fictitious energy $\mathcal{I}$ is zero for the initial geometry.

4.1 Shaft

The first model is a rotational symmetric part of a shaft with 35717 elements and 6837 nodes. The part is fixed on the left side (marked by blue nodes) and loaded on its right side by concentrated node forces of 1 kN (illustrated by red arrows), see Fig. 8a. The material parameters are set to $E = 209000$ MPa and $\nu = 0.3$. The von–Mises stress distribution is illustrated in Fig. 8b. The objective is to minimize the volume where the maximum stress must not increase. Hence the optimization problem can be formulated as

$$\begin{aligned} \min \quad & \frac{V}{V^0} + \alpha \frac{\mathcal{I}}{V^0} \\ \text{subject to} \quad & \max[\sigma_{vM}] - \max[\sigma_{vM}^0] \leq 0 \end{aligned}$$



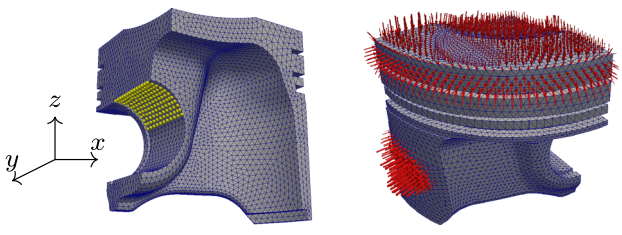
(a) fixed in x–direction (b) fixed in y–direction

Fig. 12 Symmetry boundary conditions

where $\max[\sigma_{vM}^0]$ is the maximum von–Mises stress in the initial design $\mathcal{B}^0$. The maximum in the constraint is approximated using the volume–averaged p–mean of the von–Mises stresses over all Gauss points. In case of linear shape functions and one Gauss point for each element, the p–mean norm is given by

$$\begin{aligned} P(\sigma_{vM}, p) &= \left[\frac{1}{\sum_{e=1}^{n_e} \int_{\mathcal{B}^e} \mathrm{d}V} \sum_{e=1}^{n_e} \int_{\mathcal{B}^e} [\sigma_{vM}^e]^p \mathrm{d}V \right]^{\frac{1}{p}} \\ &\approx \max_{e=1,\ldots,n_e} [\sigma_{vM}^e], \end{aligned} \tag{16}$$

where $p \in \{n \in \mathbb{N} \mid n/2 \in \mathbb{N}\}$ is a control parameter for the accuracy of the approximation and σ_{vM}^e the von–Mises stress evaluated at the Gauss point in element e. The node declaration for the optimization problem is visualized in Fig. 9a. Design nodes are marked in green. Yellow nodes are fixed in x– and z–direction and red nodes are fixed in y–direction. All remaining node coordinates are controlled. The optimization without rotational symmetry as a manufacturing constraint satisfies the stopping criterion after 19 iterations with the shape depicted in Fig. 9b. The volume has been reduced by 11.8 % but the optimized shape is not rotational symmetric anymore. The same optimization problem including the rotational symmetry constraint results in the



(a) (b)

Fig. 13 **a** Dirichlet boundary conditions for the load case at the *top side* of the pin bore and **b** *top and side* pressure

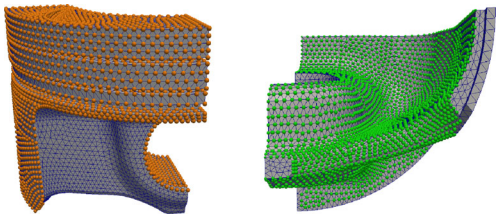


Fig. 14 boundary conditions for the optimization problem: **a** fixed nodes and **b** design nodes

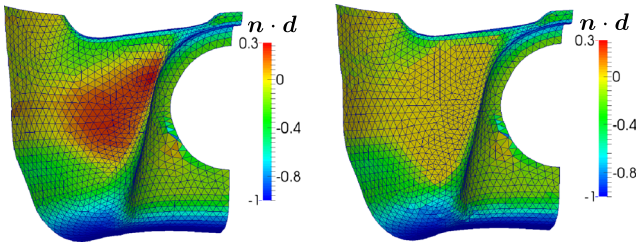


Fig. 16 visualization of the demolding constraint on the design space of the piston quarter model **a** without and **b** with molding constraint

optimized design illustrated in Fig. 10b. Thereby the design nodes have been reduced from 743 to 19 optimization nodes, marked in red in Fig. 10a. The procedure still takes 19 iterations to terminate. In this case, the volume has been reduced by 5.8 % and the new shape is still rotational symmetric. In Fig. 11a and b the convergence plots for both optimizations are shown, where the dashed lines in (a) illustrate the scaled volume $\frac{V}{V^0}$ and the solid lines include the penalizing fictitious energy $\frac{V}{V^0} + \alpha \frac{\mathcal{I}}{V^0}$. For the constraints a tolerance of 10^{-6} is adjusted to be in the feasible domain. The behavior of the volume–averaged p–mean value is displayed in Fig. 11b. A summarizing chart is depicted in Fig. 11c.

4.2 Piston

The next example is a quarter model of a piston with 33700 elements and 8657 nodes. The model and the load case are provided by courtesy of our project partner Federal–Mogul Nürnberg GmbH.

The nodes in Fig. 12 indicate symmetry boundary conditions for the quarter model, i.e. the yellow nodes in Fig. 12a are fixed in x–direction and the yellow nodes in Fig. 12b are fixed in y–direction. The presented load case is depicted in Fig. 13. The pressure of 110 MPa on the piston head up to the annular groove originates from the ignition in the combustion chamber. Due to the skew position of the piston rod and the resulting contact of the piston to the cylinder wall, a second loading acts on the piston skirt which has been approximated through a static FE–analysis (see Fig. 13b). Moreover, as illustrated in Fig. 13a, the nodes on the top side of the piston bore are fixed in x– and z–direction due to the contact to the piston rod for the force transmission. Young's modulus and Poisson's ratio are set to $E = 209000$ MPa and $\nu = 0.3$.

For the optimization problem, the crown, the side and the bore of the piston are completely fixed (see Fig. 14a). Nodes on the symmetry planes are fixed in normal direction to the plane but may be updated by the mesh movement algorithm within the plane. The nodes on the bottom of the piston are declared as design nodes as illustrated in Fig. 14b. The objective for this optimization problem is to minimize the maximum von–Mises stress, which we approximate by the volume–averaged p–mean of the von–Mises stresses over all Gauss points (16). Furthermore we apply a volume constraint such that the problem reads

$$\begin{aligned} \min \quad & \frac{P(\sigma_{vM}, p)}{P(\sigma^0_{vM}, p)} + \alpha \frac{\mathcal{I}}{V^0} \\ \text{subject to} \quad & V - V^0 \leq 0 \end{aligned}$$

where V^0 is the volume in the initial configuration. The maximum von–Mises stress distribution of the initial geometry is displayed in Fig. 15a. After 51 iterations, the maximum von–Mises stress was reduced by 8.9 %. As visible in

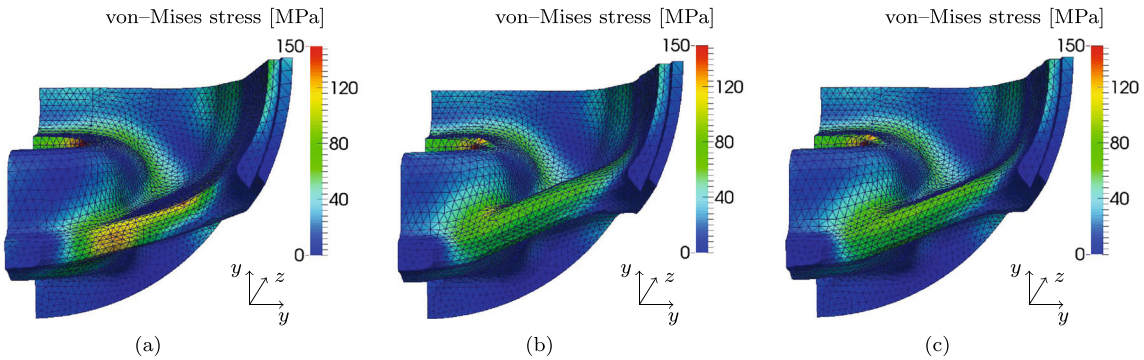
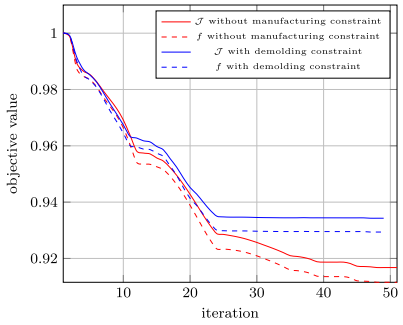


Fig. 15 Optimization results for the quarter model of a piston: stress distribution in the **a** initial design, **b** optimized without demolding constraints and **c** optimized with demolding constraints

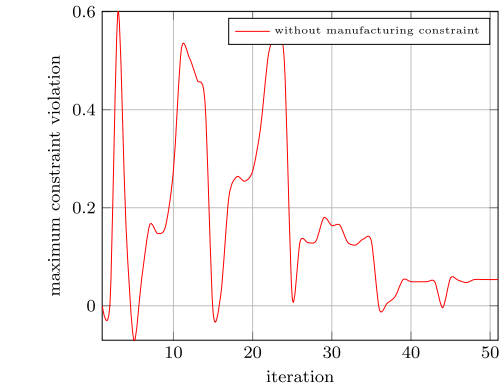


(a)

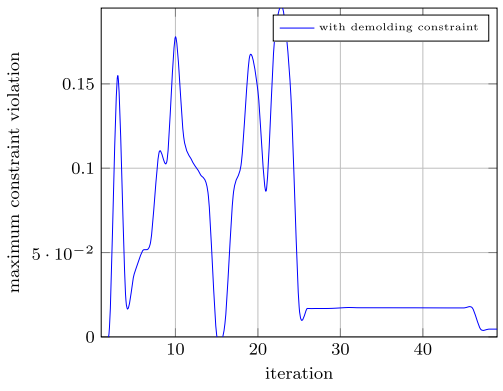
# elements	337000	
# nodes	8657	
demoldable	no	yes
# constraint eq.	1	3020
# iterations	51	49
decrease $\mathcal{J}$	$\sim 8.4\%$	$\sim 6.6\%$
decrease f	$\sim 8.9\%$	$\sim 7.1\%$
run time [h]	5.3	20.5

(b)

Fig. 17 **a** convergence plot and **b** summarizing table for the quarter model of a piston



(a)



(b)

Fig. 18 Constraint violation for the quarter model of a piston **a** without and **b** with demolding constraints

Fig. 15b an indentation occurs which will cause problems for the production process. The piston is manufactured by a casting process and therefore needs to be pulled out of the casting mould in z–direction which is not possible any more for the optimized design. Hence we incorporate a demolding constraint for each surface element in the design area and therefore append 3019 additional inequality constraints.

The maximum von–Mises stress has been reduced by 7.1 % in 49 iterations and the final shape is illustrated in Fig. 15c. In Fig. 16 the molding constraint is visualized for the optimization result (a) without and (b) with demolding constraint. The convergence plots for both optimizations as well as the behavior of the maximum constraint throughout the iterations for the demolding constraint can be observed in Figs. 17a and 18. The dashed and solid lines in Fig. 17a show the scaled maximum von–Mises stress $\frac{P(\sigma_{vM},p)}{P(\sigma^0_{vM},p)}$ and the penalized objective function $\frac{P(\sigma_{vM},p)}{P(\sigma^0_{vM},p)} + \alpha \frac{\mathcal{I}}{V^0}$ respectively, where the maximum is approximated by the volume–averaged p–mean function (16). A summarizing table is shown in (17) (b).

5 Conclusion

We introduced a novel approach to handle manufacturing constraints in node–based shape optimization, in which the design node–coordinates are split into optimization nodes and dependent nodes. The optimization nodes are declared as new design variables and the dependent nodes are related to the new design variables such that the manufacturing requirement is fulfilled. The advantages of this method are that no additional equations have to be considered in the optimization problem and the number of design variables is reduced. This new method can be combined with explicit or further implicit constraints.

Acknowledgments The funding of this research by the Federal Ministry of Education and Research in the project shapeOpt2CAD is gratefully acknowledged.

As a result of our collaboration the demolding constraint as well as the reflection and rotational symmetry are also available in SIMULIA Tosca Structure, a software system for non–parametric structural optimization by Dassault Systems.

References

Ahn HK, de Berg M, Bose P, Cheng SW, Halperin D, Matouek J, Schwarzkopf O (1997). In: Separating an object from its cast. Proceedings of the thirteenth annual symposium on Computational geometry - SCG 97. doi:10.1145/262839.262974

Allaire G, Jouve F, Michailidis G (2013) Casting constraints in structural optimization via a level-set method. In: WCSMO-10, Orlando

Allaire G, Jouve F, Michailidis G (2014) Thickness control in structural optimization via a level set method, preprint

Gill PE, Murray W, Saunders MA (2005) Snopt: an sqp algorithm for large–scale constrained optimization. SIAM Rev 47(1):99–131

Guest JK (2008) Imposing maximum length scale in topology optimization. Struct Multidiscip O 37(5):463–473. doi:10.1007/s00158-008-0250-7

Harzheim L, Graf G (2002) Topshape: an attempt to create design proposals including manufacturing constraints. Int J Vehicle Des 28(4):389–409. doi:10.1504/ijvd.2002.001997

Harzheim L, Graf G (2005a) A review of optimization of cast parts using topology optimization i - topology optimization without manufacturing constraints. Struct Multidiscip O 30(6):491–497. doi:10.1007/s00158-005-0553-x

Harzheim L, Graf G (2005b) A review of optimization of cast parts using topology optimization ii-topology optimization with manufacturing constraints. Struct Multidiscip O 31(5):388–399. doi:10.1007/s00158-005-0554-9

Hsu YL, Sheppard S, Wilde D (1995) The curvature function method for two-dimensional shape optimization under stress constraints. Comput Struct 55(4):647–657. doi:10.1016/0045-7949(94)00490-t

Hughes TJR (1987) The finite element method : linear static and dynamic finite element analysis. Dover, New York

Michailidis G (2014) Manufacturing constraints and multi-phase shape and topology optimization via a level-set method. PhD thesis, Centre de Mathematiques Appliquees, Ecole Polytechnique (X), Palaiseau, France. http://pastel.archives-ouvertes.fr/pastel-00937306

Perez RE, Jansen PW, Martins JRRA (2011) pyOpt: a python-based object-oriented framework for nonlinear constrained optimization. Struct Multidiscip O 45(1):101–118. doi:10.1007/s00158-011-0666-3

Poon NMK, Martins JRRA (2006) An adaptive approach to constraint aggregation using adjoint sensitivity analysis. Struct Multidiscip O 34(1):61–73. doi:10.1007/s00158-006-0061-7

Scherer M, Denzer R, Steinmann P (2009) A fictitious energy approach for shape optimization. Int J Numer Meth Eng 82(3):269–302. doi:10.1002/nme.2764

Schmitt O, Steinmann P (2015) On curvature control in node–based shape optimization. In: PMM-J Appl Math Mec+. (submitted)

Schmitt O, Friederich J, Steinmann P (2014) Manufacturing constraints in parameter–free sensitivity–based shape optimization. In: Proceedings of the 1st international conference on engineering and applied sciences optimization, Kos Island, pp 2532–2542

Sigmund O (2009) Manufacturing tolerant topology optimization. Acta Mech Sin 25(2):227–239. doi:10.1007/s10409-009-0240-z

Wu Z (2007) On the optimization problem of fillets and holes in plates with curvature constraints. Struct Multidiscip O 35(5):499–506. doi:10.1007/s00158-007-0139-x

Xia Q, Shi T, Wang MY, Liu S (2009) A level set based method for the optimization of cast part. Struct Multidiscip O 41(5):735–747. doi:10.1007/s00158-009-0444-7

Yao T, Choi KK (1989) 3-D shape optimal design and automatic finite element regridding. Int J Numer Meth Eng 28(2):369–384. doi:10.1002/nme.1620280209