Computer Aided Geometric Design 25 (2008) 470–488

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Shape sensitivity of constructively represented geometric models

Jiaqin Chen *, Michael Freytag, Vadim Shapiro

Spatial Automation Laboratory, 1513 University Avenue, University of Wisconsin at Madison, Madison, WI 53706, USA

ARTICLE INFO

Article history:
Received 18 August 2007
Received in revised form 29 December 2007
Accepted 6 January 2008
Available online 11 June 2008

Keywords:
Shape sensitivity
Constructive representation
Implicit representation
R-functions

ABSTRACT

Most solid models are archived using boundary representations, but they are created, edited, and optimized using high level constructive methods that rely on parameterized Boolean set operations and feature-based techniques. Downstream applications often require optimization of integral-valued performance measures over such models that include volume, mass, and energy properties, as well as more general distributed fields (stress, temperature, etc.). A key computational utility in all such applications is the computation of the sensitivity of the performance measure with respect to the parameters in the solid's construction history.

We show that for a class of performance measures defined as domain integrals, the sensitivity with respect to a parameter requires integration over a subset of the solid's boundaries that is affected by that parameter. In contrast to earlier methods, the proposed approach for computing sensitivities does not require solid's boundary to remain homeomorphic, and may be used with most types of constructive representations, including CSG and feature-based representations, where the defining Boolean expression may not be known. The simplicity and effectiveness of the proposed technique are illustrated on several common shape optimization problems.

1. Introduction

1.1. Motivation

Computer analysis and simulation of digital models is now performed routinely in an effort to predict and *improve* the performance of engineering artifacts before the designs are finalized and built. The procedure for changing the form of the solid model based on the results of the analysis is known as shape optimization: finding an optimal shape to improve a certain performance measure under some constraints. The shape optimization process itself is driven by shape (design) *sensitivity* that is broadly conceived as the derivative of a *performance measure* with respect to *geometric parameters* that define the solid shape. Formally, a performance measure is quantified as a (typically) real-valued *objective function* or *cost function*, and the computation of its sensitivities provides the rigorous basis for most deterministic optimization algorithms.

In this paper we consider the shape sensitivity computation of a general class of objective functions which can be expressed as a domain integral $F(\Omega) = \int_\Omega f\, d\Omega$, where F is the objective function, f is the integrand function which we assume to be differentiable through the paper, and Ω is the geometric (solid) domain. Many common performance measures are represented in this form. For example, volume is the integral of a unity function over the domain, mass is the integral of the density function, moment of inertia is defined as $I = \int_\Omega \rho r^2\, d\Omega$ where ρ is the density and r is the distance to the rotational axis. Other physical quantities can be written as elementary functions of integral-valued functions; for

* Corresponding author.
E-mail addresses: jiaqinchen@wisc.edu (J. Chen), freytag@cae.wisc.edu (M. Freytag), vshapiro@engr.wisc.edu (V. Shapiro).

0167-8396/$ – see front matter
doi:10.1016/j.cagd.2008.01.005

example, the center of gravity can be written as $\frac{\int_\Omega \vec{x}\rho(\vec{x})\,d\Omega}{\int_\Omega \rho(\vec{x})\,d\Omega}$, where $\vec{x}$ is the coordinate and $\rho(\vec{x})$ is the density function. An important class of performance measures involves integrals of some field functions. For example, compliance is defined as the total strain energy of a domain, the average temperature is the integral of a temperature field divided by the volume, etc. Many engineering design problems and physically based approaches to geometric modeling/graphic tasks lead to a shape optimization problem with the objective function and/or constraint functions in this form. For example, a structural design problem may seek the optimal shape with the highest stiffness, which can be written as a compliance minimization problem subject to a weight constraint (Bendsøe and Sigmund, 2003). Volume preserving deformation problems (Rappoport et al., 1995; Hirota et al., 1999) can be formulated as a deformation energy minimization problem subject to the volume constraint.

Computing mass properties (including volume, inertia, areas, etc.) over constructive representations of solids and their boundaries was one of the first applications of solid modeling (Lee and Requicha, 1982a, 1982b; Sarraga, 1982). Similar techniques may be used to compute integrals of more general functions (Freytag and Shapiro, 2004). Except in special situations, the integral functions are computed by discretizing the solid's interior and/or boundary. However, the geometric parameters of interest in shape optimization appear explicitly only as parameters in geometric *primitives* $\{\Omega_i, i = 1, \ldots, N\}$ of some *constructive representation* of the solid Ω. Such constructive representations include Constructive Solid Geometry, feature-based representations, and other procedural constructions of solids. Typical geometric parameters in constructive representations include sizes, radii, dimensions, angles, distances, control points, and so on. A constructive representation completely defines the solid Ω implicitly, and is sufficient to construct the boundary representation, $\partial\Omega$, via a suitable boundary evaluation procedure. However, every change of geometric parameters results in a boundary evaluation and may change the solid therefore change the objective function, it should also be clear that the dependence of any objective function $F(\Omega) = \int_\Omega f\,d\Omega$ on geometric parameters is also implicit. Thus, if b is a geometric parameter of some primitive Ω_i, it is not clear how the shape sensitivity $\frac{dF}{db}$ may be computed.

1.2. Contributions

In this paper, we solve the shape sensitivity problem for constructive representations with a set of primitives $\{\Omega_i\}$. We will assume that the representation of a primitive Ω_i is sufficient to determine its boundary $\partial\Omega_i$ and to compute its dependence with respect to any parameter b that affects its shape. In particular, if a primitive Ω_i is represented *implicitly* by a function Φ_i, then this dependence is reflected by $\frac{\partial\Phi_i}{\partial b}$. For convenience, we will focus mostly on such implicitly defined primitives, but extensions to free-form and mesh-based primitives are relatively straightforward and will be discussed in Section 6. Specific contributions of the paper are as follows:

- We provide a rigorous formulation of shape sensitivity that applies to any constructive representation satisfying the above assumptions. It does not assume or require knowledge of the Boolean expression or other details of the construction procedure. Consistency with the classical formulation of shape sensitivity is also shown.
- The formulation does not place artificial restrictions on the topology of the represented solid, Ω, or structure of its boundary representation, allowing to take full advantage of the versatility of constructive representations during shape optimization.
- The proposed approach to computing sensitivity leads to integration over the primitive boundaries. This suggests an efficient implementation of the method via localized algorithms, without requiring full boundary evaluation.
- Advantages and limitations of the method are explained and demonstrated by several implemented examples of shape optimization.

1.3. Previous work and state of the art

A classical result from shape (design) sensitivity analysis (SSA) shows that, for an objective function expressed as domain integration, the shape sensitivity can be expressed as a summation of two terms with one of them a domain integration and the other a boundary integration (Haug et al., 1986; Sokolowski and Zolesio, 1992; Choi and Kim, 2005a). We will give an example of SSA in Section 2. The formulation relies on a differentiable mapping between the original domain and the deformed domain and is constructed in terms of the shape (design) velocity, which determines the boundary deformation as parameters change. Direct application of the SSA method to constructive representations is difficult, because it requires full boundary evaluation. No rigorous procedures for constructing the required velocity field are known. A typical assumption that the domain remains homeomorphic under the shape changes is difficult to implement computationally, and is usually translated into the requirement that the boundary representation remains isomorphic. These restrictions undermine the *raison d'être* of constructive representations.

Much of the earlier work on shape sensitivity focused on shape optimization in the context of structural or thermal analysis. In this case, a constructive representation is converted not into a boundary representation but into a finite element mesh. Application of SSA then becomes even more problematic, because it requires relating the finite element nodal coordinates to the geometric parameters. For example, the mesh parametrization method (Yang et al., 1992; Chang and Choi, 1992; Olhoff et al., 1992) generates the mesh from a set of master-points which are used as design

variables and constructs a mapping between master-points and geometric entities. This method is difficult to use as the master-points have to be defined by users and the mesh generated may be excessively distorted. Alternately, the natural design variable method (Belegundu and Rajan, 1988; Tortorelli, 1993) uses fictitious analysis on the finite element model to ensure the geometric variations satisfy the shape constraints. This method requires defining the fictitious boundary conditions and the shapes generated are not compatible with the solid modeling data structures. In the context of solid modeling, most shape optimization approaches rely on an automatic mesh generator and then directly relate the geometric dimensions with finite element nodes based on variational geometry analysis (Kodiyalam et al., 1992; Botkin, 1992; Chen and Tortorelli, 1997). Notably, in Chen and Tortorelli (1997) the authors proposed an approach to computing design velocity using an iso-parametric mapping for geometric entities, assuming that the relationship among variational parameters and boundary variables can be determined.

All of the above methods are limited by their ability to associate geometric parameters with the finite element and boundary representations, and suffer from the same artificial limitations on topological form during the shape optimization process. In practice, most CAD systems still use the finite difference method to approximate the shape sensitivity. As simple as it is, the finite difference method has three intrinsic drawbacks: first, the accuracy is controlled by the step size which is difficult to choose a priori; second, it is computationally expensive; third, the finite difference method may fail and return incorrect values at points where the function is non-differentiable, without any warning to the user (Keane and Nair, 2005). Another brute-force approach to sensitivity computation is to use automatic differentiation (Griewank and Corliss, 1991) of the programs for computing the performance measures. Despite its generality, the approach is clearly impractical, because it requires differentiating all source code (including boundary evaluation, integration, and field simulation) with respect to every (potentially hundreds or thousands) geometric parameter of interest.

1.4. Outline

The rest of the paper is organized as follows. We use a simple example in Section 2 to explain the SSA method for calculating the shape sensitivity and its limitations. In Section 3, we formulate the shape sensitivity using implicit representations as a theoretical tool and use it to derive shape sensitivity for constructive representations. We discuss how the technical assumptions in our analysis are translated into verifiable conditions on primitives in constructive representations. The consistency between our formulation and the classical SSA method is also shown. In Section 4, several simple examples are used to illustrate the basic ideas and the computations implied by the proposed method, as well as its applicability to shapes with changing topology. Section 5 shows fully implemented numerical examples that demonstrate some of the capabilities of the proposed method. In conclusion (Section 6), we explain the application of the proposed shape sensitivity analysis to constructive representations with free-form and variationally-defined primitives.

2. Shape sensitivity analysis (SSA) and computation

In this section, we first give a brief introduction to SSA (the details can be found in the classic books (Sokolowski and Zolesio, 1992; Haug et al., 1986)). Then we use a simple example to illustrate and discuss how the sensitivity is calculated from the principle of SSA.

2.1. Shape sensitivity analysis

The basic idea in SSA is to characterize the shape deformation as a domain mapping between the original domain and the deformed domain. Under appropriate regularity assumptions, a design velocity field can be constructed from this mapping and the sensitivity analysis is carried through this velocity field.

Given a domain $\Omega \subseteq \mathbb{E}^d$ with a geometric parameter b, Ω can be regarded as a mapping from $\mathbb{R}$ to $\mathbb{E}^d$, $\Omega : b \mapsto \Omega(b)$. Let δb denote a small variation of b, and $\Omega(b+\delta b)$ the deformed domain. Assume that there is a domain mapping T : $\mathbf{x}(b) \mapsto \mathbf{x}(b+\delta b)$, $\mathbf{x} \in \Omega$, between $\Omega(b)$ and $\Omega(b+\delta b)$, $\Omega(b+\delta b) = T(\Omega(b); \delta b)$. With sufficient regularity assumptions and using Taylor series expansion, we have

$$\begin{aligned}\mathbf{x}(b+\delta b) &= T\big(\mathbf{x}(b); \delta b\big)\\ &= T\big(\mathbf{x}(b); 0\big) + \delta b \frac{\partial T}{\partial(\delta b)}\bigg|_{\delta b=0} + o(\delta b)\\ &= \mathbf{x}(b) + \delta b \frac{d\mathbf{x}(b)}{db} + o(\delta b).\end{aligned}$$

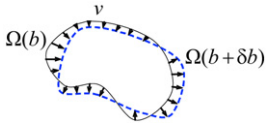

Fig. 1. The design velocity.

Defining the *design velocity* (see Fig. 1) as

$$\vec{v} = \frac{d\mathbf{x}}{db} = \frac{\partial T}{\partial(\delta b)}\bigg|_{\delta b=0} \tag{1}$$

and ignoring the high-order terms in the Taylor expansion, we have $\mathbf{x}(b+\delta b) = \mathbf{x}(b) + \vec{v}\delta b$. The Jacobian, J, of this transformation is $J = \nabla T = I + \delta b \nabla(\vec{v})$. For the objective function $F(b) = \int_{\Omega(b)} f(b, \mathbf{x}(b))\, d\Omega$, f is a differentiable function,

$$\begin{aligned} F(b+\delta b) &= \int_{\Omega(b+\delta b)} f\big(b+\delta b, \mathbf{x}(b+\delta b)\big)\, d\Omega \\ &= \int_{\Omega(b)} f\big(b+\delta b, \mathbf{x}(b) + \vec{v}\delta b\big)|J|\, d\Omega. \end{aligned}$$

So

$$\begin{aligned} \frac{dF}{db} &= \lim_{\delta b\to 0} \frac{F(b+\delta b) - F(b)}{\delta b} = \int_{\Omega(b)} \left[\frac{df}{db} + f\frac{d|J|}{db}\right] d\Omega \\ &= \int_{\Omega(b)} \left[\frac{\partial f}{\partial b} + \nabla f^T \cdot \vec{v} + f\,\mathrm{div}(\vec{v})\right] d\Omega \\ &= \int_{\Omega(b)} \left[\frac{\partial f}{\partial b} + \mathrm{div}(f\vec{v})\right] d\Omega. \end{aligned} \tag{2}$$

Applying the divergence theorem to Expression (2), we have

$$\frac{dF}{db} = \int_{\Omega(b)} \frac{\partial f}{\partial b}\, d\Omega + \int_{\partial\Omega(b)} f v_n\, d\Gamma, \tag{3}$$

where v_n is the normal component of $\vec{v}$.

Expressions (2) and (3) are classical results for shape sensitivity analysis given in Haug et al. (1986). In both Expressions (2) and (3), the first term is a domain integration of $\frac{\partial f}{\partial b}$ on Ω. If f is an explicit function of b, then the computation of this term is straightforward. In some cases, this term may be difficult to calculate if $\frac{\partial f}{\partial b}$ is not directly obtainable. For example, the function f may be a field function which depends on the solution of some boundary value problem defined on Ω, such as temperature (in a heat transfer problem), displacement (in an elasticity problem), etc. There are various techniques for computing such derivatives $\frac{\partial f}{\partial b}$, for example, the adjoint method (Haug et al., 1986). The discussion of such issues is beyond the scope of this paper. Here we only point out that for a wide class of problems, the first term can also be written as a boundary integral. For example, the sensitivity analysis for a minimum compliance optimization problem is shown to be a boundary integral in Chen et al. (2007) and is used for shape optimization in a numerical example in Section 5.

In the rest of the paper, we will focus on the computation of the second term in Expressions (2) and (3). In Expression (2) the second term is a domain integration of $\mathrm{div}(f\vec{v})$ on Ω and in Expression (3), the second term is a boundary integration of $f v_n$ on $\partial\Omega$. In either case, the computation of these terms requires the knowledge of the design velocity $\vec{v}$ (or its normal component v_n), which can be challenging as we explain in the following.

2.2. SSA for a triangle example

We now use a simple example to illustrate how the SSA can be used to compute shape sensitivities. We consider the volume (area) function $V = \int_\Omega d\Omega$ defined on a right triangle Ω with two legs a and b as shown in Fig. 2. Assume b is the parameter subject to change, the goal is to compute the sensitivity $\frac{dV}{db}$. In this example, it is easy to see that $V = \frac{1}{2}ab$ and $\frac{dV}{db} = \frac{1}{2}a$ from basic knowledge of elementary geometry. It should be obvious that for most geometries, the analytic expression of the objective function is not known or requires tedious calculation (even for the simple volume function), and the derivative cannot be directly obtained.

Substituting $f \equiv 1$ into Expressions (2) and (3), we have

$$\frac{dV}{db} = \int_\Omega \mathrm{div}\left(\frac{d\mathbf{x}}{db}\right) d\Omega = \int_{\partial\Omega} \frac{d\mathbf{x}}{db} \cdot \vec{n}\, d\Gamma, \tag{4}$$

where $\vec{n}$ is the normal vector at boundary $\partial\Omega$.

We have two choices in computing the derivative: one is to use domain integration, the other is to use boundary integration:

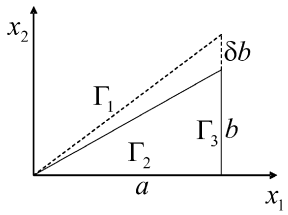

Fig. 2. The triangle with one leg b as the design variable.

(1) If we can construct the domain mapping T, then we can use the domain integration in Expression (2). In this simple example we can construct T by inspection as $T(x_1, x_2; \delta b) = (x_1, x_2) + \delta_b(0, \frac{x_2}{b})$. Then

$$\frac{d\mathbf{x}}{db} = \frac{\partial T}{\partial(\delta b)} = \left(0, \frac{x_2}{b}\right).$$

A direct calculation gives

$$\frac{dV}{db} = \int_{\Omega} \operatorname{div}\left(\frac{d\mathbf{x}}{db}\right) d\Omega = \int_{\Omega} \frac{1}{b}\, d\Omega = \frac{1}{2}a.$$

(2) From Expression (4) we see that knowing the design velocity and the normal vector of the boundary is enough to compute the shape sensitivity. For example, we may construct the velocity field $\vec{v}$ at the boundary as:

$$\vec{v} = \begin{cases} \left(0, \frac{x_1}{a}\right), & \text{if } x \in \Gamma_1, \\ (0, 0), & \text{if } x \in \Gamma_2, \\ \left(0, \frac{x_2}{b}\right), & \text{if } x \in \Gamma_3. \end{cases}$$

The corresponding normal vector is

$$\vec{n} = \begin{cases} \left(-\frac{b}{\sqrt{a^2+b^2}}, \frac{a}{\sqrt{a^2+b^2}}\right), & \text{if } x \in \Gamma_1, \\ (0, -1), & \text{if } x \in \Gamma_2, \\ (1, 0), & \text{if } x \in \Gamma_3. \end{cases}$$

So, the derivative is

$$\begin{aligned} \frac{dV}{db} &= \int_{\partial\Omega} \frac{d\mathbf{x}}{db} \cdot \vec{n}\, d\Gamma = \int_{\Gamma_1+\Gamma_2+\Gamma_3} \vec{v} \cdot \vec{n}\, d\Gamma \\ &= \int_{\Gamma_1} \frac{x_1}{\sqrt{a^2+b^2}}\, d\Gamma = \frac{1}{2}a. \end{aligned}$$

As we see, the divergence theorem tells us that only the normal component of the design velocity at the boundary contributes to the sensitivity. Thus, we only need to know the normal velocity v_n at the boundary. A simplified calculation for the sensitivity may be obtained by directly constructing a normal velocity at the boundary if possible. In this example, if we construct v_n as

$$v_n = \begin{cases} \frac{x_1}{\sqrt{a^2+b^2}}, & \text{if } x \in \Gamma_1, \\ 0, & \text{if } x \in \Gamma_2, \\ 0, & \text{if } x \in \Gamma_3, \end{cases}$$

then we have

$$\frac{dV}{db} = \int_{\partial\Omega} v_n\, d\Gamma = \int_{\Gamma_1} \frac{x_1}{\sqrt{a^2+b^2}}\, d\Gamma = \frac{1}{2}a.$$

Through this simple example, we see that using the SSA theory at least requires a boundary velocity field (or its normal component) that describes the boundary variations with parameter changes. If the boundary has explicit parametrization, for example parametric curves or surfaces, the velocity field may be obtainable.

When a solid Ω is defined constructively, its boundary $\partial\Omega$ is a union of faces that are trimmed portions of primitive boundaries $\partial\Omega_i$. In this case, it is not obvious how to construct the design velocity $\vec{v}$. Furthermore, the SSA formulation

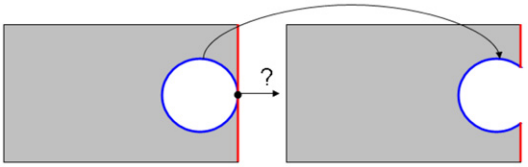

Fig. 3. A rectangle with a circular hole. When the topology changes, the velocity of singular points becomes unidentifiable.

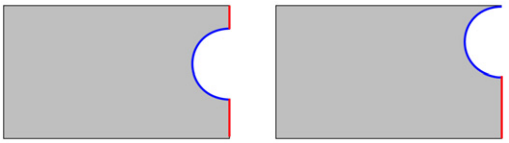

Fig. 4. A rectangle with a circular hole. When the boundary representation changes, the construction of a one-to-one mapping between the boundaries becomes challenging.

usually assumes that the topology of the domain does not change during the optimization process, undermining one of the main strengths of the constructive representations which is allowing effortless topological changes. Consider a simple example in Fig. 3. Both shapes are defined by the same constructive representation as the set difference of a rectangle and a circular disk, but the geometries and their boundaries are not homeomorphic and it is not clear if the design velocity can be defined using the SSA approach. Fig. 4 shows two more members of the same constructive representation, and the two objects *are* homeomorphic, but the red boundary segment on the right is split into two components on the left.[1] This further complicates the construction of the design velocity. Even if these formulation problems are solved in the SSA method, a typical implementation in a solid modeling system would require tracking the mapping between the corresponding portions of the primitive boundaries. This problem is known as *persistent naming* in the literature (Raghothama and Shapiro, 1997) and remains unsolved as of now. It is not surprising then that virtually all practical implementations of the SSA method in commercial boundary representation systems require that the boundary representation of the domain remains isomorphic under the deformation process, severely limiting the effectiveness of the optimization procedure.

3. Proposed approach

The difficulty in SSA lies in the construction of the velocity field, especially when the topology changes. We now show that a much simpler sensitivity computation method can be used for constructive representations without tracking the global boundary movements. Our method can easily handle topological changes and it does not need to know or maintain the boundary representation.

We first formulate the shape sensitivity analysis through the tool of implicit representations and discuss validity conditions of the given sensitivity formulation, including the differentiability of the implicit representation and the differentiability of the objective function. Then we present sufficient conditions for validating the shape sensitivity analysis in our formulation, and show that these conditions can be further localized and the shape sensitivity can be computed efficiently as an integral over a subset of boundaries which are affected by the parameter. The consistency between our formulation and the SSA formulation in the restricted form where topology does not change is also shown.

We choose the implicit representation for the geometry to derive the sensitivity analysis because of its representational advantages: the implicit representation uses functions to represent the geometry and there is no need to track each individual boundary piece during shape deformations. However, we emphasize that the implicit representation is only a convenient theoretical tool we use for formulating the sensitivity analysis. We do not need to know or construct such an implicit representation and the resulting sensitivity should not by any means depend on any particular representation.

3.1. Sensitivity formulation through implicit representation

Given a geometric domain Ω with boundary $\partial\Omega$, suppose there exists a differentiable function Φ with non-zero gradient such that $\Omega = \{x \mid \Phi(x) \geqslant 0\}$ and $\partial\Omega = \{x \mid \Phi(x) = 0\}$. Assume that b is a parameter of Ω, then Φ is also a function of b. Let D be a fixed domain that contains Ω and its variations, if we use the following characteristic (Heaviside) function

$$H\big(\Phi(x)\big) = \begin{cases} 1, & \text{if } \Phi(x) \geqslant 0, \\ 0, & \text{if } \Phi(x) < 0 \end{cases}$$

as an indicator of whether a given point x belongs to Ω or not, we have

$$\Omega = \big\{x \mid x \in D, \Phi(x) \geqslant 0\big\} = \big\{x \mid x \in D, H(\Phi) = 1\big\}.$$

[1] For colors see the web version of this article.

Employing the characteristic function, the objective function F defined on Ω can be transformed to a function defined on the reference domain:

$$F = \int_{\Omega} f \, d\Omega = \int_{D} f H(\Phi) \, d\Omega. \tag{5}$$

Using the chain rule and the fact that $\frac{dH(\Phi)}{d\Phi} = \frac{\delta(\Phi)}{|\nabla\Phi|}$, where $\delta(\cdot)$ is the Dirac Delta function, the derivative of F with respect to the parameter b is

$$\begin{aligned}\frac{dF}{db} &= \int_{D} \left[\frac{df}{db} H(\Phi) + f \frac{dH(\Phi)}{db} \right] d\Omega \\ &= \int_{D} \left[\frac{df}{db} H(\Phi) + f \frac{\delta(\Phi)}{|\nabla\Phi|} \frac{d\Phi}{db} \right] d\Omega.\end{aligned}$$

The total differential operators may be replaced by partial differential operators since they are defined on the fixed domain D, so we have

$$\begin{aligned}\frac{dF}{db} &= \int_{D} \left[\frac{\partial f}{\partial b} H(\Phi) + f \frac{\delta(\Phi)}{|\nabla\Phi|} \frac{\partial\Phi}{\partial b} \right] d\Omega \\ &= \int_{\Omega} \frac{\partial f}{\partial b} \, d\Omega + \int_{\partial\Omega} \frac{f}{|\nabla\Phi|} \frac{\partial\Phi}{\partial b} \, d\Gamma.\end{aligned} \tag{6}$$

Expression (6) gives the derivative of F with respect to b *if* it exists. We had assumed the following two conditions for Expression (6) to be valid:

1. The implicit representation function Φ is differentiable with non-zero gradient.
2. The objective function F is differentiable.

We now further analyze these two assumptions.

3.2. Assumption 1: Φ is differentiable with non-zero gradient

We had assumed that the implicit representation function Φ is differentiable (with respect to b and spatial variables) on Ω in deriving Expression (6). We also assumed than Φ has non-zero gradient so that the integrand function $\frac{f}{|\nabla\Phi|} \frac{\partial\Phi}{\partial b}$ in Expression (6) is bounded. But in general it may be difficult or impossible to construct the analytic expression of this function Φ satisfying these requirements. More importantly, we *do not* need to construct such functions even if they exist. Notice that the properties of Φ do not matter at the interior points of Ω after we use the characteristic function $H(\Phi)$, so the existence of a function Φ that is differentiable with non-zero gradient *at the boundary* is enough to validate the sensitivity formulation. In fact, the second term in Expression (6) is a boundary integration and we know that the set of measure zero at the boundary does not affect the boundary integration. So the properties of Φ can be relaxed to be differentiable with non-zero gradient *almost everywhere*[2] *at the boundary*. Therefore, the sensitivity analysis in Expression (6) remains valid when the differentiability of Φ everywhere with non-zero gradient in Ω is relaxed to the differentiability of Φ with non-zero gradient *almost everywhere at the boundary* $\partial\Omega$.

Let $\{\Phi_i, i = 1, \ldots, N\}$ be the functions implicitly defining primitives $\{\Omega_i, i = 1, \ldots, N\}$ respectively, Ω_i is the point set which satisfies $\Omega_i = \{x \mid \Phi_i(x) \geqslant 0\}$. For the geometry Ω constructed from $\{\Omega_i\}$ by Boolean set operations, a single implicit representation Φ for Ω can be constructed by many methods, for example, logical operations on inequalities $\{\Phi_i \geqslant 0\}$.

Nevertheless, the constructed function Φ has to satisfy the differentiability requirement. One way to construct such a single analytic differentiable function $\Phi = \Phi(\Phi_1, \ldots, \Phi_N)$ is to use the theory of R-functions. As is well known, the differentiability of functions constructed using R-functions depends on the particular Boolean set expression used to combine the primitives (Rvachev, 1982). Given a constructively represented geometry Ω, we assume that there exists a Boolean set expression in which each primitive Ω_i only appears once. Then by the theory of R-functions, we can guarantee that the constructed $\Phi = \Phi(\Phi_1, \ldots, \Phi_N)$ for this set expression is m-times differentiable and inherits the normal properties of corresponding Φ_i at each boundary point except those points where more than one primitive is zero. In other words, if we assume that *every point on* $\partial\Omega$ *belongs to the boundary of exactly one primitive* (*if* $\Phi_k(x) = 0$, *then* $\Phi_i(x) \neq 0, \forall i \neq k, \forall x \in \partial\Omega$) *except at a set of measure zero*, then the existence of a differentiable implicit representation almost everywhere at the boundary is guaranteed by the theory of R-functions. (Hence we exclude set complement operations because a primitive and its complement have the same boundary.) The constructions using R-functions which are sufficiently smooth and

[2] *Almost everywhere* can be determined rigorously by the measure-zero condition as we will explain later.

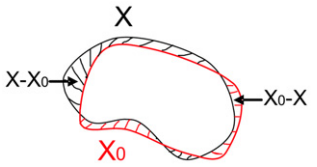

Fig. 5. The symmetric difference between two geometries.

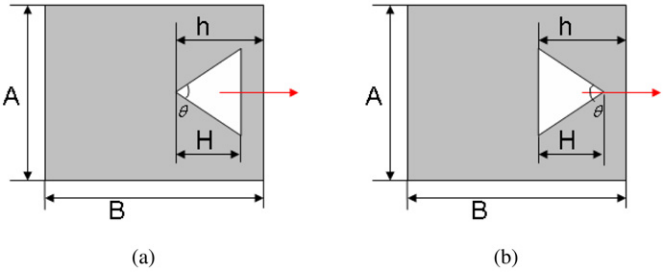

Fig. 6. Two examples to illustrate the $\mathscr{M}$-continuity of the boundary: (a) The boundary is not M-continuous when $h = H$; (b) The boundary is $\mathscr{M}$-continuous when $h = H$.

preserve normalization on the boundary have been well studied in the literature (Rvachev, 1967, 1982; Shapiro, 1988; Shapiro and Tsukanov, 1999).

In many situations, we may not know the Boolean set expression for a given Ω, which may seemingly restrict the application of R-functions to construct Φ. We will show in Section 3.4 that we only need Φ_i's in computing the sensitivity and Φ need not be constructed. Therefore, the information of the underlying Boolean set expression for Ω is not required.

3.3. Assumption 2: F is differentiable

As we know, for a function to be differentiable at a point, the derivative of the function must be continuous at that point. While it is generally difficult to characterize the differentiability of an objective function in geometric design problems, we may derive sufficient conditions for determining the differentiability of F.

Since f is a differentiable function, $\frac{\partial f}{\partial b}$ is a continuous function. If Assumption 1 is satisfied, then the integrand function $\frac{f}{|\nabla\Phi|}\frac{\partial\Phi}{\partial b}$ is also a continuous function. Observe from Expression (6) that if the integrand functions are continuous, then $\frac{dF}{db}$ is continuous if the geometry Ω and its boundary $\partial\Omega$ are "continuous" in the sense that its measure does not change much. This observation motivates the following definition of $\mathscr{M}$-continuity[3] ($\mathscr{M}$ stands for measure).

Definition 3.1. A point set $X : c \mapsto X(c)$ is $\mathscr{M}$-continuous at c_0 if $\int_{X\Delta X_0} dX \to 0$ as $c \to c_0$, where $X_0 = X(c_0)$ and $X\Delta X_0 \equiv (X - X_0) \cup (X_0 - X)$ is the symmetric difference between X and X_0.

Fig. 5 shows the symmetric difference between two geometries X_0 and X in two dimension. Intuitively, a geometry $X(c)$ is said to be $\mathscr{M}$-continuous at parameter c_0 if the area (or length in 1D, volume in 3D) of the symmetric difference between the original domain and the deformed domain goes to zero when c goes to c_0. While the $\mathscr{M}$-continuity of Ω is satisfied for most geometric deformation problems, we often see the boundary $\partial\Omega$ not being $\mathscr{M}$-continuous in many situations (especially when the topology of the geometry changes). A small perturbation of a parameter may cause sudden changes of the boundary. For example, Fig. 6(a) shows a geometry which consists of a rectangle and a triangular hole. As the position h of the triangle changes, it eventually hits the rectangle ($h = H$). Each point on the right edge of the triangle becomes an exterior point, and we can check that the $\mathscr{M}$-continuity is not satisfied for the boundary. In contrast to Fig. 6(a), the $\mathscr{M}$-continuity is satisfied for the boundary of the geometry shown in Fig. 6(b).

Using the tool of $\mathscr{M}$-continuity, we derive the following sufficient condition to guarantee the differentiability of F:

Theorem 3.2. *If there exists an implicit representation function Φ for Ω that is differentiable with non-zero gradient almost everywhere at the boundary $\partial\Omega$, then F defined in Expression* (5) *is differentiable at b_0 if Ω and $\partial\Omega$ are $\mathscr{M}$-continuous at b_0.*

Proof. With the assumption that f and Φ are differentiable at b_0, and the gradient of Φ is non-zero, the integrands in Expression (6) are bounded continuous functions. Then we can conclude from Lemma 3.3 that F is differentiable. □

[3] We used $\mathscr{G}$-continuity in place of $\mathscr{M}$-continuity in our earlier publication (Chen et al., 2007). The terminology is changed here in order to avoid the confusion with the concept of geometric continuity of curves and surfaces, which is standard in computer aided geometric design literature.

Lemma 3.3. *If a point set $X : c \mapsto X(c)$ is $\mathscr{M}$-continuous at c_0, then any function $G = \int_X g\,dX$ is continuous at c_0 for any bounded continuous function g defined on X.*

Proof. This can be easily verified by using the triangle inequality

$$\begin{aligned}
|G - G_0| &= \left| \int_X g\,dX - \int_{X_0} g_0\,dX \right| \\
&\leqslant \left| \int_X g\,dX - \int_{X_0} g\,dX \right| + \left| \int_{X_0} g\,dX - \int_{X_0} g_0\,dX \right| \\
&= \left| \int_{X-X_0} g\,dX - \int_{X_0-X} g\,dX \right| + \left| \int_{X_0} (g - g_0)\,dX \right| \\
&\leqslant \int_{X-X_0} |g|\,dX + \int_{X_0-X} |g|\,dX + \int_{X_0} |g - g_0|\,dX \\
&= \int_{X\Delta X_0} |g|\,dX + \int_{X_0} |g - g_0|\,dX \\
&\leqslant \max|g| \int_{X\Delta X_0} dX + \int_{X_0} |g - g_0|\,dX.
\end{aligned}$$

Since g is a bounded function and X_0 is a bounded point set, $\int_{X\Delta X_0} dX \to 0$ and $g \to g_0$ as $c \to c_0$, we have that the right side of the above equation goes to zero when $c \to c_0$. Therefore $G \to G_0$ when $c \to c_0$. □

Theorem 3.2 gives the sufficient condition to check the differentiability of the objective function regardless of the topological properties of the geometry. If the $\mathscr{M}$-continuity is not satisfied, then the differentiability of F becomes indeterminable. For example, we consider the volume V as the objective function for the geometries shown in Fig. 6. It can be shown that both geometries have an implicit representation function Φ that is differentiable with non-zero gradient almost everywhere at the boundary (Shapiro and Tsukanov, 1999). In Fig. 6(b), V is differentiable at $h = H$ since the $\mathscr{M}$-continuity is satisfied at the boundary. But for the geometry shown in Fig. 6(a), we have

$$V = \begin{cases} AB - H^2 \tan\frac{\theta}{2} & \text{if } h > H, \\ AB - h^2 \tan\frac{\theta}{2} & \text{if } H \geqslant h. \end{cases}$$

The derivative of V is

$$\frac{dV}{dh} = \begin{cases} 0 & \text{if } h > H, \\ -2h \tan\frac{\theta}{2} & \text{if } H \geqslant h. \end{cases}$$

So V actually is not differentiable when $h = H$. Nevertheless, even if the objective function F is not differentiable, Expression (6) can still be used to compute the one-side derivative of F ($\frac{dF}{db}\big|_{b=b_0^+}$ or $\frac{dF}{db}\big|_{b=b_0^-}$) if one wishes.

3.4. Sensitivity localized for constructive representations

Although Expression (6) is the sensitivity formulation in terms of Φ, we can further show that the sensitivity can be decoupled to sensitivities on each primitive for constructive representations. If a point x_0 on $\partial\Omega$ only belongs to the boundary of one primitive Ω_k, a basic result from the theory of R-functions states that the derivative of the function Φ at x_0 is determined by the derivative of that primitive function, i.e. $\frac{\partial\Phi}{\partial b}\big|_{x_0} = \frac{\partial\Phi_k}{\partial b}\big|_{x_0}$. Therefore, denoting $\partial\Omega_k$ as the portion of boundary $\partial\Omega$ corresponding to a primitive Ω_k and assuming that the construction of Φ preserves the gradient of each primitive at the boundary, i.e. $\nabla\Phi|_{\partial\Omega_k} = \nabla\Phi_k|_{\partial\Omega_k}$, the second term in Expression (6) can be written as

$$\int_{\partial\Omega} \frac{f}{|\nabla\Phi|} \frac{\partial\Phi}{\partial b}\,d\Gamma = \sum_{k=1}^{N} \int_{\partial\Omega_k} \frac{f}{|\nabla\Phi_k|} \frac{\partial\Phi_k}{\partial b}\,d\Gamma. \tag{7}$$

We define the *active primitive* for parameter b to be any primitive depending on b, i.e. Φ_k is a function of b. Each parameter may have more that one active primitive. We use $\mathscr{A}(b) \subseteq \{1, \ldots, N\}$ to denote the index set of active primitives for parameter b. We refer to the set of boundary points which belong to some active primitive, i.e. $\bigcup_{k\in\mathscr{A}(b)} \partial\Omega_k$, as the *active boundary* for parameter b. Expression (7) can be further simplified as

$$\int_{\partial\Omega} \frac{f}{|\nabla\Phi|}\frac{\partial\Phi}{\partial b}\,d\Gamma = \sum_{k\in\mathscr{A}(b)} \int_{\partial\Omega_k} \frac{f}{|\nabla\Phi_k|}\frac{\partial\Phi_k}{\partial b}\,d\Gamma. \tag{8}$$

Expression (8) tells us that for each parameter *only* its active primitives contribute to the boundary integral and we can ignore the remaining primitives. Therefore for constructive representations, the previous conditions and assumptions can be further localized to be on the active boundaries as follows:

- The differentiability of Φ reduces to the differentiability on the active boundaries except at a set of measure zero. Therefore, we only need the assumption that (1) there exists a Boolean expression where all active primitives appear once, and (2) every boundary point belonging to some active primitive's boundary does not belong to any other primitive's boundary except at a set of measure zero.
- The differentiability of F: the $\mathscr{M}$-continuity condition for $\partial\Omega$ in Theorem 3.2 for F to be differentiable reduces to the $\mathscr{M}$-continuity of the active boundaries.

Substituting Expression (8) into Expression (6), we have

$$\frac{dF}{db} = \int_{\Omega} \frac{\partial f}{\partial b}\,d\Omega + \sum_{k\in\mathscr{A}(b)} \int_{\partial\Omega_k} \frac{f}{|\nabla\Phi_k|}\frac{\partial\Phi_k}{\partial b}\,d\Gamma. \tag{9}$$

In summary, the localized formulation in Expression (8) essentially only relies on the assumption that *any active boundary point only belongs to one primitive except at a set of measure zero*. As long as this assumption is satisfied during the shape deformation, the shape sensitivity can be localized to the sum of sensitivities on active primitives by Expression (9). This formulation places *no* restriction on the topological properties of the domain and is general enough for handling geometric deformation problems with topological changes.

The computational advantages of Expression (8) should be obvious. We only need to know the implicit representation Φ_k and its corresponding portion of the boundary $\partial\Omega_k$ for each active primitive Ω_k of parameter b. We do not need to know or compute the implicit function Φ for the whole geometry Ω. We do not need to know the full boundary information. The derivation of Expression (8) does not require the iso-morphism of the boundary representation and the boundary representation certainly does not need to be maintained. For constructive representations, each parameter typically has only a few active primitives, which makes the computation very efficient. The integration in Expression (8) over boundary pieces can be implemented through any suitable numerical integration techniques (combined with a point membership test). Combined with the computation of the first term in Expression (9) as briefly discussed at the end of Section 2.1, the shape sensitivity $\frac{dF}{db}$ can be obtained.

3.5. Sufficient conditions for valid sensitivity analysis

We now summarize the sufficient conditions for validating the shape sensitivity computations. First, if there exists a Boolean expression where all active primitives appear only once, and all the active boundary points except a set of measure zero only belong to one primitive, then we can construct an implicit representation function Φ such that it is differentiable with non-zero gradient almost everywhere at the active boundary. Second, if there exists a Φ which is differentiable with non-zero gradient almost everywhere at the active boundary, then the objective function F is differentiable if Ω and the active boundaries are $\mathscr{M}$-continuous.

These conditions are reasonable for most practical applications and can be verified computationally. The existence of Boolean expressions is present so that we may use the theory of R-functions to prove the existence of the implicit representation which satisfies the requirements. So it serves as a theoretical tool for convenience purposes rather than an essential requirement. The following two conditions essentially guarantee the valid shape sensitivity computation:

- **Measure-zero condition:** The active boundary points only belong to one primitive except a set of measure zero.
- **$\mathscr{M}$-continuity condition:** Ω and the active boundaries are $\mathscr{M}$-continuous.

These two conditions can be checked by computing the arrangement of primitives at the active boundaries. For example, a sufficient (but not necessary) condition that implies both measure-zero condition and $\mathscr{M}$-continuity condition is that primitives intersect transversely.

3.6. The consistency with the SSA method

The proposed shape sensitivity formulation using implicit representations is consistent with the classical SSA theory in the restricted situations where the topology does not change. To see this, we compare Expression (6) with Expression (3).

For each point $x \in \partial\Omega$, we have

$$\Phi(x)|_{\partial\Omega} = 0.$$

Using the chain rule, we have[4]

$$\left.\frac{d\Phi}{db}\right|_{\partial\Omega} = \left.\frac{\partial\Phi}{\partial b}\right|_{\partial\Omega} + \nabla\Phi \cdot \left.\frac{dx}{db}\right|_{\partial\Omega} = 0, \tag{10}$$

which implies that

$$\frac{\partial\Phi}{\partial b} = -\nabla\Phi \cdot \frac{dx}{db}, \quad \forall x \in \partial\Omega.$$

By the definition of design velocity from Expression (1), $\frac{dx}{db} = \vec{v}$, and the fact that $\vec{n} = -\frac{\nabla\Phi}{|\nabla\Phi|}$, we have

$$\frac{\partial\Phi}{\partial b} = \vec{v} \cdot \vec{n} \cdot |\nabla\Phi| = v_n |\nabla\Phi|.$$

Substitute the above equation into Expression (6), it is easy to see that Expression (6) is equivalent to Expression (3).

As we discussed in Section 2, the construction of the velocity field becomes a very challenging task when the topology or the boundary representation of the geometry changes. Our proposed shape sensitivity formulation does not have restrictions on topological properties of the geometry, and is much simpler and powerful yet consistent with the SSA method.

4. Illustrative examples

In the following, we first apply the proposed shape sensitivity formulation to two examples. We illustrate the computational procedure of the proposed approach and show that there are no artificial topological restrictions on the geometry once the sufficient conditions given in Section 3.5 are satisfied. Then we show several examples where the proposed approach does not apply and explain which condition they violate. As the existence of a Boolean expression and the $\mathscr{M}$-continuity for Ω are generally satisfied for most engineering shapes (and are easy to check in the following examples), we focus on discussing the measure-zero condition and the $\mathscr{M}$-continuity of the active boundaries. In these simple examples, whether or not the measure-zero condition and $\mathscr{M}$-continuity condition are satisfied is clear. For more complicate geometries, we may use computational procedures to verify these conditions automatically as explained in Section 3.5. For simplicity, in all examples we use the volume function $V = \int_\Omega d\Omega$ as the objective function. More realistic examples are shown in Section 5.

4.1. Examples with valid sensitivity analysis

Example 1 *(Shape deformation with no topological changes (Fig. 2)).* We use the triangle example in Section 2 to illustrate how the sensitivity can be calculated using our approach. The triangle can be constructed from the three half-spaces which are defined as:

$$\Phi_1 = bx_1 - ax_2,$$
$$\Phi_2 = x_2,$$
$$\Phi_3 = a - x_1.$$

The active primitive for parameter b is Φ_1 and its corresponding boundary is Γ_1. From Expression (9), for the volume function, $V = \int_\Omega d\Omega$, we have,

$$\frac{dV}{db} = \int_{\Gamma_1} \frac{1}{|\nabla\Phi_1|} \frac{d\Phi_1}{db}\, d\Gamma.$$

Since

$$\frac{d\Phi_1}{db} = x_1,$$
$$|\nabla\Phi_1| = \sqrt{a^2 + b^2},$$

we have

$$\frac{dV}{db} = \int_{\Gamma_1} \frac{x_1}{\sqrt{a^2 + b^2}}\, d\Gamma = \frac{1}{2}a.$$

Compared with the SSA method, the simplicity of our approach is obvious since there is no need to know the full boundary information once we know the active primitives.

[4] Eq. (10) is the so-called level set equation first given by Osher and Sethian (1988), which has been widely used in interface motion problems. The associated numerical techniques can track complex fronts that develop sharp corners or topological changes.

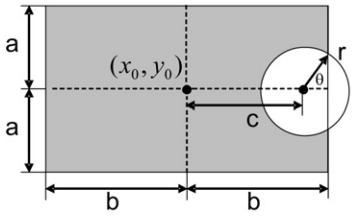

Fig. 7. A rectangle with a circular hole: when the position of the hole changes, topological changes may occur.

Example 2 *(Shape deformation with topological changes (Fig. 7)).* In this example (Fig. 7), the solid shape Ω is defined as a set difference of a rectangle and a circular disk. Recall that if the active boundary is $\mathscr{M}$-continuous and the active boundary points which belong to more than one primitives are of measure zero, the sensitivity derived in Section 3.4 is valid.

The x-position c of the hole is chosen as the design parameter. When the hole moves in/out of the rectangle, obviously the topology of the geometry changes. But regardless of the topological change, the intersection of the hole's boundary and the rectangle consists of zero, one or two points, therefore the measure-zero condition is satisfied in this case.

A direct calculation gives

$$V = 4ab - \pi r^2 + r^2 \arccos\frac{b-c}{r} - (b-c)\sqrt{r^2-(b-c)^2}$$

and

$$\frac{dV}{dc} = 2\sqrt{r^2-(b-c)^2}.$$

We now try the proposed approach. This geometry can be defined from the following three primitives:

$$\Phi_a = a^2 - (y-y_0)^2,$$
$$\Phi_b = b^2 - (x-x_0)^2,$$
$$\Phi_r = (x-x_0-c)^2 + (y-y_0)^2 - r^2.$$

The active primitive of parameter c is the circular hole and its corresponding boundary is (part) of the circle. Since

$$\frac{d\Phi_r}{dc} = -2(x-x_0-c),$$
$$|\nabla\Phi_r| = 2\sqrt{(x-x_0-c)^2+(y-y_0)^2},$$

we have

$$\begin{aligned}\frac{dV}{dc} &= \int_{\partial\Omega_r} \frac{1}{|\nabla\Phi_r|}\frac{d\Phi_r}{dc}\,d\Gamma \\ &= \int_{\partial\Omega_r} -\frac{x-x_0-c}{\sqrt{(x-x_0-c)^2+(y-y_0)^2}}\,d\Gamma \\ &= \int_{\theta}^{-\theta} -r\cos\alpha\,d\alpha = 2r\sin\theta = 2\sqrt{r^2-(b-c)^2}. \end{aligned} \tag{11}$$

At the critical points ($c = b \pm r$) where the topology changes, we have $\theta = 0$, thus the integration in Expression (11) gives zero. Fig. 8 plots the relationship between the volume function and the parameter c. We see that at the critical points, even though the topological changes do affect the behavior of the objective function, the objective function is still differentiable. With the proposed approach for sensitivity computation, we do not have to identify topological events or know the behavior of the objective function, nor do we need to maintain the boundary representation. As long as the sufficient conditions are satisfied, the sensitivity computation can be carried out.

4.2. When sensitivity analysis is invalid

Here we show examples for which the proposed approach does not apply and we explain which condition they do not satisfy.

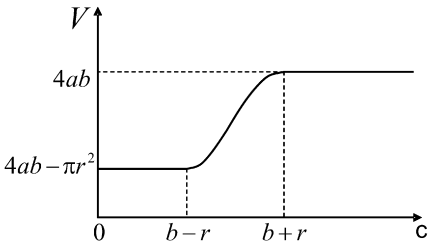

Fig. 8. The plot of the volume function with respect to variable c in Fig. 7.

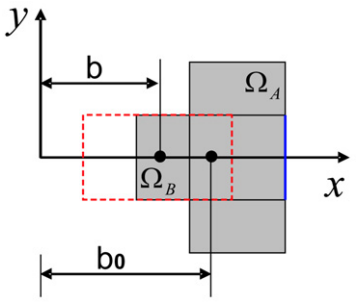

Fig. 9. The union of two rectangles: when parameter $b = b_0$, the measure-zero condition fails and the objective function is not differentiable.

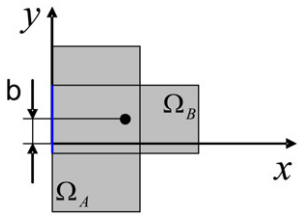

Fig. 10. The union of two rectangles: the measure-zero condition fails but the objective function is still differentiable.

Example 3 *(Measure-zero condition fails (Fig. 9)).* Fig. 9 shows a shape Ω defined as the union of two rectangles Ω_A and Ω_B. As the value of parameter b (the position of Ω_B) increases, the right edge will overlap with part of the right edge of Ω_A ($b = b_0$). At this point, though the $\mathscr{M}$-continuity condition is satisfied for the boundary, the measure-zero condition is not satisfied since every point on the right edge of Ω_B also belongs to the boundary of Ω_A. One can easily plot the volume function to see that it is not differentiable at this point.

Example 4 *(Measure-zero condition fails but F is differentiable (Fig. 10)).* The measure-zero condition is not a necessary condition for guaranteeing the differentiability of the objective function. It is easy to construct an example that does not satisfy this condition but the objective function is still differentiable, as in Fig. 10. Similar to Fig. 9, the domain Ω is the union of two rectangles Ω_A and Ω_B, but now the parameter b controls the vertical position of Ω_B. The $\mathscr{M}$-continuity condition is satisfied for the boundary and the measure-zero condition is not satisfied. It is easy to see that the volume function is constant and, therefore, differentiable as long as b is within a range that ensures the left edge of Ω_B is contained within the left edge of Ω_A.

Example 5 *($\mathscr{M}$-continuity condition fails (Fig. 11)).* In Fig. 11, the geometry Ω again is a union of two rectangle primitives. As the length b of the left rectangle increases, it eventually hits the right rectangle ($b = b_0$). Though the measure-zero condition is satisfied, the $\mathscr{M}$-continuity condition is not satisfied because the merged edges become the interior of Ω. Fig. 12 shows that the volume function is not differentiable at this transition point. In this case, the proposed approach can be used to compute a one-side derivative at b_0 (in this example, it is the right derivative $\frac{dV}{db}\big|_{b_0^+}$).

Example 6 *($\mathscr{M}$-continuity condition fails but F is differentiable (Fig. 13)).* The $\mathscr{M}$-continuity condition is not a necessary condition to guarantee the differentiability for F. In Fig. 13, we construct an example where the geometry Ω is a union of three parts: $\Omega = \Omega_A \cup (\Omega_B \cup \Omega_C)$, where Ω_B and Ω_C are rigidly attached. Parameter b controls the horizontal position of Ω_B (and Ω_C). When Ω_B moves horizontally and touches Ω_A as shown (when $b = b_0$), the measure-zero condition is satisfied but the $\mathscr{M}$-continuity condition is not satisfied. However, the volume function is differentiable at this point since

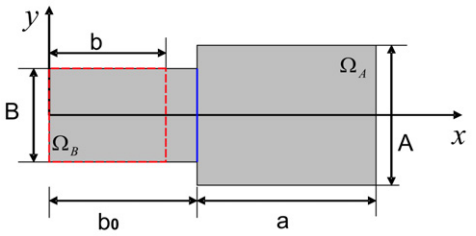

Fig. 11. The union of two rectangles: when $b = b_0$, the boundary is not $\mathscr{M}$-continuous and the objective function is not differentiable.

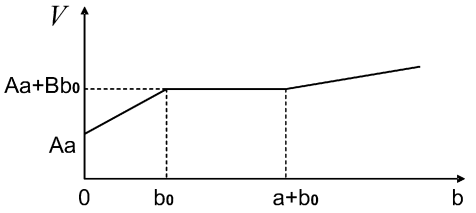

Fig. 12. The volume function corresponding to the parameter b in Fig. 11.

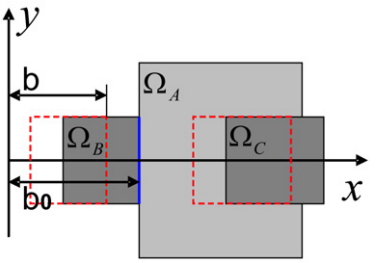

Fig. 13. The union of two rectangles: when $b = b_0$, the boundary is not $\mathscr{M}$-continuous but the objective function is still differentiable.

it is constant around $b = b_0$. This example illustrates that the $\mathscr{M}$-continuity condition is only a sufficient condition to check the differentiability of F and not a necessary condition.

5. Numerical examples

This section describes two fully implemented shape optimization examples with topological changes. In the first example, we represent the geometry in a commercial solid modeling system and optimize the geometry to minimize its volume. The shape sensitivity is computed using the proposed approach and used in a gradient-based optimization package. We will show that the topological changes of the solid can be handled naturally without any additional effort during the optimization process. In the second example, we consider minimum compliance design problems where the objective function depends on a field function—the displacement field of a linear elasticity problem. We show numerical experiment results where both parametric shapes and free-form shapes are involved. The topological changes can be handled as well.

5.1. Optimization of a hard disk arm

Fig. 14 shows the parametrization of a hard disk arm that we would like to optimize by minimizing the volume while maintaining a fixed value for the mass moment of inertia, I_z. To do this, we intersect the arm with the complement of a cylinder aligned parallel to the z-axis and optimize the x-position a and radius r of the cylinder, see Fig. 14(b). The active primitive for a and r is the cylinder and active boundaries are the part of the side face of the cylinder belonging to the arm (a and r have the same active primitive and active boundaries in this example). It is easy to check that this problem satisfies the sufficient conditions for valid sensitivity analysis (measure-zero condition and $\mathscr{M}$-continuity condition) and we can apply the proposed method to compute the sensitivities. As gradient-based optimization algorithms require the derivatives of both objective function and constraint functions, we compute the sensitivities of the objective function V and the constraint function I_z with respect to the design parameters a and r.

For the cylinder primitive, we have the following implicit representation:

$$\Phi_R = (x-a)^2 + y^2 - r^2,$$

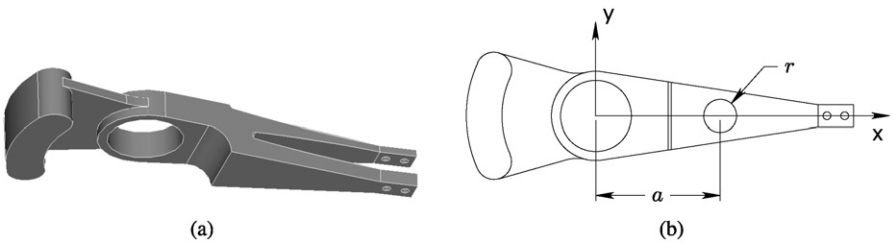

Fig. 14. A hard disk arm: (a) The 3D view of the hard disk arm; (b) The coordinate system and the hole parameters of the arm.

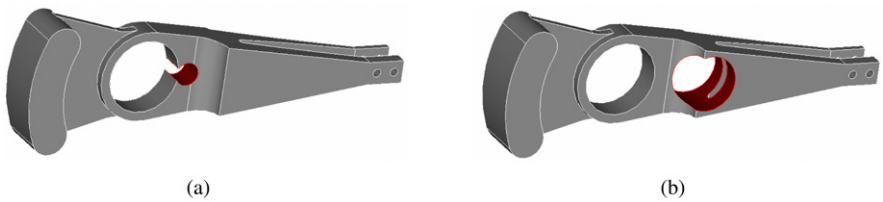

Fig. 15. Optimization of the hard disk arm: (a) The initial geometry; (b) The optimal design.

whose derivatives, with respect to the parameters a and r are:

$$\frac{d\Phi_R}{da} = -2(x-a), \qquad \frac{d\Phi_R}{dr} = -2r,$$

and its gradient is

$$|\nabla\Phi_R| = 2\sqrt{(x-a)^2+y^2}.$$

The volume and mass moment of inertia are found through the following integrals:

$$V = \iiint_{\Omega} 1\, d\Omega,$$

$$I_z = \iiint_{\Omega} (x^2+y^2)\, d\Omega,$$

which are computed directly by the geometric engine.

The sensitivities of V and I_z with respect to a and r are:

$$\frac{dV}{da} = -\iint_{\partial\Omega_R} \frac{x-a}{\sqrt{(x-a)^2+y^2}}\, dS,$$

$$\frac{dV}{dr} = -\iint_{\partial\Omega_R} \frac{r}{\sqrt{(x-a)^2+y^2}}\, dS,$$

$$\frac{dI_z}{da} = -\iint_{\partial\Omega_R} \frac{x-a}{\sqrt{(x-a)^2+y^2}}(x^2+y^2)\, dS,$$

$$\frac{dI_z}{dr} = -\iint_{\partial\Omega_R} \frac{r}{\sqrt{(x-a)^2+y^2}}(x^2+y^2)\, dS,$$

$\partial\Omega_R$ is the active boundary for a and r, i.e. the boundary point set belonging to the side face of the cylinder, which is shown in red in Fig. 15.

For experimental purposes, primitives are represented using boundary representations in the Parasolid modeling kernel (UGS PLM SOLUTIONS: Parasolid). The optimization program operates on two primitives: the arm primitive, Fig. 14(a); and our active primitive: the parameterized cylinder. Intersecting the complement of the cylinder with the arm creates one or more new faces corresponding to the active boundaries of the cylinder primitive. We integrate over these Parasolid faces using a standard quadrature rule. Though the implementation described here computes the resulting boundary representation, in principle, boundary evaluation is not necessary. The integration may be carried out over portions of the active boundaries that classify *on* with respect to a constructive representation of the arm, provided such membership classification is supported (Sarraga, 1982).

The optimization is carried out within Matlab using an interface to communicate with the Parasolid engine. Besides the moment of inertia constraint, we impose additional linear constraints on a and r such that the hole is fully contained in the arm to prevent disconnected designs. Matlab's non-linear constrained optimization function `fmincon()` uses the sensitivity information from the above integrals to converge to a final solution shown in Fig. 15 within seconds in 7 iterations. At this position, the optimum amount of material is removed, while maintaining a constant mass moment of inertia about the z-axis. We can see that the optimal geometry has a very different topology with the initial geometry.

5.2. Minimum compliance design problems

The approach we proposed in Section 3 for shape sensitivity computation can be applied to more complicated objective functions that may depend on field functions (stress, temperature, energy, etc.). We consider the following minimum compliance problem where the objective is to minimize the total strain energy:

$$\begin{aligned} &\min_{\Omega} \ J_0(u) = \iint_{\Omega} \frac{1}{2} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u)\,d\Omega \\ &\text{s.t.} \quad a(u,v) = l(v), \quad \forall v \in U, \\ &\qquad u|_{\Gamma_1} = u_0 \end{aligned}$$

where $a(u,v) = l(v)$ is the equilibrium equation, $a(u,v) = \iint_{\Omega} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(v)\,d\Omega$, and $l(v) = \iint_{\Omega}[fv + \operatorname{div}(pvn)]\,d\Omega$. The boundary $\Gamma = \Gamma_1 \cup \Gamma_2$, Dirichlet boundary condition $u = u_0$ is specified on Γ_1 and boundary traction p is specified on Γ_2, f is the body force. u is the displacement field, E is Young's modulus and ϵ is the elastic strain, v is the virtual displacement and U is the space of all admissible displacements.

If we represent the geometry Ω by the implicit surface $\Phi(x)$ and use Expression (5), the above problem can be formulated as

$$\begin{aligned} &\min_{\Phi} \ J_0(u,\Phi) = \iint_{D} \frac{1}{2} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u)H(\Phi)\,d\Omega \\ &\text{s.t.} \quad a(u,v,\Phi) = l(v,\Phi), \quad \forall v \in U, \\ &\qquad u|_{\Gamma_1} = u_0 \end{aligned}$$

where $a(u,v,\Phi) = \iint_D E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(v)H(\Phi)\,d\Omega$, $l(v,\Phi) = \iint_D [fv + \operatorname{div}(pvn)]H(\Phi)\,d\Omega$.

The derivative of the objective function $J_0(u,\Phi)$ with respect to a parameter b defining Φ can be expressed as a boundary integral (Chen et al., 2007):

$$\frac{dJ_0(u,\Phi)}{db} = \int_{\partial\Omega} \left[fu + \operatorname{div}(pun) - \frac{1}{2} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u) \right] \frac{d\Phi}{db}\,d\Gamma. \tag{12}$$

5.2.1. Example 1: Parametric shape design

Fig. 16 shows a short cantilever beam design problem. The design parameters are the center (x_c, y_c) of the circular hole and the center (x_r, y_r) of the rectangular slot. The implicit representations for the circular hole and the rectangular slot can be written as

$$\Phi_c = \frac{1}{2R_c}\left[(x - x_c)^2 + (y - y_c)^2 - R_c^2\right]$$

and

$$\Phi_r = \frac{(x - x_r)^2 - A^2}{2A} \wedge_0 \frac{(y - y_r)^2 - B^2}{2B},$$

where $\wedge_0$ is R-conjunction operation (Rvachev, 1967, 1982; Shapiro, 1988; Shapiro and Tsukanov, 1999) corresponding to logical intersection.

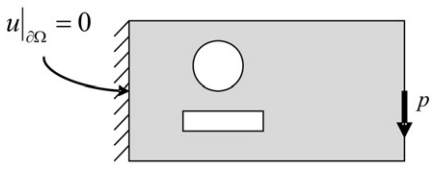

Fig. 16. A minimum compliance optimization problem.

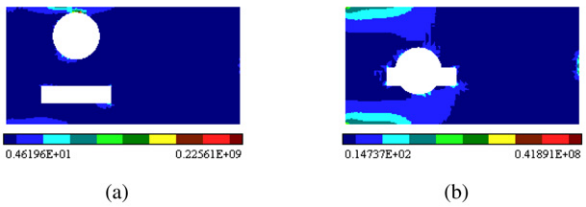

Fig. 17. Strain energy distribution of the initial and optimal shapes of a rectangle with a circular and rectangular slot. The design variables are the positions of the hole and the slot. (a) The initial design. (b) The optimal shape.

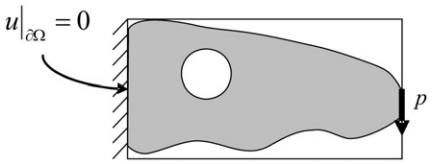

Fig. 18. A minimum compliance optimization problem.

Using Expression (8), the shape sensitivity of the compliance can be written as an integral over active boundaries, which in this case are the partial circle and partial rectangle:

$$\frac{dJ_0(u,\Phi)}{dx_c} = -\int_{\Gamma_c} \frac{x-x_c}{R_c} \cdot \left[fu + \operatorname{div}(pun) - \frac{1}{2} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u) \right] d\Gamma,$$

$$\frac{dJ_0(u,\Phi)}{dy_c} = -\int_{\Gamma_c} \frac{y-y_c}{R_c} \cdot \left[fu + \operatorname{div}(pun) - \frac{1}{2} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u) \right] d\Gamma,$$

$$\frac{dJ_0(u,\Phi)}{dx_r} = -\int_{\Gamma_r^1} \frac{x-x_r}{A} \cdot \left[fu + \operatorname{div}(pun) - \frac{1}{2} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u) \right] d\Gamma,$$

$$\frac{dJ_0(u,\Phi)}{dy_r} = -\int_{\Gamma_r^2} \frac{y-y_r}{B} \cdot \left[fu + \operatorname{div}(pun) - \frac{1}{2} E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u) \right] d\Gamma,$$

where $\Gamma_c = \{(x,y) \in \partial\Omega \mid (x-x_c)^2 + (y-y_c)^2 - R_c^2 = 0\}$, $\Gamma_r^1 = \{(x,y) \in \partial\Omega \mid (x-x_r)^2 = A^2\}$ and $\Gamma_r^2 = \{(x,y) \in \partial\Omega \mid (y-y_r)^2 = B^2\}$.

After obtaining the sensitivity information, we may use any gradient-based optimization algorithm to solve the problem. In our implementation, we regard the shape design process as a time-varying process and construct a series of descent directions based on the sensitivities to decrease the objective function. A mesh-free approach with distance fields is used to solve the structural analysis problem, whose solution is used to compute the shape sensitivity. The main optimization algorithm and numerical implementation details can be found in Chen et al. (2007). Fig. 17 shows the initial design and the optimal shape. The topological changes, such as intersection of hole and the slot, are handled easily without any additional effort.

5.2.2. Example 2: Free-form shape design with parametric features

Fig. 18 shows another short cantilever beam design problem. The difference with the previous example is in the parametrization of the geometry. In the previous example, the design variables are the geometric dimensions. In this example we have both parametric shape (the circular hole) and free-form shape. The circular hole again may be represented implicitly as

$$\Phi_c = \frac{1}{2R_c}\left[(x_1 - x_c)^2 + (x_2 - y_c)^2 - R_c^2\right].$$

We represent the free-form shape using an implicit surface approximated by a linear combination of B-spline basis functions:

$$\Phi_f = \sum c_i \chi_i(x),$$

where $\{c_i\}$ are the coefficients and $\{\chi_i(x)\}$ are linear B-spline basis functions defined on a uniform rectangular grid. Therefore, we have a free-form and parametric shape design problem with (x_c, y_c) and $\{c_i\}$ as the design variables. The shape sensitivities with respect to (x_c, y_c) are:

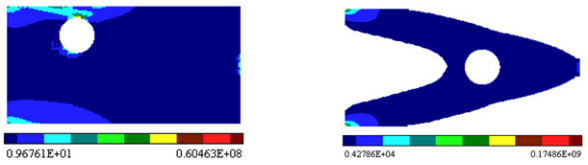

Fig. 19. Strain energy distribution of the initial and optimal shapes. The design variables are the center of the circular hole and the B-spline coefficients. (a) The initial design. (b) The optimal shape.

$$\frac{dJ_0(u,\Phi)}{dx_c} = -\int_{\Gamma_c} \frac{x-x_c}{R_c}\cdot\left[fu + \operatorname{div}(pun) - \frac{1}{2}E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u)\right]d\Gamma,$$

$$\frac{dJ_0(u,\Phi)}{dy_c} = -\int_{\Gamma_c} \frac{y-y_c}{R_c}\cdot\left[fu + \operatorname{div}(pun) - \frac{1}{2}E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u)\right]d\Gamma.$$

The shape sensitivity with respect to each c_i can be shown to be a boundary integral over its active boundaries: the boundary of the shape contained within the support of the corresponding B-spline basis function $\chi_i(x)$ (Chen et al., 2007):

$$\frac{dJ_0(u,\Phi)}{dc_i} = \int_{\partial\Omega_i} \chi_i(x)\cdot\left[fu + \operatorname{div}(pun) - \frac{1}{2}E_{ijkl}\epsilon_{ij}(u)\epsilon_{kl}(u)\right]d\Gamma,$$

where $\partial\Omega_i = \partial\Omega \cap \operatorname{support}(\chi_i(x))$.

The free-form surface representation for the geometry is similar to the concept used in the level set method developed over the past several years (Sethian and Wiegmann, 2000; Allaire et al., 2004; Wang et al., 2003). However, the level-set method relies on the boundary variation for shape sensitivity analysis, and therefore does not allow topological changes. When the topology changes, the shape sensitivity is undefined and in practice one often chooses to ignore this fact and accept shape changes. The proposed shape sensitivity analysis gives a formal justification for this action in terms of the rigorous conditions listed in Section 3.5.

Fig. 19 shows the initial and optimal shapes. Since minimum compliance design problems tend to use as much material as possible to decrease the objective function, we add a volume constraint $\iint_D H(\Phi)\,d\Omega = V_0$ to the optimization problem (V_0 is one half of the area of the rectangle). This example confirms our earlier statement that the sensitivity formulation in our method is very general and can be applied to any geometries represented implicitly by some function Φ.

6. Conclusions

We proposed a new approach for shape sensitivity computation for a general class of functions defined as domain integrals on constructive representations of solids. Using existence of a sufficiently smooth implicit representation as a theoretical tool, we showed that the shape sensitivity can be formulated as a sum of a domain integral and a boundary integral. We further showed that the boundary integration can be computed efficiently over the boundaries of the active primitives used within the constructive representation of a solid. In contrast to other methods, our method does not place artificial topological constraints on the boundary representation of the solid and in many cases does not require full boundary evaluation. The advantages of the approach are illustrated by the simple examples in Section 4 and are demonstrated by several numerical experiments in shape optimization for different objective functions. Our implementation is sufficient for demonstration purposes, but is relatively unsophisticated. A more general and efficient code would rely on automatic differentiation of arbitrary primitives (Shapiro and Tsukanov, 1999; Tsukanov and Hall, 2003) and would avoid full boundary evaluation whenever possible.

The proposed shape sensitivity formulation is both simpler and more powerful than earlier approaches. Yet, it is fully consistent with the classical SSA theory in the restricted situations implemented by others. This also indicates wide applicability of the proposed approach to constructive representations with most types of primitives. We only relied on *existence* of implicit representations, but in fact it does not matter whether the primitives are represented implicitly, parametrically, variationally, or procedurally. The proposed approach applies as long as we are able to compute the shape (design) velocity v_n in the direction normal to the primitive's boundary. For instance, when the primitive is a free-form (B-spline, Bezier, etc.) surface patch that is trimmed by the constructive representation, the normal velocity may be easily constructed as in Choi and Kim (2005b) under the assumption that the topology of the patch remains invariant. Another important class of primitives can be broadly classified as extrusions of variationally constrained sketches (Hoffmann and Juan, 1993). The design velocities for such primitives are intimately connected to constraint solving techniques (for example, see Hoffmann and Joan-Arinyo, 2005; Brüderlin and Roller, 1998 and references therein). If one does not have access to the relevant source code, the primitive design velocity may be estimated using finite difference techniques (Chen and Tortorelli, 1997). Once the primitive velocities are computed based on the properties and representations of the individual primitives, they can all be used simultaneously within the framework described in this paper.

Acknowledgements

This work was supported in part by the National Science Foundation grants CMMI-0323514, CMMI-0500380, CMMI-0621116, OCI-0537370, and Wisconsin Industrial & Economic Development Research Program (I&EDR).

References

Allaire, G., Jouve, F., Toader, A.M., 2004. Structural optimization using sensitivity analysis and a level-set method. Journal of Computational Physics 194, 363–393.

Belegundu, A.D., Rajan, S.D., 1988. A shape optimization approach based on natural design variables and shape functions. Computer Methods in Applied Mechanics 66 (1), 87–106.

Bendsøe, M.P., Sigmund, O., 2003. Topology Optimization: Theory, Methods and Applications. Springer-Verlag, Berlin, Heidelberg.

Botkin, M.E., 1992. Three-dimensional shape optimization using fully automatic mesh generation. AIAA Journal 30 (7), 1932–1934.

Brüderlin, B., Roller, D., 1998. Geometric Constraint Solving and Applications. Springer.

Chang, K.H., Choi, K.K., 1992. A geometry-based parameterization method for shape design of elastic solids. Mechanics of Structures and Machines 20 (22), 215–252.

Chen, J., Freytag, M., Shapiro, V., 2007. Shape sensitivity of constructive representations. In: Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling. ACM Press, pp. 85–95.

Chen, J., Shapiro, V., Suresh, K., Tsukanov, I., 2007. Shape optimization with topological changes and parametric control. International Journal of Numerical Methods in Engineering 71 (3), 313–346.

Chen, S., Tortorelli, D.A., 1997. Three-dimensional shape optimization with variational geometry. Structural Optimization 13, 81–94.

Choi, K.K., Kim, N.H., 2005a. Structural Sensitivity Analysis and Optimization I: Linear Systems. Springer.

Choi, K.K., Kim, N.H., 2005b. Structural Sensitivity Analysis and Optimization II: Nonlinear Systems and Applications. Springer.

Freytag, M., Shapiro, V., 2004. B-rep SE: simplicially enhanced boundary representation. In: Elber, G., Patrikalakis, N., Brunet, P. (Eds.), ACM Symposium on Solid Modeling and Applications, Genova, Italy, 2004, pp. 157–168.

Griewank, A., Corliss, G.F., 1991. Automatic Differentiation of Algorithms: Theory, Implementation, and Application. Society for Industrial and Applied Mathematics.

Haug, E.J., Choi, K.K., Komkov, V., 1986. Design Sensitivity Analysis of Structural Systems. Academic Press, New York.

Hirota, G., Maheshwari, R., Lin, M.C., 1999. Fast volume-preserving free form deformation using multi-level optimization. In: Proceedings of the Fifth ACM Symposium on Solid and Physical Modeling. ACM Press, pp. 234–245.

Hoffmann, C.M., Juan, R., 1993. Erep, an editable, high-level representation for geometric design and analysis. In: Wilson, P., Wozny, M., Pratt, M. (Eds.), Geometric and Product Modeling. North-Holland.

Hoffmann, C.M., Joan-Arinyo, R., 2005. A brief on constraint solving. Computer-Aided Design and Applications 2 (5), 655–663.

Keane, A., Nair, P., 2005. Computational Approaches for Aerospace Design: The Pursuit of Excellence. John-Wiley and Sons.

Kodiyalam, S., Kumar, V., Finnigan, P., 1992. Constructive solid geometry approach to three-dimensional structural shape optimization. AIAA Journal 30 (5), 1408–1415.

Lee, Y.T., Requicha, A.A.G., 1982a. Algorithms for computing the volume and other integral properties of solids. i. Known methods and open issues. Communications on ACM 25 (9), 635–641.

Lee, Y.T., Requicha, A.A.G., 1982b. Algorithms for computing the volume and other integral properties of solids. ii. A family of algorithms based on representation conversion and cellular approximation. Communications on ACM 25 (9), 642–650.

Olhoff, N., Lund, E., Rasmussen, J., 1992. Concurrent engineering design optimization in a cad environment. Special report 16, Institute of Mechanical Engineering, Aalborg University.

Osher, S., Sethian, J.A., 1988. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. Journal of Computational Physics 79, 12–49.

Raghothama, S., Shapiro, V., 1997. Necessary conditions for boundary representation variance. In: Proceedings of the Thirteenth ACM Symposium on Computational Geometry, June 4–6.

Rappoport, A., Sheffer, A., Bercovier, M., 1995. Volume-preserving free-form solid. In: Proceedings of the Third ACM Symposium on Solid and Physical Modeling. ACM Press, pp. 361–372.

Rvachev, V.L., 1967. Geometric Applications of Logic Algebra. Naukova Dumka (in Russian).

Rvachev, V.L., 1982. Theory of *R*-functions and Some Applications. Naukova Dumka (in Russian).

Sarraga, R.F., 1982. Computation of surface areas in gmsolid. IEEE Computer Graphics and Applications 2 (7), 65–70.

Sethian, J.A., Wiegmann, A., 2000. Structural boundary design via level set and immersed interface methods. Journal of Computational Physics 163 (2), 489–528.

Shapiro, V., 1988. Theory of R-functions and applications: A primer. Technical Report, Cornell University.

Shapiro, V., Tsukanov, I., 1999. Implicit functions with guaranteed differential properties. In: Proceedings of Fifth ACM Symposium on Solid Modeling and Applications, pp. 258–269.

Sokolowski, J., Zolesio, J.P., 1992. Introduction to Shape Optimization: Shape Sensitivity Analysis. Springer-Verlag, New York.

Tortorelli, D.A., 1993. A geometric representation scheme suitable for shape optimization. Mechanics of Structures and Machines 21, 95–121.

Tsukanov, I., Hall, M., 2003. Data structure and algorithms for fast automatic differentiation. International Journal for Numerical Methods in Engineering 56 (13), 1949–1972.

UGS PLM SOLUTIONS: Parasolid. Software.

Wang, M.Y., Wang, X.M., Guo, D.M., 2003. A level set method for structural topology optimization. Computer Methods in Applied Mechanics and Engineering 192 (1–2), 227–246.

Yang, R.J., Dewhirst, D.L., Allison, J.E., Lee, A., 1992. Shape optimization of connecting rod pin and using a generic model. Finite Elements in Analysis and Design 11, 257–264.