Proceedings of IDETC/CIE 2006
ASME 2006 International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
September 10-13, 2006, Philadelphia, Pennsylvania, USA

DETC2006-99612

PARAMETRIC AND TOPOLOGICAL CONTROL IN SHAPE OPTIMIZATION

Jiaqin Chen, Vadim Shapiro*, Krishnan Suresh, Igor Tsukanov
Spatial Automation Laboratory
University of Wisconsin-Madison
1513 University Avenue, University of Wisconsin, Madison, WI 53706, USA
Email: jiaqinchen@wisc.edu, vshapiro@engr.wisc.edu, suresh@engr.wisc.edu, tsukanov@engr.wisc.edu

ABSTRACT

We propose a novel approach to shape optimization that combines and retains the advantages of the earlier optimization techniques. The shapes in the design space are represented implicitly as level sets of a higher-dimensional function that is constructed using B-splines (to allow free-form deformations), and parameterized primitives combined with R-functions (to support desired parametric changes).

Our approach to shape design and optimization offers great flexibility because it provides explicit parametric control of geometry and topology within a large space of freeform shapes. The resulting method is also general in that it subsumes most other types of shape optimization as special cases. We describe an implementation of the proposed technique with attractive numerical properties. The effectiveness of the method is demonstrated by several numerical examples.

KEYWORDS: shape optimization, topology optimization, parametric design, level-set, implicit representation, *R*-functions, shape sensitivity analysis

1 Introduction

1.1 Shape optimization: parametric vs free-form

A *parametric* shape is defined by a finite, and usually small, set of geometric parameters called dimensions. Common examples of dimensions include sizes, radii, distances, angles, and other geometrically meaningful design and/or manufacturing variables. Most modern CAD systems represent shapes parametrically. Parametric shape optimization searches the space spanned by the design variables to minimize or maximize some externally defined objective function. In other words, parametric shape optimization is essentially a *sizing* problem that is a natural extension of parametric computer-aided design.

The downside of parametric shapes is that they do not provide any explicit information about the geometry or topology of the shape's boundaries. This, in turn, leads to at least two widely acknowledged difficulties: boundary evaluation may fail [48,25], and topological changes in the boundaries may invalidate boundary conditions or the solution procedure [9]. A common approach to dealing with these difficulties is to restrict the design space to shapes with identical parameterization and topology, as illustrated in Figure 1.

(a) A simple shape is parameterized by three dimensions that procedurally define the shape's boundary.

(b) Changes in parameter values that result in a different topology are usually not allowed.

Figure 1. Traditional parametric shape optimization and limitations.

In contrast to parametric shapes, a *free-form* shape is defined by its boundaries without any prior explicit dimensional parameterization. Free-form shape optimization searches the space of free-form shapes by incremental local motion of the free-form

*Address all correspondence to this author.

boundaries. This seemingly precludes topological changes in the boundary. The difficulty is resolved by representing the shape's boundary *implicitly* in terms of level sets (isocurves or isosurfaces) of some higher dimensional time-varying hyper-surface $\Phi(x,t)$ [37]. Topological changes in level sets are captured in terms of smooth motions of the hyper-surface, while retaining the free-form nature of the shape's boundaries. In particular, direct movement of a level set surface has been used to represent both deformations of boundaries and to create holes during the optimization process [54,3]. Figure 2 shows a typical shape deformations allowed in a free-form shape optimization (with topological changes). Free-form shapes are usually parameterized *locally* by

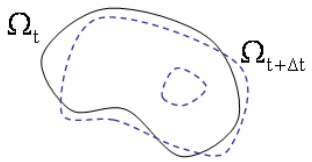


Figure 2. Free-form deformation with topological changes can be represented by a moving level set hyper-surface.

polygonal approximations or in terms of some compactly supported basis functions (radial basis functions, B-splines, etc.) It should be intuitively apparent that the space of free-form shapes is usually much larger than a typical space of parametric shapes. However, without additional constraints, such free-form representations cannot be controlled in terms of global dimensional parameters that are so critical in many design and manufacturing applications. Consequentially, free-form optimization techniques often produce optimal solutions that may not be manufacturable. Because of this limitation, free-form shape optimization (with topological changes) is particularly useful at the conceptual design stage.[1] The output of free-form optimization is then transformed into detailed shape design, typically relying on heuristic shape processing techniques that may not be consistent with the original formulation of the optimization problem.

In this paper, we combine parametric and free-form shape representations in a common unified representational framework. The new representation retains all the advantages of free-form and parametric shapes, and supports parametric, free-form, or hybrid types of shape optimization, with or without topological changes. The resulting shapes may have boundaries that are partially free-form (thus allowing sufficiently large design space) and partially parametric (thus providing control of the shape in terms of meaningful geometric parameters). The key unifying concept is that of an implicit representation $\Phi(x,t)$ that is parameterized locally in terms of B-spline functions *and/or* globally in terms of dimensional parameters. We show that such a representation may be constructed using *R*-functions [30, 39] that transform any set-theoretic construction into a sufficiently smooth hyper-surface via simple syntactic substitution.

We discuss the implementation of the proposed approach in a meshfree environment developed by the authors [46, 49] and demonstrate its performance by applying it to a widely studied minimum compliance structural shape optimization problem.

1.2 Related work

Parametric shape optimization is a well researched area that is discussed in many references [16, 9, 22] and is subject to the limitations discussed above.

Free-form shape optimization has also been studied extensively, with numerous advances during the last decade focusing on handling topological changes. To allow topological changes of the shape, early methods explicitly move the shape's boundary and introduce holes in the domain to accommodate topological changes. For example, evolutionary structural optimization (ESO) method [56], Bubble method [11], topological sensitivity analysis based method [15]. Recently proposed level set methods in shape optimization have attracted much attention, due to their ability to track evolving boundaries and handle topological changes [37]. In [38], the shape is represented as a level-set of a higher dimensional surface and a structural optimization problem is formulated and solved on this higher-dimensional surface. Holes can be created during the optimization process by modifying the surface, based on criteria that are similar to those used by other evolutionary methods. Shape sensitivity analysis in [2] and [52] show that the level-set speed function may be chosen to guarantee a descent direction of the objective function, which gives mathematical guidance on how to move the boundary. However, it has also been observed that motions of the level-set surface based on shape sensitivity alone do not appear to nucleate holes inside the domain [2]. In fact, this is not surprising, because the shape sensitivity analysis is based on small perturbations of the boundary, and therefore does not provide a mechanism for sudden topological changes such as nucleation of holes. In order to overcome the topological limitations in practice, many small holes are often inserted into the initial design and are allowed to merge under boundary motion, but the result of optimization appears to depend on the initial distribution of holes [2, 38, 52]. Another approach to handling topological changes with level sets is described in [54], where the authors use radial basis functions to represent the level set surface and extend the boundary velocity of the shape to the entire domain. The authors in [3] represent level-set function by finite element shape functions and rely on a heuristic criterion to perform shape

[1] This observation also applies to all topology optimization methods, such as homogenization, which uses material models with micro-structures and seeks an optimal layout over the design domain. Since we are mainly focusing on shape optimization at the macro (geometry) level, we refrain from discussing the topology optimization methods based on material distributions.

and topology optimization. The concept of topological derivative has been proposed to overcome the difficulty of generating interior holes [7, 1, 53] and appears to be promising, but is not fully developed at this time.

The above efforts (along with many others) have made significant contributions to the area of structural shape/topology optimization. But until now, free-form shape optimization and parametric optimization have been treated as separate and mutually exclusive techniques in shape design. Our approach builds on earlier approaches using level sets, and shares some similarities with [54] and [3], but also incorporates full power and advantages of parametric shape optimization.

1.3 Outline

The rest of the paper is organized as follows. In Section 2, we introduce the proposed shape representation method and explain its advantages. In Section 3, a minimum strain energy shape optimization problem is formulated using the proposed representation technique. Section 4 develops the optimization algorithm and shape sensitivity analysis for the formulated problem. In Section 5, we illustrate the generality and flexibility of the proposed method to shape control during the optimization process; numerical examples are given to demonstrate the correctness and effectiveness of the proposed method. Section 6 discusses the numerical implementation issues, followed by conclusions in Section 7.

2 Shape Representation

We propose to represent both free-form shapes and parametric shapes implicitly using level sets of higher-dimensional functions (hyper-surfaces) and use the theory of *R*-functions to represent arbitrary set combinations of such shapes. The key observation is that the space of level set functions that are differentiable almost everywhere is closed under *R*-functions [45].

2.1 Implicit Representations of Shapes

Implicit representations of shapes have a long tradition in geometric modeling and computer graphics, as described in several recent books [6, 51]. All such representations define a shape $\Omega \subseteq D$ implicitly in terms of non-negative values of some function $\Phi(x)$ of the spatial variable x as $\Omega = \{x \in D \,|\, \Phi(x) \geq 0\}$, where D is some predefined reference domain that contains all possible shapes Ω of interest. The boundary $\partial\Omega$ of the shape Ω is the zero level set of the function $\partial\Omega = \{x \in D \,|\, \Phi(x) = 0\}$.

This definition is consistent with the notion of level set function in [37,38,2,54,52,3], but also includes many other representations used in geometric modeling. Many techniques and transformations for constructing such representations are described in [6], including Ricci's function [28], theory of *R*-functions [30, 31, 39, 40], and convolution methods. More recent notable methods include exact and approximate distance fields [14, 5], blending of implicit primitives like blobs, spheres, quadrics, and local quadratics that have been fit to the points [21, 20, 24], radial basis functions with both global [50] and compact support [35, 19], and multi-variate B-splines to represent scalar fields whose zero-sets represent the boundary of sculpted geometry [26, 36]. Implicit representations may be constructed from both Constructive Solid Geometry and Boundary Representations of geometric objects [40, 42, 43]. Although implicit representations lack explicit boundary information [44], we will show in Section 6 that our implementation does not require it.

2.2 Parametric and Free-form Primitive Shapes

A shape optimization process is an iterative procedure, where the shape Ω and its implicit representation can be considered as time-dependent functions $\Omega(t)$ and $\Phi(x,t)$.

A parametric level-set of a function $\Phi(x,t)$ is parameterized in terms of geometrically meaningful variables $\{b_i\}$. Familiar examples of implicitly defined parametric shapes include conic sections and quadric surfaces, super-ellipses and super-quadrics, tori, as well as local and global transformations of these simple shapes [6]. The corresponding functions Φ for these primitive shapes are well known. The geometric parameters (radii, focal distances, angles, positions, etc.) of these implicit representations serve as time-dependent design variables that evolve during the search for optimal shape. We will use $\Phi_p(x, \mathbf{b}(t))$ to denote the level-set functions for parametric shapes, where x is the spatial variable and $\mathbf{b}(t) = \{b_1(t), b_2(t), \ldots, b_M(t)\}$ is the set of geometric parameters. Parametric implicit representations for more complex shapes can be built from primitive shapes using a variety of blending, convolution, and set-theoretic techniques [6, 46]. If the implicit function $\Phi_p(x, \mathbf{b}(t))$ is constructed from two primitive implicit representations $\Phi_p^1(x, \mathbf{b}^1(t))$ and $\Phi_p^2(x, \mathbf{b}^2(t))$, then the vector of parameters $\mathbf{b}$ is simply an (ordered) union of $\mathbf{b}^1$ and $\mathbf{b}^2$.

A free-form implicit representation relies on a function $\Phi(x,t)$ that is constructed as a linear combination of basis functions $\{\chi_i(x), i = 1, \ldots, N\}$ from some complete space:

$$\Phi(x,t) = \sum_{i=1}^{N} c_i(t)\chi_i(x) \tag{1}$$

The associated (free-form) shape optimization problem is to determine the unknown coefficients $\{c_i(t)\}$ for an optimal shape. The term "free-form" is consistent with the fact that the parameters $\{c_i\}$ do not have intuitive geometric meaning. Popular choices of the basis functions $\{\chi_i(x)\}$ include polynomials, trigonometric, B-splines, radial basis functions, etc. For our implementation we chose multivariate B-splines on a uniform grid subdividing the reference domain D because of their well-

understood smoothness and local control properties [10]. The local control is particularly useful for performing local shape deformations and for forcing or disallowing some topological changes by manipulating a particular subset of the coefficients. A free-form implicit representation (1) parameterizes the shape in terms of the coefficients $\{c_i\}$ of basis functions $\{\chi_i(x)\}$. This parametrization effectively transforms the difficult free-form shape optimization problem into an easier problem of "sizing" coefficients $\{c_i\}$. To distinguish free-form representations from parametric ones, in the remainder of the paper, we will use $\Phi_f(x,\mathbf{c}(t))$ to denote the implicit functions for free-form shapes, where x is the spatial variable and $\mathbf{c}(t) = \{c_1(t), c_2(t), \ldots, c_N(t)\}$ is the set of B-spline coefficients.

2.3 Composition of level-set functions with *R*-functions

Complex geometric shapes can be constructed using Boolean set operations $\cap$ and $\cup$. For example, the geometric domain in Figure 3(a) is described by the Boolean expression $\Omega = (\Omega_1 \cap \Omega_2) \cap \Omega_3$, where Ω_i, $i = 1,2,3$ are primitives shapes represented implicitly by the corresponding level-set functions Φ^i:

$$\Phi^1(x,y) = b^2 - y^2 \geq 0;$$
$$\Phi^2(x,y) = a^2 - x^2 \geq 0;$$
$$\Phi^3(x,y) = (x-x_c)^2 - (y-y_c)^2 - r^2 \geq 0$$

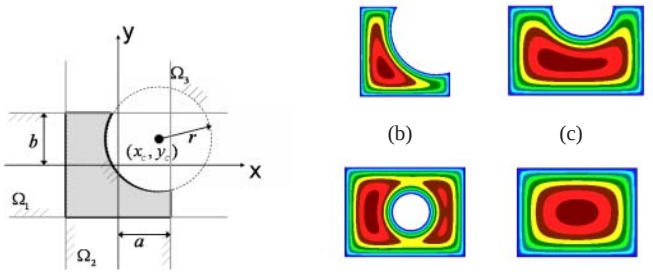
More generally, most shapes in geometric modeling belong to the class of semi-analytic sets that, by definition, can be constructed using logical operations on equalities and inequalities with analytic functions [27, 44]. Constructive Solid Geometry representations rely on such logic expressions explicitly, but they can also be constructed automatically from a variety of other representations [47, 41, 42].

While such Boolean expressions are perfectly adequate for most geometric computations, they cannot be differentiated. Recall that differential properties of the level-set function Φ are essential for shape sensitivity analysis. Fortunately, the theory of *R*-functions allows to translate any logical composition of level-set functions into a single sufficiently smooth function via straightforward syntactic substitution. *R*-functions are real-valued functions whose signs are completely determined by the signs of their arguments [29, 39, 40]. They were discovered by Rvachev who developed the theory specifically for solution of boundary value problems in mechanics [31, 32]. For example, multiplication xy of two arguments x and y is an *R*-function because it is positive only when x and y are both positive or negative. Considering the sign of a function as its "logical" attribute, the relationship between logical expressions and *R*-functions becomes apparent; for example, multiplication corresponds to the logical equivalence operation. In fact, every Boolean function corresponds to a space of *R*-functions, but they can be studied and classified in terms of their logical and differential properties. Properties of the most popular *R*-functions have been studied extensively in [30, 31, 39, 45]. A popular system of these functions includes:

$$f_1 \wedge_0 f_2 \equiv f_1 + f_2 - \sqrt{f_1^2 + f_2^2};$$
$$f_1 \vee_0 f_2 \equiv f_1 + f_2 + \sqrt{f_1^2 + f_2^2} \qquad (2)$$

It is easy to check that $f_1 \wedge_0 f_2$ is positive if and only if both f_1 and f_2 are positive; likewise, $f_1 \vee_0 f_2$ is positive if and only if f_1 or f_2 are positive. In addition, these functions are analytic everywhere except where $f_1 = f_2 = 0$. Using *R*-functions, any set theoretic expression can be translated into a real-valued function by *syntactically* replacing Boolean operations by the corresponding *R*-functions. This is one of the major outcomes of the theory of *R*-functions.

The composition of primitive implicit representations by *R*-functions is another implicit representation that is parameterized by the union of parameters in the primitives. Furthermore, topological changes in the level sets of the composite function correspond precisely to the changes in the respective parameter values. For example, Figure 3 (b-d) shows the isolines of the composite function (positive part) corresponding to the set-theoretic construction $\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3$ in Figure 3(a) with different values for the geometric parameters $\{a, b, r, x_c, y_c\}$. Significant shape changes are obtained without any additional effort to track the boundary movement or topological changes. All geometric and topological information is implied by the geometric parameters.



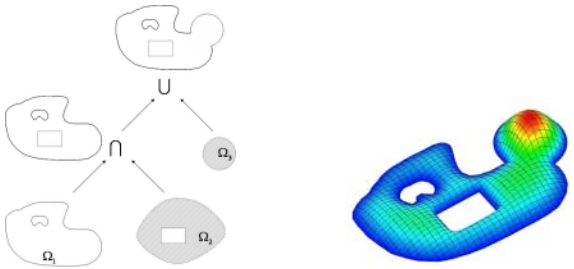
(a) Parametric shape defined as $\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3$.

Figure 3. Parametric shape deformations corresponding to changes in values of geometric parameters

The above approach with *R*-functions supports composition of arbitrary parametric and free-form level-set functions. Suppose we already have implicit representations for K_f free-form shapes $\left\{\Phi_f^1, \ldots, \Phi_f^{K_f}\right\}$, and a collection of K_p parametric shapes $\left\{\Phi_p^1, \ldots, \Phi_p^{K_p}\right\}$. Each primitive free-form function is defined by a linear combination of some basis functions $\Phi_f^k =$

$\Phi_f^k(x,\mathbf{c}^k(t)) = \sum_{i=1}^{N_k} c_i^k(t)\chi_i^k(x)$ for $k = 1,\ldots,K_f$, and each parametric function is parameterized in terms of geometric parameters as $\Phi_p^k = \Phi_p^k(x,\mathbf{b}^k(t))$ for $k = 1,\ldots,K_p$. If a composite shape is defined using some Boolean function of all free-form and parametric primitives, then the composite level-set function is immediately obtained as $\Phi(x,t) = \Phi\left(\Phi_f^1,\ldots,\Phi_f^{K_f},\Phi_p^1,\ldots,\Phi_p^{K_p}\right)$, where Φ is an *R*-function corresponding to the Boolean function.

Figure 4(b) shows a plot of the constructed level-set function (positive part) for the geometric domain $\Omega = \Omega_1 \cap \Omega_2 \cup \Omega_3$ in Figure 4(a). Ω_1 is a free-form shape, Ω_2 is a rectangular hole and Ω_3 is a circular disk attached to Ω_1.



(a) A two dimensional free-form shape combined with two parametric shapes.

(b) The implicit function constructed using R-functions.

Figure 4. Using *R*-functions to combine free-form and parametric shapes.

The constructed implicit representation allows a large class of shapes and shape deformations, accommodating a wide range of implicit modeling techniques. The topological changes can be handled naturally for both free-from shapes and parametric shapes. The resulting level-set functions are differentiable, allowing rigorous shape sensitivity analysis and supporting many gradient-based optimization methods. Last, but not least, we will show that the constructed representation provides flexibility in using parametric design features directly with shape optimization, for example, as obstacles, attachments, to enforce exact geometric requirements, etc.

3 Optimization problem formulation

In the rest of this paper, we will study a particular shape optimization problem in order to illustrate the capability of the proposed representation in shape design. The new representation supports a variety of design problems and design optimization methods with provable properties. For concreteness, we will demonstrate it on a compliance minimization problem with volume constraint that has been studied by others and is well understood. But we stress that our approach can also be applied to other structural optimization problems with similar benefits.

We assume that the shape Ω we seek is contained within a given domain D, $\Omega \subseteq D$. The shape Ω with boundary $\Gamma = \partial\Omega$ is the optimal shape for the following compliance minimization problem:

$$\begin{aligned} \textit{Minimize}\ \ & J_0(u) = \iint_\Omega \frac{1}{2} E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u)d\Omega \\ \textit{subject to}:\ \ & a(u,v) = l(v),\ \forall v \in U \\ & u|_{\Gamma_1} = u_0 \\ & \iint_\Omega d\Omega = V_0 \end{aligned} \tag{3}$$

where $a(u,v) = l(v)$ is the equilibrium equation, $a(u,v) = \iint_\Omega E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(v)d\Omega$, $l(v) = \iint_\Omega fvd\Omega + \int_{\Gamma_2} pvd\Gamma$. The boundary $\Gamma = \Gamma_1 + \Gamma_2$, 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 elastic strain, v is the virtual displacement and U is the space of all admissible displacements. $\iint_\Omega d\Omega = V_0$ is simply the volume constraint.

We represent the shape Ω as a level set of a higher-dimensional function $\Phi(x,t)$ that evolves over time t so that

$$\begin{cases} x \text{ inside } \Omega, & \text{if } \Phi(x,t) > 0 \\ x \text{ on } \Gamma, & \text{if } \Phi(x,t) = 0 \\ x \text{ outside } \Omega, & \text{if } \Phi(x,t) < 0 \end{cases}. \tag{4}$$

If we use the following characteristic function

$$H(\Phi(x,t)) = \begin{cases} 1, \text{ if } \Phi(x,t) \geq 0 \\ 0, \text{ if } \Phi(x,t) < 0 \end{cases}, \tag{5}$$

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

$$\Omega = \{x\,|\,x \in D, \Phi(x,t) \geq 0\} = \{x\,|\,x \in D, H(\Phi) = 1\}. \tag{6}$$

We emphasize that $\Phi(x,t)$ is a general level set function that can represent free-form shapes, parametric shapes or any combinations of both. The shape design space is determined by how this level set function is constructed and how it is allowed to vary. Following [3, 52], we can reformulate Problem (3) as the following:

$$\begin{aligned} \textit{Minimize}\ \ & J_0(u,\Phi) = \iint_D \frac{1}{2} E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u)H(\Phi)d\Omega \\ \textit{subject to}:\ \ & a(u,v,\Phi) = l(v,\Phi),\ \forall v \in U \\ & u|_{\Gamma_1} = u_0 \\ & \iint_D H(\Phi)d\Omega = V_0 \end{aligned} \tag{7}$$

where

$$a(u,v,\Phi) = \iint_D E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(v)H(\Phi)d\Omega \tag{8}$$

$$l(v,\Phi) = \iint_D [fv + div(pvn)]H(\Phi)d\Omega. \tag{9}$$

Note that the traction p is only defined over the traction boundary Γ_2, but in Expression (9), we have an integral of $div(pvn)$ over the entire domain D [2]. Thus, the traction p must be extended from the boundary to D. This can be accomplished, for example, using transfinite interpolation with approximate distance fields as described in [34].

The level set function in Problem (7) can be constructed by:

$$\Phi(x,t) = \Phi(\Phi_f^1, \ldots, \Phi_f^{K_f}, \Phi_p^1, \ldots, \Phi_p^{K_p}), \tag{10}$$

where each $\Phi_f^k(x,t) = \Phi_f^k(x, \mathbf{c}^k(t)) = \sum_{i=1}^{N_k} c_i^k(t)\chi_i^k(x)$, $k = 1,\ldots,K_f$, is an implicitly represented free-form shape, and each $\Phi_p^k(x,t) = \Phi_p^k(x, \mathbf{b}^k(t))$, $k = 1,\ldots,K_p$, is an implicitly represented parametric shape.

By using Expression (10), the original shape optimization formulation as in Problem (3) becomes a fully parameterized optimization formulation as in Problem (7) where the parameters are the coefficients of B-spline basis functions $\{c_i^k, i = 1,\ldots,N_k\}$ in $\Phi_f^k(x, \mathbf{c}^k(t))$, $k = 1,\ldots,K_f$, and geometric dimensions $\{b_j^k, j = 1,\ldots,M_k\}$ in $\Phi_p^k(x, \mathbf{b}^k(t))$, $k = 1,\ldots,K_p$. This parameterization applies to the level set surface $\Phi(x,t)$ instead of the shape itself, allowing topological changes in the shape without the need to track the shape's boundary.

4 Optimization Procedure

Many optimization methods can be used to solve the problem. Since an equality volume constraint is usually difficult to enforce during the optimization process, we use the augmented Lagrangian multiplier method, which is well understood and is widely used (for example, see [23]). By imposing the volume constraint as a penalty term in the objective function, we obtain the following formulation:

$$\begin{aligned} \textit{Minimize} \quad & J(u,\Phi) = J_0(u,\Phi) + \lambda\left(\iint_D H(\Phi)d\Omega - V_0\right) \\ & \qquad + \frac{1}{2\gamma}\left(\iint_D H(\Phi)d\Omega - V_0\right)^2 \\ \textit{subject to}: \quad & a(u,v,\Phi) = l(v,\Phi),\ \forall v \in U \\ & u|_{\Gamma_1} = u_0, \end{aligned} \tag{11}$$

where λ is the Lagrangian multiplier and γ is a pre-defined parameter (typically a very small number). At each iteration, we fix λ and solve Problem (11) for Φ, then we update λ and check for termination criteria. If the termination criteria are not satisfied, we go to the next iteration.

Because the implicit function Φ is fully parameterized, solution of Problem (11) reduces to searching for an optimal shape in the design space spanned by parameters $\{c_i^k\}$ in Φ_f^k and $\{b_j^k\}$ in each Φ_p^k. The differentiability of Φ supports rigorous sensitivity analysis as described in Section 4.2.

4.1 Algorithm

To solve the augmented Lagrangian multiplier subproblem in (11), we use an iterative gradient search method: in each iteration, we find a descent direction (where the objective function decreases) and move the design variables along this descent direction. In the following, we state a generic algorithm for solving Problem (7) considering the most general case: combination of free-form and parametric shape optimization. The algorithm can handle many special cases by updating a chosen subset of the design variables during the optimization process. The basic algorithm consists of the following steps:

1. Initialize the implicit function $\Phi(x,0)$ and stepsize Δt, choose λ and γ.
2. Solve the augmented Lagrangian multiplier subproblem Problem (11)
   - (2.1) Solve the equilibrium equation.
   - (2.2) For **a chosen set** of parameters,
     - (2.2a) Calculate derivatives $\frac{dc_i(t)}{dt}$ and $\frac{db_j(t)}{dt}$.
     - (2.2b) Update the parameters $c_i(t) = c_i(t) + \frac{dc_i(t)}{dt}\cdot\Delta t$ and $b_j = b_j + \frac{db_j(t)}{dt}\cdot\Delta t$
   - (2.3) Check termination criteria for the subproblem. If not satisfied, go to (2.1). The termination criteria is defined as $|\Delta J| \le \varepsilon$, where ε is a predefined small positive number.
3. Update Lagrangian multiplier $\lambda = \lambda + \frac{1}{\gamma}\left(\iint_D H(\Phi)d\Omega - V_0\right)$
4. Check termination condition. If not satisfied, go to 1. The termination criteria is defined as $|\Delta\lambda| \le \delta$, where δ is a predefined small positive number.

This generic algorithm can handle both free-form and parametric shape optimization problems. If only free-form shape optimization is desired, we may only use $\Phi_f(x, \mathbf{c}(t)) = \sum_{i=1}^{N} c_i(t)\chi_i(x)$ to represent the shape. If pure parametric shape optimization is preferred, $\Phi_p(x, \mathbf{b}(t))$ may be used. The algorithm also supports additional control of topological events and how the parametric shapes may be used. For example, if nucleation of holes inside the domain is undesirable, we may choose to perform the sensitivity calculation and update only those parameters that affect the shape's boundary. Or if parametric shapes are used as obstacles, we may simply force the geometric parameters to be constant. These additional controls provide much flexibility in shape optimization. We will illustrate these features of the proposed method in Section 5.

4.2 Sensitivity analysis

The full parameterization of the shape optimization problem transforms it into a sizing problem which has two set of design variables: one is the coefficients of the B-spline basis functions, the other is the geometric dimensions. Because the constructed

level set function Φ is differentiable we can perform rigorous the sensitivity analysis for Problem (11). Since the number of these parameters is typically large (mainly because of the number of B-spline coefficients), for simplicity, we transform the sensitivity with respect to each design variable (coefficient or geometric dimension) to sensitivity with respect to time since each design variable can be regarded as a time-dependent function.

The results of sensitivity analysis are summarized in the following Lemma and Theorems. Lemma 4.1 shows the time derivative of the objective function in Problem (11) for a generic Φ. With the parameterization of Φ, Theorems 4.2, 4.3, and 4.4 derive explicit expressions for shape sensitivities and establish how to compute the descent direction for the objective function in case of free-form, parametric, or combined shape optimization respectively. The proofs of all results are provided in [8].

Lemma 4.1. *For Problem (11), the time derivative of the objective function is*

$$\frac{dJ(u,\Phi)}{dt} = -\int_{\partial\Omega} \frac{d\Phi}{dt} R d\Gamma, \tag{12}$$

where

$$\begin{aligned} R = \; & -fu - div(pun) + \frac{1}{2} E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u) \\ & -\lambda - \frac{1}{\gamma}\left(\iint_D H(\Phi) d\Omega - V_0\right). \end{aligned} \tag{13}$$

Theorem 4.2. (Free-form shape optimization) *In Problem (11), if* $\Phi(x,t) = \Phi_f(x,\mathbf{c}(t)) = \sum_{i=1}^{N} c_i(t)\chi_i(x)$*, then*

$$\left\{ \frac{dc_i(t)}{dt} = \int_{\partial\Omega} \chi_i(x) R d\Gamma, \; i = 1,\ldots N \right\} \tag{14}$$

is a descent direction of Problem (11).

Theorem 4.3. (Parametric shape optimization) *In Problem (11), if we have* $\Phi(x,t) = \Phi_p(x,\mathbf{b}(t))$*, then*

$$\left\{ \frac{db_j(t)}{dt} = \int_{\partial\Omega} \frac{d\Phi_p(x,\mathbf{b}(t))}{db_j} R d\Gamma, \; j = 1,\ldots M \right\} \tag{15}$$

is a descent direction of Problem (11).

In the general case, the boundary $\partial\Omega$ of the shape is defined by the zero level set of the composite function $\Phi(x,t) = \Phi(\Phi_f^1,\ldots,\Phi_f^{K_f},\Phi_p^1,\ldots,\Phi_p^{K_p})$. The boundary $\partial\Omega$ consists of $K_f + K_p$ pieces, $\Gamma_f^k = \partial\Omega \cap \partial\Omega_f^k$, $k = 1,\ldots,K_f$, and $\Gamma_p^k = \partial\Omega \cap \partial\Omega_p^k$, $k = 1,\ldots,K_p$. When every point on $\partial\Omega$ belongs to the boundary of exactly one primitive, a basic result from the theory of R-functions states that the derivative of the function Φ on the boundary $\partial\Omega$ is determined by the derivative of the primitive level set function defining that portion of the boundary. This allows us to decouple the derivative of Φ into derivatives of individual primitives (see proof in [8]).

Theorem 4.4. (General case) *In Problem (11), if* $\Phi(x,t) = \Phi(\Phi_f^1,\ldots,\Phi_f^{K_f},\Phi_p^1,\ldots,\Phi_p^{K_p})$*, and almost every point of* $\partial\Omega$ *belongs to the boundary of exactly one primitive, then*

$$\left\{ \frac{dc_i^k(t)}{dt} = \int_{\Gamma_f^k} \chi_i^k(x) R d\Gamma, \; i = 1,\ldots N_k \right\}, \; k = 1,\ldots,K_f \tag{16}$$

and

$$\left\{ \frac{db_j^k(t)}{dt} = \int_{\Gamma_p^k} \frac{d\Phi_p^k(x,\mathbf{b}^k(t))}{db_j^k} R d\Gamma, \; j = 1,\ldots M_k \right\}, k = 1,\ldots,K_p \tag{17}$$

is a descent direction of Problem (11).

Theorems 4.2 and 4.3 are special cases of the last, most general, Theorem 4.4. In an unlikely situation that the condition in Theorem 4.4 does not hold, i.e. boundaries from different primitives may become coincident, the derivative on the overlapping boundaries becomes indeterminate. The ambiguity may be resolved in several ways, for example, by perturbing the boundaries or by making additional assumptions about the primitives' behavior at the boundary.

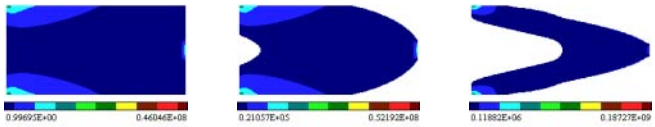
5 Shape control with numerical examples

We now demonstrate shape optimization with parametric, free-form, and topological controls in various combinations, depending on particular design preferences. The design variables (B-spline coefficients and/or geometric dimensions) are updated based on the sensitivity analysis shown in Section 4.2. Observe that the right hand side of Expression (14) is zero if support of a particular function $\chi_i(x)$ does not intersect the boundary. This is to be expected in the case of a free-form shape optimization because the changes in shape are defined in terms of moving the existing boundaries, and is consistent with [16]. It also means that a direct implementation of Expression (14), while theoretically well founded, does not allow changes in topology. We therefore propose an update strategy that permits the evolution of all parameters $\{c_i(t)\}$, including those in the interior. Observing that Expression (14) is simply a measurement of the total effect of the integrand function over the boundary, we define the new updating criteria as the average value of function R over each individual B-spline's support. This is consistent with (14) when the domain is replaced by the boundary, and the integrand function R is well defined at all interior points. In this sense, the proposed strategy is an extension of the sensitivity based approach implied by Expression (14). The numerical results for a short cantilever beam design problem (see Figure 5) are used here to illustrate

the effectiveness of the proposed method. If not specified, the volume constraint is half the area of the design domain. The following parameters are consistently assumed in the examples: the Young's elasticity modulus $E = 1$, Poisson's ratio $\nu = 0.3$, the domain D is of size 0.1×0.05, a distributed force $p = 200$ is applied in a interval of 0.005 around the middle point of the right edge of D and the left edge of D is fixed, the body force $f = 0$. The numerical implementation details are described in the next section.

Figure 5. Definition of the minimum compliance problem for a short cantilever beam.

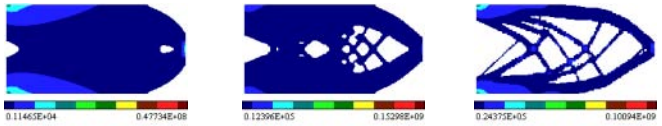
Case 1: Free-form, without nucleation of holes In case of pure free-form shape design, we only have the B-spline coefficients $\{c_1, c_2, \ldots, c_N\}$ as design variables. Only the coefficients of those B-splines whose support intersect the boundary are updated. Figure 6 shows the results at different stages of the optimization process. (The color map shows the distribution of strain energy in this example and all examples that follow.)



(a) Initial design (b) Iteration 60 (c) Iteration 141

Figure 6. Strain energy distribution of the shapes at different iterations from the free-form shape optimization process without nucleation of holes. Only the coefficients of those B-spline basis functions that have support intersecting the boundary are updated. Grid size = 100×50.

Case 2: Free-form, with topological changes In this case, all the B-spline coefficients are updated to allow nucleation of holes in the interior of the domain. Figure 7 shows the results at different stages of the optimization process.



(a) Iteration 28 (b) Iteration 40 (c) Iteration 111

Figure 7. Strain energy distribution of the shapes at different iterations from the free-form shape optimization process with nucleation of holes. Every B-spline coefficient is updated during the optimization process. Grid size = 100×50.

Case 3: Parametric, with or without topological changes For pure parametric design, the design variables are geometric parameters $\{b_1, b_2, \ldots, b_M\}$. We use Theorem 4.3 to update these parameters. Figure 8 shows the optimal shape of a rectangle with a circular hole, with the position of the hole as the only design parameter. Figure 9 shows the optimal shape of a rectangle with a circular hole and a rectangular slot, where positions of both hole and slot are used as design variables. Due to the nature of the design problem, no volume constraint is imposed for these two examples. The topological changes, such as intersection of hole and the slot, are handled easily without any additional effort.

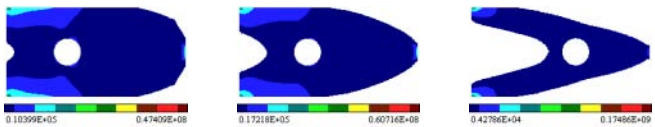
(a) Initial design (b) Result at iteration 9

Figure 8. Strain energy distribution of the initial and optimal shapes of a rectangle with a circular hole. The design variables are the position of the hole. Grid size = 50×25.

(a) Initial design (b) Result at iteration 21

Figure 9. 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. Grid size = 50×25.

Case 4: Free-form and parametric, without nucleation of holes In this case, both the B-spline coefficients $\{c_1, c_2, \ldots, c_N\}$ and geometric parameters $\{b_1, b_2, \ldots, b_M\}$ are used as design variables. To prevent nucleation of holes, we only update the boundary B-spline coefficients (as in Case 1) and geometric parameters. Figure 10 shows the example of a moving circular hole inside the domain where the position of the hole needs to be optimized.



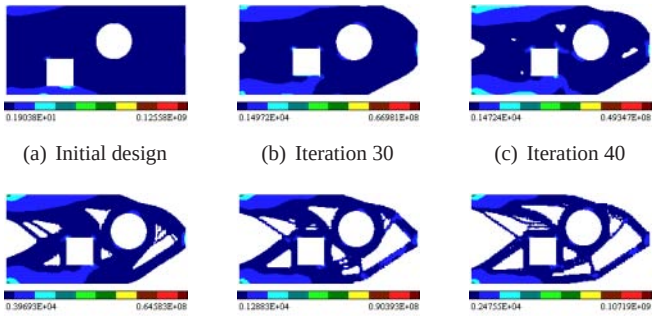
(a) Iteration 40 (b) Iteration 80 (c) Iteration 135

Figure 10. Strain energy distribution at different iterations during the optimization process for free-form shape with a circular hole. Grid size = 100×50.

Case 5: Free-form and parametric, with topological changes

Just as in Case 2, every B-spline coefficient is updated to allow nucleation of holes. Geometric parameters are updated to

optimize the parametric shapes. Figure 11 shows the results at different stages of the optimization process for a shape with a circular hole and a rectangular slot. In this example, we protect the parametric shapes throughout the optimization process (which means the circular hole remains circular and the rectangular slot remains rectangular). This is achieved by simply placing a tolerance zone around each parametric shape and force the B-spline coefficients inside the tolerance zone to be positive.



(a) Initial design (b) Iteration 30 (c) Iteration 40

(d) Iteration 60 (e) Iteration 80 (f) Iteration 134

Figure 11. Strain energy distribution at different iterations during the optimization process for free-form shape with a circular hole and a rectangular slot. Nucleation of holes is allowed. Grid size = 100×50.

Case 6: Parametric features as obstacles or attachments Figure 12 shows the optimal shape with a circular hole as a moving obstacle. Figure 13 shows the optimal shape with a circular disk as a fixed attachment. The obstacle can be combined with free-from shape design using the intersection operation while the attachment can be implemented by the union operation.

(a) Initial design (b) Optimal shape

Figure 12. Strain energy distribution of the initial and optimal free-form shape with a circular hole as a moving obstacle. Grid size = 50×25.

(a) Initial design (b) Optimal shape

Figure 13. Strain energy distribution of the initial and optimal free-form shape with a disk as an attachment. Grid size = 100×50.

6 Implementation

The proposed approach to shape optimization can be implemented in many environments that support stress/strain analysis, allow some programmability for parametric functions, and provide tools for differentiation, and boundary and volume integration. A potentially challenging task that is likely to dominate any implementation is numerical integration over an evolving (a priori unknown) geometric domain Ω. The task makes implementation with finite elements challenging because, as the shape changes, it would require either frequent remeshing or (re) approximation of the domain by piecewise linear functions.

6.1 Meshfree approach with distance fields

The meshfree method with approximated distances described in [13] is based on the original idea by Kantorovich [18] for solving simple Dirichlet problems, but was fully developed by Rvachev and his students for general boundary conditions and problems [31, 32]. To paraphrase Rvachev, a physical field can be represented by a generalized Taylor series by powers of an approximate distance field to the boundary [31, 33].[2] Once such distance fields are constructed, they can be used to construct solutions to boundary value problems that satisfy the prescribed boundary conditions exactly on all points where the distance field vanishes. The remainder term in the Taylor series contains degrees of freedom necessary to approximate differential equation(s), and it also assures completeness of the solution [33]. The method is essentially meshfree, though a background mesh may be used for integration and visualization purposes. A restricted implementation of the method with WEB-Splines is described in [17], and a complete programming environment supporting construction, differentiation, and integration of all required functions at run time is described in [49] and was used to implement the proposed approach to shape optimization.

6.2 Solution procedure

In the context of the structural analysis problem solved in this paper, we represent components of the displacement vector $\mathbf{u} = (u_1, u_2)$ as products of two functions $u_i = \omega_i \Psi_i$, $i = 1, 2$, where ω_i are distance functions to the fixed portions of the boundary of the domain Ω, and functions $\Psi_i = \sum_{j=1}^{k} a_j^{\Psi_i} \xi_j$ are linear combinations of basis functions used to approximate solution of the differential equation. Generally, these basis functions can be defined on a grid that does not conform to the geometric domain and are *not* related to the basis functions used to construct the level set functions Φ_i. In this paper we approximated components of the displacement vector using uniform cartesian grid of

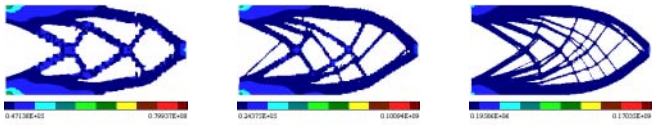
[2] Rvachev does not use distance fields directly but employs so called "normalized functions" that are constructed using theory of *R*-functions [31]. With that terminology, a level set function is 0-th level approximation of a distance from any particular level set.

bilinear B-splines. Numerical values of the coefficients $a_j^{\Psi_i}$ are determined by a standard technique that requires minimization of an energy functional [55]. As a result we obtain a system of linear algebraic equations whose solution gives numerical values of the coefficients $a_j^{\Psi_i}$. Assembly of the matrix and vector of this system of equations requires differentiation of the approximate distance fields ω_i and basis functions with respect to spatial coordinates and integration over non-meshed geometric domain and its boundary that is represented by a level set function. Once numerical values of the coefficients $a_j^{\Psi_i}$ are computed, they are substituted into expressions for components of the displacement vector $\mathbf{u}$.

The free-form level set function is initialized as a B-spline surface with constant coefficients over the rectangular domain. It is then combined with the parametric shapes to obtain the initial level set function Φ using R-conjunction and/or R-disjunction as appropriate. At each step of optimization, the structural problem is solved using the meshfree method as described above. The value of R in Expression (13) is computed from the solution field, then the derivatives $\{\frac{dc_i(t)}{dt}\}$ can be directly calculated from Expression (16). Note that the derivatives $\{\frac{db_j(t)}{dt}\}$ require boundary integration. The integrand in Expression (17) is available since the derivatives of $\{\frac{d\Phi_p}{db_j}\}$ can be easily calculated from the explicit expression of parametric level set functions. The boundary of each parametric shape is known a priori, however some points of the parametric boundary may not lie on the boundary of the final shape Ω. For example, in Figure 9 only a portion of the circle is a subset of the boundary after the circle and the rectangle merge. The boundary points are identified through a simple point membership test against the implicitly represented level set $\Phi = 0$. This eliminates the need to track parametric boundaries – a difficult task associated with traditional parametric optimization.

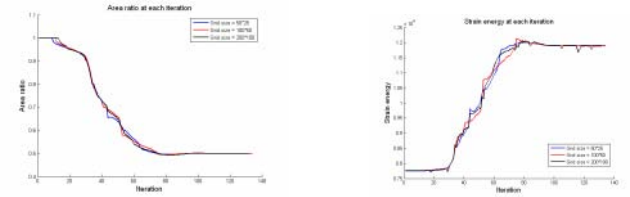
6.3 Dependence on grid size

The algorithm converges rapidly and smoothly to (local) minima for the examples shown in Section 5. For problems that involve free-form boundaries and allow topological changes, the results are clearly dependent on the number of B-splines $\chi_i(x)$ used to represent the free-form component Φ_f of the shape. Figure 14 shows three different shapes resulting from the same optimization problem that also produced the shape in Figure 7, but with different grids of B-splines: 50×25, 100×50 and 200×100. Although the final shapes are very different, the values of the objective function are very close to each other. Figure 15 shows the values of the objective function and the area ratios for the shapes during the optimization process for the three grid sizes. The observed grid dependence should be expected, since it is well known that the original shape optimization problem (3) is not well posed as stated [12, 4]. Without changing the structure volume, introduction of more holes will generally increase the efficiency of the structure *ad infinitum*. The size of the grid defines the number of B-splines, which serve as degrees of freedom for the level set surface, and is directly related to the maximum number of possible holes. If we disallow introduction of new holes, the grid dependence disappears (Figure 16 shows two similar shapes from different grid size.) The limit of the continuous grid refinement is often associated with micro-structured materials, the latter also provide the basis for the homogenization method.



(a) 50×25 grid (b) 100×50 grid (c) 200×100 grid

Figure 14. Strain energy distribution of the optimal shapes from different grid size in the free-form shape optimization with nucleation of holes.



(a) The area ratio during optimization process (b) The strain energy during optimization process

Figure 15. The area ratio and strain energy during optimization process for different grid size.

(a) Result from 50×25 grid (b) Result from 100×50 grid

Figure 16. Strain energy distribution of the optimal shapes from different grid size in the free-form shape optimization without nucleation of holes.

7 Conclusions

We proposed a new method for shape optimization that combines and subsumes free-form and parametric shape optimization approaches. The resulting space of shapes is fully parameterized by the B-spline coefficients (for the free-form boundaries) and geometric dimensional parameters, without restricting the parameterization to any particular topology. This parametrization transforms the difficult shape and topology optimization problems into a relatively straightforward sizing problem to which

many gradient-base optimization techniques can be applied. Further the differentiability of the constructed level set function supports rigorous shape sensitivity analysis, where free-form and parametric shape sensitivity can be treated simultaneously.

The generality and flexibility of the proposed approach are demonstrated by numerical examples for a two dimensional minimum compliance problem. To our knowledge, this is the first work on shape optimization that combines free-form shape optimization and parametric shape optimization. Many existing shape optimization methods can be treated as special cases in our approach. In level set method, a boundary velocity V_n is constructed to guarantee a descent direction [52, 2]. This technique can also be implemented in our approach if we apply level set equation $\frac{d\Phi}{dt} = -|\nabla\Phi| \cdot V_n$ in Expression (12). Use of shape functionals to represent level set surfaces for shape optimization as proposed in [54] and [3] corresponds to special cases of free-form optimization into our approach. Last, but not least, traditional parametric shape optimization is already built in our approach, but with added ability to control topological changes throughout the optimization process.

ACKNOWLEDGMENT

This research is supported in part by the National Science Foundation grants DMI-0323514, DMI-0115133 and DMI-0500380, and Wisconsin Industrial & Economic Development Research Program (I&EDR).

REFERENCES

[1] G. Allaire, F. de Gournay, F. Jouve, and A. M. Toader. Structural optimization using topological and shape sensitivity via a level set method. *Control and Cybernetics*, 34:59–80, 2005.

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

[3] T. Belytschko, S. P. Xiao, and C. Parimi. Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering*, 57:1177–1196, 2003.

[4] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer Verlag, Berlin Heidelberg, 2003.

[5] A. Biswas and V. Shapiro. Approximate distance fields with non-vanishing gradients. *Graphical Models*, 66(3):133–159, May 2004.

[6] J. Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers, 1997.

[7] M. Burger, B. Hackl, and W. Ring. Incorporating topological derivatives into level set methods. *Journal of Computational Physics*, 194:344–362, 2004.

[8] J. Chen, V. Shapiro, K. Suresh, and I. Tsukanov. Shape optimization with topological changes and parametric control. Technical Report SAL 2006-1, Spatial Automation Laboratory, University of Wisconsin, February 2006.

[9] S. Chen and D. A. Tortorelli. Three-dimensional shape optimization with variational geometry. *Structural Optimization*, 13:81–94, 1997.

[10] Carl de Boor. *A Practical Guide to Splines*. Springer-Verlag, 2001.

[11] H. A. Eschenauer, H. A. Kobelev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural Optimization*, 8:142–151, 1994.

[12] H. A. Eschenauer and N. Olhoff. Topology optimization of continuum structures: A review. *Applied Mechanics Review*, 54:331–390, 2001.

[13] M. Freytag, V. Shapiro, and I. Tsukanov. Field modeling with sampled distances. *Computer Aided Design*, 38(2):87–100, 2006.

[14] Sarah F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 249–254, 2000.

[15] S. Garreau, M. Masmoudi, and P. Guillaume. The topological sensitivity for linear isotropic elasticity. *ECCM'99*. Munich, Germany.

[16] E. J. Haug, K. K. Choi, and V. Komkov. *Design Sensitivity Analysis of Structural Systems*. Academic Press, New York, NY, 1986.

[17] K. Höllig. *Finite Element Methods with B-Splines*. Number 26 in Frontiers in Applied Mathematics. SIAM, 2003.

[18] L. V. Kantorovich and V. I. Krylov. *Approximate Methods of Higher Analysis*. Interscience Publishers, 1958.

[19] N. Kojekine, I. Hagiwara, and V. Savchenko. Software tools using csrbf's for processing scattered data. *Computers and Graphics*, 27(2), 2003.

[20] C. Lim, G. M. Turkiyyah, M. A. Ganter, and D. W. Storti. Implicit reconstruction of solids from cloud point sets. In *Proceedings of the Third Symposium on Solid Modeling and Applications*, pages 393–402. ACM Press, 1995.

[21] S. Muraki. Volumetric shape description of range data using "Blobby Model. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 25(4):227–235, July 1991.

[22] D. Natekar, X. Zhang, and G. Subbarayan. Constructive solid analysis: A hierarchical, geometry based meshless procedure for integrated design and analysis. *Computer Aided Design*, 36:472–486, 2004.

[23] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag, 1999.

[24] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Transactions*

on Graphics (TOG), 22(3):463–470, 2003.

[25] S. Raghothama and V. Shapiro. Consistent updates in dual representation systems. *Computer-Aided Design*, 32(8–9):463–477, 2000. A preliminary version appeared in Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications, Ann Arbor, MI, June 1999.

[26] Alon Raviv and Gershon Elber. Three-dimensional freeform sculpting via zero sets of scalar trivariate functions. *Computer-Aided Design*, 32:513–526, 2000.

[27] A. Requicha. Representations for rigid solids: Theory, methods, and systems. *Computing Surveys*, 12(4):437–464, 1980.

[28] A. Ricci. A constructive geometry for computer graphics. *Computer Journal*, 16(3):157–160, May 1973.

[29] V. L. Rvachev. Analytical description of some geometric objects. *Dokl AS USSR*, 153(4):765–768, 1963.

[30] V. L. Rvachev. *Geometric Applications of Logic Algebra*. Naukova Dumka, 1967. In Russian.

[31] V. L. Rvachev. *Theory of R-functions and Some Applications*. Naukova Dumka, 1982. In Russian.

[32] V. L. Rvachev and T. I. Sheiko. *R*-functions in boundary value problems in mechanics. *Applied Mechanics Reviews*, 48(4):151–188, 1996.

[33] V. L. Rvachev, T. I. Sheiko, V. Shapiro, and I. Tsukanov. On completeness of RFM solution structures. *Computational Mechanics*, 25:305–316, 2000.

[34] V. L. Rvachev, T. I. Sheiko, V. Shapiro, and I. Tsukanov. Transfinite interpolation over implicitly defined sets. *Computer Aided Geometric Design*, 18(4):195–220, 2001.

[35] Vladimir V. Savchenko, Alexander A. Pasko, Oleg G. Okunev, and Tosiyasu L. Kunii. Function Representation of Solids Reconstructed from Scattered Surface Points and Contours. *Computer Graphics Forum*, 14(4):181–188, 1995.

[36] Benjamin Schmitt, Alexander Pasko, and Christophe Schlick. Constructive sculpting of heterogeneous volumetric objects using trivariate b-splines. *The Visual Computer*, 20(2):130–148, may 2004.

[37] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

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

[39] V. Shapiro. Theory of R-functions and applications: A primer. *Technical Report, Cornell University*, November 1988.

[40] V. Shapiro. Real functions for representation of rigid solids. *Computer-Aided Geometric Design*, 11(2):153–175, 1994.

[41] V. Shapiro. Maintenance of geometric representations through space decompositions. *International Journal on Computational Geometry and Applications*, 7(4):383–418, 1997.

[42] V. Shapiro. Well-formed set representations of solids. *International Journal on Computational Geometry and Applications*, 9(2):125–150, 1999.

[43] V. Shapiro. A convex deficiency tree algorithm for curved polygons. *International Journal of Computational Geometry and Applications*, 11(2):215–238, 2001.

[44] V. Shapiro. Solid modeling. In G. Farin, J. Hoschek, and M. S. Kim, editors, *Handbook of Computer Aided Geometric Design*, pages 473–518. Elsevier Science Publishers, 2002.

[45] V. Shapiro and I. Tsukanov. Implicit functions with guaranteed differential properties. In *Fifth ACM Symposium on Solid Modeling and Applications*, pages 258–269, Ann Arbor, MI, 1999.

[46] V. Shapiro and I. Tsukanov. Meshfree simulation of deforming domains. *Computer Aided Design*, 31:459–471, 1999.

[47] V. Shapiro and D. L. Vossler. Construction and optimization of CSG representations. *Computer-Aided Design*, 23(1):4–20, January/February 1991.

[48] V. Shapiro and D. L. Vossler. What is a parametric family of solids? In *3rd ACM Symposium on Solid Modeling and Applications, Salt Lake City, Utah, May*, 1995.

[49] I. Tsukanov and V. Shapiro. The architecture of SAGE – a meshfree system based on RFM. *Engineering with Computers*, 18(4):295–311, 2002.

[50] G. Turk and J. O'Brien. Modeling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, 21(4):855–873, October 1999.

[51] Luiz Velho, Jonas Gomes, and Luiz H. de Figueiredo. *Implicit Objects in Computer Graphics*. Springer, 2002.

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

[53] M. Y. Wang and P. Wei. Topology optimization with level set method incorporating topological derivative. In *6th World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 2005.

[54] S. Y. Wang and M. Y. Wang. Radial basis functions and level set method for structural topology optimization. *International Journal for Numerical Methods in Engineering*, 65(12):2060–2090, 2005.

[55] K. Washizu. *Variational methods in elasticity and plasticity*. Oxford, Eng. ; New York : Pergamon Press, 3rd edition, 1982.

[56] Y. M. Xie and G. P. Steven. A simple evolutionary procedure for structural optimization. *Computers and Structures*, 49(5):885–896, 1993.