INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng (in press)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1943

Shape optimization with topological changes and parametric control

Jiaqin Chen, Vadim Shapiro*,†, Krishnan Suresh and Igor Tsukanov

Spatial Automation Laboratory, 1513 University Avenue, University of Wisconsin, Madison, WI 53706, U.S.A.

SUMMARY

Recent advances in shape optimization rely on free-form implicit representations, such as level sets, to support boundary deformations and topological changes. By contrast, parametric shape optimization is formulated directly in terms of meaningful geometric design variables, but usually does not support free-form boundary and topological changes. 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 free-form 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 explicit construction of an implicit representation supports straightforward sensitivity analysis that can be used with most gradient-based optimization methods. Furthermore, our implementation does not require any error-prone polygonization or approximation of level sets (isocurves and isosurfaces). The effectiveness of the method is demonstrated by several numerical examples.

Received 17 February 2006; Revised 10 October 2006; Accepted 23 October 2006

KEY WORDS: shape optimization; topology optimization; parametric design; level set; implicit representation; *R*-functions; shape sensitivity analysis

*Correspondence to: Vadim Shapiro, Spatial Automation Laboratory, 1513 University Avenue, University of Wisconsin, Madison, WI 53706, U.S.A.
†E-mail: vshapiro@engr.wisc.edu

Contract/grant sponsor: National Science Foundation; contract/grant numbers: DMI-0621116, OCI-0537370, DMI-0323514, DMI-0500380
Contract/grant sponsor: Wisconsin Industrial & Economic Development Research Program (I&EDR)

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 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. Rather, the boundaries are defined only procedurally (i.e. *via* some additional construction procedure) by the allowed parameter values. Consequently, in parametric shape optimization, every change in the parameter values requires re-evaluation of the corresponding boundary representation. This, in turn, leads to at least two widely acknowledged difficulties: boundary evaluation may fail [1, 2], and topological changes in the boundaries may invalidate boundary conditions or the solution procedure [3]. 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.

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 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 $\Phi(x, t)$ [4]. 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. The advantages of using this implicit representation and posing the optimization problem on the implicit surface have been well demonstrated by recent advances in shape optimization methods [5–7]. 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 [8, 9]. 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 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.[‡] The output of free-form optimization is then transformed into detailed shape design, typically relying

[‡]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.

DOI: 10.1002/nme

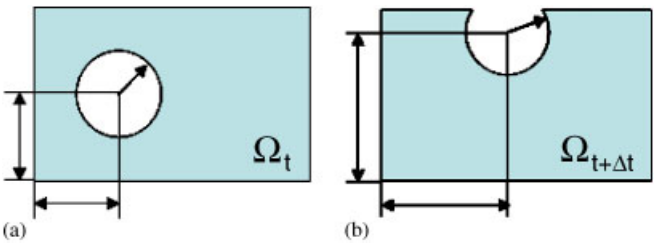

Figure 1. Traditional parametric shape optimization and limitations: (a) a simple shape is parameterized by three dimensions that procedurally define the shape's boundary; and (b) changes in parameter values that result in a different topology are usually not allowed.

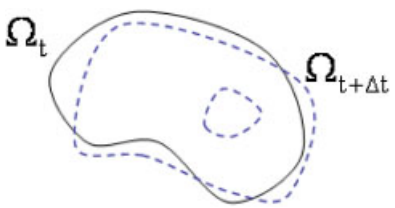

Figure 2. Free-form deformation with topological changes can be represented by a moving level set hyper-surface.

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 construct such representations using Boolean set operations on simpler implicitly represented and parameterized shapes (free form and/or parametric). The usual difficulty with this approach is that the resulting representation is a Boolean expression and is *not* a hyper-surface. It cannot be differentiated as required for sensitivity analysis [10], and cannot be treated as a single analytic representation of a shape. We show that this problem is readily solved using R-functions [11, 12] that transform any set-theoretic construction into a sufficiently smooth hyper-surface *via* simple syntactic substitution.

The proposed approach to shape optimization may be implemented using standard shape optimization tools, has many advantages in terms of shape control, and can be applied to a large class of shape design and optimization problems. We discuss its implementation in a mesh-free environment developed by the authors [13, 14] and demonstrate its performance by applying it to a widely studied minimum compliance structural shape optimization problem.

DOI: 10.1002/nme

1.2. Related work

Parametric shape optimization is a well understood subject that can be found in many texts (for example, see [10, 15]). Parametric shape optimization may be implemented within any CAD system supporting automated analysis in an obvious and brute force fashion, using finite differences to compute shape sensitivities with respect to the parameters and (re)meshing the shape at every time step as needed. A method for computing velocity of the shape's boundaries in terms of dimensional variations is described in [3]. In [16], the parametric approach is extended to include free-form primitives represented by NURBS (non-uniform rational B-splines) with iso-parametric mapping of the geometry and the behaviour field, in support of simultaneous analysis and design. As we already discussed above, all these approaches suffer from the limitations associated with any parametric shape optimization.

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, proposed in [17] and improved by many other authors, relies on criteria for element addition and removal based on structural analysis results. Bubble method [18] uses iterative introduction, positioning of new holes and parametrization of the boundaries to achieve an optimal design. A similar approach using the topological sensitivity analysis is also presented in [19] and is given a mathematical justification in [20] using the concept of topological derivative.

Recently proposed level set methods in shape optimization have attracted much attention, due to their ability to track evolving boundaries and handle topological changes [4]. In [5], 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 [6, 7] shows 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 [6]. 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 [5–7]. Another approach to handling topological changes with level sets is described in [8], 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 [9] represent level set function by finite element shape functions and rely on a heuristic criterion to perform shape and topology optimization. The concept of topological derivative has been proposed to overcome the difficulty of generating interior holes [21–23] and appears to be promising.

Another important class of methods for topology optimization is the material distribution-based methods such as homogenization method and SIMP (solid isotropic material with penalization) method. Homogenization method uses material model with micro-structures to seek the optimal layout over the design domain and generates a porous medium. The SIMP method uses a continuous material density distribution resulting in a grey-scaled-image-like material density distribution.

DOI: 10.1002/nme

A comprehensive description on homogenization method and SIMP method can be found in the book by Bendsøe and Sigmund [24]. We will not consider material distribution methods in this paper, but focus exclusively on geometric methods that formulate shape optimization in terms of moving boundaries and (dis)appearance of voids.

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 [8, 9], 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

The main difficulty in combining free-form shapes and parametric shapes lies in finding a way to unite these two seemingly different representations into a single analytical and computational framework which supports free form and parametric shape control simultaneously, and lends itself to rigorous shape sensitivity analysis to guide the shape optimization procedure. 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 [25].

A typical shape in this space of level sets has boundaries that are partially free form and partially parametric, with purely parametric and purely free-form shapes as special cases. The parametric boundaries provide control of the shape in terms of meaningful engineering and geometric features, while the free-form boundaries expand the design space to allow virtually unlimited shape variations. Topological changes are easily handled as with any implicit shape representation.

2.1. Implicit representations of shapes

Implicit representations of shapes have a long tradition in geometric modelling and computer graphics, as described in several recent books [26, 27]. 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 $\Omega = \{x \in D | \Phi(x) \geqslant 0\}$, 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 $\partial\Omega = \{x \in D \mid \Phi(x) = 0\}$. Figure 3(b) shows a level set function for the geometric domain in 3(a).

DOI: 10.1002/nme

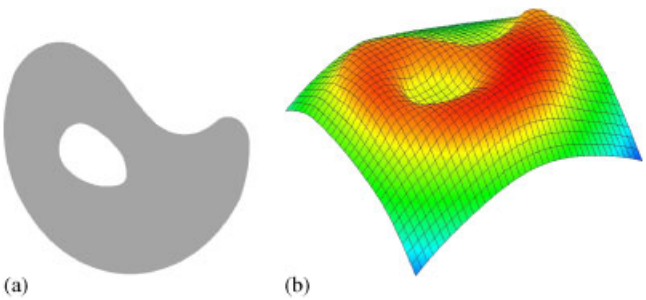

Figure 3. Implicit representation for the shape: (a) a two-dimensional shape defined by the level set of the function in (b); and (b) a three-dimensional level-set function defines the two-dimensional shape by its zero level set.

This definition is consistent with the notion of level set function in [4–9], but also includes many other representations used in geometric modelling. Many techniques and transformations for constructing such representations are described in [26], including Ricci's function [28], theory of *R*-functions [11, 12, 29, 30], and convolution methods. More recent notable methods include exact and approximate distance fields [31, 32], blending of implicit primitives like blobs, spheres, quadrics, and local quadratics that have been fit to the points [33–35], radial basis functions with both global [36] and compact support [37, 38], and multi-variate B-splines to represent scalar fields whose zero sets represent the boundary of sculpted geometry [39, 40]. Implicit representations may be constructed from both constructive solid geometry and boundary representations of geometric objects [30, 41, 42]. Although implicit representations lack explicit boundary information [43], 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 $\Phi(x, t)$. There are, of course, infinitely many functions Φ that correspond to the same geometry. Our goal is to construct a large and useful class of functions $\{\Phi\}$ that are suitable for shape optimization, and to define an optimization procedure that finds a function corresponding to the optimal shape. We will do so by combining primitive level set functions Φ that can be free form or parametric.

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 [26]. 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 $\Phi_{\mathrm{p}}(x, \mathbf{b}(t))$ 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 [13, 25, 26]. If the implicit function $\Phi_{\mathrm{p}}(x, \mathbf{b}(t))$ is constructed from two primitive implicit representations $\Phi^1_{\mathrm{p}}(x, \mathbf{b}^1(t))$ and $\Phi^2_{\mathrm{p}}(x, \mathbf{b}^2(t))$, then the vector of parameters $\mathbf{b}$ is simply an (ordered) union of $\mathbf{b}^1$ and $\mathbf{b}^2$. For

DOI: 10.1002/nme

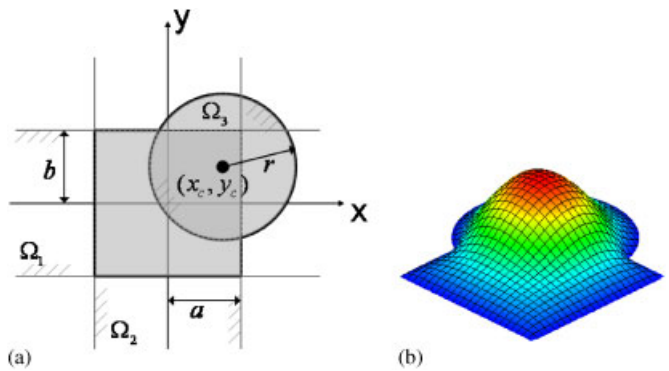

Figure 4. Using R-functions to construct complex parametric shapes: (a) a two-dimensional parametric shape; and (b) a level set function constructed using R-functions.

example, Figure 4(a) shows a shape defined by a union of an origin-centred rectangle and a circle. The corresponding space of shapes is completely parameterized in terms of dimensions of the rectangle a and b, the circle's radius r and its position x_c, y_c. Figure 4(b) shows one corresponding level set function $\Phi(x, a, b, r, x_c, y_c)$ that is parameterized in terms of the same geometric variables. We will explain how to construct such functions automatically using R-functions in Section 2.3.

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 [44]. 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 with coefficients $\{c_i\}$ as the design variables. 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 4(a) is described by the Boolean expression $\Omega = (\Omega_1 \cap \Omega_2) \cup \Omega_3$,

DOI: 10.1002/nme

where Ω_i, $i = 1, 2, 3$ are primitives shapes represented implicitly by the corresponding level-set functions Φ^i:

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

More generally, most shapes in geometric modelling belong to the class of semi-analytic sets that, by definition, can be constructed using logical operations on equalities and inequalities with analytic functions [43, 45]. Constructive solid geometry representations rely on such logic expressions explicitly, but they can also be constructed automatically from a variety of other representations [41, 46, 47].

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 [12, 30, 48]. They were discovered by Rvachev who developed the theory specifically for solution of boundary value problems in mechanics [29, 49]. 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 [11, 12, 25, 29].

Because composition of *R*-functions is another *R*-function, it suffices to choose only two *R*-functions: *R*-conjunction corresponding to logical $\wedge$ and *R*-disjunction corresponding to logical $\vee$. By Boolean properties, *R*-functions for all other Boolean expressions are simply compositions of these two *R*-functions. A popular system of these functions includes:

$$f_1 \wedge_0 f_2 \equiv f_1 + f_2 - \sqrt{f_1^2 + f_2^2}, \quad f_1 \vee_0 f_2 \equiv f_1 + f_2 + \sqrt{f_1^2 + f_2^2} \tag{2}$$

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. Thus, the function shown in Figure 4(b) was constructed as $\Phi(x, y) = (\Phi^1(x, y) \wedge_0 \Phi^2(x, y)) \vee_0 \Phi^3(x, y)$.

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 5(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 5(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.

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

DOI: 10.1002/nme

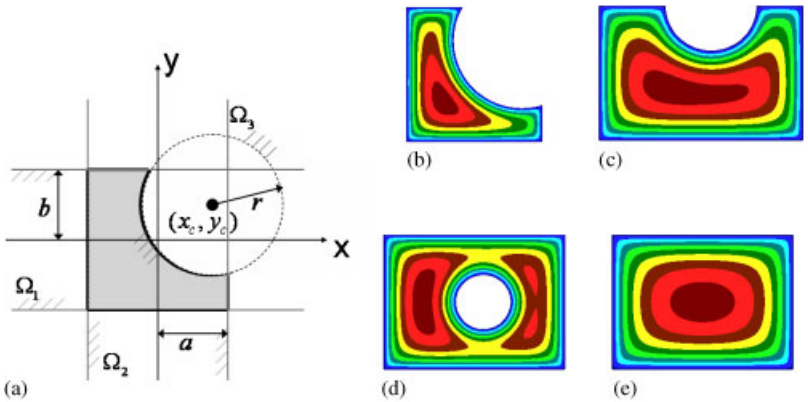

Figure 5. Parametric shape deformations corresponding to changes in values of geometric parameters: (a) parametric shape defined as $\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3$; and (b-e) different shapes generated by parameter changes.

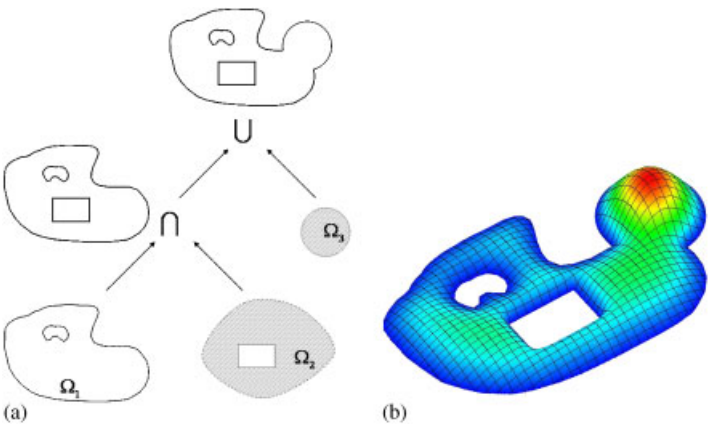

Figure 6. Using R-functions to combine free form and parametric shapes: (a) a two-dimensional free-form shape combined with two parametric shapes; and (b) the implicit function constructed using R-functions.

shapes $\{\Phi_{\mathrm{f}}^1, \ldots, \Phi_{\mathrm{f}}^{K_{\mathrm{f}}}\}$, and a collection of K_{p} parametric shapes $\{\Phi_{\mathrm{p}}^1, \ldots, \Phi_{\mathrm{p}}^{K_{\mathrm{p}}}\}$. Each primitive free-form function is defined by a linear combination of some basis functions $\Phi_{\mathrm{f}}^k = \Phi_{\mathrm{f}}^k(x, \mathbf{c}^k(t)) = \sum_{i=1}^{N_k} c_i^k(t)\chi_i^k(x)$ for $k = 1, \ldots, K_{\mathrm{f}}$, and each parametric function is parameterized in terms of geometric parameters as $\Phi_{\mathrm{p}}^k = \Phi_{\mathrm{p}}^k(x, \mathbf{b}^k(t))$ for $k = 1, \ldots, K_{\mathrm{p}}$. 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 $\Phi(x, t) = \Phi(\Phi_{\mathrm{f}}^1, \ldots, \Phi_{\mathrm{f}}^{K_{\mathrm{f}}}, \Phi_{\mathrm{p}}^1, \ldots, \Phi_{\mathrm{p}}^{K_{\mathrm{p}}})$, where Φ is an R-function corresponding to the Boolean function.

Figure 6(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 6(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.

DOI: 10.1002/nme

To summarize, combining the implicit representations for free-form shapes and parametric shapes using R-functions offers much flexibility and generality in shape optimization. Free-form shape design and parametric shape design can be treated as special cases, and one may choose to select either one of them in certain situations without any additional effort. The constructed implicit representation allows a large class of shapes and shape deformations, accommodating a wide range of implicit modelling 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:

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

where $a(u, v) = l(v)$ is the equilibrium equation, $a(u, v) = \int\!\!\int_\Omega E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(v)\,\mathrm{d}\Omega$, $l(v) = \int\!\!\int_\Omega f v\,\mathrm{d}\Omega + \int_{\Gamma_2} pv\,\mathrm{d}\Gamma$. The boundary $\Gamma = \Gamma_1 + \Gamma_2$, Dirichlet 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. $\int\!\!\int_\Omega \mathrm{d}\Omega = V_0$ 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

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

DOI: 10.1002/nme

If we use the following characteristic function:

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

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

$$\Omega = \{x \mid x \in D, \Phi(x,t) \geqslant 0\} = \{x \mid x \in D, H(\Phi) = 1\} \tag{6}$$

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 [7, 9], we can reformulate Problem (3) as the following:

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

where

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

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

Note that the traction p is only defined over the traction boundary Γ_2, but in Expression (9), we have an integral of $\mathrm{div}(pvn)$ over the entire domain D [6]. 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 [50].

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

$$\Phi(x,t) = \Phi(\Phi_{\mathrm{f}}^1, \ldots, \Phi_{\mathrm{f}}^{K_{\mathrm{f}}}, \Phi_{\mathrm{p}}^1, \ldots, \Phi_{\mathrm{p}}^{K_{\mathrm{p}}}) \tag{10}$$

where each $\Phi_{\mathrm{f}}^k(x,t) = \Phi_{\mathrm{f}}^k(x, \mathbf{c}^k(t)) = \sum_{i=1}^{N_k} c_i^k(t)\chi_i^k(x)$, $k = 1, \ldots, K_{\mathrm{f}}$, is an implicitly represented free-form shape, and each $\Phi_{\mathrm{p}}^k(x,t) = \Phi_{\mathrm{p}}^k(x, \mathbf{b}^k(t))$, $k = 1, \ldots, K_{\mathrm{p}}$, 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 $\Phi_{\mathrm{f}}^k(x, \mathbf{c}^k(t))$, $k = 1, \ldots, K_{\mathrm{f}}$, and geometric dimensions $\{b_j^k, j = 1, \ldots, M_k\}$ in $\Phi_{\mathrm{p}}^k(x, \mathbf{b}^k(t))$, $k = 1, \ldots, K_{\mathrm{p}}$. Also, due to the differentiability of function $\Phi(x,t)$, the shape sensitivity analysis is available by computing the

DOI: 10.1002/nme

derivatives of the objective function with respect to these parameters, as will be explained in next section. 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 [51]). By imposing the volume constraint as a penalty term in the objective function, we obtain the following formulation:

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

where λ is the Lagrangian multiplier and γ is a predefined 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:

Step 1: Initialize the implicit function $\Phi(x, 0)$ and stepsize Δt, choose λ and γ.
Step 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 $\frac{\mathrm{d}c_i(t)}{\mathrm{d}t}$ and $\frac{\mathrm{d}b_j(t)}{\mathrm{d}t}$.
(2.2b) Update the parameters $c_i(t) = c_i(t) + \frac{\mathrm{d}c_i(t)}{\mathrm{d}t} \cdot \Delta t$ and $b_j = b_j + \frac{\mathrm{d}b_j(t)}{\mathrm{d}t} \cdot \Delta t$

DOI: 10.1002/nme

(2.3) Check termination criteria for the subproblem. If not satisfied, go to (2.1). The termination criteria is defined as $|\Delta J| \leqslant \varepsilon$, where ε is a predefined small positive number.

Step 3: Update Lagrangian multiplier $\lambda = \lambda + \frac{1}{\gamma}(\int\int_D H(\Phi)\,\mathrm{d}\Omega - V_0)$

Step 4: Check termination condition. If not satisfied, go to *Step* 2. The termination criteria is defined as $|\Delta\lambda| \leqslant \delta$, 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 $\Phi_{\mathrm{f}}(x, \mathbf{c}(t)) = \sum_{i=1}^{N} c_i(t)\chi_i(x)$ to represent the shape. If pure parametric shape optimization is preferred, $\Phi_{\mathrm{p}}(x, \mathbf{b}(t))$ 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 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 rigorously 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 4.2–4.4 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 Appendix A.

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

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

where

$$R = -fu - \mathrm{div}(pun) + \frac{1}{2}E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u) - \lambda - \frac{1}{\gamma}\left(\int\int_D H(\Phi)\,\mathrm{d}\Omega - V_0\right) \tag{13}$$

DOI: 10.1002/nme

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

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

is a descent direction of Problem (11).

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

$$\left\{\frac{\mathrm{d}b_j(t)}{\mathrm{d}t}=\int_{\partial\Omega}\frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}(t))}{\mathrm{d}b_j}R\,\mathrm{d}\Gamma,\ \ j=1,\ldots,M\right\} \tag{15}$$

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 $\Phi(x,t)=\Phi(\Phi_{\mathrm{f}}^1,\ldots,\Phi_{\mathrm{f}}^{K_{\mathrm{f}}},\Phi_{\mathrm{p}}^1,\ldots,\Phi_{\mathrm{p}}^{K_{\mathrm{p}}})$. The boundary $\partial\Omega$ consists of $K_{\mathrm{f}}+K_{\mathrm{p}}$ pieces, $\Gamma_{\mathrm{f}}^k=\partial\Omega\cap\partial\Omega_{\mathrm{f}}^k,\ k=1,\ldots,K_{\mathrm{f}}$, and $\Gamma_{\mathrm{p}}^k=\partial\Omega\cap\partial\Omega_{\mathrm{p}}^k,\ k=1,\ldots,K_{\mathrm{p}}$. 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 (see Lemma A.1). This allows us to decouple the derivative of Φ into derivatives of individual primitives (see proof of Theorem 4.4 in Appendix A).

Theorem 4.4 (*General case*)
In Problem (11), if $\Phi(x,t)=\Phi(\Phi_{\mathrm{f}}^1,\ldots,\Phi_{\mathrm{f}}^{K_{\mathrm{f}}},\Phi_{\mathrm{p}}^1,\ldots,\Phi_{\mathrm{p}}^{K_{\mathrm{p}}})$, and almost every point of $\partial\Omega$ belongs to the boundary of exactly one primitive, then

$$\left\{\frac{\mathrm{d}c_i^k(t)}{\mathrm{d}t}=\int_{\Gamma_{\mathrm{f}}^k}\chi_i^k(x)R\,\mathrm{d}\Gamma,\ \ i=1,\ldots,N_k\right\},\quad k=1,\ldots,K_{\mathrm{f}} \tag{16}$$

and

$$\left\{\frac{\mathrm{d}b_j^k(t)}{\mathrm{d}t}=\int_{\Gamma_{\mathrm{p}}^k}\frac{\mathrm{d}\Phi_{\mathrm{p}}^k(x,\mathbf{b}^k(t))}{\mathrm{d}b_j^k}R\,\mathrm{d}\Gamma,\ \ j=1,\ldots,M_k\right\},\quad k=1,\ldots,K_{\mathrm{p}} \tag{17}$$

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' behaviour at the boundary. Through Theorem 4.4, we see that the theory of R-functions provides a convenient tool for decoupling the derivative of Φ into derivatives of each individual primitives at the boundary points. This implies that the computations of these sensitivities do not require explicitly constructing the composite implicit representation. Thus, the main utility of R-function in shape optimization is in the problem formulation and does not introduce any additional computational burden.

DOI: 10.1002/nme

The above sensitivity analysis is sufficient to support shape optimization procedure that combines parametric and free-form shapes with topological changes. However, it does not provide an efficient mechanism for introducing new holes (or voids in 3D) in the interior of the design domain. Consider the special case where only free-form shape is involved (Theorem 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 [10]. It also means that a direct implementation of Expression (14), while theoretically is well founded, does not allow direct nucleation of holes. When this is desired, it is convenient to use an alternative 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 criterion as the average value of function R over each individual B-spline's support (the effect of basis function $\chi_i(x)$ is negligible):

$$\frac{\mathrm{d}c_i(t)}{\mathrm{d}t} = \frac{\int\int_{\mathrm{support}(\chi_i)\bigcap\Omega} R\,\mathrm{d}\Omega}{\int\int_{\mathrm{support}(\chi_i)\bigcap\Omega} \mathrm{d}\Omega} \tag{18}$$

This is consistent with (14) in the sense that the signs of Expressions (14) and (18) are the same (thus the decent direction maintained) in the limit (as the grid size goes to infinity) for smooth integrand function (such as R) and smooth boundary $\partial\Omega$ (such as the one represented implicitly by the B-spline surfaces Φ_f). This modified version is well defined at all interior points and provides an updating criterion for every B-spline coefficient as well as those that have support intersecting the boundary. In this sense, the proposed strategy is an extension of the sensitivity-based approach implied by Expression (14).

For a general case where both free-form shapes and parametric shapes are involved, the derivative of Φ cannot be decoupled into derivatives of individual primitives at the interior points since Lemma A.1 is only valid at boundary points. To obtain an extended derivative as in Expression (18), we have two choices. One is to construct the composite implicit surface Φ, use it to calculate the boundary sensitivities as in Expression (A8) and to extend them to the entire domain. Alternatively, we can calculate the boundary sensitivities separately for free-form primitives Φ_f and parametric primitives Φ_p as in Expression (A13) and (A14) and only extend Expression (A13) to the interior of the domain, ignoring the properties of the composite Φ in the interior. It can be shown that both approaches will give legitimate descent directions for the objective function. We adopt the second approach in our implementation due to its simplicity.

4.3. Examples

We conclude this section, by illustrating the results of sensitivity analysis on two simple examples.

Example 1

Suppose we would like to incorporate a solid circular disk as an additive feature that is attached (*via* union operation) to a free-form design. The disk is parameterized by its position (x_c, y_c) and radius R_c, where the position (x_c, y_c) are design variables. In this case, the level set function Φ_p for the circular disk can be written as

$$\Phi_\mathrm{p}(x, t, x_c, y_c) = \frac{1}{2R_c}\left[R_c^2 - (x_1 - x_c)^2 - (x_2 - y_c)^2\right]$$

DOI: 10.1002/nme

where $x = (x_1, x_2)$ is spatial variable. The level set function for the shape is

$$\Phi(x, t) = \Phi_{\mathrm{f}}(x, t) \vee_0 \Phi_{\mathrm{p}}(x, t, x_c, y_c)$$

While the time derivative of Φ_{p} is

$$\frac{\mathrm{d}\Phi_{\mathrm{p}}(x, t, x_c, y_c)}{\mathrm{d}t} = \frac{1}{R_c}\left[(x_1 - x_c)\frac{\mathrm{d}x_c}{\mathrm{d}t} + (x_2 - y_c)\frac{\mathrm{d}y_c}{\mathrm{d}t}\right]$$

we have

$$\frac{\mathrm{d}J(u, \Phi)}{\mathrm{d}t} = -\sum_{i=1}^{N}\int_{\Gamma_{\mathrm{f}}}\frac{\mathrm{d}c_i(t)}{\mathrm{d}t}\chi_i(x)R\,\mathrm{d}\Gamma - \int_{\Gamma_{\mathrm{p}}}\frac{1}{R_c}\left[(x_1 - x_c)\frac{\mathrm{d}x_c}{\mathrm{d}t} + (x_2 - y_c)\frac{\mathrm{d}y_c}{\mathrm{d}t}\right]R\,\mathrm{d}\Gamma$$

Thus, a descent direction of Problem (11) can be constructed by the following:

$$\begin{bmatrix} \dfrac{\mathrm{d}c_i}{\mathrm{d}t} \\ \dfrac{\mathrm{d}x_c}{\mathrm{d}t} \\ \dfrac{\mathrm{d}y_c}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \displaystyle\int_{\Gamma_{\mathrm{f}}}\chi_i(x)R\,\mathrm{d}\Gamma \\ \displaystyle\int_{\Gamma_{\mathrm{p}}}(x_1 - x_c)R\,\mathrm{d}\Gamma \\ \displaystyle\int_{\Gamma_{\mathrm{p}}}(y_1 - y_c)R\,\mathrm{d}\Gamma \end{bmatrix}$$

$\Gamma_{\mathrm{p}} = \{x \in \partial\Omega | R_c^2 - (x_1 - x_c)^2 - (x_2 - y_c)^2 = 0\}$ is the set of boundary points of the shape where $\Phi_{\mathrm{p}} = 0$.

Example 2

A rectangular slot is parameterized by its position (x_r, y_r) and size $(2A, 2B)$, where the position (x_r, y_r) are design parameters and $(2A, 2B)$ remain constant. This parameterized rectangular slot is incorporated in a free-form shape *via* intersection operation, as $\Phi(x, t) = \Phi_{\mathrm{f}}(x, t) \wedge_0 \Phi_{\mathrm{p}}(x, t, x_r, y_r)$. The primitive parametric function

$$\Phi_{\mathrm{p}}(x, t, x_r, y_r) = \frac{(x_1 - x_r)^2 - A^2}{2A} \wedge_0 \frac{(x_2 - y_r)^2 - B^2}{2B}$$

implicitly represents the exterior of the rectangle. Thus, we have

$$\Phi(x, t) = \Phi_{\mathrm{f}}(x, t) \wedge_0 \left(\frac{(x_1 - x_r)^2 - A^2}{2A} \wedge_0 \frac{(x_2 - y_r)^2 - B^2}{2B}\right)$$

The time derivative of the objective function in Problem (11) is

$$\frac{\mathrm{d}J(u, \Phi)}{\mathrm{d}t} = -\sum_{i=1}^{N}\int_{\Gamma_{\mathrm{f}}}\frac{\mathrm{d}c_i(t)}{\mathrm{d}t}\chi_i(x)R\,\mathrm{d}\Gamma + \int_{\Gamma_{\mathrm{p}}^1}\frac{x_1 - x_r}{A}R\,\mathrm{d}\Gamma + \int_{\Gamma_{\mathrm{p}}^2}\frac{x_2 - y_r}{B}R\,\mathrm{d}\Gamma$$

DOI: 10.1002/nme

Therefore, a decent direction of Problem (11) can be constructed by

$$\begin{bmatrix} \dfrac{\mathrm{d}c_i}{\mathrm{d}t} \\ \dfrac{\mathrm{d}x_r}{\mathrm{d}t} \\ \dfrac{\mathrm{d}y_r}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \displaystyle\int_{\Gamma_\mathrm{f}} \chi_i(x) R \,\mathrm{d}\Gamma \\ \displaystyle\int_{\Gamma_\mathrm{p}^1} -(x_1 - x_r) R \,\mathrm{d}\Gamma \\ \displaystyle\int_{\Gamma_\mathrm{p}^2} -(x_2 - y_r) R \,\mathrm{d}\Gamma \end{bmatrix}$$

where $\Gamma_\mathrm{p}^1 = \{x \in \partial\Omega | (x_1 - x_r)^2 = A^2\}$ and $\Gamma_\mathrm{p}^2 = \{x \in \partial\Omega | (x_2 - y_r)^2 = B^2\}$.

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 numerical results for a short cantilever beam design problem (see Figure 7) 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 $v = 0.3$, 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.

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, thus only the coefficients of those B-splines whose support intersect the boundary are updated; the rest of the coefficients need not be changed. The resulting optimization process allows boundary deformations and no new holes can appear inside the domain. This is consistent with traditional shape optimization where only boundary variations can be handled since the shape sensitivity analysis is based on small boundary perturbations. Figure 8 shows the results at different stages of the optimization process. (The colour map shows the distribution of strain energy in this example and all examples that follow.)

Figure 7. Definition of the minimum compliance problem for a short cantilever beam.

DOI: 10.1002/nme

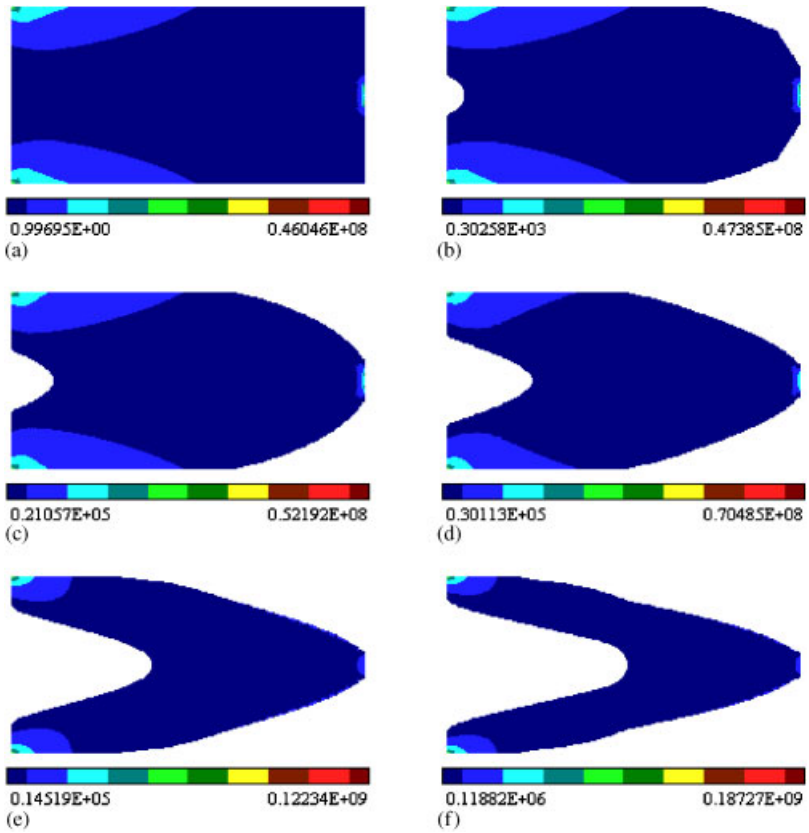

Figure 8. Strain energy distribution of the shapes at different iterations from the free-form shape optimization process without nucleation of holes. Only the coefficients of those B-spline basis functions that have support intersecting the boundary are updated. Grid size $= 100 \times 50$: (a) initial design; (b) iteration 40; (c) iteration 60; (d) iteration 90; (e) iteration 120; and (f) iteration 141.

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. Although the sensitivity analysis only involves the coefficients of those B-spline basis functions whose support intersects the boundary, we extend the updating criterion to all other coefficients as in Expression (18). The small changes of those coefficients far away from the boundary may not affect the shape itself, but we still choose to update them and hope that eventually some of them may cause nucleation of new holes. The reasoning behind this approach is similar to the one used in evolutionary methods: if the material is not efficiently used at some point ($R<0$, which means strain energy is below some threshold), we may create a hole, if the material is over stressed at some point ($R>0$, which means strain energy is above some threshold), we may add more material. A similar approach was adopted in [8]. Figure 9 shows the results at different stages of the optimization process.

Case 3: *Parametric*, *with or without topological changes*. For pure parametric design, the design variables are geometric parameters $\{b_1, b_2, \ldots, b_M\}$. Figure 10 shows the optimal shape

DOI: 10.1002/nme

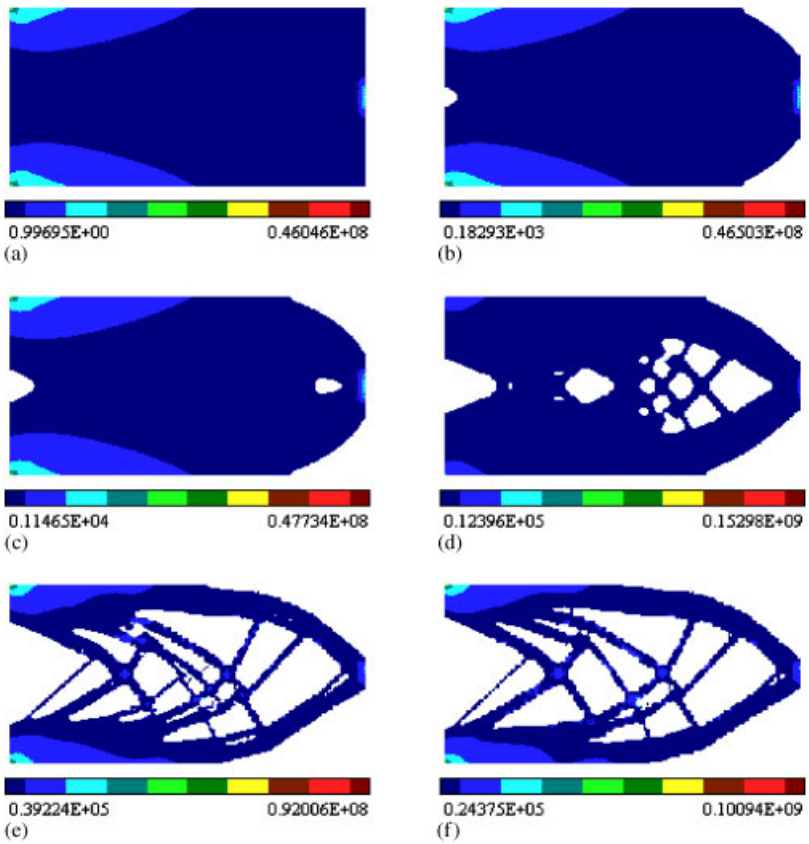

Figure 9. Strain energy distribution of the shapes at different iterations from the free-form shape optimization process with nucleation of holes allowed. Every B-spline coefficient is updated during the optimization process. Grid size $= 100 \times 50$: (a) initial design; (b) iteration 20; (c) iteration 28; (d) iteration 40; (e) iteration 80; and (f) iteration 111.

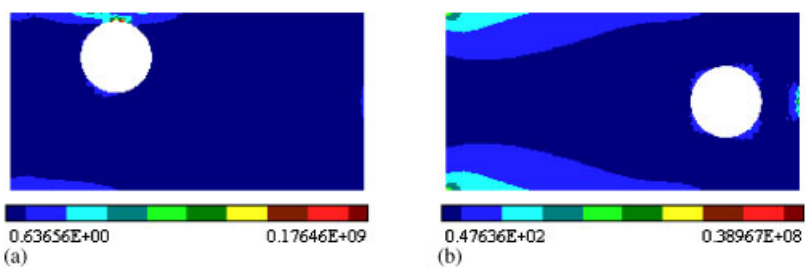

Figure 10. Strain energy distribution of the initial and optimal shapes of a rectangle with a circular hole. The design variables are the position of the hole. Grid size $= 50 \times 25$: (a) initial design; and (b) result at iteration 9.

DOI: 10.1002/nme

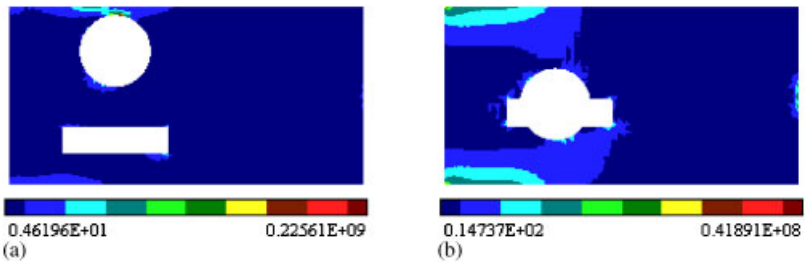

Figure 11. Strain energy distribution of the initial and optimal shapes of a rectangle with a circular and rectangular slot. The design variables are the positions of the hole and the slot. Grid size $= 50 \times 25$: (a) initial design; and (b) result at iteration 21.

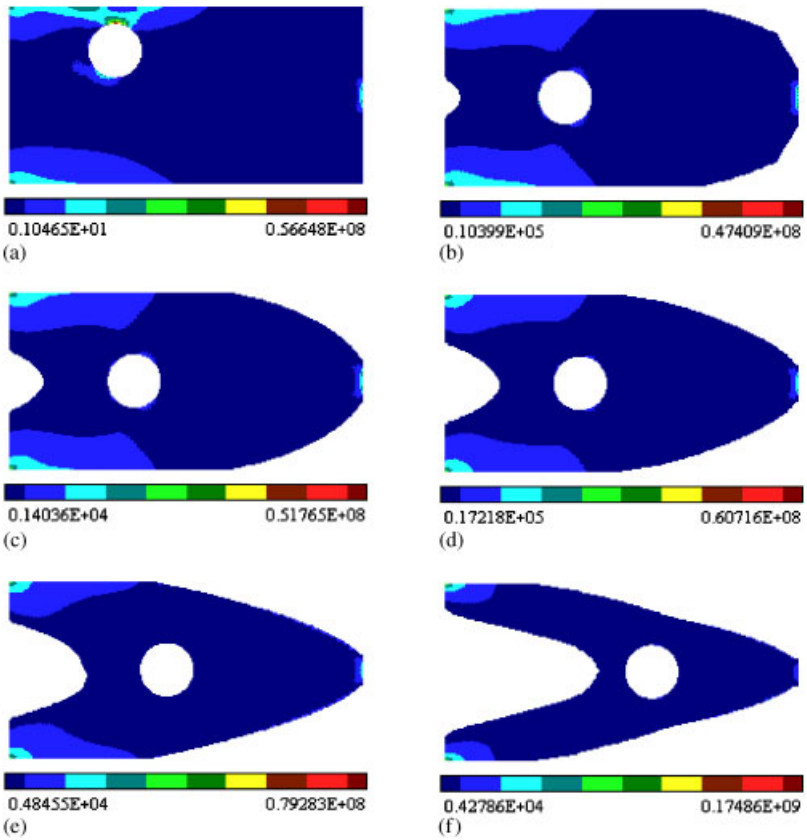

Figure 12. Strain energy distribution at different iterations during the optimization process for free-form shape with a circular hole. Grid size $= 100 \times 50$: (a) initial design; (b) iteration 40; (c) iteration 60; (d) iteration 80; (e) iteration 100; and (f) iteration 135.

of a rectangle with a circular hole, with the position of the hole as the only design parameter. Figure 11 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

DOI: 10.1002/nme

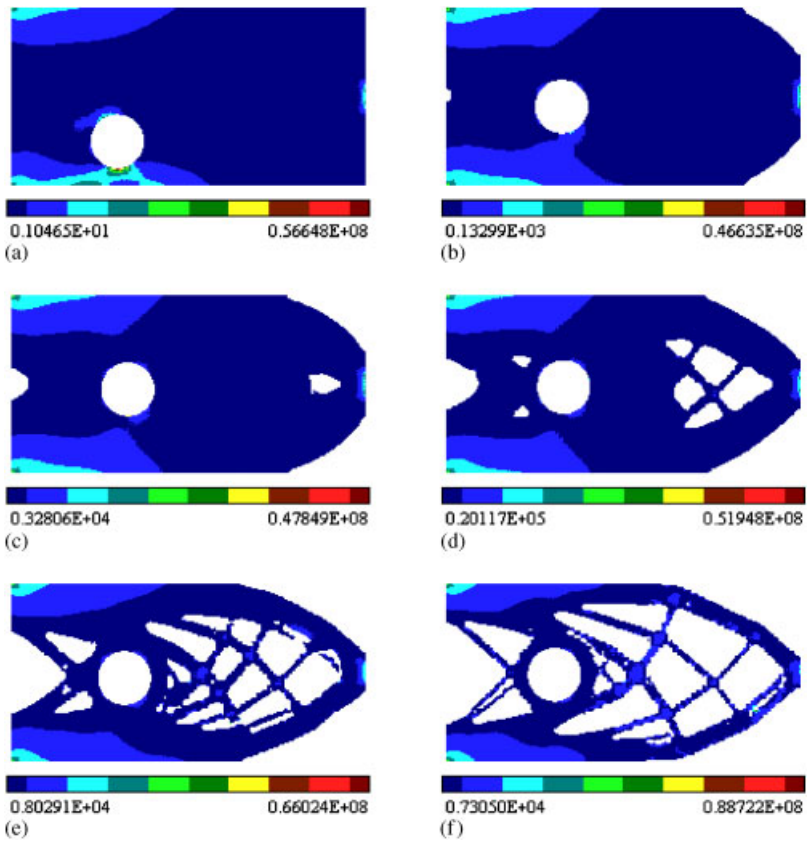

Figure 13. Strain energy distribution at different iterations during the optimization process for free-form shape with a circular hole. Nucleation of holes inside the domain is allowed. Grid size $= 100 \times 50$: (a) initial design; (b) iteration 20; (c) iteration 30; (d) iteration 40; (e) iteration 60; and (f) iteration 90.

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. The high-level geometric parametrization allows additional controls, such as preventing certain topological events, by adding some geometric constraints. For example, in Figure 11, if we do not want the hole and the slot to intersect, we could impose an additional geometric constraint such as minimum distance between two holes. We note that the final designs for these two examples are both locally optimal. (Without volume constraint or additional geometric constraints, the hole(s) should move out the domain to give the minimum objective.)

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 and geometric parameters. Figure 12 shows the example of a moving circular hole inside the domain where the position of the hole needs to be optimized.

DOI: 10.1002/nme

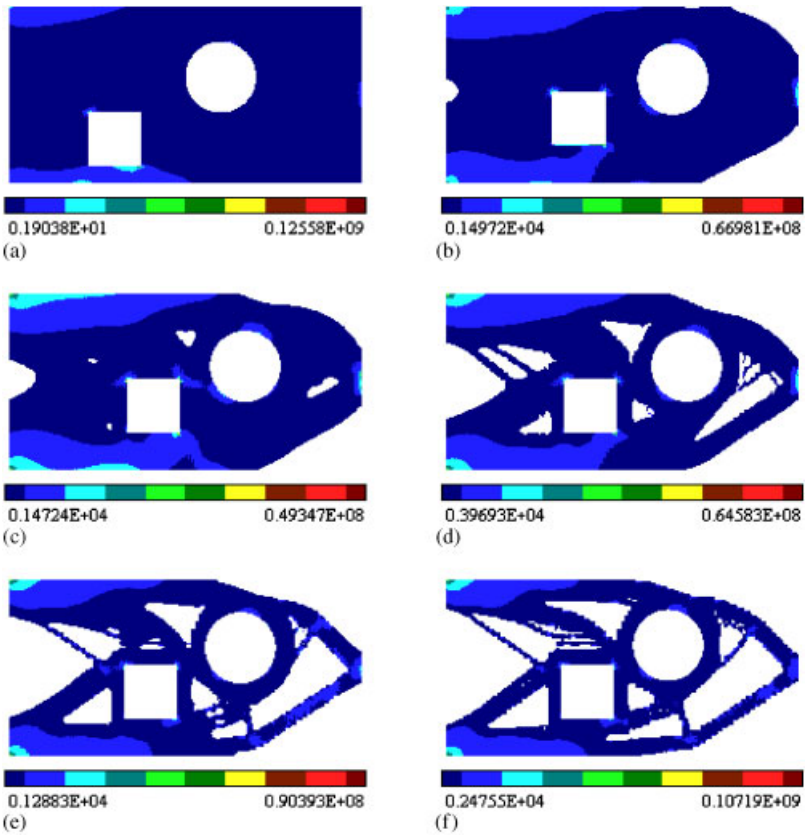

Figure 14. Strain energy distribution at different iterations during the optimization process for free-form shape with a circular hole and a rectangular slot. Nucleation of holes is allowed. Grid size $= 100 \times 50$: (a) initial design; (b) iteration 30; (c) iteration 40; (d) iteration 60; (e) iteration 80; and (f) iteration 134.

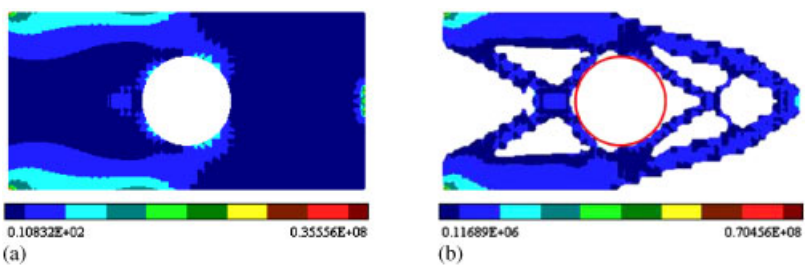

Figure 15. Strain energy distribution of the initial and optimal free-form shape with a circular hole as a fixed obstacle. Grid size $= 50 \times 25$: (a) initial design; and (b) optimal shape.

DOI: 10.1002/nme

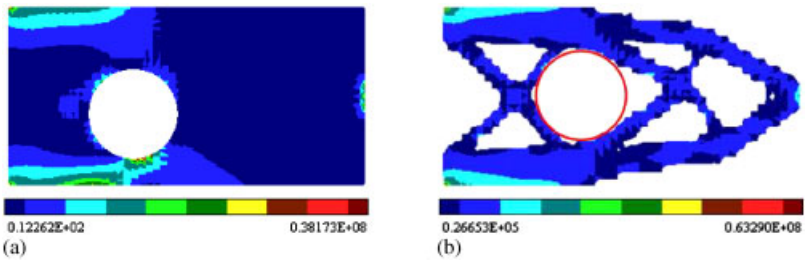

Figure 16. Strain energy distribution of the initial and optimal free-form shape with a circular hole as a moving obstacle. Grid size $= 50 \times 25$: (a) initial design; and (b) optimal shape.

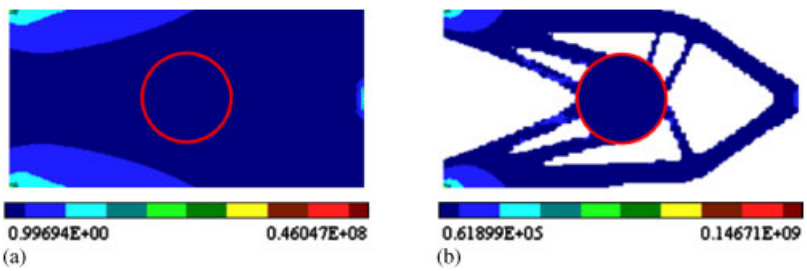

Figure 17. Strain energy distribution of the initial and optimal free-form shape with a disk as an attachment. Grid size $= 100 \times 50$: (a) initial design; and (b) optimal shape.

Case 5: *Free form and parametric*, *with topological changes*. In this case, all B-spline coefficients are updated to allow nucleation of holes. Geometric parameters are updated to optimize the parametric shapes. As the shape evolves, free form and parametric boundaries may intersect, for example, free-form holes may appear and intersect with the parametric shapes. In some situations this may not be desirable. Figures 13 and 14 show the results at different stages of the optimization process for two different initial designs. In these examples, 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 [24] and force the B-spline coefficients inside the tolerance zone to be positive.

Case 6: Parametric features as obstacles or attachments. Figure 15 shows the optimal shape with a circular hole as a fixed obstacle. Figure 16 shows the result from the same problem except that the obstacle can move around. Figure 17 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.

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

DOI: 10.1002/nme

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. Both alternatives are computationally expensive, and could lead to additional errors and instabilities.

In this section we briefly describe how the proposed approach to shape optimization is implemented in a meshfree environment developed by the authors [14] and used earlier to solve the problems with shape deformations and moving boundary conditions [13].

6.1. Meshfree approach with distance fields

Since we represent the boundary of a geometric domain by the zero level set of an evolving scalar function, it is natural to use an engineering analysis method that can work with the same geometric representation. A meshfree method with approximated distances described in [52] appears to be well suited for the task. This method is based on the original idea by Kantorovich and Krylov [53] for solving simple Dirichlet problems, but was fully developed by Rvachev and his students for general boundary conditions and problems [29, 49]. To paraphrase Rvachev, a physical field can be represented by a generalized Taylor series by powers of an approximate distance field to the boundary [29, 54].[§] 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 [54]. The method is essentially meshfree, though a background mesh may be used for integration and visualization purposes. The required distance fields may be constructed from a wide range of sources and by variety of methods, including R-functions [25, 32], special purpose constructions [55], as well as sampling and fitting [52]. A restricted implementation of the method with WEB-Splines is described in [55], and a complete programming environment supporting construction, differentiation, and integration of all required functions at run time is described in [14].

The implementation of the general meshfree solution method relies on automatic differentiation techniques [56, 57] that propagates derivatives from independent to dependent variables.[¶] Geometrically adaptive integration [14] uses hierarchical space decomposition together with marching cubes algorithms in order to integrate over non-meshed geometric domain. These algorithms are well suited for integration over geometric domains whose boundary is described by a level set function Φ. Hierarchical space decompositions (quadtree in two dimension or octree in three dimension) help to resolve ambiguous intersection cases of the integration cell with the domain's boundary. When ambiguity is eliminated by hierarchical cell subdivision, an appropriate coordinate system is imposed in the integration cell according to marching cubes classification. Then

[§] Rvachev does not use distance fields directly but employs the so-called 'normalized functions' that are constructed using theory of R-functions [29]. With that terminology, a level set function is 0th level approximation of a distance from any particular level set.

[¶] Fast Forward Automatic Differentiation Library (FFADLib) [57] is publicly available for download at our web site http://sal-cnc.me.wisc.edu.

DOI: 10.1002/nme

integration points are allocated in this co-ordinate system using Gauss integration rules. Additional details can be found in [14].

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. Basis functions $\{\xi_j\}_{j=1}^{k}$ can be chosen from B-splines, polynomials, radial basis functions or even finite elements. 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 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 [58]. 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 co-ordinates and integration over non-meshed geometric domain and its boundary that is represented by a level set function. Use of B-splines as basis functions results in matrices that possess block-diagonal sparse structure. Algebraic systems with such matrices can be efficiently solved by a conjugate gradient algorithm [59]. 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 $\{\mathrm{d}c_i(t)/\mathrm{d}t\}$ can be directly calculated by integrating R over each B-spline support as in Expression (18). Note that the derivatives $\{\mathrm{d}b_j(t)/\mathrm{d}t\}$ require boundary integration. The integrand in Expression (17) is available since the derivatives of $\{\mathrm{d}\Phi_\mathrm{p}/\mathrm{d}b_j\}$ 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 11 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. After we compute the derivatives, updating of parameters and Lagrangian multiplier λ is straightforward as described in Section 4.1.

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 18 shows three different shapes and the corresponding B-spline

DOI: 10.1002/nme

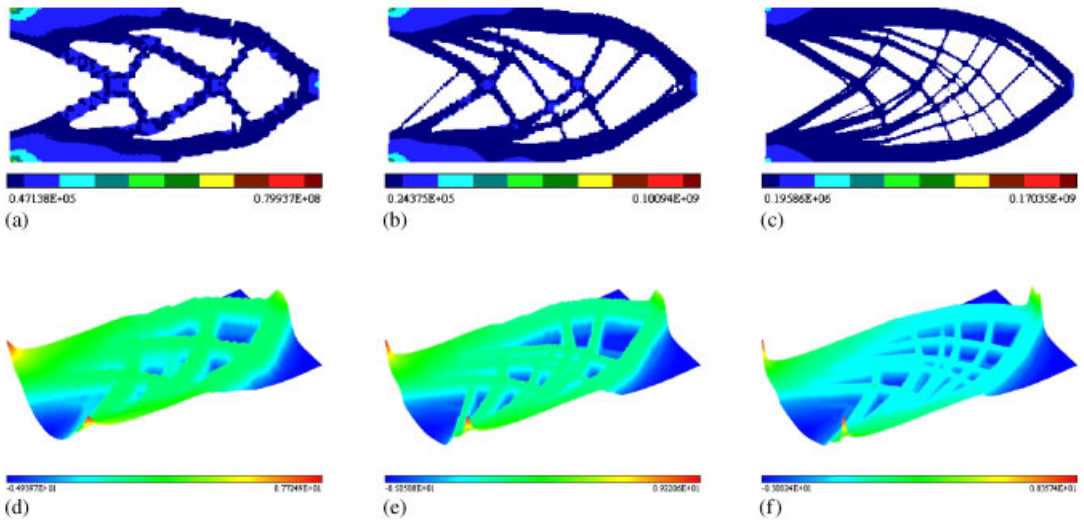

Figure 18. Strain energy distribution and the B-spline surfaces of the optimal shapes from different grid sizes in the free-form shape optimization with nucleation of holes: result from (a) 50×25 grid; (b) 100×50 grid; (c) 200×100 grid; (d) 50×25 grid; (e) 100×50 grid; and (f) 200×100 grid.

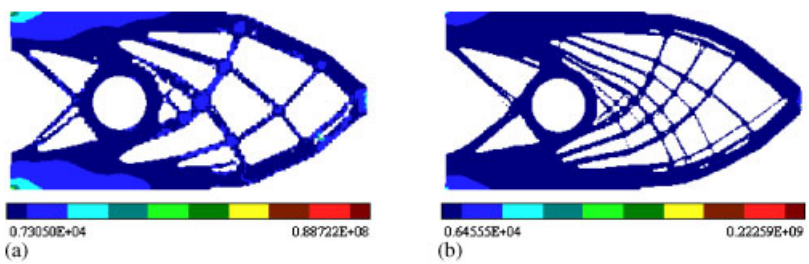

Figure 19. Strain energy distribution of the optimal shapes from different grid sizes in the free-form shape optimization with nucleation of holes. A circular hole is incorporated: (a) result from 100×50 grid; and (b) result from 200×100 grid.

surfaces resulting from the same optimization problem that also produced the shape in Figure 9, but with different grids of B-splines: 50×25, 100×50 and 200×100. The effects of grid dependence can be also observed in Figure 19 for the example from Figure 13. Although the final shapes are very different, the values of the objective function are very close to each other. Figure 20 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 [24, 60]. 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

DOI: 10.1002/nme

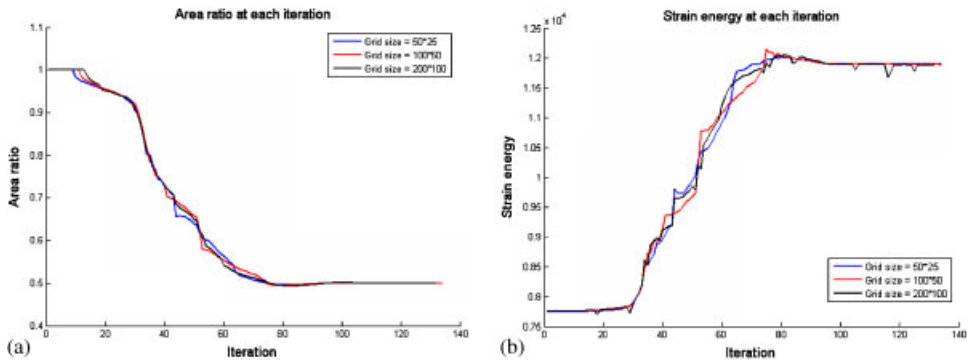

Figure 20. The area ratio and strain energy during optimization process for different grid sizes: (a) the area ratio during optimization process; and (b) the strain energy during optimization process.

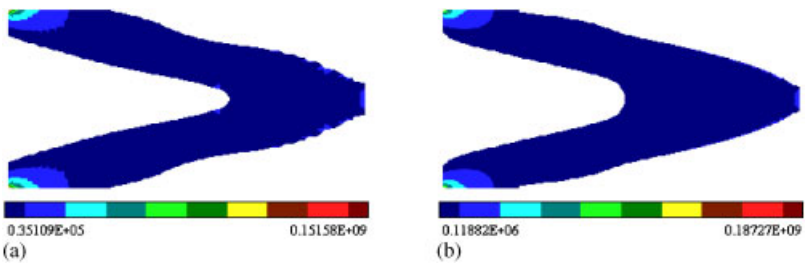

Figure 21. Strain energy distribution of the optimal shapes from different grid sizes in the free-form shape optimization without nucleation of holes: (a) result from 50×25 grid; and (b) result from 100×50 grid.

holes. If we disallow introduction of new holes, the grid dependence disappears (Figure 21 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. We conjecture that, in the limit, our method produces results similar to those obtained by the homogenization method. To make the optimization problem well posed, regularization and filtering techniques (such as perimeter control or gradient restriction) have been widely used within homogenization method [24] and should be equally applicable in our approach to shape optimization.

7. CONCLUSIONS

We proposed a new method for shape optimization that combines and subsumes free-form and parametric shape optimization approaches. A key element of the new approach is an implicit level set representation that relies on *R*-functions to compose and combine free-form and parametric implicit representations. The resulting space of shapes is fully parameterized by the B-spline coefficients (for the free-form boundaries) and geometric dimensional parameters, without restricting

DOI: 10.1002/nme

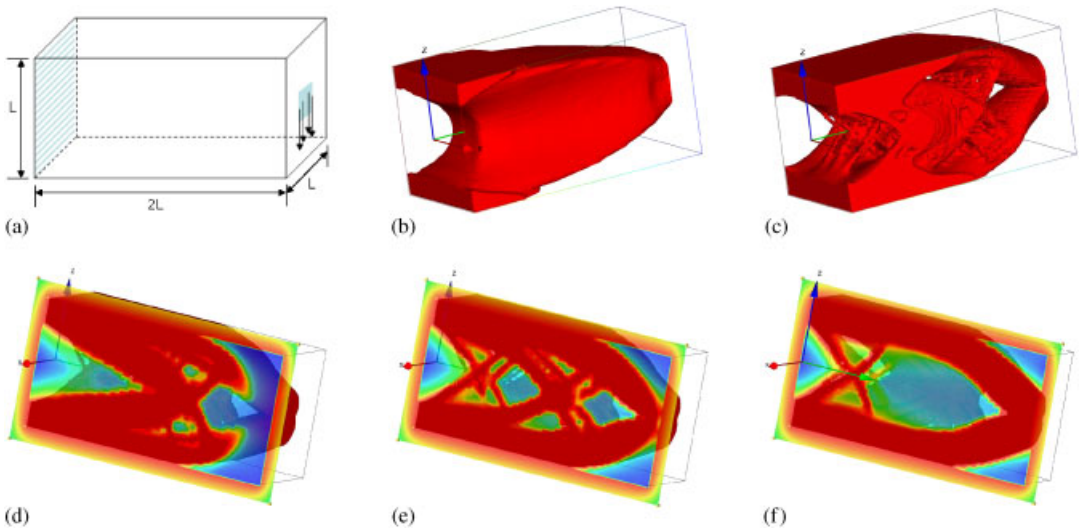

Figure 22. The optimal shape of a 3D beam with $L = 1$. The free-form shape optimization is applied with or without nucleation of holes allowed. The grid size is $40 \times 80 \times 40$: (a) the problem definition; (b) the optimal shape without nucleation of holes; (c) the optimal shape with nucleation of holes allowed; (d) the internal structure of (c) at $x = 0.25$; (e) the internal structure of (c) at $x = 0.15$; and (f) the internal structure of (c) at $x = 0$.

the parameterization to any particular topology. This parametrization transforms the difficult shape and topology optimization problems into a relatively straightforward parameterized problem to which many gradient-based 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 [6, 7]. This technique can also be implemented in our approach if we apply level set equation $\mathrm{d}\Phi/\mathrm{d}t = -|\nabla\Phi| \cdot V_n$ in Expression (12). Use of shape functionals to represent level set surfaces for shape optimization as proposed in [8, 9] 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.

Extending the proposed method to three-dimensional problems is straightforward, provided that the computational utilities described in Section 6 are available. Figure 22 shows the shape optimization results for a 3D cantilever beam design problem, defined by the boundary conditions in (a). The left face of the beam is fixed and a loading is applied on a small area of the right face. The volume constraint is set to be one half of the original solid. Figure 22(b) shows the result of a free-form shape optimization without topological changes, and Figure 22(c) shows the result

DOI: 10.1002/nme

of the same optimization procedure but with nucleation of holes allowed. The beam shapes are represented implicitly using trilinear B-splines on a $40 \times 80 \times 40$ uniform grid, with zero level set polygonized for visualization purposes. It is instructive to compare these experiments to the analogous two-dimensional results shown in Figures 8 and 9, respectively. Figure 22(d)–(f) shows several sections of the beam shape in Figure 22(c).

We have not considered material-based approaches to topology optimization problem, such as the homogenization method and the SIMP (solid isotropic microstructure with penalty) method, which use material models to seek an optimal design. At present, it is not clear if there is a rigorous connection between our approach and material-based shape optimization techniques. If properties (for example, height) of the level set function can be correlated with the material properties (for example, material density), then the material-based shape optimization and geometry-based shape optimization can be treated in a single framework. Analogies between material models and level set formulations have also been observed in [9], which explains a dual relationship between the level set approach therein and the SIMP method.

Application of the proposed method to shape optimization problems with additional geometric constraints and more general objective functions is also of interest. Other promising directions for future work include investigation of shape optimization problems with design-dependent load conditions, topology preserving shape optimization, sizing control during the optimization process (for example, suppressing thin structures), and multi-resolution shape optimization.

APPENDIX A

A.1 Proof of Lemma 4.1

Proof

Take the derivative of $J_0(u, \Phi)$, $a(u, v, \Phi)$, and $l(v, \Phi)$ with respect to t and use the fact that $\mathrm{d}H(\Phi)/d\Phi = \delta(\Phi)$, δ is the Dirac delta function, we have

$$\begin{aligned}\frac{\mathrm{d}J_0(u,\Phi)}{\mathrm{d}t} &= \int\!\!\int_D E_{ijkl}\varepsilon_{ij}(\dot{u})\varepsilon_{kl}(u)H(\Phi)\,\mathrm{d}\Omega + \int\!\!\int_D \frac{1}{2}E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u)\delta(\Phi)\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Omega \\ &= \int\!\!\int_D E_{ijkl}\varepsilon_{ij}(\dot{u})\varepsilon_{kl}(u)H(\Phi)\,\mathrm{d}\Omega + \int_{\partial\Omega}\frac{1}{2}E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u)\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Gamma \qquad \text{(A1)}\end{aligned}$$

$$\begin{aligned}\frac{\mathrm{d}a(u,v,\Phi)}{\mathrm{d}t} &= \int\!\!\int_D E_{ijkl}\varepsilon_{ij}(\dot{u})\varepsilon_{kl}(v)H(\Phi)\,\mathrm{d}\Omega + \int\!\!\int_D E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(\dot{v})H(\Phi)\,\mathrm{d}\Omega \qquad \text{(A2)} \\ &\quad + \int\!\!\int_D E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(v)\delta(\Phi)\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Omega\end{aligned}$$

$$\begin{aligned}\frac{\mathrm{d}l(v,\Phi)}{\mathrm{d}t} &= \int\!\!\int_D [fv + \mathrm{div}(pvn)]\,\delta(\Phi)\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Omega \\ &\quad + \int\!\!\int_D [f\dot{v} + \mathrm{div}(p\dot{v}n)]\,H(\Phi)\,\mathrm{d}\Omega \qquad \text{(A3)}\end{aligned}$$

DOI: 10.1002/nme

Since $\dot{v} \in U$, $\iint_D E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(\dot{v})H(\Phi)\,\mathrm{d}\Omega = \iint_D [f\dot{v} + \mathrm{div}(p\dot{v}n)]H(\Phi)\,\mathrm{d}\Omega$. From Expression (A3) and (A3) we get

$$\iint_D E_{ijkl}\varepsilon_{ij}(\dot{u})\varepsilon_{kl}(v)H(\Phi)\,\mathrm{d}\Omega = \iint_D \left[fv + \mathrm{div}(pvn) - E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(v)\right]\delta(\Phi)\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Gamma$$

$$= \int_{\partial\Omega} \left[fv + \mathrm{div}(pvn) - E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(v)\right]\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Gamma \tag{A4}$$

Since the minimum compliance problem is self-adjoint,

$$\iint_D E_{ijkl}\varepsilon_{ij}(\dot{u})\varepsilon_{kl}(u)H(\Phi)\,\mathrm{d}\Omega = \int_{\partial\Omega} \left[fu + \mathrm{div}(pun) - E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u)\right]\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Gamma \tag{A5}$$

Substitute Expression (A5) in (A1), we have

$$\frac{\mathrm{d}J_0(u,\Phi)}{\mathrm{d}t} = \int_{\partial\Omega}\left[fu + \mathrm{div}(pun) - \frac{1}{2}E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u)\right]\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Gamma \tag{A6}$$

We also have that

$$\frac{\mathrm{d}(\iint_D H(\Phi)\,\mathrm{d}\Omega - V_0)}{\mathrm{d}t} = \int_{\partial\Omega}\frac{\mathrm{d}\Phi}{\mathrm{d}t}\,\mathrm{d}\Gamma \tag{A7}$$

It is easy to see that

$$\frac{\mathrm{d}J(u,\Phi)}{\mathrm{d}t} = \int_{\partial\Omega}\frac{\mathrm{d}\Phi}{\mathrm{d}t}\left[fu + \mathrm{div}(pun) - \frac{1}{2}E_{ijkl}\varepsilon_{ij}(u)\varepsilon_{kl}(u) + \lambda + \frac{1}{\gamma}\left(\iint_D H(\Phi)\,\mathrm{d}\Omega - V_0\right)\right]\mathrm{d}\Gamma$$

$$= -\int_{\partial\Omega}\frac{\mathrm{d}\Phi}{\mathrm{d}t}R\,\mathrm{d}\Gamma \tag{A8}$$

A.2. Proof of Theorem 4.2

Proof
Since

$$\frac{\mathrm{d}\Phi(x,t)}{\mathrm{d}t} = \frac{\mathrm{d}\Phi_{\mathrm{f}}(x,t)}{\mathrm{d}t} = \sum_{i=1}^{N}\frac{\mathrm{d}c_i(t)}{\mathrm{d}t}\chi_i(x) \tag{A9}$$

by substituting Expression (A9) into (12) we get

$$\frac{\mathrm{d}J(u,\Phi)}{\mathrm{d}t} = -\sum_{i=1}^{N}\int_{\partial\Omega}\frac{\mathrm{d}c_i(t)}{\mathrm{d}t}\chi_i(x)R\,\mathrm{d}\Gamma \tag{A10}$$

If we define $\{\mathrm{d}c_i(t)/\mathrm{d}t\}$ as in Expression (14), then we have

$$\frac{\mathrm{d}J}{\mathrm{d}t} = -\sum_{i=1}^{N}\left(\int_{\partial\Omega}\chi_i(x)R\,\mathrm{d}\Gamma\right)^2 \leqslant 0$$

DOI: 10.1002/nme

since $\chi_i(x) \geqslant 0$, which proves that $\{\mathrm{d}c_i(t)/\mathrm{d}t = \int_{\partial\Omega} \chi_i(x) R \,\mathrm{d}\Gamma,\ i = 1, \ldots N\}$ is a descent direction of Problem (11). □

A.3. Proof of Theorem 4.3

Proof
Since

$$\frac{\mathrm{d}\Phi(x,t)}{\mathrm{d}t} = \frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}(t))}{\mathrm{d}t} = \sum_{j=1}^{M} \frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}(t))}{\mathrm{d}b_j} \frac{\mathrm{d}b_j(t)}{\mathrm{d}t} \tag{A11}$$

by substituting Expression (A11) into (12) we get

$$\frac{\mathrm{d}J(u,\Phi)}{\mathrm{d}t} = -\sum_{j=1}^{M} \int_{\partial\Omega} \frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}(t))}{\mathrm{d}b_j} \frac{\mathrm{d}b_j(t)}{\mathrm{d}t} R \,\mathrm{d}\Gamma \tag{A12}$$

If we define $\{\mathrm{d}b_j(t)/\mathrm{d}t\}$ as in Expression (15), then we have

$$\frac{\mathrm{d}J}{\mathrm{d}t} = -\sum_{j=1}^{M} \left(\int_{\partial\Omega} \frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}(t))}{\mathrm{d}b_j} R \,\mathrm{d}\Gamma \right)^2 \leqslant 0$$

which proves that

$$\left\{ \frac{\mathrm{d}b_j(t)}{\mathrm{d}t} = \int_{\partial\Omega} \frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}(t))}{\mathrm{d}b_j} R \,\mathrm{d}\Gamma,\ \ j = 1, \ldots M \right\}$$

is a descent direction of Problem (11). □

A.4. Proof of Theorem 4.4

The following Lemma is a basic result from the theory of R-functions. For proof, refer to [29].

Lemma A.1
Suppose $\Phi(\Phi^1, \ldots, \Phi^K)$ is an R-functions of $\Phi^1, \ldots, \Phi^K$. If $\Phi^i(x_0) = 0$ and $\Phi^j(x_0) \neq 0$ for $\forall j \neq i$, $j = 1, \ldots, K$, then $D(\Phi)|_{x_0} = D(\Phi^i)|_{x_0}$.

Proof
From Lemma A.1 and the fact that isolated points do not affect boundary integration, we have

$$\begin{aligned} \int_{\Gamma_{\mathrm{f}}^k} \frac{\mathrm{d}\Phi(x,t)}{\mathrm{d}t} R \,\mathrm{d}\Gamma &= \int_{\Gamma_{\mathrm{f}}^k} \frac{\mathrm{d}\Phi_{\mathrm{f}}^k(x,\mathbf{c}^k(t))}{\mathrm{d}t} R \,\mathrm{d}\Gamma \\ &= \sum_{i=1}^{N_k} \int_{\Gamma_{\mathrm{f}}^k} \frac{\mathrm{d}c_i^k(t)}{\mathrm{d}t} \chi_i^k(x) R \,\mathrm{d}\Gamma, \quad k = 1, \ldots, K_{\mathrm{f}} \end{aligned} \tag{A13}$$

DOI: 10.1002/nme

and

$$\int_{\Gamma_{\mathrm{p}}^k} \frac{\mathrm{d}\Phi(x,t)}{\mathrm{d}t} R\,\mathrm{d}\Gamma = \int_{\Gamma_{\mathrm{p}}^k} \frac{\mathrm{d}\Phi_{\mathrm{p}}^k(x,\mathbf{b}^k(t))}{\mathrm{d}t} R\,\mathrm{d}\Gamma = \sum_{j=1}^{M_k} \int_{\Gamma_{\mathrm{p}}^k} \frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}^k(t))}{\mathrm{d}b_j^k} \frac{\mathrm{d}b_j^k(t)}{\mathrm{d}t} R\,\mathrm{d}\Gamma$$
$$k = 1, \ldots, K_{\mathrm{p}} \tag{A14}$$

so

$$\begin{aligned}\frac{\mathrm{d}J(u,\Phi)}{\mathrm{d}t} &= -\left(\sum_{k=1}^{K_{\mathrm{f}}} \int_{\Gamma_{\mathrm{f}}^k} + \sum_{k=1}^{K_{\mathrm{p}}} \int_{\Gamma_{\mathrm{p}}^k}\right) \frac{\mathrm{d}\Phi(x,t)}{\mathrm{d}t} R\,\mathrm{d}\Gamma \\ &= -\sum_{k=1}^{K_{\mathrm{f}}} \sum_{i=1}^{N_k} \int_{\Gamma_{\mathrm{f}}^k} \frac{\mathrm{d}c_i^k(t)}{\mathrm{d}t} \chi_i^k(x) R\,\mathrm{d}\Gamma - \sum_{k=1}^{K_{\mathrm{p}}} \sum_{j=1}^{M_k} \int_{\Gamma_{\mathrm{p}}^k} \frac{\mathrm{d}\Phi_{\mathrm{p}}(x,\mathbf{b}^k(t))}{\mathrm{d}b_j^k} \frac{\mathrm{d}b_j^k(t)}{\mathrm{d}t} R\,\mathrm{d}\Gamma \end{aligned} \tag{A15}$$

If we define $\{\mathrm{d}c_i^k(t)/\mathrm{d}t\}$ as in Expression (16) and $\{\mathrm{d}b_j^k(t)/\mathrm{d}t\}$ as in Expression (17) then

$$\frac{\mathrm{d}J(u,\Phi)}{\mathrm{d}t} = -\sum_{k=1}^{K_{\mathrm{f}}} \sum_{i=1}^{N_k} \left(\int_{\Gamma_{\mathrm{f}}^k} \chi_i^k(x) R\,\mathrm{d}\Gamma\right)^2 - \sum_{k=1}^{K_{\mathrm{p}}} \sum_{j=1}^{M_k} \left(\int_{\Gamma_{\mathrm{p}}^k} \frac{\mathrm{d}\Phi_{\mathrm{p}}^k(x,\mathbf{b}^k(t))}{\mathrm{d}b_j^k} R\,\mathrm{d}\Gamma\right)^2 \leqslant 0 \qquad \square$$

ACKNOWLEDGEMENTS

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

REFERENCES

1. Shapiro V, Vossler DL. What is a parametric family of solids?. *Third ACM Symposium on Solid Modeling and Applications*, Salt Lake City, Utah, May 1995.
2. Raghothama S, Shapiro V. Consistent updates in dual representation systems. *Computer-Aided Design* 2000; **32**(8–9):463–477. A preliminary version appeared in *Proceedings of the Fifth ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, June 1999.
3. Chen S, Tortorelli DA. Three-dimensional shape optimization with variational geometry. *Structural Optimization* 1997; **13**:81–94.
4. Sethian JA. *Level Set Methods and Fast Marching Methods*: *Evolving Interfaces in Computational Geometry*, *Fluid Mechanics*, *Computer Vision*, *and Materials Science*. Cambridge University Press: Cambridge, 1999.
5. Sethian JA, Wiegmann W. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics* 2000; **163**(2):489–528.
6. Allaire G, Jouve F, Toader AM. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics* 2004; **194**:363–393.
7. Wang MY, Wang XM, Guo DM. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**(1–2):227–246.
8. Wang SY, Wang MY. Radial basis functions and level set method for structural topology optimization. *International Journal for Numerical Methods in Engineering* 2005; **65**(12):2060–2090.
9. Belytschko T, Xiao SP, Parimi C. Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering* 2003; **57**:1177–1196.

DOI: 10.1002/nme

10. Haug EJ, Choi KK, Komkov V. *Design Sensitivity Analysis of Structural Systems*. Academic Press: New York, NY, 1986.
11. Rvachev VL. *Geometric Applications of Logic Algebra*. Naukova Dumka, 1967 (in Russian).
12. Shapiro V. Theory of R-functions and applications: a primer. *Technical Report*, Cornell University, November 1988.
13. Shapiro V, Tsukanov I. Meshfree simulation of deforming domains. *Computer-Aided Design* 1999; **31**:459–471.
14. Tsukanov I, Shapiro V. The architecture of SAGE—a meshfree system based on RFM. *Engineering with Computers* 2002; **18**(4):295–311.
15. Bennet JA, Botkin MA (eds). *The Optimum Shape*: *Automated Structural Design*, *General Motors Research Laboratory Symposia Series*. Plenum Press: New York, 1986.
16. Natekar D, Zhang X, Subbarayan G. Constructive solid analysis: a hierarchical, geometry based meshless procedure for integrated design and analysis. *Computer-Aided Design* 2004; **36**:472–486.
17. Xie YM, Steven GP. A simple evolutionary procedure for structural optimization. *Computers and Structures* 1993; **49**(5):885–896.
18. Eschenauer HA, Kobelev HA, Schumacher A. Bubble method for topology and shape optimization of structures. *Structural Optimization* 1994; **8**:142–151.
19. Garreau S, Masmoudi M, Guillaume P. The topological sensitivity for linear isotropic elasticity. *ECCM'99*. Munich, Germany.
20. Sokolowski J, Zochowski A. On topological derivative in shape optimization. *SIAM Journal on Control and Optimization* 1999; **37**:1251–1272.
21. Allaire G, de Gournay F, Jouve F, Toader AM. Structural optimization using topological and shape sensitivity via a level set method. *Control and Cybernetics* 2005; **34**:59–80.
22. Burger M, Hackl B, Ring W. Incorporating topological derivatives into level set methods. *Journal of Computational Physics* 2004; **194**:344–362.
23. Wang MY, Wei P. Topology optimization with level set method incorporating topological derivative. *Sixth World Congress on Structural and Multidisciplinary Optimization*, Rio de Janeiro, Brazil, 2005.
24. Bendsøe MP, Sigmund O. *Topology Optimization*: *Theory*, *Methods and Applications*. Springer: Berlin, Heidelberg, 2003.
25. Shapiro V, Tsukanov I. Implicit functions with guaranteed differential properties. *Fifth ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, 1999; 258–269.
26. Bloomenthal J. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers: Los Altos, CA, 1997.
27. Gomes J, Velho L, de Figueiredo LH. *Implicit Objects in Computer Graphics*. Springer: Berlin, 2002.
28. Ricci A. A constructive geometry for computer graphics. *Computer Journal* 1973; **16**(3):157–160.
29. Rvachev VL. *Theory of R-functions and Some Applications*. Naukova Dumka, 1982 (in Russian).
30. Shapiro V. Real functions for representation of rigid solids. *Computer-Aided Geometric Design* 1994; **11**(2): 153–175.
31. Frisken SF, Perry RN, Rockwood AP, Jones TR. Adaptively sampled distance fields: a general representation of shape for computer graphics. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 2000; 249–254.
32. Biswas A, Shapiro V. Approximate distance fields with non-vanishing gradients. *Graphical Models* 2004; **66**(3):133–159.
33. Muraki S. Volumetric shape description of range data using Blobby Model. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics* 1991; **25**(4):227–235.
34. Lim C, Turkiyyah GM, Ganter MA, Storti DW. Implicit reconstruction of solids from cloud point sets. *Proceedings of the Third Symposium on Solid Modeling and Applications*. ACM Press: New York, 1995; 393–402.
35. Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H-P. Multi-level partition of unity implicits. *ACM Transactions on Graphics* (*TOG*) 2003; **22**(3):463–470.
36. Turk G, O'Brien J. Modeling with implicit surfaces that interpolate. *ACM Transactions on Graphics* 1999; **21**(4):855–873.
37. Savchenko VV, Pasko AA, Okunev OG, Kunii TL. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum* 1995; **14**(4):181–188.
38. Kojekine N, Hagiwara I, Savchenko V. Software tools using csrbf's for processing scattered data. *Computers and Graphics* 2003; **27**(2):311–319.
39. Raviv A, Elber G. Three-dimensional freeform sculpting via zero sets of scalar trivariate functions. *Computer-Aided Design* 2000; **32**:513–526.

DOI: 10.1002/nme

40. Schmitt B, Pasko A, Schlick C. Constructive sculpting of heterogeneous volumetric objects using trivariate b-splines. *The Visual Computer* 2004; **20**(2):130–148.
41. Shapiro V. Well-formed set representations of solids. *International Journal on Computational Geometry and Applications* 1999; **9**(2):125–150.
42. Shapiro V. A convex deficiency tree algorithm for curved polygons. *International Journal of Computational Geometry and Applications* 2001; **11**(2):215–238.
43. Shapiro V. Solid modeling. In *Handbook of Computer-Aided Geometric Design*, Farin G, Hoschek J, Kim MS (eds). Elsevier Science Publishers: Amsterdam, 2002; 473–518.
44. de Boor C. *A Practical Guide to Splines*. Springer: Berlin, 2001.
45. Requicha A. Representations for rigid solids: theory, methods, and systems. *Computing Surveys* 1980; **12**(4): 437–464.
46. Shapiro V, Vossler DL. Construction and optimization of CSG representations. *Computer-Aided Design* 1991; **23**(1):4–20.
47. Shapiro V. Maintenance of geometric representations through space decompositions. *International Journal on Computational Geometry and Applications* 1997; **7**(4):383–418.
48. Rvachev VL. Analytical description of some geometric objects. *Doklady AS USSR* 1963; **153**(4):765–768.
49. Rvachev VL, Sheiko SI. *R*-functions in boundary value problems in mechanics. *Applied Mechanics Reviews* 1996; **48**(4):151–188.
50. Rvachev VL, Sheiko TI, Shapiro V, Tsukanov I. Transfinite interpolation over implicitly defined sets. *Computer-Aided Geometric Design* 2001; **18**(4):195–220.
51. Nocedal J, Wright SJ. *Numerical Optimization*. Springer: Berlin, 1999.
52. Freytag M, Shapiro V, Tsukanov I. Field modeling with sampled distances. *Computer-Aided Design* 2006; **38**(2):87–100.
53. Kantorovich LV, Krylov VI. *Approximate Methods of Higher Analysis*. Interscience Publishers: New York, 1958.
54. Rvachev VL, Sheiko TI, Shapiro V, Tsukanov I. On completeness of RFM solution structures. *Computational Mechanics* 2000; **25**:305–316.
55. Höllig K. *Finite Element Methods with B-Splines*. Frontiers in Applied Mathematics, vol. 26. SIAM: Philadelphia, PA, 2003.
56. Tsukanov I, Hall M. Data structure and algorithms for fast automatic differentiation. *International Journal for Numerical Methods in Engineering* 2003; **56**(13):1949–1972.
57. Tsukanov I, Hall M. Fast forward automatic differentiation library (FFADLib): a user manual. *Technical Report 2000-4*, Spatial Automation Laboratory, http://sal-cnc.me.wisc.edu, 2000.
58. Washizu K. *Variational Methods in Elasticity and Plasticity* (3rd edn). Pergamon Press/Oxford: New York, England, 1982.
59. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in C* (2nd edn). Cambridge University Press: Cambridge, MA, 1992.
60. Eschenauer HA, Olhoff N. Topology optimization of continuum structures: a review. *Applied Mechanics Review* 2001: **54**:331–390.

DOI: 10.1002/nme