Struct Multidisc Optim 22, 65–82 © Springer-Verlag 2001

Integration of topology and shape optimization for design of structural components

Poh-Soong Tang and Kuang-Hua Chang

Abstract This paper presents an integrated approach that supports the topology optimization and CAD-based shape optimization. The main contribution of the paper is using the geometric reconstruction technique that is mathematically sound and error bounded for creating solid models of the topologically optimized structures with smooth geometric boundary. This geometric reconstruction method extends the integration to 3-D applications. In addition, commercial Computer-Aided Design (CAD), finite element analysis (FEA), optimization, and application software tools are incorporated to support the integrated optimization process. The integration is carried out by first converting the geometry of the topologically optimized structure into smooth and parametric B-spline curves and surfaces. The B-spline curves and surfaces are then imported into a parametric CAD environment to build solid models of the structure. The control point movements of the B-spline curves or surfaces are defined as design variables for shape optimization, in which CAD-based design velocity field computations, design sensitivity analysis (DSA), and nonlinear programming are performed. Both 2-D plane stress and 3-D solid examples are presented to demonstrate the proposed approach.

Key words CAD, design sensitivity analysis, FEA, shape optimization, topology optimization

Received January 27, 2000
Communicated by J. Sobieski

Poh-Soong Tang[1] and Kuang-Hua Chang[2]

Concurrent Design and Manufacturing Research Laboratory and School of Aerospace and Mechanical Engineering, The University of Oklahoma, Norman, OK 73019
[1] e-mail: pstang@ou.edu
[2] e-mail: khchang@ou.edu

1 Introduction

Topology optimization has drawn significant attention in the recent development of structural optimization. This method has been proven very effective in determining the initial geometric shape for structural designs. The main drawback of the method, however, is that the topology optimization always leads to a nonsmooth structural geometry, while most of engineering applications require a smooth geometric shape, especially for manufacturing. On the other hand, shape optimization starts with a smooth geometric model that can be manufactured much more easily. However, the optimal shape is confined to the topology of the initial structural geometry. No additional holes can be created during the shape optimization process. Topology and shape optimization can be combined to support structural design effectively by taking advantage of both methods.

In order to obtain a smooth structural shape, a number of methods have been proposed. Marsan and Dutta (1996) proposed to use geometric modeling techniques, such as curve fitting and surface skinning, to reconstruct the geometry of the topologically optimized structures in a CAD environment. Contour matching and branch transition techniques from image processing are also used to reconstruct the geometry of the branches. Olhoff *et al.* (1991), Kumar and Gossard (1996) and Maute and Ramm (1997) on the other hand, not only smoothed the coarse structural boundaries but also integrated the topology optimization with shape optimization. Their methods, however, focus on 2-D applications. Olhoff *et al.* (1991) is among the first few who investigated the integration of topology and shape optimizations in CAD environment. In their work, the topology pre-processor HOMOPT is integrated with the structural shape optimization system CAOS (Computer-Aided Optimization of Shapes). The smooth design boundaries for shape optimization are sketched manually based on the topology optimization result. Kumar and Gossard

(1996), proposed a design optimization method that optimizes both shape and topology of the structural components. In their research, a shape density function is defined over a feasible domain and is represented by a continuous piece-wise interpolation function over the triangular finite elements used for structural analysis. Similarly, Maute and Ramm (1997) proposed a mixed Lagrangian-Eulerian design model that enables the topology and shape optimization to be performed in parallel or alternately.

Following the previous work of integrating topology and shape optimization, this paper presents an integrated approach that smoothes the boundary geometry of the topologically optimized structure, imports the geometry into a CAD environment, and employs CAD-based shape optimization (Hardee *et al.* 1999) to further fine tune the structural geometric shape for a desired structural performance. The proposed approach aims at supporting designs of both 2-D and 3-D structural components.

The proposed approach consists of five main steps, design problem definition, topology optimization using the homogenization method (Bendsøe and Kikuchi 1988) or the Solid Isotropic Microstructure with Penalty (SIMP) method (Rozvany *et al.* 1995), boundary smoothing techniques (Park and Kim 1996), geometric reconstruction in Computer-Aided Design (CAD) solid model form (Tang 1999), and CAD-based shape optimization (Hardee *et al.* 1999), as shown in Fig. 1. Adding strength to practical engineering applications, commercial CAD, FEA, optimization, and application software tools are incorporated into the approach. At the beginning of the process, topology optimization tool OptiStruct from Altair Computing, Inc. (1997) is used to obtain an optimal structural layout based on a finite element model. The coarse layout of the topology result is then smoothed using curve fitting and surface skinning techniques. In this paper, parametric cubic B-spline curves and surfaces are utilized for curve fitting and surface skinning, respectively, which approximate the boundary edges and surfaces of the coarse structural layout. During this approximation process, the control points, which govern the shape of the B-spline curves and surfaces, as well as basis functions are acquired. Then, through CAD API (Application Programming Interface), these control points and basis functions are imported into the CAD environment to reconstruct solid models that are enclosed by the B-spline curves or surfaces. For structures with branches, Boolean operations are employed. These imported control points also parameterize the boundary curves or surfaces of the reconstructed solid model and later serve as design variables for CAD-based shape optimization.

The CAD-based shape optimization uses the reconstructed structural solid model for design. The PolyFEM (PolyFEM Development Group 1994) kernel of *DesignWorks* (CADSI 1998), a p-adaptive solver, supports FEA's in the shape optimization process. Before the shape optimization iteration, CAD-based design velocity field (Hardee *et al.* 1999) is computed to characterize material point movements due to shape design change, using parametric mapping and boundary displacement methods (Chang and Choi 1992). Note that in general the design velocity field is not linearly dependent on design changes when Boolean subtraction (difference) operations are involved in generating the solid models. With the velocity field computed, an overall finite difference method is utilized to support shape design sensitivity analysis (DSA) during the optimization process. Both batch mode and interactive design optimization methods are used to perform the shape optimization. At the end of the shape optimization, the process returns to the CAD environment and updates the solid model according to the changes of the control point positions.

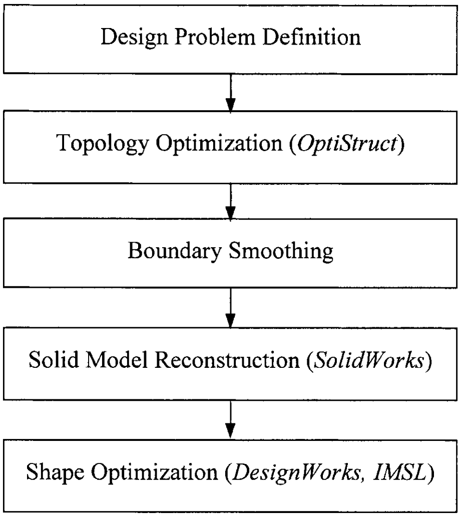

Design Problem Definition
Topology Optimization (*OptiStruct*)
Boundary Smoothing
Solid Model Reconstruction (*SolidWorks*)
Shape Optimization (*DesignWorks, IMSL*)

Fig. 1 Integrated optimization process

This paper will focus on presenting the proposed integration approach for topology and shape design optimization. For details of topology optimization methods the reader is referred to Bendsøe *et al.* (1995), Bendsøe and Kikuchi (1988), Rozvany *et al.* (1995), Yang (1997), Marsan and Dutta (1996), Maute and Ramm (1997), Olhoff *et al.* (1991), Kumar and Gossard (1996). For shape optimization, refer to Yao and Choi (1989), Braibant and Fleury (1984), Chang and Choi (1992), Botkin, (M.E. 1991), Kodiyalam *et al.* (1992), Hardee *et al.* (1999). The rest of the paper is organized as follows. In Sect. 2, the boundary smoothing techniques, including curve fitting and surface skinning, will be presented. The solid model reconstruction will be discussed in Sect. 3. The CAD-based design velocity field computation that supports the shape design optimization will be discussed in Sect. 4. Design examples, including a 2-D planar beam and a 3-D tracked vehicle roadarm, are presented in Sect. 5 to demonstrate the proposed method. A summary and future research is given in Sect. 6.

2
Boundary smoothing methods

The basic ideas of the proposed method for boundary smoothing include:

1. boundary node averaging,
2. the improved least square fitting using B-spline curves, and
3. surface skinning using B-spline surfaces.

The improved least square curve fitting employs a robust knot vector construction that guarantees existence of a set of unique control points of the B-spline fitting curve. In addition, this method allows designers to control the curve fitting error and to minimize the number of control points of the B-spline curve.

2.1
Averaging coordinates of boundary nodes

A set of finite element nodes that describe the boundary of the topologically optimized structure are identified first, for example, a 2-D planar beam shown in Fig. 2a. In Fig. 2a, a set of nodes are stored separately in proper orders for each of the four boundary edges identified. For 3-D structures, the boundary nodes are collected separately on pre-selected sections since curve fitting will be performed on each section separately. These boundary nodes will be referred to as geometric points throughout the paper. The Cartesian coordinates of the geometric points are then acquired for curve fitting. A simple averaging filter is first applied to smooth the coordinates of the geometric points by

$$\mathbf{P}_j = \frac{1}{5}(\mathbf{P}_{j-2}+\mathbf{P}_{j-1}+\mathbf{P}_j+\mathbf{P}_{j+1}+\mathbf{P}_{j+2})\,,$$
$$j=2,\ldots,r-2\,, \tag{1}$$

where $\mathbf{P}_j$ is the position vector of the j-th geometry point in the geometric point set, and $r+1$ is the total number of geometric points in the set. This averaging process reduces the wiggle property inherited from the finite element mesh. The smoothed sets of geometric points of the 2-D beam are shown in Fig. 2b.

2.2
Least square fitting using B-spline curves

Using the least square fitting for geometric points on a pre-selected structural section, the best fitting curve is obtained by minimizing the distance sum f between the curve and the geometric points. The distance sum f is defined as

$$f = \sum_{j=0}^{r} \|\mathbf{P}_j - \mathbf{x}(u_j)\|^2\,, \tag{2}$$

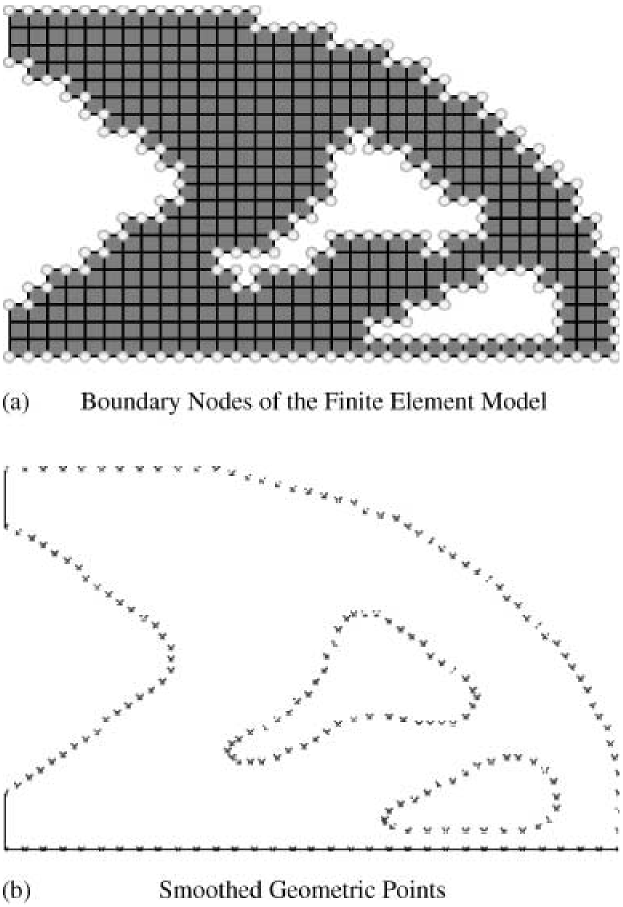

(a) Boundary Nodes of the Finite Element Model
(b) Smoothed Geometric Points

Fig. 2 Boundary nodes of the 2-D beam example

where $\|\bullet\|$ is the norm of the vector $\bullet$, $\mathbf{x}(u)$ is the fitting B-spline curve, $\mathbf{x}(u_j)=[x_1(u_j),x_2(u_j),x_3(u_j)]$ is the position vector of the fitting B-spline curve at u_j, where u is the parametric coordinate of the curve, and u_j is defined by the length ratio of the polygon formed by the geometric points $\mathbf{P}_j$, as illustrated in Fig. 3a. The values of u_j are determined by

$$u_0 = 0\,,$$
$$u_j = (r+1)\frac{\sum_{k=0}^{j-1} |\mathbf{P}_{(k+1)\mathrm{mod}(r+1)} - \mathbf{P}_k|}{\sum_{k=0}^{r} |\mathbf{P}_{(k+1)\mathrm{mod}(r+1)} - \mathbf{P}_k|}\,,$$
$$j=1,\ldots,r\,. \tag{3}$$

Note that u_j can be further adjusted for reducing the curve fitting error using, for example, the chord length parameterization method (Tang 1999). Mathematically, the B-spline curve is defined as (Mortenson 1985),

$$\mathbf{x}(u) = \sum_{i=0}^{n} \mathbf{B}_i N_{i,k}(u)\,, \tag{4}$$

where $\mathbf{B}_i$ is the i-th control point shown in Fig. 3b,

$n+1$ is the number of control points, and $N_{i,k}(u)$ is the basis function of the B-spline curve defined recursively as

$$N_{i,k}(u) = \frac{(u-t_i)N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}}$$

and

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } t_i \leq u \leq t_{i+1}\,, \\ 0 & \text{otherwise}\,, \end{cases} \tag{5}$$

where $[t_i, t_{i+1})$ is a knot span formed by the two consecutive knots t_i and t_{i+1}, and $k-1$ is the polynomial order of the basis functions.

In order to minimize f, the derivative of f with respect to the $n+1$ control points is set to zero. For simplicity, considering only the ℓ-th control point, one has

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{B}_\ell} = \sum_{j=0}^{r} \left\| -2\mathbf{P}_j \sum_{i=0}^{n} N_{i,k}(u_j) + \right.$$

$$\left. 2\sum_{i=0}^{n} N_{i,k}(u_j) \left(\sum_{i=0}^{n} N_{i,k}(u_j)\mathbf{B}_\ell \right) \right\| = 0\,. \tag{6}$$

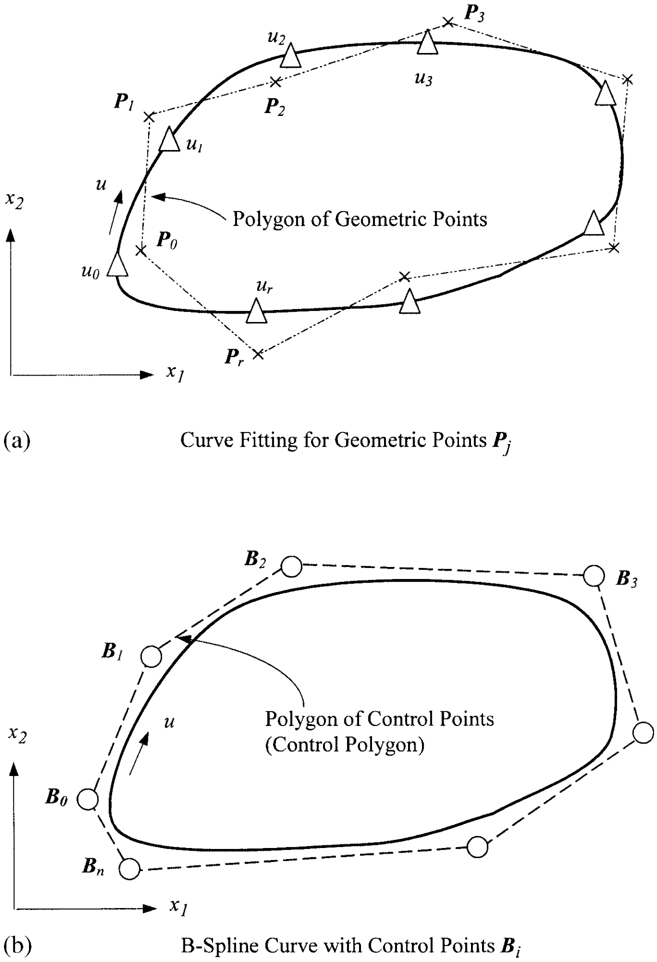
For $\ell = 0, \ldots, n$, the above expression can be rewritten in a matrix form as

$$\mathbf{N}^T\mathbf{N}\mathbf{B} = \mathbf{N}^T\mathbf{P}\,, \tag{7}$$

where $\mathbf{N} \in \mathbf{R}^{(r+1)\times(n+1)}$, $\mathbf{B} \in \mathbf{R}^{(n+1)\times 3}$, $\mathbf{P} \in \mathbf{R}^{(r+1)\times 3}$, and

$$\mathbf{N} = \begin{bmatrix} N_{0,k}(u_0) & N_{1,k}(u_0) & \ldots & N_{n,k}(u_0) \\ N_{0,k}(u_1) & N_{1,k}(u_1) & \ldots & N_{n,k}(u_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,k}(u_r) & N_{1,k}(u_r) & \ldots & N_{n,k}(u_r) \end{bmatrix}_{(r+1)\times(n+1)} \tag{8}$$

According to DeBoor (1978), $\mathbf{N}^T\mathbf{N}$ is invertible if $N_{i,k}$ $(u_j) \neq 0$. This is true if and only if $t_{i-k+1} < u_j < t_{i+1}$, for $i = 0, \ldots, n$, and $j = 0, \ldots, r$. This implies that there must exist at least one u_j in at least one knot span so that $N_{i,k}(u_j) \neq 0$ for all basis functions. This requirement can be achieved by adjusting the knot values of the basis functions (Tang 1999). As shown in the computational algorithm listed in Fig. 4, the curve fitting error can be controlled by adjusting the polynomial order and the number of control points. This curve fitting method allows for minimizing the number of control points for a required error bound, which avoids the difficulty of having a large number of design variables for shape optimization. The output of the curve fitting is a set of control points and basis functions that describe the smoothed contour. The smoothed boundary of the 2-D beam is shown in Fig. 5, where four boundary edges are approximated using cubic B-spline curves. The curve fitting method using B-spline curves provides a direct link to parametric CAD models and supports CAD-based design optimization for 2-D structures.



(a) Curve Fitting for Geometric Points $\boldsymbol{P}_j$

(b) B-Spline Curve with Control Points $\boldsymbol{B}_i$

Fig. 3 B-spline curve fitting

1. Collect and sort geometric points;
2. Smooth geometric points by averaging their positions with two previous and two following nodes (optional);
3. Specify desired error bound ε;
4. Determine polynomial order (if a cubic B-Spline curve is employed, $k = 4$, and the curve is C^2-continuous);
5. Determine number of control points, $n+1$, as small as possible;
6. Construct the B-Spline curve, reduce fitting error by adjusting u_j values using for example the chord length parameterization method;
7. Calculate $e = f/(r+1)$, if $e < \varepsilon$, acquire control points $\boldsymbol{B}_i$ to construct B-Spline curve;
8. Otherwise, increase k or n, repeat steps 4 to 7.

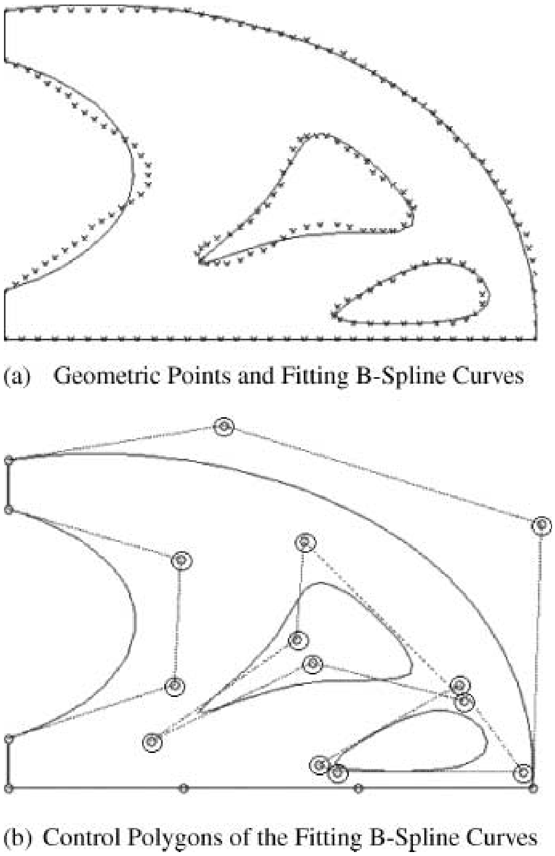
Fig. 4 Computational algorithm of B-spline curve fitting

2.3 Surface skinning

The fitting B-spline curves discussed above are then related across sections to form an open B-spline surface, as shown in Fig. 6, using the surface skinning technique. Note that in this process, the number of control points of the B-spline curves must be kept identical across sections. In addition, the polynomial order of the basis functions and knot values of the B-spline curves must be identical on all sections. The control points are connected to their corresponding points across sections, as shown in Fig. 6, to form a control polyhedron. The enclosed B-spline surface is then constructed by

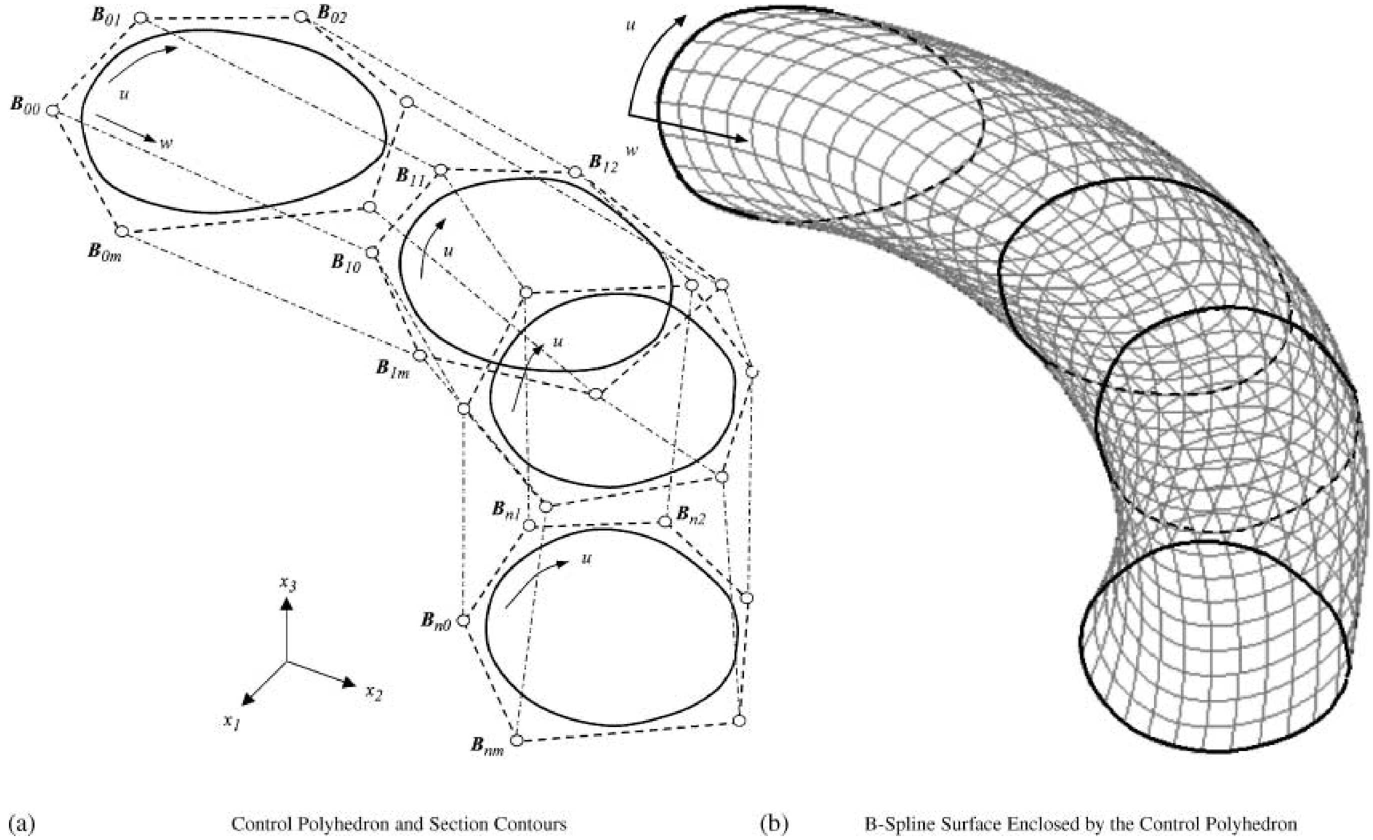
$$\mathbf{x}(u,w) = \sum_{i=0}^{n} \sum_{j=0}^{m} \mathbf{B}_{ij} N_{i,k}(u) M_{j,\ell}(w)\,, \tag{9}$$

where $n+1$ and $m+1$ are the numbers of control points in the u and w parametric directions, respectively; and $k-1$ and $\ell-1$ are the polynomial orders of the basis functions $N_{i,k}(u)$ and $M_{j,\ell}(w)$, respectively. Note that the B-spline surface constructed is C^2-continuous in both u and w parametric directions, if cubic basis functions are assumed. The control points and basis functions of the B-spline surface can be imported into CAD tools to support solid modeling and shape optimization. The com-


(a) Geometric Points and Fitting B-Spline Curves

(b) Control Polygons of the Fitting B-Spline Curves

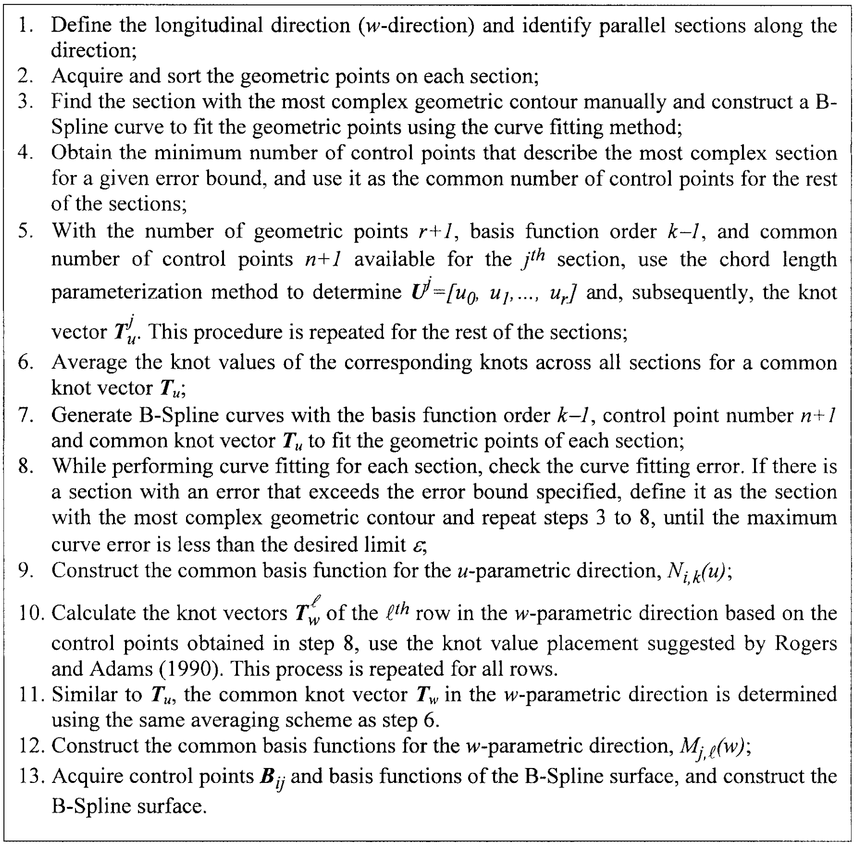
Fig. 5 Smoothed boundary of the 2-D beam example



(a) Control Polyhedron and Section Contours

(b) B-Spline Surface Enclosed by the Control Polyhedron

Fig. 6 B-spline surface skinning



1. Define the longitudinal direction (w-direction) and identify parallel sections along the direction;
2. Acquire and sort the geometric points on each section;
3. Find the section with the most complex geometric contour manually and construct a B-Spline curve to fit the geometric points using the curve fitting method;
4. Obtain the minimum number of control points that describe the most complex section for a given error bound, and use it as the common number of control points for the rest of the sections;
5. With the number of geometric points $r+1$, basis function order $k-1$, and common number of control points $n+1$ available for the j^{th} section, use the chord length parameterization method to determine $\boldsymbol{U}^j=[u_0, u_1, ..., u_r]$ and, subsequently, the knot vector $\boldsymbol{T}_u^j$. This procedure is repeated for the rest of the sections;
6. Average the knot values of the corresponding knots across all sections for a common knot vector $\boldsymbol{T}_u$;
7. Generate B-Spline curves with the basis function order $k-1$, control point number $n+1$ and common knot vector $\boldsymbol{T}_u$ to fit the geometric points of each section;
8. While performing curve fitting for each section, check the curve fitting error. If there is a section with an error that exceeds the error bound specified, define it as the section with the most complex geometric contour and repeat steps 3 to 8, until the maximum curve error is less than the desired limit ε;
9. Construct the common basis function for the u-parametric direction, $N_{i,k}(u)$;
10. Calculate the knot vectors $\boldsymbol{T}_w^{\ell}$ of the ℓ^{th} row in the w-parametric direction based on the control points obtained in step 8, use the knot value placement suggested by Rogers and Adams (1990). This process is repeated for all rows.
11. Similar to $\boldsymbol{T}_u$, the common knot vector $\boldsymbol{T}_w$ in the w-parametric direction is determined using the same averaging scheme as step 6.
12. Construct the common basis functions for the w-parametric direction, $M_{j,\ell}(w)$;
13. Acquire control points $\boldsymbol{B}_{ij}$ and basis functions of the B-Spline surface, and construct the B-Spline surface.

Fig. 7 Computational algorithm of B-spline surface skinning

putation algorithm of the surface skinning is summarized in Fig. 7.

3 Solid model reconstruction

The control points $\mathbf{B}_{ij}$ and basis functions $N_{i,k}(u)$ and $M_{j,\ell}(u)$ are imported into a CAD environment, for example SolidWorks (SolidWorks Corporation 1998), through API to construct the B-spline surface. The solid model enclosed by the B-spline surface can be created by extruding a solid block over the B-spline surface and then using the surface cut feature in CAD to trim the solid block to the shape enclosed by the surface. Note that the surface skinning and solid model construction processes directly support structures with flat ends and no branches. Solid model construction for structure with nonflat end and multiple branches is explained next.

3.1 End capping

For models with nonflat ends, caps can be added to approximate the structural boundary using the "dome" feature available in CAD, such as SolidWorks. The dome feature maintains a C^0-continuity from the flat end face, as shown in Fig. 8. Note that the height of the cap can be adjusted to achieve the best approximation. The dome feature is fully parameterized, which means that the cap will be automatically updated due to changes of the geometric shape of the end face while maintaining C^0-continuity. In addition, C^1-continuity is achievable by creating fillets at the junction edges, i.e. the boundary edges of the flat end face.

In addition to the dome feature, end caps can be generated by introducing additional control points on the end face of the control polyhedron as part of the surface skinning process (Fig. 9). Note that more than one control points can be added to the end face and each of them can be adjusted to best fit the geometry of the structure. Note that a C^2-continuity is maintained between the cap and the flat end face if a bi-cubic B-spline surface is employed. This method provides a greater flexibility

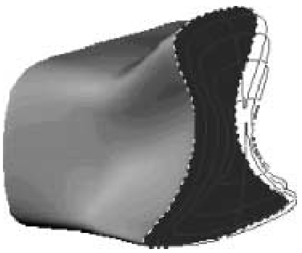
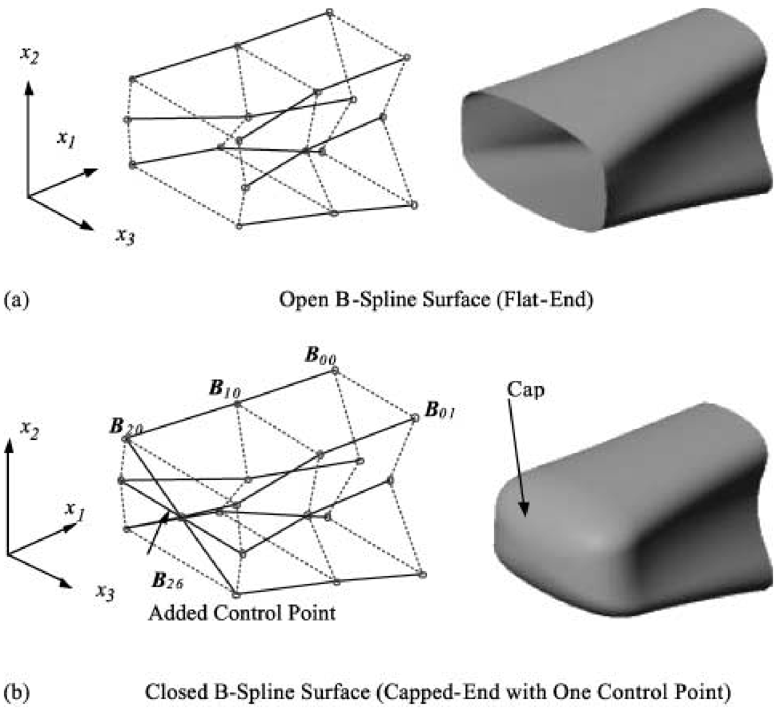


Fig. 8 Flat-end with a dome feature



(a) Open B-Spline Surface (Flat-End)

(b) Closed B-Spline Surface (Capped-End with One Control Point)

Fig. 9 B-spline surface with and without end cap

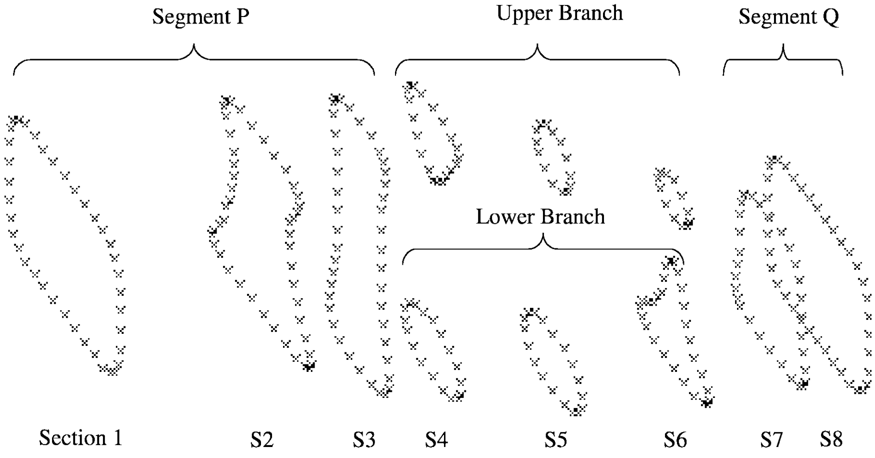


Fig. 10 Smoothed geometric points on serial cross-sections

in controlling the shape of the end caps and generates smoother caps that are more desirable for shape optimization. When a more complicated structure is encountered, for example, a structure with branches, Boolean operations, including union and subtraction (difference), can be employed for creating the solid models. These operations are discussed next.

3.2 Multiple branches: union

For models with multiple branches, the C^0-continuity between the segments and the branches can be maintained by constructing a transition section at the mid-plane. The transition contour is constructed using methods proposed by Marsan and Dutta (1996). In the example shown in Fig. 10, eight sections are smoothed with 11 closed cubic B-spline curves. Two branches are merged into the segments between sections 3 and 4 as well as sections 6 and 7. In this paper, the transition section between sections 3 and 4 will be discussed. Note that the solid models of the branches and the segment P can be generated separately following the methods discussed earlier, as illustrated in Fig. 11. In addition, the number of control points of every section in the segments must be identical to the sum of those of the upper and lower contours in every branch section.

The first step of creating the transition section is to connect the centroids of the top and bottom control poly-

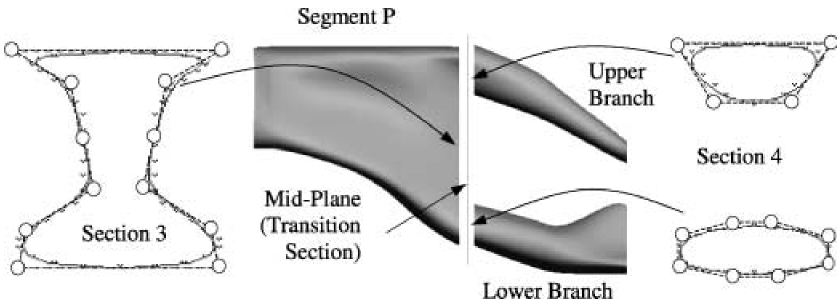
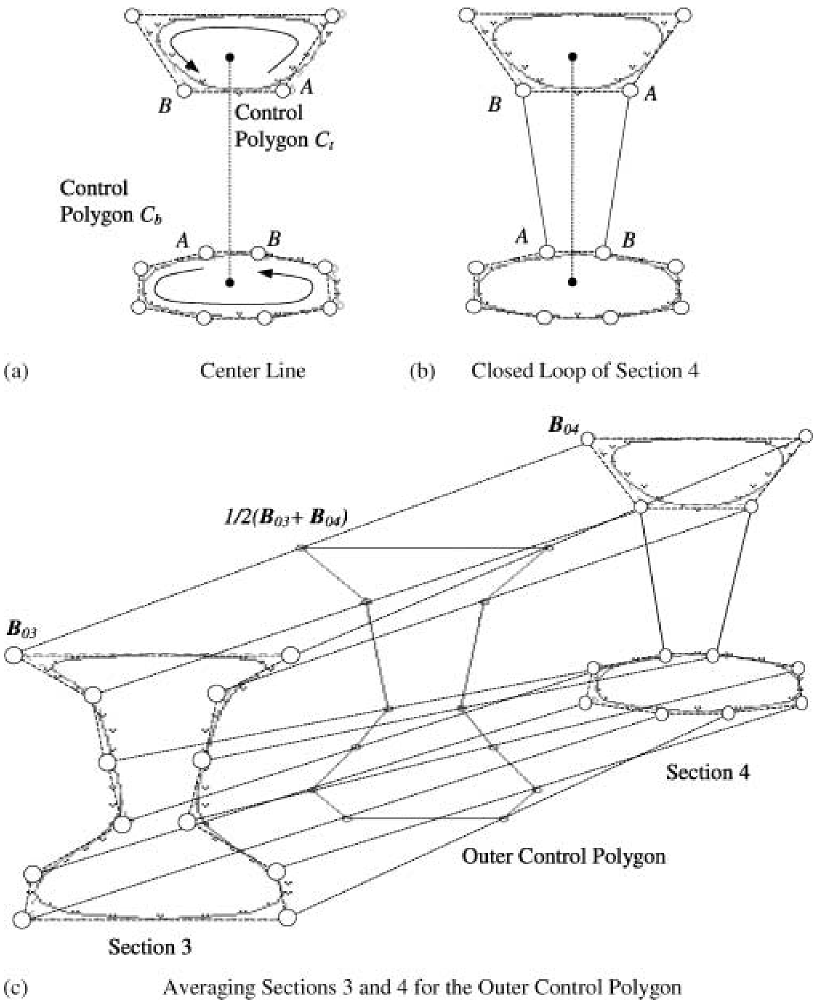


Fig. 11 Transition section between P and branches

gons C_t and C_b, respectively, in section 4 to form a centreline, as shown in Fig. 12a. The control points that are closer to the centreline are identified as points A and B in each polygon. Point A of a polygon is the first closer control point to the centreline in the counter-clockwise direction, as shown in Fig. 12a. Points B are connected to points A of the other polygon to form a closed loop of section 4 (Fig. 12b). The outer control polygon of the transition section is constructed by averaging the coordinates of the control points of the polygons of section 3 and the closed loop of section 4, as shown in Fig. 12c. The inner control polygon of the transition section is constructed following the same procedure as the outer one, except that the averaging is performed with the control point positions of the closed loop of section 4 and the centroid of section 3, as shown in Fig. 13a. Twelve control points are selected manually to form the control polygon of the transition section, as shown in Fig. 13b. A closed B-spline curve can be formed using the control polygon on the transition section.

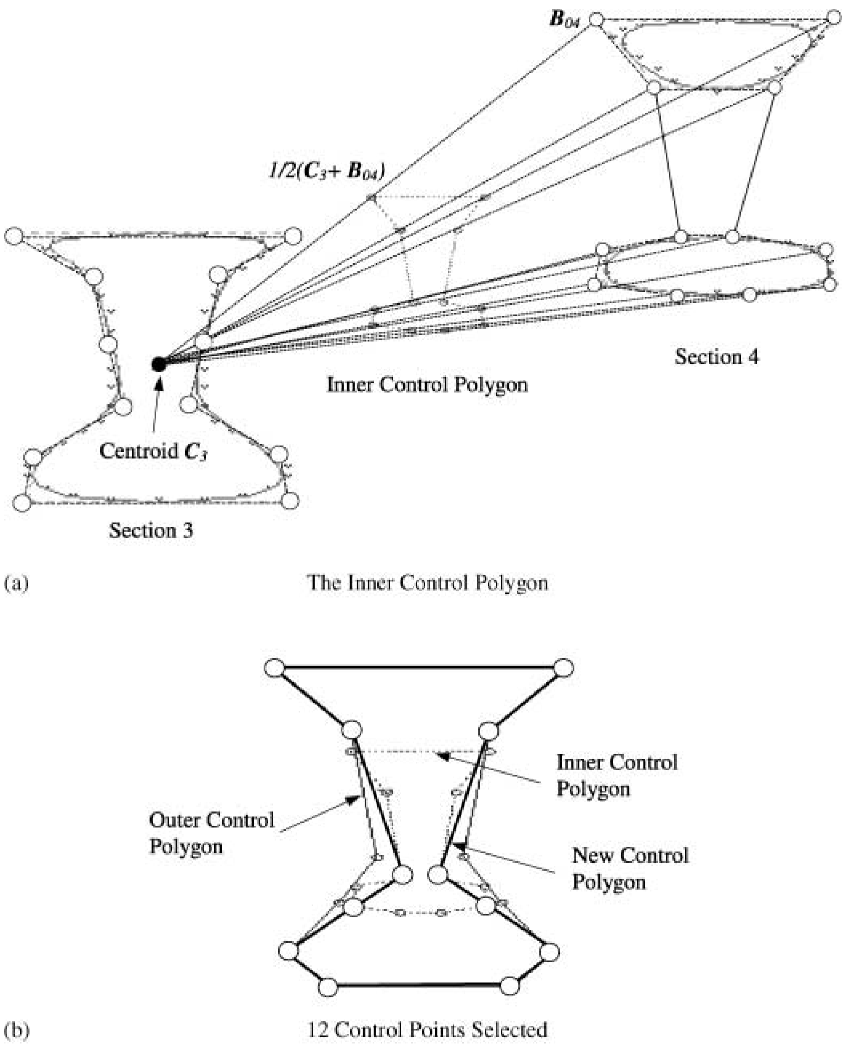
The transition between sections 3 and 4 is then constructed by extending the segment P from section 3 and



(a) Center Line (b) Closed Loop of Section 4

(c) Averaging Sections 3 and 4 for the Outer Control Polygon

Fig. 12 Construction of control polygons at the transition section



(a) The Inner Control Polygon

(b) 12 Control Points Selected

Fig. 13 Construction of control polygons at the transition section (cont'd)

the branches from section 4 to the B-spline curve enclosed by the control polygon on the transition section, as shown in Fig. 14. Note that this method provides a smooth transition between the branches and the segments. A C^1-continuity at the junction areas is preserved by creating fillets. Note that in this method, a significant number of control points and geometric parameters are generated. This increases the difficulty in shape design optimization. In addition, manual work is involved in determining the control polygon on the transition section.

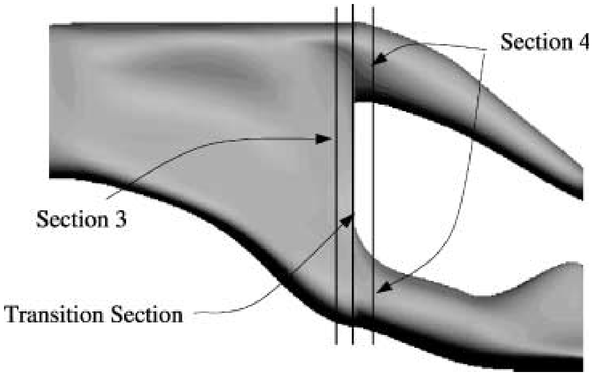


Fig. 14 Solid model with smooth transition

3.3 Multiple branches: subtraction

The same example discussed in Sect. 3.2 is used. Geometric points of five representative sections, S1, S4, S5, S6, and S8 are selected and fitted with closed B-spline curves, as shown in Fig. 15. Following the surface skinning method, an outer polyhedron formed by 6×5 manually selected control points and the enclosed B-spline surface are created (Fig. 16a). Similarly, an inner B-spline surface (enclosed by the 6×2 control points) that represents the middle hole is created (Fig. 16b). Note that the inner surface is a cylindrical surface. The solid models enclosed by the outer and inner B-spline surfaces, respectively, are created using solid extrusion and surface cut. The geometry of the final solid model is obtained by subtracting the inner solid from the outer one, as shown in Fig. 16c.

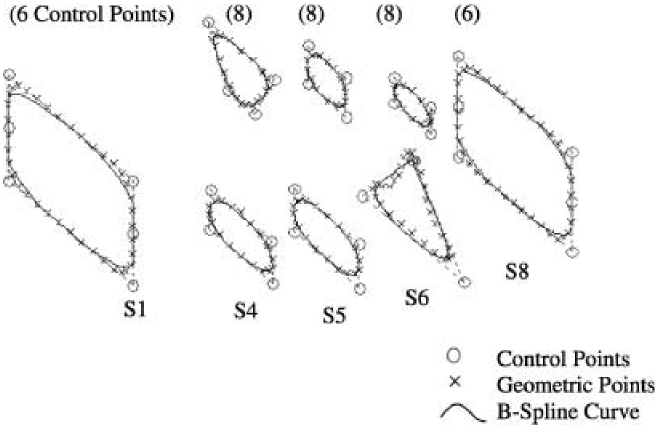


Fig. 15 Section contours and fitting curves

Note that the boundary surfaces of the solid model is C^2-continuous, except on the intersection edges, where C^0-continuity is observed.

In this method, the number of control points can be minimized to facilitate the shape optimization, yet capture essential geometric features. Even though the subtracted model does not capture detailed geometric features as those of the union model, shape optimization will be conducted to further improve structural performance by tuning up the structural geometry. Therefore, this subtracted model is more desirable for shape optimization. However, a linear design velocity field cannot be ensured for the subtracted model. A simple example will be presented in Sect. 4.5 to illustrate the case. Special treatment must be determined and applied. This treatment is problem dependent. A more general example will be discussed in Sect. 5.2.

4
CAD-based design velocity field computation

The design velocity field governs the movement of the structural material points due to design changes in shape optimization. In practice, FEA is used to evaluate the structural responses in each optimization iteration. When the structural shape is changed, the finite element mesh of the structure must be updated, and the local performance measuring points, such as stress measuring points, need to be changed accordingly.

In this paper, a hybrid design velocity computation method proposed by Choi and Chang (1994) is employed. The hybrid method is a combination of isoparametric mapping and boundary displacement methods. Isoparametric mapping method is first used to determine the boundary velocity field. Based on the boundary design velocity field, the boundary displacement method is employed to solve an auxiliary elasticity problem for the domain velocity. Note that the design velocity field must depend linearly on the variation of the shape design variable. This requirement stems from the fact that the sensitivity information predicts linear variations of the performance measures with respect to the variation of the shape design variables. Further details can be referred to Choi and Chang (1994).

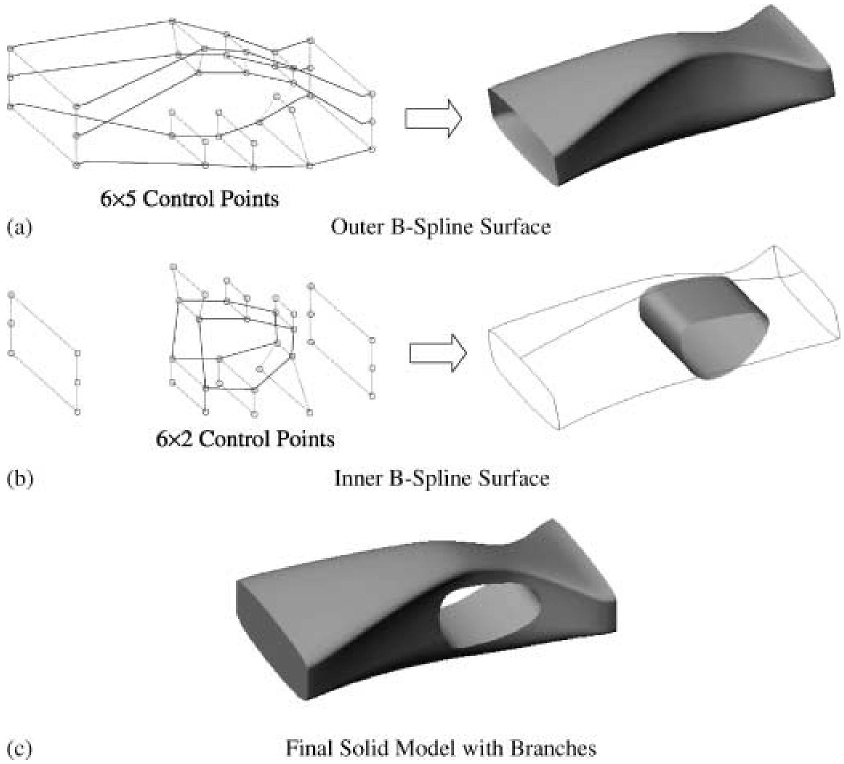


Fig. 16 Solid model created using Boolean subtraction

4.1 Isoparametric mapping method

The isoparametric mapping method relates the movement of boundary finite element nodes, including corner and edge nodes, to the changes of shape design variables based on geometric governing equation of the design boundary. In this research, the boundary curves and surfaces of the solid models are represented by B-spline entities, as expressed in (4) and (9), respectively. Converting (4) and (9) into matrix form, one has

$$\mathbf{x}(u) = [N_{0,k}(u)\ N_{1,k}(u)\ \ldots\ N_{n,k}(u)] \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_n \end{bmatrix}, \tag{10}$$

and

$$\mathbf{x}(u,w) = [N_{0,k}(u)\ N_{1,k}(u)\ \ldots\ N_{n,k}(u)] \times \begin{bmatrix} \mathbf{B}_{00} & \mathbf{B}_{01} & \ldots & \mathbf{B}_{0m} \\ \mathbf{B}_{10} & \mathbf{B}_{11} & \ldots & \mathbf{B}_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{n0} & \mathbf{B}_{n1} & \ldots & \mathbf{B}_{nm} \end{bmatrix} \begin{bmatrix} M_{0,\ell}(w) \\ M_{1,\ell}(w) \\ \vdots \\ M_{m,\ell}(w) \end{bmatrix}. \tag{11}$$

The movements of the selected control points of the B-spline curves or surfaces are defined as design variables $\mathbf{b}$ for shape optimization. Note that the set of design variables is a collection of B_k^i or B_{ij}^k, where $k = 1, 2, 3$, representing x_1, x_2, and x_3 directions, respectively. The design velocity field of a node defined by u or (u, w) parameters of the B-spline curve or surface can be computed, respectively, by

$$\mathbf{V} = \delta\mathbf{x}(u) = [N_{0,k}(u)\ N_{1,k}(u)\ \ldots\ N_{n,k}(u)] \begin{bmatrix} \delta\mathbf{B}_0 \\ \delta\mathbf{B}_1 \\ \vdots \\ \delta\mathbf{B}_n \end{bmatrix}, \tag{12}$$

and

$$\mathbf{V} = \delta\mathbf{x}(u,w) = [N_{0,k}(u)\ N_{1,k}(u)\ \ldots\ N_{n,k}(u)] \times \begin{bmatrix} \delta\mathbf{B}_{00} & \delta\mathbf{B}_{01} & \ldots & \delta\mathbf{B}_{0m} \\ \delta\mathbf{B}_{10} & \delta\mathbf{B}_{11} & \ldots & \delta\mathbf{B}_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ \delta\mathbf{B}_{n0} & \delta\mathbf{B}_{n1} & \ldots & \delta\mathbf{B}_{nm} \end{bmatrix} \begin{bmatrix} M_{0,\ell}(w) \\ M_{1,\ell}(w) \\ \vdots \\ M_{m,\ell}(w) \end{bmatrix}. \tag{13}$$

where $\delta\mathbf{B}_i$ or $\delta\mathbf{B}_{ij}$ represents the movements of the control points. From (12) and (13), it is observed that the isoparametric mapping method generates a linear design velocity field at the boundary curve or surface with respect to the variation of design variables. Note the design velocity field is computed for each design variable, i.e. movement of a single or a number of control points in certain direction, individually.

For h-FEA, finite elements are bounded by linear or quadratic edges and faces for 2-D and 3-D structures, respectively. The finite element nodes are the endpoints and midpoints of the element edges. For p-FEA, more midpoints are employed for interpolation of the curve element boundary using higher order polynomials. The isoparametric mapping method supports both h- and p-FEA's, provided that the parametric locations of the boundary nodes on the B-spline curves or surfaces are available.

4.2 Boundary displacement method

Based on the boundary velocity field, the boundary displacement method is employed to calculate the domain velocity field. The domain velocity field is obtained by solving an auxiliary elasticity problem using the boundary velocity field as the prescribed displacements on the design boundary. The discretized equilibrium equation of the auxiliary problem can be written in finite element matrix form as

$$\begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{db} \\ \mathbf{K}_{db} & \mathbf{K}_{dd} \end{bmatrix} \begin{bmatrix} \mathbf{V}_b \\ \mathbf{V}_d \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b \\ \mathbf{0} \end{bmatrix} \tag{14}$$

where $\mathbf{K}$ is the reduced stiffness matrix of the auxiliary structure, $\mathbf{V}_b$ denotes the boundary velocity, $\mathbf{V}_d$ is the domain velocity, and $\mathbf{f}_b$ denotes the fictitious force induced by the pre-described displacement field. Note that the domain velocity depends on the boundary velocity linearly. Since the boundary velocity is linearly dependent on the variation of design variable, the linear dependency of design velocity field on design variation is maintained. More details can be referred to Choi and Chang (1994).

4.3 Implementation with SolidWorks and DesignWorks

In this research, the DesignWorks p-FEA tool is employed for structural analysis. A seamless integration between DesignWorks and SolidWorks facilitates the implementation. The CAD solid model created in SolidWorks can be imported into DesignWorks for FEA accurately. The finite element mesh is created automatically with brick, wedge, and tetrahedron elements by DesignWorks. Note that the solid model can be partitioned into several C^0-continuous blocks to facilitate the mesh generation and to ensure a more regular mesh.

For the present research, the parametric coordinates of a given finite element boundary node are determined

by SolidWorks via its API function,
`GetClosestPointsOn`(x_1, x_2, x_3),
where (x_1, x_2, x_3) are the Cartesian coordinates of the finite element node. In order to use the API function, a design boundary curve or surface where the node resides has to be pre-selected. For each boundary node, the API function takes the x_1, x_2, and x_3 coordinates of the node and gives the parametric coordinates u and w on the selected surface (or u on the curve). With the parametric coordinates available, the boundary design velocity field can be computed using (12) or (13).

The domain velocity field can be computed using the boundary displacement method by employing a p-version FEA with a lower polynomial order, for example cubic function in this paper, and the pre-specified boundary displacement field. The elements of p-version FEA are generally large, and the larger elements tend to experience less distortion when the design boundary changes.

4.4 CAD and finite element model updates

The velocity field and design perturbation are used to update the finite element mesh for the new design by

$$\mathbf{x}(\mathbf{b}+\delta\mathbf{b}) = \mathbf{x}(\mathbf{b}) + \delta\mathbf{x} = \mathbf{x}(\mathbf{b}) + \sum_{k=1}^{s} \mathbf{V}_k \delta b_k \,, \tag{15}$$

where $\mathbf{x}(\mathbf{b}+\delta\mathbf{b})$ and $\mathbf{x}(\mathbf{b})$ are the coordinates of the node $\mathbf{x}$ of the perturbed and the current designs, respectively; $\delta\mathbf{x}$ is the nodal point movement due to the design changes; $\mathbf{V}_k$ and $\delta\mathbf{b}_k$ are the design velocity field and the perturbation of the k-th design variable, respectively; and s is the total number of design variables. The nodal point coordinates can be calculated using (15) to update the finite element input data file for FEA's in the optimization iteration.

The finite element model updates in the optimization process are conducted externally to the CAD environment. When an optimal design is obtained, the control point positions of the B-spline curves or surfaces are imported into the CAD environment to update the CAD solid model. Note that the optimized finite element model is consistent to the CAD solid model, i.e. the finite element boundary nodes, calculated using (15) stay on the boundary surface of the CAD solid model. This is because a linear design velocity field is assumed, and no topological change is expected to occur in the shape optimization process (Hardee *et al.* 1999).

4.5 Nonlinear boundary design velocity field

When the Boolean subtraction operations are employed in creating the solid models, the linear design velocity

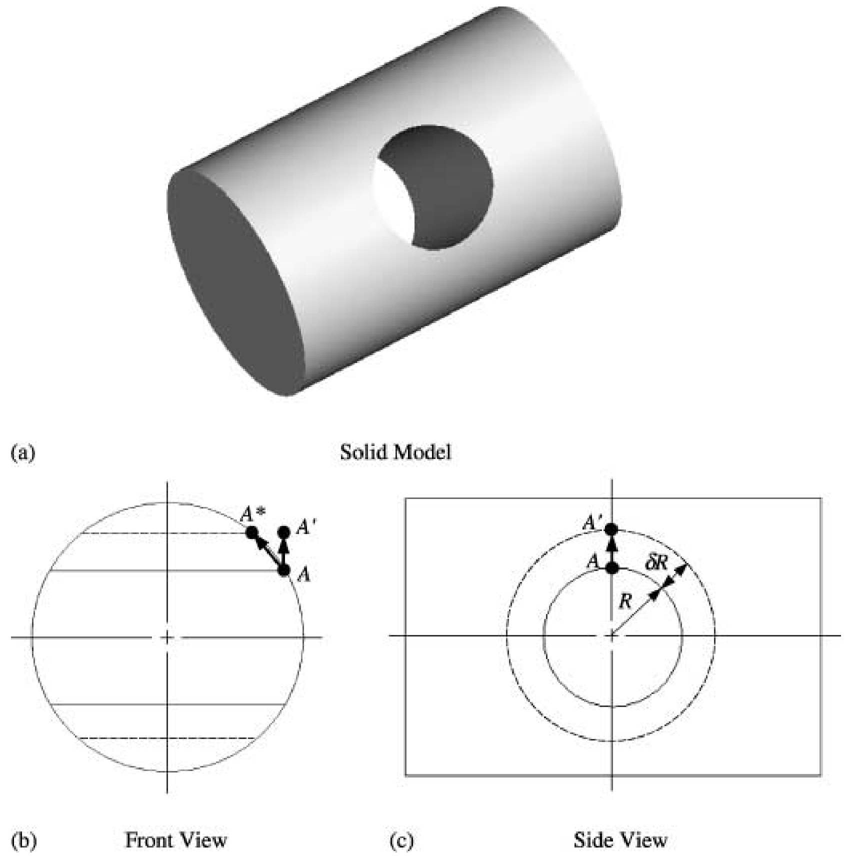


Fig. 17 Example for illustration of nonlinear design velocity

field may lead to an updated finite element model that is inconsistent to the CAD solid model, even without topological changes. To illustrate the point, consider a cylindrical bar that is cut through by a circular hole, as shown in Fig. 17a. Let the radius of the circular hole R be the design variable, and the design perturbation be δR. If the linear design velocity is assumed, point A on the intersection edge of the hole and the cylindrical bar will move to point A', which is out of the design domain, as shown in Figs. 17b and 17c. Note that the design velocity field, which gives the desired material movement, A to A*, is not linearly dependent on the design variation δR.

A very general design velocity field computation method that supports design problems such as the circular cylinder with a through hole is not straightforward. When such situation occurs, special treatment must be developed to ensure that a linear design velocity field is able to capture the design intents and lead to a desired optimal design.

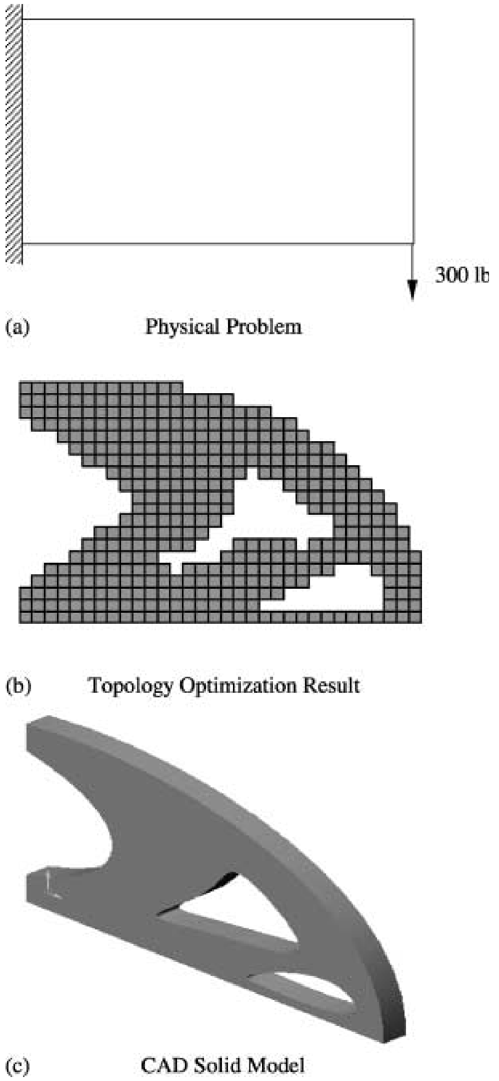
5 Design examples

Two design examples, a 2-D beam and a 3-D roadarm, are presented in this paper. To the authors' knowledge, this roadarm is the first 3-D design example presented in the context of integrated topology and shape optimization.

5.1 2-D plane stress example

A short beam of $16'' \times 14''$ with a point force shown in Fig. 18a, is employed to illustrate the proposed design process for 2-D applications. For topology optimization, the design problem is defined as minimizing the structural compliance subject to a 40% material reduction. There are 20×32 plane stress finite elements defined for the topology optimization. OptiStruct is employed for topology optimization. The finite elements with a material fraction less than 0.5 are eliminated. After the topology optimization result is obtained (Fig. 18b), the boundary averaging and curve fitting techniques are applied to construct smooth boundary edges, as shown in Fig. 5a, for shape optimization. Note that two open and two closed B-spline curves are created by the curve fitting technique. Seventeen control points are generated, as shown in Fig. 5b. The boundary geometry of the beam is then imported into SolidWorks to create a solid model in CAD (Fig. 18c) with a unit thickness.

The CAD solid model is transferred to DesignWorks for FEA. The finite element model is then created in DesignWorks with the mesh generated automatically, as shown in Fig. 19a. The von Mises stress distribution obtained from the p-version FEA is shown in Fig. 19b. The highest stress is found 5.43×10^2 psi, and is located at the bottom surface of the lower fixed end. The p-version FEA converges at the 4-th polynomial order with a 3.9% change in global strain energy.



(a) Physical Problem

(b) Topology Optimization Result

(c) CAD Solid Model

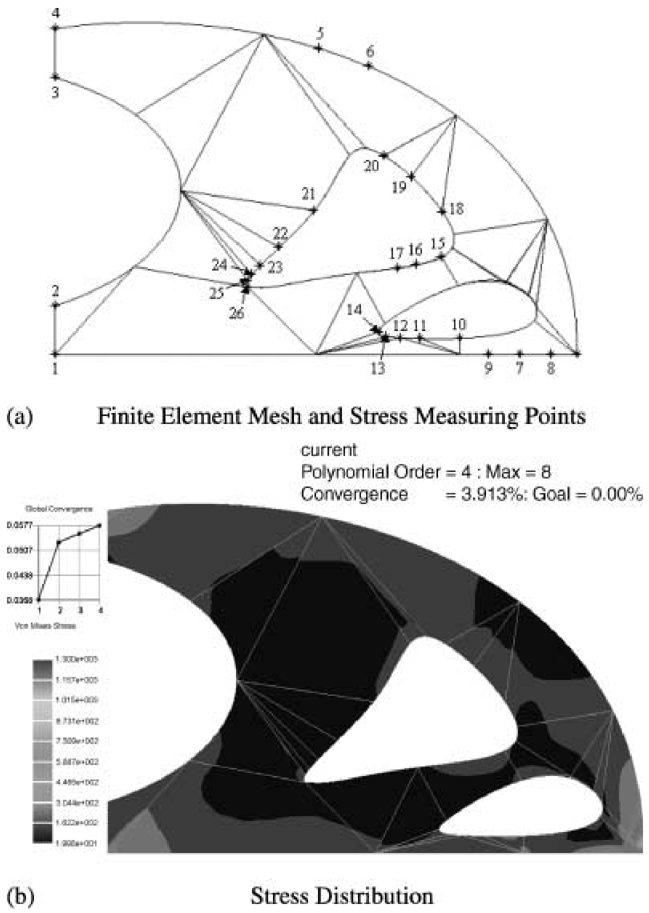
Fig. 18 2D beam example

Shape optimization is then conducted. In shape optimization, the objective function is structural mass, and constraint functions consist of the structural compliance measure obtained from topology optimization and 26 stress measures located at nodes labeled in Fig. 19a. Note that the stress upper bound is defined as 3,000 psi, and the beam material is SAE 1020 carbon steel with a yield strength 30 ksi. Hence, a safety factor of 10 is employed for design. The x_1- and x_2-movements of the 13 control points enclosed by circles in Fig. 5b are defined as the shape design variables. The design velocity fields are computed using the hybrid method. The overall finite difference method is employed for shape DSA.

The IMSL (Visual Numerics, Inc. 1998) optimization routine N0ONF is used for the nonlinear programming of the shape optimization process. An optimal design is obtained in 34 design iterations. Shape variations and stress distribution of the optimal design are shown in Figs. 20a and 20b, respectively. The highest stress is re-

Table 1 Changes of the mass and compliance vlues of the 2-D beam

	Initial design	Topology optimization	Shape optimization
Total mass	1.168×10^{-1} lb (100%)	7.300×10^{-2} lb (63%)	6.457×10^{-2} lb (55%)
Compliance	0.535 lb-in (100%)	0.123 lb-in (23%)	1.234×10^{-1} lb-in (23%)



(a) Finite Element Mesh and Stress Measuring Points

(b) Stress Distribution

Fig. 19 Finite Element mesh and stress distribution of the 2-D beam example

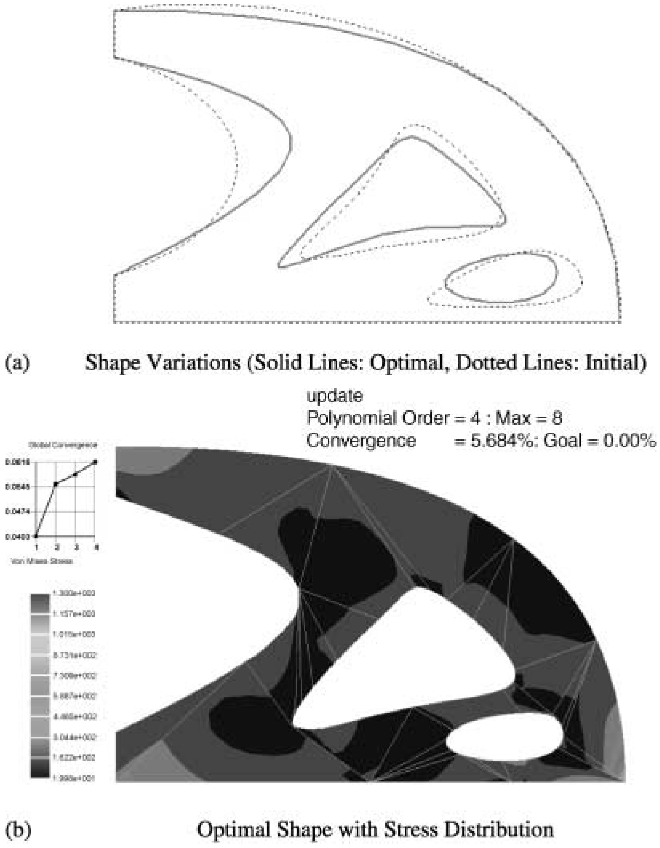
duced to 5.04×10^2 psi, and the highest stress is shifted to the top surface of the upper fixed end. The changes of mass and compliance throughout the integrated topology and shape optimizations are summarize in Table 1. The total mass and compliance reductions are 45% and 77%, respectively.

5.2 3-D tracked-vehicle roadarm example

A tracked-vehicle roadarm example shown in Fig. 21a is selected to demonstrate the proposed method for 3D applications. At the beginning of the topology optimization, a rough shape is assumed (Fig. 21b). The end of the torsion bar is fully constrained and an axial force of 2.15×10^4 lbs is applied at the wheel shaft, as shown in Fig. 21b.

The objective of the topology optimization is to minimize the roadarm compliance subject to a 50% mass reduction. The topologically optimized layout is shown in Fig. 21c. Note that a centre hole is generated as the result of the topology optimization. The section contours in the middle arm, as shown in Fig. 10, are first approximated using the averaging and curve fitting techniques. Both Boolean union and subtraction methods discussed in Sect. 3 are employed for reconstructing the solid models of the middle arm.

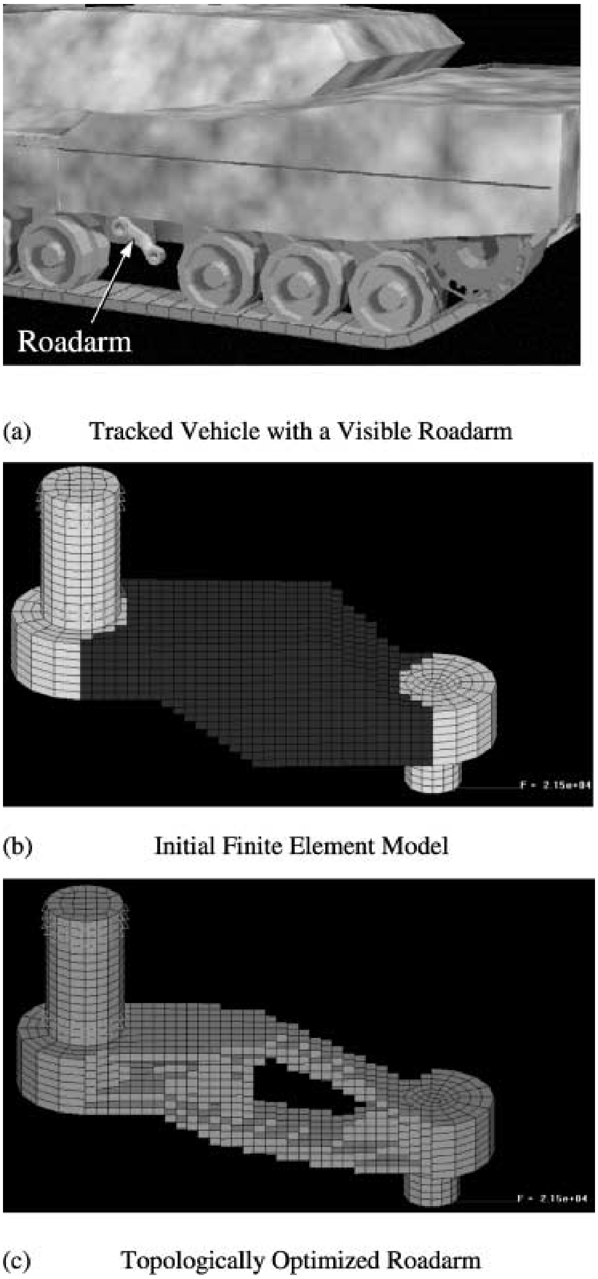
For the Boolean union technique, the middle arm sections are divided into 4 groups, i.e. segment P (sections 1 to 3), upper and lower branches (sections 4 to 6), and segment Q (sections 7 and 8). The surface skinning and solid extrusion/cut techniques are employed for creating solid models of the four pieces individually. The union operation described in Sect. 3.2 is conducted to create transition sections between sections 3 and 4, as well as sections 7 and 8, respectively. Fillets are created to smooth out the junction edges for a C^1-continuity. The cylinders at the ends of the middle arm are added in SolidWorks.



(a) Shape Variations (Solid Lines: Optimal, Dotted Lines: Initial)

(b) Optimal Shape with Stress Distribution

Fig. 20 Design optimization of the 2-D beam example

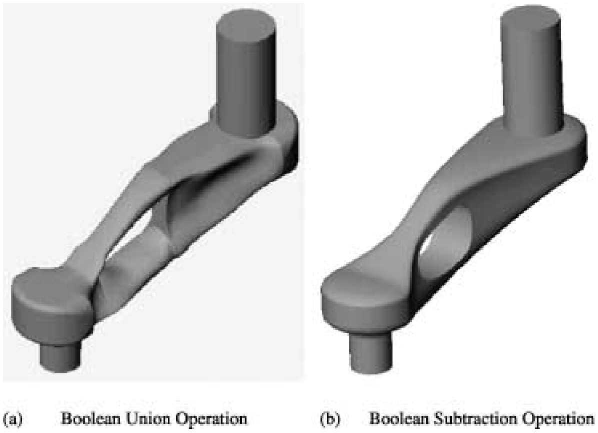


(a) Tracked Vehicle with a Visible Roadarm

(b) Initial Finite Element Model

(c) Topologically Optimized Roadarm

Fig. 21 The tracked vehicle roadarm

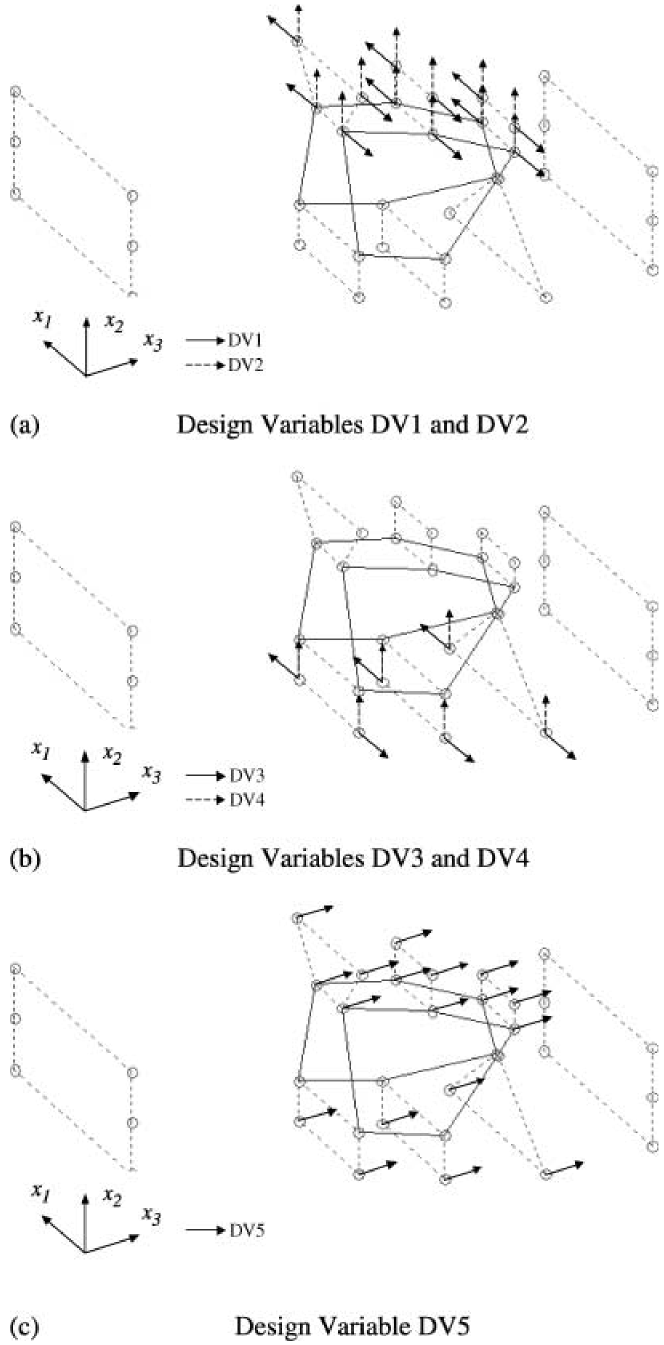


(a) Boolean Union Operation (b) Boolean Subtraction Operation

Fig. 22 Reconstructed solid models of the roadarm

The final solid model reconstructed is shown in Fig. 22a. Note that there are too many control points and geometric dimensions, such as the fillet radius, created in the process. This presents potential difficulty in shape optimization.

For the Boolean subtraction operation, five representative sections are selected for constructing the middle arm, as shown in Fig. 14. As discussed in Sect. 3.3, the outer and inner B-spline surfaces are constructed. The solid model of the middle arm is constructed by subtracting the inner solid from the outer one. The end cylinders are again added in SolidWorks. The final solid model reconstructed, as shown in Fig. 22b, is much smoother than



(a) Design Variables DV1 and DV2

(b) Design Variables DV3 and DV4

(c) Design Variable DV5

Fig. 23 Shape design parameterization

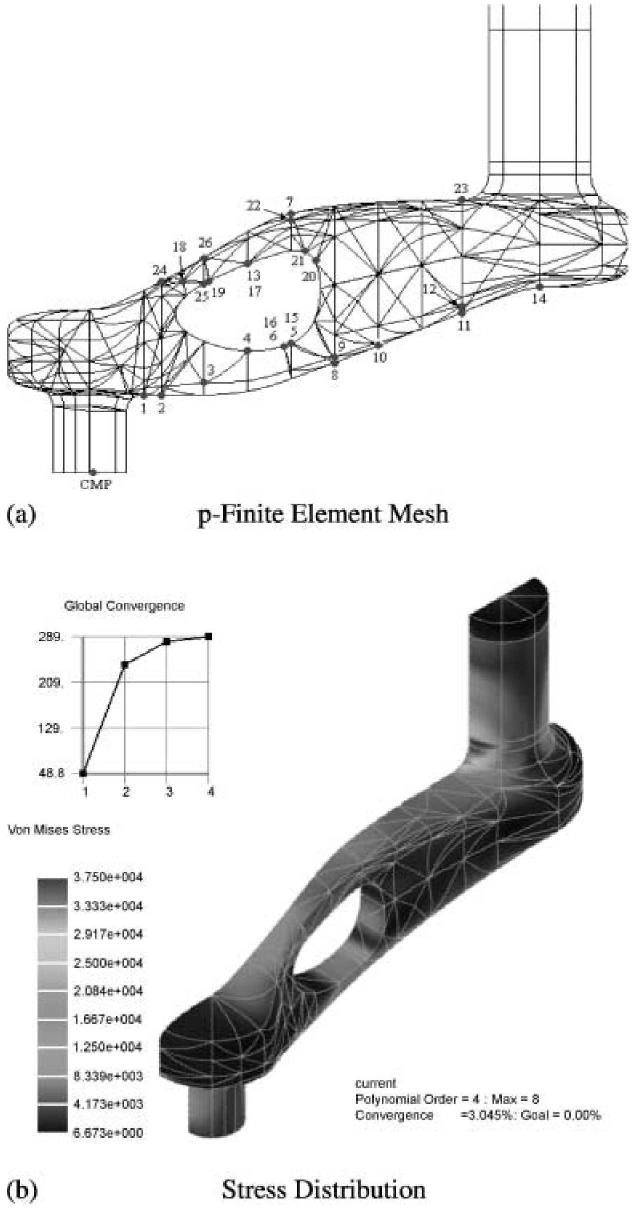
the one constructed by union. Note that there are much less number of control points in the solid model since less number of sections are used for surface skinning and no fillets are created.

Control point positions of the B-spline surfaces are combined for defining design variables for shape design optimization. There are five design variables. The first two characterize the shape and size of the upper branch, as shown in Fig. 23a. Similarly, the third and fourth characterize the shape and size of the lower branch, as shown in Fig. 23b. The last one defines the position of the inner hole along the longitudinal direction (x_3-direction) of the roadarm, as shown in Fig. 23c.

Due to symmetry, only half of the roadarm is meshed for FEA, as shown in Fig. 23a. Note that the half roadarm domain is subdivided into 12 smaller blocks to facilitate the mesh generation for a more regular finite element mesh. The p-version FEA is performed for the roadarm using DesignWorks. The solution converges at the 4-th polynomial order with a less than 5% change in the global strain energy. The stress contours of the initial design are shown in Fig. 24b. High stress areas are found around points 1, 6, 8, 9, 16, and 23, as identified in Fig. 24a.

According to the design variables shown in Fig. 22, the outer and inner B-spline surfaces will vary due to design changes. The material points along the intersection edges of the outer and inner surfaces are not linearly dependent on the design variables. When the control points shared by the outer and inner B-spline surfaces move, the material points on the intersection edges can move following either the outer or inner surface. Such movements are not unique. In order to avoid the problem, the boundary velocity field of both the outer and inner surfaces is determined by the outer surface only. The parametric locations of the finite element nodes of the intersection edges on the outer B-spline surface are first identified along with the other boundary nodes. The boundary velocity field on the outer surface, including the intersection edges, can be computed. The velocity field on the inner cylindrical surface is determined by the linear interpolation of the edge velocity toward the interior of the surface (along the x_1-direction). This can be easily achieved for the roadarm example since the inner surface is a cylindrical surface. Once the boundary velocity field is computed, the domain velocity can be calculated using the boundary displacement method, as discussed earlier.

Shape optimization is then conducted. In shape optimization, the objective function is structural mass, and constraint functions are the structural compliance measure obtained from topology optimization and 26 stress measures shown in Fig. 24a. Note that the stress upper



(a) p-Finite Element Mesh

(b) Stress Distribution

Fig. 24 Roadarm Finite Element analysis

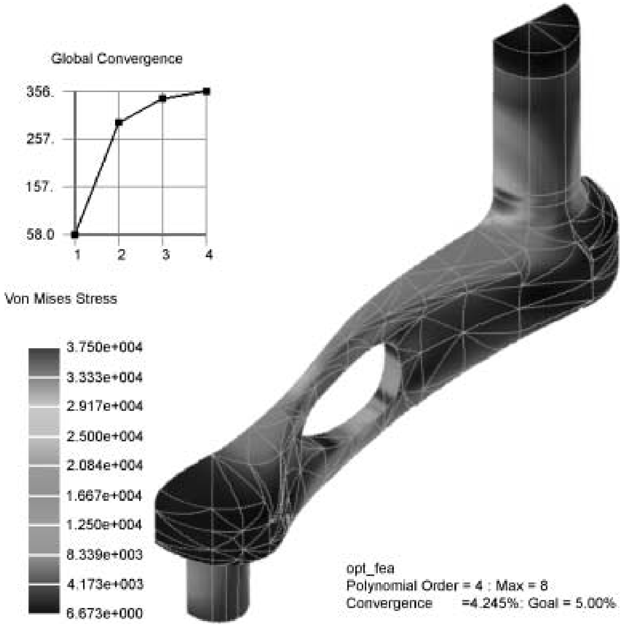
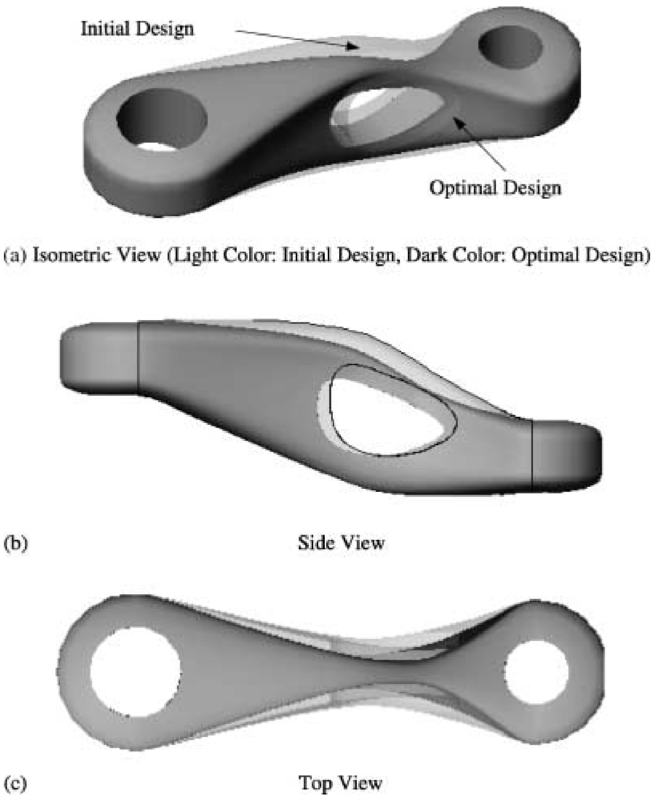


Fig. 25 Von Mises stress of the optimal roadarm

Table 2 Changes of the mass and compliance vlues of the 2-D beam

	Initial design	Topology optimization	Shape optimization
Total mass	9.132×10^{-1} lb (100%)	5.000×10^{-1} lb (55%)	4.868×10^{-1} lb (53%)
Compliance	7.705×10^{3} lb-in (100%)	1.434×10^{3} lb-in (19%)	1.434×10^{3} lb-in (19%)



(a) Isometric View (Light Color: Initial Design, Dark Color: Optimal Design)

(b) Side View

(c) Top View

Fig. 26 Comparison of the 3-D roadarm at initial and optimal designs

bound is defined as 37.5 ksi, and the material is SAE 1045 carbon steel with a yield strength 45 ksi. Hence, a safety factor of 1.2 is employed for the design.

The interactive design method, including design sensitivity display and what-if study (Chang *et al.* 1995), instead of the batch mode optimization, is employed for the design of roadarm. This is because that the finite element mesh tends to get distorted in shape optimization, even when the p-version finite element method is employed. The optimal shape is obtained in five design iterations. The von Mises stress distribution of the optimal design is shown in Fig. 25. The high stress areas are distributed around the upper and lower branch of the roadarm. The highest stress is found located at the top surface of lower branch with a value of 3.69×10^4 psi. The roadarm shape variation between initial and optimal designs is shown in Fig. 26. The changes of mass and compliance throughout the integrated topology and shape optimizations are summarized in Table 2. The total mass and compliance reductions are 47% and 81%, respectively.

6 Conclusions

An integrated approach that supports structural topology and shape optimizations has been presented. The geometric smoothing technique converts rough structural boundary obtained from topology optimization into smooth curves and surfaces for solid modeling and shape optimization. Both 2-D beam and 3-D roadarm examples have been presented to demonstrate the proposed design method. Even though this integrated capability is currently implemented with SolidWorks and DesignWorks, it can be extended to other CAD and FEA tools, as long as the CAD API functions are accepting the B-spline curves and surfaces and the FEA accepts the CAD solid models for mesh generation.

The integrated design approach has been demonstrated to be feasible for structural optimization. However, three fundamental issues require further study. The first issue is that the solid model generation methods proposed requires significant users interactions and decision-makings, especially when branches are involved. The second issue is that even when the p-FEA is employed for shape optimization, mesh distortion occurs. This issue makes the batch mode shape optimization impossible. Note that the newly developed meshless method holds the potential for resolving this issue. Initial work has been conducted recently (Grindeanu *et al.* 1999). The third issue is that special treatments are needed for ensuring that a linear design velocity field can be generated to support design of solid models involving branches.

Acknowledgements The authors would like to thank Altair Computing, Inc., Computer-Aided Design Software, Inc. (CADSI) and SolidWorks, Inc., for their support of using OptiStruct, DesignWorks and SolidWorks software tools for this research.

References

Altair Computing, Inc. 1997: *Altair OptiStruct 3.1 User's Manual*

Bendsøe, M.P.; Diaz, A.R.; Lipton, R.; Taylor, J.E. 1995: Optimal design of material properties and material distribution for multiple loading conditions. *Int. J. Numer. Meth. Engrg.* **38**, 1149–1170

Bendsøe, M.P.; Kikuchi, N. 1988: Generating optimal topologies in structural design using a homogenization method. *Comp. Meth. Appl. Mech. Eng., ASME* **71**, 197–224

Botkin, M.E. 1991: Shape design modeling using fully automatic three-dimensional mesh generation. *Finite Elements in Analysis and Design* **10**, 165–181

Braibant, V.; Fleury, C. 1984: Shape optimal design using B-splines. *Comp. Meth. App. Mech. Eng.* **44**, 247–267

Chang, K.H.; Choi, K.K. 1992: A geometry-based parameterization method for shape design of elastic solids. *Mech. Struct. & Mach.* **20**, 215–252

Chang, K.H.; Choi, K.K.; Tsai, C.S.; Chen, C.J.; Choi, B.S.; Yu, X. 1995: Design sensitivity analysis and optimization tool (DSO) for shape design applications. *Computing Systems Eng.* **6**, 151–175

Choi, K.K; Chang, K.H. 1994: A study of design velocity field computation for shape optimal design. *Finite Elements in Analysis and Design* **15**, 317–342

Computer-Aided Design Software, Inc. (CADSI) 1998: *DesignWorks98 tutorial and users guide.*

DeBoor, C. 1978: *A practical guide to splines*. Berlin, Heidelberg, New York: Springer

Grindeanu, I.; Choi, K.K.; Chen, J.S.; Chang, K.H. 1999: Shape design optimization of hyperelastic structures using a meshless method. *AIAA J.* **37**, 990–997

Hardee, E.; Chang, K.H.; Tu, J.; Choi, K.K.; Grindeanu, I.; Yu, X. 1999: CAD-based shape design sensitivity analysis and optimization. *Advances Eng. Software* **30**, 153–175

Kodiyalam, S.; Kumar, V.; Finnigan, P.M. 1992: A constructive solid geometry approach to three-dimensional shape optimization, *AIAA J.* **30**, 1408–1415

Kumar, A.V.; Gossard, D.C. 1996: Synthesis of optimal shape and topology of structures. *Trans. ASME* **118**, 68–74

Marsan, A.L.; Dutta, D. 1996: Construction of a surface model and layered manufacturing data from 3D homogenization output. *J. Mech. Des.* **118**, 412–418

Maute, K.; Ramm, E. 1997: Adaptive topology optimization of shell structures. *AIAA J.* **35**, 1767–1773

Mortenson, M.E. 1985: *Geometric modeling*. New York: John Wiley & Sons

Olhoff, N.; Bendsøe, M.P.; Rasmussen, J. 1991: On CAD-integrated structural topology and design optimization. *Comp. Meth. Appl. Mech. Eng.* **89**, 259–279

Park, H.; Kim, K. 1996: Smooth surface approximation to serial cross-sections. *J. Computer Aided Design* **28**, 995–1005

PolyFEM Development Group 1994: *PolyFEM-Pro/ENGINEER user guide and reference, release 2.0*. IBM Almaden Research Center

Rozvany, G.I.N.; Bendsøe, M.P.; Kirsch, U. 1995: Layout optimization of structures. *Appl. Mech. Rev. (ASME)* **48**, 41–119

SolidWorks Corporation 1998: *SolidWorks98 user's guide*

Tang, P.-S. 1999: *Integration of topology and shape optimizations for design of structural components*. M.S. Thesis, The University of Oklahoma

Visual Numerics, Inc. 1998: *IMSL Fortran 90 MP library 3.0 help*. Houston, TX

Yang, R.J. 1997: Multidiscipline topology optimization. *Comp. & Struct.* **63**, 1205–1212

Yao, T.M.; Choi, K.K. 1989: 3-D shape optimal design and automatic finite element regridding. *Int. J. Numer. Meth. Engrg.* **28**, 369–384