On increasing the developability of a trimmed NURBS surface

Charlie C. L. Wang*

Department of Automation and Computer-Aided Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong

Yu Wang **Matthew M. F. Yuen**

Department of Mechanical Engineering, Hong Kong University of Science and Technology, Clear Water Bay, N.T., Hong Kong

Abstract

Developable surfaces are desired in designing products manufactured from planar sheets. Trimmed NURBS surface patches are widely adopted to represent 3D products in CAD/CAM. This paper presents a new method to increase the developability of an arbitrarily trimmed NURBS surface patch. With this tool, designers can first create and modify the shape of a product without thinking about the developable constraint. When the design is finished, our approach is applied to increase the developability of the designed surface patches. Our method is an optimization based approach. After defining a function to identify the developability of a surface patch, the objective function for increasing the developability is derived. During the optimization, the positions and weights of the free control points are adjusted. When increasing the developability of a given surface patch, its deformation is also minimized and the singular points are avoided. G^0 continuity is reserved on the boundary curves during the optimization, and the method to reserve G^1 continuity across the boundaries is also discussed in this paper. Compared to other exist methods, our approach solves the problem in a novel way that is close to the design convention, and we are dealing with the developability problem of an arbitrarily trimmed NURBS patch.

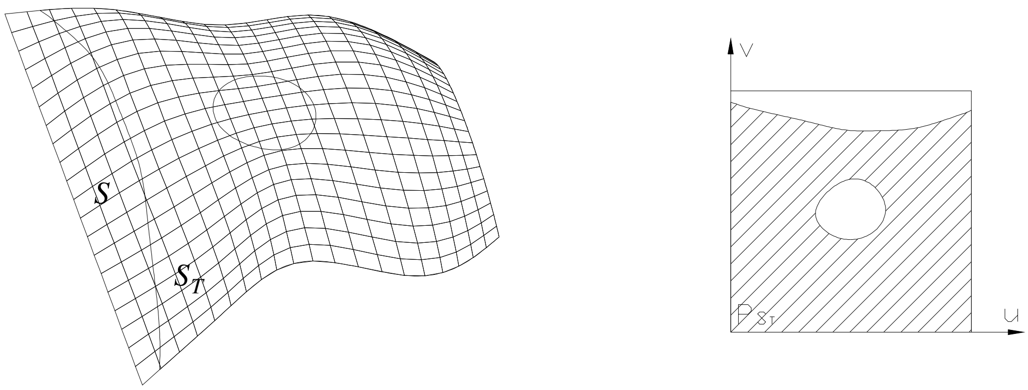
Keywords: developable surface, Gaussian curvature, trimmed surface, NURBS, and optimization.

* Corresponding author; e-mail: cwang@acae.cuhk.edu.hk

1. Introduction

Non-Uniform Rational B-Spline (NURBS) is the most popular representation method in CAD/CAM due to its generality and excellent properties. It is also a major geometric element in international product data transfer standards such as STEP (Standard for the Exchange of Product Model Data) and IGES (Initial Graphics Exchange Specification). In computer-aided engineering, geometric modeling, computer graphics, and many other applications, a parametric surface patch usually intersects with other surfaces, and thus only a portion of the surface patch is used in defining a meaningful shape [1]. The remaining portion of parametric surface patch S after trimming by other surfaces is called a trimmed (parametric) surface patch S_T. S_T is constrained by the same mathematical surface equation as $S(u,v)$, but its parametric domain is only a portion of that of S (see Fig. 1a). In current CAD/CAM systems, a trimmed parametric patch S_T is represented by a trimmed NURBS surface, where $S(u,v)$ is a NURBS surface, the parametric area of S_T – P_{S_T} lies inside $(u,v) \in [0,1] \times [0,1]$, and P_{S_T} is bounded by a number of curves (see Fig. 1b). Each boundary curve of P_{S_T} is expressed as a parametric equation of the form $b_i(t) = [u_i(t) \quad v_i(t)]$, where $t \in [0,1]$.

Bending or rolling a planar surface without stretching or tearing produces a developable surface; so developable surfaces are widely used in manufactured items from materials that are not allowed to have large stretches (e.g., ship hulls, ducts, shoes, clothing, aircraft and automobile parts [2-6]). Since developable surfaces are desired in manufacturing and the NURBS representation is widely used in CAD/CAM, this paper addresses the problem of increasing the developability of a trimmed NURBS surface – so as to decrease the stretches when forming its shape from a planar sheet in manufacturing.



(a) a trimmed surface **(b) the parametric area in $u-v$ domain**

Fig. 1 A trimmed surface and its related parametric area

Research related to *Computer Aided Geometric Design*, in particular those concerning the design and approximation of developable surfaces, are found in [6-15]. They are mostly in terms of NURBS or its special case – B-spline or Bézier surfaces [6-12]. Aumann [7] proposed the condition under which a developable Bézier surface can be constructed with two boundary curves. The boundary curves in his approach are restricted to lie in parallel planes; the projection of the boundary curves on the *x-y* plane must be a rectangle. In the work of Frey and Bindschadler [8], the results of Aumann are extended by generalizing the degree of the directions. Their system requires solving non-linear system equations to find the Bézier control points. Chu and Séquin [9] recently proposed a new method to design a developable Bézier patch. In their method, after one boundary curve is freely specified, five more degrees of free are available for a second boundary curve of the same degree. Chalfant and Maekawa [2, 6] presented a method to design developable B-spline surfaces where boundary curves do not necessarily lie in parallel planes. In the work of [10-12], the approximation methods are used to design developable B-Spline surfaces based on projective geometry. Other approaches are based on alternative perspective: Redont [13] constructs developable surfaces by specifying tangent planes along a geodesic of a surface, Randrup [14] approximates a given surface by cylinders in its Gaussian image, and Park et al. [15] design developable surfaces by the methods from optimal control theory.

All the above approaches try to utilize developable surfaces to construct the shape of a product. However, in practice, designers sometimes handle the problem in a reverse way. When they begin to design a product, they do not wish to have too many constraints; so they only utilize all kinds of modeling tools in the geometric modeling systems to design the product. After that, the final shapes of surfaces, which are planned to be manufactured by rolling a planar sheet, are modified to reduce stretches when rolling – the more developable the less stretches. Our approach automates and optimizes this modification process to increase the developability of a given surface – this is close to the design convention.

As mentioned above, trimmed NURBS surface patches are widely used in CAD/CAM systems. However, currently, there is no available method to make a trimmed NURBS patch developable in the literature. Also, the available methods that consider the whole B-spline surface patches cannot be directly applied on a trimmed patch since it is hard to reserve continuities on the boundary curves. Therefore, we develop a new method to increase the developability of a trimmed NURBS surface patch by adjusting the positions and weights of its control points. Our approach is function optimization based. When increasing the developability of the given surface patch, we also try to reduce the deformation of the original patch and avoid the singular points at the same time. G^0 continuity is reserved across the boundaries of the given trimmed NURBS surface patch. Our method presented in this paper can deal with an arbitrarily trimmed NURBS surface patch while current exiting approaches in literature can only handle the parametric patches with regular shape (3-sided or 4-sided).

We organize the paper as follows. In section 2, some necessary preliminary definitions and formulas are first reviewed. Next, the developability function and the objective function are defined in section 3 to minimize the overall Gaussian curvature, to reduce the change of the surface shape, and to maintain the surface to be regular. After that, the detailed numerical implementation to minimize the objective function is presented in section 4. In section 5, some experimental results are shown, and the modification method of our work to reserve G^1 continuity across patch boundaries is also discussed.

2. Preliminary

We first recall the following necessary definitions and formulas.

Definition 1 A NURBS surface of order k in the *u*- direction and order l in the *v*- direction is a vector-valued piecewise rational function of the form [16]

$$S(u,v)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,k}(u)N_{j,l}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,k}(u)N_{j,l}(v)w_{i,j}} \qquad 0\le u,v\le 1. \tag{1}$$

$\{P_{i,j}\}$ form a bi-directional control net, $\{w_{i,j}\}$ are the related weights of every control points, k and l are the order of the surface in the *u*- and *v*- directions, and $\{N_{i,k}(u)\}$ and $\{N_{j,l}(v)\}$ are the B-spline basis functions defined on the knot vectors: $U=\{u_0,u_1,\ldots,u_n,\ldots,u_{n+k}\}$ and $V=\{v_0,v_1,\ldots,v_m,\ldots,v_{m+l}\}$, where the definition of $N_{i,k}(u)$ is

$$\begin{aligned}
N_{i,1}(u)&=\begin{cases}1 & (u_i\le u<u_{i+1})\\ 0 & (otherwise)\end{cases}\\
N_{i,k}(u)&=\frac{u-u_i}{u_{i+k-1}-u_i}N_{i,k-1}(u)+\frac{u_{i+k}-u}{u_{i+k}-u_{i+1}}N_{i+1,k-1}(u)
\end{aligned}.$$

Definition 2 The Gaussian curvature of a parametric surface $S(u,v)$ at the point (u,v) is [17]

$$K=\frac{eg-f^2}{EG-F^2}, \tag{2}$$

where $e=N\cdot S_{uu}$, $f=N\cdot S_{uv}$, $g=N\cdot S_{vv}$, $E=S_u\cdot S_u$, $F=S_u\cdot S_v$, $G=S_v\cdot S_v$, and $N=\dfrac{S_u\times S_v}{\|S_u\times S_v\|}$.

To compute the Gaussian curvature at a point on a NURBS surface, we need the formulas of $S_u(u,v)$, $S_v(u,v)$, $S_{uu}(u,v)$, $S_{vv}(u,v)$, and $S_{uv}(u,v)$. Let

$$A(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u)N_{j,l}(v)w_{i,j}P_{i,j} \text{ and } w(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u)N_{j,l}(v)w_{i,j},$$

then,

$$S(u,v) = \frac{A(u,v)}{w(u,v)},\ S_u(u,v) = \frac{A_u(u,v) - w_u(u,v)S(u,v)}{w(u,v)},\ S_v(u,v) = \frac{A_v(u,v) - w_v(u,v)S(u,v)}{w(u,v)},$$

$$S_{uv}(u,v) = \frac{A_{uv}(u,v) - w_{uv}(u,v)S(u,v) - w_u(u,v)S_v(u,v) - w_v(u,v)S_u(u,v)}{w(u,v)},$$

$$S_{uu}(u,v) = \frac{A_{uu}(u,v) - 2w_u(u,v)S_u(u,v) - w_{uu}(u,v)S(u,v)}{w(u,v)},$$

$$S_{vv}(u,v) = \frac{A_{vv}(u,v) - 2w_v(u,v)S_v(u,v) - w_{vv}(u,v)S(u,v)}{w(u,v)},$$

where

$$A_u(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}^{(1)}(u)N_{j,l}(v)w_{i,j}P_{i,j},\ w_u(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}^{(1)}(u)N_{j,l}(v)w_{i,j},$$

$$A_v(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u)N_{j,l}^{(1)}(v)w_{i,j}P_{i,j},\ w_v(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u)N_{j,l}^{(1)}(v)w_{i,j},$$

$$A_{uu}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}^{(2)}(u)N_{j,l}(v)w_{i,j}P_{i,j},\ w_{uu}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}^{(2)}(u)N_{j,l}(v)w_{i,j},$$

$$A_{vv}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u)N_{j,l}^{(2)}(v)w_{i,j}P_{i,j},\ w_{vv}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}(u)N_{j,l}^{(2)}(v)w_{i,j},$$

$$A_{uv}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}^{(1)}(u)N_{j,l}^{(1)}(v)w_{i,j}P_{i,j},\ w_{uv}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,k}^{(1)}(u)N_{j,l}^{(1)}(v)w_{i,j}.$$

The formula to compute the derivatives of a B-spline basis function from [16] is

$$N_{i,k}^{(p)}(u) = (k-1)\left(\frac{N_{i,k-1}^{(p-1)}}{u_{i+k-1} - u_i} - \frac{N_{i+1,k-1}^{(p-1)}}{u_{i+k} - u_{i+1}}\right).$$

3. Objective Function

In this section, after defining the developability function of a parametric surface, we give the objective function of the optimization to increase the developability of a given surface. The objective function consists of a Gaussian term, a distance term, and a regularity term. Let us first recall the following theorem in differential geometry [17].

Theorem At regular points, the Gaussian curvature of a developable surface is identically *zero*.

By this theorem, we can detect whether a surface is developable or non-developable according to its overall Gaussian curvature. However, simply stating whether the surface is developable or non-developable is insufficient to identify the developability degree of a surface. Thus, we define the *developability function* of a parametric surface to describe it.

Definition 3 The developability function of a given parametric surface $S(u,v)$ is defined as

$$D[S(u,v)] = \frac{1}{A}\int_{\Omega} \delta(\kappa_1 \cdot \kappa_2) d\Omega \tag{3}$$

where $\delta(t)$ is the impulse function, Ω is the parametric domain of $S(u,v)$, A is the area of Ω, and $\kappa_1 \cdot \kappa_2$ is the Gaussian curvature at the point (u,v) on $S(u,v)$.

The value of the developability function gives a progressive representation of the developable property of a parametric surface. When $D[S(u,v)] = 1$, the Gaussian curvature of the whole surface is zero; in other words, $S(u,v)$ is developable. When $D[S(u,v)] = 0$, it means that we cannot find any point where the Gaussian curvature is zero on the surface – $S(u,v)$ is absolutely non-developable. If $D[S(u,v)] \in (0,1)$, there are some zero Gaussian curvature points on $S(u,v)$. The larger the value of $D[S(u,v)]$, the more developable is surface $S(u,v)$. When computing the developability function of a trimmed parametric surface S_T, Ω is the meaningful parametric domain P_{S_T} of S_T; and A is the area of P_{S_T}.

To increase the developability of a trimmed NURBS surface $S_T(u,v)$ (i.e., to increase $D[S_T(u,v)]$), we define the Gaussian term of the objective function as

$$J_G = \frac{1}{G_0}\int_{\Omega} (\kappa_1 \cdot \kappa_2)^2 d\Omega \tag{4}$$

where κ_1 and κ_2 are two principle curvatures, $\kappa_1 \cdot \kappa_2$ equals K in equation (2), Ω is the valid parametric region of the given surface $S_T(u,v)$, and G_0 equals $\int_{\Omega} (\kappa_1 \cdot \kappa_2)^2 d\Omega$ on the initially given surface. Since the Gaussian curvature at a point can be either positive or negative when it is non-zero, the quadric power is adopted in J_G to lead the optimization procedure to achieve the zero Guassian curvature at every point on $S_T(u,v)$. The G_0 in J_G alters the scale of J_G to be between 0 and 1.

When increasing the developability, the deformation between the initial surface and the optimized surface is to be controlled by the distance term of the objective function as

$$J_D = \frac{1}{D_0}\int_{\Omega} \|S_0 - S\|^2 d\Omega \tag{5}$$

where Ω has the same meaning as in J_G, and $D_0 = \int_{\Omega} {d_0}^2 d\Omega$ (d_0 is the tolerance of the distance error).

When optimizing the shape of $S_T(u,v)$, we must maintain the surface to be regular. This is because at the irregular points, $S_u // S_v$, which leads to $EG - F^2 = 0$. From the definition of K in equation (2), this creates the singularity on the objective function. Therefore, this should be prevented during the function optimization. To conduct a simpler form and to avoid irregular points, it is necessary to evade the condition $\frac{S_u}{\|S_u\|} \cdot \frac{S_v}{\|S_v\|} = 1$. Thus, the regularity term of the objective function is defined as

$$J_R = \int_{\Omega} \delta\left(1 - \frac{S_u}{\|S_u\|} \cdot \frac{S_v}{\|S_v\|}\right) d\Omega \tag{6}$$

In order that the term computed is a global minimum, the following approximation for $\delta(t)$ is usually adopted [18],

$$\delta_{\varepsilon}(t) = \frac{\varepsilon}{\pi(\varepsilon^2 + t^2)} . \tag{7}$$

As $\varepsilon \to 0$, the approximation converges to the theoretical $\delta(t)$. In our experimental examples, we usually use $\varepsilon = \frac{1}{\pi}$, which makes $\delta_{\varepsilon}(0) = 1$. The figure of $\delta_{\varepsilon}(t)$ with $\varepsilon = \frac{1}{\pi}$ is given below.

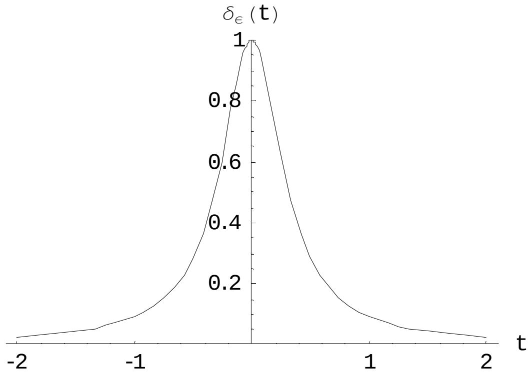


Fig. 2 Approximation of $\delta_{\varepsilon}(t)$

In summary, the final objective function of optimization is defined as

$$J = \frac{1}{G_0}\int_\Omega (\kappa_1 \cdot \kappa_2)^2 d\Omega + \frac{\lambda_1}{D_0}\int_\Omega \|S_0 - S\|^2 d\Omega + \lambda_2 \int_\Omega \delta\left(1 - \frac{S_u}{\|S_u\|} \cdot \frac{S_v}{\|S_v\|}\right) d\Omega \tag{8}$$

where λ_1 and λ_2 are the weighting factors to balance the relative importance of the three terms. In our testing examples, we usually use $\lambda_1 = 0.25$ and $\lambda_2 = 0.5$ so that the most important term is J_G, the second is J_R, and finally the least important one is J_D. Varying the value of λ_1 and λ_2 will effect the result of optimization. For example, if your designed object allows some distortion on the surface and you wants the optimized surface to be very close to the original design, you can change the balance among J_G, J_R, and J_D to let $\lambda_1 = 2.0$ and $\lambda_2 = 0.5$, so this time the most important term during optimization becomes J_D.

4. Numerical Implementation

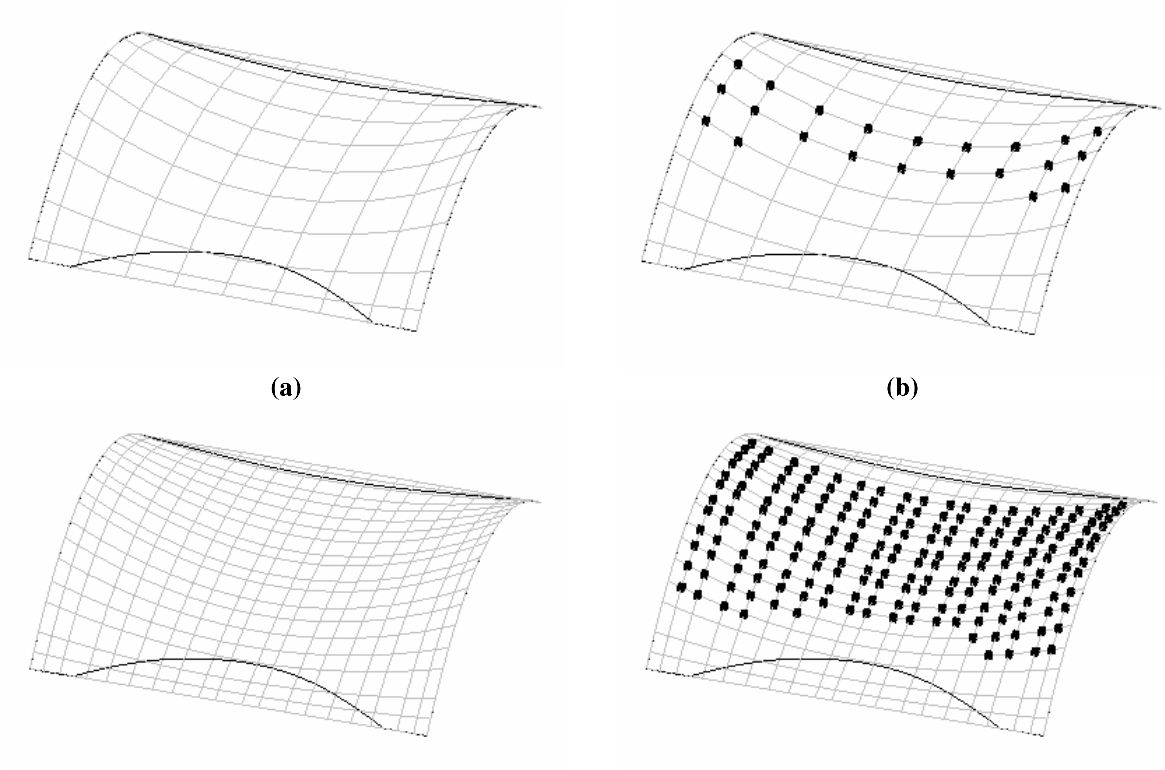
Our numerical implementation of minimizing the objective function adopts the conjugate gradient method. When applying this method to our optimization system, we face the following problems: 1) what are the free variables in the optimization system – in other words, we need to determine the control points and weight factors that can be adjusted to minimize the objective function; 2) how to compute the integration in the objective function; 3) how to compute the gradients of the free variables; 4) what is the terminal condition during the iteration. The methods for solving the four problems are detailed as follows.

Free variables

Our approach minimizes the objective function by adjusting the control points and weights of the given trimmed NURBS patch S_T. However, during this process, in order to maintain G^0 continuity on boundary curves, the position of the points on the boundary curves $b_i(t) = [u_i(t) \quad v_i(t)]$ of S_T must not be changed. Therefore, only part of the control points and its weight can be modified. They are the free variables in our function optimization.

Proposition 1 In order to achieve G^0 continuity across the surface boundaries, if the point $b_B(t_0)$ is on the boundary of S_T (the two components of $b_B(t_0)$ are $u_B(t_0) \in [u_i, u_{i+1})$ and $v_B(t_0) \in [v_i, v_{i+1})$), the related control points $P_{\xi,\eta}$ must *not* be moved and the weights $w_{\xi,\eta}$ must *not* be changed; where $\xi = (i-k+1), \cdots, i$ and $\eta = (j-l+1), \cdots, j$, k is the order of S_T in the u direction, and l is the order of S_T in the v direction.

From the property of NURBS surfaces, we know that if $P_{\xi,\eta}$ is moved or $w_{\xi,\eta}$ is changed, only the region $[u_\xi, u_{\xi+k}) \times [v_\eta, v_{\eta+l})$ on the surface will be affected [16]. In other words, for any control point $P_{\xi,\eta}$ or its weight $w_{\xi,\eta}$ where $\xi = (i-k+1), \cdots, i$ and $\eta = (j-l+1), \cdots, j$, this change will alter the shape of the surface in the region $[u_i, u_{i+1}) \times [v_j, v_{j+1})$; so the position of point $b_B(t_0)$ is changed. In order to reserve G^0 continuity across the surface boundaries, the position of any point on $b_i(t)$ must not be changed. Therefore, the position of $P_{\xi,\eta}$ and the value of $w_{\xi,\eta}$ should be fixed during the function optimization.



(a) **(b)**

(c) **(d)**

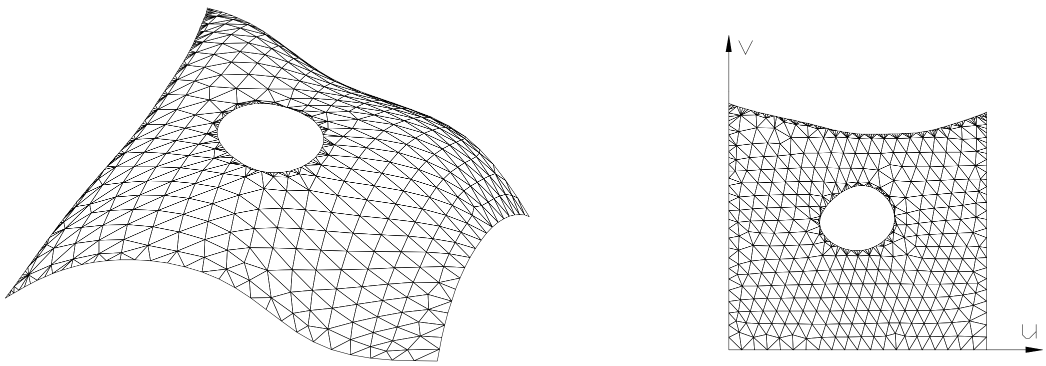
Fig. 3 Free control points before and after refinement

By Proposition 1, we obtain the algorithm to determine the fixed control points and weights on S_T by a recursive search along each boundary curve of S_T. The pseudo-code of the algorithm is given in Table 1. Fig. 3a shows an example trimmed NURBS patch (the grey lines are the control net of S_T); the free control points of S_T are illustrated by solid cubes in Fig. 3b. If more freeform variables are desired, the knot insertion technique of NURBS [16] can be applied to insert knots in the middle of the parameter intervals – so S_T is represented by more control points and weight factors. Fig. 3c shows the control points of a patch in Fig. 3a after twice subjecting the knot vectors to uniform refinement; and Fig. 3d gives the free control points after refinement.

Table 1 The pseudo-code of ***Algorithm*** DetermineStaticControlPointsWeights(S_T)

Algorithm DetermineStaticControlPointsWeights(S_T)
Input: A trimmed NURBS surface patch S_T with n boundary curves $b_i(t)$
Output: The control points and weights cannot be changed
1. **for** $i = 0$ **to** $i < n$ {
2. **Call** RecursiveDetermination(S_T , $b_i(t)$, 0, 1);
3. } **return**;

Function RecursiveDetermination(S_T , $b(t)$, t_0 , t_1)
Input: A trimmed patch S_T , one boundary curve $b(t) = [u(t) \quad v(t)]$, and two end parameters t_0 and t_1
Output: The static control points and weights determined by $b(t)$
1. Determine the fixed control points and weights by $b(t_0)$ according to Proposition 1;
2. Determine the fixed control points and weights by $b(t_1)$ according to Proposition 1;
3. **if** $b(t_0)$ and $b(t_1)$ are in the same interval in both u- and v- directions, **then return;**
4. $t_m \leftarrow \frac{1}{2}(t_0 + t_1)$;
4. Call RecursiveDetermination(S_T , $b(t)$, t_0 , t_m);
5. Call RecursiveDetermination(S_T , $b(t)$, t_m , t_1);
6. **return**;



(a) in spatial domain **(b) in parametric domain**

Fig. 4 Tessellation of trimmed NURBS surface patch S_T

Numerical integration

During the function optimization, the developability and the objective function are evaluated on the meaningful integral of the trimmed NURBS patch. Since the shape of the meaningful integral is irregular, it is impossible to compute the analytical integral automatically. Thus, we carry out the numerical integration to evaluate the objective function. Firstly, the tessellation of the surface S_T is generated in the parametric domain and has its quality enhanced in the spatial domain by the method in [19]. The continuous domain P_{S_T} is tessellated into m small triangles (e.g., Fig. 4b shows the tessellation of P_{S_T} in Fig. 1b), where each triangle T_i has three nodes $n_1 = (u_1, v_1)$, $n_2 = (u_2, v_2)$, and $n_3 = (u_3, v_3)$. After assuming the function value of any point, $f_{T_i}(u, v)$ in T_i is a linear function of its coordinate, it can be computed by [20]

$$\left.\begin{aligned} f_{T_i}(u,v) &= \sum N_k f_{T_i k} \\ N_k &= \frac{1}{2A_{T_i}}(a_k + b_k u + c_k v) \end{aligned}\right\} \quad k = 1, 2, 3 \tag{9}$$

where

$$\begin{array}{lll} a_1 = u_2 v_3 - u_3 v_2 & b_1 = v_2 - v_3 & c_1 = u_3 - u_2 \\ a_2 = u_3 v_1 - u_1 v_3 & b_2 = v_3 - v_1 & c_2 = u_1 - u_3 \\ a_3 = u_1 v_2 - u_2 v_1 & b_3 = v_1 - v_2 & c_3 = u_2 - u_1 \end{array}$$

$f_{T_i 1}$, $f_{T_i 2}$, and $f_{T_i 3}$ are the function values on the three nodes of T_i, (u,v) is the coordinate of any point in the triangle, and A_{T_i} is the area of the triangle. The integration of function $f(u,v)$ in P_{S_T} is approximated by

$$\int_\Omega f(u,v)d\Omega \approx \sum_{i=1}^{m} \int_{T_i} f_{T_i}(u,v)dudv\,. \tag{10}$$

By the coordinate of the three nodes in T_i and equation (9), we get $\int_{T_i} f_{T_i}(u,v)dudv = \frac{A_{T_i}}{3}(f_{T_i 1} + f_{T_i 2} + f_{T_i 3})$.

Thus, the integration of any function $f(u,v)$ in the P_{S_T} region can be computed by

$$\sum_{i=1}^{m} \frac{A_{T_i}}{3}(f_{T_i 1} + f_{T_i 2} + f_{T_i 3})\,. \tag{11}$$

Numerical gradient

It is necessary to compute the gradient of the objective function with respect to all the free variables during the optimization process. In our optimization to increase the developability of a trimmed NURBS patch, the objective function is of such complexity that it is impractical to compute the gradient analytically. Instead, we compute the central differences [21] to approximate the partial derivatives. The standard central difference formula for computing the derivative of $f(t)$ with respect to t is

$$f'(t) \approx \frac{f(t+h) - f(t-h)}{2h} \tag{12}$$

with $E(f) = -\frac{h^2}{6} f'''(\eta)$, the approximation error where $|\eta - t| < |h|$. The optimum value of h is defined as the value for which the sum of the magnitudes of the round-off error and of the discretization error is minimized. Since the value of the objective function in equation (8) is around 1.0, when using single precision computing (10^{-8}), the round-off error R is approximately

$$R = \pm \frac{2 \times 10^{-8}}{2h}\,;$$

the discretization error T is approximately

$$T = -\frac{1}{6}h^2 .$$

To find the optimum h we must minimize

$$g(h) = |R| + |T| = \frac{10^{-8}}{h} + \frac{1}{6}h^2$$

By computing the positive root of $g'(h) = 0$, we have

$$h = \sqrt[3]{3\times 10^{-8}}$$

This is the optimum value of h found in our approach.

Terminal conditions

During the iteration of function optimization, the value of the objective function decreases while the step number of iteration increases. Usually, these two factors are utilized to give the terminal condition of the iteration. Here, we employ a combination of them. Either $\frac{J_i}{J_0} < \varepsilon$ or the iteration steps is greater than $N_{\max}$, the iteration stops, where J_i is the value of the objective function in the *i*th iteration (current value), J_0 is the value of the objective function before optimization, $N_{\max}$ is the maximum iteration number, and ε is a small number (we usually choose $\varepsilon = 10\%$ or $\varepsilon = 15\%$ in our testing examples). In practice, the iteration usually stops when the maximum iteration steps condition is reached. Whether we can arrive the $\frac{J_i}{J_0} < \varepsilon$ terminal condition of the developability optimization does really depends on the number of variables, when the number of free variables is not enough, the problem cannot be fully optimized – we stop at $N_{\max}$. Also, when the termination of iteration depends on $N_{\max}$, the results are improved when $N_{\max}$ increased.

5. Experimental Results and Discussion

Fig. 5 gives the function optimization result to increase the developability of a trimmed NURBS patch with order 4 – example I. Fig. 5a gives the control network and the boundary curves of the given trimmed patch, Fig. 5b shows the grid of its knot vectors,

$$U = \{0,0,0,0,1,1,1,1\} \text{ and } V = \{0,0,0,0,1,1,1,1\},$$

in the parametric domain, Fig. 5c is the Gaussian curvature map of the given trimmed surface, and Fig. 5d gives the side view of its initial shape. By the ***Algorithm*** DetermineStaticControlPointsWeights(S_T), no control point and its weight factor can be changed in the initial given surface of example I. Thus, we uniformly insert knots to increase the free variables of the given trimmed surface patch. When converting the knot vectors to

$$U = \{0,0,0,0,\frac{1}{8},\frac{1}{4},\frac{3}{8},\frac{1}{2},\frac{5}{8},\frac{3}{4},\frac{7}{8},1,1,1,1\} \text{ and } V = \{0,0,0,0,\frac{1}{8},\frac{1}{4},\frac{3}{8},\frac{1}{2},\frac{5}{8},\frac{3}{4},\frac{7}{8},1,1,1,1\},$$

the new control network becomes denser as shown in Fig. 5e; Fig. 5f gives its related grids of knot vectors. Applying the function optimization approach to this dense control network (with $N_{\max} = 100$, $\varepsilon = 10\%$ and $d_0 = 1\text{mm}$), the optimized surface is computed iteratively. Fig. 5g shows the Guassian curvature map of the result surface after 100 iterations, and the side view shape is in Fig. 5h. The computation statistics is given in Table 2. With 88 free variables (22 control points and 22 weight factors), the computing time is about 3 hours. However, in the u direction of the trimmed surface, not that many control points are required to increase the developability. As mentioned in [22], choosing the appropriate variables can greatly increase the convergent speed of the optimization system. After adjusting the knot vectors to

$$U = \{0,0,0,0,\frac{1}{2},1,1,1,1\} \text{ and } V = \{0,0,0,0,\frac{1}{5},\frac{1}{4},\frac{3}{10},\frac{4}{5},\frac{17}{20},\frac{9}{10},1,1,1,1\},$$

we have 36 variables (9 control points and 9 weight factors) to be optimized. Fig. 6a and 6b shows the related control network and the grids of the knot vectors in the $u-v$ domain. Since the number of variables has decreased, the computing time decreased from about 3 hours to 1 hour 37 minutes (see Table 2). At the same time, the developability after optimization increased from 0.978 to 0.992 by conducting the same iteration terminal condition – $N_{\max} = 100$ and $\varepsilon = 10\%$. The Gaussian curvature map and the side view of the result surface patch are given in Fig. 6c and 6d.

The surface patches in example II is from a practical product (Fig. 7a), which is to be manufactured by rolling and welding metal sheets. The Gaussian curvature maps of two patches from the shape of the initial design are shown in Fig. 7b and 7c. After increasing their developability by our approach, the values of their developability function increase from 0.765 to 0.810 and from 0.741 to 0.981 respectively. The Gaussian curvature maps of the optimized patches with refined control networks (the refined patch A has 432 variables, and the refined patch B has 564 variables) are given in Fig. 7d and 7e. When placing the result patches together (Fig. 7f), it is easy to find that the two shared curve edges on the given patches are still coincident – G^0 continuity are reserved.

Our approach has the ability to reserve G^1 continuity after extension. If the cross tangent on a boundary curve $b_{\tan}(t)$ has to be reserved during the optimization of the given patch S_T, we can add an offset curve with a small distance to $b_{\tan}(t) - b_{offset}(t)$ on the inner surface of $b_{\tan}$ in P_{S_T} when determining the free variables by the ***Algorithm*** DetermineStaticControlPointsWeights(S_T). We call this modification G^1 *boundary extension*. After that, the same optimization approach presented in this paper will reserve the cross tangent on $b_{\tan}(t)$ of S_T. Fig. 8 demonstrates this. In Fig. 8a, two patches are connected with G^1 continuity; their related Gaussian curvature maps are shown in Fig. 8b and 8c. Without the G^1 boundary modification on the given patches, the G^0 reserved optimization results (with 8 variables) are shown in Fig. 8d-8f – the iteration stops at the $\varepsilon = 10\%$ terminal condition. After adding $b_{offset}(t) = [0.95 \quad t]$ on patch A and $b_{offset}(t) = [0.05 \quad t]$ on patch B, their control networks are refined to have enough degree of free (12 variables) during the optimization. The final optimization result with G^1 continuity reserved is shown in Fig. 8g-8i. The computation statistics of G^0 and G^1 reserved optimizations are also shown in Table 2. The G^0 reserved optimization takes only 4 seconds; this is because that the iteration stops at the 2nd step by the $\varepsilon = 10\%$ terminal condition. From the Gaussian maps (Fig. 8h and 8i), it is easy to find that although the overall developability of the given surfaces has been increased, the Gaussian curvature may increase at some places – this happens when the number of free variables are not enough. Introducing more variables near the places with a high Gaussian curvature will be helpful to solve this problem. Both the G^0 and G^1 reserved optimizations have the opportunity to increase the Gaussian curvature at some location. In this example, the number of free variables is enough for the G^0 reserved optimization, but not enough for G^1 reserved optimization, so the Gaussian curvature increases locally in the G^1 reserved tests only.

Table 2 Computation statistics of the examples

Example	Figure	Developability - $D[S(u,v)]$		Tessellation		Free Variable No.	Computing Time
		Before optimization	After optimization	Node No.	Triangle No.		
I	5	0.940	0.978	808	1431	88	2 hr. 49 min.
	6	0.940	0.992	808	1431	36	1 hr. 37 min.
II	7b	0.765	0.810	864	1486	432	13 hr. 21 min.
	7c	0.741	0.981	819	1454	564	18 hr. 37 min.
III - G^0	8b, 8e	0.775	0.974	506	930	8	4 sec.
	8c, 8f	0.776	0.974	506	930	8	4 sec.
III - G^1	8b, 8h	0.775	0.894	506	930	12	10 min.
	8c, 8i	0.776	0.892	506	930	12	10 min.

* All with $N_{\max} = 100$ and $\varepsilon = 10\%$ on a PIII 500 PC with a program written in C++.

6. Conclusion

In this paper, we present an optimization method to increase the developability of an arbitrarily trimmed NURBS surface patch. It is a useful tool for designing products manufactured from planar sheets. With this tool, designers can firstly create and modify the shape of a product without considering the developable constraint. After a design is finished, our approach is applied to increase the developability. Different from other approaches, our method optimizes a trimmed NURBS surface patch, which is now almost a 3D geometry representation standard in CAD/CAM systems. Therefore, our method can be integrated into any current popular commercial geometric modeling system to benefit designers. In our method, the developability of trimmed NURBS surface patches are increased by adjusting the positions and weights of their control points during the optimization approach. When increasing the developability of a given surface patch, we also try to minimize the deformation of the given patch and avoid the singular points at the same time. Our approach reserves G^0 continuity on the boundaries of given patches, and also can be further modified to reserve G^1 continuity across a boundary curve. The initial testing results are encouraging, after optimization, the developability of all testing surface patches are significantly improved. Our approach is new, which can optimize arbitrary trimmed NURBS patch while others only deal with the whole parametric patches with regular shape. Our method gives a solution that is more close to the design convention.

The major disadvantage of the developability increasing approach is the computing time. As shown in Table 2, the processing time ranges from several minutes to several hours. In the current implementation, we conduct primitive numerical methods to compute the function optimization. It is believed that with more efficient optimization algorithms and with the increasing processing power available on the desktop, the running time can be shortened significantly. Moreover, the following topics or improvements are worth conducting future research:

- As shown in example I, the optimally refined control network can greatly improve the convergent speed. It is possible to design an algorithm to automatically refine the control network. The basic idea is that more control points are required near the boundary curves and places with a relative high Gaussian curvature.

- From the Gaussian maps of our testing examples, we can find that the overall developability of the given surfaces has been increased after optimization, however, the Gaussian curvature may increase locally at some places, this happens in both the G^0 and G^1 reserved optimizations when the number of variables are not enough. This comes out the research about how to automatically refine the control network of the given surface to provide enough variables for the optimization.

- In our approach, we do not address the problem of tolerance, i.e., how good is good enough? For example, the result of example II improves developability from 0.765 to 0.810, however, we cannot say that 0.810 is good or not. The answer is application dependent and material dependent. When using steel, no amount of stretching is acceptable if the object is to be made purely by rolling or pressing, but there will be some tolerance due to the thickness of the steel. Therefore, the tolerance problem is worth to be further studied.
- Also, it will be interesting to see if the idea proposed in this paper can be extended onto the tessellated surface; in other words, using the optimization approach to increase the developability of triangular mesh surfaces. Then, the optimization approach will be more flexible to increase local developability – the number of free variables equals to 3 times the number of inner triangular nodes.

References

[1] Mortenson, M. E. (1997) Geometric Modeling (2nd Edition). New York, Wiley.

[2] Chalfant, J. S.; Maekawa, T. (1998) Design for manufacturing using B-spline developable surfaces. Journal of Ship Research, vol.42, no.3, pp.207-215.

[3] Azariadis, P. N.; Aspragathos, N. A. (2001) Geodesic curvature preservation in surface flattening through constrained global optimization. Computer-Aided Design, vol. 33, no.8, pp. 581-591.

[4] McCartney, J.; Hinds, B. K.; Seow, B. L. (1999) The flattening of triangulated surfaces incorporating darts and gussets, Computer-Aided Design, vol.31, no.4, pp.249-260.

[5] Wang, C. C. L.; Smith, S. S. F.; Yuen, M. M. F.; (2002) Surface flattening based on energy model. Computer-Aided Design, vol.34, no.11, pp.823-833.

[6] Maekawa, T.; Chalfant, J. (1998) Design and tessellation of B-spline developable surfaces. ASME Transaction Journal of Mechanical Design, vol.120, pp.453-461.

[7] Aumann, G. (1991) Interpolation with developable Bézier patches. Computer Aided Geometric Design, vol.8, pp.409-420.

[8] Frey, W. H.; Bindschadler D. (1993) Computer aided design of a class of developable Bézier surfaces. GM Research Publication 1993; GMR-8057.

[9] Chu, C. H.; Séquin C. H. (2002) Developable Bézier patches: properties and design. Computer-Aided Design, vol.34, no.7, pp.511-527.

[10] Hoschek, J.; Pottmann, H. (1995) Interpolation and approximation with developable B-spline surfaces. Mathematical Methods in CAGD III, M. Daehlen, T. Lyche, and L. Schumaker, eds., Vanderbilt University Press.

[11] Chen, H. Y.; Lee, I. K; Leopoldseder, S.; Pottmann, H.; Randrup, T.; Wallner, J. (1999) On surface approximation using developable surfaces. Graphical Models & Image Processing, vol.61, no.2, pp.110-124.

[12] Pottmann, H.; Wallner, J. (1999) Approximation algorithms for developable surfaces. Computer Aided Geometric Design, vol.16, no.6, pp.539-556.

[13] Redont, P. (1989) Representation and deformation of developable surfaces. Computer-Aided Design, vol.21, no.1, pp13-20.

[14] Randrup, T. (1998) Approximation of surfaces by cylinders. Computer-Aided Design, vol.30, no.10, pp.807-812.

[15] Park, F. C.; Yu, J.; Chun, C. (2002) Design of developable surfaces using optimal control. ASME Transaction Journal of Mechanical Design, vol.124, pp.602-608.

[16] Piegl L.; Tiller W. (1997) The NURBS Book (2nd ed). Berlin; Hong Kong: Springer.

[17] do Carmo M. P. (1976) Differential Geometry of Curves and Surfaces. Englewood Cliffs, N.J.: Prentice-Hall.

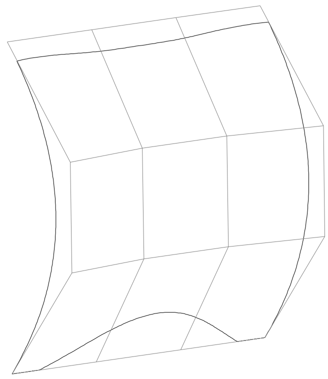
[18] Chan T. F.; Vese L. A. (2001) Active contours without edges. IEEE Transactions of Image Processing, vol.10, pp.266-277.

[19] Peiró, J. (1999) Surface grid generation. Handbook of grid generation, J.F. Thompson, B.K. Soni, and N.P. Weatherill, eds., Boca Raton, FL: CRC Press.

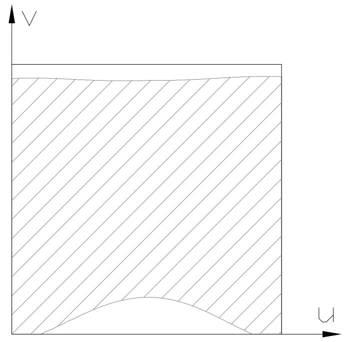
[20] Lewis, P. E.; Ward, J. P. (1991) The finite element method: principles and applications. Reading, Mass.: Addison-Wesley.

[21] Conte, S. D.; de Boor, C. (1980) Elementary numerical analysis: an algorithm approach, McGraw-Hill, New York, N.Y.

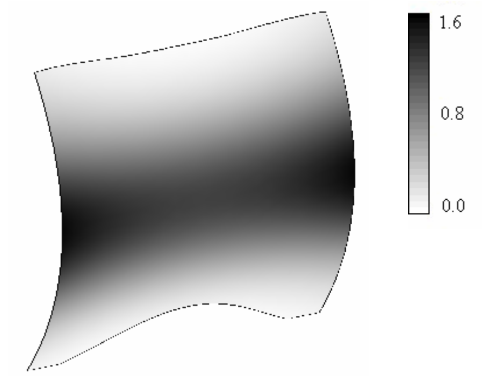
[22] Vanderplaats, G. N. (1984) Numerical optimization techniques for engineering design: with applications. McGraw-Hill, New York, N.Y.



(a) initial control network with boundary curves

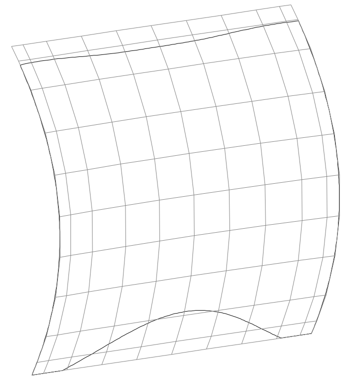


(b) grids of knot vectors of initial control network in the $u-v$ domain

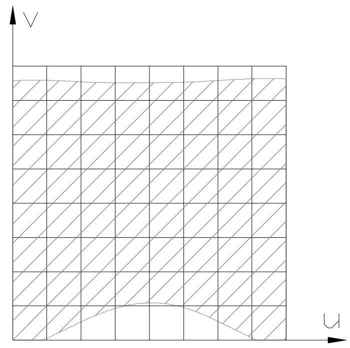


(c) Guassian curvature map+ of initial surface

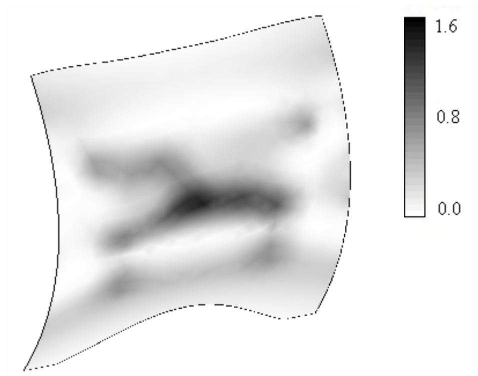
(d) side view of initial shape



(e) uniform refined control network with boundary curves



(f) grids of knot vectors of initial control network in the $u-v$ domain

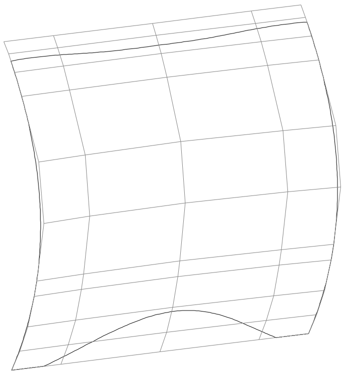


(g) Guassian curvature map of optimized surface

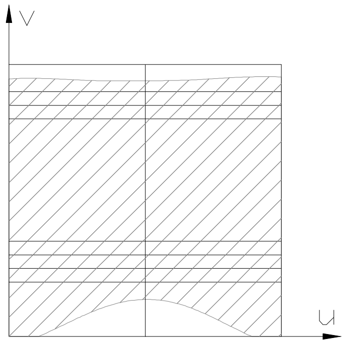
(h) side view of optimized surface

Fig. 5 Example I – before and after optimization (d_0 = 1mm)

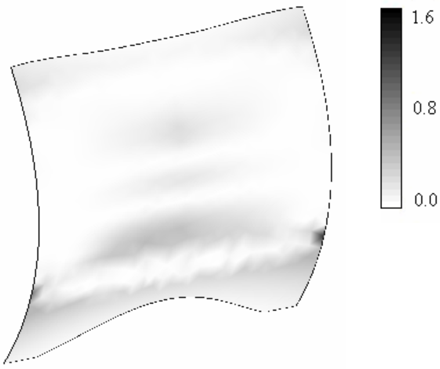
+ In a Gaussian curvature map, the grey-scales represent the value of K^2 at every point on the surface.



(a) appropriately refined control network with boundary curves



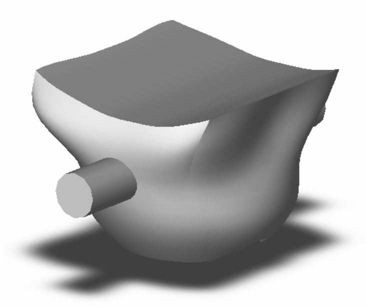
(b) grids of knot vectors of appropriately refined control network in the $u-v$ domain



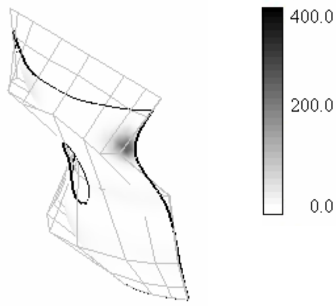
(c) Guassian curvature map of optimized surface

(d) side view of optimized surface

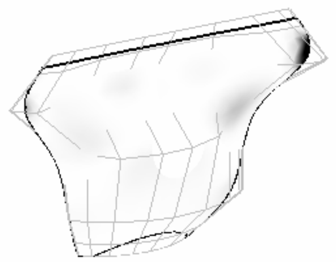
Fig. 6 Example I (continue) – choosing appropriate variables can greatly improve the convergent speed



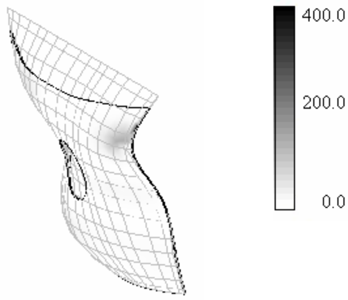
(a) given model



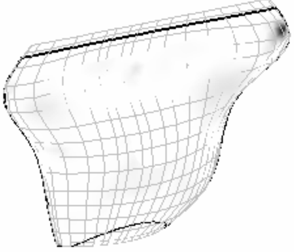
(b) initial Gaussian curvature map of patch A



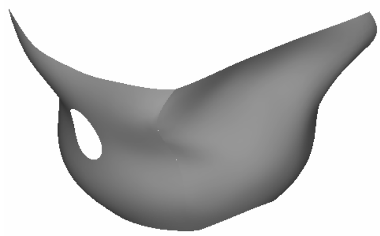
(c) initial Gaussian curvature map of patch B



(d) Gaussian curvature map of patch A after optimization

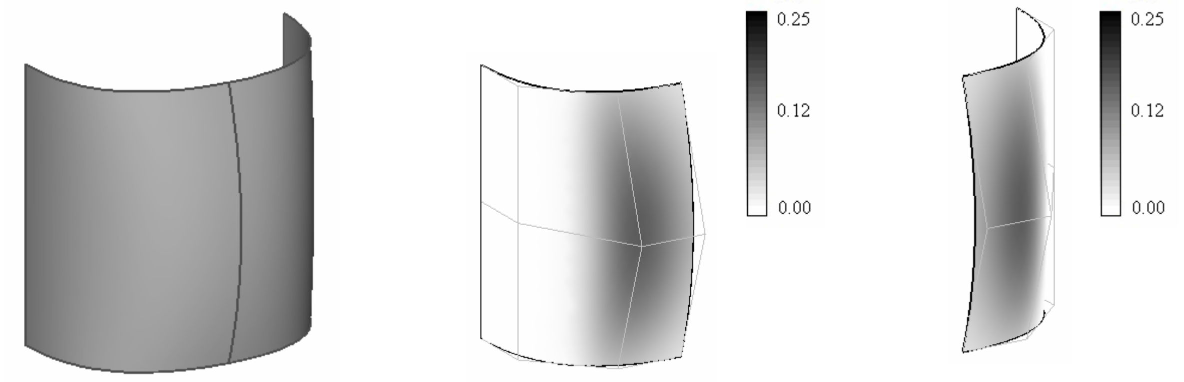


(e) Gaussian curvature map of patch B after optimization

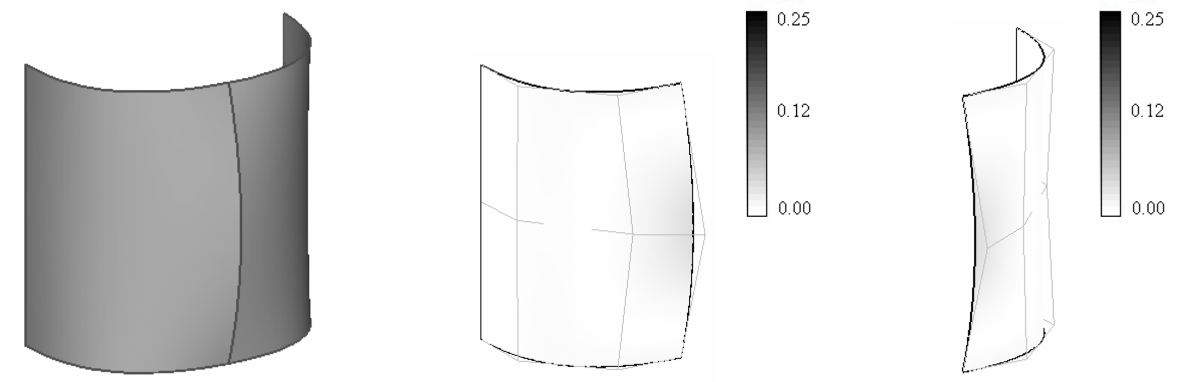


(f) assembly of optimized patch A and B

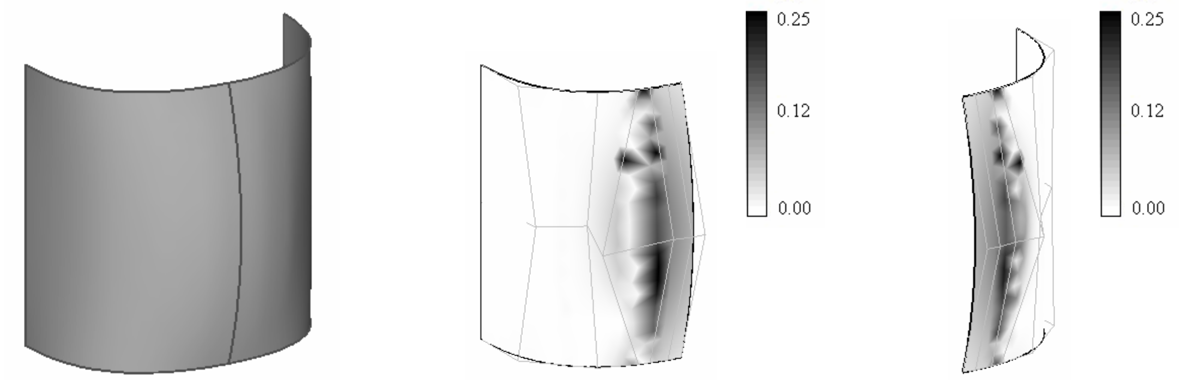
Fig. 7 Example II – practical trimmed NURBS surface patches (d_0 = 5mm)



(a) initially given patches **(b) Gaussian curvature map of patch A in (a)** **(c) Gaussian curvature map of patch B in (a)**



(d) G^0 **reserved optimization result** **(e) Gaussian curvature map of patch A in (d)** **(f) Gaussian curvature map of patch B in (d)**



(g) G^1 **reserved optimization result** **(h) Gaussian curvature map of patch A in (g)** **(i) Gaussian curvature map of patch B in (g)**

Fig. 8 Example III – G^0 **vs.** G^1 **reserved optimization (** d_0 **= 1mm)**