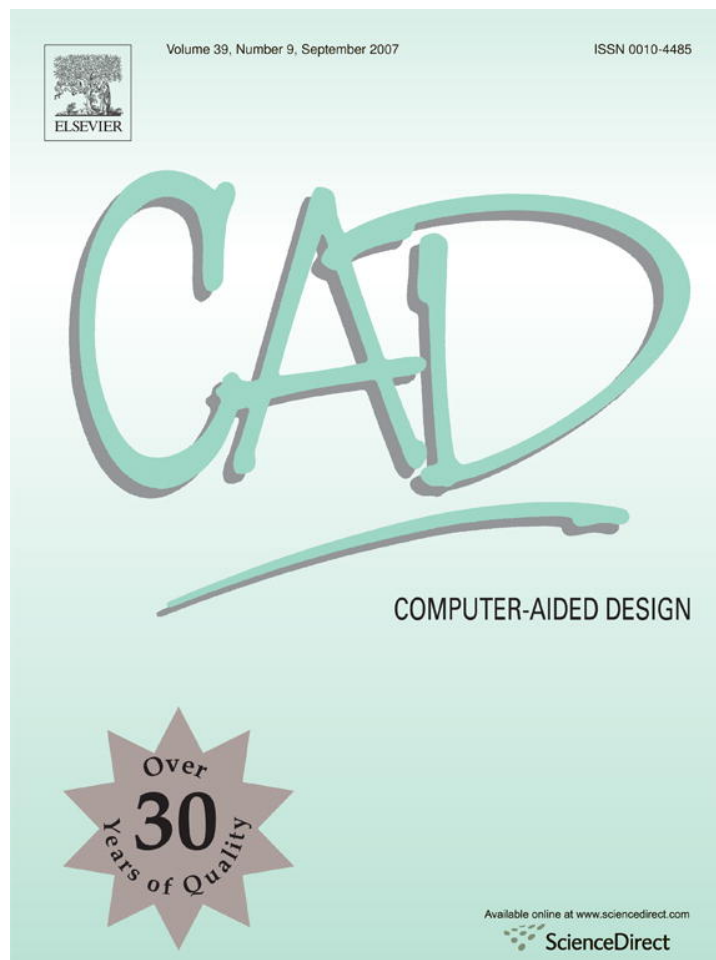


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article was published in an Elsevier journal. The attached copy is furnished to the author for non-commercial research and education use, including for instruction at the author's institution, sharing with colleagues and providing to institution administration.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



ELSEVIER

Available online at www.sciencedirect.com

Computer-Aided Design 39 (2007) 709–718

COMPUTER-AIDED
DESIGNwww.elsevier.com/locate/cad

Generating strictly non-self-overlapping structured quadrilateral grids

Hongwei Lin^{a,*}, Kai Tang^b, Ajay Joneja^c, Hujun Bao^a

^a State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, 310027, China

^b Department of Mechanical Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, NT, Hong Kong, China

^c Department of Industrial Engineering and Engineering Management, The Hong Kong University of Science and Technology, Clear Water Bay, NT, Hong Kong, China

Received 2 August 2005; accepted 4 February 2007

Abstract

In this paper, we present a BPM (Bézier patch mapping) algorithm which generates a strictly non-self-overlapping structured quadrilateral grid in a given four-sided planar region. Given four pieces of polynomial curves which enclose a simple region in the plane, the algorithm first constructs a Bézier patch which interpolates the four curves (as its four boundary curves), while the inner control points of its control grid remain unknown. In this paper, we show that, for the bijective condition to be satisfied, it is sufficient that the interior points satisfy a set of quadratic inequality equations. Exploiting this key result, we formulate the mapping algorithm as an optimization problem where the constraints are the bijective condition of the Bézier patch mapping (BPM), and the objective is to find out the best from all of the non-self-overlapping grids. Thus, commercial optimization solvers can be used to find the bijective mapping. If a solution to the optimization problems exists, then so does a solution to the mapping problem, and vice-versa. The BPM method is simple and intuitive, and some examples presented in this paper demonstrate its effectiveness.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Structured quadrilateral grid; Four-sided region; Non-self-overlapping; Boundary-conforming mapping

1. Introduction

Structured quadrilateral grid is often used in the finite element method and the finite difference method, for it usually generates smaller analysis errors and provides a structure with which one can easily calculate the derivative employed in the finite difference method. When the finite element method or the finite difference method works on a four-sided planar region, the region should be first mapped into a unit square to get structured quadrilateral grid on the four-sided region. Additionally, in computer aided design, designers often use surface trimming operation to obtain four-sided trimming surfaces satisfying some requirements. A four-sided trimming surface corresponds to a four-sided region in its 2D parameter field. The 2D four-sided parameter region also needs to be reparameterized to a unit square for further process of the trimming surface. This is equivalent to generating a structured quadrilateral grid in the 2D four-sided parameter region.

Currently, there are mainly two classes of methods for generating structured quadrilateral grids in a planar four-sided region, the partial differential equation (PDE) method [1–5], and the boundary-conforming mapping (BCM) method [6–9]. The PDE method generates a grid by first distributing points on the boundary curves and then solving elliptic PDEs in the field. It requires delicate numerical solutions and usually runs very slow; moreover, the PDE method can only give a discrete grid, for only numerical methods can be used to solve elliptical PDEs and solutions only at discrete points in the four-sided region can be obtained. The BCM method requires no algorithmic computation like the PDE method, thus it is fast and free of any numerical instability problem. Basically, a BCM is a continuous mapping from a square to a planar four-sided region R . By sampling the square into a rectangular grid and then mapping it into R , one obtains a quadrilateral grid of R .

However, although simple and robust, BCM methods suffer from a common deficiency — it is difficult to ensure that the generated grid is free of self-overlapping. Self-overlapping means some quadrilateral elements in the grid may overlap with each other, which makes the grid useless to downstream

* Corresponding author. Tel.: +86 571 88206681 518; fax: +86 571 88206680.

E-mail address: hwlin@cad.zju.edu.cn (H. Lin).

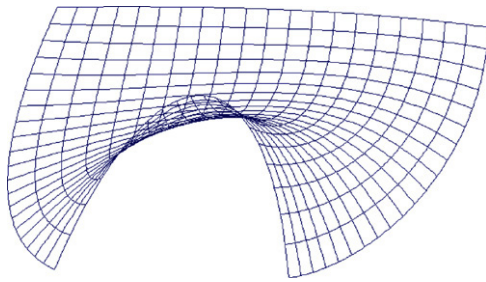


Fig. 1. Self-overlapping structured quadrilateral grid. Self-overlapping means some quadrilateral elements in the grid overlap with each other.

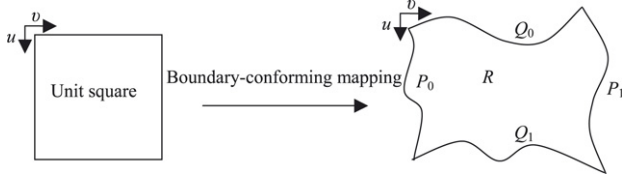


Fig. 2. A boundary-conforming mapping maps from a unit square to a 4-sided region R .

applications (see Fig. 1). Generating non-self-overlapping structured quadrilateral grid by a BCM method is equivalent to constructing a continuous bijective mapping. But checking whether a continuous mapping is bijective is difficult because it requires the solution of a nonlinear equation, which can be solved only by numerical methods in general.

Additionally, a more complicated problem is grid generation in an n -sided planar region [10]. It can be solved either by segmenting the n -sided planar region into a series of four-sided regions, and generating grid on each four-sided region; or, by directly constructing an n -sided continuous mapping on the n -sided region, such as Gregory patch mapping. However, the same problem, namely, how to check a continuous mapping is bijective, still remains.

In this paper, we develop a bijective boundary-conforming mapping method, that is, Bézier patch mapping (BPM) method, to generate strictly non-self-overlapping grid in a planar four-sided region R , whose boundaries are polynomial curves. First, the four boundary polynomial curves are converted into Bézier curves; their four control polygons connect sequentially, forming the boundary of a Bézier patch. Second, by setting the initial values of the inner control points, we get an initial Bézier patch, which can be considered as a mapping from the unit square to the four-sided region R . Third, by solving a optimization problem with constraints, which takes the inner control points as unknowns, the inner control points can be adjusted to the optimal state, leading to the optimal non-self-overlapping Bézier patch mapping (BPM). Here, the constraints of the optimization problem are the bijective conditions of BPM, which ensure that the generated grid is non-self-overlapping; furthermore, the objective function improves the quality of the generated grid. Thus, not only is the non-self-overlapping property of the mapping guaranteed, but also the overall quality of the generated grid is maximized simultaneously.

This paper is arranged as follows. In Section 2, we describe the overview of the BPM method and the constraints on the given four boundary curves. In Section 3, the bijective conditions of BPM are deduced and the initial inner control points are constructed. In Section 4, the problem on generating and improving the structured quadrilateral grid is formulated as an optimization problem with constraints. Some examples are presented in Section 5 and Section 6 concludes the paper.

2. Overview of BPM method and constraints on four boundary curves

2.1. Overview of BPM method

As illustrated in Fig. 2, four pieces of planar curves which connect sequentially form a simple and closed curve, thus enclosing a four-sided planar region R . One pair of opposite curves, P_0 and P_1 , has the same degree m , and the other pair of curves, Q_0 and Q_1 , has the degree n :

$$\begin{aligned}
 P_0: & \begin{cases} x = f_0(t) \\ y = g_0(t) \end{cases}, & t \in [a_0, b_0]; \\
 P_1: & \begin{cases} x = f_1(t) \\ y = g_1(t) \end{cases}, & t \in [a_1, b_1]; \\
 Q_0: & \begin{cases} x = e_0(t) \\ y = h_0(t) \end{cases}, & t \in [c_0, d_0]; \\
 Q_1: & \begin{cases} x = e_1(t) \\ y = h_1(t) \end{cases}, & t \in [c_1, d_1].
 \end{aligned} \tag{1}$$

The boundary-conforming mapping (BCM) method seeks a continuous map from the unit square to the four-sided planar region R , which generates a structured quadrilateral grid over the region R , by sampling the unit square into a rectangular grid and then mapping it into R . Conventionally, the BCM method calculates a Coons patch interpolating the four given curves P_0 , P_1 , Q_0 and Q_1 by transfinite interpolation [11,12], and the parameter grid of the Coons patch is taken as the structured grid over R . The bijectiveness of the Coons patch mapping is equivalent to the regularity of the Coons patch, which ensures that there is no self-overlapping in the generated structured grid. However, as stated above, it is impossible to determine the regularity of the Coons patch theoretically when its degree is even modestly high. Therefore, all of the BCM methods only try to reduce the self-overlapping region as small as possible, using some kinds of numerical energy minimization method, rather than to strictly guarantee the non-self-overlapping property [11,12].

Thus, developing a BCM method which can generate non-self-overlapping structured grid theoretically becomes one important issue in grid generation. In this paper, based on the convex-hull properties of a Bernstein polynomial, we present the Bézier patch mapping (BPM) method to generate non-self-overlapping structured quadrilateral grids. Specifically, BPM method constructs a regular Bézier patch interpolating the four given boundary curves, which maps the unit square to the four-sided region R and generates a non-self-overlapping structured quadrilateral grid on it.

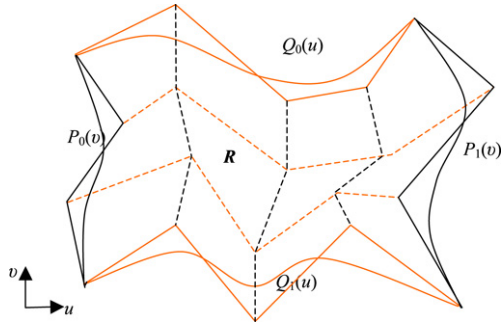


Fig. 3. The boundary curves P_0 and P_1 are converted into $(n + 1)$ -order Bézier curves, while Q_0 and Q_1 $(m + 1)$ -order Bézier curves. Their control polygons connect sequentially, forming the boundary ring of the control grid of a $(m + 1) \times (n + 1)$ -order Bézier patch. BPM method adjusts the inner control points to generate a regular Bézier patch.

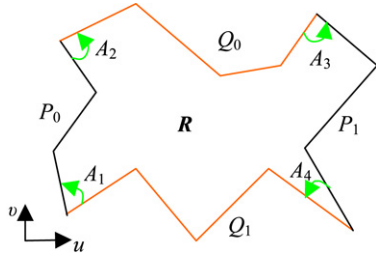


Fig. 4. Boundary constraint 3: the four corner angles A_1, A_2, A_3 and A_4 are less than 180 deg, respectively.

If the four given boundary curves are in power polynomial form, BPM method first converts the four power polynomials into Bernstein form, resulting in four Bézier curves: one pair of $(m + 1)$ -order curves, and the other pair of $(n + 1)$ -order curves (cf. Fig. 3). Their control polygons connect sequentially and construct the boundary ring of the control grid of the Bézier patch, which maps the unit square to the given four-sided planar region R (cf. Fig. 2). The inner control points of the Bézier patch remain unknown. In fact, the essence of the BPM method is to generate the regular Bézier patch by selecting the inner control points.

2.2. Boundary constrains

In order to generate a regular Bézier patch taking the four given curves as its boundary curves, the four given curves should satisfy some constraints. Suppose the four given curves are in Bernstein form, (a power basis polynomial can be converted into a Bernstein polynomial using the basis transformation formula [13,14]), that is, four Bézier curves, $Q_0(u), Q_1(u), u \in [0, 1]$, and $P_0(v), P_1(v), v \in [0, 1]$ (cf. Fig. 3). They must satisfy the following four *boundary constraints*:

- (1) Each of the four curves is non-self-intersecting;
- (2) The opposite curves have same degrees m and n respectively;
- (3) Consistence condition (cf. Fig. 4): The four *corner angles*, each formed by the two edges adjacent to a corner, are less

than 180 deg, respectively. It implies the four *corner normal vectors*, that is, $\mathbf{n}_{00} = Q'_1(0) \times P'_0(0)$, $\mathbf{n}_{10} = Q'_1(1) \times P'_1(1)$, $\mathbf{n}_{11} = Q'_0(1) \times P'_1(1)$ and $\mathbf{n}_{01} = Q'_0(0) \times P'_0(1)$, have the same direction;

- (4) The four curves are all regular curves. That is, the tangent vector at any point of each curve is not null vector.

Constraints (1) and (2) are intuitive, and Constraint (3) is depicted in Fig. 4. However, Constraint (4) is somewhat complicated. We only give some sufficient geometric conditions for generating the regular Bézier curves here, and the reader can refer to Ref. [15] for details.

Suppose the given Bézier curve is:

$$\mathbf{r}(t) = \sum_{i=0}^n \mathbf{P}_i B_i^n(t), \quad t \in [0, 1]. \quad (2)$$

Its derivative vector curve is:

$$\begin{aligned} \mathbf{r}'(t) &= n \sum_{i=0}^{n-1} (\mathbf{P}_{i+1} - \mathbf{P}_i) B_i^{n-1}(t) \\ &= n \sum_{i=0}^{n-1} (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i) B_i^{n-1}(t). \end{aligned} \quad (3)$$

And the square of the scaled norm of the derivative vector curve is:

$$\begin{aligned} f(t) &= \|\mathbf{r}'(t)\|^2 / n^2 \\ &= \sum_{m=0}^{2n-2} \sum_{j+k=m} [(x_{j+1} - x_j)(x_{k+1} - x_k) \\ &\quad + (y_{j+1} - y_j)(y_{k+1} - y_k) + (z_{j+1} - z_j) \\ &\quad \times (z_{k+1} - z_k)] \frac{\binom{n-1}{j} \binom{n-1}{k}}{\binom{2n-2}{j+k}} B_{j+k}^{2n-2}(t) \\ &= \sum_{m=0}^{2n-2} \sum_{j+k=m} \frac{\binom{n-1}{j} \binom{n-1}{k}}{\binom{2n-2}{j+k}} \\ &\quad \times [(\mathbf{P}_{j+1} - \mathbf{P}_j) \cdot (\mathbf{P}_{k+1} - \mathbf{P}_k)] B_{j+k}^{2n-2}(t) \\ &= \sum_{m=0}^{2n-2} f_m B_{j+k}^{2n-2}(t). \end{aligned} \quad (4)$$

Here, $B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$, and $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ is the binomial coefficient.

From formulae (3) and (4), according to the convex hull property of Bézier curves and Bernstein polynomials, some conditions can be deduced to guarantee the regularity of a Bézier curve. For example,

Condition (1): Let $\{\mathbf{P}_i = \{x_i, y_i, z_i\} | i = 0, 1, \dots, n\}$ be the control points of the Bézier curve (2), and let

$$\begin{aligned} X &= \{x_0, x_1, \dots, x_n\}, & Y &= \{y_0, y_1, \dots, y_n\}, \\ Z &= \{z_0, z_1, \dots, z_n\}. \end{aligned} \quad (5)$$

If one of the three sequences is strictly monotone, the Bézier curve (2) is regular.

Condition (2): If $(P_{i+1} - P_i) \cdot (P_{j+1} - P_j) > 0, i = 0, 1, \dots, n - 1, j = 0, 1, \dots, n - 1$, the Bézier curve (2) is regular. Here, $P_i, i = 0, 1, \dots, n$ are the control points of the Bézier curve (2).

3. Bijective Bézier patch mapping and initial construction of inner control points

3.1. Conditions for bijective Bézier patch mapping

First, the right-handed Cartesian coordinate system (x, y, z) should be so established that the four-sided region R is on the (x, y) -plane, and z -axis points to the direction of the corner normal vectors.

In this paper, the $(m + 1) \times (n + 1)$ -order Bézier patch, which interpolates the four given boundary curves, is regarded as a mapping from the unit square U to R (cf. Fig. 1), named *Bézier Patch Mapping* (BPM),

$$r(u, v) = (x(u, v), y(u, v)) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_i^m(u) B_j^n(v) : U \mapsto R, \quad (6)$$

where $P_{ij} = (x_{ij}, y_{ij})$, $B_i^m(u)$ and $B_j^n(v)$ are Bernstein basis. Note that the boundary control points, namely $P_{0j}, P_{m,j}, j = 0, 1, \dots, n$, and $P_{i0}, P_{i,n}, i = 0, 1, \dots, m$, are just the control points of the four given boundary Bézier curves, while the other inner control points remain unknown.

It is well-known that the above mapping is bijective if and only if its Jacobi determinant:

$$J(u, v) = \det \begin{pmatrix} x_u & x_v \\ y_u & y_v \end{pmatrix} = \sum_{i=0}^{2m} \sum_{k=0}^{2m-1} \sum_{j=0}^{2n-1} \sum_{l=0}^{2n} (\nabla_i x_{ij} \nabla_l y_{kl} - \nabla_i y_{ij} \nabla_l x_{kl}) B_i^{2m}(u) \times B_k^{2m-1}(u) B_j^{2n-1}(v) B_l^{2n}(v), \quad (7)$$

is not equal to zero in $U = [0, 1] \times [0, 1]$. Here:

$$\begin{aligned} \nabla_i x_{ij} &= x_{i+1,j} - x_{ij}, & \nabla_i y_{ij} &= y_{i+1,j} - y_{ij}, \\ i &= 0, 1, \dots, m - 1; & j &= 0, 1, \dots, n; \\ \nabla_l x_{kl} &= x_{k,l+1} - x_{kl}, & \nabla_l y_{kl} &= y_{k,l+1} - y_{kl}, \\ k &= 0, 1, \dots, m, & l &= 0, 1, \dots, n - 1. \end{aligned} \quad (8)$$

Due to:

$$B_i^{m-1}(u) B_k^m(u) = \frac{\binom{m-1}{i} \binom{m}{k} B_{i+k}^{2m-1}(u)}{\binom{2m-1}{i+k}},$$

$$B_j^n(v) B_l^{n-1}(v) = \frac{\binom{n}{j} \binom{n-1}{l} B_{j+l}^{2n-1}(v)}{\binom{2n-1}{j+l}}, \quad (9)$$

the Jacobi determinant (7) can be converted into:

$$J(u, v) = \sum_{p=0}^{2m-1} \sum_{q=0}^{2n-1} \left[\sum_{i+k=p} \sum_{j+l=q} (\nabla_i x_{ij} \nabla_l y_{kl} - \nabla_i y_{ij} \nabla_l x_{kl}) \times \frac{\binom{m-1}{i} \binom{m}{k} \binom{n}{j} \binom{n-1}{l}}{\binom{2m-1}{i+k} \binom{2n-1}{j+l}} \right] \times B_{i+k}^{2m-1}(u) B_{j+l}^{2n-1}(v). \quad (10)$$

It is a Bernstein polynomial with two variables, whose control values are:

$$T_{pq} = \sum_{i+k=p} \sum_{j+l=q} (\nabla_i x_{ij} \nabla_l y_{kl} - \nabla_i y_{ij} \nabla_l x_{kl}) \frac{\binom{m-1}{i} \binom{m}{k}}{\binom{2m-1}{i+k}} \times \frac{\binom{n}{j} \binom{n-1}{l}}{\binom{2n-1}{j+l}}; \quad p = 0, 1, \dots, 2m - 1; \quad q = 0, 1, \dots, 2n - 1. \quad (11)$$

Obviously, the Jacobi determinant $J(u, v)$ not being equal to zero in $[0, 1] \times [0, 1]$ implies either $J(u, v) > 0$ or $J(u, v) < 0, (u, v) \in [0, 1] \times [0, 1]$. In the following, by exploring the geometric meanings of the Jacobi determinant (7), we explain that only the first inequality $J(u, v) > 0$ is possible.

In order to investigate the geometric meanings of the Jacobi determinant (7), the control points of the Bézier patch (6) should be rewritten as: $P_{ij} = (x_{ij}, y_{ij}, 0)$. Thus, the normal vector surface of (6) is:

$$n(u, v) = r_u \times r_v = \sum_{p=0}^{2m-1} \sum_{q=0}^{2n-1} \left[\sum_{i+k=p} \sum_{j+l=q} (\nabla_i P_{ij} \times \nabla_l P_{kl}) \times \frac{\binom{m-1}{i} \binom{m}{k} \binom{n}{j} \binom{n-1}{l}}{\binom{2m-1}{i+k} \binom{2n-1}{j+l}} \right] \times B_{i+k}^{2m-1}(u) B_{j+l}^{2n-1}(v), \quad (12)$$

which is also a Bézier patch. On the other hand,

$$n(u, v) = r_u \times r_v = (x_u, y_u, 0) \times (x_v, y_v, 0) = (0, 0, \det \begin{pmatrix} x_u & x_v \\ y_u & y_v \end{pmatrix}) = (0, 0, J(u, v)). \quad (13)$$

Therefore, we have the following proposition:

Proposition 1. The Jacobi determinant (7) is the z -component of the normal vector of the Bézier patch (6).

That is to say, the Bézier patch mapping is bijective iff the corresponding Bézier patch is regular. Further,

Proposition 2. Only $J(u, v) > 0$ can exist in the so established coordinates system.

First, based on the end-interpolation property of Bézier surfaces, the four corner vectors of the normal vector surface (12), that is, $\mathbf{n}(0, 0)$, $\mathbf{n}(0, 1)$, $\mathbf{n}(1, 1)$, and $\mathbf{n}(1, 0)$, are just the four corner normal vectors \mathbf{n}_{00} , \mathbf{n}_{01} , \mathbf{n}_{11} , \mathbf{n}_{10} . (cf. Section 2.2). Together with that the z -axis points to the direction of the corner normal vectors, the normal vectors (12) or (13) can only point to the positive direction of the z -axis. In other words, only $J(u, v) > 0$ can exist. Otherwise, if $J(u, v) < 0$, the four corner vectors of the normal vector surface (12) (or (13)) would point to the negative direction of the z -axis. It contradicts the fact that the z -axis points to the direction of the four corner normal vectors.

Finally, based on the convex hull property of the Bernstein polynomial and the above two propositions, the following bijective condition of the Bézier patch mapping can be deduced:

Proposition 3 (Bijective Condition of BPM). If

$$T_{ij} \geq 0, \quad i = 0, 1, \dots, 2m - 1, \quad j = 0, 1, \dots, 2n - 1,$$

except that $T_{00} > 0, \quad T_{2m-1,0} > 0, T_{0,2n-1} > 0,$

$$T_{2m-1,2n-1} > 0,$$

the Bézier patch mapping (6) is bijective.

The proof of the above proposition is straightforward. If the condition is true, the Jacobi determinant (7) is greater than zero based on the convex hull property, that is $J(u, v) > 0$. Therefore, the Bézier patch mapping (6) is bijective. The bijective condition implies that the normal vectors of the Bézier patch point to the positive direction of the z -axis.

A geometric view of the above proposition is that, if the normal vector (12) points to the positive direction of the z -axis, that is, the direction of the corner normal vectors, the BPM (6) is bijective. In addition, for convenience of computation, the matrix form of T_{pq} (see Formula (11)) should be constructed, that is:

$$T = [T_{pq}]_{2m \times 2n} = D_1 \cdot A \cdot (E_1 \otimes F_1 - E_2 \otimes F_2) \cdot B \cdot D_2. \tag{14}$$

Here, D_1 and D_2 are the diagonal matrices,

$$D_1 = \text{diag} \left(1 / \binom{2m-1}{0}, 1 / \binom{2m-1}{1}, \dots, 1 / \binom{2m-1}{2m-1} \right),$$

$$D_2 = \text{diag} \left(1 / \binom{2n-1}{0}, 1 / \binom{2n-1}{1}, \dots, 1 / \binom{2n-1}{2n-1} \right); \tag{15}$$

\otimes denotes the kronecker product between the matrices E_1 and F_1 , E_2 and F_2 ,

$$E_1 = \begin{bmatrix} \binom{m-1}{m-1} \binom{n}{0} \nabla_i x_{00} & \cdots & \binom{m-1}{0} \binom{n}{n} \nabla_i x_{0,n} \\ \vdots & & \vdots \\ \binom{m-1}{m-1} \binom{n}{0} \nabla_i x_{m-1,0} & \cdots & \binom{m-1}{m-1} \binom{n}{n} \nabla_i x_{m-1,n} \end{bmatrix}, \tag{16}$$

$$F_1 = \begin{bmatrix} \binom{m}{0} \binom{n-1}{n-1} \nabla_l y_{00} & \cdots & \binom{m}{0} \binom{n-1}{0} \nabla_l y_{0,n-1} \\ \vdots & & \vdots \\ \binom{m}{m} \binom{n-1}{0} \nabla_l y_{m,0} & \cdots & \binom{m}{m} \binom{n-1}{n-1} \nabla_l y_{m,n-1} \end{bmatrix};$$

$$E_2 = \begin{bmatrix} \binom{m-1}{0} \binom{n}{0} \nabla_i y_{00} & \cdots & \binom{m-1}{0} \binom{n}{n} \nabla_i y_{0,n} \\ \vdots & & \vdots \\ \binom{m-1}{m-1} \binom{n}{0} \nabla_i y_{m-1,0} & \cdots & \binom{m-1}{m-1} \binom{n}{n} \nabla_i y_{m-1,n} \end{bmatrix}, \tag{17}$$

$$F_2 = \begin{bmatrix} \binom{m}{0} \binom{n-1}{n-1} \nabla_l x_{00} & \cdots & \binom{m}{0} \binom{n-1}{0} \nabla_l x_{0,n-1} \\ \vdots & & \vdots \\ \binom{m}{m} \binom{n-1}{0} \nabla_l x_{m,0} & \cdots & \binom{m}{m} \binom{n-1}{n-1} \nabla_l x_{m,n-1} \end{bmatrix}.$$

A is a $2m \times m(m+1)$ order matrix, and B is a $n(n+1) \times 2n$ order matrix,

$$A = \begin{pmatrix} \boxed{I_{m+1}} & & & \\ & \boxed{I_{m+1}} & & \\ & & \ddots & \\ & & & \boxed{I_{m+1}} \end{pmatrix}, \quad B = \begin{pmatrix} \boxed{I_n} & & & \\ & \boxed{I_n} & & \\ & & \ddots & \\ & & & \boxed{I_n} \end{pmatrix}. \tag{18}$$

Specifically, the matrix A consists of m identity matrices with order $m+1$. As shown in (18), each identity matrix shifts downwards one row from its previous identity matrix. For example, the first one locates from the first row to the $(m+1)$ -th row, the second one locates from the second row to the $(m+2)$ -th row, and so on. The matrix B consists of $n+1$ identity matrices with order n . Also as shown in (18), each identity matrix shifts rightwards one column from its previous identity matrix. For example, the first one occupies from the first column to the n -th column, the second one from the second column to the $(n+1)$ -th column, and so on. It should be pointed out that, since the diagonal entries of the diagonal matrices D_1 and D_2 are all greater than 0, they can be deleted from the Eq. (14), and only the remaining product of matrices needs to be considered.

3.2. Initial construction of inner control points

The BPM method adjusts the inner control points of the control grid to generate a regular Bézier patch (therefore, a bijective Bézier patch mapping). Hence, after getting the boundary control polygon ring of the control grid, the initial inner control points should be constructed as a start for BPM by the following method, which calculates the initial set of inner control points by an averaging method.

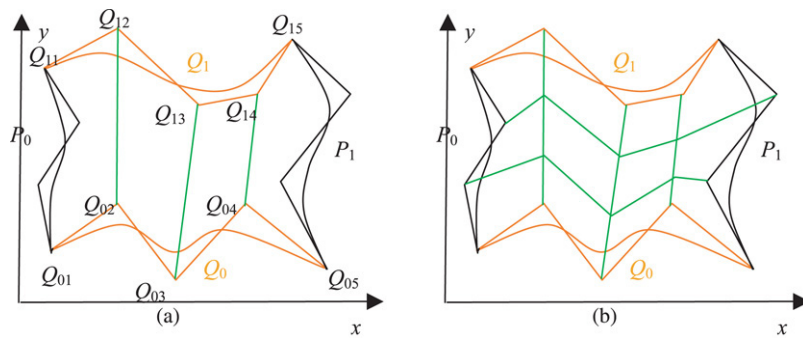


Fig. 5. Generation of the initial control grid. (a) Three line segments are constructed by connecting the corresponding points in a pair of opposite boundary polygons. (b) The initial control grid is formed by orderly connecting the corresponding points in the other pair of boundary polygons and the newly constructed control points.

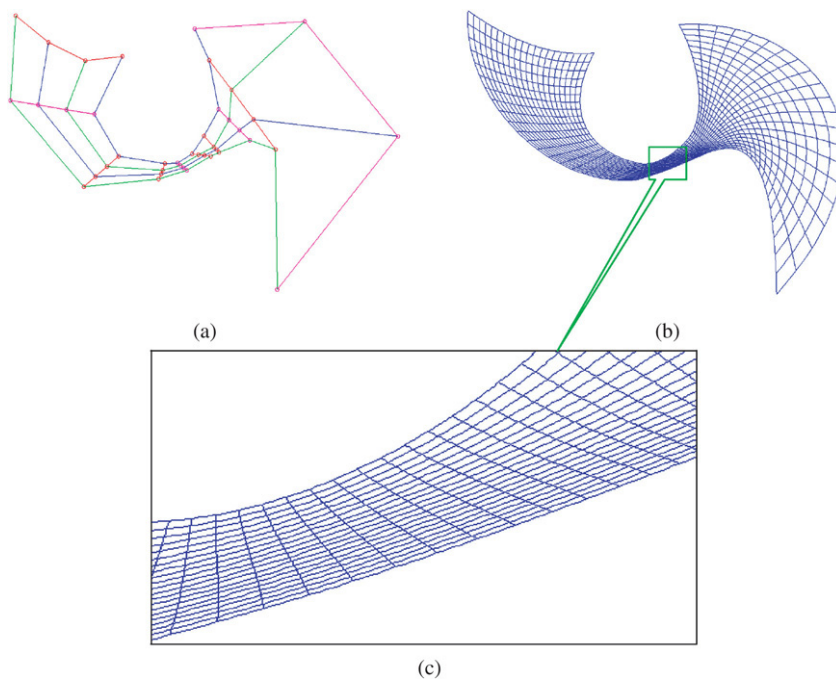


Fig. 6. Even the initial control grid can generate a satisfactory structured quadrilateral grid. The initial control grid satisfies the bijective condition, so the generated structured quadrilateral grid has no self-overlapping: (a) The initial control grid; (b) The structured quadrilateral grid; (c) Zoom of part of the structured grid.

Specifically, given a boundary control polygon ring (cf. Fig. 5), m being the number of control points on each of a pair of opposite boundary polygons, and n for the other pair of opposite boundary polygons, the initial inner control points are calculated as follows:

- (1) Choose the pair of opposite boundary control polygons which has more control points than the other pair, e.g. suppose $m > n$, we choose the pair of opposite boundary polygons each of which has m control points. In the case of $m = n$, an arbitrary pair is chosen. In Fig. 5(a), the pair of opposite boundary polygons, Q_0 and Q_1 , is chosen, where

$$Q_0 = \{Q_{01}, Q_{02}, \dots, Q_{05}\}, \quad \text{and} \\ Q_1 = \{Q_{11}, Q_{12}, \dots, Q_{15}\}.$$

- (2) Connect each pair of control points in the selected opposite boundary polygons with line segment in turn except the two

pairs of end points, producing $m - 2$ pieces of line segments. For example, three line segments, $L_2 = Q_{02}Q_{12}$, $L_3 = Q_{03}Q_{13}$, $L_4 = Q_{04}Q_{14}$, are generated for the case in Fig. 5(a).

- (3) Divide each of the generated $m - 2$ pieces of line segments into $n - 1$ portions of equal length, generating $(m - 2) \times (n - 1)$ points. They are taken as the initial inner control points. In the case of Fig. 5, each of the line segments L_2 , L_3 , and L_4 are divided into three equal portions, generating six inner control points (cf. Fig. 5(b)).
- (4) Connect the corresponding control points in the other pair of boundary control polygons and the newly generated inner control points orderly, creating the initial $m \times n$ order control grid. Such as the example in Fig. 5(b), by connecting the corresponding points in the other opposite boundary polygons, namely P_0 and P_1 , and the newly

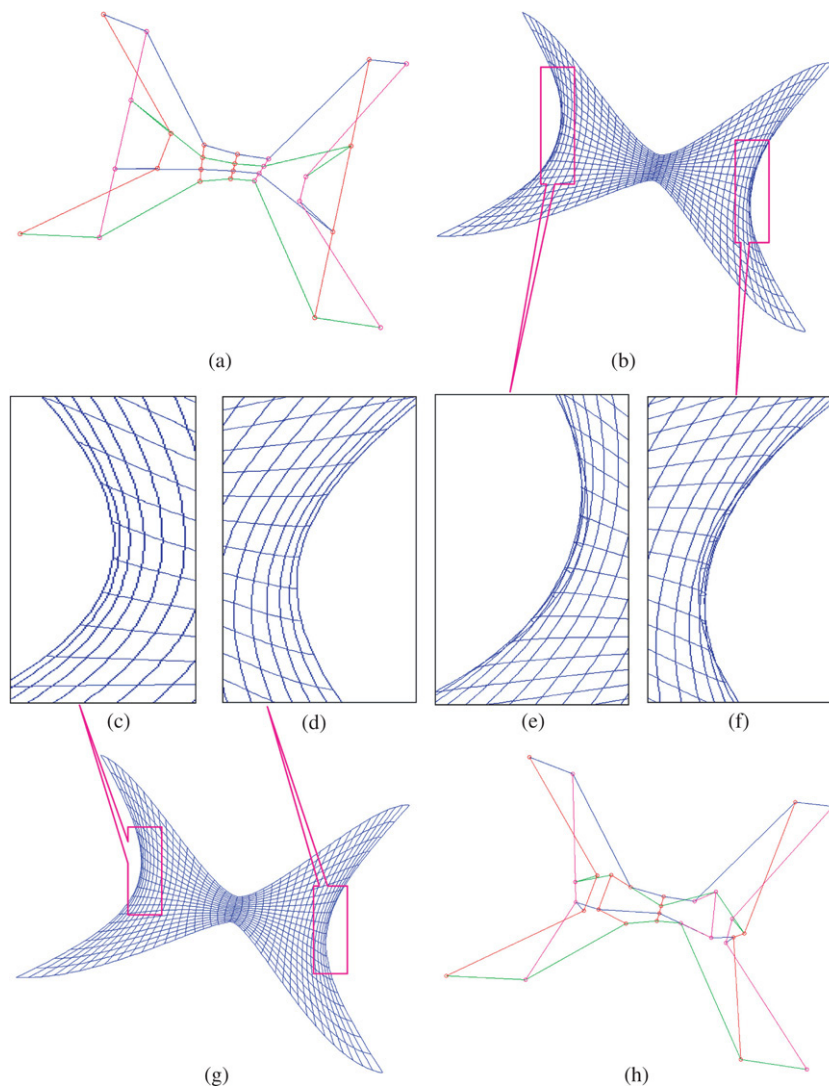


Fig. 7. The poor configuration of the initial control grid (a) leads to the structured quadrilateral grid (b) with self-overlapping, the close-ups of which are shown in (e) and (f). By adjusting the inner control points to attain the solution to the optimization problem (20), the configuration of the control grid (h) is improved, and self-overlapping is eliminated from its corresponding structured quadrilateral grid (g), the close-ups of which are shown in (c) and (d).

generated inner control points orderly, a 5×4 control grid is presented.

The above construction starts at the pair of boundary polygons with more control points. However, if the boundary control polygon ring has very complex shape, some user interaction may be required to select a reasonable pair of boundary polygons for start. The initial control grid so constructed may have self-overlapping. Although not ideal, it is acceptable as a starting point. From the initial control grid, the BPM method adjusts its inner control points until the final control grid is the solution to a constrained optimization problem as described in the following section.

4. Non-self-overlapping structured grid generation and improvement

Based on the analysis in Section 3, we are now at the position to compute the bijective *Bézier Patch Mapping* (BPM) (6), which has $2 \times (m - 1) \times (n - 1)$ scalar unknowns in

$P_{11}, P_{12}, \dots, P_{m-1,n-1}$. As discussed in the previous section, this is equivalent to making its Jacobi determinant (7) greater than 0. Suppose the four boundary curves satisfy the four constraints as presented in Section 2.2, based on the bijective condition (Proposition 3), the Jacobi determinant (7) is greater than 0, if:

$$T_{pq} \geq 0, \quad p = 0, 1, \dots, 2m - 1, \quad q = 0, 1, \dots, 2n - 1,$$

$$\text{except that } T_{00} > 0, \quad T_{0,2n-1} > 0, \quad T_{2m-1,0} > 0,$$

$$\text{and } T_{2m-1,2n-1} > 0.$$

On the other hand, the structured quadrilateral grid generated by bijective BPM should be further optimized in order to improve the quality of the grid. In general, a good grid (actually the parametric curve grid of the Bézier patch $r(u, v)$ (6)) should at least have no self-overlapping, and then additionally be as orthogonal as possible. Therefore, we adopt the optimization objective function [16,17]:

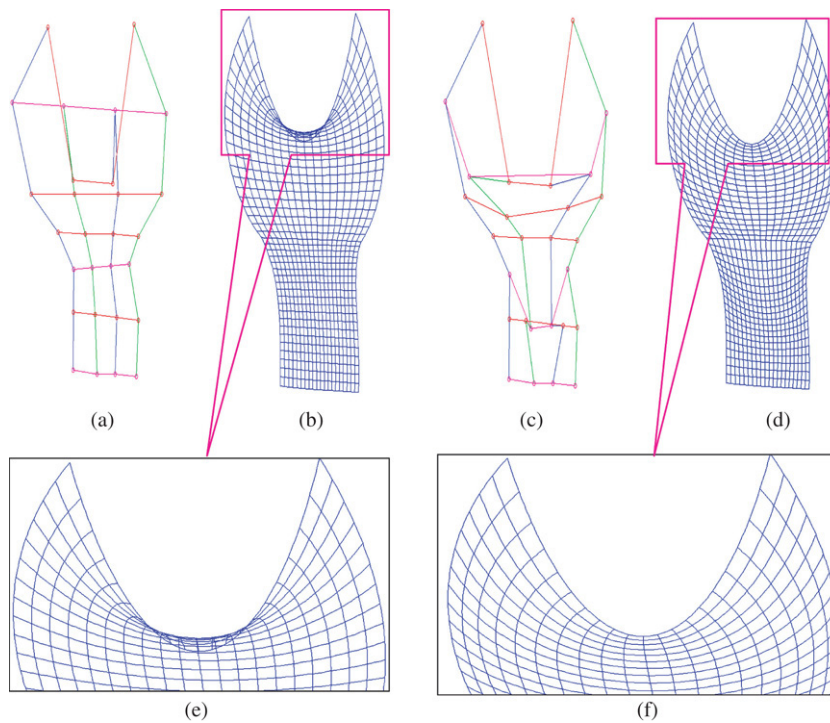


Fig. 8. The initial control grid (a) leads to the self-overlapping structured quadrilateral grid (b), by adjusting the inner control points to attain the solution to the optimization problem (20), the obtained control grid (c) generates the non-self-overlapping structured quadrilateral grid (d). (e) is the close-up of the self-overlapping portion in (b), and (f) is the close-up of the same portion in (d), but without self-overlapping.

$$\begin{aligned}
 & \arg \min_{\mathbf{P}_{11}, \mathbf{P}_{12}, \dots, \mathbf{P}_{m-1, n-1}} \int_{[0,1] \times [0,1]} \left(\left\| \frac{\partial \mathbf{r}}{\partial u} \right\|^2 + \left\| \frac{\partial \mathbf{r}}{\partial v} \right\|^2 \right) dudv. \quad (19)
 \end{aligned}$$

The reason for adopting the above objective function is twofold. First, as a whole, the formula (19) is the membrane model which is a small deflection approximation of the surface area. However, if we examine the objective function (19) part by part, we can find out that the function $\arg \min_{\mathbf{P}_{11}, \mathbf{P}_{12}, \dots, \mathbf{P}_{m-1, n-1}} \iint_{[0,1] \times [0,1]} \left\| \frac{\partial \mathbf{r}}{\partial u} \right\|^2 dudv$ is related to the arc-length of the u -directional iso-parameter lines of the Bézier patch (6), and it tries to make the u -directional iso-parameter lines as short as possible. Similarly, the function $\arg \min_{\mathbf{P}_{11}, \mathbf{P}_{12}, \dots, \mathbf{P}_{m-1, n-1}} \iint_{[0,1] \times [0,1]} \left\| \frac{\partial \mathbf{r}}{\partial v} \right\|^2 dudv$ attempts to make the v -directional iso-parameter lines as short as possible.

Second, since $\left\| \frac{\partial \mathbf{r}}{\partial u} \right\|^2 + \left\| \frac{\partial \mathbf{r}}{\partial v} \right\|^2 \geq 2 \left\| \frac{\partial \mathbf{r}}{\partial u} \cdot \frac{\partial \mathbf{r}}{\partial v} \right\|$, the objective function (19) has another effect that makes the intersecting angle between the u - and v -directional iso-parameter lines as close to the right-angle as possible.

Therefore, the whole problem, that is, non-self-overlapping structured quadrilateral grid generation and improvement, can be formulated as:

$$\begin{aligned}
 & \arg \min_{\mathbf{P}_{11}, \mathbf{P}_{12}, \dots, \mathbf{P}_{m-1, n-1}} \int_{[0,1] \times [0,1]} \left(\left\| \frac{\partial \mathbf{r}}{\partial u} \right\|^2 + \left\| \frac{\partial \mathbf{r}}{\partial v} \right\|^2 \right) dudv \\
 & \text{subject to : } T_{pq} \geq 0, \quad p = 0, 1, \dots, 2m - 1,
 \end{aligned}$$

$$\begin{aligned}
 & q = 0, 1, \dots, 2n - 1, \text{ except that } T_{00} > 0, \\
 & T_{0, 2n-1} > 0, \quad T_{2m-1, 0} > 0, \quad \text{and} \quad T_{2m-1, 2n-1} > 0. \quad (20)
 \end{aligned}$$

This is a constrained optimization problem. The constraints ensure the bijectiveness of the Bézier patch mapping, which in theory guarantees that the generated structured quadrilateral grid will have no self-overlapping. Furthermore, the objective function tries to find the best grid from all of the grids with this property.

To solve this constrained optimization problem, we employ the Matlab function *fmincon* under its default setting. It starts at an initial solution and finds a constrained minimum of the function (20). Here, the initial inner control points constructed by the method in the Section 3.2 can act as the initial solution. Details on the function *fmincon* can be found in Ref. [18]. Finally, we conclude the whole procedure by the following Algorithm 1.

Algorithm 1 (Generating Bijective BPM).

1. Take as input the four planar curves enclosing a four-sided region R , which satisfy the four boundary constraints (Section 2.2);
2. Construct the initial inner control points (Section 3.2);
3. Construct the matrices T (formula (14));
4. Solve the optimization problem with constraints (20) using the MatLab function *fmincon*; the independent variables in the optimization are $\mathbf{P}_{11}, \mathbf{P}_{12}, \dots, \mathbf{P}_{m-1, n-1}$, as given in the formula (6);
5. Generate the non-self-overlapping structured quadrilateral grid using the bijective BPM (formula (6)).

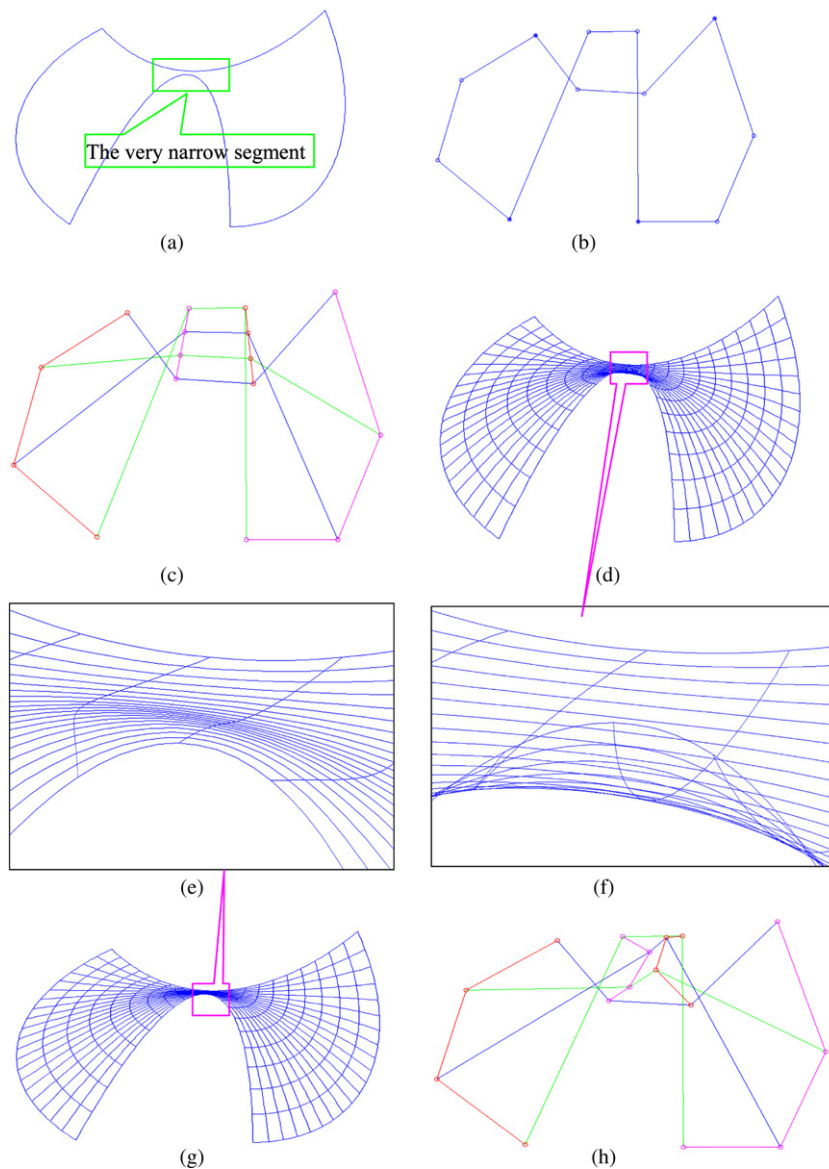


Fig. 9. Even though the boundary polygons (b) of the 4-sided region (a) with very narrow segment, have self-overlapping, and the initial control grid (c) gives a self-overlapping structured quadrilateral grid (d), by adjusting the inner control points to attain the solution to the optimization problem (20), a new control grid (h) is obtained, which generates the structured grid (g) without self-overlapping. (f) is the close-up of the self-overlapping portion in (d), and (e) is the close-up of the same portion in (g), which has no self-overlapping.

5. Examples

Several test examples are given in Figs. 6–9 to validate the presented BPM grid generation algorithm. The example in Fig. 6 demonstrates that, in some special cases, even the initial control grid can generate a satisfactory structured quadrilateral grid. The initial control grid satisfies the bijective condition, so it guarantees that the generated structured grid has no self-overlapping. In Figs. 7 and 8, the poor configuration of the initial control grid leads to a self-overlapping structured quadrilateral grid. By adjusting the inner control points so as to attain the solution to the constrained optimization problem (20), a non-self-overlapping structured quadrilateral grid can be generated. In the last example, shown in Fig. 9, the very narrow channel in the region forces even the four boundary

polygons to intersect each other, thus leading to a very poor initial control grid, the BPM method is still able to produce a good grid without self-overlapping.

6. Conclusion

The boundary-conforming mapping (BCM) method is often used to generate structured quadrilateral grids on planar four-sided regions. However, the bijective boundary-conforming mapping is difficult to obtain, because checking whether a map is bijective requires solving a highly nonlinear system, which can only be done by numerical methods. Therefore, generally speaking, the BCM method is not able to ensure the non-self-overlapping property in the generated grid. In this paper, based on the convex-hull property of Bernstein, we develop a bijective

Bézier patch mapping (BPM) algorithm to generate strictly non-self-overlapping structured quadrilateral grid in a planar four-sided region. The BPM method is simple and intuitive, and can generate true non-self-overlapping structured quadrilateral grid. Test examples are also given to validate the effectiveness and efficiency of the presented algorithm.

Acknowledgements

We thank the reviewers for their valuable comments. This work is supported by the Foundation of State Key Basic Research 973 Item (No. 2004CB719400), the National Natural Science Foundations (No. 60503057, No. 60021201).

References

- [1] Thompson JF. A survey of dynamically-adaptive grids in the numerical solution of partial differential equations. *Appl Numer Math* 1985;1:3–27.
- [2] Soni BK. Elliptic grid generation system: Control functions revisited. I. *Appl Math Comput* 1993;59:151–63.
- [3] Khamayseh A, Hamann B. Elliptic grid generation using NURBS surfaces. *Comput Aided Geom Design* 1996;13:369–86.
- [4] Knupp PM. Jacobian-weighted elliptic grid generation. *SIAM J Sci Comput* 1996;17:1475–90.
- [5] Kim S. Control functions and grid qualities measurements in the elliptic grid generation around arbitrary surfaces. *Internat J Numer Methods Fluids* 2000;33:81–8.
- [6] Smith RE, Eriksson LE. Algebraic grid generation. *Comput Methods Appl Mech Engrg* 1987;64:285–300.
- [7] Shih TIP, Bailey RT, Nguyen HL, Roelke RJ. Algebraic grid generation for complex geometries. *Internat J Numer Methods Fluids* 1991;13:1–31.
- [8] Brakhage KH, Muller S. Algebraic-hyperbolic grid generation with precise control of intersection of angles. *Internat J Numer Methods Fluids* 2000;33:89–123.
- [9] Yang DCH, Chuang J-J, OuLee TH. Boundary-conformed toolpath generation for trimmed free-form surfaces. *Comput-Aided Design* 2003;35(2):127–39.
- [10] Wang C, Tang K. Non-self-overlapping structured grid generation on an n -sided surface. *Internat J Numer Methods Fluids* 2004;46(9):961–82.
- [11] Wang C, Tang K. Non-self-overlapping Hermite interpolation mapping: A practical solution for structured quadrilateral meshing. *Comput-Aided Design* 2005;37(2):271–83.
- [12] Wang C, Tang K. Algebraic grid generation on trimmed surface using non-self-overlapping Coons patch mapping. *Internat J Numer Methods Engrg* 2004;60(7):1259–86.
- [13] Farouki RT, Rajan VT. On the numerical condition of polynomials in Bernstein form. *Comput Aided Geom Design* 1987;4:191–216.
- [14] Farouki RT, Rajan VT. Algorithms for polynomials in Bernstein form. *Comput Aided Geom Design* 1988;5:1–26.
- [15] Lin H, Bao H. Regular curve: Some geometric conditions and a necessary and sufficient condition. In: *Proceedings of ninth international conference on computer aided design and computer graphics' 2005*. HongKong: IEEE Press; 2005. p. 52–7.
- [16] Wang XF, Cheng F, Barsky BA. Energy and B -spline interproximation. *Comput-Aided Design* 1997;29(7):485–96.
- [17] Cheng C, Zhang YF. Thin plate spline surface approximation using Coons patches. *Comput Aided Geom Design* 1994;11:269–87.
- [18] Matlab online manual. The MathWorks, Inc.