IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 19, 2021, accepted August 7, 2021, date of publication August 16, 2021, date of current version September 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104909

Solving the Boundary Value Problem of Curves With Prescribed Geodesic Curvature Based on a Cubic B-Spline Element Method

PENG ZHANG, LAIRONG YIN, ZHENHUA ZHOU, AND LONG HUANG
College of Automotive and Mechanical Engineering, Changsha University of Science and Technology, Changsha 410114, China
Key Laboratory of Lightweight and Reliability Technology for Engineering Vehicle, Education Department of Hunan Province, Changsha University of Science and Technology, Changsha 410114, China

Corresponding author: Lairong Yin (yinlairong@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51705036 and Grant 51705034, in part by Hunan Provincial Natural Science Foundation of China under Grant 2018JJ3542, and in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 17C0045.

ABSTRACT Composite manufacturing processes such as automated tape placement (ATP) and filament winding process (FW) put forward specific requirements for the geodesic curvature along the layup paths of the composite tows and tapes. In this paper, a non-uniform cubic b-spline element method is proposed for solving the boundary value problem of curves with prescribed geodesic curvature. The differential equation system of the target curve is discretized through the point collocation method, and a quasi-Newton iteration scheme is adopted to approach the real solution from an initial approximation. The proposed method is proved to have third order accuracy, which shows more superiorities comparing with existing numerical methods. Simulations and experiments on a series of parametric surfaces are performed to investigate the performance of the proposed approach and the results verify the high efficiency. The proposed method could cope with the BVP for curves no matter their geodesic curvature vanishes or not. At the same time, the computed curves are natural and smooth such that interpolation technique is unnecessary to ensure the continuity of target curves. One potential application of this method is trajectory optimization for automated tape placement process.

INDEX TERMS Cubic b-splines, geodesic curvature, boundary value problem (BVP), automated tape placement (ATP).

I. INTRODUCTION

Curves lying on surfaces show many applications related to design and manufacture, such as surface trimming [1], surface blending [2], NC tool path generation [3], [4], and so on. According to the designing manner, curves on a surface can be the offset of a given curve on a surface [3], the intersection curve of two surfaces [4], the projection curve of a spatial curve onto a surface [5]–[7], or the image of a curve in the parametric domain of a parametric surface [1], [2].

Geodesic curvature is an intrinsic geometric feature of a surface curve which can be defined as the curvature of the curve projected onto the surface's tangent plane. A curve whose geodesic curvature is zero everywhere is called a geodesic curve, and it is locally the shortest distance between two points on the surface. The concept of geodesic curve finds its place in computer vision, image processing and various industrial applications, such as trajectory planning in automated tape placement (ATP) and filament winding process (FW). For ATP, the geodesic curve allows minimizing the steering of the tape. For FW, it would be impossible to lay a filament in any way other than along a geodesic curve on a frictionless convex surface.

The geodesic computation involves in ATP includes initial value problem (IVP) and boundary value problem (BVP). The IVP for geodesic solves a uniquely determined curve with zero geodesic curvature once the initial condition is provided. As a fundamental problem in computational geometry and geometric modeling, the IVP for geodesic has been studied extensively in the past three decades. To date, many elegant methods have been proposed. Representative works include the numerical approaches [8], [9], the discrete methods [10], [11], and the geometric methods [12], [13]. The BVP for geodesic finds the shortest path between two points

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

on a surface. It is well known that the BVP is more challenging than IVP and may have non-unique solution. In general, available approaches for BVP of geodesic curve could be classified broadly into two types. First, solve the nonlinear differential equations of geodesic by using a finite difference method [14], [15] or an optimization strategy [16]. Second, approximate accurate geodesic curve on discrete surfaces by following some rules [17], [18]. The first type aimed at the smooth surface and usually shows better performance than the latter one [19].

In this paper, we extend the idea of geodesic curve to curves with prescribed geodesic curvature. The geodesic curvature along such curve is provided in terms of a specific application and not necessarily zero. The curves with prescribed geodesic curvature show three potential applications. Firstly, although geodesic did minimize the distortion of composite tape, it might lead to overlaps and excessive gaps between tapes. The curves with prescribed geodesic curvature could be utilized to steer the composite tapes such that eliminating the layup defects and ensuring high quality of composite components [20]. Secondly, the algorithm for curves with prescribed geodesic curvature could be extended to solve the non-geodesic winding trajectory in FW. In this process, the requirement for geodesic curvature is that the ratio of geodesic curvature to the normal curvature at each point of the winding path should not exceed the coefficient of maximum static friction between the fiber and the mandrel surface [21]. Thirdly, Azariadis and Aspragathos [22] pointed out that to flatten an arbitrary non-developable surface with as little distortion as possible, a good approach should try to preserve the geodesic curvatures on the surface. Thus, the curve with prescribed geodesic curvature, i.e. constant geodesic curvature might be a candidate to evaluate the performance of the surface flattening algorithm.

This paper solves the boundary value problem of curves with prescribed geodesic curvature on smooth parametric surfaces. The discrete approximation method shows difficulties in determining how the curve with non-zero geodesic curvature should pass on surfaces. However, the numerical approach introduced by Maekawa, Kasap and Chen for BVP of geodesic could be extended to curves with prescribed geodesic curvature directly. The finite difference method proposed by Maekawa and Kasap obtains discrete points along geodesic curve, and an interpolation technique is needed to ensure the continuity of desired curve. The optimization strategy delivered by Chen is accurate and could solve natural and smooth geodesic curves. In this paper, we prove that Chen's approach is equivalent to solving the differential equation of geodesics based on the Galerkin method. The Galerkin method is inconvenient when the basis function is chosen inadequately due to that the differential equation of curves with prescribed geodesic curvature is complicated and highly nonlinear.

This work is based on the cubic b-spline element method. The method has been applied to solve the heat and wave equation [23], the Burgers' equation [24], the 2D-Poisson equation [25] and so on. However, in most cases, uniform B-splines are adopted, which is not suitable for the BVP of curves with prescribed geodesic curvature. This is mainly due to that the desired curve is arc length parametrization, and the arc length is variable during the computation. The rest of this paper is organized as follows: Section 2 describes the problem in more detail; Section 3 presents the principle of our algorithm; Section 4 performs error analysis of the algorithm; Section 5 tests the proposed approach on several parametric surfaces, and some of the results are demonstrated; and Section 6 provides the discussion and conclusion.

II. DESCRIPTION OF THE PROBLEM

ATP is an important automated process used for fabrication of large composite structures in aeronautical industry. This process utilizes a composite tape layup end-effector mounted on a large robotic manipulator, and adds material layer by layer to build the composite part. As illustrated in Fig. 1, the end-effector moves over the mould surface in direct contact, laying a successive of continuous unidirectional composite tapes [26]. During the placement process, the centerlines of composite tapes generally follow geodesics as they allow minimizing the steering of the tape.

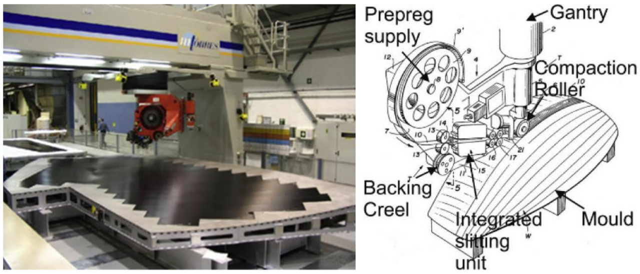


FIGURE 1. Schematic diagram of automated tape placement [27].

Due to the Gaussian curvature of the mould surface, the geodesics are not parallel to each other, which might lead to overlaps and excessive gaps between adjacent tapes. Engineering cases indicate that this phenomenon is more notable at the end of laid tapes on a complex mould surface. Figure 2 presents the manufacture process of a complex composite part through ATP. Note the excessive gaps occur at the end of laid tapes. Engineering experience reveals that

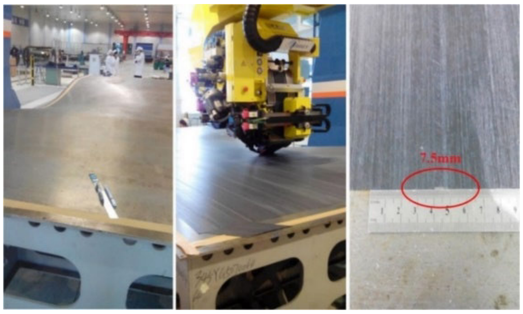


FIGURE 2. Excessive gaps appear at the end of laid tapes.

a good lay-up should prevent overlaps and constrain the gaps within 2.5mm.

Overlaps and excessive gaps are highly undesirable as they will result in a weaken part. To tackle with this dilemma, Hogg [28] constructed the lay-up trajectories with a plurality of geodesic segments, each of which defined an offset angle relative to the adjacent segments. The offset angle was used to adjust the shape of lay-up path so as to minimize the gaps or overlaps. However, the offset angle results in an unsmoothed curve. Although Hogg indicated that the offset angle of each geodesic segment should be less than a predetermined maximum value, the concentration of distortion at the connecting points is inevitable, which could possibly lead to the appearance of wrinkles in the composite tape. Alternatively, we could utilize the curve with very small geodesic curvature to ameliorate the distortion at the connect point, as seen in Fig. 3. Recalling that the appearance of wrinkles is related to the geodesic curvature along the lay-up paths, the curve with small geodesic curvature would suppress the occurrence of the deposition problem.

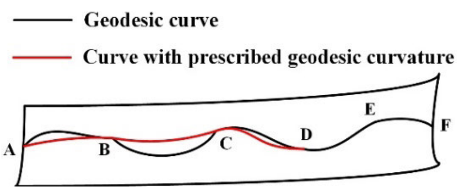


FIGURE 3. Optimizing the layup trajectory for ATP.

A preliminary numerical model for the curve with prescribed geodesic curvature was established in Reference [20]. After several modifications, this numerical model is simplified as follows:

$$\begin{cases} \dfrac{\mathrm{d}^2u}{\mathrm{d}s^2}+\Gamma_{11}^1\left(\dfrac{\mathrm{d}u}{\mathrm{d}s}\right)^2+2\Gamma_{12}^1\dfrac{\mathrm{d}u}{\mathrm{d}s}\dfrac{\mathrm{d}v}{\mathrm{d}s}+\Gamma_{22}^1\left(\dfrac{\mathrm{d}v}{\mathrm{d}s}\right)^2 \\ +\dfrac{k_g}{\sqrt{EG-F^2}}\left(G\dfrac{\mathrm{d}v}{\mathrm{d}s}+F\dfrac{\mathrm{d}u}{\mathrm{d}s}\right)=0, \\ \dfrac{\mathrm{d}^2v}{\mathrm{d}s^2}+\Gamma_{11}^2\left(\dfrac{\mathrm{d}u}{\mathrm{d}s}\right)^2+2\Gamma_{12}^2\dfrac{\mathrm{d}u}{\mathrm{d}s}\dfrac{\mathrm{d}v}{\mathrm{d}s}+\Gamma_{22}^2\left(\dfrac{\mathrm{d}v}{\mathrm{d}s}\right)^2 \\ -\dfrac{k_g}{\sqrt{EG-F^2}}\left(F\dfrac{\mathrm{d}v}{\mathrm{d}s}+E\dfrac{\mathrm{d}u}{\mathrm{d}s}\right)=0, \end{cases} \tag{1}$$

where s is the arc length, and $\Gamma_{jk}^i=\Gamma_{jk}^i(u,v)\ (i,j,k=1,2)$ are the Christoffel symbols which are defined as follows:

$$\Gamma_{11}^1=\frac{GE_u-2FF_u+FE_v}{2(EG-F^2)},\quad \Gamma_{11}^2=\frac{2EF_u-EE_v+FE_u}{2(EG-F^2)},$$
$$\Gamma_{12}^1=\frac{GE_v-FG_u}{2(EG-F^2)},\quad \Gamma_{12}^2=\frac{EG_u-FE_v}{2(EG-F^2)},$$
$$\Gamma_{22}^1=\frac{2GF_v-GG_u+FG_v}{2(EG-F^2)},\quad \Gamma_{22}^2=\frac{EG_v-2FF_v+FG_u}{2(EG-F^2)}.$$

In Fig. 3, the problem involves connecting discrete points on mould surface by curves with prescribed geodesic curvature, which is equivalent to solving the boundary value problem of the Eq. (1).

III. PRINCIPLE OF THE ALGORITHM

A. THE NON-UNIFORM CUBIC B-SPLINES

In this paper, non-uniform cubic B-spline functions are used to solve the differential equation (1).

Let us consider a partition $0=s_0<s_1<\ldots s_{N-1}<s_N=L$ on a given interval $[0,L]$. B-splines are defined through a recursive relation introduced by Boor [30] in the early 1970s. The B-splines of degree 3 are given by:

$$B_i^3(s)=\begin{cases} \dfrac{(s-s_i)^3}{(s_{i+3}-s_i)(s_{i+2}-s_i)(s_{i+1}-s_i)} \\ \qquad s\in[s_i,s_{i+1}], \\ \dfrac{(s-s_i)^2(s_{i+2}-s)}{(s_{i+3}-s_i)(s_{i+2}-s_i)(s_{i+2}-s_{i+1})} \\ +\dfrac{(s-s_i)(s_{i+3}-s)(s-s_{i+1})}{(s_{i+3}-s_i)(s_{i+3}-s_{i+1})(s_{i+2}-s_{i+1})} \\ +\dfrac{(s_{i+4}-s)(s-s_{i+1})^2}{(s_{i+4}-s_{i+1})(s_{i+3}-s_{i+1})(s_{i+2}-s_{i+1})} \\ \qquad s\in[s_{i+1},s_{i+2}], \\ \dfrac{(s-s_i)(s_{i+3}-s)^2}{(s_{i+3}-s_i)(s_{i+3}-s_{i+1})(s_{i+3}-s_{i+2})} \\ +\dfrac{(s_{i+4}-s)(s-s_{i+1})(s_{i+3}-s)}{(s_{i+4}-s_{i+1})(s_{i+3}-s_{i+1})(s_{i+3}-s_{i+2})} \\ +\dfrac{(s_{i+4}-s)^2(s-s_{i+2})}{(s_{i+4}-s_{i+1})(s_{i+4}-s_{i+2})(s_{i+3}-s_{i+2})} \\ \qquad s\in[s_{i+2},s_{i+3}], \\ \dfrac{(s_{i+4}-s)^3}{(s_{i+4}-s_{i+1})(s_{i+4}-s_{i+2})(s_{i+4}-s_{i+3})} \\ \qquad s\in[s_{i+3},s_{i+4}], \\ 0 \qquad \text{otherwise.} \end{cases} \tag{2}$$

The last equation is a cubic spline with knots s_i, s_{i+1}, s_{i+2}, s_{i+3}, s_{i+4}. Note that the cubic B-spline is zero except on the interval $[s_i,s_{i+4})$.

B. THE BOUNDARY VALUE PROBLEM

The boundary value problem in this section is stated as follows: Finding a curve with prescribed geodesic curvature between two boundary points $(u_0,\ v_0)$ and $(u_n,\ v_n)$ on a parametric surface.

The solution of Eq. (1) is approximated by:

$$\begin{cases} u(s)=\displaystyle\sum_{i=0}^{n}B_i^3(s)u_i, \\ v(s)=\displaystyle\sum_{i=0}^{n}B_i^3(s)v_i, \end{cases} \tag{3}$$

where u_i, $(1\le i\le n-1)$ and v_i, $(1\le i\le n-1)$ are the unknown real coefficients and $B_i^3(s)$ are the basis functions of the cubic B-spline defined on a knot vector $U_1 = \left\{ {}^1t_1, {}^1t_2 \ldots {}^1t_{N+6}, {}^1t_{N+7} \right\} = \left\{ 0, 0, 0, 0, \ s_1, \ s_2 \ldots s_{N-2}, s_{N-1}, L, L, L, L \right\}$.

For the cubic B-spline, $N = n - 2$.

Note the first knot and the last knot are of multiplicity 4 so that the clamped b-spline curves $u(s)$ and $v(s)$ pass through the end control points u_0, v_0, u_n, v_n.

With reference [31], the first order derivative and second order derivative of $u(s)$ and $v(s)$ are denoted as:

$$\begin{cases} u'(s) = \sum\limits_{i=0}^{n-1} B_i^2(s) p_i, \\ v'(s) = \sum\limits_{i=0}^{n-1} B_i^2(s) q_i, \end{cases} \tag{4}$$

where $B_i^2(s)$ are the basis functions of the quadratic B-spline defined on a knot vector $U_2 = \left\{ {}^2t_1, {}^2t_2 \ldots {}^2t_{N+4}, {}^2t_{N+5} \right\} = \left\{ 0, 0, 0, s_1, \ s_2 \ldots s_{N-2}, \ s_{N-1}, L, L, L \right\}$, at the same time $p_i = 3\frac{u_{i+1}-u_i}{{}^2t_{i+4}-{}^2t_{i+1}}$ and $q_i = 3\frac{v_{i+1}-v_i}{{}^2t_{i+4}-{}^2t_{i+1}}$.

$$\begin{cases} u''(s) = \sum\limits_{i=0}^{n-2} B_i^1(s) l_i, \\ v''(s) = \sum\limits_{i=0}^{n-2} B_i^1(s) m_i, \end{cases} \tag{5}$$

where $B_i^1(s)$ are the basis functions of the linear B-spline defined on a knot vector $U_3 = \left\{ {}^3t_1, {}^3t_2 \ldots {}^3t_{N+2}, {}^3t_{N+3} \right\} = \left\{ 0, 0, s_1, \ s_2 \ldots s_{N-2}, \ s_{N-1}, L, L \right\}$. Meanwhile, it is obtained that $l_i = 2\frac{p_{i+1}-p_i}{{}^3t_{i+3}-{}^3t_{i+1}}$ and $m_i = 2\frac{q_{i+1}-q_i}{{}^3t_{i+3}-{}^3t_{i+1}}$ in this equation.

The basis functions of $u(s)$, $u'(s)$ and $u''(s)$ are illustrated in Fig. 4.

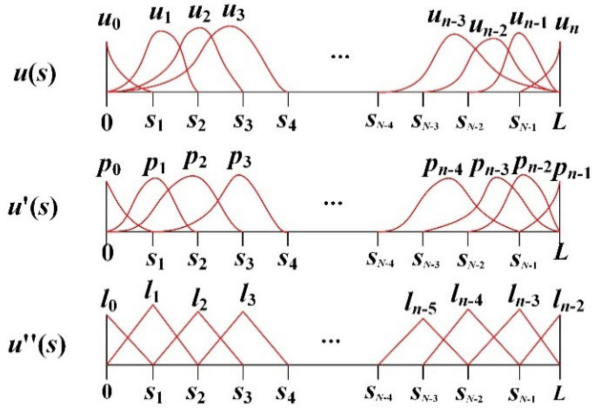

u_0 u_1 u_2 u_3
u_{n-3} u_{n-2} u_{n-1} u_n
$u(s)$
...
0 s_1 s_2 s_3 s_4 s_{N-4} s_{N-3} s_{N-2} s_{N-1} L
p_0 p_1 p_2 p_3
p_{n-4} p_{n-3} p_{n-2} p_{n-1}
$u'(s)$
...
0 s_1 s_2 s_3 s_4 s_{N-4} s_{N-3} s_{N-2} s_{N-1} L
l_0 l_1 l_2 l_3
l_{n-5} l_{n-4} l_{n-3} l_{n-2}
$u''(s)$
...
0 s_1 s_2 s_3 s_4 s_{N-4} s_{N-3} s_{N-2} s_{N-1} L

FIGURE 4. The basis functions of $u(s)$, $u'(s)$ and $u''(s)$.

In order to solve the $2n - 2$ unknown control points in the Eq. (3), $2n - 2$ Equations should be established. The equations could be established through the point collocation method, the subdomain method, the least square method or the Galerkin method. The last three methods are difficult to implement as the numerical integration of Eq. (1) is rather tedious. Thus, the point collocation method is preferred in this application. To obtain the $2n - 2$ equations, we choose $n - 1$ collocation points $0, s_1, \ s_2 \ldots s_{N-2}, \ s_{N-1}, L$, at which we enforce the approximate solution (3) satisfying the differential equation (1). Note the collocation points are in accordance with the knot points, which enables the evaluation of $u(s)$, $v(s)$ and their derivatives more cheaply.

In the subsequent part, the discretization of Eq. (1) is examined. The length of the interval $[s_i, \ s_{i+1}]$, $i = 0 \ldots \ N - 1$ is denoted as ss_i which is approximated by the chord length between the knot points.

With Eq. (4) and Eq. (5), the first order derivative and the second order derivative of $u(s)$ and $v(s)$ at the knot point s_i could be expressed as follows:

$$\begin{cases} u'(s_i) = \dfrac{ss_i}{ss_{i-1} + ss_i} p_i + \dfrac{ss_{i-1}}{ss_{i-1} + ss_i} p_{i+1}, \\ v'(s_i) = \dfrac{ss_i}{ss_{i-1} + ss_i} q_i + \dfrac{ss_{i-1}}{ss_{i-1} + ss_i} q_{i+1}. \end{cases} \quad 0 \le i \le N \tag{6}$$

$$\begin{cases} u''(s_i) = 2\dfrac{p_{i+1} - p_i}{ss_{i-1} + ss_i}, \\ v''(s_i) = 2\dfrac{q_{i+1} - q_i}{ss_{i-1} + ss_i}. \end{cases} \quad 0 \le i \le N \tag{7}$$

The relationship between p_i, q_i and the control point is stated as follows:

$$\begin{cases} p_i = 3\dfrac{u_{i+1} - u_i}{ss_{i-2} + ss_{i-1} + ss_i}, \\ q_i = 3\dfrac{v_{i+1} - v_i}{ss_{i-2} + ss_{i-1} + ss_i}. \end{cases} \quad 0 \le i \le N + 1 \tag{8}$$

In this paper, it is appointed that $ss_i \equiv 0$ when $i < 0$ or $i > N - 1$. For example, if i =0, then Eq. (8) will degenerate as the following equation:

$$\begin{cases} p_0 = 3\dfrac{u_1 - u_0}{ss_0}, \\ q_0 = 3\dfrac{v_1 - v_0}{ss_0}. \end{cases} \tag{9}$$

With Eq. (6), Eq. (7) and Eq. (8), through simplification, the discretization of Eq. (1) at the knot point s_i is denoted as:

$$\begin{cases} f_i = \dfrac{6}{ss_{i-1} + ss_i} \left[V_2 \left(u_{i+2} - u_{i+1} \right) - V_1 \left(u_{i+1} - u_i \right) \right] \\ \qquad + 9\Gamma_{11}^1 {H_1}^2 + 18\Gamma_{12}^1 H_1 H_2 + 9\Gamma_{22}^1 {H_2}^2 \\ \qquad + \dfrac{3k_g}{\sqrt{EG - F^2}} \left(FH_1 + GH_2 \right) = 0, \\ g_i = \dfrac{6}{ss_{i-1} + ss_i} \left[V_2 \left(v_{i+2} - v_{i+1} \right) - V_1 \left(v_{i+1} - v_i \right) \right] \\ \qquad + 9\Gamma_{11}^2 {H_1}^2 + 18\Gamma_{12}^2 H_1 H_2 + 9\Gamma_{22}^2 {H_2}^2 \\ \qquad - \dfrac{3k_g}{\sqrt{EG - F^2}} \left(EH_1 + FH_2 \right) = 0. \end{cases} \tag{10}$$

In Eq. (10), $H_1 = W_2 V_1 \left(u_{i+1} - u_i \right) + W_1 V_2 \left(u_{i+2} - u_{i+1} \right)$ and $H_2 = W_2 V_1 \left(v_{i+1} - v_i \right) + W_1 V_2 \left(v_{i+2} - v_{i+1} \right)$.

Meanwhile, $V_1 = \frac{1}{ss_{i-2}+ss_{i-1}+ss_i}$, $V_2 = \frac{1}{ss_{i-1}+ss_i+ss_{i+1}}$, $W_1 = \frac{ss_{i-1}}{ss_{i-1}+ss_i}$, $W_2 = \frac{ss_i}{ss_{i-1}+ss_i}$.

The boundary condition is converted as the following governing equation:

$$\begin{cases} f_{B1} = u_0 - u_A = 0, \\ g_{B1} = v_0 - v_A = 0, \end{cases} \\ \begin{cases} f_{B2} = u_{N+2} - u_B = 0, \\ g_{B2} = v_{N+2} - v_B = 0. \end{cases} \tag{11}$$

With Eq. (10) and Eq. (11), we have $2(N+3)$ nonlinear algebraic equations:

$$\boldsymbol{F} = (f_{B1},\ g_{B1},\ f_0,\ g_0 \ldots f_i,\ g_i \ldots f_N,\ g_N,\ f_{B2},\ g_{B2})^T = \boldsymbol{0}. \tag{12}$$

Eq. (12) can be computed by the Newton's method, if a starting vector $\boldsymbol{Y}^{(0)} = (u_0^{(0)},\ v_0^{(0)} \ldots\ u_i^{(0)},\ v_i^{(0)} \ldots\ u_{N+2}^{(0)},\ v_{N+2}^{(0)})^T$ is provided. The Newton iteration scheme is delivered by:

$$[\boldsymbol{J}^{(j)}]\Delta\boldsymbol{Y}^{(j)} = -F^{(j)}. \tag{13}$$

$$\boldsymbol{Y}^{(j+1)} = \boldsymbol{Y}^{(j)} + \lambda\Delta\boldsymbol{Y}^{(j)}. \tag{14}$$

where we let $0 < \lambda \leq 1$ to achieve more stability for the highly nonlinear problem. If $\lambda = 1$ the equation reduces to a regular Newton's method, while if $\lambda < 1$ the rate of convergence will be less than quadratic. Here $[\boldsymbol{J}^{(j)}]$ is the $2(N+3)$ by $2(N+3)$ Jacobian matrix of $\boldsymbol{F}^{(j)}$ with respect to $\boldsymbol{Y}^{(j)}$.

From Eq. (10) and Eq. (11), the Jacobian matrix should have the following seven-diagonal structure:

$$\begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & \\ & & \Delta_{2i+3,2i+1} & \bullet & \bullet & \bullet & \bullet & \Delta_{2i+3,2i+6} & & \\ & & \Delta_{2i+4,2i+1} & \bullet & \bullet & \bullet & \bullet & \Delta_{2i+4,2i+6} & & \\ & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & & & & & & & 1 & \\ & & & & & & & & & 1 \end{bmatrix}, \tag{15}$$

where Δ indicates non-zero value in the Jacobian matrix.

In the Jacobian matrix, the $2i+3$, $0 \leq i \leq N$ row is related to f_i at the knot point s_i, which is the function of u_i, v_i, u_{i+1}, v_{i+1}, u_{i+2}, v_{i+2}. Meanwhile, the $2i+4$, $0 \leq i \leq N$ row is related to g_i at this knot point, which is also the function of u_i, v_i, u_{i+1}, v_{i+1}, u_{i+2}, v_{i+2}. Recalling the definition of Jacobian matrix, the $2i+3$ row includes six non-zero values: $\Delta_{2i+3,2i+1}$, $\Delta_{2i+3,2i+2} \ldots \Delta_{2i+3,2i+6}$. Here, we have the following:

$$\begin{aligned} \Delta_{2i+3,2i+1} &= \frac{\partial f_i}{\partial u_i} = \frac{6V_1}{ss_{i-1}+ss_i} - W_2V_1 \\ &\quad \times \left(18\Gamma_{11}^1 H_1 + 18\Gamma_{12}^1 H_2 + \frac{3k_gF}{\sqrt{EG-F^2}}\right) \end{aligned}$$

$$\begin{aligned} \Delta_{2i+3,2i+2} &= \frac{\partial f_i}{\partial v_i} = -W_2V_1 \\ &\quad \times \left(18\Gamma_{12}^1 H_1 + 18\Gamma_{22}^1 H_2 + \frac{3k_gG}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+3,2i+3} &= \frac{\partial f_i}{\partial u_{i+1}} = -\frac{6\left(V_2+V_1\right)}{ss_{i-1}+ss_i} + \left(W_2V_1 - W_1V_2\right) \\ &\quad \times \left(18\Gamma_{11}^1 H_1 + 18\Gamma_{12}^1 H_2 + \frac{3k_gF}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+3,2i+4} &= \frac{\partial f_i}{\partial v_{i+1}} = \left(W_2V_1 - W_1V_2\right) \\ &\quad \times \left(18\Gamma_{12}^1 H_1 + 18\Gamma_{22}^1 H_2 + \frac{3k_gG}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+3,2i+5} &= \frac{\partial f_i}{\partial u_{i+2}} = \frac{6V_2}{ss_{i-1}+ss_i} + W_1V_2 \\ &\quad \times \left(18\Gamma_{11}^1 H_1 + 18\Gamma_{12}^1 H_2 + \frac{3k_gF}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+3,2i+6} &= \frac{\partial f_i}{\partial v_{i+2}} = W_1V_2 \\ &\quad \times \left(18\Gamma_{12}^1 H_1 + 18\Gamma_{22}^1 H_2 + \frac{3k_gG}{\sqrt{EG-F^2}}\right) \end{aligned}$$

At the same time, the $2i+4$ row also includes six non-zero values: $\Delta_{2i+4,2i+1}$, $\Delta_{2i+4,2i+2} \ldots \Delta_{2i+4,2i+6}$. Here,

$$\begin{aligned} \Delta_{2i+4,2i+1} &= \frac{\partial g_i}{\partial u_i} = -W_2V_1 \\ &\quad \times \left(18\Gamma_{11}^2 H_1 + 18\Gamma_{12}^2 H_2 - \frac{3k_gE}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+4,2i+2} &= \frac{\partial g_i}{\partial v_i} = \frac{6V_1}{ss_{i-1}+ss_i} - W_2V_1 \\ &\quad \times \left(18\Gamma_{12}^2 H_1 + 18\Gamma_{22}^2 H_2 - \frac{3k_gF}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+4,2i+3} &= \frac{\partial g_i}{\partial u_{i+1}} = \left(W_2V_1 - W_1V_2\right) \\ &\quad \times \left(18\Gamma_{11}^2 H_1 + 18\Gamma_{12}^2 H_2 - \frac{3k_gE}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+4,2i+4} &= \frac{\partial g_i}{\partial v_{i+1}} = -\frac{6\left(V_2+V_1\right)}{ss_{i-1}+ss_i} + \left(W_2V_1 - W_1V_2\right) \\ &\quad \times \left(18\Gamma_{12}^2 H_1 + 18\Gamma_{22}^2 H_2 - \frac{3k_gF}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+4,2i+5} &= \frac{\partial g_i}{\partial u_{i+2}} = W_1V_2 \\ &\quad \times \left(18\Gamma_{11}^2 H_1 + 18\Gamma_{12}^2 H_2 - \frac{3k_gE}{\sqrt{EG-F^2}}\right) \\ \Delta_{2i+4,2i+6} &= \frac{\partial g_i}{\partial v_{i+2}} = \frac{6V_2}{ss_{i-1}+ss_i} + W_1V_2 \\ &\quad \times \left(18\Gamma_{12}^2 H_1 + 18\Gamma_{22}^2 H_2 - \frac{3k_gF}{\sqrt{EG-F^2}}\right) \end{aligned}$$

C. THE INITIAL APPROXIMATION

As illustrated in Fig. 5, the two end points in the parametric space are connected by a straight line or a circular arc and a uniform mesh is defined by a set of discrete points. These points are the initial knot points $(u(s_i)^{(0)},\ v(s_i)^{(0)})$ and they are utilized to compute the initial control points $(u_i^{(0)},\ v_i^{(0)})$. In this paper, the right superscript (0) indicates the initial step.

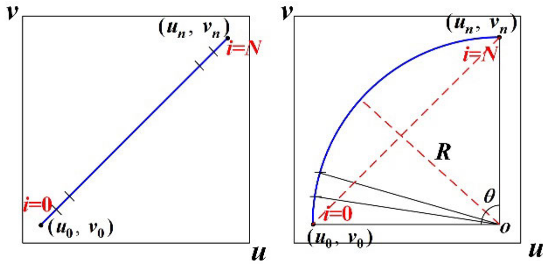


FIGURE 5. Straight line approximation and circular arc approximation.

Straight line approximation is the simplest and most often provides a good initial approximation. Wherein, the initial knot points is obtained as:

$$\begin{cases} u(s_i)^{(0)} = u_0 + i\dfrac{u_n - u_0}{N}, \\ v(s_i)^{(0)} = v_0 + i\dfrac{v_n - v_0}{N}. \end{cases} \quad 0 \le i \le N \tag{16}$$

The circular arc approximation is more reliable than the straight line approximation when there exists more than one solutions [14]. For this kind of initial approximation, an auxiliary circular with radius R is drawn between the two end points. Through computation, the center coordinates of the arc is expressed as (u_R, v_R). At the same time, the arc angles corresponding to (u_0, v_0) and (u_n, v_n) are denoted as a and β, respectively. Thereby, the initial knot points is delivered as:

$$\begin{cases} u(s_i)^{(0)} = u_R + R\cos\left(a + i\dfrac{\beta - a}{N}\right), \\ v(s_i)^{(0)} = v_R + R\sin\left(a + i\dfrac{\beta - a}{N}\right). \end{cases} \quad 0 \le i \le N \tag{17}$$

With Eq. (2) and Eq. (3), the relationship between the knot point $(u(s_i),\ v(s_i))$ at s_i and the control point $(u_i,\ v_i)$ is stated as:

$$\begin{cases} A_i u_i + B_i u_{i+1} + C_i u_{i+2} = u(s_i), \\ A_i v_i + B_i v_{i+1} + C_i v_{i+2} = v(s_i). \end{cases} \quad i = 1 \ldots N-1 \tag{18}$$

Here $A_i,\ B_i,\ C_i$ are acquired through the following formulas:

$$\begin{cases} A_i = \dfrac{1}{ss_{i-1} + ss_i}\left(\dfrac{{ss_i}^2}{ss_{i-2} + ss_{i-1} + ss_i}\right), \\ B_i = \dfrac{1}{ss_{i-1} + ss_i} \\ \left[\dfrac{(ss_{i-2} + ss_{i-1})\, ss_i}{ss_{i-2} + ss_{i-1} + ss_i} + \dfrac{(ss_i + ss_{i+1})\, ss_{i-1}}{ss_{i-1} + ss_i + ss_{i+1}}\right], \\ C_i = \dfrac{1}{ss_{i-1} + ss_i}\left(\dfrac{{ss_{i-1}}^2}{ss_{i-1} + ss_i + ss_{i+1}}\right). \end{cases}$$
$$i = 1 \ldots N-1 \tag{19}$$

In Eq. (18), there are $2N+2$ unknown values while $2N-2$ formulas. In order to evaluate $(u_i^{(0)},\ v_i^{(0)})$, another four formulas is obtained by providing the tangent vector of the initial B-spline curve at the boundary points:

$$\begin{cases} (p_0^{(0)},\ q_0^{(0)}) = (P_A,\ Q_A), \\ (p_{N+1}^{(0)},\ q_{N+1}^{(0)}) = (P_B,\ Q_B). \end{cases} \tag{20}$$

With Eq. (8), Eq. (18) and Eq. (20), $u_i^{(0)}$ can be solved conveniently through the following matrix operation:

$$\begin{bmatrix} u_1^{(0)} \\ \cdots \\ u_i^{(0)} \\ \cdots \\ u_{N+1}^{(0)} \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ \cdot & \cdot & \cdot & & \\ & A_i^{(0)} & B_i^{(0)} & C_i^{(0)} & \\ & & \cdot & \cdot & \cdot \\ & & & & 1 \end{bmatrix}^{-1}$$
$$\times \begin{bmatrix} u_0 + \dfrac{1}{3}P_A.ss_0^{(0)} \\ \cdots \\ u(s_i)^{(0)} \\ \cdots \\ u_{N+2} - \dfrac{1}{3}Q_A.ss_{N-1}^{(0)} \end{bmatrix}. \tag{21}$$

At the same time, the formula for $v_i^{(0)}$ can be obtained by replacing symbol u with symbol v, P_A with P_B, and Q_A with Q_B in Eq. (21). Note that Eq. (21) will be utilized only once in the computation.

D. UPDATING THE KNOT POINTS

During the iteration, the interval length ss_i is variable as the b-spline curve is evolving. As the interval length is approached by the chord length between the knot points, these points should also be updated in the iteration.

The knot points $(u(s_i)^{(j+1)},\ v(s_i)^{(j+1)})$ in iteration $j+1$ could be evaluated with Eq. (18) directly when the control points $(u_i^{(j+1)},\ v_i^{(j+1)})$ in this iteration is acquired through Eq. (14). However, it is found that the following correction procedure would be more effective.

With Eq. (18), the relationship between the knot point and the control point can be transformed into a linear equation:

$$\begin{cases} L_i = u(s_i) - A_i u_i - B_i u_{i+1} - C_i u_{i+2} = 0, \\ M_i = v(s_i) - A_i v_i - B_i v_{i+1} - C_i v_{i+2} = 0. \end{cases}$$
$$i = 1 \ldots N-1 \tag{22}$$

With Eq.(22), once the updating vector $\Delta Y^{(j)}$ of the control point is evaluated through Eq.(13), the knot point $(u(s_i),\ v(s_i))$ is acquired by The Newton iteration scheme:

$$\begin{cases} \Delta u(s_i)^{(j)} = {A_i}^{(j)} \Delta {u_i}^{(j)} + {B_i}^{(j)} \Delta {u_{i+1}}^{(j)} \\ + {C_i}^{(j)} \Delta {u_{i+2}}^{(j)} - {L_i}^{(j)}, \\ \Delta v(s_i)^{(j)} = {A_i}^{(j)} \Delta {v_i}^{(j)} + {B_i}^{(j)} \Delta {v_{i+1}}^{(j)} \\ + {C_i}^{(j)} \Delta {v_{i+2}}^{(j)} - {M_i}^{(j)}. \end{cases} \tag{23}$$

$$\begin{cases} u(s_i)^{(j+1)} = u(s_i)^{(j)} + \lambda \Delta u(s_i)^{(j)}, \\ v(s_i)^{(j+1)} = v(s_i)^{(j)} + \lambda \Delta v(s_i)^{(j)}, \end{cases} \tag{24}$$

where, $i = 1\ \ldots\ N-1$, and j indicates the iteration times.

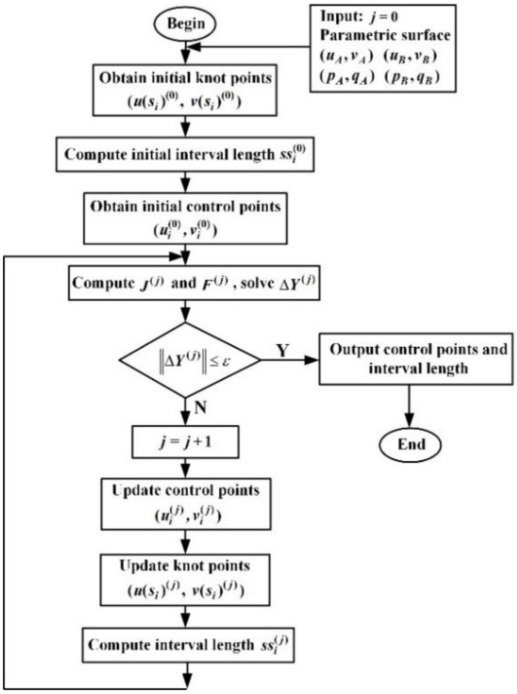

Begin
Input: $j=0$
Parametric surface
(u_A, v_A) (u_B, v_B)
(p_A, q_A) (p_B, q_B)
Obtain initial knot points $(u(s_i)^{(0)},\ v(s_i)^{(0)})$
Compute initial interval length $ss_i^{(0)}$
Obtain initial control points $(u_i^{(0)}, v_i^{(0)})$
Compute $J^{(j)}$ and $F^{(j)}$, solve $\Delta Y^{(j)}$
$\|\Delta Y^{(j)}\| \le \varepsilon$
Y
Output control points and interval length
End
N
$j=j+1$
Update control points $(u_i^{(j)}, v_i^{(j)})$
Update knot points $(u(s_i)^{(j)},\ v(s_i)^{(j)})$
Compute interval length $ss_i^{(j)}$

FIGURE 6. Flow chart of our algorithm.

The flow chart of our algorithm is illustrated in Fig. 6. We detail our approach into Algorithm 1 as follows:

As seen in Fig. 6 and Algorithm 1, the input values are two boundary points on the target parametric surface. After several iterations and a predetermined accuracy level is reached, the control points and interval length are output to build the solution of Eq. (1).

IV. ERROR ANALYSIS

In this section, the computation accuracy of the proposed method is analyzed. First of all, let $\widetilde{u}(s)$, $\widetilde{v}(s)$ denote the exact solution of Eq. (1).

With Eq. (6)-Eq. (8), we can obtain the following:

$$u'(s_i) = \frac{3}{ss_{i-1}+ss_i} \times \left[\frac{ss_{i-1}\,(u_{i+2}-u_{i+1})}{ss_{i-1}+ss_i+ss_{i+1}} + \frac{ss_i\,(u_{i+1}-u_i)}{ss_{i-2}+ss_{i-1}+ss_i} \right]. \tag{25}$$

$$u''(s_i) = \frac{6}{ss_{i-1}+ss_i} \times \left(\frac{u_{i+2}-u_{i+1}}{ss_{i-1}+ss_i+ss_{i+1}} - \frac{u_{i+1}-u_i}{ss_{i-2}+ss_{i-1}+ss_i} \right). \tag{26}$$

The Eq. (18) can be rewritten as follows:

$$u(s_i) = \frac{1}{ss_{i-1}+ss_i} \left[2u_{i+1} + \frac{{ss_{i-1}}^2\,(u_{i+2}-u_{i+1})}{ss_{i-1}+ss_i+ss_{i+1}} - \frac{{ss_i}^2\,(u_{i+1}-u_i)}{ss_{i-2}+ss_{i-1}+ss_i} \right]. \tag{27}$$

Algorithm 1 Solving the Boundary Value Problem of Curves With Prescribed Geodesic Curvature Based on a Cubic B-Spline Element Method

```
Input: A parametric surface $S(u, v)$, two boundary points $(u_A, v_A)$ and $(u_B, v_B)$, number of knot points, $j = 0$, an error tolerance $\varepsilon$.
Output: A B-spline curve with prescribed geodesic curvature to connect the user-specified boundary points.
Obtain the initial knot points $(u(s_i)^{(0)},\ v(s_i)^{(0)})$ by using the straight line approximation (16) or circular arc approximation (17);
Compute the chord lengths between the initial knot points $(u(s_i)^{(0)},\ v(s_i)^{(0)})$, and use them as the initial interval lengths $ss_i^{(0)}$;
Compute the initial control points $(u_i^{(0)},\ v_i^{(0)})$ by using Eq. (21);
While 1 do
    Evaluate the Jacobian matrix $[\boldsymbol{J}^{(j)}]$ and $\boldsymbol{F}^{(j)}$ by using Eq. (15) and Eq. (12), respectively;
    Solve $\Delta\boldsymbol{Y}^{(j)}$ by using Eq. (13);
    If $\|\Delta\boldsymbol{Y}^{(j)}\| \le \varepsilon$ Break;
    Else Do
        Let $j = j + 1$;
        Update the control points $(u_i^{(j)},\ v_i^{(j)})$ in the iteration $j$ by using Eq. (14);
        Update the knot points $(u(s_i)^{(j)},\ v(s_i)^{(j)})$ in the iteration $j$ by using Eq. (23) and Eq. (24);
        Compute the chord lengths between the knot points $(u(s_i)^{(j)},\ v(s_i)^{(j)})$, and use them as the interval lengths $ss_i^{(j)}$ in the iteration $j$;
End While
Return the control points and the interval length to build the solution.
```

Then, the following relationships can be obtained:

$$-\frac{3}{{ss_{i-1}}^2}u(s_{i-1}) + \left(\frac{3}{{ss_{i-1}}^2} - \frac{3}{{ss_i}^2}\right)u(s_i) + \frac{3}{{ss_i}^2}u(s_{i+1}) = \frac{1}{ss_{i-1}}u'(s_{i-1}) + \left(\frac{2}{ss_{i-1}} + \frac{2}{ss_i}\right)u'(s_i) + \frac{1}{ss_i}u'(s_{i+1}). \tag{28}$$

$$\frac{6}{ss_{i-1}}u(s_{i-1}) - \frac{6(ss_{i-1}+ss_i)}{ss_{i-1}ss_i}u(s_i) + \frac{6}{ss_i}u(s_{i+1}) = ss_{i-1}u''(s_{i-1}) + 2(ss_{i-1}+ss_i)u''(s_i) + ss_iu''(s_{i+1}). \tag{29}$$

With the Taylor Expansion at the collocation point, the following equations are expressed:

$$\begin{cases} u(s_{i-1}) = \widetilde{u}(s_{i-1}) = \widetilde{u}(s_i) - ss_{i-1}\widetilde{u}'(s_i) + \frac{1}{2}{ss_{i-1}}^2\widetilde{u}''(s_i) \\ -\frac{1}{6}{ss_{i-1}}^3\widetilde{u}'''(s_i) + \frac{1}{24}{ss_{i-1}}^4\widetilde{u}''''(s_i) + O(h^5), \\ \qquad u(s_i) = \widetilde{u}(s_i), \\ u(s_{i+1}) = \widetilde{u}(s_{i+1}) = \widetilde{u}(s_i) + ss_i\widetilde{u}'(s_i) + \frac{1}{2}{ss_i}^2\widetilde{u}''(s_i) \\ +\frac{1}{6}{ss_i}^3\widetilde{u}'''(s_i) + \frac{1}{24}{ss_i}^4\widetilde{u}''''(s_i) + O(h^5), \end{cases} \tag{30}$$

where $O(\,)$ is the big O notation.

$$\begin{cases} u'(s_{i-1}) = u'(s_i) - ss_{i-1}u''(s_i) + \frac{1}{2}ss_{i-1}{}^2u'''(s_i) \\ -\frac{1}{6}ss_{i-1}{}^3u''''(s_i) + O(h^5), \\ u'(s_{i+1}) = u'(s_i) + ss_iu''(s_i) + \frac{1}{2}ss_i{}^2u'''(s_i) \\ +\frac{1}{6}ss_i{}^3u''''(s_i) + O(h^5). \end{cases} \tag{31}$$

$$\begin{cases} u''(s_{i-1}) = u''(s_i) - ss_{i-1}u'''(s_i) \\ +\frac{1}{2}ss_{i-1}{}^2u''''(s_i) + O(h^5), \\ u''(s_{i+1}) = u''(s_i) + ss_iu'''(s_i) \\ +\frac{1}{2}ss_i{}^2u''''(s_i) + O(h^5). \end{cases} \tag{32}$$

Substitute Eq. (30) and Eq. (31) into Eq. (28), through simplification, the following relationship is deduced:

$$\begin{aligned} &\left[1 + \frac{ss_{i-1}ss_i}{6}D^2 + \frac{ss_{i-1}ss_i(ss_i - ss_{i-1})}{18}D^3 \right. \\ &\left. + \frac{ss_{i-1}ss_i(ss^3_{i-1} + ss^3_i)}{72(ss_{i-1} + ss_i)}D^4 \dots \right] u'(s_i) \\ &= \left[D + \frac{ss_{i-1}ss_i}{6}D^3 + \frac{ss_{i-1}ss_i(ss_i - ss_{i-1})}{24}D^4 \right. \\ &\left. + \frac{ss_{i-1}ss_i(ss_{i-1}{}^3 + ss_i{}^3)}{120(ss_{i-1} + ss_i)}D^5 \dots \right] \widetilde{u}(s_i), \end{aligned} \tag{33}$$

where $D \equiv \frac{\mathrm{d}}{\mathrm{d}s}$.

With Eq. (29), Eq. (30) and Eq. (32), through simplification, the following formula is obtained:

$$\begin{aligned} &\left[1 + \frac{(ss_i{}^2 - ss_{i-1}{}^2)}{3(ss_{i-1} + ss_i)}D + \frac{(ss_i{}^3 + ss_{i-1}{}^3)}{6(ss_{i-1} + ss_i)}D^2 + \cdots .\right] u''(s_i) \\ &= \left[D^2 + \frac{(ss_i{}^2 - ss_{i-1}{}^2)}{3(ss_{i-1} + ss_i)}D^3 + \frac{(ss_i{}^3 + ss_{i-1}{}^3)}{12(ss_{i-1} + ss_i)}D^4 + \cdots\right] \widetilde{u}(s_i). \end{aligned} \tag{34}$$

With Eq. (33) and Eq. (34), through simplification, the following relations can be derived:

$$\begin{cases} u'(s_i) = \widetilde{u}'(s_i) - \frac{ss_{i-1}ss_i(ss_i - ss_{i-1})}{72}\widetilde{u}^4(s_i) \\ -\frac{ss_{i-1}ss_i(ss_{i-1}{}^3 + ss_i{}^3)}{180(ss_{i-1} + ss_i)}\widetilde{u}^5(s_i) + O(h^4), \\ u''(s_i) = \widetilde{u}''(s_i) - \frac{(ss_i{}^3 + ss_{i-1}{}^3)}{12(ss_{i-1} + ss_i)}\widetilde{u}^4(s_i) + O(h^3). \end{cases} \tag{35}$$

Now, we define the computation error of u as $u\_e(s) = u(s) - \widetilde{u}(s)$, and with the Taylor Expansion, it is acquired:

$$\begin{aligned} u\_e(s_i + \theta ss_i) &= u(s_i) - \widetilde{u}(s_i) + \theta ss_i\left[u'(s_i) - \widetilde{u}'(s_i)\right] \\ &\quad + \frac{1}{2}\theta^2 ss_i{}^2\left[u''(s_i) - \widetilde{u}''(s_i)\right] + \cdots, \end{aligned} \tag{36}$$

where $0 \le \theta \le 1$.

Substitute Eq. (35) into Eq. (36), the computation error of u is expressed as:

$$\begin{aligned} &u\_e(s_i + \theta ss_i) \\ &= \left[-\theta\frac{ss_{i-1}ss_i{}^2(ss_i - ss_{i-1})}{72} - \theta^2\frac{ss_i{}^2(ss_i{}^3 + ss_{i-1}{}^3)}{24(ss_{i-1} + ss_i)}\right] \\ &\quad \times \widetilde{u}^4(s_i) + O(h^5). \end{aligned} \tag{37}$$

Similarly, the computation error of v is solved as:

$$\begin{aligned} &v\_e(s_i + \theta ss_i) \\ &= \left[-\theta\frac{ss_{i-1}ss_i{}^2(ss_i - ss_{i-1})}{72} - \theta^2\frac{ss_i{}^2(ss_i{}^3 + ss_{i-1}{}^3)}{24(ss_{i-1} + ss_i)}\right] \\ &\quad \times \widetilde{v}^4(s_i) + O(h^5), \end{aligned} \tag{38}$$

where $v\_e(s) = v(s) - \widetilde{v}(s)$ denotes the computation error of v.

With Eq. (37) and Eq. (38), it is deduced that our proposed approach is generally $O(h^4)$ accurate.

V. VALIDATION

It is difficult for the discrete approximation method of geodesic curves to determine how the curves with non-zero geodesic curvature should pass on surfaces. However, the numerical approach introduced by Maekawa, Kasap and Chen for BVP of geodesic could be extended to curves with non-vanishing geodesic curvature directly. Both of Maekawa's and Kasap's methods are based on the finite difference method, and an interpolation technique is needed to ensure the continuity of desired curve. Chen's algorithm utilizes an optimization strategy to find geodesics. In Appendix, we prove the optimization strategy is equivalent to solving the differential equation of geodesics based on the Galerkin method.

In the subsequent part, the proposed method is compared with Maekawa's method, Kasap's method and Chen's geodesic-like curve as these approaches are elegant and aim for the smooth parametric surface. Curves with prescribed geodesic curvature between two boundary points on several parametric surfaces are constructed with the methods. For the purpose of equity, all of the methods share the same number of iterations and the time costs are recorded. In order to measure the accuracy of different algorithms, the length difference between the computed curve and ideal curve is evaluated as:

$$\text{Length error} = \frac{\left|\text{Length}_{\text{computed}} - \text{Length}_{\text{ideal}}\right|}{\text{Length}_{\text{ideal}}} \times 100\%, \tag{39}$$

where $\text{Length}_{\text{computed}}$ and $\text{Length}_{\text{ideal}}$ indicate the length of the computed curve and the target curve, respectively. Here, the length of the computed curve is approximated by the sum of the chord length between the knot points. The ideal curves with prescribed geodesic curvature on B-spline surfaces are not readily available, so we use the curves computed through our method using more knot points and iteration times as the ideal curves.

Next, various surfaces are adopted to validate the proposed method. For simplicity, the straight line approximation is adopted. All of the codes are implemented in MATLAB 2016b with NURBS Toolbox 1.0 and run on a personal computer [Intel Core (TM) i5 Duo Processor, 2.8GHz, 16GB memory] to test the accuracy and the efficiency.

As illustrated in Fig. 7, the first example computes curves with prescribed geodesic curvature on a sphere surface. The target curves include a geodesic curve and a curve with constant geodesic curvature ($k_g = 1/4000$). The length of the two curves are 3141.5927 and 3261.9341, respectively. The length variations of the computed curves with respect to the iteration times for the four methods are recorded in Table 1. The differences on the accuracy and time costs of the four approaches are demonstrated in Table. 2.

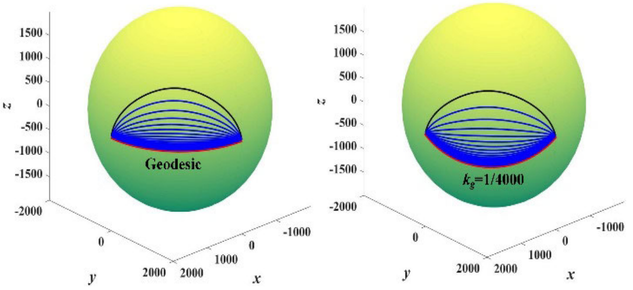


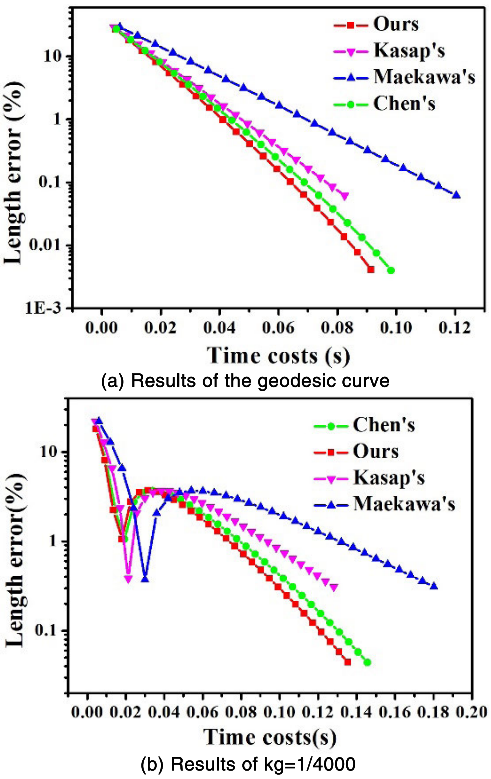
FIGURE 7. Curves with prescribed geodesic curvature on a sphere surface.

TABLE 1. Length variations with respect to the iterations in example 1.

Method	Knot points: 101 λ=0.2 kg=0			
Iterations	1	5	10	15
Maekawa's	4054.37	3400.88	3193.09	3151.67
Kasap's	4054.39	3398.87	3192.67	3151.60
Chen's	3986.26	3310.80	3161.19	3143.54
Ours	3987.56	3310.86	3161.44	3143.57
Method	Knot points: 101 λ=0.2 kg=1/4000			
Iterations	1	5	10	15
Maekawa's	3974.77	3249.84	3143.33	3183.37
Kasap's	3974.79	3249.43	3143.20	3183.38
Chen's	3848.35	3171.88	3166.23	3220.13
Ours	3848.95	3172.00	3166.83	3219.82

TABLE 2. Results of comparison on accuracy and time costs in example 1.

Method	Maekawa's	Kasap's	Chen's	Ours
Knot points: 101 λ=0.2 Iteration times:20 kg=0				
Time costs	0.1204	0.0824	0.0982	0.0915
Length	3143.54	3143.53	3141.72	3141.72
Length error	0.0620%	0.0616%	0.0040%	0.0041%
Knot points: 101 λ=0.2 Iteration times:30 kg=1/4000				
Time costs	0.1802	0.1281	0.1457	0.1355
Length	3251.83	3251.84	3260.50	3260.49
Length error	0.310%	0.310%	0.0440%	0.0443%



(a) Results of the geodesic curve

(b) Results of kg=1/4000

FIGURE 8. Results of comparison on efficiency in example 1.

As shown in Fig. 7 and Table. 1, the initial curve approaches the desire curve gradually with the iteration. For geodesic, with the increase of the number of iterations, the length of the curve decreases and the error of the curve length decreases. This is due to that the geodesic curve generally minimizes the distance between two points on surfaces. However, for the curve with prescribed geodesic curvature, the error of curve length will gradually decrease after a certain number of iterations, and the computed curve is relatively close to the target curve at this time. With Table. 2, the accuracy of our method is comparable with Chen's method, and the accuracy of Maekawa's method is close to Kasap's method. At the same time, the accuracy of the first two methods is significantly better than that of the latter two methods. In Table 2, the computation accuracy of the curve with prescribed geodesic curvature is worse than that of geodesic curve. This is mainly due to that the differential equation of the former curve is more complicated than the latter. As can be seen form Table. 2, Kasap's method is the fastest, while Maekawa's approach is the slowest. Meanwhile, our algorithm is faster than Chen's method.

In order to show the efficiency of different algorithms, Fig. 8 records their computation time required to achieve a given level of accuracy. With this figure, it is noticed that our approach is always able to achieve better accuracy in a same computation time, which indicates that our method is more efficient.

As shown in Fig. 9, the second example evaluates curves with prescribed geodesic curvature on a cubic B-spline surface. The target curves also include a geodesic curve and a curve with constant geodesic curvature ($k_g = 1/500$). The length of the two curves are 199.3348 and 200.9313, respectively.

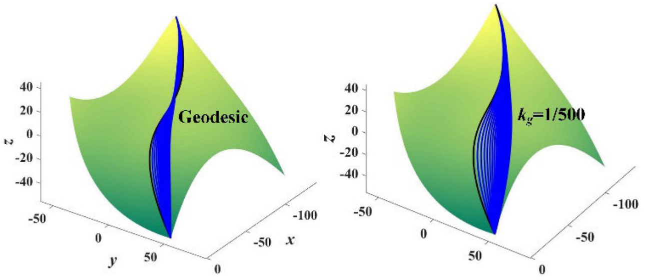


FIGURE 9. Curves with prescribed geodesic curvature on a cubic B-spline surface.

The length variations of the computed curves with respect to the iteration times for the four methods are recorded in Table 3. The differences on the accuracy and time costs of the four approaches are demonstrated in Table. 4. The Results of comparison on efficiency are illustrated in Fig. 10.

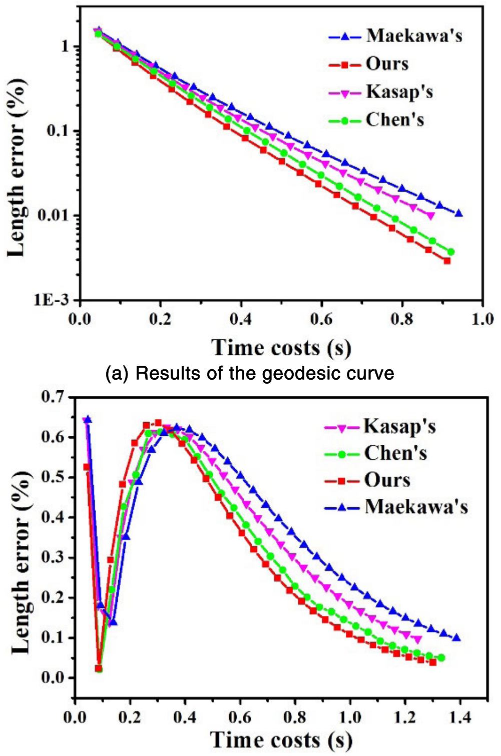
TABLE 3. Length variations with respect to the iterations in example 2.

Method	Knot points: 101 λ=0.2 kg=0			
Iterations	1	5	10	15
Maekawa's	202.353	200.207	199.558	199.400
Kasap's	202.351	200.206	199.556	199.399
Chen's	202.139	200.058	199.482	199.368
Ours	202.139	199.955	199.453	199.361
Method	Knot points: 101 λ=0.2 kg=1/500			
Iterations	1	5	10	15
Maekawa's	202.222	199.953	199.729	200.065
Kasap's	202.221	199.953	199.724	200.060
Chen's	201.987	199.833	199.802	200.254
Ours	201.986	199.756	199.843	200.288

TABLE 4. Results of comparison on accuracy and time costs in example 2.

Method	Maekawa's	Kasap's	Chen's	Ours
Knot points: 101 λ=0.2 Iteration times:20 kg=0				
Time costs	0.940	0.870	0.921	0.912
Length	199.3555	199.3550	199.3422	199.3405
Length error	0.0104%	0.0101%	0.0037%	0.0029%
Knot points: 101 λ=0.2 Iteration times:30 kg=1/500				
Time costs	1.386	1.245	1.332	1.301
Length	200.7341	200.7357	200.8313	200.8546
Length error	0.0981%	0.0973%	0.0498%	0.0382%

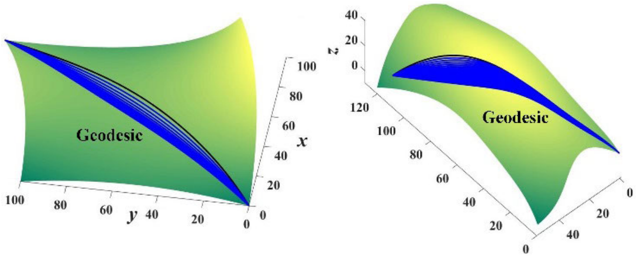
As can be seen from Table. 3, Table. 4 and Figure. 10, the experimental results reflect similar trends on accuracy and efficiency as those in the first example. However, it is also noticed that the divergence between our method and Chen's method is more obvious, and our method shows superiority on accuracy.



(a) Results of the geodesic curve

(b) Results of kg=1/500

FIGURE 10. Results of comparison on efficiency in example 2.



(a) A 2×2 B-spline surface

(b) A 6×7 B-spline surface

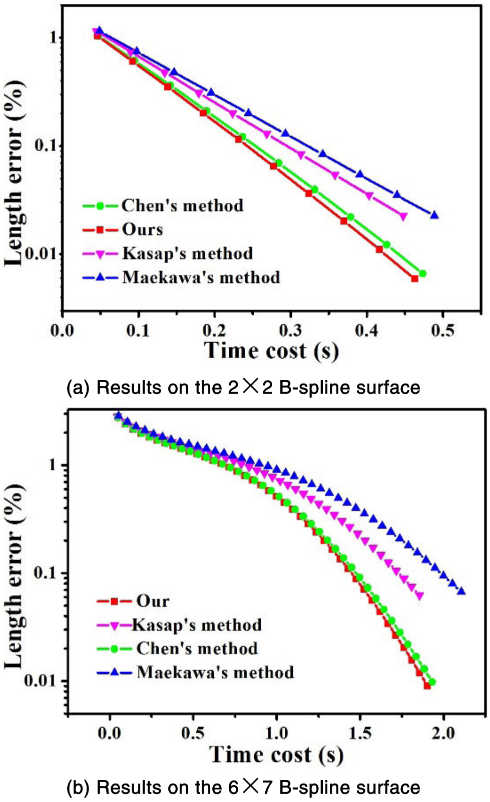
FIGURE 11. Computation of geodesic curves on B-spline surfaces.

In the third example, the influence of underlying parametric surfaces on the computation process is investigated. As illustrated in Fig. 11, curves with zero geodesic curvature are computed on a 2 × 2 B-spline surface and a 6 × 7 B-spline surface. The curvature of the latter surface is larger than that of the former surface. As the length error is related to the length of the ideal curve, the two target curves share an approximately equal length for equity, which are 145.7069 and 140.4017, respectively.

With Fig. 12 and Table. 5, the differences between the four algorithms are similar to that in the first two examples. Meanwhile, it is noticed that more iterations and time consumption are needed to achieve a given level of accuracy on the surface with large curvature for all of the methods. In Fig. 13, the influence of the number of knot points on the performance of our proposed method is investigated.

TABLE 5. Results of comparison on accuracy and time costs in example 3.

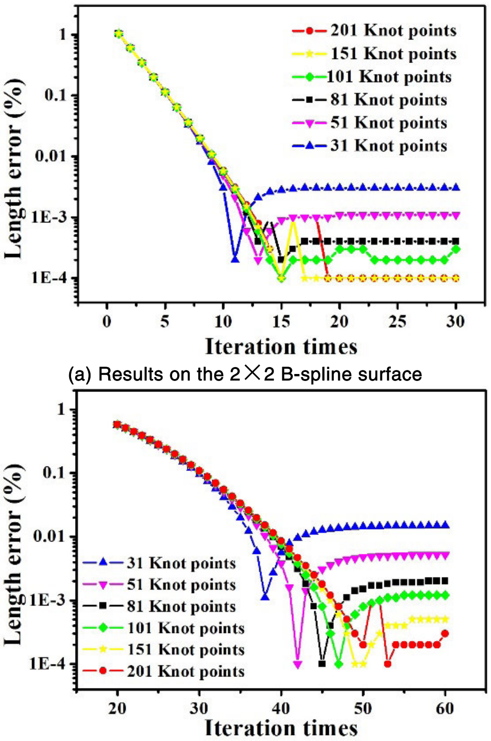
Method	Maekawa's	Kasap's	Chen's	Ours
Knot points: 101 $\lambda=0.2$ Iteration times:10 kg=0				
Time costs	0.489	0.447	0.474	0.463
Length	145.7395	145.7395	145.7162	145.7152
Length error	0.0224%	0.0224%	0.0064%	0.0057%
Knot points: 101 $\lambda=0.2$ Iteration times:40 kg=0				
Time costs	2.105	1.855	1.932	1.905
Length	140.4935	140.4873	140.4137	140.4125
Length error	0.0654%	0.0610%	0.0085%	0.0077%



(a) Results on the 2×2 B-spline surface

(b) Results on the 6×7 B-spline surface

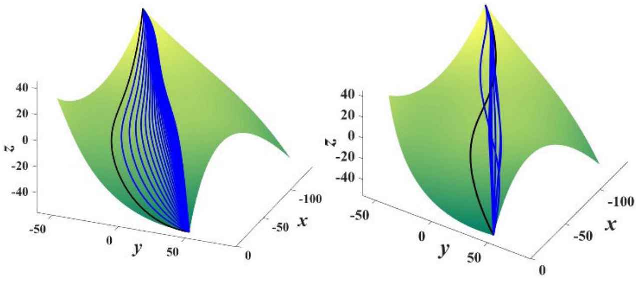
FIGURE 12. Results of comparison on efficiency in example 3.



(a) Results on the 2×2 B-spline surface

(b) Results on the 6×7 B-spline surface

FIGURE 13. Influence of the number of knot points on the computation accuracy.



(a) Circular arc approximation

(b) Correction factor $\lambda=1.0$

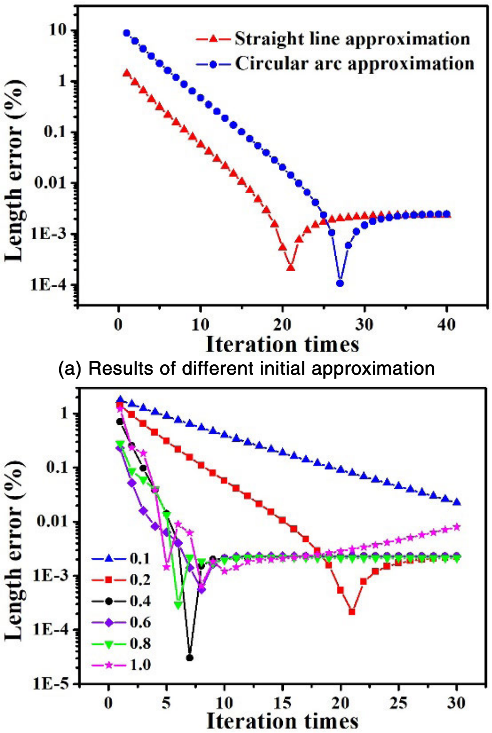
FIGURE 14. Computation of geodesic curves on the cubic B-spline surface.

With Fig. 13, it is known that the computation accuracy generally increases with the increase of knot points. At the same time, it is noticed that it requires more iteration times to achieve a stable lever of length error for the computation with more knot points. In Fig. 13(a), no obvious improvement of computation accuracy is noticed when the number of knot points is increased from 150 to 200, which is not the case in Fig. 13(b). Thus, for a surface with large curvature, more iteration times and knot points are preferred to achieve a given level of computation accuracy.

The last example evaluates curves with zero geodesic curvature by using our proposed approach on the cubic B-spline surface investigated in Example 2. The number of knot points is 101. In Fig. 14(a), the circular arc approximation is adopted and the correction factor λ is set as 0.2. In Fig. 14(b), the straight line approximation is utilized and the correction factor λ is set as 1.0. The corresponding experimental results are illustrated in Fig. 15(a) and Fig. 15(b), respectively.

With Figure. 15(a), it can be seen that the straight line approximation provides better initial approximation than the circular arc approximation as it acquires better computation accuracy in a same iteration times. The main advantage of circular arc approximation is that it is more reliable when there exists more than one target path. In Figure. 15(b), it is found that with the increase of the value of λ, the convergence speed generally increases. However, the fluctuation of the computation accuracy is more significant. When the value of λ is 1.0, the Eq. (13) reduces to a regular Newton's method, which is difficult to find a reliable solution as seen in Fig. 14(b).

To sum up, our proposed method is generally more accurate than the algorithms proposed by Maekawa, Kasap and



(a) Results of different initial approximation

(b) Results of different correction factors

FIGURE 15. **Experimental results on the cubic B-spline surface.**

Chen. This is mainly due to that our approach has third order accuracy, while Maekawa and Kasap's finite difference methods are with second order accuracy. Meanwhile, Chen's optimization strategy is based on a quadratic B-spline basis functions. As for the time costs, Maekawa's method is slow as the iteration process involves the coordinates and the directions of the target curve. In this case, the Jacobian matrix is about twice the size of that in the other three methods. Kasap's method is the fastest thanks to that it is independent of updating the knot points. The experimental results also demonstrate followings: Firstly, our approach is always able to achieve better accuracy in a same computation time, which indicates that our method is more efficient. Secondly, the computation accuracy of the curve with non-zero geodesic curvature is worse than that of geodesic curve. Thirdly, for a surface with large curvature, more iteration times and knot points are preferred to achieve a given level of computation accuracy. Lastly, the straight line approximation generally provides better initial approximation than the circular arc approximation. At the same time, large correction factors λ are more likely to result in a notable fluctuation of the computation accuracy.

VI. DISCUSSION AND CONCLUSION

In this paper, a non-uniform cubic b-spline element method is proposed for solving the boundary value problem (BVP) of curves with prescribed geodesic curvature. Simulation and experimental results show that the proposed method is generally more efficient than existing numerical methods.

The advantages of the proposed method are concluded as follows:

1) The method could cope with the BVP for curves no matter their geodesic curvature vanishes or not.
2) The method has third order accuracy, which shows more superiorities than existing numerical methods.
3) The computed curves are natural and smooth, and no interpolation technique is needed to ensure the continuity of target curves.
4) The collocation points are always in accordance with the knot points, which facilitates the deduction process a lot.

This paper focuses on providing an efficient numerical framework for constructing curves with prescribed geodesic curvature between two points on parametric surfaces. In future research, the proposed method would be applied to path optimization in ATP. The desired trajectories designed with the proposed method could be adopted to steer the composite tapes such that eliminating the gaps and improving the quality of the resulting laminate.

APPENDIX

Chen's algorithm utilizes an optimization strategy to find geodesic curves on parametric surfaces. Let $\boldsymbol{r} = \boldsymbol{r}(u, v)$ denote the underlying parametric surface. With reference [16], the geodesic curve is approximated as a Bezier curve or a b-spline curve C:

$$\begin{cases} u(s) = \sum_{i=0}^{n} B_i(s)u_i, \\ v(s) = \sum_{i=0}^{n} B_i(s)v_i. \end{cases} \tag{A1}$$

The energy of curve C is expressed as:

$$E\left(u_1, u_2, \ldots .u_{n-1}, v_1, v_2 \ldots .v_{n-1}\right) = \frac{1}{2}\int_a^b \left(r_u\frac{\mathrm{d}u}{\mathrm{d}s} + r_v\frac{\mathrm{d}v}{\mathrm{d}s},\ r_u\frac{\mathrm{d}u}{\mathrm{d}s} + r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right)\mathrm{d}s. \tag{A2}$$

The geodesic curve is the solution of the system of geodesic-like equations (A3), which minimizes the energy of curve C.

$$\begin{cases} Eu_i = \int_a^b \left[\left(r_{uu}\frac{\mathrm{d}u}{\mathrm{d}s} + r_{uv}\frac{\mathrm{d}v}{\mathrm{d}s}\right)B_i + r_u\left(B_i\right)', \right. \\ \left. r_u\frac{\mathrm{d}u}{\mathrm{d}s} + r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s = 0, \\ Ev_i = \int_a^b \left[\left(r_{uv}\frac{\mathrm{d}u}{\mathrm{d}s} + r_{vv}\frac{\mathrm{d}v}{\mathrm{d}s}\right)B_i + r_v\left(B_i\right)', \right. \\ \left. r_u\frac{\mathrm{d}u}{\mathrm{d}s} + r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s = 0. \end{cases} \tag{A3}$$

In Eq. (A3), Eu_i is rewritten as:

$$Eu_i = \int_a^b \left[\left(r_{uu}\frac{\mathrm{d}u}{\mathrm{d}s} + r_{uv}\frac{\mathrm{d}v}{\mathrm{d}s}\right)B_i, r_u\frac{\mathrm{d}u}{\mathrm{d}s} + r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s + \int_a^b \left[r_u\left(B_i\right)', r_u\frac{\mathrm{d}u}{\mathrm{d}s} + r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s. \tag{A4}$$

In Eq. (A4), we have:

$$\begin{aligned}\int_a^b &\left[\left(r_{uu}\frac{\mathrm{d}u}{\mathrm{d}s}+r_{uv}\frac{\mathrm{d}v}{\mathrm{d}s}\right)B_i, r_u\frac{\mathrm{d}u}{\mathrm{d}s}+r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s\\ &= \int_a^b B_i\left[r_{uu}r_u\left(\frac{\mathrm{d}u}{\mathrm{d}s}\right)^2+r_{uv}r_v\left(\frac{\mathrm{d}v}{\mathrm{d}s}\right)^2\right.\\ &\qquad\left.+r_{uu}r_v\frac{\mathrm{d}u}{\mathrm{d}s}\frac{\mathrm{d}v}{\mathrm{d}s}+r_{uv}r_u\frac{\mathrm{d}u}{\mathrm{d}s}\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s. \qquad \text{(A5)}\end{aligned}$$

$$\begin{aligned}\int_a^b &\left[r_u\left(B_i\right)', r_u\frac{\mathrm{d}u}{\mathrm{d}s}+r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s\\ &= \left(r_u, r_u\frac{\mathrm{d}u}{\mathrm{d}s}+r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right)\left(B_i\right)|_a^b\\ &\quad-\int_a^b\left(B_i\right)\mathrm{d}\left(r_u, r_u\frac{\mathrm{d}u}{\mathrm{d}s}+r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right)\\ &= -\int_a^b\left(B_i\right)\mathrm{d}\left(r_u, r_u\frac{\mathrm{d}u}{\mathrm{d}s}+r_v\frac{\mathrm{d}v}{\mathrm{d}s}\right)\\ &= -\int_a^b\left(B_i\right)\left[E\frac{\mathrm{d}^2u}{\mathrm{d}s^2}+F\frac{\mathrm{d}^2v}{\mathrm{d}s^2}+E_u\left(\frac{\mathrm{d}u}{\mathrm{d}s}\right)^2\right.\\ &\qquad\left.+F_v\left(\frac{\mathrm{d}v}{\mathrm{d}s}\right)^2+\left(E_v+F_u\right)\frac{\mathrm{d}u}{\mathrm{d}s}\frac{\mathrm{d}v}{\mathrm{d}s}\right]\mathrm{d}s \qquad \text{(A6)}\end{aligned}$$

Substitute Eq. (A5) and Eq. (A6) into Eq. (A4), Eu_i is expressed as:

$$\begin{aligned}Eu_i = &-\int_a^b\left(B_i\right)\\ &\times\left[\begin{aligned}&E\frac{\mathrm{d}^2u}{\mathrm{d}s^2}+F\frac{\mathrm{d}^2v}{\mathrm{d}s^2}+\left(E_v+F_u-r_{uu}r_v-r_{uv}r_u\right)\frac{\mathrm{d}u}{\mathrm{d}s}\frac{\mathrm{d}v}{\mathrm{d}s}\\ &+\left(E_u-r_{uu}r_u\right)\left(\frac{\mathrm{d}u}{\mathrm{d}s}\right)^2+\left(F_v-r_{uv}r_v\right)\left(\frac{\mathrm{d}v}{\mathrm{d}s}\right)^2\end{aligned}\right]\mathrm{d}s.\end{aligned} \qquad \text{(A7)}$$

Similarly, Ev_i is expressed as:

$$\begin{aligned}Ev_i = &-\int_a^b\left(B_i\right)\\ &\times\left[\begin{aligned}&F\frac{\mathrm{d}^2u}{\mathrm{d}s^2}+G\frac{\mathrm{d}^2v}{\mathrm{d}s^2}+\left(F_v+G_u-r_{uv}r_v-r_{vv}r_u\right)\frac{\mathrm{d}u}{\mathrm{d}s}\frac{\mathrm{d}v}{\mathrm{d}s}\\ &+\left(F_u-r_{uv}r_u\right)\left(\frac{\mathrm{d}u}{\mathrm{d}s}\right)^2+\left(G_v-r_{vv}r_v\right)\left(\frac{\mathrm{d}v}{\mathrm{d}s}\right)^2\end{aligned}\right]\mathrm{d}s.\end{aligned} \qquad \text{(A8)}$$

With Eq. (A7) and Eq. (A8), solving the system of geodesic-like equations (A3) is equivalent to solve the following:

$$\begin{aligned}\frac{-GEu_i+FEv_i}{EG-F^2} &= \int_a^b\left(B_i\right)\left[\frac{\mathrm{d}^2u}{\mathrm{d}s^2}+\Gamma_{11}^1\left(\frac{\mathrm{d}u}{\mathrm{d}s}\right)^2\right.\\ &\qquad\left.+2\Gamma_{12}^1\frac{\mathrm{d}u}{\mathrm{d}s}\frac{\mathrm{d}v}{\mathrm{d}s}+\Gamma_{22}^1\left(\frac{\mathrm{d}v}{\mathrm{d}s}\right)^2\right]\mathrm{d}s=0,\end{aligned}$$

$$\begin{aligned}\frac{FEu_i-EEv_i}{EG-F^2} &= \int_a^b\left(B_i\right)\left[\frac{\mathrm{d}^2v}{\mathrm{d}s^2}+\Gamma_{11}^2\left(\frac{\mathrm{d}u}{\mathrm{d}s}\right)^2\right.\\ &\qquad\left.+2\Gamma_{12}^2\frac{\mathrm{d}u}{\mathrm{d}s}\frac{\mathrm{d}v}{\mathrm{d}s}+\Gamma_{22}^2\left(\frac{\mathrm{d}v}{\mathrm{d}s}\right)^2\right]\mathrm{d}s=0.\end{aligned} \qquad \text{(A9)}$$

With Eq. (A1) and Eq. (A9), we know that Chen's optimization strategy for finding geodesic curves is equivalent to solving the differential equation of geodesics based on the Galerkin method.

DECLARATION OF CONFLICTING INTERESTS

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

[1] T. W. Sederberg, G. T. Finnigan, X. Li, H. Lin, and H. Ipson, "Watertight trimmed NURBS," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–8, Aug. 2008.

[2] J.-H. Chuang, C.-H. Lin, and W.-C. Hwang, "Variable-radius blending of parametric surfaces," *Vis. Comput.*, vol. 11, no. 10, pp. 513–525, Oct. 1995.

[3] H.-Y. Tam, H.-W. Law, and H. Xu, "A geometric approach to the offsetting of profiles on three-dimensional surfaces," *Comput.-Aided Des.*, vol. 36, no. 10, pp. 887–902, Sep. 2004.

[4] H.-Y. Xu, H. Y. Tam, X. Fang, and L. Hu, "Quart-parametric interpolations for intersecting paths," *Comput.-Aided Des.*, vol. 41, no. 6, pp. 432–440, Jun. 2009.

[5] J. Pegna and F.-E. Wolter, "Surface curve design by orthogonal projection of space curves onto free-form surfaces," *J. Mech. Des.*, vol. 118, no. 1, pp. 45–52, Mar. 1996.

[6] W. Xiaoping, A. Luling, Z. Laishui, and Z. Liyan, "Constructing G2 continuous curve on freeform surface with normal projection," *Chin. J. Aeronaut.*, vol. 23, no. 1, pp. 137–144, Feb. 2010.

[7] H.-Y. Xu, X. Fang, H.-Y. Tam, X. Wu, and L. Hu, "A second-order algorithm for curve orthogonal projection onto parametric surface," *Int. J. Comput. Math.*, vol. 89, no. 1, pp. 98–111, Jan. 2012.

[8] J. M. Beck, R. T. Farouki, and J. K. Hinds, "Surface analysis methods," *IEEE Comput. Graph. Appl.*, vol. 6, no. 12, pp. 18–36, Dec. 1986.

[9] N. M. Patrikalakis and L. Bardis, "Offsets of curves on rational B-spline surfaces," *Eng. Comput.*, vol. 5, no. 1, pp. 39–46, Dec. 1989.

[10] G. V. V. R. Kumar, P. Srinivasan, V. D. Holla, K. G. Shastry, and B. G. Prakash, "Geodesic curve computations on surfaces," *Comput. Aided Geometric Des.*, vol. 20, no. 2, pp. 119–133, May 2003.

[11] K. Polthier and M. Schmies, *Mathematical Visualization*. Berlin, Germany: Springer, 1998.

[12] I. Hotz and H. Hagen, "Visualizing geodesics," in *Proc. 11th IEEE Vis. Conf.*, Washington, DC, USA, Oct. 2000, pp. 311–318.

[13] P. Zhang, R. Sun, and T. Huang, "A geometric method for computation of geodesic on parametric surfaces," *Comput. Aided Geometric Des.*, vol. 38, pp. 24–37, Oct. 2015.

[14] T. Maekawa, "Computation of shortest paths on free-form parametric surfaces," *J. Mech. Des.*, vol. 118, no. 4, pp. 499–508, Dec. 1996.

[15] E. Kasap, M. Yapici, and F. T. Akyildiz, "A numerical study for computation of geodesic curves," *Appl. Math. Comput.*, vol. 171, no. 2, pp. 1206–1213, Dec. 2005.

[16] S.-G. Chen, "Geodesic-like curves on parametric surfaces," *Comput. Aided Geometric Des.*, vol. 27, no. 1, pp. 106–117, Jan. 2010.

[17] D. Martínez, L. Velho, and P. C. Carvalho, "Computing geodesics on triangular meshes," *Comput. Graph.*, vol. 29, no. 5, pp. 667–675, Oct. 2005.

[18] B. Liu, S. Chen, S.-Q. Xin, Y. He, Z. Liu, and J. Zhao, "An optimization-driven approach for computing geodesic paths on triangle meshes," *Comput.-Aided Des.*, vol. 90, pp. 105–112, Sep. 2017.

[19] K. Crane, M. Livesu, E. Puppo, and Y. Qin. (2020). *A Survey of Algorithms for Geodesic Paths and Distances*. [Online]. Available: https://www.researchgate.net/publication/343124798_A_Survey_of_Algorithms_for_Geodesic_Paths_and_Distances

[20] P. Zhang, Z. Zhou, G. Chen, and S. Chen, "Optimizing the lay-up of composite tapes based on improved geodesic strategy for automated tape placement," *Proc. Inst. Mech. Eng. C, J. Mech. Eng. Sci.*, vol. 232, no. 22, pp. 4084–4097, Nov. 2018.

[21] L. Zu, "Stability of fiber trajectories for winding toroidal pressure vessels," *Compos. Struct.*, vol. 94, no. 5, pp. 1855–1860, Apr. 2012.

[22] P. N. Azariadis and N. A. Aspragathos, "Geodesic curvature preservation in surface flattening through constrained global optimization," *Comput. Aided Des.*, vol. 33, no. 8, pp. 581–591, 2001.

[23] J. Goh, A. A. Majid, and A. I. M. Ismail, "Numerical method using cubic B-spline for the heat and wave equation," *Comput. Math. Appl.*, vol. 62, no. 12, pp. 4492–4498, Dec. 2011.

[24] E. N. Aksan, "Quadratic B-spline finite element method for numerical solution of the Burgers' equation," *Appl. Math. Comput.*, vol. 174, no. 2, pp. 884–896, Mar. 2006.

[25] A. M. Elsherbeny, R. M. I. El-hassani, H. El-badry, and M. I. Abdallah, "Solving 2D-Poisson equation using modified cubic B-spline differential quadrature method," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 2879–2885, Dec. 2018.

[26] B. Shirinzadeh, G. Cassidy, D. Oetomo, G. Alici, and M. H. Ang, Jr., "Trajectory generation for open-contoured structures in robotic fibre placement," *Robot. Comput.-Integr. Manuf.*, vol. 23, no. 4, pp. 380–394, Aug. 2007.

[27] W. B. Goldsworthy, "Geodesic path length compensator for composite tape placement method," U.S. Patent 3 810 805, May 14, 1974.

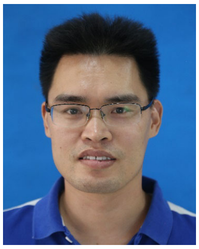
[28] M. W. Hogg, "The Boeing company. Method for laying composite tape," U.S. Patent 7 842 145, Apr. 6, 2006.

[29] N. M. Patrikalakis and T. Maekawa. *Shape Interrogation for Computer Aided Design and Manufacturing*. Berlin, Germany: Springer, 2002.

[30] C. R. de Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978.

[31] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. New York, NY, USA: Springer, 1997.

PENG ZHANG received the B.E. and Ph.D. degrees in mechanical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2009 and 2016, respectively. He is currently a Lecturer with Changsha University of Science and Technology. His research interests include trajectory planning for automated tape placement and automated fiber placement.



LAIRONG YIN received the B.E. degree in mechanical engineering from Xiangtan University, Xiangtan, China, in 2005, and the Ph.D. degree in mechanical engineering from the University of Science and Technology Beijing, in 2011. He is currently an Associate Professor with Changsha University of Science and Technology. His research interests include theory of mechanism and advanced robotics.

ZHENHUA ZHOU received the B.E. degree in mechanical engineering from Hunan University of Science and Technology, Xiangtan, China, in 2006, and the Ph.D. degree in mechanical engineering from Huazhong University of Science and Technology, Wuhan, in 2013. He is currently a Lecturer with Changsha University of Science and Technology. His research interest includes theory of advanced robotics.

LONG HUANG received the B.E. and Ph.D. degrees in mechanical engineering from Beihang University, Beijing, China, in 2010 and 2017, respectively. He is currently a Lecturer with Changsha University of Science and Technology. His research interests include mechanism design and robot design.