Journal of Mechanical Design, ASME Transactions, Vol. 118 No. 4, pages 499-508, 1996.

Computation of Shortest Paths on Free-Form Parametric Surfaces

Takashi Maekawa

Massachusetts Institute of Technology
Department of Ocean Engineering
Design Laboratory, Cambridge, MA 02139-4307, USA

Abstract

Computation of shortest paths on free-form surfaces is an important problem in ship design, robot motion planning, computation of medial axis transforms of trimmed surface patches, terrain navigation and NC machining. The objective of this paper is to provide an efficient and reliable method for computing the shortest path between two points on a free-form parametric surface and the shortest path between a point and a curve on a free-form parametric surface. These problems can be reduced to solving a two point boundary value problem. Our approach for solving the two point boundary value problem is based on a relaxation method relying on finite difference discretization. Examples illustrate our method.

Keywords: geodesics, boundary value problem, finite difference method, relaxation method

1 Introduction

The history of geodesic lines begins with the study by Johann Bernoulli, who solved the problem of the shortest distance between two points on a convex surface in 1697, according to [15]. He showed that the osculating plane of the geodesic line must always be perpendicular to the tangent plane. The equation of the geodesics was first obtained by Euler (1732) for a surface given by $F(x, y, z) = 0$. His attention to the problem was due to Johann Bernoulli, probably through the aid of his nephew Daniel, who was at St. Petersburg with Euler [14]. Bliss [2] obtained the geodesic lines on the anchor ring, which has a torus shape, analytically. Munchmeyer and Haw [10] were the first to introduce the geodesic curve to the CAGD community. They applied the geodesic curve in ship design, namely to find out the precise layout of the seams and butts in the ship hull. Beck et al [1] computed both the initial-value integration and boundary-value integration of geodesic paths, using the fourth order Runge-Kutta method on a bicubic spline surface. However their paper does not provide any details of the shooting method for the solution of the boundary value problem. Patrikalakis and Bardis computed geodesic offsets of curves on rational B-spline surfaces using the initial-value integration [11]. One application of such offsets is automated construction of linkage curves for free-form procedural blending surfaces. Sneyd and Peskin [13] investigated the computation of geodesic paths on a generalized cylinder as initial value problem using a second order Runge-Kutta method. Their work was motivated by constructing the great vessels of the heart out of geodesic fibers. Very recently, Kimmel et al. [7] presented a new numerical method for finding the shortest path on surfaces by calculating the propagation of an equal geodesic-distance contour from a point or a source region on the surface. The algorithm works on a rectangular grid using the finite difference approximation. The shortest path problem is also very active among the robot motion planning and terrain navigation communities,

however they usually represent the surface as a polyhedral surface and solve the problem in the field of linear computational geometry [9].

In this work a new approach for finding the shortest path between two points on a free-form parametric surface is introduced. Also a method for computing the shortest path between a point and a curve on a free-form surface is described. The geodesic path between two points is obtained by first discretizing the governing equations by finite difference on a mesh with m points. Secondly we start with an initial guess and improve the solution iteratively or in other words relax to the true solution. The shortest path between a point and a curve is obtained in a similar way but with nested iterations, since we do not know the point on the curve which forms the shortest path beforehand.

This paper is structured as follows. Section 2 begins with a brief review of relevant differential geometry properties of geodesics on a parametric surface. Section 3 describes the finite difference approach to the two point boundary value problem. Section 4 explains how to obtain a good initial approximation for the two point boundary value problem and provides a robust algorithm to find all the geodesic paths between two points on a surface. Section 5 discusses an algorithm to obtain a geodesic path from a point to a curve on a surface. Finally, section 6 illustrates our method with examples and concludes the paper.

2 Review of Differential Geometry

The books by Struik [15], Kreyszig [8], doCarmo [4] offer firm theoretical basis to the differential geometry aspects of geodesics. In this section, we summarize the relevant definitions employed in this work.

A general parametric surface S can be defined as a vector-valued mapping from two-dimensional uv parameter domain to a set of three-dimensional xyz coordinates

$$\mathbf{r}(u,v) = [x(u,v), \quad y(u,v), \quad z(u,v)]^T. \tag{1}$$

Let C be an arc length parametrized regular curve on surface $\mathbf{r}(u,v)$ which passes through point P as shown in Figure 1 and denote by

$$\mathbf{r}(s) = \mathbf{r}(u(s), v(s)). \tag{2}$$

Let $\mathbf{t}$ be a unit tangent vector of C at P, $\mathbf{n}$ be a unit normal vector of C at P, $\mathbf{N}$ be a unit surface normal vector

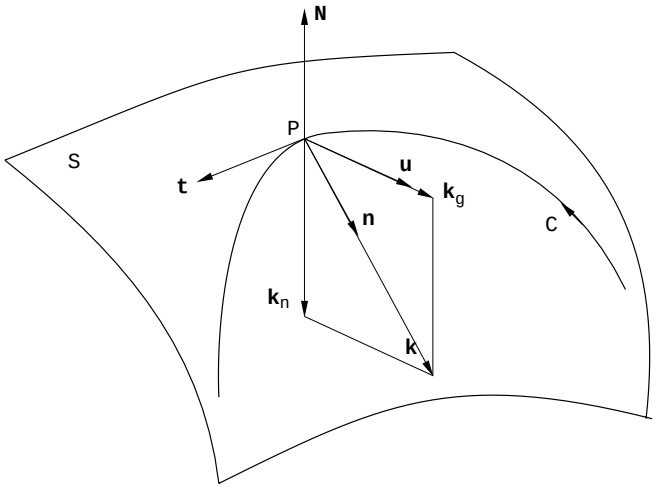

Figure 1: Definition of geodesic curvature

of S at P and $\mathbf{u}$ be a unit vector perpendicular to $\mathbf{t}$ in the tangent plane defined by $\mathbf{N} \times \mathbf{t}$. We can decompose the curvature vector $\mathbf{k}$ of C into $\mathbf{N}$ component $\mathbf{k}_n$, which is called normal curvature vector, and $\mathbf{u}$ component $\mathbf{k}_g$, which is called geodesic curvature vector as follows:

$$\mathbf{k} = \mathbf{k}_n + \mathbf{k}_g = \kappa_n \mathbf{N} + \kappa_g \mathbf{u} \tag{3}$$

Here κ_n and κ_g are the normal and geodesic curvatures, respectively and defined as follows:

$$\kappa_n = -\mathbf{k}\cdot\mathbf{N} \tag{4}$$
$$\kappa_g = \mathbf{k}\cdot\mathbf{u} \tag{5}$$

The minus sign in equation (4) ensures that if κ_n is positive, the center of curvature lies opposite to the direction of the surface normal. Consequently,

$$\kappa_g = \frac{d\mathbf{t}}{ds}\cdot(\mathbf{N}\times\mathbf{t}) \tag{6}$$

The unit tangent vector of the curve C can be obtained by differentiating equation (2) with respect to the arc length.

$$\mathbf{t} = \frac{d\mathbf{r}(u(s),v(s))}{ds} = \mathbf{r}_u\frac{du}{ds} + \mathbf{r}_v\frac{dv}{ds} \tag{7}$$

where the chain rule is used and the subscripts u and v denote the first partial derivatives with respect to u and v parameter respectively. Hence

$$\frac{d\mathbf{t}}{ds} = \mathbf{r}_{uu}\left(\frac{du}{ds}\right)^2 + 2\mathbf{r}_{uv}\frac{du}{ds}\frac{dv}{ds} + \mathbf{r}_{vv}\left(\frac{dv}{ds}\right)^2 + \mathbf{r}_u\frac{d^2u}{ds^2} + \mathbf{r}_v\frac{d^2v}{ds^2} \tag{8}$$

so that

$$\begin{aligned}\kappa_g &= [(\mathbf{r}_u\times\mathbf{r}_{uu})\left(\frac{du}{ds}\right)^3 + (2\mathbf{r}_u\times\mathbf{r}_{uv} + \mathbf{r}_v\times\mathbf{r}_{uu})\left(\frac{du}{ds}\right)^2\frac{dv}{ds} + (\mathbf{r}_u\times\mathbf{r}_{vv} + 2\mathbf{r}_v\times\mathbf{r}_{uv})\frac{du}{ds}\left(\frac{dv}{ds}\right)^2 \\ &+ (\mathbf{r}_v\times\mathbf{r}_{vv})\left(\frac{dv}{ds}\right)^3]\cdot\mathbf{N} + (\mathbf{r}_u\times\mathbf{r}_v)\cdot\mathbf{N}\left(\frac{du}{ds}\frac{d^2v}{ds^2} - \frac{d^2u}{ds^2}\frac{dv}{ds}\right)\end{aligned} \tag{9}$$

We can easily notice that the coefficients of $\left(\frac{du}{ds}\right)^3$, $\left(\frac{du}{ds}\right)^2\frac{dv}{ds}$, $\frac{du}{ds}\left(\frac{dv}{ds}\right)^2$, $\left(\frac{dv}{ds}\right)^3$, $\left(\frac{du}{ds}\frac{d^2v}{ds^2} - \frac{d^2u}{ds^2}\frac{dv}{ds}\right)$ are all functions of the coefficients of the first fundamental form E, F and G and their derivatives, E_u, F_u, G_u, E_v, F_v, G_v. It is interesting to note that the normal curvature κ_n depends on both the first and second fundamental forms, while the geodesic curvature depends only on the first fundamental form. Using the Christoffel symbols Γ^i_{jk} $(i,j,k=1,2)$ defined as follows

$$\begin{aligned}\Gamma^1_{11} &= \frac{GE_u - 2FF_u + FE_v}{2(EG-F^2)}, & \Gamma^2_{11} &= \frac{2EF_u - EE_v + FE_u}{2(EG-F^2)} \\ \Gamma^1_{12} &= \frac{GE_v - FG_u}{2(EG-F^2)}, & \Gamma^2_{12} &= \frac{EG_u - FE_v}{2(EG-F^2)} \\ \Gamma^1_{22} &= \frac{2GF_v - GG_u + FG_v}{2(EG-F^2)}, & \Gamma^2_{22} &= \frac{EG_v - 2FF_v + FG_u}{2(EG-F^2)}\end{aligned} \tag{10}$$

geodesic curvature can be reduced to

$$\begin{aligned}\kappa_g &= [\Gamma^2_{11}\left(\frac{du}{ds}\right)^3 + (2\Gamma^2_{12} - \Gamma^1_{11})\left(\frac{du}{ds}\right)^2\frac{dv}{ds} + (\Gamma^2_{22} - 2\Gamma^1_{12})\frac{du}{ds}\left(\frac{dv}{ds}\right)^2 \\ &- \Gamma^1_{22}\left(\frac{dv}{ds}\right)^3 + \frac{du}{ds}\frac{d^2v}{ds^2} - \frac{d^2u}{ds^2}\frac{dv}{ds}]\sqrt{EG-F^2}\end{aligned} \tag{11}$$

Geodesic paths are sometimes defined as shortest path between points on a surface, however this is not always a satisfactory definition. In this paper we define as follows [15]:

Definition 2.1 *Geodesics are curves of zero geodesic curvature.*

In other words, the osculating planes of a geodesic contain the surface normal. From this definition we can easily see that the geodesic between two points on a sphere is a great circle. But there are two arcs of a great circle between two of their points, and only one of them provides the shortest distance, except that the two points are

the end points of a diameter. This example indicates that there may exist more than one geodesic between two points.

According to the definition, we can determine the differential equation that any geodesic on a surface must satisfy by simply setting $\kappa_g = 0$ in equation (11) and obtain

$$\frac{du}{ds}\frac{d^2v}{ds^2} - \frac{d^2u}{ds^2}\frac{dv}{ds} = -\Gamma_{11}^2\left(\frac{du}{ds}\right)^3 - (2\Gamma_{12}^2 - \Gamma_{11}^1)\left(\frac{du}{ds}\right)^2\frac{dv}{ds} + (2\Gamma_{12}^1 - \Gamma_{22}^2)\frac{du}{ds}\left(\frac{dv}{ds}\right)^2 + \Gamma_{22}^1\left(\frac{dv}{ds}\right)^3. \tag{12}$$

Alternatively we can derive the differential equation for geodesics by considering that the surface normal $\mathbf{N}$ has the direction of normal of the geodesic line $\pm\mathbf{n}$

$$\mathbf{n}\cdot\mathbf{r}_u = 0, \qquad \mathbf{n}\cdot\mathbf{r}_v = 0. \tag{13}$$

Since $k\mathbf{n} = \frac{d\mathbf{t}}{ds}$, equation (13) can be rewritten as

$$\frac{d\mathbf{t}}{ds}\cdot\mathbf{r}_u = 0, \qquad \frac{d\mathbf{t}}{ds}\cdot\mathbf{r}_v = 0 \tag{14}$$

By substituting equation (8) into equations (14) we obtain

$$(\mathbf{r}_{uu}\cdot\mathbf{r}_u)\left(\frac{du}{ds}\right)^2 + 2(\mathbf{r}_{uv}\cdot\mathbf{r}_u)\frac{du}{ds}\frac{dv}{ds} + (\mathbf{r}_{vv}\cdot\mathbf{r}_u)\left(\frac{dv}{ds}\right)^2 + E\frac{d^2u}{ds^2} + F\frac{d^2v}{ds^2} = 0 \tag{15}$$

$$(\mathbf{r}_{uu}\cdot\mathbf{r}_v)\left(\frac{du}{ds}\right)^2 + 2(\mathbf{r}_{uv}\cdot\mathbf{r}_v)\frac{du}{ds}\frac{dv}{ds} + (\mathbf{r}_{vv}\cdot\mathbf{r}_v)\left(\frac{dv}{ds}\right)^2 + F\frac{d^2u}{ds^2} + G\frac{d^2v}{ds^2} = 0 \tag{16}$$

By eliminating $\frac{d^2v}{ds^2}$ from equation (15) using equation (16), and eliminating $\frac{d^2u}{ds^2}$ from equation (16) using equation (15) and employing the Christoffel symbols, we obtain

$$\frac{d^2u}{ds^2} + \Gamma_{11}^1\left(\frac{du}{ds}\right)^2 + 2\Gamma_{12}^1\frac{du}{ds}\frac{dv}{ds} + \Gamma_{22}^1\left(\frac{dv}{ds}\right)^2 = 0 \tag{17}$$

$$\frac{d^2v}{ds^2} + \Gamma_{11}^2\left(\frac{du}{ds}\right)^2 + 2\Gamma_{12}^2\frac{du}{ds}\frac{dv}{ds} + \Gamma_{22}^2\left(\frac{dv}{ds}\right)^2 = 0 \tag{18}$$

Equations (17) and (18) are related by the first fundamental form $ds^2 = Edu^2 + 2Fdudv + Gdv^2$ and if we eliminate ds from both equations, the equations reduce to equation (12) with u taken as parameter. These two second order differential equations can be rewritten as a system of four first order differential equations [3].

$$\frac{du}{ds} = p \tag{19}$$

$$\frac{dv}{ds} = q \tag{20}$$

$$\frac{dp}{ds} = -\Gamma_{11}^1p^2 - 2\Gamma_{12}^1pq - \Gamma_{22}^1q^2 \tag{21}$$

$$\frac{dq}{ds} = -\Gamma_{11}^2p^2 - 2\Gamma_{12}^2pq - \Gamma_{22}^2q^2 \tag{22}$$

3 Two Point Boundary Value Problem

We can solve a system of four first order ordinary differential equations (19) to (22) as an initial-value problem (IVP), where all four boundary conditions are given at one point, or as a boundary-value problem (BVP), where four boundary conditions are specified at two distinct points. Most of the problems that arise in applications of geodesics are not IVP but BVP, which are much more difficult to solve. It is well known that the solution of an IVP is unique, however for a BVP it is possible that the differential equations have many solutions or even no solution [6]. General methods for the solutions of two-point BVPs can be found in [6, 5].

It is convenient to write a system of differential equations in vector form, since we can describe the equations for systems as just for a single vector equation. If we set

$$\begin{aligned} \mathbf{y} &= (y_1, y_2, \ldots, y_n)^T \quad , \mathbf{g} = (g_1, g_2, \ldots, g_n)^T && (23) \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_n)^T \quad , \boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)^T && (24) \\ s &\in [A, B] && (25) \end{aligned}$$

Then the general first order differential equation for a boundary value problem can be written as:

$$\frac{d\mathbf{y}}{ds} = \mathbf{g}(s, \mathbf{y}), \quad \mathbf{y}(A) = \boldsymbol{\alpha}, \quad \mathbf{y}(B) = \boldsymbol{\beta} \tag{26}$$

$$\tag{27}$$

There are two commonly used approaches to the numerical solution of BVP. The idea of the first technique is that if all values of $\mathbf{y}(s)$ are known at $s = A$, then the problem can be reduced to an IVP. However, $\mathbf{y}(A)$ can be found only by solving the problem. Therefore an iterative procedure must be used. We assume values at $s = A$, which are not given as boundary conditions at $s = A$ and compute the solution of the resulting IVP to $s = B$. The computed values of $\mathbf{y}(B)$ will not, in general, agree with the corresponding boundary condition at $s = B$. Consequently, we need to adjust the initial values and try again. The process is repeated until the computed values at the final point agree with the boundary conditions and referred as shooting method. The second method is based on a finite difference approximation to $\frac{d\mathbf{y}}{ds}$ on a mesh of points in the interval $[A, B]$. This method starts with an initial guess and improves the solution iteratively and referred as, direct method, relaxation method or finite difference method. We have implemented both methods and found that the finite difference method is much more reliable than the shooting method. The shooting method is often very sensitive to the unknown initial angles at point A and unless a good initial guess is provided, the integrated path will never reach the other point B, while the finite difference method starts with two end points fixed and relaxes to the true solution and hence it is much more stable. Therefore we will focus only on the finite difference method.

Let us consider a mesh of points satisfying $A = s_1 < s_2 < \ldots < s_m = B$. We approximate the n first order differential equations by the trapezoidal rule [5].

$$\frac{\mathbf{Y}_k - \mathbf{Y}_{k-1}}{s_k - s_{k-1}} = \frac{1}{2}[\mathbf{G}_k + \mathbf{G}_{k-1}], \qquad k = 2, 3, \ldots, m \tag{28}$$

with boundary conditions

$$\mathbf{Y}_1 = \boldsymbol{\alpha}, \quad \mathbf{Y}_m = \boldsymbol{\beta} \tag{29}$$

Here the n-vectors $\mathbf{Y}_k$, $\mathbf{G}_k$ are meant to approximate $\mathbf{y}(s_k)$ and $\mathbf{g}(s_k)$. $\mathbf{Y}_1$ has n_1 known components, while $\mathbf{Y}_m$ has $n_2 = n - n_1$ known components. This discrete approximation will be accurate to the order of h^2 ($h = \max_k\{s_k - s_{k-1}\}$). Equation (28) forms a system of $(m-1)n$ nonlinear algebraic equations with mn unknowns $\mathbf{Y}_k = (Y_1, Y_2, \ldots, Y_n)_k^T$ $(k = 1, \ldots, m)$. The remaining n equations come from boundary conditions (29). Let us refer to equation (28) as

$$\mathbf{F}_k = (F_{1,k}, F_{2,k}, \ldots, F_{n,k})^T = \frac{\mathbf{Y}_k - \mathbf{Y}_{k-1}}{s_k - s_{k-1}} - \frac{1}{2}[\mathbf{G}_k + \mathbf{G}_{k-1}] = 0, \qquad k = 2, 3, \ldots, m \tag{30}$$

and equations (29) as

$$\mathbf{F}_1 = (F_{1,1}, F_{2,1}, \ldots, F_{n_1,1})^T = \mathbf{Y}_1 - \boldsymbol{\alpha} = 0, \quad \mathbf{F}_{m+1} = (F_{1,m+1}, F_{2,m+1}, \ldots, F_{n_2,m+1})^T = \mathbf{Y}_m - \boldsymbol{\beta} = 0 \tag{31}$$

then we have mn nonlinear algebraic equations

$$\mathbf{F} = (\mathbf{F}_1^T, \mathbf{F}_2^T, \ldots, \mathbf{F}_{m+1}^T)^T = \mathbf{0} \tag{32}$$

and can be computed by quadratically convergent Newton iteration, if a sufficiently accurate starting vector $\mathbf{Y}^{(0)} = (\mathbf{Y}_1^T, \mathbf{Y}_2^T, \ldots, \mathbf{Y}_m^T)^T$ is provided. The Newton iteration scheme is given by

$$\mathbf{Y}^{(i+1)} = \mathbf{Y}^{(i)} + \Delta\mathbf{Y}^{(i)} \tag{33}$$

$$[\mathbf{J}^{(i)}]\Delta\mathbf{Y}^{(i)} = -\mathbf{F}^{(i)} \tag{34}$$

where $[\mathbf{J}^{(i)}]$ is the mn by mn Jacobian matrix of $\mathbf{F}^{(i)}$ with respect to $\mathbf{Y}^{(i)}$.

If we apply this finite difference method to the geodesics path problem, we have

$$\mathbf{y} = (y_1, y_2, y_3, y_4)^T = (u, v, p, q)^T \tag{35}$$

$$\mathbf{g} = (g_1, g_2, g_3, g_4)^T = (p,\ q,\ -\Gamma^1_{11}p^2 - 2\Gamma^1_{12}pq - \Gamma^1_{22}q^2,\ -\Gamma^2_{11}p^2 - 2\Gamma^2_{12}pq - \Gamma^2_{22}q^2)^T \tag{36}$$

hence

$$F_{1,k} = u_k - u_{k-1} - \frac{1}{2}(s_k - s_{k-1})(p_k + p_{k-1}) = 0 \tag{37}$$

$$F_{2,k} = v_k - v_{k-1} - \frac{1}{2}(s_k - s_{k-1})(q_k + q_{k-1}) = 0 \tag{38}$$

$$\begin{aligned} F_{3,k} &= p_k - p_{k-1} + \frac{1}{2}(s_k - s_{k-1})(\Gamma^1_{11,k}p_k^2 + 2\Gamma^1_{12,k}p_kq_k + \Gamma^1_{22,k}q_k^2 \\ &+ \Gamma^1_{11,k-1}p_{k-1}^2 + 2\Gamma^1_{12,k-1}p_{k-1}q_{k-1} + \Gamma^1_{22,k-1}q_{k-1}^2) = 0 \end{aligned} \tag{39}$$

$$\begin{aligned} F_{4,k} &= q_k - q_{k-1} + \frac{1}{2}(s_k - s_{k-1})(\Gamma^2_{11,k}p_k^2 + 2\Gamma^2_{12,k}p_kq_k + \Gamma^2_{22,k}q_k^2 \\ &+ \Gamma^2_{11,k-1}p_{k-1}^2 + 2\Gamma^2_{12,k-1}p_{k-1}q_{k-1} + \Gamma^2_{22,k-1}q_{k-1}^2) = 0 \end{aligned} \tag{40}$$

for $k = 2, 3, \ldots, m$. Note that the mesh points are defined in $3D$ space and not on the parameter domain. In this paper we assume that the coordinates of the two end points A and B in the parameter domain are given as follows:

$$F_{1,1} = u_1 - u_A = 0 \tag{41}$$

$$F_{2,1} = v_1 - v_A = 0 \tag{42}$$

$$F_{1,m+1} = u_m - u_B = 0 \tag{43}$$

$$F_{2,m+1} = v_m - v_B = 0 \tag{44}$$

Now we can evaluate the Jacobian Matrix $[\mathbf{J}]$. Since $\mathbf{F}_k$ are evaluated using information from points k and $k-1$, we only need to evaluate

$$\frac{\partial F_{i,k}}{\partial Y_{j,k-1}}, \quad \frac{\partial F_{i,k}}{\partial Y_{j,k}}, \qquad i, j = 1, 2, 3, 4 \qquad k = 2, 3, \ldots, m \tag{45}$$

for non-boundary equations. For the two boundary points A and B, we evaluate

$$\frac{\partial F_{i,1}}{\partial Y_{j,1}}, \quad \frac{\partial F_{i,m+1}}{\partial Y_{j,m}}, \qquad i = 1, 2 \qquad j = 1, 2, 3, 4 \tag{46}$$

Therefore the Jacobian matrix has a banded structure or more specifically it has the following block structure.

$$\begin{pmatrix} B_1 & & & & & & & \\ A_2 & B_2 & & & & & & \\ & A_3 & B_3 & & & & & \\ & & A_4 & \ddots & & & & \\ & & & & \ddots & & & \\ & & & & & \ddots & & \\ & & & & & & B_{m-1} & \\ & & & & & & A_m & B_m \\ & & & & & & & A_{m+1} \end{pmatrix} \tag{47}$$

where A_k, B_k $(k = 2, \ldots, m)$ are 4×4 submatrices obtained by evaluating equations (45) and B_1, A_{m+1} are 2×4 submatrices obtained by evaluating equations (46). The columns correspond to mesh points, while the rows correspond to equations. If we set

$$h_k = s_k - s_{k-1} \tag{48}$$

$$\omega^j_{k-1} = \Gamma^j_{11,k-1}p_{k-1} + \Gamma^j_{12,k-1}q_{k-1}, \qquad j = 1, 2 \tag{49}$$

$$\omega_k^j = \Gamma_{11,k}^j p_k + \Gamma_{12,k}^j q_k, \qquad j = 1,2 \tag{50}$$

$$\theta_{k-1}^j = \Gamma_{12,k-1}^j p_{k-1} + \Gamma_{22,k-1}^j q_{k-1}, \qquad j = 1,2 \tag{51}$$

$$\theta_k^j = \Gamma_{12,k}^j p_k + \Gamma_{22,k}^j q_k, \qquad j = 1,2 \tag{52}$$

$$\tag{53}$$

then A_k, B_k, B_1 and A_{m+1} can be written as follows:

$$A_k = \begin{pmatrix} -1 & 0 & [-\frac{h_k}{2}] & 0 \\ 0 & -1 & 0 & [-\frac{h_k}{2}] \\ 0 & 0 & h_k\omega_{k-1}^1 - 1 & h_k\theta_{k-1}^1 \\ 0 & 0 & h_k\omega_{k-1}^2 & h_k\theta_{k-1}^2 - 1 \end{pmatrix}, \qquad k = 2,\ldots,m \tag{54}$$

$$B_k = \begin{pmatrix} 1 & 0 & -\frac{h_k}{2} & 0 \\ 0 & 1 & 0 & -\frac{h_k}{2} \\ [0] & 0 & h_k\omega_k^1 + 1 & h_k\theta_k^1 \\ 0 & [0] & h_k\omega_k^2 & h_k\theta_k^2 + 1 \end{pmatrix}, \qquad k = 2,\ldots,m \tag{55}$$

$$B_1 = \begin{pmatrix} [1] & 0 & 0 & 0 \\ 0 & [1] & 0 & 0 \end{pmatrix} \tag{56}$$

$$A_{m+1} = \begin{pmatrix} 1 & 0 & [0] & 0 \\ 0 & 1 & 0 & [0] \end{pmatrix} \tag{57}$$

Here the arc length h_k is approximated by the chord length. The elements surrounded by the brackets [ ] in the submatricies correspond to diagonal elements in the global matrix. The linear system (34) can be solved by a special form of Gaussian elimination technique to minimize the total number of operations [12]. Since there are zero pivot elements in the submatrices B_k and A_{m+1}, we need to interchange columns so that there will be only nonzero elements on the diagonal to ensure numerical stability. The columns which include the zero pivot elements are columns $(4k-3)$, $(4k-2)$ $(k = 2,\ldots,m)$, $(4m-1)$ and $4m$. If we interchange the columns $(4k-3)$ by $(4k-1)$ and $(4k-2)$ by $(4k)$ for $(k = 2,\ldots,m)$, we obtain

$$A_k = \begin{pmatrix} -\frac{h_k}{2} & 0 & [-1] & 0 \\ 0 & -\frac{h_k}{2} & 0 & [-1] \\ h_k\omega_{k-1}^1 - 1 & h_k\theta_{k-1}^1 & 0 & 0 \\ h_k\omega_{k-1}^2 & h_k\theta_{k-1}^2 - 1 & 0 & 0 \end{pmatrix}, \qquad k = 3,\ldots,m \tag{58}$$

$$B_k = \begin{pmatrix} -\frac{h_k}{2} & 0 & 1 & 0 \\ 0 & -\frac{h_k}{2} & 0 & 1 \\ [h_k\omega_k^1 + 1] & h_k\theta_k^1 & 0 & 0 \\ h_k\omega_k^2 & [h_k\theta_k^2 + 1] & 0 & 0 \end{pmatrix}, \qquad k = 2,\ldots,m \tag{59}$$

$$A_{m+1} = \begin{pmatrix} 0 & 0 & [1] & 0 \\ 0 & 0 & 0 & [1] \end{pmatrix}. \tag{60}$$

Consequently the correction vector $\Delta\mathbf{Y}$ becomes as follows:

$$\Delta\mathbf{Y} = (\Delta u_1, \Delta v_1, \Delta p_1, \Delta q_1, \Delta p_2, \Delta q_2, \Delta u_2, \Delta v_2, \ldots, \Delta p_m, \Delta q_m, \Delta u_m, \Delta v_m)^T \tag{61}$$

Since the corrections are based on a first-order Taylor approximation, a regular Newton method may not be sufficient for a very complex nonlinear surface unless a good initial approximation is provided. If the vector norm of the correction vector is large, then it is an indication that the problem is highly nonlinear and may produce a divergent iteration. We shall say that the iteration is converging if $\| \Delta\mathbf{Y}^{(i+1)} \|_1 < \| \Delta\mathbf{Y}^{(i)} \|_1$, where $\| \Delta\mathbf{Y} \|_1$ is a scaled vector norm and defined as

$$\| \Delta\mathbf{Y} \|_1 = \sum_{i=1}^{m} \left(\frac{|\Delta u_i|}{M_u} + \frac{|\Delta v_i|}{M_v} + \frac{|\Delta p_i|}{M_p} + \frac{|\Delta q_i|}{M_q} \right) \tag{62}$$

where M_u, M_v, M_p and M_q are the scale factors for each variable. In this work we set $M_u = M_v = 1$ and $M_p = M_q = 10$, since we find that the magnitude of Δp_i and Δq_i are roughly ten times larger than that of Δu_i and Δv_i through the numerical experiments. To achieve more stability our method employs a step correction procedure

$$\mathbf{Y}^{(i+1)} = \mathbf{Y}^{(i)} + \mu\Delta\mathbf{Y}^{(i)} \tag{63}$$

where $0 < \mu \leq 1$. If $\mu = 1$ the equation reduces to a regular Newton's method, while if $\mu < 1$ the rate of convergence will be less than quadratic. When the problem is not highly nonlinear, we set

$$\begin{aligned} \mu &= 0.2 \qquad if \quad 0.05 < \frac{\| \Delta \mathbf{Y} \|_1}{4m} \\ \mu &= 0.4 \qquad if \quad 0.005 < \frac{\| \Delta \mathbf{Y} \|_1}{4m} \leq 0.05 \\ \mu &= 0.6 \qquad if \quad \frac{\| \Delta \mathbf{Y} \|_1}{4m} \leq 0.005 \end{aligned} \tag{64}$$

to accelerate the convergence when the solution vector is close to the true solution, otherwise we use

$$\mu = 0.2 \tag{65}$$

through out the iteration. The Newton's method terminates when $\frac{\|\Delta \mathbf{Y}\|_1}{4m}$ is smaller than the pre-specified tolerance ϵ_N. The order of ϵ_N should be proportional to h^2, since we are using the trapezoidal rule (see equation (28)).

4 Initial Approximation

4.1 Straight Line Approximation

Straight line approximation is the simplest and most often provides a good initial approximation, since it is a solution to equations (19) to (22) when we neglect all the nonlinear terms in the right hand sides. We connect the two end points in the parameter space by straight line and define a uniform mesh or grid by a set of $k = 1, 2, \ldots, m$ points as shown in Figure 2 (a). Therefore we have

$$u_k = u_A + \frac{u_B - u_A}{m-1}(k-1) \tag{66}$$

$$v_k = v_A + \frac{v_B - v_A}{m-1}(k-1). \tag{67}$$

If we assume that $u_A \neq u_B$ then

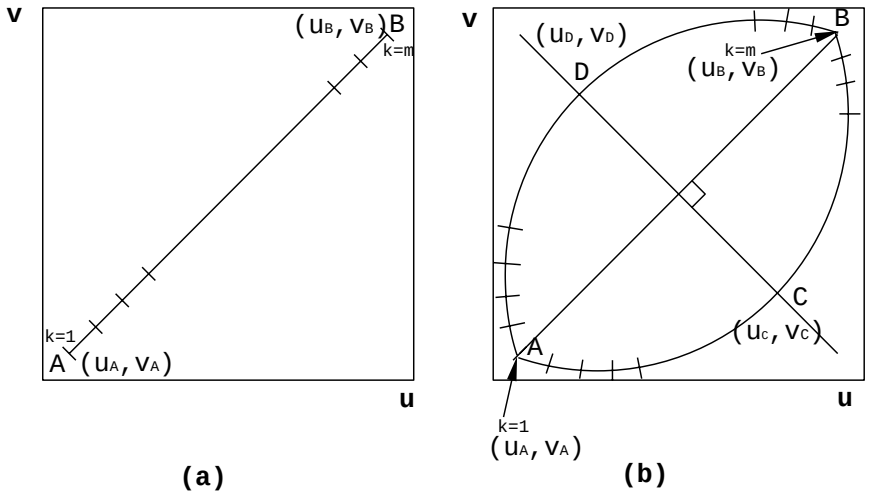

Figure 2: Initial approximations, (a) straight line approximation, (b) circular arc approximation

$$\frac{dv}{du} = \frac{v_B - v_A}{u_B - u_A} \equiv \phi \tag{68}$$

hence

$$\frac{dv}{ds} = \phi \frac{du}{ds} \tag{69}$$

If we substitute this relation into the first fundamental form we obtain

$$E\left(\frac{du}{ds}\right)^2 + 2F\phi\left(\frac{du}{ds}\right)^2 + G\phi^2\left(\frac{du}{ds}\right)^2 = 1 \tag{70}$$

Thus

$$p_k = \frac{du}{ds} = \pm\frac{1}{\sqrt{E_k + 2F_k\phi + G_k\phi^2}} \tag{71}$$

$$q_k = \frac{dv}{ds} = \pm\frac{\phi}{\sqrt{E_k + 2F_k\phi + G_k\phi^2}} \tag{72}$$

When $u_A = u_B$, it is easy to find that $p_k = 0$ and $q_k = \frac{1}{\sqrt{G_k}}$. It is well known that conjugate points do not exist on regions of a surface where Gaussian curvature is negative [15]. Therefore the straight line approximation will provide good initial approximation to the geodesic path in those regions.

4.2 Circular Arc Approximation

The problem of the straight line approximation is that when there are more than one path, it cannot capture the other paths. To make the algorithm more reliable, we have developed the following algorithm. First we pick two points C and D in the parameter domain, which are on the bisector of the two end points A and B, such that $\overline{AC} = \overline{AD}$ or $\overline{BC} = \overline{BD}$ as illustrated in Figure 2. Then we determine two circular arcs which pass through the three points A, C, B and A, D, B. If C and D are taken large enough, all the geodesic paths in the parameter domain between points A and B may lie within or close to the region surrounded by the two circular arcs. Notice that the algorithm fails once the circular arcs go outside the domain. The uv coordinates in the parameter domain i.e. (u_k, v_k), $k = 1, \ldots, m$ can be obtained by equally distributing the points along the circular arc in the parameter domain. Once we have a set of points in the parameter domain, we can easily evaluate p_k, q_k by using the central difference formula for $k = 2, \ldots, m-1$, the forward difference formula for $k = 1$ and the backward difference formula for $k = m-1$. The difference formulae for non-uniform mesh points are listed in the Appendix. Note that the step length h_k is evaluated by computing the chord length between the successive points on the surface. Even if the mesh points are equally distributed along the circular arc in the parameter domain, h_k is not in general constant.

We give the flow chart of **Algorithm A** for computing the geodesic path between given two points in Figure 3.

5 Shortest Path between a Point and a Curve

In this section we solve a problem of finding a shortest path between a point and a curve on a free-form parametric surface, which utilizes the method we have developed in sections 3 and 4. This concept is important in robot motion planning and constructing a medial axis on a free-form surface. Suppose we have a curve C on a parametric surface $\mathbf{r}(u, v)$ defined as $\mathbf{r}^c(t) = \mathbf{r}(u(t), v(t))$, we want to compute the shortest path between point A and curve C as shown in Figure 4. Let us denote the intersection point with curve C and the shortest path by B. Wolter [16] proved that the shortest path from point A to curve C is given by the geodesic path from A to B where the tangent vectors of the geodesic curve and C are orthogonal to each other. If point B were known, the problem could be reduced to IVP, since at point B u, v, p and q are all known. However, point B can be found only by solving the problem. We guess a parameter value $t = t_B$ for point B and solve the BVP. In general the unit tangent vector of the curve $\mathbf{r}^c(t)$ and the unit tangent vector of the geodesic curve at the guessing point will not be orthogonal to each other. Consequently, we need to adjust the parameter value t_B and iterate until those two unit tangent vectors become orthogonal. Therefore for each iteration, we need to solve a two point boundary value problem, which also requires iterations (i.e., nested iteration). If we denote these two unit tangent vectors

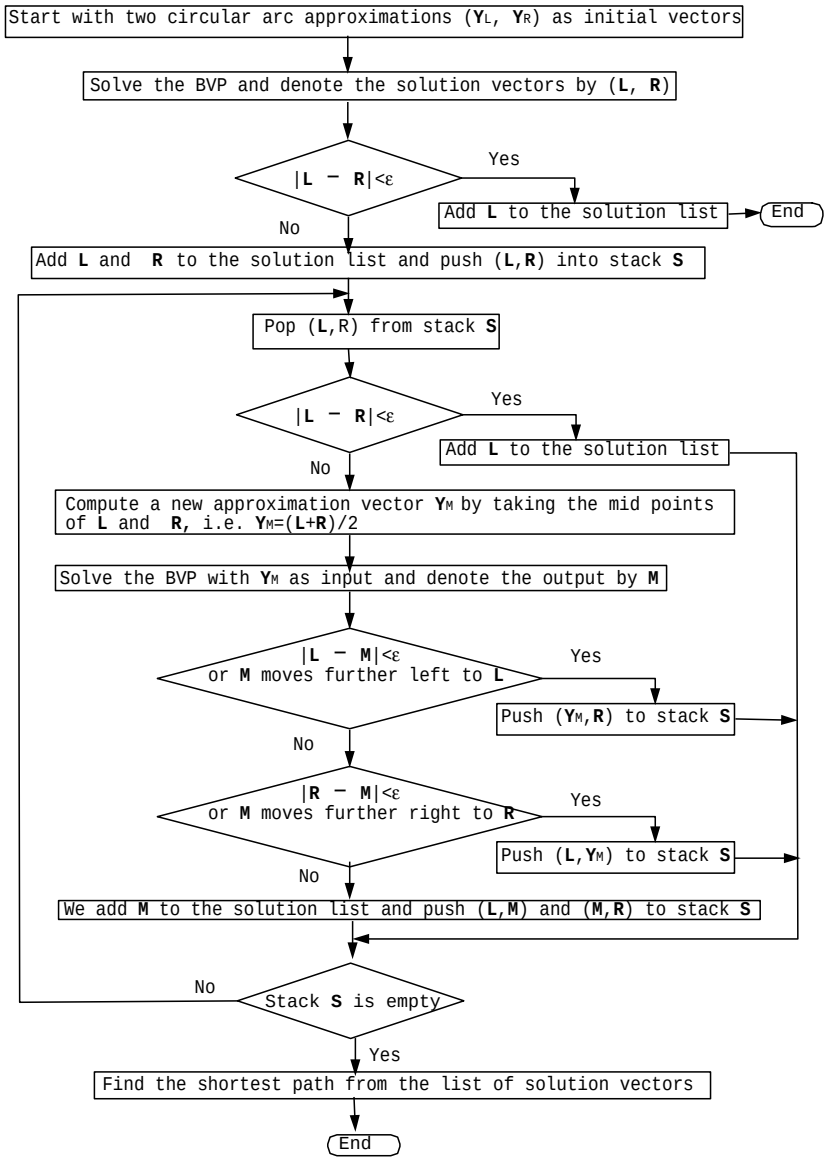

Figure 3: Flow chart of **Algorithm A** for computing the geodesic path between given two points.

as $\mathbf{t}^c$ and $\mathbf{t}^g$, then they can be expressed as follows:

$$\mathbf{t}^c = \frac{\frac{d\mathbf{r}^c(t)}{dt}}{\left|\frac{d\mathbf{r}^c(t)}{dt}\right|} = \frac{\mathbf{r}_u\frac{du}{dt} + \mathbf{r}_v\frac{dv}{dt}}{\sqrt{(\frac{du}{dt}x_u + \frac{dv}{dt}x_v)^2 + (\frac{du}{dt}y_u + \frac{dv}{dt}y_v)^2}} \tag{73}$$

$$\mathbf{t}^g = \frac{d\mathbf{r}^g(s)}{ds} = \mathbf{r}_u\frac{du}{ds} + \mathbf{r}_v\frac{dv}{ds} \tag{74}$$

Therefore the orthogonality condition can be written as

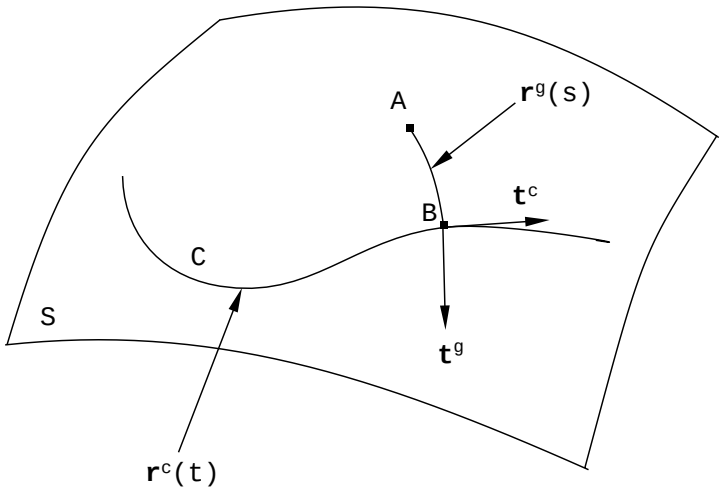

Figure 4: Shortest path between point A and curve C

$$\begin{aligned}\omega(t) &= \mathbf{t}^c \cdot \mathbf{t}^g \\ &= \frac{\frac{du}{ds}\frac{du}{dt}E + (\frac{dv}{ds}\frac{du}{dt} + \frac{dv}{dt}\frac{du}{ds})F + \frac{dv}{ds}\frac{dv}{dt}G}{\sqrt{(\frac{du}{dt}x_u + \frac{dv}{dt}x_v)^2 + (\frac{du}{dt}y_u + \frac{dv}{dt}y_v)^2}} \\ &= 0\end{aligned} \tag{75}$$

Consequently we need to find a parameter value t_B such that $\omega(t_B) = 0$. Since the relationship described above is implicit, we use the secant method instead of Newton's method to obtain t_B. The secant method can be derived from Newton's method by replacing the derivative $\frac{d\omega^{(n)}}{dt}$ by the quotient $(\omega^{(n)} - \omega^{(n-1)})/(t^{(n)} - t^{(n-1)})$. This leads to the following scheme.

$$t^{(n+1)} = t^{(n)} + \Delta t^{(n)}, \qquad \Delta t^{(n)} = -\frac{t^{(n)} - t^{(n-1)}}{\omega^{(n)} - \omega^{(n-1)}}\omega^{(n)}, \qquad \omega^{(n)} \neq \omega^{(n-1)} \tag{76}$$

Notice that the secant method requires two initial approximations. Since the secant method has an order of convergence of $\frac{1}{2}(1+\sqrt{5}) \simeq 1.618$, it converges within a reasonable number of iterations. If the correction $\Delta t^{(n)}$ is large, it is again an indication that problem is highly nonlinear. In such case we also employ a step correction procedure

$$t^{(n+1)} = t^{(n)} + \nu\Delta t^{(n)} \tag{77}$$

where ν is a correction factor. Since we do not know how many solutions are there and the corresponding parameter values of the curve beforehand, we can use a similar algorithm to Algorithm A. Notice that unlike the geodesic path between two points, there is only a unique solution between A and B since at the solution it is a IVP. If the range of the parameter value of the curve is $0 \leq t \leq 1$, we start from both ends of the curve (i.e., t_1=0, t_2=0.02 and $t_1 = 1$, t_2=0.98). Then we recursively find the solutions.

Points	Tolerance	Correction Factor	Iterations			Geodesic Distance		
m	ϵ_N	μ	L	M	R	L	M	R
101	1.0E-3	eqn. (64)	10	1	10	1.661	1.865	1.661
		eqn. (65)	22	1	22			

Table 1: Numerical conditions and results for the computation of the geodesic path between corner points of the wave-like surface.

6 Numerical Applications and Conclusions

6.1 Geodesic Path between Two Points

The first example is a wave-like bicubic B-spline surface, whose control polyhedron is a lattice of 7×7 vertices with uniform knot vectors in both directions and spans $0 \le x \le 1$, $0 \le y \le 1$.

We have computed the geodesic path between two corner points, (u_A, v_A)= (0,0) and (u_B, v_B)=(1,1). We choose (u_C, v_C) to be (0.7, 0.3) and (u_D, v_D) to be (0.3, 0.7). Algorithm A finds three geodesic paths, as shown in Figure 5 (solid thick lines). The computational conditions such as number of mesh points, tolerance and correction factor for the Newton's method ϵ_N, μ, as well as computational results such as number of iterations for convergence and the geodesic distances are listed in Table 1. Characters L, M, N refer to left, middle and right geodesic paths in Figure 5. The middle geodesic path is not a minimal path (s=1.865), while the other two paths are the shortest path (s = 1.661) due to symmetry. Since the surface is relatively simple, the Newton's scheme converges with both correction factors. We can observe from Table 1 that the three step correction factor is two times faster than the constant correction factor. Figure 6 shows how the initial approximation path (the most right thick solid line) converges gradually to the final solution (wavy thick solid line). The intermediate paths are illustrated by the thin solid lines.

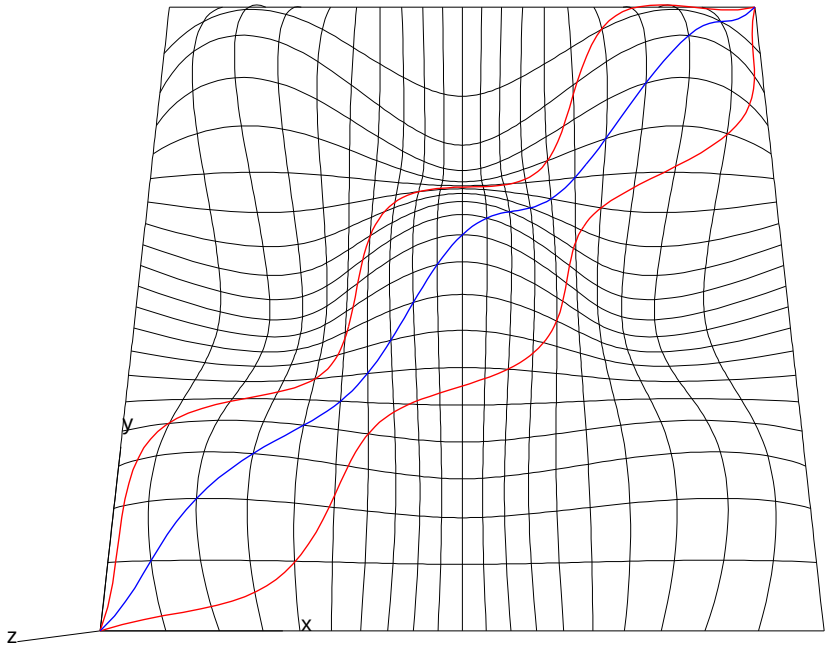

Figure 5: Geodesic paths on the wave-like bicubic B-spline surface between points of two corners.

The second example is a biquadratic rational B-spline surface, whose control polyhedron is a lattice of 6×9 vertices in u, v. This surface was constructed by sweeping a circle of radius 0.5, the generatrix, along a helix

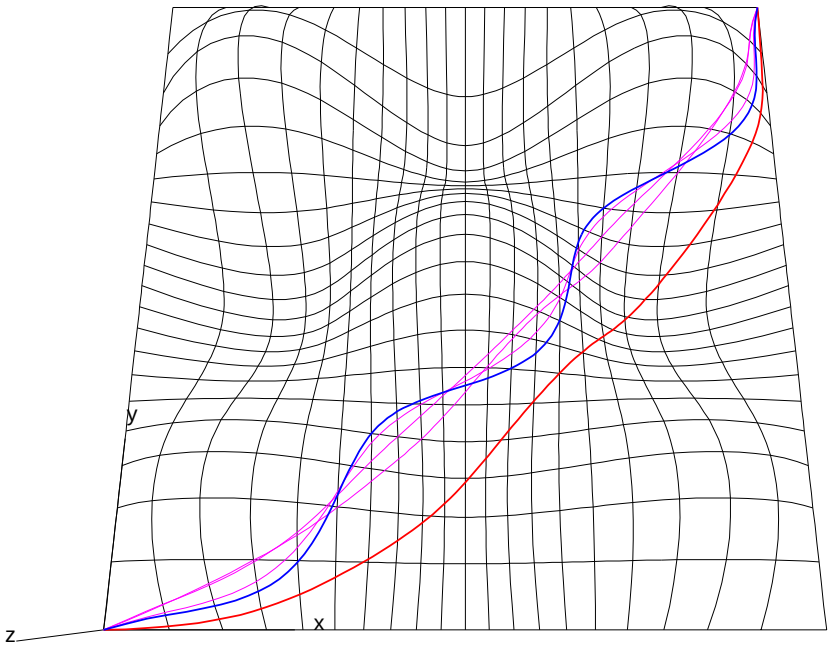

Figure 6: Convergence of the right geodesic path in Figure 4.

Correction Factor	Points	Tolerance	Iterations			Geodesic Distance		
μ	m	ϵ_N	L	M	R	L	M	R
eqn. (65)	501	1.0E-2	8	10	8	5.860	6.983	5.860
	1001	5.0E-3	10	10	10	5.843	6.956	5.843

Table 2: Numerical conditions and results for the computation of the geodesic path between points on generalized cylinder

($x = \cos t$, $y = -\sin t$, $z = \frac{t}{\pi}$, $0 \leq t \leq 2\pi$), the spine, and is approximated by rational B-spline interpolating a number of generatricies. When we keep u constant, we obtain a curve on the surface which depends only on v. This curve coincides with the generatrix. Similarly $v = constant$ represents another isoparametric curve which is parallel to the spine. Two end points are chosen to be (u_A, v_A)=(0, 0.4) and (u_B, v_B)=(1, 0.6) as shown in Figure 7. In Figure 7 three initial approximations are illustrated by the thin solid lines, while the final solutions are illustrated by the thick solid lines. Two circular arcs are determined by setting $(u_C, v_C) = (0.578, 0.108)$, $(u_D, v_D) = (0.422, 0.892)$. Algorithm A starts with these two circular arcs and converges to the two minimal geodesic paths which are shown as thick solid lines close to the initial circular arcs. Then the algorithm computes the mid-points of these solutions, which is shown as a thin straight line connecting the two end points. This initial approximation converges to a sine wave-like solution. This solution is a geodesic path but it does not provide the shortest distance. Table 2 shows the list of computational conditions and results as in Table 1. Since the surface is relatively complex, the constant step correction equation (65) is used.

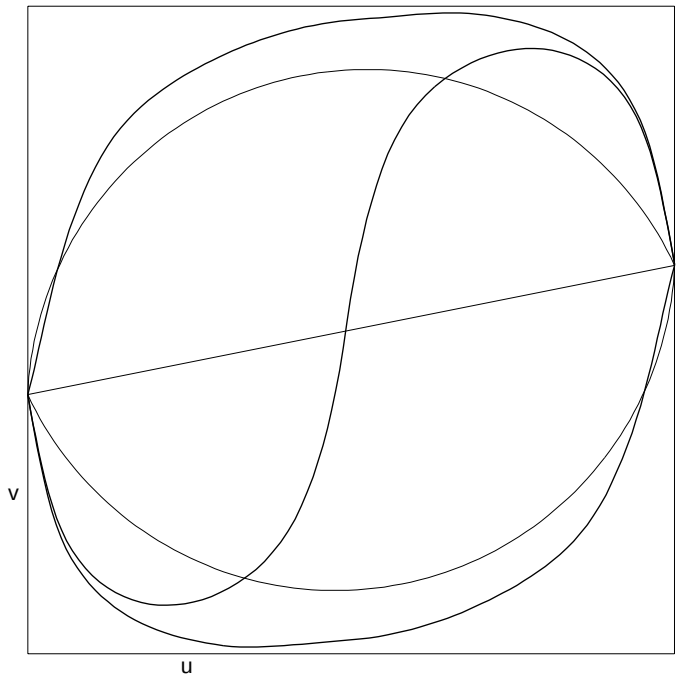

Figure 7: Geodesic paths in the parameter domain of the generalized cylinder.

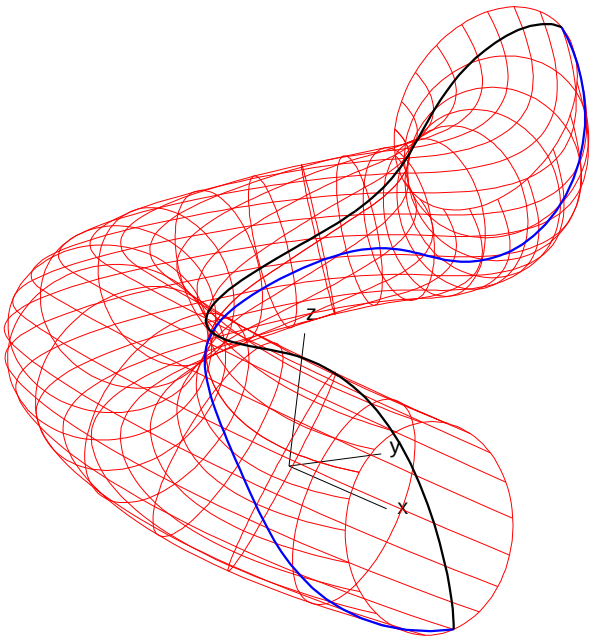

Figure 8: Minimal geodesic paths on the generalized cylinder between points of two circular edges.

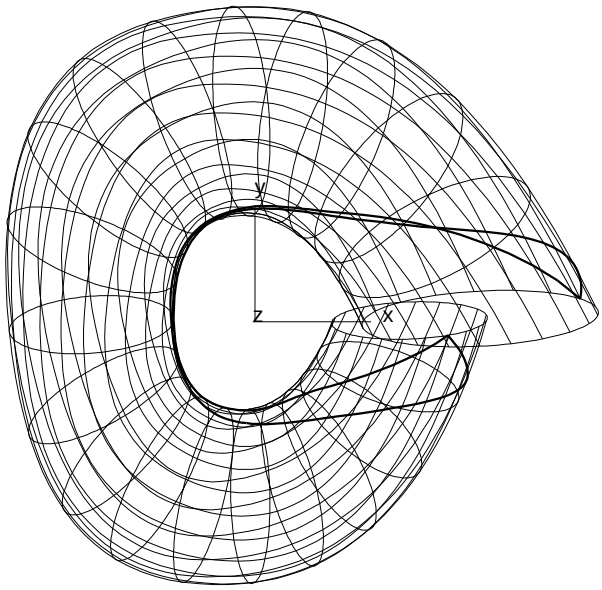

Figure 9: Top view of the minimal geodesic paths on the generalized cylinder between points of two circular edges.

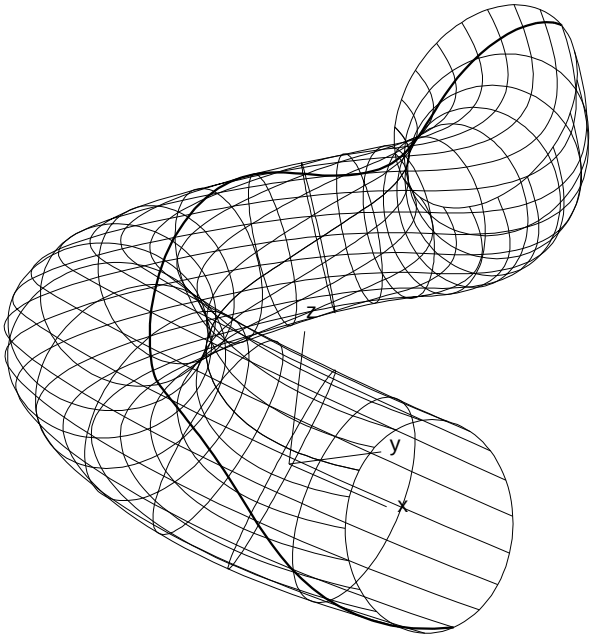

Figure 10: Geodesic path on the generalized cylinder which is not the shortest path.

t_1	t_2	t_B	m	μ	ν	ϵ_N	ϵ_S	Iteration	Geodesic Distance
0	0.02	0.2663	101	0.2	0.05	1.0E-3	1.0E-6	16	0.2754
1	0.98	0.7266	101	0.2	0.05	1.0E-3	1.0E-6	14	0.3712
0.4964	0.5164	0.5788	101	0.2	0.05	1.0E-3	1.0E-6	8	0.3874

Table 3: Numerical conditions and results for the computation of the geodesic path between a point and a curve on wave-like surface

6.2 Geodesic Path between a Point and a Curve

Figure 11 shows a planar cubic Bézier curve and point A (0.3, 0.2) in the uv parameter domain, which will be mapped onto the wave-like B-spline surface as shown in Figure 12. The algorithm finds three geodesic paths AB, AB' and AB'', whose tangent vectors at B, B' and B'' are orthogonal to the tangent vectors at the curve. Table 3 shows the list of computational conditions and results. The entries t_1, t_2 and t_B are the parameter values of the curve corresponding to the first two initial approximations for the secant method and the solution value. The following entries, m, μ and ν are the number of mesh points, correction factors for the Newton and secant methods. Tolerances for the convergence of Newton and secant methods are given by ν, ϵ_N.

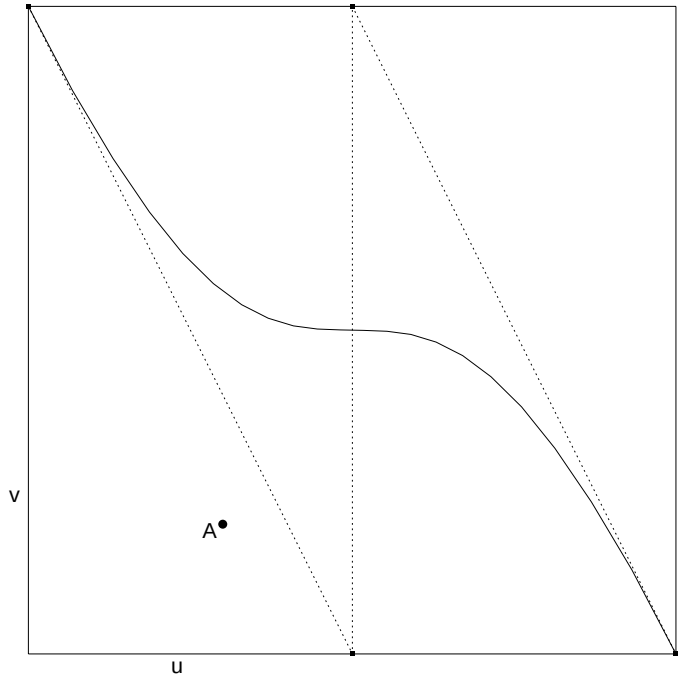

Figure 11: Cubic Bézier curve in the parameter domain.

6.3 Conclusions and Recommendations

The method developed in this work allows efficient and reliable computation of the shortest path between two points and a point to a curve on a free-form parametric surfaces. A basic element of our method is to solve the two point boundary value problem by the relaxation method based on a finite difference discretization. Step controlled correction procedure in the Newton's method makes it possible to start with initial vectors which are

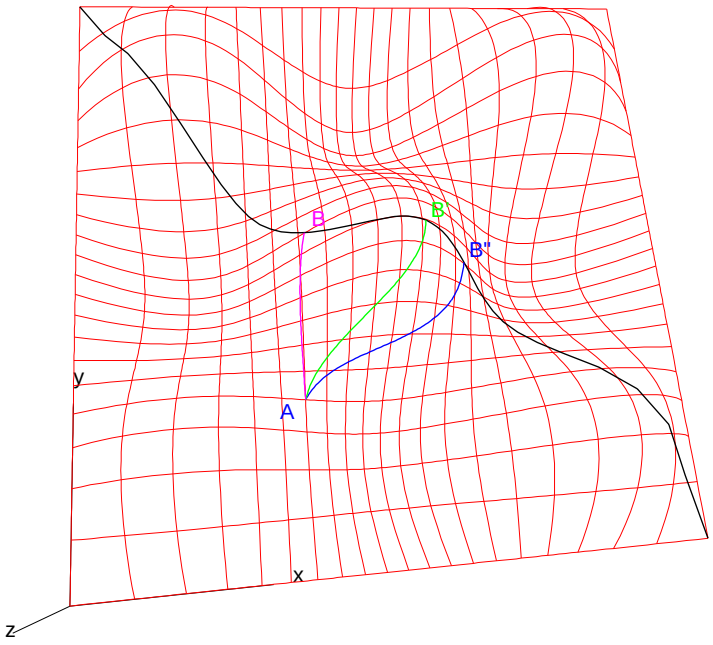

Figure 12: Geodesic paths from point A to Bézier curve on the wave-like bicubic B-spline surface.

not necessarily close to the solution vector, which helps to find all the geodesic paths between two points and a point to a curve.

Finally, the following topics are recommended for further study.

- Adaptive step size: We can achieve greater efficiency if we could provide more mesh points where the variables are changing rapidly to achieve accuracy, while less mesh points should be provided in regions where the variables are changing slowly. Therefore it is recommended to allocate mesh points during the relaxation process.
- Our algorithms fail once the intermediate path goes outside of the parameter domain. This can be fixed by gluing the surface patches to all the four boundaries.

Acknowledgements

I would like to thank Prof. Patrikalakis for productive discussions. Funding was provided by the Office of Naval Research (under grant number N00014-94-1-1001).

Appendix: Difference Formulas for Non-Uniform Grids

Forward Difference

$$f'(x_i) = \frac{-\frac{h_i}{h_{i+1}} f_{i+1} + \left(\frac{h_i}{h_{i+1}} + \frac{h_{i+1}}{h_i} + 2\right) - \left(2 + \frac{h_{i+1}}{h_i}\right) f_{i-1}}{h_i + h_{i+1}} \tag{78}$$

Central Difference

$$f'(x_i) = \frac{\frac{h_i}{h_{i+1}}(f_{i+1} - f_i) - \frac{h_{i+1}}{h_i}(f_{i-1} - f_i)}{h_i + h_{i+1}} \tag{79}$$

Backward Difference

$$f'(x_i) = \frac{\frac{h_{i+1}}{h_i} f_{i+1} - \left(\frac{h_i}{h_{i+1}} + \frac{h_{i+1}}{h_i} + 2\right) + \left(2 + \frac{h_i}{h_{i+1}}\right) f_{i-1}}{h_i + h_{i+1}} \tag{80}$$

References

[1] J. M. Beck, R. T. Farouki, and J. K. Hinds. Surface analysis methods. *IEEE Computer Graphics and Applications*, 6(12):18–36, December 1986.

[2] G. A. Bliss. The geodesic lines on the anchor ring. *Annals of Mathematics*, 4:1–21, October 1902.

[3] G. Dahlquist and A. Björck. *Numerical Methods.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.

[4] P. M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[5] J. H. Ferziger. *Numerical Methods for Engineering Applications.* Wiley, 1981.

[6] H. B. Keller. *Numerical Methods for Two-Point Boundary Value Problems.* Blaisdell, 1968.

[7] R. Kimmel, A. Amir, and A. M. Bruckstein. Finding shortest paths on surfaces using level sets propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):635–640, June 1995.

[8] E. Kreyszig. *Differential Geometry.* University of Toronto Press, Toronto, 1959.

[9] J. S. B. Mitchell. An algorithmic approach to some problems in terrain navigation. *Artificial Intelligence*, 37:171–201, 1988.

[10] F. C. Munchmeyer and R. Haw. Applications of differential geometry to ship design. In D. F. Rogers, B. C. Nehring, and C. Kuo, editors, *Proceedings of Computer Applications in the Automation of Shipyard Operation and Ship Design IV*, volume 9, pages 183–196, Annapolis, Maryland, USA, June 1982.

[11] N. M. Patrikalakis and L. Bardis. Offsets of curves on rational B-spline surfaces. *Engineering with Computers*, 5:39–46, 1989.

[12] W. H. Press et al. *Numerical Recipes in C.* Cambridge University Press, 1988.

[13] J. Sneyd and C. S. Peskin. Computation of geodesic trajectories on tubular surfaces. *SAIM Journal of Scientific Statistical Computing*, 11(2):230–241, March 1990.

[14] D. J. Struik. Outline of a history of differential geometry. *Isis*, 19:92–120, 1933.

[15] D. J. Struik. *Lectures on Classical Differential Geometry.* Addison-Wesley, Cambridge Mass., 1950.

[16] F.-E. Wolter. *Cut Loci in Bordered and Unbordered Riemannian Manifolds.* PhD thesis, Technical University of Berlin, Department of Mathematics, December 1985.