Dimas Martínez Morera · Paulo Cezar Carvalho · Luiz Velho

Modeling on Triangulations with Geodesic Curves

Abstract In the first part of this paper we define a new class of curves, called *geodesic Bézier curves*, that are suitable for modeling on manifold triangulations. As a natural generalization of Bézier curves, the new curves are as smooth as possible. In the second part we discuss the construction of C^0 and C^1 piecewise Bézier splines. We also describe how to perform editing operations, such as trimming, using these curves. Special care is taken to achieve interactive rates for modeling tasks. The third part is devoted to the definition and study of convex sets on triangulated surfaces. We derive the convex hull property of geodesic Bézier curves.

Keywords Geodesic Bézier Curve · Discrete Geodesic · de Casteljau Algorithm · Spline Curves · Free-Form Design · Convex Sets on Surfaces

1 Introduction

Designing free-form curves is a basic operation in Geometric Modeling. Doing so in Euclidean space is a widely studied problem; see [7,10] and many others. The problem becomes harder, however, if we wish to design on a curved geometry, such as triangulated surfaces. Most existing work for the later task, relies on imposing a suitable parameterization, which is usually an unintuitive approach that leads to a series of "trial and error" operations. We pursue instead a direct design in the geometry of the surface, an approach that has received much less attention.

In this paper we introduce a new class of curves that are suitable for free-form modeling directly on the geometry of a manifold mesh. Defined by means of the the de Casteljau Algorithm, they are a natural generalization of Bézier curves. Thus, we call them "Geodesic Bézier Curves". In order to stablish the convex hull property of these curves, we give an intrinsic definition of convex sets on manifold triangulations, and make a deep study of their properties. As a side result we prove the convergence of our geodesic algorithm [21].

1.1 Overview

We begin in section 2 with a summary of related work. In section 3 we give a brief overview of the state of the art of Discrete Geodesic computation and present a simple description of the algorithm given by [21], which will be used in sections 4 and 5. After summarizing classical Bézier curves theory, we define in section 4 the new class of curves and compare them to the classical ones. A study of their use in modeling operations is done in section 5. Section 6 describes the construction of piecewise Bézier spline curves on triangulations. In section 7 we define convex sets in the context of triangulations and study their properties. Convex hulls in that context are presented in section 8. In section 9 we present the two applications of convex sets mentioned above. Finally in section 11 we give concluding remarks and indicate potential further research.

1.2 Notation and Preliminary Definitions

A polygonal line Γ on a triangle mesh $\mathcal{S}$ is defined as a sequence of nodes $\{\Gamma_0, \Gamma_1, \ldots, \Gamma_n\} \subset \mathcal{S}$ such that every line segment $\overline{\Gamma_i \Gamma_{i+i}}$ is also contained in $\mathcal{S}$. We refer to polygonal vertices as nodes, in order to differentiate them from mesh vertices.
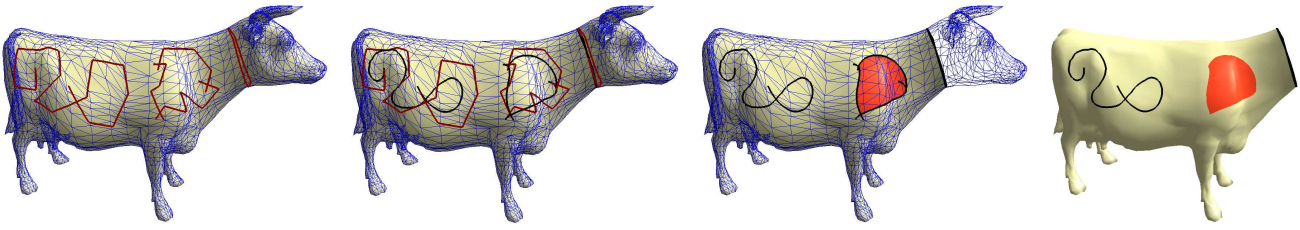
D. Martínez
E-mail: dimasmm@impa.br

P. C. Carvalho
E-mail: pcezar@impa.br

L. Velho
E-mail: lvelho@impa.br

Instituto Nacional de Matemática Pura e Aplicada - IMPA
Estrada Dona Castorina, 110, Jardim Botânico, Rio de Janeiro, RJ. 22460-320
Tel.: +55-21-25295080

Fig. 1 Modeling on the surface of a cow. From left: the control polygons of some curves and a C^1 spline, the corresponding curves, a region is filled and other is trimmed, final result.

Curves defined over a triangulation $\mathcal{S}$ cannot be smooth, the only exception being when they are completely defined on a planar part of $\mathcal{S}$, which is usually not the case. However, as continuity is a local property we can analyze the geodesic behavior of a curve $\mathcal{C}$ on $\mathcal{S}$ by looking at the intersection of the neighborhoods of each curve point with $\mathcal{S}$.

We define C^k continuity at points having a neighborhood isometric to a plane as the usual C^k continuity in the unfolding of that neighborhood. In particular, this applies to points lying in the interior of mesh edges and faces.

In the case of mesh vertices, continuity is not well defined. However, geodesic Bézier curves in practice are well behaved when they pass through mesh vertices. Furthermore, we are currently investigating the theoretical aspects of curve continuity at mesh vertices.

2 Related Work

The de Casteljau algorithm has been adapted to Riemannian manifolds using geodesic interpolation [24]. However, it has been applied only to some surfaces where geodesic computation is relatively easy, such as spheres or Lie groups; see [8,27] and the references therein for details. The only modeling operation studied in those works is the construction of cubic splines. Furthermore, trimming seems to be very hard to perform in this setting. We define curves using the same idea as in [24], but we consider manifolds triangulations.

Another generalization of cubic splines to smooth manifolds is given in [11,26]. The curves are defined by interpolating a set of knot points on the surface, minimizing an energy functional. These results are also applicable to triangulations, but they require the computation of local smooth approximations. If the position of a knot is changed, the whole curve must be recomputed since there is no local control of its shape. On the other hand, the interpolation of tangent vectors or higher derivatives has not been studied. Using geodesic Bézier curves we can overcome these difficulties, and it becomes possible to prescribe derivatives at knot points and to have local control of the segments of the Bézier spline curve.

Trimming is a very important application in CAGD. This operation is usually done by means of a parame-terization. One has to figure out which curve in the parameter space corresponds to the desired curve on the surface, which is generally a difficult problem. Most of the time these curves are obtained as the result of CSG operations, or more general surface intersections [17,6]. An approach for subdivision surfaces is to modify the original (coarse) mesh in order to obtain a trimmed limit surface [3,19]. Our trimming operations are done directly on the mesh and no parameterization is needed.

Recent works [31,30,29] define a general framework for curve subdivision schemes. Geodesic Bézier curves fits into this framework. However, smoothness of limit curves is only studied – and proved – in the case of smooth manifolds, considering meshes as an approximation of smooth surfaces, what is not always the case. For example, a potential user may be interested in modeling on a coarse mesh instead of a refined one. Besides, models with sharp features are best approximated with non-smooth surfaces. In this paper we analyze the smoothness of geodesic Bézier curves in the context of triangular meshes, and also how to handle modifications in the position of control points at interactive rates, what is not done in those works. Our results are also applicable to the other geodesic-based subdivision schemes fitting into this framework.

3 Geodesic Curves

The problem of computing locally shortest geodesic paths on discrete geometries, particularly meshes, has been addressed in many works [2,5,12,15,23], and it is still subject of active research [21,28]. Most algorithms use the so-called Continuous Dijkstra Technique. The algorithm proposed in [21] adopts an iterative curve correction strategy that we believe is suitable to our curve design algorithm, with the goal of reaching responsive user interaction without increasing storage space. We will go back to this subject in section 4.

3.1 Iterative Curve Correction Algorithm

In this section we summarize the Iterative Curve Correction Algorithm; for details see [21]. Geodesic computation is performed in two steps. In the first step, an initial

polygonal curve, joining two points in the mesh, is computed using a front propagation strategy. In the second step, all the nodes of the initial curve are put in a priority queue; then the node with largest error is corrected and the error at neighboring nodes is updated. This process is repeated until a small error is attained.

Nodes are constrained to lie on mesh edges since a geodesic must coincide with a line segment in the interior of each face, and the extremes of the curve are added as new vertices to the mesh. Errors at curve nodes are computed based on discrete geodesic curvature [25]. A node position is corrected by unfolding a subset of the faces adjacent to it and moving it to the line joining its neighboring nodes in the unfolded part of the mesh.

Since our curves are allowed to pass through the interior of a face – not necessarily as a line segment – we would need to add new vertices to the mesh any time we subdivide a control polygon. However, a careful implementation would allow geodesic nodes to lie in the interior of mesh faces, leaving the underlying mesh intact.

4 Geodesic Bézier Curves

Bézier curves are of great importance when modeling on $\mathbb{R}^n$. A natural question is how to generalize them to curved geometries. In this section we propose a class of curves that generalize Bézier curves to manifold triangulations.

4.1 Classical Bézier Curves

Given n+1 control points $P_0, P_1, \ldots, P_n$ in $\mathbb{R}^d$, they define a curve given by the following parametric expression:

$$P(u) = \sum_{i=0}^{n} \binom{n}{i} (1-u)^{n-i} u^i P_i, \quad 0 \le u \le 1, \qquad (1)$$

P is known as Bézier curve of degree n, and the set of control points $P_0, P_1, \ldots, P_n$ forms its control polygon. Note that P interpolates the two extreme control points P_0 and P_n, being tangent to the control polygon at these points. It also "imitates" the form of the control polygon, making the task of designing with Bézier curves very intuitive. That's the reason why Bézier curves are so popular for CAD/CAGD applications. More information about this subject can be found in [7,10]; figure 2 shows an example of a Bézier curve of degree 3.

The de Casteljau Algorithm [4] provides a geometric procedure to evaluate a Bézier curve at any parameter $u \in [0,1]$, using repeated linear interpolation:

Algorithm 1 : *de Casteljau*

Input: The control points $P_0, P_1, \ldots, P_n$ and a parameter $u \in [0,1]$
Output: The point $P(u)$.
   step 1. for $i = 0, \ldots, n$ set $P_i^{[0]}(u) = P_i$
   step 2. for $j = 1, \ldots, n$
        for $i = j, \ldots, n$
          $P_i^{[j]} =$**interpolate**$(P_{i-1}^{[j-1]}(u), P_i^{[j-1]}(u), u)$
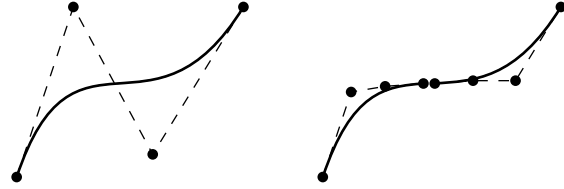   **step 3.** $P(u) = P_n^{[n]}$

Fig. 2 A subdivision step of a control polygon and its Bézier curve.

In step 2 we use the function **interpolate**(A, B, u), which performs a linear interpolation between A and B with parameter u: **interpolate**$(A, B, u) = (1-u)A + uB$.

From de Casteljau's algorithm one can define a subdivision scheme whose limit curve is the Bézier curve given by the control polygon. Given a parameter value u and a control polygon, we can obtain two new control polygons for the segments $P([0, u])$ and $P([u, 1])$:

Algorithm 2 : *Subdivision of a control polygon*

Input: The control points $P_0, P_1, \ldots, P_n$ and a parameter $u \in [0,1]$
Output: Two sets of control points defining $P([0, u])$ and $P([u, 1])$.
   step 1. deCasteljau$((P_0, P_1, \ldots, P_n), u)$.
   step 2.
        $P([0, u]) =$**bezier**$(P_0^{[0]}, P_1^{[1]}, \ldots, P_n^{[n]})$
        $P([u, 1]) =$**bezier**$(P_n^{[n]}, P_n^{[n-1]}, \ldots, P_n^{[0]})$

Evaluating the curve at u, using de Casteljau's algorithm, provides the intermediary interpolated points $P_i^{[j]}$. The output (step 2) are the control polygons defining both (Bézier) segments of the curve. Figure 2 shows a subdivision step of the control polygon of a degree 3 Bézier curve.

Algorithm 2 provides the rule for the subdivision scheme converging to the curve. Additionally, this scheme can be made adaptive by stopping the subdivision whenever a control polygon can be considered as "almost straight".

4.2 Bézier Curves on Triangulations

Geodesic Bézier curves are defined by means of the subdivision algorithm for classical Bézier curves. Given a control polygon $P_0, P_1, \ldots, P_n$ on a surface $\mathcal{S}$, we want to compute a curve $\mathcal{C}$ on $\mathcal{S}$ interpolating P_0 and P_n, whose shape is controlled by the position of the interior points $P_1, P_2, \ldots, P_{n-1}$.

The curve $\mathcal{C}$ is defined as the limit of the subdivision scheme given in algorithm 2 of section 4.1. The main difference is that the sides of the control polygon are no longer line segments, but geodesics connecting the control points. This imposes the necessity of modifying the interpolation

Algorithm 3 : *Interpolation on Triangulations*

Input: A manifold triangulation $\mathcal{S}$, two points Q_1 and Q_2 on it and a parameter $u \in [0, 1]$
Output: A point Q on $\mathcal{S}$ interpolating Q_1 and Q_2.
  step 1. $\gamma =$**ComputeGeodesic**(Q_1, Q_2).
  step 2. $Q =$ the point of γ such that
$$\mathrm{d}_\gamma(Q_1, Q) = u\mathrm{d}_\gamma(Q_1, Q_2)$$

---

step on algorithm 1. The equivalent to linear interpolation in the geometry of the surface is the interpolation along geodesic lines. Algorithm 3 describes the interpolation step in the case of manifold triangulations. In this algorithm, $\mathrm{d}_\gamma(A, B)$ computes the distance between A and B along γ. Note that since γ is a polygonal line, it is very simple to perform step 2.

To compute an approximation of $\mathcal{C}$, we can use the subdivision adaptive algorithm. It stops at some prescribed level of subdivision or when the control polygon can be considered as a geodesic segment; i.e., when all of its control vertices have error smaller than a prescribed tolerance. Figure 3 shows some geodesic Bézier curves along with their control polygons.
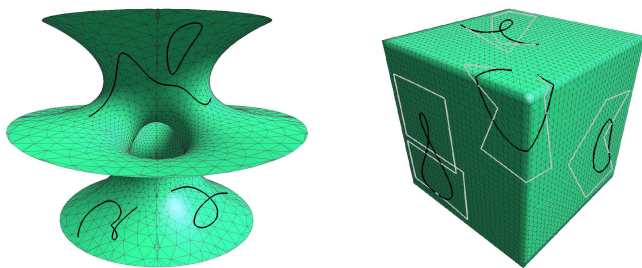

Fig. 3 Some geodesic Bézier curves. Control polygons are also shown in the Cube model.

The use of de Casteljau's algorithm in the definition of geodesic Bézier curves makes them a generalization of planar Bézier curves. Note that a geodesic on a plane is a straight line. Therefore, when the triangulation is planar both concepts coincide; see for example the curves designed on the (planar) faces of the cube in figure 3. As in the case of classical Bézier curves, we have a parameterization of our curves with parameter $u \in [0, 1]$. Evaluation at any particular parameter value can be performed easily by subdividing the corresponding control polygon at each level of subdivision. Previous calculations can be used to evaluate at new parameter values, this can be useful when performing many evaluations.

It is known that shortest geodesics are not unique on triangulations. Consequently, the use of a different subdivision parameter may lead to a different curve. So geodesic Bézier curves depends on the control points and the chosen subdivision parameter u. To our experience, the curves obtained with different values of u are very close to each other. We are currently studying the theoretical issues related to the choice of u. In practice, selecting a fixed value of u gives us a subdivision curve. In this paper we have chosen $u = 0.5$ and therefore we have a midpoint subdivision scheme. All figures of this paper were generated using this scheme.

Geodesic algorithm selection

There are several algorithms to compute geodesics (see section 3) and any of them could be used both to compute geodesic Bézier curves and to perform user interaction. We chose the algorithm of [21] for two reasons. In first place, it relies on the correction of an initial curve assumed to be close to the true geodesic. Since de Casteljau's algorithm is a sort of corner-cutting process, a part of each control polygon can be used as initial curve to compute the geodesics needed in the computation of control polygons in the following level of subdivision. On the other hand, during interaction the new control polygons are very close to the previous ones and they can be computed very fast. Using other algorithms as [28] also permits very fast interaction, but at the cost of storing a tree for each control point. The tree associated with a control point must be updated any time its position is changed.

4.3 Properties of Geodesic Bézier Curves

Geodesic Bézier curves share some properties with classical ones. The proofs of the following propositions can be found in [20].

Proposition 1 *Geodesic Bézier curves interpolate P_0 and P_n and are tangent to the control polygon at these points.*

Proof Consider the first line segment of the first side (geodesic line) of the control polygon. It is entirely contained in a face F and for N large enough, the initial sub-polygon at level N is entirely contained in F. So, the corresponding curve segment is a segment of a planar Bézier curve, which is tangent to the first segment of

the Nth level subpolygon, that is contained on the first side of the 1st level polygon. So, the curve is tangent to its control polygon at P_0; the same applies to P_n. □

Each interior point of an edge has a neighborhood which is isometric to an open disc in the plane. By means of this isometry, we can analyze the behavior of a curve when passing through the interior of mesh edges. If the curve is smooth in the plane, it will have smooth appearance in the mesh. The following proposition address this property of geodesic Bézier curves.

Proposition 2 *A geodesic Bézier curve has (at least) C^1 continuity when intersecting an edge of the mesh.*

Proof Let s_0 be the parameter value corresponding to the point $P(s_0)$ of $\mathcal{C}$ intercepting the edge e of mesh $\mathcal{S}$. We have two cases: (i) for N large enough, there is a segment $\mathcal{C}_{N,i}$ of $\mathcal{C}$ completely contained in the union of the two faces adjacent to e, and such that $P(s_0) \in \mathcal{C}_{N,i}$; or (ii) case (i) does not hold and $s_0 = k/2^N$, where k is an integer such that $0 < k < N$.

In case (i) $\mathcal{C}$ is C^∞ at $P(s_0)$, since it is a point of a plane Bezier curve defined in the unfolding of e. If case (ii) holds, there is an N large enough such that each of the two polygons corresponding to level N of subdivision is contained in one of the two faces adjacent to edge e. So, in the unfolding of the two faces, the two segments of control polygons at level N with $P(s_0)$ as endpoint are collinear and have the same length. As a consequence the plane spline defined by these two control polygon is (at least) C^1 at its junction point $P(s_0)$, and hence $\mathcal{C}$ is C^1 at $P(s_0)$ too. □

Proposition 3 *Each connected component of the intersection of a geodesic Bézier curve with the interior of a mesh face is a C^∞ plane curve, except for (at most) a countable set of points, where it is C^1.*

Proof Let f be a face of $\mathcal{S}$, and s the parameter value corresponding to the point $P(s)$ of $\mathcal{C} \bigcap f$.

As in the preceding proposition, there are two cases: (i) for N large enough, there is a segment $\mathcal{C}_{N,i}$ of $\mathcal{C}$ completely contained in f and such that $P(s) \in \mathcal{C}_{N,i}$; or (ii) case (i) does not hold and $s = k/2^N$, where k is an integer such that $0 < k < N$.

In case (i) $\mathcal{C}$ is C^∞ at $P(s)$, since it is a point of a plane Bezier curve defined in f. If case (ii) holds, there is an N large enough such that each of the two polygons corresponding to level N of subdivision is contained in f. So, the two segments of control polygons at level N with $P(s)$ as endpoint are collinear and have the same length. As a consequence the plane spline defined by these two control polygon is (at least) C^1 at its junction point $P(s)$, and hence $\mathcal{C}$ is C^1 at $P(s)$ too.

Note that the set of points satisfying condition (ii) is finite in almost all the cases. When $\mathcal{C}$ pass through a vertex of f, this set can be infinite. However, it is countable, since it is a subset of $\{s = k/2^N, \text{ s.t. } k, N \in \mathbb{N}, k < N\}$ which is a countable set. □

As a consequence of previous propositions we have that $\mathcal{C}$ is as smooth as possible in the interior of faces and when crossing a mesh edge. The analysis of the passage of $\mathcal{C}$ through mesh vertices is more complicated and is part of our current research.

The Convex Hull property of Bézier curves has a huge importance in modeling. The adaptive version of de Casteljau's algorithm relies on this property. It is not trivial to give a proper definition of *convex set* in a curved geometry. However, in section 7 we give definitions of convex set and convex hull that are appropriated for the study of our curves. Propositions 6 and 9 guarantee the correctness of the adaptive version of geodesic Bézier curves.

5 Modeling

In order to model with geodesic Bézier curves we must be able to perform the usual modeling operations. Moreover, the user should be allowed to modify a curve at interactive rates. In this section we first describe how to efficiently handle user interaction. Following that, we present a simple algorithm for region fill and trimming.

5.1 User Interaction

Fast user interaction is very important in free-form design operations. The user should be able to modify any previously defined curve by changing the position of some of its control points. This operation should be as fast and easy as possible; for example, it must be possible to select and drag any control point using the mouse. Every time the position of a control point is changed, its neighboring sides in the control polygon should be recomputed. These (at most two) sides are geodesic lines and we must recompute them very fast, at least approximately. Each new (recomputed) geodesic is very close to the old (original) one, since one of their extremes remain fixed while the other one is very close to the corresponding extreme in the old curve. Hence we use, as initial approximation for the algorithm described in section 3, the original curve after adding to it the line segment joining its extreme to the new control point position. Because the initial segment is very close to the recomputed one, this update process runs very fast. Additionally, we force the geodesic computation to perform fewer iteration steps during interaction since the user only needs to have a good idea of the shape of the control polygon. When the user releases the mouse, full-precision geodesics are computed and the curve is then recomputed. Figure 4 shows three different positions for the middle node during user interaction with a third order curve.
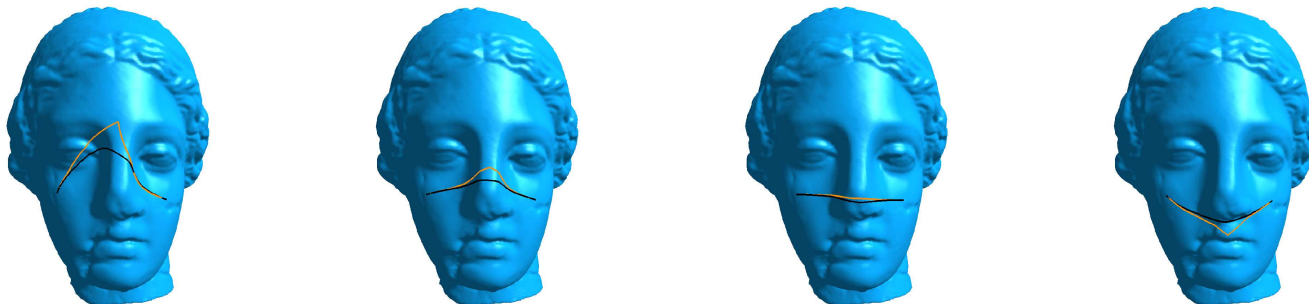
Fig. 4 Igea model: Four different positions for the middle control point in a curve with three control points.

5.2 Region Fill and Trimming

We are now concerned with the problem of identifying a piece of a surface $\mathcal{S}$ limited by one or more curves defined on it. Solving this problem allows us to trim (cut) a piece of $\mathcal{S}$, to paint it with a certain color, or to map a texture to it. Given a point P in $\mathcal{S}$, typically obtained by a mouse click, the idea is to use a flood-fill algorithm, propagating a wavefront from this point until it reaches the boundary curves. Algorithm 4 describes how to identify the faces in the region $\mathcal{R}$ that contains the point P.

Algorithm 4 : *Identify region*

Input: A point $P \in \mathcal{S}$
Output: The set $S_{\mathcal{R}} = \{f \in \{\text{faces of } \mathcal{S}\}, \text{s.t.} f \bigcap \mathcal{R} \neq \emptyset\}$
   step 1. f = face containing P.
   step 2. push($f, L_{\mathcal{R}}$)
   step 3. while $L_{\mathcal{R}}$ is not empty
         $g = \textbf{pop}(L_{\mathcal{R}})$
         push($g, S_{\mathcal{R}}$)
         for $h \in \{\text{neighbors of } g\}$
            if(**can_propagate**($g \mapsto h$))
               push($h, L_{\mathcal{R}}$)

In algorithm 4 above, $L_{\mathcal{R}}$ is an auxiliary list of faces. The function **can_propagate** returns **true** if the following three conditions hold:

1. h does not belong to $L_{\mathcal{R}}$,
2. h does not belong to $S_{\mathcal{R}}$, and
3. $\mathcal{R}$ contains the edge common to g and h, or part of it.

In practice it is not necessary to know if condition 3 holds. Instead we only consider the faces that are adjacent to edges intersecting region $\mathcal{R}$, see figure 6. When $L_{\mathcal{R}}$ becomes empty we have in $S_{\mathcal{R}}$ all the faces contained in the interior of $\mathcal{R}$ and also the faces cutting the boundary curves. They are colored red and green respectively in figure 5 (left). Once we have identified the set $S_{\mathcal{R}}$ of faces cutting $\mathcal{R}$, we must decide which part of the boundary faces belongs to $\mathcal{R}$. To do that, during propagation we mark each portion of an edge intersecting $\mathcal{R}$, see figure 6. With this information we can decide which part of

Fig. 6 Propagation directions. Arrows indicate by what edges can the wave be propagated. Bullets indicate what portion of the edges belong to $\mathcal{R}$ (shadowed region).

the planar subdivision defined by the boundary curves in each face belongs to $\mathcal{R}$.

The above described process can easily be performed if the seed point P belongs to a face which is entirely contained in $\mathcal{R}$. If P belongs to a face crossed by the boundary of $\mathcal{R}$, we subdivide it until P is inside an interior face (see figure 7). For texture mapping or trimming
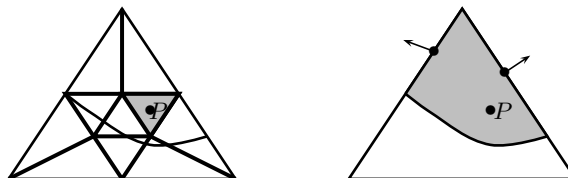

Fig. 7 Locating seed point

it is not sufficient to identify the part of $\mathcal{S}$ (i.e., the region $\mathcal{R}$) selected by the user. It is also necessary to have a model of it. In those cases we can triangulate the corresponding part of each face crossed by the boundary of $\mathcal{R}$. Figure 8 shows some regions filled or trimmed in the Cube and the Bunny models.

6 Piecewise Bézier Spline Curves

A powerful tool for modeling is the use of piecewise spline curves, allowing local control of the shape of the curve as well as faster computations by means of segments of low degree. We want to compute piecewise spline curves of geodesic Bézier curves , so next we investigate how to guarantee some continuity at junction points.
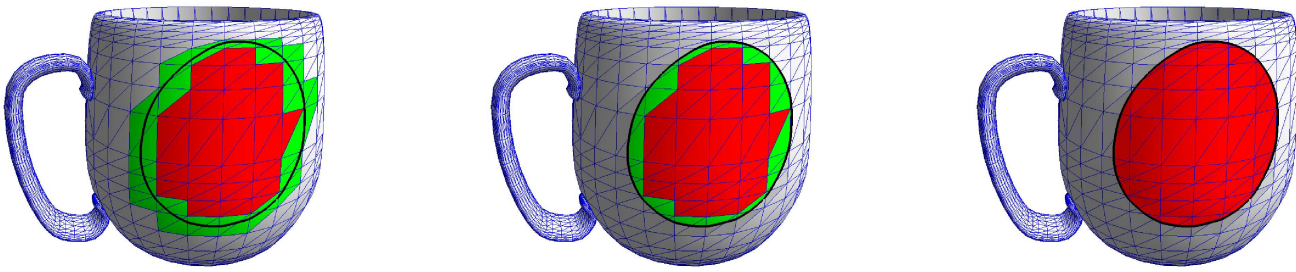
Fig. 5 Region finding stages. Left: set $S_{\mathcal{R}}$ with boundary faces highlighted. Middle and Right: the region $\mathcal{R}$ after eliminating the part of boundary faces not belonging to it.
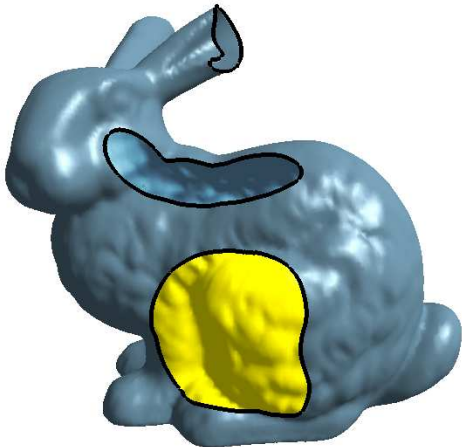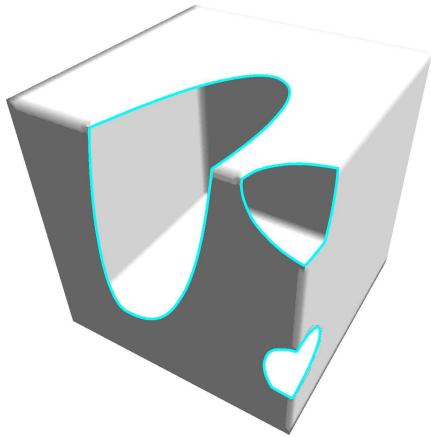

Fig. 9 C^0 and C^1 splines on the surface of the bunny.

Fig. 8 Filled and trimmed regions. Up: Trimmed Cube. Down: Stanford's bunny with two trimmed regions and a filled one.

As usual, C^0 continuity is reached by defining the first control point of a segment $\mathcal{C}_{i+1}$ to be the same as the last control point of its previous segment $\mathcal{C}_i$. To guarantee C^1 continuity is harder because we must have the last side of the control polygon of $\mathcal{C}_i$ aligned with the first side of the control polygon of $\mathcal{C}_{i+1}$. Moreover, the length of these two sides must be the same. This means that we need to locate three control points in the same geodesic line. In other words, the position of the two first control vertices of the segment $\mathcal{C}_{i+1}$ are determined by the position of the control vertices of the previous segment $\mathcal{C}_i$.

Given the control polygon of the ith segment $\mathcal{C}_i$ of a spline curve $\mathcal{C}$, how to compute the two first control points P_0^{i+1} and P_1^{i+1} of $\mathcal{C}_{i+1}$? Its first control point P_0^{i+1} is the same as the last control point P_m^i of $\mathcal{C}_i$. The second one, P_1^{i+1}, is hard to find because we do not know how to continue the geodesic line between P_{m-1}^i and P_m^i. For smooth surfaces we can compute the unique geodesic passing by a point in a direction. This is not the case for shortest geodesics on meshes. Nevertheless the straightest geodesics defined by [25] have this nice property. For that reason we define the first side of the control polygon of $\mathcal{C}_{i+1}$ as the straightest geodesic continuing the last side of the control polygon of $\mathcal{C}_i$. It is known [25] that if a straightest geodesic does not pass by a spherical vertex, it is also a shortest geodesic. So we can expect that most of the times our control polygon will be defined by means of shortest geodesics. It is important to note that all the properties of section 4.3 are also satisfied if we replace one or more of the shortest geodesics by straightest ones. Thus, the relaxation we did to the definition of the control polygon, in order to have C^1 continuity, is more than justified.

Finally note that modifying the position of P_{m-1}^i modifies the position of P_1^{i+1} and vice versa. In the last case, the last side of the control polygon of C_i will be a straightest geodesic. Modifying the position of the junction point conduce us to modify at least the position of one of the control points P_{m-1}^i and P_1^{i+1}. Figure 9 show the use of C^1 splines to write in the surface of the Stanford's bunny model. The middle curve in figure 1 is a C^1 spline, composed by 8 Bézier segments.

7 Convex Sets on Triangulations

The adaptive version of de Casteljau's algorithm relies in the convex hull property of classical Bézier curves. In order to have such a property for geodesic Bézier curves we need first to establish what do *convex set* mean in the context of triangualtions.

To define convex sets on triangulations requires some care. A naive definition, as the intersection of the surface with a convex set, will not work because it ignores the intrinsic geometry of the surface. There are different definitions of convex sets on triangulations, obtained by imitating the behavior of plane convex sets [2], but they are too restrictive. For example, non-optimal shortest geodesics are not convex in the usual sense. Our definition allow them to be convex.

In this section we define convex sets based on the geodesic curvature of the boundary curves, and derive in next section the concept of convex hull in manifolds. Those definitions are very useful to prove the convergence of the geodesic algorithm given by the authors in [21], and the convex hull property of geodesic Bézier curves.

7.1 Discrete Geodesic Curvature

Polthier and Schmies [25] defined *straightest geodesics* as curves with zero geodesic curvature

$$\kappa_g(P) = \frac{2\pi}{\theta}\left(\frac{\theta}{2} - \theta_r\right) = -\frac{2\pi}{\theta}\left(\frac{\theta}{2} - \theta_l\right),$$

where θ is the total angle at P and θ_r and θ_l are the angle formed by the curve to the right and left side respectively. In order to obtain a similar characterization for shortest geodesics, we must employ an alternative definition of discrete geodesic curvature:

Definition 1 The *discrete geodesic curvature* of a polygonal curve at a given point P is given by the following expression.

$$\kappa_s(P) = \begin{cases} 0, & \text{if } \theta(P) > 2\pi, \ \theta_r(P) \geq \pi \\ & \text{and } \theta_l(P) \geq \pi \\ \infty, & \text{if } \theta(P) < 2\pi \ \text{and } \theta_r(P) = \theta_l(P) \\ \kappa_g(P), & \text{otherwise} \end{cases}$$

Remark 1 We use the notation κ_s to distinguish this new definition of discrete geodesic curvature from Polthier's one. The subindex s stands for shortest.

Remark 2 Boundary points are treated as hyperbolic, as done in [21].

With this alternative definition of geodesic curvature at hand, we can characterize discrete shortest geodesics on triangulations:

Proposition 4 *A polygonal curve Γ contained in a manifold triangulation $\mathcal{S}$ is a shortest geodesic if and only if its geodesic curvature κ_s is identically zero.*

Proof Let Γ be a shortest geodesic on $\mathcal{S}$ and P a node of Γ. If P does not coincide with a mesh vertex, or coincides with an euclidean one, then θ_r and θ_l are equals; consequently $\kappa_s(P) = \kappa_g(P) = 0$. If P coincides with a hyperbolic vertex then both θ_r and θ_l must be greater than π, hence $\kappa_s(P) = 0$. Finally, P cannot coincide with a spherical vertex.

On the other hand, suppose that Γ have zero geodesic curvature κ_s everywhere. That means that a small perturbation in the position of any vertex will increase the length of Γ. Thus, it is a local minimum of the length functional and consequently a shortest geodesic. $\square$

7.2 Convex Sets

It is known that if the boundary of a plane convex set is a smooth curve, then its curvature does not change its sign [9]. If the boundary is a polygonal line, then at each vertex the angle corresponding to the interior of the set is smaller that the exterior one. Based on these facts, we define convex sets on triangulations.

Definition 2 Let $\mathcal{C}$ be a connected subset of a surface $\mathcal{S}$. $\mathcal{C}$ is *convex* if its boundary $\partial\mathcal{C}$ can be parametrized by a closed curve $\Gamma : I \longrightarrow \partial\mathcal{C}$ such that:

1. $\forall t \in I \ \kappa_s(\Gamma(t)) \leq 0$, and
2. the interior of set $\mathcal{C}$ is always situated in the left side of Γ.

Remark 3 If we replace'$\leq$' by '$\geq$' and 'left' by 'right' in definition 2, we get an equivalent definition of convex set.

Remark 4 We do not require the curve Γ parameterizing $\partial\mathcal{C}$ to be simple. Otherwise, a simple geodesic curve would not be convex. Figure 10 shows two convex sets in a plane with a hole where the boundary curves are not simple.

Remark 5 Convex sets in a plane triangulation (without holes) are also convex in the usual sense. Unfortunately, the usual properties of planar convex sets do not hold in the general case for convex sets on triangulations. In section 8 we are going to see an example where the intersection of two convex sets is not connected. Besides, there are convex sets having two different points that
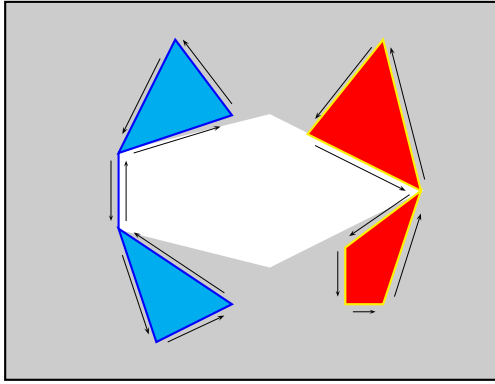
Fig. 10 In a plane with a hole, two convex sets whose boundary curves are not simple.

Fig. 11 A set in plane with a hole. It contains a shortest geodesic between any pair of points although it is not convex.

cannot be joined by an optimal shortest geodesic. However, a convex set always contains a shortest geodesic, not necessary optimal, joining two any points in it.

Proposition 5 *Let $\mathcal{C}$ be a convex set on the surface $\mathcal{S}$ and let A and B be two points of $\mathcal{C}$. Then there exists a shortest geodesic joining A and B that is completely contained in $\mathcal{C}$.*

Proof Consider the surface $\mathcal{S}'$ defined by the set $\mathcal{C}$. There exists a geodesic Γ joining A and B on $\mathcal{S}'$, see [23]. We must prove that Γ is a geodesic on $\mathcal{S}$.

If Γ does not touch the boundary of $\mathcal{S}'$ then it is also a geodesic on $\mathcal{S}$. Suppose now that the point $P \in \Gamma$ belongs to the boundary of $\mathcal{S}'$. If we choose an orientation of Γ such that the angle θ_l of Γ at P is the one in the interior of $\mathcal{S}'$, then θ_l must be greater than or equal to π, because boundary points are treated as hyperbolic vertices. This means that the angle θ_l of $\partial\mathcal{C}$ at P is also greater than or equal to π, but as $\mathcal{C}$ is a convex set then, looking to $\partial\mathcal{C}$ as a curve on $\mathcal{S}$, the angle θ_r of $\partial\mathcal{C}$ at P must also be greater than or equal to π. Hence, P is a hyperbolic or euclidean point on $\mathcal{S}$ and the geodesic curvature $\kappa_s(P)$ of Γ at P is also zero when looking to Γ as a curve on $\mathcal{S}$. This means that Γ is a geodesic on $\mathcal{S}$. □

Remark 6 The converse of proposition 5 does not hold. In figure 11 we show a non-convex set, contained in a plane with a hole, where every pair of points can be joined by a shortest geodesic.
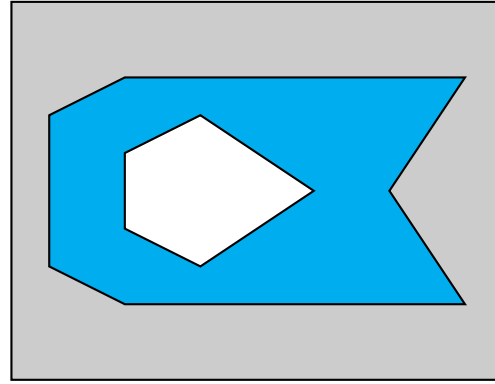
8 Convex Hull

We want to define convex hull as the intersection of convex sets. To do that, we must study if convexity is preserved by this set operation. Unfortunately, this is not true; for example, the intersection of two meridians in a sphere results in a two-point set which is not even connected.

Although intersection does not preserve convexity, we still can use it to define convex hull. The following lemma states that each connected component of the intersection is a convex set:

Lemma 1 *The intersection of two convex sets $\mathcal{C}_1$ and $\mathcal{C}_2$ is a collection of convex sets.*

Proof We know that the intersection of two convex sets may have more than one connected component. Let $\mathcal{C}$ be one of such connected components; we must prove that it is a convex set.

The boundary of $\mathcal{C}$ is formed by a sequence of curves belonging to the boundaries of $\mathcal{C}_1$ and $\mathcal{C}_2$. So, if we prove that κ_s remains smaller than or equal to zero at the intersection of the boundaries of $\mathcal{C}_1$ and $\mathcal{C}_2$, then we are done. But this is true because the left angle $\theta_l(\partial\mathcal{C}(P))$ at those points is smaller that both $\theta_l(\partial\mathcal{C}_1(P))$ and $\theta_l(\partial\mathcal{C}_2(P))$, and thus the geodesic curvature of $\partial\mathcal{C}$ at P is also negative:

$$\kappa_s(P) = \frac{2\pi}{\theta}\left(\frac{\theta}{2} - \theta_r\right) = \frac{2\pi}{\theta}\left(\frac{\theta_l - \theta_r}{2}\right) \leq 0.$$

The above formula holds everywhere, except at some hyperbolic vertices where θ_l is greater than π, but in those cases $\kappa_s(P) = 0$. Note that if P coincides with a spherical mesh vertex, then θ_l is necessary smaller than θ_r. □

After lemma 1 we can define the convex hull of curves on manifold triangulations:

Definition 3 *Let Γ be a curve defined on a surface $\mathcal{S}$. The convex hull $\overline{\Gamma}$ of Γ is the intersection of all convex sets containing Γ.*

$$\overline{\Gamma} = \bigcup_{\Gamma \subset \mathcal{C}_i} \mathcal{C}_i.$$

Remark 7 Note that we cannot extend the concept of convex hull in manifolds to arbitrary sets. For instance, the convex hull of two poles of a sphere would be the two points, which is not a connected set. However, this concept is well defined for curves, for which the convex hull is always a convex connected set. The following proposition will be useful in section 9.

Proposition 6 *Given a simple polygonal curve Γ, joining two different points A and B, the area of its convex hull $\mathcal{A}(\overline{\Gamma})$ is equal to zero if and only if Γ is a simple shortest geodesic.*

Proof If Γ is a simple shortest geodesic, then it is a convex set and consequently $\overline{\Gamma} = \Gamma$, so $\mathcal{A}(\overline{\Gamma}) = 0$.

Now assume that $\mathcal{A}(\overline{\Gamma}) = 0$, and suppose it is not a shortest geodesic. There exists an interior node $P_i \in \Gamma$ where $\kappa_s(P) \neq 0$. Hence, the set bounded by the geodesic joining P_{i-1} and P_{i+1}, and the segment of Γ between P_{i-1} and P_{i+1} belongs to $\overline{\Gamma}$. But the area of that set, which is a subset of $\overline{\Gamma}$, is not zero, which is a contradiction. $\qquad\square$

After defining convex hull, it is interesting to know how its shape depends on the curve. Next lemma answer this question for polygonal curves.

Lemma 2 *The boundary of the convex hull $\overline{\Gamma}$ of a polygonal curve Γ is composed of a sequence of geodesic segments whose extremes are nodes of Γ.*

Proof We know that $\kappa_s(\partial\overline{\Gamma}(.)) \leq 0$, suppose that $\kappa_s(P) < 0$ at a point $P \in \partial\overline{\Gamma}$. If $P \notin \Gamma$, then we can join two points of its neighboring sides in $\partial\overline{\Gamma}$ by a geodesic segment Γ_0 not crossed by Γ. Substituing the portion of $\partial\overline{\Gamma}$ between those points by Γ_0 we obtain a convex set $\mathcal{C}_0 \subset \overline{\Gamma}$ containing Γ, but then $\overline{\Gamma}$ would not be the convex hull of Γ. $\qquad\square$

If the boundary $\partial\mathcal{C}$ of a convex set is a polygonal curve, we can distinguish two type of nodes in it, those where the geodesic curvature κ_s is zero, and those with non-null geodesic curvature. From now on, we refer to the last ones as vertices. Doing this, we can look at $\partial\mathcal{C}$ as a polygon whose sides are geodesics.

---

9 Applications

In this section we study two important applications of the theory of convex hulls on triangulations presented in last sections. One of them is the establishment of the convex hull property of geodesic Bézier curves, which is very important for the adaptive implementation. The other result is the proof of convergence of the geodesic algorithm [21] given in a previos work. We present this result here due to its importance, even when it is not directly related to geodesic Bézier curves.

9.1 On the Metric of Discrete Surfaces

In section 9 we will need some metric properties of triangulated surfaces as distance and limit. In this section we study them.

The length functional $L(\Gamma) = \text{length}(\Gamma)$ can be used to define a distance in a connected triangulation:

Definition 4 The *distance* between two points P and Q on a connected discrete surface $\mathcal{S}$ is defined as the length of an optimal shortest arc joining P and Q.

$$\rho(P,Q) = \inf_{\Gamma_{PQ} \subset \mathcal{S}} \{L(\Gamma_{PQ})\}$$

It is known –see [23]– that if $\mathcal{S}$ is connected there always exists a shortest arc joining P and Q. Hence the distance ρ is well defined. The surface $\mathcal{S}$, provided with ρ, becomes a metric space [1,2]. A metric like this one, where the distance between two points is the same as the length of the shortest arcs joining them, is called *intrinsic*.

A metric space M is complete if every Cauchy sequence converges to a point of M. For a space provided with an intrinsic metric we can give an equivalent definition of completeness, see [1].

Definition 5 A metric of a space M is called *complete* if the Weierstrass theorem holds for M, i.e, if each infinite bounded set in M has an accumulation point.

The following lemma is rather obvious and is probably proved somewhere else. However, we have not seen it proved and will give a proof here.

Lemma 3 *A discrete surface $\mathcal{S}$, provided with its intrinsic metric, is a complete metric space.*

Proof From definition 5, it is enough to prove that every infinite set has an accumulation point.[1] Suppose that the set $A \in \mathcal{S}$ is infinite. Since $\mathcal{S}$ has a finite number of faces, there is a face $f \in \mathcal{S}$ which has infinite points of A. As in f the euclidean and intrinsic metric coincide, A has an accumulation point on f, and so it has an accumulation point on $\mathcal{S}$. $\qquad\square$

The following proposition will be useful in next section.

Proposition 7 *Let $\{\mathcal{C}_n\}_{n=0}^{\infty}$ be a sequence of closed sets in a discrete surface $\mathcal{S}$ such that $\mathcal{C}_{n+1} \subset \mathcal{C}_n$ and $\mathcal{C}_n \neq \emptyset$. There exists a closet set $\mathcal{C} \neq \emptyset$ such that $\mathcal{C} = \bigcap_{n=0}^{\infty} \mathcal{C}_n$ and for all $\varepsilon > 0$ there exists an $n \in \mathbb{N}$ such that if $N \geq n$ then $\rho(P,\mathcal{C}) < \varepsilon$ for all $P \in \mathcal{C}_N$.*

Proof The closed set $\mathcal{C}$ exists and is non-empty, see [18]. Lets prove the second part. Suppose that it does not hold, i.e, there is an $\varepsilon > 0$ such that $\forall n \in \mathbb{N}$ there exists a point $P_n \in \mathcal{C}_n$ such that $\rho(P_n, \mathcal{C}) \geq \varepsilon$. Consider the sequence $\{P_n\}$, from lemma 3 it has a subsequence $\{P_{k_n}\}$ convergent to a point P and $\rho(P,\mathcal{C}) \geq \varepsilon$. Given a $k \in \mathbb{N}$, $P_{k_n} \in \mathcal{C}_k$ for all $k_n \geq k$ and hence $P \in \mathcal{C}_k$, so $P \in \mathcal{C}$ and $\rho(P_n, \mathcal{C}) = 0$. But this is a contradiction with $\rho(P,\mathcal{C}) \geq \varepsilon$. $\qquad\square$

---

[1] Since we are considering the case of discrete surfaces, which are bounded, every infinite set on them is also bounded.

9.2 Results

The definition of convex hull of curves on triangulations help us to study the convergence of the geodesic algorithm presented by the authors in [21]; and also to stablish the convex hull property of geodesic Bézier curves.

In the rest of this section, $\{\Gamma_n\}_{n\in\mathbb{N}}$ denotes the sequence generated by the geodesic algorithm –see [21]. The following lemma will be used later:

Lemma 4 *In a triangulation $\mathcal{S}$, Γ_{n+1} belongs to the convex hull of Γ_n.*

Proof At each vertex correction, for example at P_{nj}, the sequence $P_{n,j-1}P_{nj}P_{n,j+1}$ goes to the geodesic segment joining the points $P_{n,j-1}$ and $P_{n,j+1}$, which is entirely contained in the convex hull of the current curve Γ_n. So, each vertex of Γ_{n+1} belongs to the convex hull of Γ_n and so does Γ_{n+1}. □

Proposition 8 *If every Γ_n is simple, then the sequence converges to a geodesic line.*

Proof Let $\mathcal{C}_n$ be the convex hull of Γ_n. Consider the set

$$\mathcal{C} = \bigcap_{n=0}^{\infty} \mathcal{C}_n.$$

From lemma 4 we know that $\mathcal{C}_n \bigcap \mathcal{C}_{n+1} = \mathcal{C}_{n+1}$. From proposition 7 we know that $\mathcal{C}$ exists and is closed. Each $\mathcal{C}_n$ is convex and so does $\mathcal{C}$.

We must prove that $\mathcal{A}(\mathcal{C}) = 0$ since, by proposition 6, only simple geodesic lines have convex hull with area equal to zero.
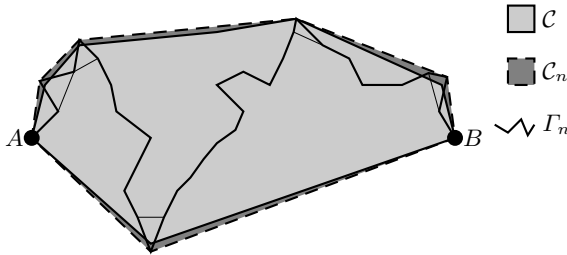

Fig. 12 The convex hull of Γ_n is very close to C.

By proposition 7, for an arbitrary $\varepsilon > 0$ we can find a natural number n such that all the vertices in the boundary $\partial\mathcal{C}_n$ of $\mathcal{C}_n$ are at distance smaller than ε from the boundary $\partial\mathcal{C}$ of $\mathcal{C}$.

Suppose that $\mathcal{A}(\mathcal{C}) \neq 0$. Each vertex of $\partial\mathcal{C}_n$ is also a node of Γ_n, see lemma 2. There is an n large enough such that the vertices of $\partial\mathcal{C}_n$ are very close to $\partial\mathcal{C}$ and such that for each node $P_{nj} \in \Gamma_n \cap \partial\mathcal{C}_n$ the geodesic line joining its neighbors $P_{n,j-1}$ and $P_{n,j+1}$ in Γ_n intersects $\mathcal{C}$. When we compute the next curve Γ_{n+1}, it will pass through $\mathcal{C}$, and as P_{in} belongs to the boundary of the convex hull of Γ_n, there will be points of $\mathcal{C}$ outside of $\mathcal{C}_{n+1}$, but this is impossible since $\forall n \ \mathcal{C} \subset \mathcal{C}_n \ \Rightarrow \mathcal{C} \subset \mathcal{C}_{n+1}$. As a consequence $\mathcal{A}(\mathcal{C})$ must be zero and hence $\Gamma_n \xrightarrow[n\to\infty]{} \Gamma$ which is a geodesic. □

Proposition 8 looks very restrictive, since we require that every curve Γ_n be simple. However, this seems to be the general case. When we look for an initial approximation, using FMM, the resulting curve Γ_0 is certainly simple. After that, each correction is performed in a neighborhood of a curve node and it us unlikely that self-intersection will occur.

To finish this section, we present a very important application of convex hulls: the convex hull property of geodesic Bézier curves. The following proposition is responsable for the correctness of the adavtive version of geodesic Bézier curves.

Proposition 9 *Geodesic Bézier curves satisfy the convex hull property. This is, a geodesic Bézier curve is completely inside the convex hull of its control polygon.*

Proof This proposition is a direct consequence of lemma 4. □

10 Convex Sets with Non-Polygonal Boundaries

Up to now, all the material in this section was mainly dedicated to polygonal curves. In fact, as far as this work is concerned, polygonal curves are enough. Nevertheless, we can extend our study about convexity to sets with non-polygonal boundaries. We can also extend it to smooth manifolds. For the sake of completeness, we consider them here.

Consider a piecewise smooth curve Γ on a triangulation $\mathcal{S}^2$. At points where the curve is smooth, it has neighborhoods which are isometric to a piece of plane, so we define the geodesic curvature κ_s of Γ at those points as the corresponding curvature in the unfolding. At non-smooth points, we define κ_s as the curvature corresponding to the polygonal line defined by the left and right tangent on these points, when it exists. Otherwise κ_s is not defined.

On smooth manifolds, the geodesic curvature is always defined for smooth curves. Consider a piece-wise smooth curve γ, which is simple and closed. In points where γ is not smooth, we can look at the corresponding left and right tangent lines in the plane tangent to the manifold. If the angle[3] between them is smaller than π, and the geodesic curvature at smooth points does not change its sign, then the set bounded by γ is convex.

---

[2] Note that polygonal curves are special cases of piecewise smooth curves.

[3] This angle is measured in the same direction of the curve.

11 Conclusion

We have defined geodesic Bézier curves, which are a generalization of Euclidean Bézier curves to manifold triangulations, and studied some properties of them. They have the advantage of being defined geodesically, which makes them independent of any parameterization. We have shown how to use them to define pieces or regions of a surface, allowing trimming, local texture mapping, and region coloring. Fast user interaction joined with the possibility of constructing C^0 and C^1 splines make of them a powerful tool for free-form modeling on manifold triangulations.

11.1 Further Research

There remain some theoretical issues associated with geodesic Bézier curves; it will be very interesting to see which other properties of classical Bézier curves hold for the new curves and also which concepts can be generalized to the geometry of manifold triangulations. For example, it is not clear how to define the control polygons if we want C^2 continuity or higher. The continuity of the curves at mesh vertices has to be studied. Is there something equivalent to affine invariance of Bézier curves in the case of geodesic Bézier curves?

There are some works about geodesic computation in geometries other than manifold triangulations. So, we can define geodesic Bézier curves for point clouds [22], for Riemannian manifolds [16], and for smooth surfaces [14,13]. The next step is to study how to handle user interaction in a fast way and what properties of classical Bézier curves are inherited by geodesic Bézier curves on those geometries. A good point to start could be subdivision surfaces where the extension of the ideas in this paper seems to be straightforward, with the nice property that user interaction could be handled at low resolution, making it faster.

12 Acknowledgement

References

1. Aleksandrov, A.D.: A. D. Aleksandrov selected works. Intrinsic Geometry of Convex Surfaces. Chapman & Hall/CRC (2006)

2. Aleksandrov, A.D., Zalgaller, V.A.: Intrinsic Geometry of Surfaces, *Translation of Mathematical Monographs*, vol. 15. AMS (1967)

3. Biermann, H., Martin, I.M., Zorin, D., Bernardini, F.: Sharp features on multiresolution subdivision surfaces. Graph. Models **64**(2), 61–77 (2002). DOI http://dx.doi.org/10.1006/gmod.2002.0570

4. de Casteljau, P.: Outillage Méthodes Calcul. Internes Dokument P2108, SA André Citroën, Paris (1959)

5. Chen, J., Han, Y.: Shortest paths on a polyhedron. In: Proceedings of 6th Annu. ACM Sympos. Comput. Geom, pp. 360–369 (1990)

6. Coelho, L.C.G., Gattass, M., de Figueiredo, L.H.: Intersecting and trimming parametric meshes on finite-element shells. International Journal for Numerical Methods in Engineering **47**(4), 777–800 (2000)

7. Cohen, E., Riesenfeld, R.F., Elber, G.: Geometric Modeling with Splines: An introduction. A K Peters, Ltd., 63 South Avenue, Natick, MA 01760 (2001)

8. Crouch, P., Kun, G., Leite, F.S.: The de Casteljau algorithm on Lie groups and spheres. Journal of Dynamical and Control Systems **5**(3), 397–429 (1999)

9. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall (1976)

10. Farin, G.: Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann Publishers Inc., San Francisco (2002)

11. Hofer, M., Pottmann, H.: Energy-minimizing splines in manifolds. ACM Trans. Graph. **23**(3), 284–293 (2004). DOI http://doi.acm.org/10.1145/1015706.1015716

12. Kapoor, S.: Efficient computation of geodesic shortest paths. In: Proceedings of 31st Annu. ACM Sympos. Theory Comput., pp. 770–779 (1999)

13. Kasap, E., Yapici, M., Akyildiz, F.T.: A numerical study for computation of geodesic curves. Applied Mathematics and Computation **171**(2), 1206–1213 (2005)

14. Kimmel, R., Sapiro, G.: Shortening three-dimensional curves via two-dimensional flows. Computers and Mathematics with Applications **29**(3), 49–62 (1995)

15. Kimmel, R., Sethian, J.: Computing geodesic paths on manifolds. *In* Proceedings of the National Academy of Sciences of the USA **95**(15), 8431–8435 (1998). URL http://citeseer.nj.nec.com/kimmel98computing.html

16. Klassen, E., Srivastava, A., Mio, W., Joshi, S.: Analysis of planar shapes using geodesic paths on shape manifolds. IEEE Trans. Pattern Analysis and Machine Intelligence **26**(3), 372–384 (2004)

17. Krishnan, S., Manocha, D.: An efficient surface intersection algorithm based on lower dimensional formulation. ACM Trans. on Computer Graphics **16**(1), 74–106 (1997)

18. Lages, E.: Curso de análise, vol. 2, 5 edn. Projeto Euclides (1999)

19. Litke, N., Levin, A., Schröder, P.: Trimming for subdivision surfaces. Computer Aided Geometric Design **18**(5), 463–481 (2001)

20. Martínez, D.: Geodesic-based modeling on manifold triangulations. Ph.D. thesis, IMPA, Rio de Janeiro, Brazil (2006)

21. Martínez, D., Velho, L., Carvalho, P.C.: Computing geodesics on triangular meshes. Computer and Graphics **29**(5), 667–675 (2005)

22. Mémoli, F., Sapiro, G.: Distance functions and geodesics on submanifolds of $\mathbb{R}^d$ and point clouds. SIAM Journal on Applied Mathematics **65**(4), 1227–1260 (2005)

23. Mitchell, J.S.B., Mount, D.M., Papadimitriou, C.H.: The discrete geodesic problem. SIAM J. COMPUT. **16**, 647–668 (1987)

24. Park, F.C., Ravani, B.: Bezier curves on Riemannian manifolds and Lie groups with kinematic applications. ASME Journal of Mechanical Design **117**, 36–40 (1995)

25. Polthier, K., Schmies, M.: Straightest geodesics on poly-
    hedral surfaces. In: H.C. Hege, K. Polthier (eds.) Visual-
    ization and Mathematics, pp. 135–150. Springer Verlag,
    Heidelberg (1998)
26. Pottmann, H., Hofer, M.: A variational aproach to spline
    curves on surfaces. Computer Aided Geometric Design
    22(7), 693–709 (2005)
27. Rodriguez, R.C., Leite, F.S., Jacubiak, J.: A new geomet-
    ric algorithm to generate smooth interpolating curves on
    riemannian manifolds. LMS Journal of Computation and
    Mathematics **8**, 251–266 (2005)
28. Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S.,
    Hoppe, H.: Fast exact and approximate geodesics on
    meshes. In: Proceedings of ACM SIGGRAPH 2005, pp.
    553–560 (2005)
29. Wallner, J.: Smoothness analysis of subdivision schemes
    by proximity. Constr. Approx. **24**(3), 289–318 (2006)
30. Wallner, J., Dyn, N.: Convergence and C^1 analysis of
    subdivision schemes on manifolds by proximity. Comput.
    Aided Geom. Design **22**(7), 593–622 (2005)
31. Wallner, J., Pottmann., H.: Intrinsic subdivision with
    smooth limits for graphics and animation. ACM Trans.
    Graphics **25**(2), 356–374 (2006)