Università di Milano–Bicocca
Quaderni di Matematica

Convexity preserving interpolatory subdivision with conic precision

Gudrun Albrecht and Lucia Romani

Quaderno n. 8/2010 (arXiv:1004.1295)

Stampato nel mese di aprile 2010
presso il Dipartimento di Matematica e Applicazioni,
Università degli Studi di Milano-Bicocca, via R. Cozzi 53, 20125 Milano, ITALIA.

Disponibile in formato elettronico sul sito www.matapp.unimib.it.
Segreteria di redazione: Ada Osmetti - Giuseppina Cogliandro
tel.: +39 02 6448 5755-5758 fax: +39 02 6448 5705

Esemplare fuori commercio per il deposito legale agli effetti della Legge 15 aprile 2004 n.106.

CONVEXITY PRESERVING INTERPOLATORY SUBDIVISION WITH CONIC PRECISION

Gudrun Albrecht
Univ. Lille Nord de France, UVHC, LAMAV-CGAO
FR no. 2956, F-59313 Valenciennes, France
gudrun.albrecht@univ-valenciennes.fr

Lucia Romani
Univ. of Milano-Bicocca, Dept. of Mathematics and Applications
Via R. Cozzi 53, 20125 Milano, Italy
lucia.romani@unimib.it

Abstract

The paper is concerned with the problem of shape preserving interpolatory subdivision. For arbitrarily spaced, planar input data an efficient non-linear subdivision algorithm is presented that results in G^1 limit curves, reproduces conic sections and respects the convexity properties of the initial data. Significant numerical examples illustrate the effectiveness of the proposed method.

Keywords: Subdivision, interpolation, convexity preservation, conics reproduction

2010 Mathematics Subject Classification: 41A05, 65D05, 65D17, 51N15

1 Introduction and state of the art

Subdivision schemes constitute a powerful alternative for the design of curves and surfaces over the widely studied parametric and implicit forms. In fact, they offer a really versatile tool that is, at the same time, very intuitive and easy to use and implement. This is due to the fact that subdivision schemes are defined via iterative algorithms which exploit simple refinement rules to generate denser and denser point sequences that, under appropriate hypotheses, converge to a continuous, and potentially smooth, function.

In the univariate case, the iteration starts with a sequence of points denoted by $\mathbf{p}^0 = (\mathbf{p}_i^0 \ : \ i \in \mathbb{Z})$, attached to the integer grid, and then for any $k \geq 0$ one subsequently computes a sequence $\mathbf{p}^{k+1} = S\mathbf{p}^k$, where $S : \ell(\mathbb{Z}) \to \ell(\mathbb{Z})$ identifies the so-called subdivision operator.

Subdivision operators can be broadly classified into two main categories: *interpolating* and *approximating* [12, 26]. Interpolating schemes are required to generate limit curves passing through all the vertices of the given polyline $\mathbf{p}^0$. Thus they are featured by refinement rules maintaining the points generated at each step of the recursion in all the successive iterations. Approximating schemes, instead, are not required to match the original position of vertices on the assigned polyline and

Date: January 26, 2010

thus they adjust their positions aiming at very smooth and visually pleasing limit shapes. As a consequence, while in the case of *approximating* subdivision the refinement rules rely on a recursive corner cutting process applied to the starting polygon $\mathbf{p}^0$, in the case of *interpolatory* subdivision, in every iteration a finer data set $\mathbf{p}^{k+1}$ is obtained by taking the old data values $\mathbf{p}^k$ and inserting new points in between them. Every such new point is calculated using a finite number of existing, usually neighboring points. In particular, if the computation of the new points is carried out through a linear combination of the existing points, the scheme is said to be *linear*, otherwise *non-linear*.

Then, inside the above identified categories, the schemes can also be further classified. More specifically, they can be distinguished between *stationary* (when the refinement rules do not depend on the recursion level) and *non-stationary*; between *uniform* (when the refinement rules do not vary from point to point) and *non-uniform*; between *binary* (when the number of points is doubled at each iteration) and *N-ary*, namely of arity $N > 2$.

Most of the univariate subdivision schemes studied in the literature are binary, uniform, stationary and linear. These characteristics, in fact, ease to study the mathematical properties of the limit curve, but seriously limit the applications of the scheme. Exceptions from a binary, or a uniform, or a stationary, or a linear approach, have recently appeared (see for example [24] and references therein), but none of the proposed methods provides an interpolatory algorithm that can fulfill the list of *all* fundamental features considered essential in applications. These features can briefly be summarized as:

(i) generating a visually-pleasing limit curve which faithfully mimics the behaviour of the underlying polyline without creating unwanted oscillations;

(ii) preserving the shape, i.e., the convexity properties of the given data;

(iii) identifying geometric primitives like circles and more generally conic sections, the starting polyline had been sampled from, and reproducing them.

Requirement (i) derives from the fact that, despite interpolating schemes are considered very well-suited for handling practical models to meet industrial needs (due to their evident link with the initial configuration of points representing the object to be designed), compared to their approximating counterparts, they are more difficult to control and tend to produce bulges and unwanted folds when the initial data are not uniformly spaced. Recently this problem has been addressed by using non-uniform refinement rules [6] opportunely designed to take into account the irregular distribution of the data. But, although their established merit of providing visually pleasing results, there is no guarantee that such methods are convexity-preserving, i.e., that if a convex data set is given, a convex interpolating curve can be obtained. This is due to the fact that, such non-uniform schemes are linear and, as it is well-known [17], linear refinement operators that are C^1 cannot preserve convexity in general.

The property (ii) of convexity preservation is of great practical importance in modelling curves and surfaces tailored to industrial design (e.g. related to car,

aeroplane or ship modelling where convexity is imposed by technical and physical conditions as well as by aesthetic requirements). In fact, if shape information as convexity is not enforced, interpolatory curves, though smooth, may not be satisfactory as they may contain redundant wiggles and bumps rather than those suggested by the data points, i.e., they feature unacceptable visual artifacts. Preserving convexity, while a curve is interpolated to a given data set, is far from trivial. But much progress has been made in this field, evidence of which is given by the recent burgeoning literature. In most publications, the introduction of subdivision schemes fulfilling requirement (ii) has been achieved through the definition of non-linear refinement rules. In fact, despite linear subdivision schemes turn out to be simple to implement, easy to analyze and affine invariant, they have many difficulties to control the shape of the limit curve and avoid artifacts and undesired inflexions that usually occur when the starting polygon $\mathbf{p}^0$ is made of highly non-uniform edges. Non-linear schemes, instead, offer effective algorithms to be used in shape-preserving data interpolation [9, 11, 13, 17, 18, 19].

On the basis of the well–known, linear Dubuc-Deslauriers interpolatory 4-point scheme [10] for example, several non-linear analogues have been presented in order to solve at least one of the three above listed properties. On the one hand, non-linear modifications of the classical 4-point scheme have been introduced to reduce the oscillations that usually occur in the limit curve when applying the refinement algorithm to polylines with short and long adjacent edges. These have been presented in [8] and [14], and as concerns the case of convexity-preserving strategies (which are the ones capable of completely eliminating the artifacts arising during the subdivision process), we find the papers [15] and [19]. On the other hand, for the purpose of enriching the Dubuc-Deslauriers 4-point scheme with the property (iii) of geometric primitives preservation, a non-linear 4-point scheme reproducing circles and reducing curvature variation for data off the circle, has been defined [25]. With the same intent, another modification of the classical 4-point scheme in a non-linear fashion, had been given in [3].

With these papers, the theoretical investigation of non-linear interpolatory subdivision has only begun. A lot is still to be done, in particular as concerns the use of non-linear rules for reproducing salient curves other than circles, considered of fundamental importance in several applications. So far, it has been shown that non-linear updating formulas can be used in the definition of non-stationary subdivision schemes aimed at reproducing polynomials and some common transcendental functions. In particular, [23] respectively [4, 5, 22] present subdivision algorithms that turn out to be circle-preserving respectively able to exactly represent any conic section. While the first is able to guarantee reproduction starting from given samples with any arbitrary spacing, for the latter ones the property of conics precision is confined to the case of equally-spaced samples. Most recently a shape and circle preserving scheme has been presented in [9].

Therefore, an outstanding issue that should be considered is the possibility of defining an interpolatory subdivision scheme that is at the same time shape-preserving and artifact free, as well as capable of generating conic sections starting from any arbitrarily-spaced samples coming from a conic. This is exactly the pur-

pose of this paper. Based on an approximation order four strategy presented in [1] for estimating tangents to planar convex data sequences, we propose a convexity-preserving interpolatory subdivision scheme with conic precision. This turns out to be a new kind of non-linear and geometry-driven subdivision method for curve interpolation. In Section 2 we start by describing the refinement strategy which relies on a classical cross-ratio property for conic sections and uses the tangent estimator from [1], for the case of globally convex data. In Section 3 we adapt the scheme to general, not necessarily convex data by segmenting the given polygon into convex segments and by presenting new refinement rules next to the junction points. In Section 4 we summarize the whole subdivision algorithm in all its steps. Then, in Section 5 we present an adaptive variant of the scheme which is aimed at producing regularly spaced points in every round of subdivision. Section 6 contains proofs for the scheme's shape preservation and conic reproducing properties, as well as for the G^1 continuity of its limit curve. Section 7 is devoted to illustrating the scheme by several significant application examples, and we conclude in Section 8.

2 Definition of the scheme for globally convex data

In this section we define a convexity preserving subdivision scheme for globally convex data which will then be the basis for the final shape preserving subdivision algorithm for general data.

Curve subdivision schemes iteratively apply a subdivision operator S to a starting point sequence $\mathbf{p}^0 = (\mathbf{p}_i^0 \ : \ i \in \mathbb{Z})$ yielding a new sequence $\mathbf{p}^{k+1} = S\mathbf{p}^k$ for any level $k \geq 0$. Our scheme being interpolatory, has refinement rules of the following form:

$$\begin{aligned} \mathbf{p}_{2i}^{k+1} &= \mathbf{p}_i^k, \\ \mathbf{p}_{2i+1}^{k+1} &= \varphi(\mathbf{p}_{i-\nu}^k, ..., \mathbf{p}_i^k, \mathbf{p}_{i+1}^k, ..., \mathbf{p}_{i+\nu}^k, \mathbf{p}_{i+\nu+1}^k;\ \mathbf{p}) \end{aligned} \tag{1}$$

where $\nu = 2$ is the number of points taken into account in the left and right hand neighborhoods of the segment $\mathbf{p}_i^k\mathbf{p}_{i+1}^k$ in order to define the newly inserted vertex $\mathbf{p}_{2i+1}^{k+1}$, and $\mathbf{p}$ is a parameter point specified later. φ is a non linear function, which we will define in form of an algorithm.

In order to detail the idea of this convexity–preserving scheme, we thus consider the following problem where, for simplicity, we omit the upper indices k.

Problem 1. *Given n points $\mathbf{p}_i((p_i)_x, (p_i)_y)$, $i = 1, \ldots, n$ ($n \geq 5$), in convex position in the affine plane, we wish to obtain one new point $\mathbf{u}_i$ related to the i-th edge $\mathbf{p}_i\,\mathbf{p}_{i+1}$.*

We will carry out the construction of the new points in the projectively extended affine plane. To this end we denote the projective counterparts of the affine points $\mathbf{p}_i((p_i)_x, (p_i)_y)$, $i = 1, \ldots, n$ by $P_i(p_{i,0}, p_{i,1}, p_{i,2})$ where $p_{i,0} = 1, p_{i,1} = (p_i)_x, p_{i,2} = (p_i)_y$.

By projective geometry's principle of duality, a line L may be represented either by a linear equation

$$l_0 x_0 + l_1 x_1 + l_2 x_2 = 0$$

in variable *point coordinates* (x_0, x_1, x_2) or by a triple (l_0, l_1, l_2) of constant *line coordinates*. The line coordinates of the line $L(l_0, l_1, l_2)$ joining two points $X_1(x_{1,0}, x_{1,1}, x_{1,2})$ and $X_2(x_{2,0}, x_{2,1}, x_{2,2})$ may simply be calculated by the vector product $X_1 \wedge X_2 = L$. In the same way, the point coordinates of the intersection point $P(p_0, p_1, p_2)$ of two lines $L_1(l_{1,0}, l_{1,1}, l_{1,2})$ and $L_2(l_{2,0}, l_{2,1}, l_{2,2})$ is obtained as $L_1 \wedge L_2 = P$. Without loss of generality we apply a normalization to the homogeneous point coordinates such that $x_0 \in \{0, 1\}$ for all calculated points.

In order to preserve global convexity of the points we apply the following preprocessing procedure to the point set. In every given point P_i we estimate a tangent from a subset of five points (including the point P_i) by the conic tangent estimator presented in [1]. If the given points represent a closed polygon, then the five-point subset is composed of the point P_i, its preceding two points P_{i-1}, P_{i-2} as well as its successive two points P_{i+1}, P_{i+2} (by considering $P_0 = P_n$, $P_{-1} = P_{n-1}$, $P_{n+1} = P_1$, $P_{n+2} = P_2$). If the given points represent an open polygon, then for $i = 3, \ldots, n-2$ we proceed as above and for $i \in \{1, 2\}$ (respectively $i \in \{n-1, n\}$) the points $\{P_1, P_2, P_3, P_4, P_5\}$ (respectively $\{P_{n-4}, P_{n-3}, P_{n-2}, P_{n-1}, P_n\}$) are taken.

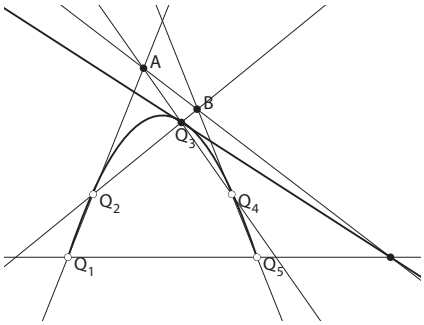


Figure 1: Illustration of the tangent construction.

In order to apply the conic tangent estimator from [1] we locally rename the five points around P_i by $Q_3 = P_i$, and by arbitrarily mapping the remaining four points to Q_1, Q_2, Q_4, Q_5 by a one–to–one map. The desired tangent in the point Q_3 is then calculated by the formula (see [1])

$$M_{33} := Q_3 \wedge (M_{15} \wedge (A \wedge B)) \,, \tag{2}$$

where $M_{ij} = Q_i \wedge Q_j$ for $i \neq j$, $A = M_{12} \wedge M_{34}$, and $B = M_{54} \wedge M_{32}$. See Figure 1 for an illustration.
We then denote the obtained line in the point P_i by L_i and intersect every two consecutive lines generating the intersection points

$$T_i = L_i \wedge L_{i+1} \,,$$

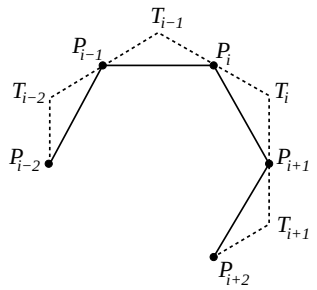


Figure 2: Convex delimiting polygon (dashed line) for a given convex data set (solid line).

see Figure 2.

The dashed lines in Figure 2 constitute a convex delimiting polygon for the new points generated in the next subdivision level. If the initial points come from a conic section, the constructed lines L_i are the tangents to this conic in the respective points P_i. Otherwise the lines L_i approximate the tangents with approximation order 4, see [1].

After this preprocessing step we now get back to the initial subdivision problem 1, i.e., between every two points P_i and P_{i+1} insert a new point U_i by applying a classical result from projective geometry which, for the readers convenience, we recall in the following theorem (see e.g., [2, 7, 21]).

Theorem 2. *a) Let X, E, E_0, E_1 be four points of a projective line $\mathcal{P}_1$, where the points E, E_0, E_1 are mutually distinct, and let (x_0, x_1) be the projective coordinates of the point X with respect to the projective coordinate system $\{E_0, E_1; E\}$ of $\mathcal{P}_1$. Then, the cross ratio $cr(X, E, E_0, E_1)$ of the four points X, E, E_0, E_1 in this order is defined by*

$$cr(X, E, E_0, E_1) = \frac{x_1}{x_0}.$$

b) Let P_1, P_2 be two points on a conic section c, and t_1, t_2 the tangents of c in P_1, P_2 respectively, and let T be the intersection point of t_1 and t_2 ($T = t_1 \cap t_2$). Then, the point T and the line P_1P_2 are pole and polar with respect to the conic c. Let's further denote the intersection points of any line l_T through T with the conic c by P and U, and the intersection point of l_T and T's polar P_1P_2 by X ($l_T \cap c = \{P, U\}$, $l_T \cap P_1P_2 = X$). Then

$$cr(U, P, X, T) = -1,$$

and the four points (U, P, X, T) are said to be in harmonic position.

Theorem 2 gives us the means of constructing a point U from three known collinear points P, X, T such that the harmonic cross ratio condition for conic sections is satisfied. For an illustration see Figure 3.

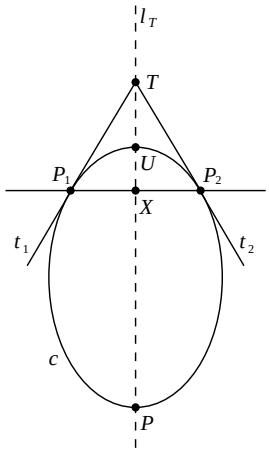


Figure 3: Harmonic cross ratio condition for conic sections.

For the choice of the parameter point P which is needed for the construction of the new point U_i inside the triangle $\Delta P_i T_i P_{i+1}$, the whole region bounded by the lines L_i, L_{i+1} and $P_i P_{i+1}$ containing the given convex polygon (see Figure 4) is suitable. In particular, every point P_{j_i} of the given convex polygon can be taken as parameter point (with exception of P_i and P_{i+1}), and since such a choice guarantees the reproduction of conic sections we opt for it.

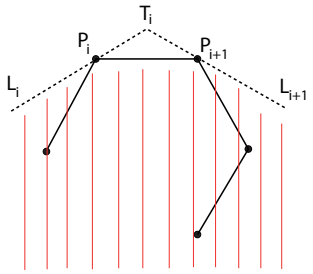


Figure 4: Suitable region for the choice of the parameter point P.

Let's thus denote by X_i the intersection point of the lines $P_i P_{i+1}$ and $P_{j_i} T_i$ for a chosen $j_i \in \{1, \ldots, n\} \setminus \{i, i+1\}$, i.e., $X_i = P_i P_{i+1} \cap P_{j_i} T_i$.

In order to guarantee a regular distribution of the inserted points we propose to choose the index j_i by the following angle criterion. To this end we temporarily come back to the Euclidean plane and introduce the midpoint $\mathbf{m}_i$ for each segment $\mathbf{p}_i \mathbf{p}_{i+1}$. Let $g_i = \mathbf{t}_i \mathbf{m}_i$, $h_{ij} = \mathbf{t}_i \mathbf{p}_j$ be the connecting lines of the points $\mathbf{t}_i$ and $\mathbf{m}_i$ respectively $\mathbf{t}_i$ and $\mathbf{p}_j$, and $\alpha_j^i = \angle(g_i, h_{ij})$ the angle between these two lines[1] for $j = i+2, \ldots, n+i-1$ by considering $\mathbf{p}_{n+r} = \mathbf{p}_r$ for $r \geq 1$. For every $i \in \{1, \ldots, n\}$ in the case of a closed polygon and for every $i \in \{1, \ldots, n-1\}$ in the case of an open polygon, we then obtain a value j_i from the condition

$$\alpha_{j_i}^i = \min_{j=i+2,\ldots,n+i-1} \alpha_j^i. \tag{3}$$

[1]The smaller one of the two complementary angles is taken in each case.

Once the point P_{j_i} has been selected by exploiting the illustrated criterion, we then establish the projective coordinate system $\{X_i, T_i; P_{j_i}\}$ on the straight line $P_{j_i}T_i$ by calculating the projective representatives of X_i and T_i by solving

$$\gamma_i X_i + \mu_i T_i = P_{j_i}$$

for γ_i and μ_i. We obtain

$$\gamma_i = D_{i,1}/D_i\,, \quad \mu_i = D_{i,2}/D_i\,,$$

where

$$D_i = \det \begin{pmatrix} x_{i,l} & t_{i,l} \\ x_{i,m} & t_{i,m} \end{pmatrix}\,,\; D_{i,1} = \det \begin{pmatrix} p_{j_i,l} & t_{i,l} \\ p_{j_i,m} & t_{i,m} \end{pmatrix}\,,\; D_{i,2} = \det \begin{pmatrix} x_{i,l} & p_{j_i,l} \\ x_{i,m} & p_{j_i,m} \end{pmatrix}\,, \tag{4}$$

$l \neq m \in \{0, 1, 2\}$.
By Theorem 2 the point U_i is thus obtained as

$$U_i = D_{i,1}X_i - D_{i,2}T_i.$$

3 Modification of the scheme for non-convex data

If the input data are not convex, we segment them according to the following criteria in order to obtain piecewise convex segments.
First, consecutive collinear points are identified in the following way.

> By applying a dominant points selection algorithm like, for instance, the one in [20], we can easily detect the end points of a sequence containing at least 3 collinear vertices. Let us denote them by $\mathbf{p}_j^0$ and $\mathbf{p}_l^0$. (5)

Since in CAD applications as we have in mind linear features are usually intentional, we do not smooth the angle between a subpolygon consisting of (at least 3) collinear points and its neighbors. The insertion rule for these straight line subpolygons between the points $\mathbf{p}_j^k$ and $\mathbf{p}_l^k$ $(k = 0, 1, 2, 3, \ldots)$ simply reads as:

$$\mathbf{p}_{2i+1}^{k+1} = \frac{1}{2}(\mathbf{p}_i^k + \mathbf{p}_{i+1}^k)\,,\; i = j, \ldots, l-1. \tag{6}$$

The remaining subpolygons do not anymore contain collinear segments. For these remaining subpolygons, inflection edges are identified by the following criterion, see also [9, 16].

> An edge $\mathbf{p}_i^0\mathbf{p}_{i+1}^0$ is identified as *inflection edge* if the points $\mathbf{p}_{i-1}^0$ and $\mathbf{p}_{i+2}^0$ lie in different half planes with respect to it. (7)

On an inflection edge we insert a new point, e.g., as midpoint of the edge corners, and we thus have a sequence of subpolygons without inflections. Next, we check

each of these subpolygons $\mathbf{p}_j^0 \dots \mathbf{p}_l^0$ for total convexity by the following criterion:

> If for every edge of the subpolygon all the points of the subpolygon lie either on the edge or in the same half plane with respect to the edge, then the subpolygon is totally convex, otherwise it is only locally convex. (8)

If a subpolygon is only locally convex we divide it into two new subpolygons $\mathbf{p}_j^0 \dots \mathbf{p}_i^0$ and $\mathbf{p}_i^0 \dots \mathbf{p}_l^0$, where $i = j + \llcorner \frac{l-j+1}{2} \lrcorner$. We repeat this procedure until we have a sequence of totally convex subpolygons, each of which we suppose to be composed of at least five points.

Whereas in the interior of every convex subpolygon we apply the algorithm detailed in the previous section, we now define the method next to

1. an inserted junction point on an inflection edge, which we refer to as *inflection point*,

2. a junction point between two convex subpolygons, which we refer to as *convex junction point*.

In the case of an *inflection point* $\mathbf{p}_i^0$ let us denote the line of the inflection edge $\mathbf{p}_{i-1}^0\mathbf{p}_i^0\mathbf{p}_{i+1}^0$ by $\mathbf{e}_i$, and the convex subpolygons meeting in $\mathbf{p}_i^0$ by $\mathbf{s}_{l,i}^0$ and $\mathbf{s}_{r,i}^0$. We estimate a left and a right tangent in $\mathbf{p}_i^0$, $\mathbf{l}_{l,i}^0$ and $\mathbf{l}_{r,i}^0$ respectively, by applying the tangent estimation method from [1] to the five points $\mathbf{p}_{i-4}^0, \dots, \mathbf{p}_{i-1}^0, \mathbf{p}_i^0$ of polygon $\mathbf{s}_{l,i}^0$, respectively $\mathbf{p}_i^0, \dots, \mathbf{p}_{i+3}^0, \mathbf{p}_{i+4}^0$ of polygon $\mathbf{s}_{r,i}^0$.

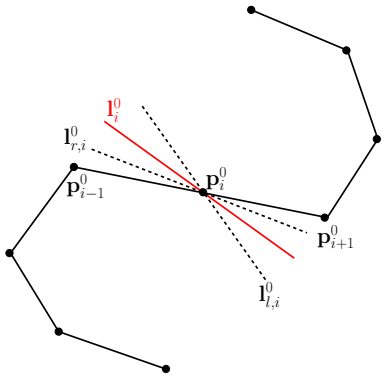


Figure 5: Definition of an initial tangent $\mathbf{l}_i^0$ in an inflection point $\mathbf{p}_i^0$.

We then combine these two lines $\mathbf{l}_{l,i}^0$ and $\mathbf{l}_{r,i}^0$ for defining an initial tangent $\mathbf{l}_i^0$ in $\mathbf{p}_i^0$ (see Figure 5):

$$\mathbf{l}_i^0 = \lambda_i^0 \mathbf{l}_{l,i}^0 + \mu_i^0 \mathbf{l}_{r,i}^0 \,, \tag{9}$$

where $\lambda_i^0 + \mu_i^0 = 1$, $\lambda_i^0, \mu_i^0 > 0$. The pair (λ_i^0, μ_i^0) thus plays the role of a shape parameter. The tangents in the other vertices of the polygons $\mathbf{s}_{l,i}^0$ and $\mathbf{s}_{r,i}^0$, and thus their new vertices, are calculated as described in the previous section by treating $\mathbf{s}_{l,i}^0$ and $\mathbf{s}_{r,i}^0$ separately. We obtain the new polygons $\mathbf{s}_{l,i}^1$ and $\mathbf{s}_{r,i}^1$. Let us now describe

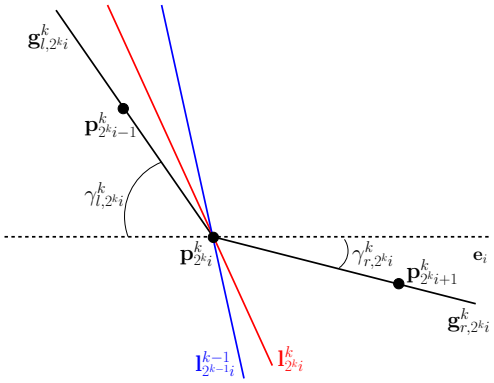


Figure 6: Tangent definition in an inflection point.

how to obtain the tangents $\mathbf{l}^k_{2^k i}$ in the point $\mathbf{p}^0_i = \mathbf{p}^k_{2^k i}$ in the following iterations ($k = 1, 2, 3, \ldots$), see Figure 6 for an illustration.

Let

$$\mathbf{g}^k_{l,2^k i} = \mathbf{p}^k_{2^k i-1}\mathbf{p}^k_{2^k i} \text{ and } \mathbf{g}^k_{r,2^k i} = \mathbf{p}^k_{2^k i}\mathbf{p}^k_{2^k i+1} \tag{10}$$

be the edges that are incident in $\mathbf{p}^0_i = \mathbf{p}^k_{2^k i}$, and

$$\gamma^k_{r,2^k i} = \angle(\mathbf{g}^k_{r,2^k i}, \mathbf{e}_i) \text{ and } \gamma^k_{l,2^k i} = \angle(\mathbf{g}^k_{l,2^k i}, \mathbf{e}_i) \tag{11}$$

their respective angles with the initial inflection edge $\mathbf{e}_i$. We then define the line $\mathbf{g}^k_{2^k i}$ by choosing that one of the lines $\mathbf{g}^k_{l,2^k i}$ and $\mathbf{g}^k_{r,2^k i}$ from (10) yielding the maximum angle

$$\gamma^k_{2^k i} = \max\{\gamma^k_{l,2^k i}, \gamma^k_{r,2^k i}\}\,. \tag{12}$$

The tangent $\mathbf{l}^k_{2^k i}$ in the point $\mathbf{p}^0_i = \mathbf{p}^k_{2^k i}$ is then defined as

$$\mathbf{l}^k_{2^k i} = \lambda^k_{2^k i}\mathbf{l}^{k-1}_{2^{k-1} i} + \mu^k_{2^k i}\mathbf{g}^k_{2^k i}\,, \tag{13}$$

where $\lambda^k_{2^k i} + \mu^k_{2^k i} = 1$ and $\lambda^k_{2^k i}, \mu^k_{2^k i} > 0$. In the other vertices of $\mathbf{s}^k_{l,i}$ and $\mathbf{s}^k_{r,i}$ we proceed as in section 2 for estimating the tangents; this allows us to calculate the new polygons $\mathbf{s}^{k+1}_{l,i}$ and $\mathbf{s}^{k+1}_{r,i}$ by separately applying the "convex" procedure from the previous section.

In the case of a *convex junction point* $\mathbf{p}^0_i$ we suppose our data to be such that the following condition holds[2] (see Figure 7):

$$\begin{aligned}&\text{the intersection points } \mathbf{p}^0_{i-2}\mathbf{p}^0_{i-1} \cap \mathbf{p}^0_i\mathbf{p}^0_{i+1} \text{ and } \mathbf{p}^0_{i-1}\mathbf{p}^0_i \cap \\ &\mathbf{p}^0_{i+1}\mathbf{p}^0_{i+2} \text{ lie in the same half plane with respect to the} \\ &\text{line } \mathbf{p}^0_{i-1}\mathbf{p}^0_{i+1} \text{ as the point } \mathbf{p}^0_i.\end{aligned} \tag{14}$$

Let us denote the convex subpolygons meeting in $\mathbf{p}^0_i$ by $\mathbf{s}^0_{l,i}$ and $\mathbf{s}^0_{r,i}$. As in the case of an inflection point we estimate a left and a right tangent in $\mathbf{p}^0_i$, $\mathbf{l}^0_{l,i}$ and

[2]This condition is guaranteed by a sufficiently dense sampling of the initial data points.

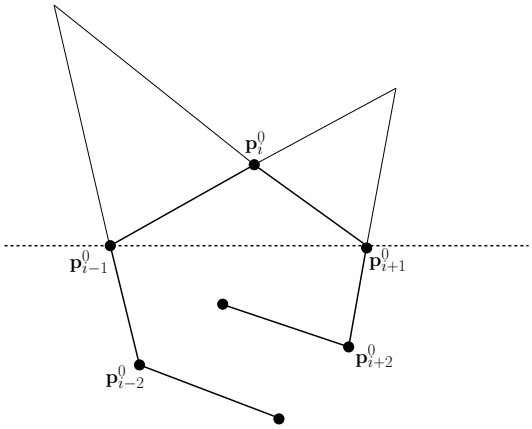


Figure 7: Situation in a convex junction point.

$\mathbf{l}_{r,i}^0$ respectively, by applying the tangent estimation method from [1] to the five points $\mathbf{p}_{i-4}^0, \ldots, \mathbf{p}_{i-1}^0, \mathbf{p}_i^0$ of polygon $\mathbf{s}_{l,i}^0$, respectively $\mathbf{p}_i^0, \ldots, \mathbf{p}_{i+3}^0, \mathbf{p}_{i+4}^0$ of polygon $\mathbf{s}_{r,i}^0$. If the points $\mathbf{p}_{i-1}^0$ and $\mathbf{p}_{i+1}^0$ lie in different half planes with respect to $\mathbf{l}_{l,i}^0$ ($\mathbf{l}_{r,i}^0$ respectively) we replace $\mathbf{l}_{l,i}^0$ ($\mathbf{l}_{r,i}^0$ respectively) by the line $\mathbf{p}_i^0\mathbf{p}_{i+1}^0$ ($\mathbf{p}_{i-1}^0\mathbf{p}_i^0$ respectively). We then combine these two lines $\mathbf{l}_{l,i}^0$ and $\mathbf{l}_{r,i}^0$ as in (9) for defining an initial tangent $\mathbf{l}_i^0$ in $\mathbf{p}_i^0$. The tangents in the other vertices of the polygons $\mathbf{s}_{l,i}^0$ and $\mathbf{s}_{r,i}^0$, and thus their new vertices, are calculated as described in the previous section by treating $\mathbf{s}_{l,i}^0$ and $\mathbf{s}_{r,i}^0$ separately. We obtain the new polygons $\mathbf{s}_{l,i}^1$ and $\mathbf{s}_{r,i}^1$. We then iterate this procedure and obtain the tangents $\mathbf{l}_{2^k i}^k$ in the point $\mathbf{p}_i^0 = \mathbf{p}_{2^k i}^k$ in the following iterations ($k = 1, 2, 3, \ldots$) as

$$\mathbf{l}_{2^k i}^k = \lambda_{2^k i}^k \mathbf{l}_{l,2^k i}^k + \mu_{2^k i}^k \mathbf{l}_{r,2^k i}^k \,, \tag{15}$$

where $\mathbf{l}_{l,2^k i}^k$ and $\mathbf{l}_{r,2^k i}^k$ is the respective left and right tangent in $\mathbf{p}_{2^k i}^k$ and $\lambda_{2^k i}^k + \mu_{2^k i}^k = 1$, $\lambda_{2^k i}^k, \mu_{2^k i}^k > 0$ (see Figure 8 for an illustration).

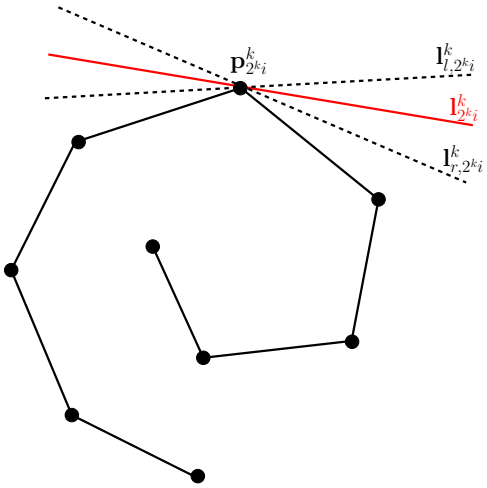


Figure 8: Tangent definition in a convex junction point.

In order to avoid a rather slow convergence of the newly inserted points towards the junction points (inflection points or convex junction points), see Figure 10, we modify the point insertion rule for the points $\mathbf{p}^{k+1}_{2^{k+1}i-1}$ and $\mathbf{p}^{k+1}_{2^{k+1}i+1}$ in the tangent triangles adjacent to the junction point $\mathbf{p}^k_{2^k i}$ formed by the lines $\mathbf{l}^k_{2^k i-1}$, $\mathbf{l}^k_{2^k i}$ and the edge $\mathbf{p}^k_{2^k i-1}\mathbf{p}^k_{2^k i}$, and $\mathbf{l}^k_{2^k i}$, $\mathbf{l}^k_{2^k i+1}$ and the edge $\mathbf{p}^k_{2^k i}\mathbf{p}^k_{2^k i+1}$ respectively. The modified rule for the first and the last new point of a subpolygon reads as follows

$$\mathbf{p} = \rho\mathbf{t} + \sigma\mathbf{m}\,, \quad \rho+\sigma = 1\,, \quad \rho,\sigma > 0\,, \tag{16}$$

where $\mathbf{p}$ stands for the new point $\mathbf{p}^{k+1}_{2^{k+1}i-1}$ respectively $\mathbf{p}^{k+1}_{2^{k+1}i+1}$, $\mathbf{t}$ designates the intersection point of the two tangents in the corner points of the corresponding edge and $\mathbf{m}$ is the mid-point of this edge, see Figure 9.

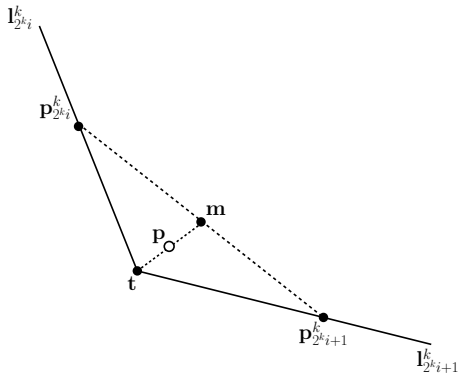
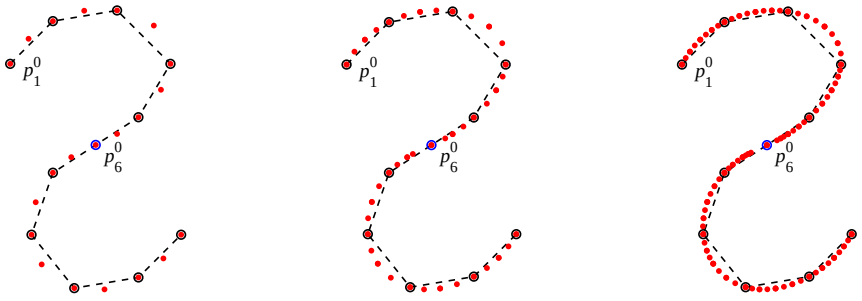


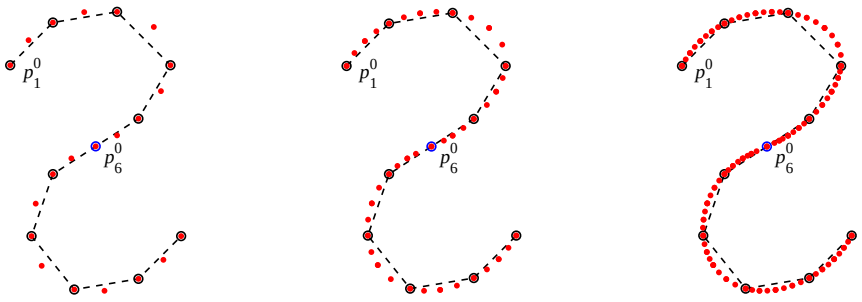
Figure 9: Modified point insertion rule next to a junction point.

This new "end point rule" avoids holes around the junction points (see Figure 11).



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 10: Example of slow convergence of the newly inserted points towards the inflection point p^0_6.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 11: Application example of the new "end point rule" in equation (16) to accelerate convergence of the newly inserted points towards the inflection point p_6^0.

4 The subdivision algorithm

The above detailed procedure may be summarized in the following algorithm, which we apply to the given point sequence $\mathbf{p}^0 = (\mathbf{p}_i^0 \ : \ i \in \mathbb{Z})$. For the sake of clarity we will first describe the procedure to be used in the case of totally convex data (Subsection 4.1), which constitutes an essential ingredient of the final algorithm for general non-convex data presented in Subsection 4.2.

4.1 Algorithm for totally-convex data

Let $\mathbf{p}^k = (\mathbf{p}_i^k \ : \ i = 1, \ldots, n_k)$ be the vertices of the totally convex polygon at the k-level refinement. Hereafter we denote by P_i^k the projective counterparts of the affine points $\mathbf{p}_i^k$, see section 2.
The algorithm that implements the function φ from (1) proceeds as follows in order to calculate the points $\mathbf{p}_{2i+1}^{k+1} = \varphi(\mathbf{p}_{i-2}^k, \mathbf{p}_{i-1}^k, \mathbf{p}_i^k, \mathbf{p}_{i+1}^k, \mathbf{p}_{i+2}^k, \mathbf{p}_{i+3}^k \,;\, \mathbf{p})$.

Step 1: Preprocessing

a) For a closed polygon we set $P_0^k = P_{n_k}^k$, $P_{-1}^k = P_{n_k-1}^k$, $P_{n_k+1}^k = P_1^k$, $P_{n_k+2}^k = P_2^k$.
For an open polygon we set $P_0^k = P_5^k$, $P_{-1}^k = P_4^k$, $P_{n_k+1}^k = P_{n_k-4}^k$, $P_{n_k+2}^k = P_{n_k-3}^k$.
For $i = 1, \ldots, n_k$ we then assign $Q_1 = P_{i-2}^k$, $Q_2 = P_{i-1}^k$, $Q_3 = P_i^k$, $Q_4 = P_{i+1}^k$, $Q_5 = P_{i+2}^k$, and apply formula (2) for obtaining the tangent L_i^k in P_i^k.

b) According to the angle criterion, we calculate the angles α_j^i for $j = i + 2, \ldots, n_k + i - 1$ and for $i = 1, \ldots, n_k$ in the case of a closed polygon, while for $i = 1, \ldots, n_k - 1$ in the case of an open polygon. We then select the value $\alpha_{j_i}^i$ satisfying condition (3).

Step 2: For a closed polygon we set $L^k_{n_k+1} = L^k_1$ and we consider $i = 1, \ldots, n_k$, whereas for an open polygon we consider $i = 1, \ldots, n_k - 1$. We thus calculate the intersection points

$$T^k_i = L^k_i \wedge L^k_{i+1}\,.$$

Step 3: Calculate the lines (for $i = 1, \ldots, n_k$ for a closed polygon, and for $i = 1, \ldots, n_k - 1$ for an open polygon)

$$N^k_i = P^k_i \wedge P^k_{i+1}\,,\ \Lambda^k_i = P^k_{j_i} \wedge T^k_i\,,$$

as well as their intersection points

$$X^k_i = N^k_i \wedge \Lambda^k_i.$$

Step 4: Calculate the points P^{k+1}_{2i+1} as

$$P^{k+1}_{2i+1} = D^k_{i,1} X^k_i - D^k_{i,2} T^k_i$$

with $D^k_{i,1}, D^k_{i,2}$ according to (4).

4.2 Algorithm for non-convex data

Step 1: Preprocessing

We preprocess the data according to the criteria (5), (7) and (8) thus identifying subpolygons contained in a straight line and introducing and /or identifying inflection vertices and convex junction vertices within the initial data points of the remaining subpolygons yielding a sequence composed of totally convex subpolygons and "straight line" subpolygons.

Step 2:

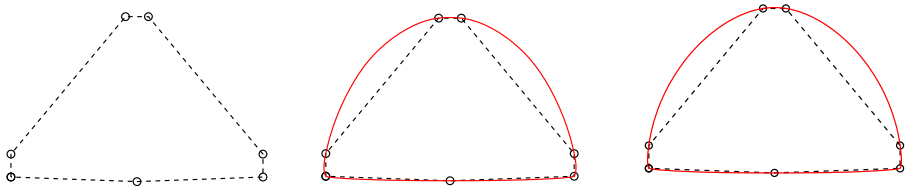
- For a "straight line" subpolygon consisting of collinear segments we fix the tangents at its end points equal to the straight line passing through them and we apply everywhere the insertion rule (6).
- For a totally convex subpolygon $\mathbf{p}^0_j \ldots \mathbf{p}^0_l = \ldots = \mathbf{p}^k_{2^k j} \ldots \mathbf{p}^k_{2^k l}$ between the junction points $\mathbf{p}^0_j$ and $\mathbf{p}^0_l$ (where $l \geq j + 4$) we apply the algorithm of Subsection 4.1 in order to calculate the tangents in the points $\mathbf{p}^k_{2^k j+1}, \ldots, \mathbf{p}^k_{2^k l-1}$ and the new points (with upper index $k + 1$) between $\mathbf{p}^k_{2^k j+1}$ and $\mathbf{p}^k_{2^k l-1}$. According to the type of the subpolygon end points $\mathbf{p}^k_{2^k j}$ and $\mathbf{p}^k_{2^k l}$, i.e., inflection point or convex junction point, we apply the rule (13) for an inflection point and (15) for a convex junction point in order to calculate the respective tangent, and apply rule (16) for calculating the first respectively last new point, i.e., $\mathbf{p}^{k+1}_{2^{k+1} j+1}$ respectively $\mathbf{p}^{k+1}_{2^{k+1} l-1}$. If $\mathbf{p}^k_{2^k j}$ or $\mathbf{p}^k_{2^k l}$ is a junction point with a straight line subpolygon, we fix the tangent at such an end point equal to the straight line passing through it

and we define the new point in the adjacent triangle using the standard rule. Finally, if $\mathbf{p}^k_{2^k j}$ or $\mathbf{p}^k_{2^k l}$ coincides with an end point of the whole sequence, we proceed by computing the tangent at that location and the new point closest to it exactly following the same procedure we described in the open polygon case of Subsection 4.1.

5 Adaptive version of the scheme

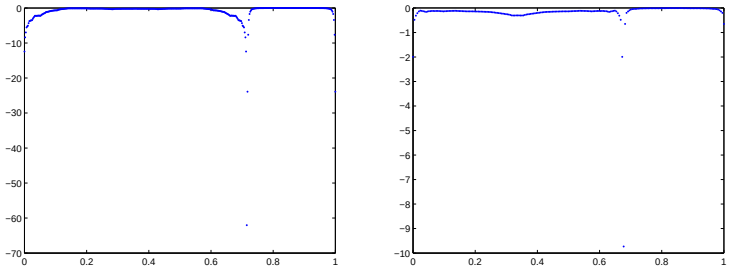
Subdivision curves are visualized by drawing a polyline on a level of refinement which evokes the impression of sufficient approximation of the given data. For a high quality rendering, the task is therefore to calculate and draw a level of subdivision which is a visually sufficient approximation of the limit shape. The subdivision algorithm presented in Section 4 provides a process of global refinement at every level. Therefore, when the starting polyline is highly non-uniform, the required level of refinement is determined by those locations which approximate the limit curve most unfavorably. Obviously, these may cause unnecessary fine subdivisions at other locations of the curve, thus leading to an unreasonable resource demanding algorithm.
To overcome this problem we propose an adaptive version of the subdivision algorithm previously described. Adaptive subdivision is achieved by applying the mechanism of subdivision only locally, i.e. only at those locations of the initial polyline that are not approximated with the desired quality. The decision where high resolution refinement is needed, strongly depends on the underlying application. As concerns our algorithm, adaptivity may be controlled either by the user or by an automatic criterion. In fact, the user may specify which portions of the polygon should be subdivided or the process may be automated by controlling whether the length of an edge is greater or not than a specified threshold. Only in the positive case we insert a new point in correspondence of the considered edge. Of course, besides improving the visual quality of the limit curve, the adaptive version of the scheme reduces the computational cost of the algorithm.
In the remainder of this section we take highly non-uniform polylines and compare our adaptive refinement algorithm with the basic one. Figure 12 shows the comparison between the refined polylines obtained by applying the two algorithms to a totally convex set of points, while Figure 14 compares the two algorithms on a highly non-uniform polyline with convex junction points and inflection points.

Figures 13 and 15 illustrate the corresponding discrete curvature plots. It can be easily seen that the use of the adaptive refinement scheme results in a considerable improvement of the curvature behaviour.



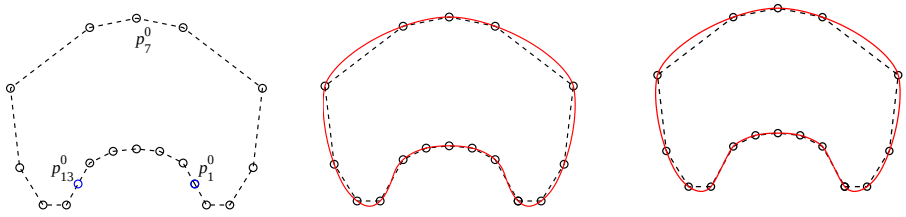
(a) (b) (c)

Figure 12: Application example of the adaptive algorithm: (a) starting polyline; (b) refined polyline after 6 steps of the basic algorithm; (c) refined polyline after 6 steps of the adaptive algorithm.



(a) (b)

Figure 13: Comparison between the discrete curvature plots of the refined polylines in Figure 12 (b) and 12 (c).

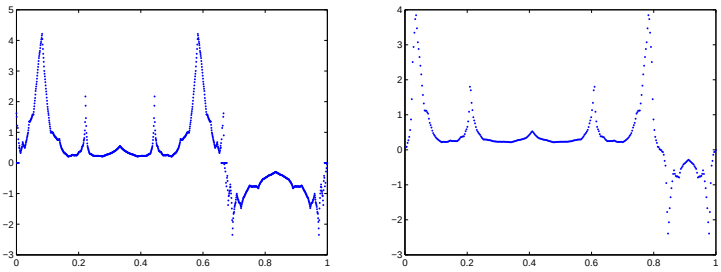


(a) (b) (c)

Figure 14: Application example of the adaptive algorithm: (a) starting polyline; (b) refined polyline after 6 steps of the basic algorithm; (c) refined polyline after 6 steps of the adaptive algorithm. The labels in Subfigure (a) denote the end points of consecutive subpolygons: while p_7^0 is a convex junction point, p_1^0 and p_{13}^0 are inflection points.

6 Properties of the scheme and smoothness analysis

By construction the presented scheme enjoys the properties of being shape preserving and conic reproducing as summarized in the following proposition.



(a) (b)

Figure 15: Comparison between the discrete curvature plots of the refined polylines in Figure 14 (b) and 14 (c).

Proposition 1. *The presented interpolatory curve subdivision scheme*

a) is shape preserving, i.e., if the starting point sequence $\mathbf{p}^0 = (\mathbf{p}_i^0 \ : \ i = 1, \ldots, n_0)$ *consists of convex, straight line and concave segments, then all generated subsequent point sequences* $\mathbf{p}^k = (\mathbf{p}_i^k \ : \ i = 1, \ldots, n_k)$*, for* $k = 1, 2, 3, \ldots$ *respect the same behavior;*

b) reproduces conic sections, i.e., if the starting point sequence $\mathbf{p}^0$ *is sampled from a conic section* c*, then the limit curve coincides with* c*.*

Proof. *a)* In the preprocessing step the initial point sequence is segmented into straight line segments and totally convex segments, which are not changed during the subdivision procedure. A straight line subpolygon is reproduced as such. For a totally convex subpolygon $\mathbf{p}^k = (\mathbf{p}_i^k \ : \ i = j_k, \ldots, l_k)$ the algorithm first generates a line $\mathbf{l}_i^k$ in every point $\mathbf{p}_i^k$. By construction (see Section 2 and (13), (15)) these lines form a convex delimiting polygon for the points of the sequence $\mathbf{p}^{k+1}$ of the next subdivision level, see Figure 2. The fact that by construction the point $\mathbf{p}_{2i+1}^{k+1}$ is contained in the triangle formed by the points $\mathbf{p}_i^k, \mathbf{t}_i^k, \mathbf{p}_{i+1}^k$ ($\mathbf{p}_{2i+1}^{k+1} \in \Delta(\mathbf{p}_i^k, \mathbf{t}_i^k, \mathbf{p}_{i+1}^k)$) guarantees convexity of the sequence $\mathbf{p}^{k+1}$.

b) If the point sequence $\mathbf{p}^k$ comes from a conic section c, then the lines $\mathbf{l}_i^k$ generated in the preprocessing step are the tangents of c in the points $\mathbf{p}_i^k$. By Theorem 2 the constructed points $\mathbf{p}_{2i+1}^{k+1}$ lie on c. □

The special set-up of the presented subdivision scheme allows to obtain a result on the smoothness of the limit curve as formulated in the following proposition for which we suppose our data not to contain any subpolygons that consist of consecutive collinear segments. The transitions from and to such subpolygons are purposely C^0 by construction, and these "straight line" subpolygons are exactly reproduced by the subdivision algorithm. We can thus restrict our attention to polygons not containing any straight line subpolygons.

Proposition 2. *a) For the presented subdivision scheme the polygon series* $\mathbf{p}^k$ *converges to a continuous curve.*

b) The limit curve of the presented subdivision scheme is of continuity class G^1*.*

Proof. The proof is formulated for the non–adaptive version of the subdivision algorithm where every polygon edge is replaced by two new edges in every step. For the sake of simplicity we omit the details for the adaptive case, where the indices change but not the general idea. We proceed by demonstrating the C^0 and G^1 continuity in two steps. First, we consider (locally) convex segments (including convex junction points), and then we treat inflection points. The locally convex segments are composed of several totally convex segments joined together at convex junction points.

We introduce the following notation. See Figure 16 for an illustration.

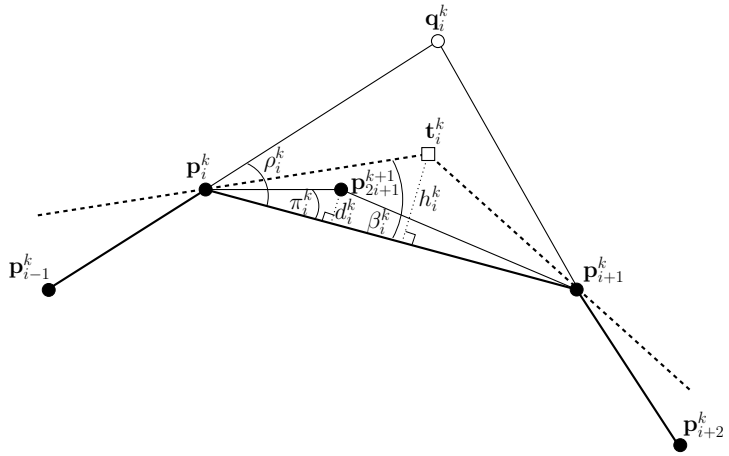


Figure 16: Notation used throughout the proof.

$\mathbf{q}_i^k$	intersection point of the edges $\mathbf{p}_{i-1}^k\mathbf{p}_i^k$ and $\mathbf{p}_{i+1}^k\mathbf{p}_{i+2}^k$
ρ_i^k	inner angle in $\mathbf{p}_i^k$ of the triangle $\Delta(\mathbf{p}_{i+1}^k\mathbf{p}_i^k\mathbf{q}_i^k)$
β_i^k	inner angle in $\mathbf{p}_i^k$ of the triangle $\Delta(\mathbf{p}_{i+1}^k\mathbf{p}_i^k\mathbf{t}_i^k)$
π_i^k	inner angle in $\mathbf{p}_i^k$ of the triangle $\Delta(\mathbf{p}_{i+1}^k\mathbf{p}_i^k\mathbf{p}_{2i+1}^{k+1})$
h_i^k	height of the triangle $\Delta(\mathbf{p}_{i+1}^k\mathbf{p}_i^k\mathbf{t}_i^k)$ from $\mathbf{t}_i^k$ onto the edge $\mathbf{p}_i^k\mathbf{p}_{i+1}^k$
d_i^k	height of the triangle $\Delta(\mathbf{p}_{i+1}^k\mathbf{p}_i^k\mathbf{p}_{2i+1}^{k+1})$ from $\mathbf{p}_{2i+1}^{k+1}$ onto the edge $\mathbf{p}_i^k\mathbf{p}_{i+1}^k$
c_i^k	length of the edge $\mathbf{p}_i^k\mathbf{p}_{i+1}^k$
l_i^k	length of the line segment $\mathbf{p}_i^k\mathbf{t}_i^k$

C^0 continuity for (locally) convex segments:

In order to show the continuity of the limit curve we compute the distance d^k between the polygon $\mathbf{p}^{k+1}$ and the polygon $\mathbf{p}^k$. According to the above notation we define

$d^k = \max_i\{d_i^k\}$. By construction[3] the point $\mathbf{t}_i^k$ lies inside the triangle $\Delta(\mathbf{p}_{i+1}^k\mathbf{p}_i^k\mathbf{q}_i^k)$ and the point $\mathbf{p}_{2i+1}^{k+1}$ lies inside the triangle $\Delta(\mathbf{p}_{i+1}^k\mathbf{p}_i^k\mathbf{t}_i^k)$. Thus (see Figures 16, 17)

$$0 < \pi_i^k < \beta_i^k < \rho_i^k \tag{17}$$

and

$$\rho_{2i}^{k+1} < \rho_i^k \,. \tag{18}$$

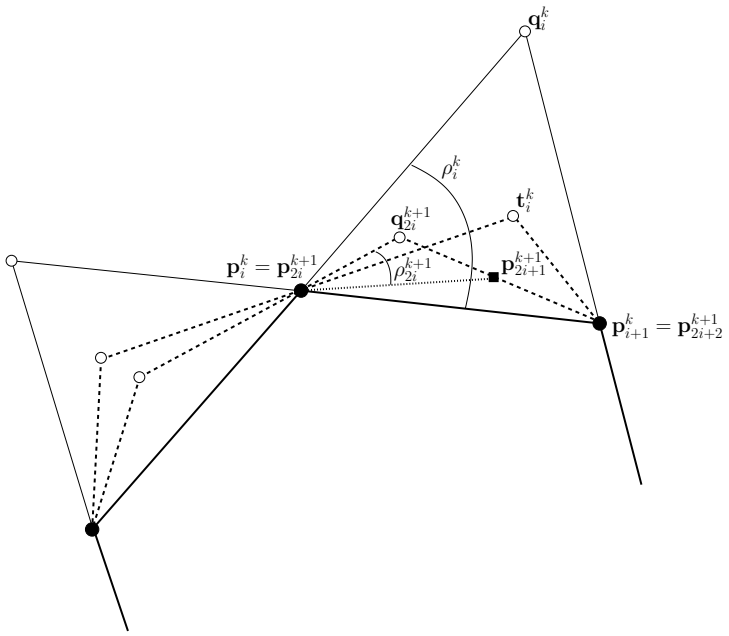


Figure 17: Relation between the angles ρ_i^k and ρ_{2i}^{k+1}.

Let $\mathbf{p}_j^{k_0}$ be a point that appears in the k_0–th iteration. Since the presented scheme is interpolatory we have

$$\mathbf{p}_j^{k_0} = \mathbf{p}_{2j}^{k_0+1} = \mathbf{p}_{4j}^{k_0+2} = \ldots = \mathbf{p}_{2^n j}^{k_0+n} \,.$$

From (18) we obtain in the point $\mathbf{p}_j^{k_0}$:

$$\rho_{2^n j}^{k_0+n} \leq (\epsilon_j)^n \rho_j^{k_0} \,, \tag{19}$$

and by (17) we have

$$\beta_{2^n j}^{k_0+n} \leq (\epsilon_j)^n \rho_j^{k_0} \,,\ \pi_{2^n j}^{k_0+n} \leq (\epsilon_j)^n \rho_j^{k_0} \,, \tag{20}$$

where the ϵ_j's are constants such that $0 < \epsilon_j < 1$. Let $\epsilon := \max_j\{\epsilon_j\}$ and $\rho^{k_0} := \max_j\{\rho_j^{k_0}\}$, where $0 < \epsilon < 1$. Then,

$$\rho_{2^n j}^{k_0+n} \leq \epsilon^n \rho^{k_0} \,, \tag{21}$$

$$\beta_{2^n j}^{k_0+n} \leq \epsilon^n \rho^{k_0} \,, \tag{22}$$

$$\pi_{2^n j}^{k_0+n} \leq \epsilon^n \rho^{k_0} \,. \tag{23}$$

[3]In a convex junction point this is guaranteed by condition (14) and the respective tangent definition.

In the triangle $\Delta(\mathbf{p}_{i+1}^k \mathbf{p}_i^k \mathbf{p}_{2i+1}^{k+1})$ we have

$$\sin(\pi_i^k) = \frac{d_i^k}{c_{2i}^{k+1}}$$

and thus

$$d_{2^n j}^{k_0+n} = c_{2^{n+1} j}^{k_0+n+1} \sin(\pi_{2^n j}^{k_0+n}). \tag{24}$$

Since the angles ρ_i^k become smaller in every step and the longest edge of a triangle is the one opposite to its biggest inner angle there exists an index k' such that for all $k \geq k'$ the circle centered in $\mathbf{p}_i^k$ with radius c_i^k contains the triangle $\Delta(\mathbf{p}_{i+1}^k \mathbf{p}_i^k \mathbf{q}_i^k)$. Thus there exists an index n_0 such that for all $n \geq n_0$ and for all j we have

$$c_{2^{n+1} j}^{k_0+n+1} \leq c_{2^{n_0+1} j}^{k_0+n_0+1} \leq \max_j \{c_{2^{n_0+1} j}^{k_0+n_0+1}\} =: c^{k_0+n_0+1}.$$

Since $\sin(x) \leq x$ for $x \geq 0$, we deduce from (24) and (23) for $n \geq n_0$ that:

$$\begin{aligned} d_{2^n j}^{k_0+n} &\leq c^{k_0+n_0+1} \pi_{2^n j}^{k_0+n} \\ &\leq c^{k_0+n_0+1} \epsilon^n \rho^{k_0}. \end{aligned}$$

Since this holds for all j we have

$$d^k \leq c^{k_0+n_0+1} \epsilon^{k-k_0} \rho^{k_0}\,, \tag{25}$$

where $k \geq k_0 + n_0$. The polygons $\{\mathbf{p}^k\}$ thus form a Cauchy sequence and this sequence of polygons converges uniformly. Each polygon being piecewise linear, the limit curve is continuous.

G^1 continuity for (locally) convex segments:

Let

$$h^k = \max_i \{h_i^k\}\,.$$

For the triangle $\Delta(\mathbf{p}_{i+1}^k \mathbf{p}_i^k \mathbf{t}_i^k)$ it holds

$$\sin(\beta_i^k) = \frac{h_i^k}{l_i^k}\,.$$

By the analogous reasoning as above for d^k we obtain for h^k (since there exists an index k' such that for all $k \geq k'$: $l_i^k \leq c_i^k$):

$$h^k \leq c^{k_0+\bar{n}} \epsilon^{k-k_0} \rho^{k_0}\,,$$

where $k \geq k_0 + \bar{n}$ for a certain value of $\bar{n}$ and

$$c^{k_0+\bar{n}} = \max_j \{c_{2^{\bar{n}} j}^{k_0+\bar{n}}\}\,.$$

In every iteration level the presented subdivision scheme constructs the new points by sampling them from a G^1–continuous conic spline Γ^k. By its Bézier construction the distance of the conic segment of Γ^k corresponding to the edge $\mathbf{p}_i^k\mathbf{p}_{i+1}^k$ is bounded by h_k. The sequence of conic splines $\{\Gamma^k\}$ thus converges to the same limit curve as the sequence of polygons $\{\mathbf{p}^k\}$.
It thus remains to be shown that the sequence of tangents $\{\mathbf{l}^k\}$, where $\mathbf{l}^k = (\mathbf{l}_i^k : i \in \mathbb{Z})$ converges uniformly. Let ξ_{2i}^{k+1} be the angle between the tangents $\mathbf{l}_i^k$ and $\mathbf{l}_{2i}^{k+1}$ in $\mathbf{p}_i^k$. Then,

$$|\xi_{2i}^{k+1}| = |\beta_{2i}^{k+1} + \pi_i^k - \beta_i^k|\,,$$

see Figure 18.

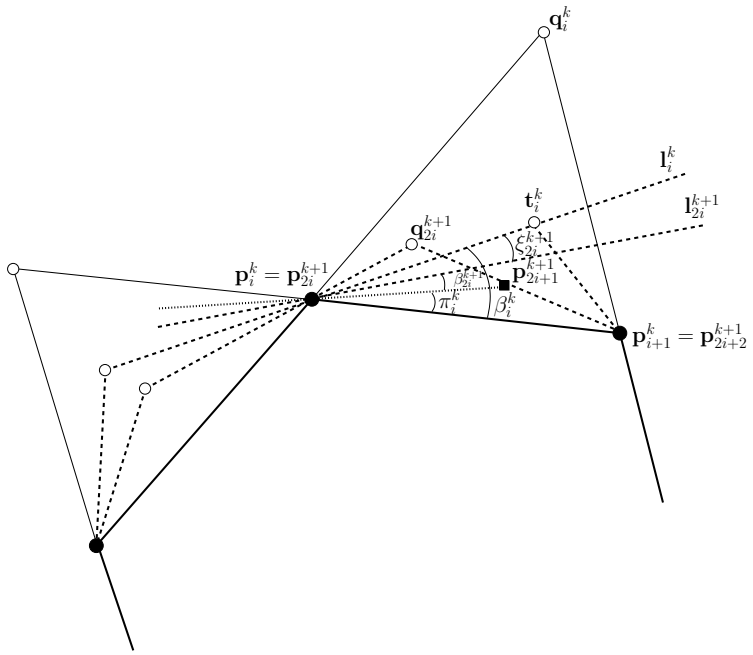


Figure 18: Illustration of the uniform convergence of the tangent sequence.

Thus by (22) and (23) with $i = 2^n j$ and $k = k_0 + n$:

$$\begin{aligned} |\xi_{2^{n+1}j}^{k_0+n+1}| &\leq |\beta_{2^{n+1}j}^{k_0+n+1}| + |\pi_{2^n j}^{k_0+n}| + |\beta_{2^n j}^{k_0+n}| \\ &\leq \epsilon^n(\epsilon+2)|\rho^{k_0}|. \end{aligned}$$

The set of tangents $\{\mathbf{l}^k\}$ thus forms a Cauchy sequence and this sequence converges uniformly. The limit curve is therefore of continuity class G^1.

C^0 and G^1 continuity for inflection points:

In an inflection point $\mathbf{p}_i^0 = \mathbf{p}_{2^k i}^k$ we define

$$\begin{aligned}
\sigma_{2^k i}^k &= \angle(\mathbf{l}_{2^{k-1} i}^{k-1}, \mathbf{g}_{2^k i}^k)\,, \\
\tau_{2^k i}^k &= \angle(\mathbf{l}_{2^k i}^k, \mathbf{e}_i)\,, \\
\xi_{2^k i}^k &= \angle(\mathbf{l}_{2^{k-1} i}^{k-1}, \mathbf{l}_{2^k i}^k)\,, \\
\beta_{l,2^k i}^k &= \angle(\mathbf{l}_{2^k i}^k, \mathbf{g}_{l,2^k i}^k)\,, \\
\beta_{r,2^k i}^k &= \angle(\mathbf{l}_{2^k i}^k, \mathbf{g}_{r,2^k i}^k)\,,
\end{aligned}$$

where $\mathbf{l}_{2^j i}^j$, $\mathbf{g}_{2^k i}^k$, $\mathbf{g}_{l,2^k i}^k$, and $\mathbf{g}_{r,2^k i}^k$ are defined as in section 3, see Figure 19. We have

$$\sigma_{2^k i}^k = \tau_{2^{k-1} i}^{k-1} - \gamma_{2^k i}^k\,, \tag{26}$$

with $\gamma_{2^k i}^k$ from (12).

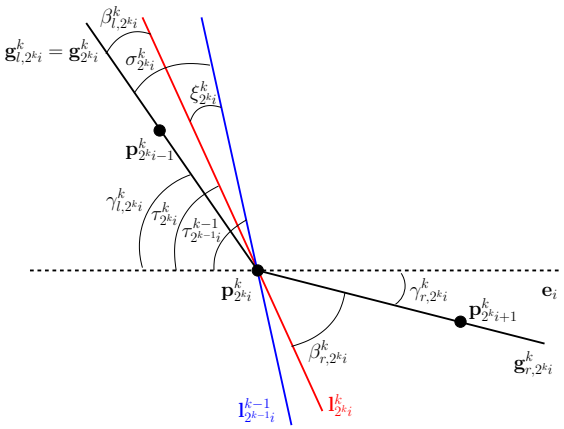


Figure 19: Illustration of C^0 and G^1 continuity for inflection points.

By construction these angles satisfy the following inequalities (see Figure 19):

$$\begin{aligned}
0 < \xi_{2^k i}^k &< \sigma_{2^k i}^k\,, && (27)\\
0 < \tau_{2^k i}^k &< \tau_{2^{k-1} i}^{k-1}\,, && (28)\\
0 < \gamma_{2^k i}^k &< \gamma_{2^{k+1} i}^{k+1}\,, && (29)\\
0 < \beta_{l,2^{k+1} i}^{k+1} &< \beta_{l,2^k i}^k\,, && (30)\\
0 < \beta_{r,2^{k+1} i}^{k+1} &< \beta_{r,2^k i}^k\,. && (31)
\end{aligned}$$

The relations (30) and (31) imply that the tangent triangles adjacent to the inflection point $\mathbf{p}_i^0$, used for the determination of the new points next to $\mathbf{p}_i^0$, become continuously flatter, thus yielding C^0 continuity in $\mathbf{p}_i^0$.
Furthermore, we have

$$\sigma_{2^k i}^k = \tau_{2^{k-1} i}^{k-1} - \gamma_{2^k i}^k > \tau_{2^k i}^k - \gamma_{2^{k+1} i}^{k+1} = \sigma_{2^{k+1} i}^{k+1} > 0\,. \tag{32}$$

By introducing an ϵ with $0 < \epsilon < 1$ we thus have

$$\sigma^k_{2^k i} \leq \epsilon^{k-1} \sigma^1_{2i}$$

and consequently

$$0 < \xi^k_{2^k i} < \sigma^k_{2^k i} \leq \epsilon^{k-1} \sigma^1_{2i} \, .$$

The angle between two consecutive tangents in an inflection point $\mathbf{p}^0_i$ thus continuously decreases yielding a G^1 continuous inflection joint in the limit. □

7 Numerical examples

In this section we want to illustrate the performance of the basic and adaptive subdivision algorithms presented in Sections 4 and 5, respectively. Generally, when the starting points are uniformly spaced we use the first proposal, while in the case of irregularly distributed vertices we apply the second one.

As concerns the forthcoming examples, we start by applying the subdivision algorithm of Subsection 4.1 to totally convex closed and open polylines with nearly uniform edges (Figures 20, 22), and successively we exploit the adaptive version of the scheme in the case of polylines with highly non-uniform edges (Figures 21, 23). As it appears, the generated curves are always convex and visually pleasing.
Then, concerning Figures 24 and 25, the goal is to illustrate the conic precision property of the proposed algorithms in both the uniform and non-uniform cases. Moving from top to bottom, the four plots in the figures have been generated by repeated application of the subdivision scheme to points sampled from a circle, an ellipse, a parabola and a hyperbola.
We then continue by showing that the curves computed through the algorithms presented in Subsection 4.1 and Section 5 are really artifact free. In fact, although a limit curve can be apparently artifact free, it is hard to tell from the display on the screen if it is acceptable or not. Two curves may look very similar on the screen, but their curvature plots may reveal important differences. The most commonly used tool for revealing significant shape differences is provided by the curvature comb of the curve. In pictures 26, 27 we have used the graphs of the discrete curvature combs of the refined polylines to show that the limit curves generated by our algorithms are indeed artifact free.
In particular, if we compare the results we get by refining the polyline in Figure 27(a) for data, that do not come from a conic section, through our adaptive algorithm and through the subdivision algorithms in [6], [14] and [25] (Figures 27(b), (c) and (d) respectively), they are only apparently very similar. Yet their curvature combs reveal substantial differences showing that neither every non-linear nor every non-uniform subdivision scheme is indeed artifact free (see Figures 27 (e)-(f)-(g)-(h)).
We close this section by illustrating the results of the subdivision algorithm of Subsection 4.2. In the first example we take the D-shape polyline of [9] in order to show the ability of the scheme to reproduce collinear vertices (see Figure 28). In the second and third examples we apply the generalized subdivision scheme to a closed,

respectively open, sequence of non-convex data to illustrate its G^1 continuity and shape-preserving interpolation properties (Figures 29, 30). The last three examples in Figures 31 and 32 deal with more complex polylines that respectively represent the cover of a mobile phone, Micky Mouse face and a bottle opener.
The shapes in Figure 31 are designed by a collection of rectilinear and convex segments where the most of the latter ones have been sampled from conic sections. The shape in Figure 32 is made of 4 independent closed polygons, some of which are non-convex. The data of the first example in Figure 31 and that of Figure 32 are courtesy of the CAD company think3 (www.think3.com). In all the considered experiments the proposed subdivision scheme turns out to work very well and clearly manifests all its characteristic features described in the previous sections.

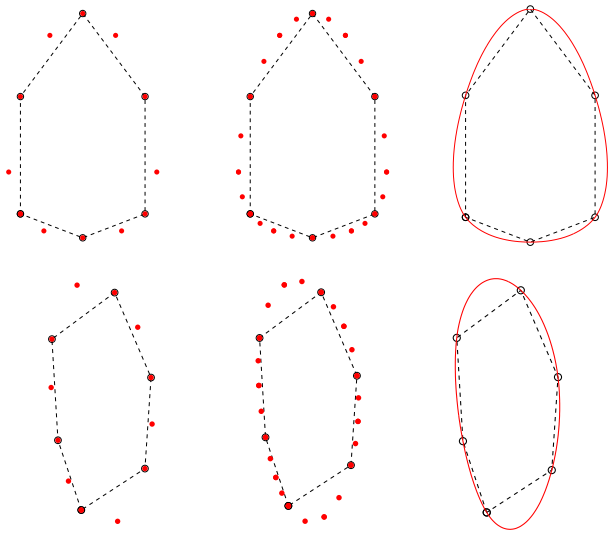


Figure 20: Application examples of the subdivision algorithm of Subsection 4.1 to nearly uniform closed polylines. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

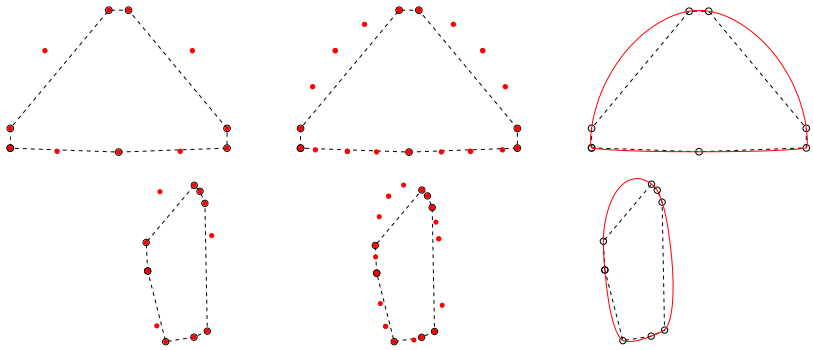


Figure 21: Application examples of the adaptive subdivision algorithm of Section 5 to highly non-uniform closed polylines. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

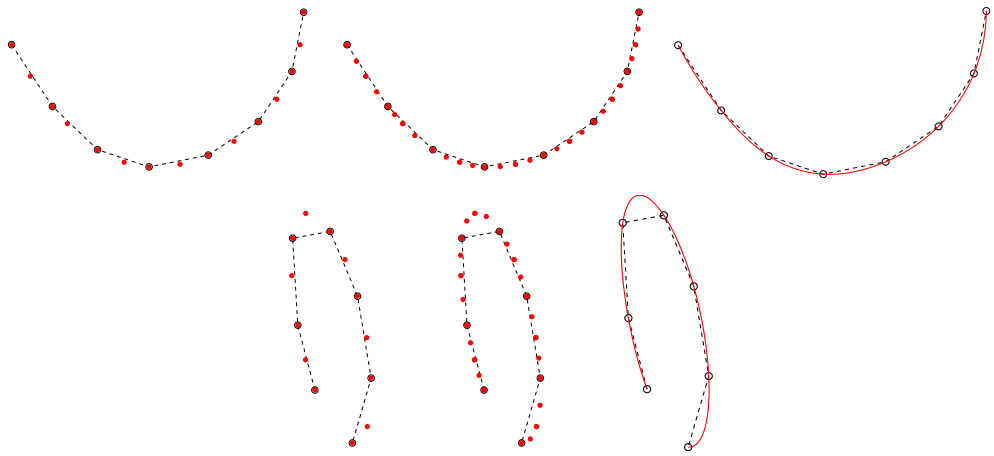


Figure 22: Application examples of the subdivision algorithm of Subsection 4.1 to nearly uniform open polylines. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

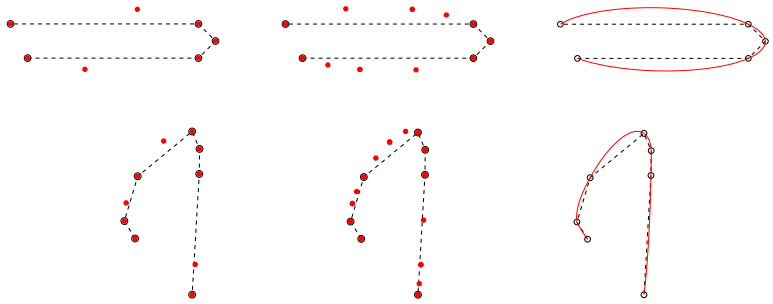


Figure 23: Application examples of the adaptive subdivision algorithm of Section 5 to highly non-uniform open polylines. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

8 Conclusions

Even though in the last years important steps forward have been taken both in the construction and analysis of interpolatory subdivision schemes [24], several problems are still open and need to be tackled in order to increase the strength and popularity of subdivision in more and more fields of application.

First of all, unlike the non-interpolatory subdivision schemes, the interpolatory ones usually generate shapes of inferior quality because, if applied to points with an irregular distribution, they provide a limit curve with more convexity changes than the starting polygon. Since, in several applications it is important to guarantee shape preservation, in this paper we have described a new interpolating subdivision algorithm enjoying this important property.

Because in CAGD it is also often necessary to have schemes able to generate classical geometric shapes, we have enriched our interpolating subdivision scheme with

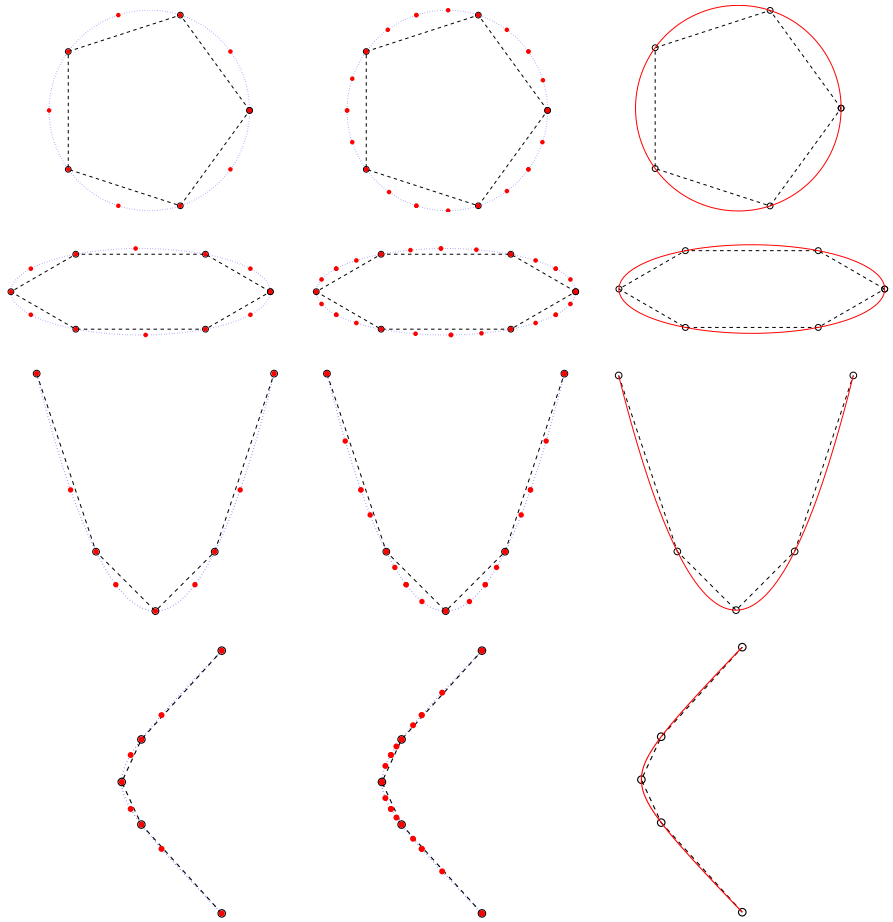


Figure 24: Uniform case: reproduction of conic sections from uniform samples by applying the subdivision algorithm of Subsection 4.1. The dotted blue line is the conic section to be reconstructed. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

the capability of including the exact representation of all conic sections. The different methods of the literature [4, 5, 22] give solutions only when the assigned points have a regular distribution. These linear subdivision schemes use non-stationary refinement rules associated with functional spaces defined via the union between polynomial and exponential functions with a free parameter.

The idea we have explored in this paper is to provide a subdivision scheme in which the conic section reproduction is obtained by adaptive geometric constructions on the given points. The advantage of doing this is that the presented non–linear scheme is able to adapt itself to any data configuration, i.e., to arbitrary irregularly distributed point sequences. Due to the underlying construction, the properties of shape preservation, conic reproduction as well as the proof of G^1 continuity of the limit curve, follow straightforwardly. The method has been illustrated by several significant examples.

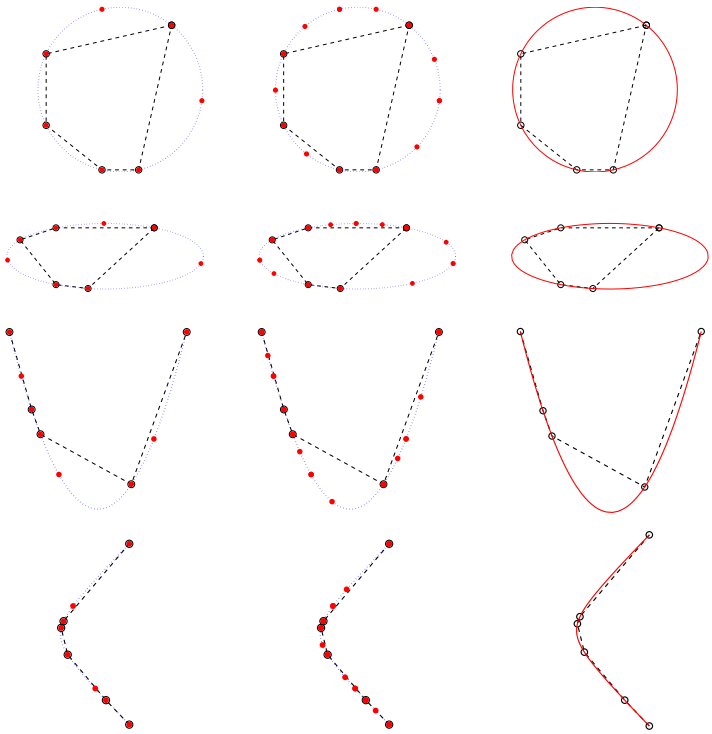


Figure 25: Non-uniform case: reproduction of conic sections from non-equispaced samples by applying the adaptive subdivision algorithm of Section 5. The dotted blue line is the conic section to be reconstructed. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

References

[1] Albrecht, G., Bécar, J.-P., Farin, G., Hansford, D., 2008. On the approximation order of tangent estimators. Comput. Aided Geom. Design 25, 80-95.

[2] Albrecht, G., 2008. Géométrie projective. In Encyclopédie Techniques de l'Ingénieur, AF 206, Editions T.I., http://www.techniques-ingenieur.fr/book/af206/geometrie-projective.html.

[3] Aspert, N., Ebrahimi, T., Vandergheynst, P., 2003. Non-linear subdivision using local spherical coordinates. Comput. Aided Geom. Design 20(3), 165-187.

[4] Beccari, C., Casciola, G., Romani, L., 2007. A non-stationary uniform tension controlled interpolating 4-point scheme reproducing conics. Comput. Aided Geom. Design 24(1), 1-9.

[5] Beccari, C., Casciola, G., Romani, L., 2009. Shape-controlled interpolatory ternary subdivision. Appl. Math. Comput. 215, 916-927.

[6] Beccari, C., Casciola, G., Romani, L., Non-uniform interpolatory curve subdivision with edge parameters built-upon compactly supported cardinal splines. Submitted.

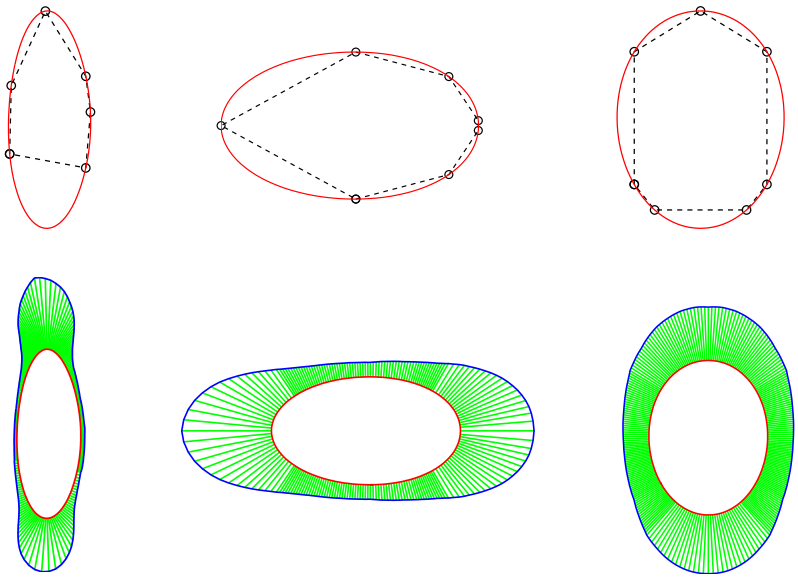


Figure 26: First row: application examples of the subdivision algorithms of Subsection 4.1 (left) and Section 5 (center and right) to totally convex closed polylines that do not come from a conic section. Second row: curvature combs of the considered examples.

[7] Chasles, M., 1865. Traité des sections coniques. Paris, Gauthier Villars.

[8] Cohen, A., Dyn, N., Matei, B., 2003. Quasilinear subdivision schemes with applications to ENO interpolation. ACHA 15, 89-116.

[9] Deng, C., Wang, G., 2010. Incenter subdivision scheme for curve interpolation. Comput. Aided Geom. Design 27(1), 48-59.

[10] Deslauriers, G., Dubuc, S., 1989. Symmetric iterative interpolation processes. Constr. Approx. 5, 49-68.

[11] Dyn, N., Levin, D., Liu, D., 1992. Interpolatory convexity-preserving subdivision schemes for curves and surfaces. Comput. Aided Design 24(4), 211-216.

[12] Dyn, N., Levin, D., 2002. Subdivision schemes in geometric modelling. Acta Numer. 11, 73-144.

[13] Dyn, N., 2005. Three families of nonlinear subdivision schemes. In: Jetter, K., Buhmann, M., Haussman, W., Schaback, R., Stoeckler, J. (Eds.), Topics in Multivariate Approximation and Interpolation, Elsevier, pp. 23-38.

[14] Dyn, N., Floater, M., Hormann, K., 2009. Four-point curve subdivision based on iterated chordal and centripetal parameterizations. Comput. Aided Geom. Design 26(3), 279-286.

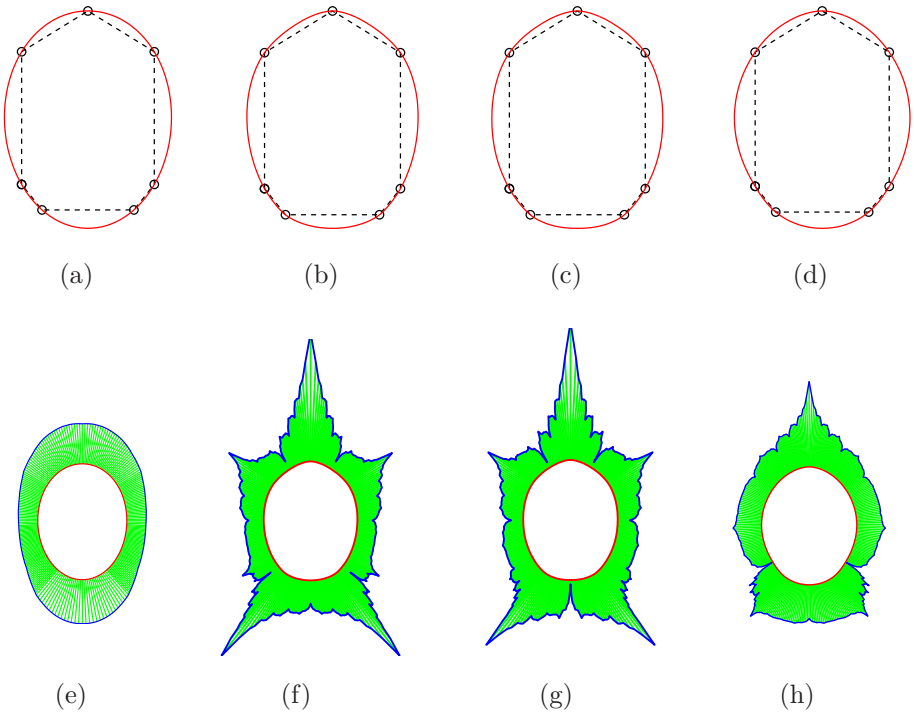


Figure 27: Comparison with the linear, non-uniform scheme in [6] and with the non-linear schemes in [14] and [25]. First row: refined polylines obtained after 6 steps of (a) our adaptive algorithm of Section 5; (b) algorithm in [6] with chord length parameterization; (c) algorithm in [14] with chord length parameterization; (d) algorithm in [25]. Second row: corresponding curvature combs.

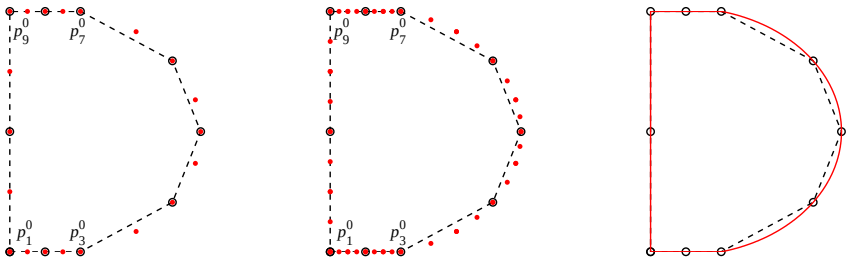


Figure 28: Application example of the subdivision algorithm of Subsection 4.2 to a closed sequence of data containing collinear vertices. The labels in the figure denote the end points of consecutive subpolygons. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

[15] Floater, M., Micchelli, C., 1998. Nonlinear stationary subdivision. In: Govil, N., Mohapatra, R., Nashed, Z., Sharma, A., Szabados, J. (Eds.), Approximation Theory: in memory of A.K. Varma, Elsevier, pp. 209-224.

[16] Hernández, V., Estrada, J., Ivrissimtzis, I., Morales, S., 2009. Curve subdivision with arc-length control. Computing 86(2-3), 151169.

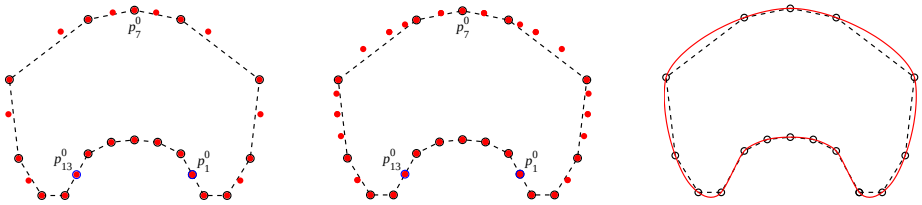


Figure 29: Application example of the adaptive subdivision algorithm of Section 5 to a closed sequence of non-convex data. The labels in the figure denote the end points of consecutive subpolygons. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

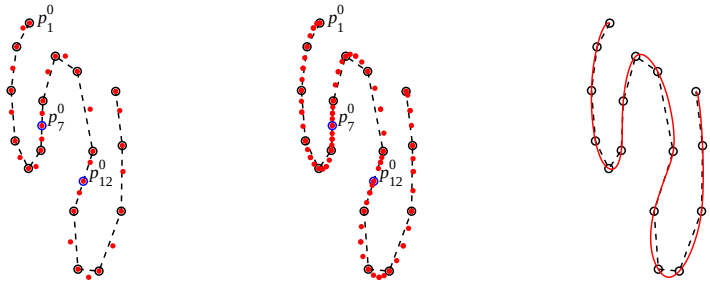


Figure 30: Application example of the subdivision algorithm of Subsection 4.2 to an open sequence of non-convex data. The labels in the figure denote the end points of consecutive subpolygons. From left to right: points at 1st and 2nd level of refinement; refined polyline after 6 steps of the algorithm.

[17] Kuijt, F., 1998. Convexity preserving interpolation - stationary nonlinear subdivision and splines. PhD Thesis, University of Twenty.

[18] Kuijt, F., Van Damme, R., 2002. Shape-preserving interpolatory subdivision schemes for nonuniform data. J. Approx. Theory 114, 1-32.

[19] Marinov, M., Dyn, N., Levin, D., 2005. Geometrically controlled 4-point interpolatory schemes. In: Dodgson, N.A., Floater, M.S., Sabin, M.A. (Eds.), Advances in Multiresolution for Geometric Modelling, Springer-Verlag, pp. 301-315.

[20] Marji, M., Siy, P., 2003. A new algorithm for dominant points detection and polygonization of digital curves. Pattern Recognition 36, 2239-2251.

[21] Pascal, E., 1910. Repertorium der Höheren Mathematik. Teubner, B.G., Leipzig/Berlin.

[22] Romani, L., 2009. From approximating subdivision schemes for exponential splines to high-performance interpolating algorithms. J. Comput. Appl. Math. 224(1), 383-396.

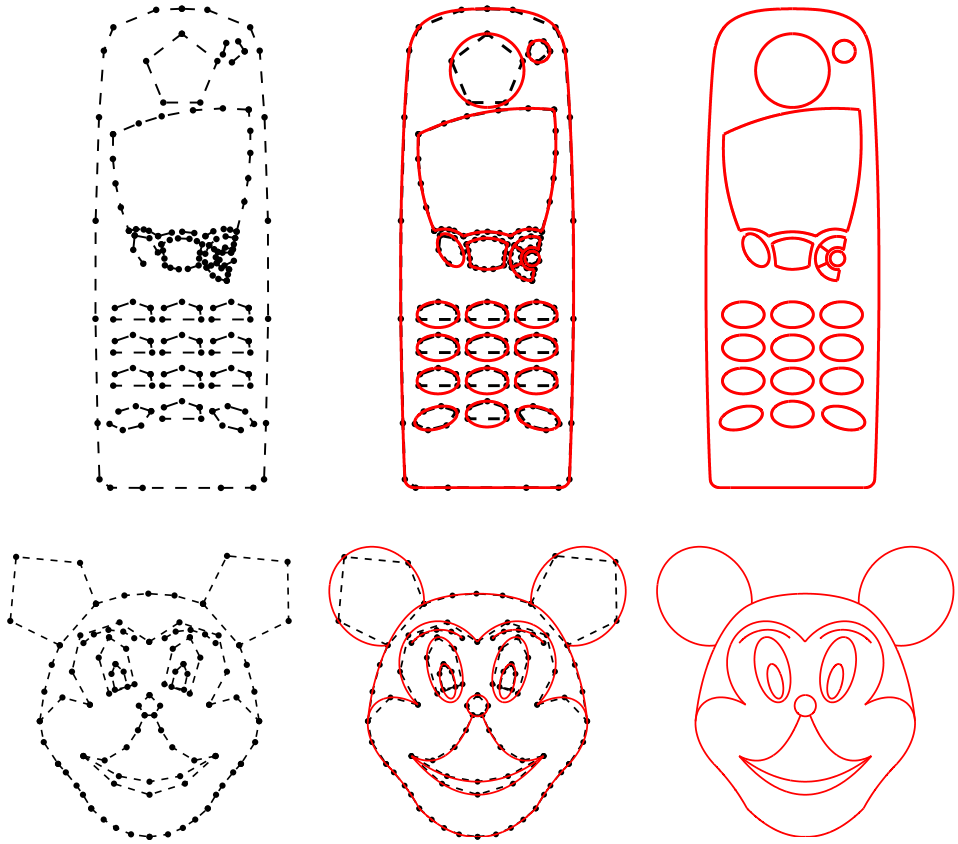


Figure 31: Application examples of the subdivision algorithm of Subsection 4.2 to complex data. Left: starting polylines. Center and Right: refined polylines obtained after 7 steps of our algorithm. Data of first row: courtesy of think3.

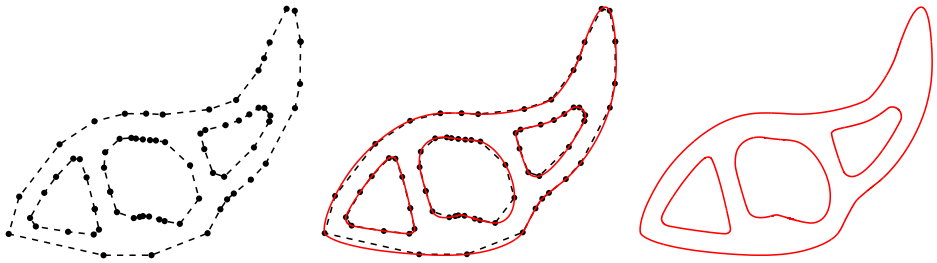


Figure 32: Application example of the subdivision algorithm of Subsection 4.2 to complex data. Left: starting polylines. Center and Right: refined polylines obtained after 7 steps of our algorithm. Data are courtesy of think3.

[23] Romani, L., 2010. A circle-preserving C^2 Hermite interpolatory subdivision scheme with tension control. Comput. Aided Geom. Design 27(1), 36-47.

[24] Sabin, M.A., 2005. Recent progress in subdivision: a survey. In: Dodgson, N.A., Floater, M.S., Sabin, M.A. (Eds.), Advances in Multiresolution for Geometric Modelling, Springer-Verlag, pp. 203-230.

[25] Sabin, M.A., Dodgson, N., 2005. A circle-preserving variant of the four-point subdivision scheme. In: Dæhlen, M., Mørken, K., Schumaker, L.L. (Eds.), Mathematical Methods for Curves and Surfaces: Tromsø 2004, Nashboro Press, pp. 275-286.

[26] Zorin, D., Schröder, P., 2000. Subdivision for modeling and animation. In SIGGRAPH Course Notes.