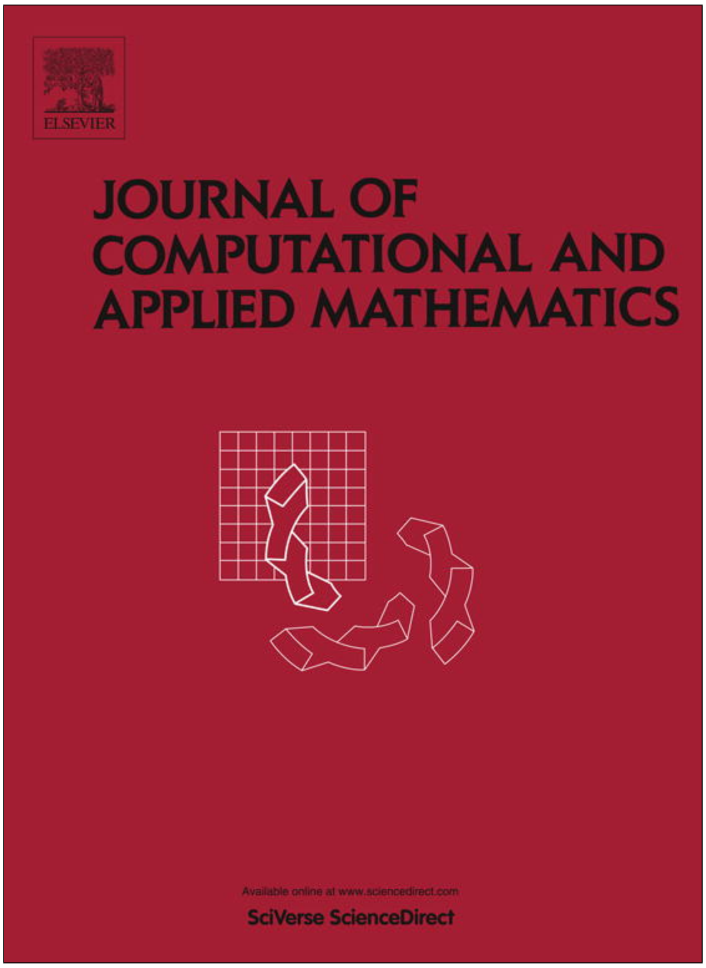

ELSEVIER
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Journal of Computational and Applied Mathematics 244 (2013) 36–48

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Interproximate curve subdivision

Xin Li [a], Jianmin Zheng [b,*]

[a] *Department of Mathematics, University of Science and Technology of China, China*
[b] *School of Computer Engineering, Nanyang Technological University, Singapore*

ARTICLE INFO

Article history:
Received 25 July 2011
Received in revised form 8 September 2012

Keywords:
Interproximate subdivision
4-point scheme
Cubic *B*-spline refinement
Tension parameter
Convexity preserving

ABSTRACT

This paper presents a new curve subdivision algorithm called *interproximate subdivision* for generating curves that interpolate some given vertices and approximate the other vertices. By the interproximate subdivision, only the vertices specified to be interpolated are fixed and the other vertices are updated at each refinement step. The refinement rules are derived to ensure that the eigenvalues of the refinement matrix satisfy the necessary condition of C^2 continuity. The interproximate subdivision also contains tension parameters assigned to vertices or edges for shape adjustment. Compared to the 4-point interpolatory subdivision scheme, the interproximate subdivision does not force the new inserted vertices to be interpolated and is thus expected to have improved behavior; and compared to the cubic *B*-spline refinement scheme, the interproximate subdivision is able to generate curves interpolating user-specified vertices. In addition, the paper also presents two extensions of the interproximate subdivision: one automatically adapts the tension parameters locally according to the geometry of the control polygon during the refinement to achieve convexity preservation and the other automatically relaxes the interpolating property of some vertices to achieve better shape behavior.

1. Introduction

Subdivision is a process that starts from a given control polygon and recursively refines the polygons by adding new vertices as linear combinations of old vertices and meanwhile fixing or updating the positions of the old vertices [1]. The process results in a sequence of refined polygons that converges to a limit curve. It is a simple and popular way to generate freeform curves. In general, subdivision schemes can be divided into two categories: approximate and interpolatory. Approximate subdivision generates curves that approximate the control polygons. Two well-known approximate curve subdivision schemes are Chaikin's algorithm [2] and the cubic *B*-spline refinement algorithm based on knot insertion of *B*-splines [3]. These algorithms actually produce uniform quadratic and cubic *B*-spline curves with C^1 continuity and C^2 continuity, respectively. Interpolatory subdivision does not move the existing vertices at each refinement step and thus the limit curve interpolates the vertices of the given control polygon. A famous interpolatory curve subdivision algorithm is the 4-point interpolatory subdivision scheme (or 4-point scheme for shorthand) [4] whose functional version was described by Dubuc [5]. The 4-point scheme is very simple and intuitive, but generates only C^1 continuous limit curves. Various modifications on the 4-point scheme have been proposed to improve the quality of curve shapes. For example, Marinov et al. presented geometrically controlled 4-point schemes in which the tension parameters vary according to the local geometry in each subdivision step [6]. Dyn et al. revised the 4-point refinement rules based on iterated chordal and centripetal parameterization [7]. Augsdörfer et al. presented six variants of the 4-point scheme based on a three stage construction [8].

* Corresponding author. Tel.: +65 67906257; fax: +65 67926559.
E-mail address: asjmzheng@ntu.edu.sg (J. Zheng).

0377-0427/$ – see front matter
doi:10.1016/j.cam.2012.11.012

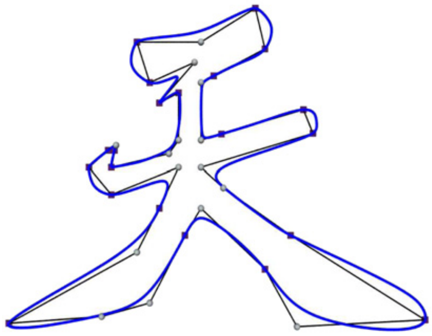


Fig. 1. A subdivision curve interpolates some control vertices shown by squares and approximates the others shown by spheres.

However, one case has been missed out, in which a subdivision curve is required to interpolate some given vertices, but is only required to approximate the other vertices (see Fig. 1). This is useful in many applications such as the shape reconstruction where some of the sample data contain noise. In this paper, we develop subdivision algorithms for this case, which we call *interproximate subdivision*. The term "interproximate" originally appeared in [9], where an interproximate spline curve was constructed to interpolate several given points and pass through some specified regions at other points. Nevertheless, we here do not specify regions at the points that are approximated.

This paper is also motivated by an apparent observation that in order to accomplish the purpose of interpolating given control points, there is no need for a subdivision scheme to freeze those vertices inserted during the refinement although the 4-point scheme always fixes all the old vertices at each subdivision step. With the interproximate subdivision, only the initially given control points are frozen, the inserted vertices during the refinement are relaxed from the interpolation requirement and thus they can be updated by some low-pass filters in the subsequent subdivision iterations. We postulate that this can lead to improved behavior.

Since the 4-point and cubic *B*-spline refinement schemes are two representative schemes of interpolatory and approximate subdivision, respectively, it is tempting to combine them to construct the interproximate subdivision. The idea of combining the 4-point and cubic *B*-spline refinement schemes has been explored in [10,11], where the results of the 4-point and cubic *B*-spline refinement schemes at each refinement step are linearly blended and consequently a smoother limit curve is yielded. Different in spirit from these approaches, our interproximate scheme combines the two subdivision schemes in a new way that for a portion of the control polygon consisting of vertices required to be interpolated, the 4-point scheme is used; for a portion consisting of vertices required only to be approximated, the cubic *B*-spline subdivision is used; and for a portion connecting an interpolatory vertex and an approximate vertex, new rules are applied. The new rules are derived to assure that the eigenvalues of the refinement matrix satisfy the necessary condition of C^2 continuity. As a result, the interproximate subdivision bridges the gap between the 4-point interpolatory subdivision and the cubic *B*-spline approximate subdivision. When all the vertices of a given control polygon are not required to be interpolated, the interproximate subdivision reduces to the cubic *B*-spline refinement. The interproximate subdivision also provides tension parameters that can be used to adjust the shape of the limit curve. Furthermore, two extensions of the interproximate subdivision are presented, which automatically adapt the tension parameters locally according to the geometry of the control polygon during the refinement to achieve convexity preservation and automatically relax the interpolating property of some vertices to achieve better shape behavior, respectively. The experiment shows that the interproximate subdivision usually produces curves with better curvature behavior than the 4-point scheme.

2. Interproximate subdivision

This section first reviews the 4-point and cubic *B*-spline refinement schemes, and then derives geometric rules for interproximate subdivision. Based on the derived geometric rules, an interproximate subdivision algorithm is presented, which is followed by continuity analysis.

2.1. 4-point scheme

Given a control polygon with vertices $\{P_i^k\}$ where k denotes the subdivision level and i is the index of vertices, the 4-point scheme generates a refined polygon with vertices $\{P_i^{k+1}\}$ by the following rules:

$$\begin{cases} P_{2i+1}^{k+1} = (P_i^k + P_{i+1}^k)\left(\dfrac{1}{2} + w\right) - (P_{i-1}^k + P_{i+2}^k)w \\ P_{2i}^{k+1} = P_i^k \end{cases} \tag{1}$$

where w is a tension parameter.

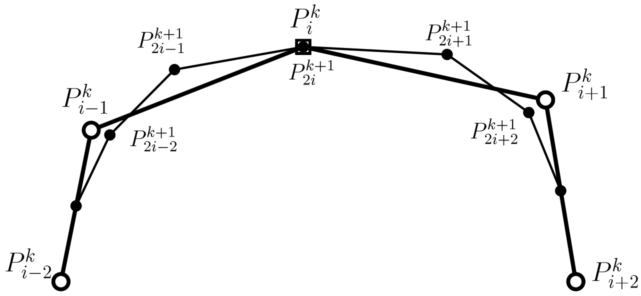


Fig. 2. Refinement in the neighborhood of an '*I*' vertex P_i^k depicted by a square.

The refined polygon is then regarded as a new control polygon and the same rules apply to it, and so on repeatedly, resulting in a sequence of polygons that converges to a limit curve. It is known that the 4-point scheme generates a C^1 continuous limit curve for $0 < w < \frac{1}{8}$. In general, $w = \frac{1}{16}$ is a popular choice, which gives the best continuity properties and positions P_{2i+1}^{k+1} on the Lagrange cubic through P_{i-1}^k, P_i^k, P_{i+1}^k and P_{i+2}^k.

2.2. Cubic B-spline refinement

B-splines can be refined. In particular, assume a uniform cubic *B*-spline curve is defined by control points $\{P_i^k\}$. The polygon consisting of $\{P_i^k\}$ approximates the *B*-spline curve. If we insert a new knot midway through each knot interval, we obtain a new set of control points $\{P_i^{k+1}\}$ which can be computed by the refinement rules:

$$\begin{cases} P_{2i+1}^{k+1} = \dfrac{P_i^k + P_{i+1}^k}{2} \\ P_{2i}^{k+1} = \dfrac{1}{2}P_i^k + \dfrac{1}{2}\dfrac{P_{2i-1}^{k+1} + P_{2i+1}^{k+1}}{2}. \end{cases} \tag{2}$$

The new control points together with a denser knot vector define the same *B*-spline curve. The new control points also form a refined polygon that is closer to the *B*-spline curve than the original control polygon. This refinement process continues, resulting in a sequence of refined polygons that converges to the *B*-spline curve.

2.3. Geometric rules for interproximate subdivision

One common pattern of the 4-point and cubic *B*-spline refinement schemes is that for each edge a new vertex called an edge point is created and each vertex in the old vertex sequence has a corresponding vertex called a vertex point in the new vertex sequence. For the 4-point scheme, each edge point is a linear combination of 4 old vertices and each vertex point is the same as an old vertex. For the cubic *B*-spline refinement scheme, each edge point is a linear combination of 2 old vertices and each vertex point is a linear combination of 3 old vertices. Now we devise geometric rules for interproximate subdivision by mixing the 4-point and cubic *B*-spline refinement schemes. Our objective is to make refinement rules have a pattern similar to that of the 4-point and cubic *B*-spline refinement schemes and to generate good curve shapes.

Given a vertex sequence $\{P_i\}$, we classify the vertices into two categories: interpolatory and approximate. An interpolatory vertex is labeled by '*I*' and remains at the same location during the refinement as in the 4-point scheme. An approximate vertex is labeled by '*A*' and is replaced by a new vertex that is a linear combination of the approximate vertex itself and the two neighboring new edge points as in the cubic *B*-spline refinement scheme. For an edge, there are three situations. The first situation is that the edge is bounded by two '*I*' vertices. Then we use the 4-point scheme to create the new edge point. The second situation is that the edge is bounded by two '*A*' vertices. We use the cubic *B*-spline refinement scheme to create the new edge point. The third situation is that the edge is bounded by one '*I*' vertex and one '*A*' vertex. In this case we need a new rule. We let the new edge point be a linear combination of three vertices among which one is the '*I*' vertex and the other two are on the '*A*' vertex side including the '*A*' vertex itself. This rule is kind of a mixture of the 4-point and cubic *B*-spline refinement schemes.

We next determine the coefficients of the linear combinations described above. Consider five successive vertices $P_{i-2}^k, P_{i-1}^k, P_i^k, P_{i+1}^k$ and P_{i+2}^k where P_i is an '*I*' point. Noting that our basic idea is that the new edge points are not needed to be interpolated, we assume that the other four vertices and P_{i-3}, P_{i+3} as well are '*A*' vertices without loss of generality. After refinement, we have five new vertices $P_{2i-2}^{k+1}, P_{2i-1}^{k+1}, P_{2i}^{k+1}, P_{2i+1}^{k+1}$ and P_{2i+2}^{k+1} (see Fig. 2). They are linear combinations of $P_{i-2}^k, P_{i-1}^k, P_i^k, P_{i+1}^k$ and P_{i+2}^k. In particular, $P_{2i}^{k+1} = P_i^k$; $P_{2i-1}^{k+1} = \alpha P_{i-1}^k + (1-\alpha-\beta)P_i^k + \beta P_{i+1}^k$ where α and β are two coefficients and the coefficients of $P_{i-1}^k, P_i^k, P_{i+1}^k$ sum to one to ensure that the rule is translation-invariant; and $P_{2i-2}^{k+1} = (1-\gamma)P_{i-1}^k + \gamma\frac{P_{2i-3}^{k+1}+P_{2i-1}^{k+1}}{2} = \frac{\gamma}{4}P_{i-2}^k + (1 - \frac{3\gamma}{4} + \frac{\alpha\gamma}{2})P_{i-1}^k + \frac{(1-\alpha-\beta)\gamma}{2}P_i^k + \frac{\beta\gamma}{2}P_{i+1}^k$ where γ is another coefficient.

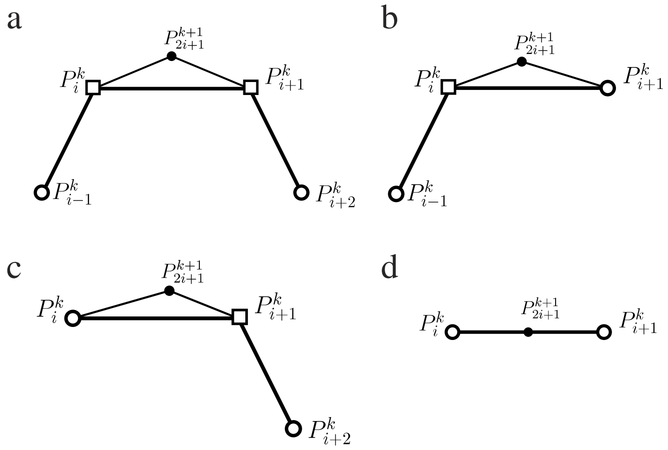


Fig. 3. Four cases for a new edge point.

P_{2i+1}^{k+1} and P_{2i+2}^{k+1} can be obtained by symmetry. These linear relationships can be written by

$$[P_{2i-2}^{k+1}, P_{2i-1}^{k+1}, P_{2i}^{k+1}, P_{2i+1}^{k+1}, P_{2i+2}^{k+1}]^T = M[P_{i-2}^{k}, P_{i-1}^{k}, P_{i}^{k}, P_{i+1}^{k}, P_{i+2}^{k}]^T \tag{3}$$

where

$$M = \begin{bmatrix} \frac{\gamma}{4} & 1-\frac{3\gamma}{4}+\frac{\alpha\gamma}{2} & \frac{(1-\alpha-\beta)\gamma}{2} & \frac{\beta\gamma}{2} & 0 \\ 0 & \alpha & 1-\alpha-\beta & \beta & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \beta & 1-\alpha-\beta & \alpha & 0 \\ 0 & \frac{\beta\gamma}{2} & \frac{(1-\alpha-\beta)\gamma}{2} & 1-\frac{3\gamma}{4}+\frac{\alpha\gamma}{2} & \frac{\gamma}{4} \end{bmatrix} \tag{4}$$

is a 5×5 matrix called the refinement matrix. The refinement matrix is useful in analyzing the continuity of the limit curve. Suppose the five eigenvalues of the matrix are $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ and λ_4 in a decreasing order, where $\lambda_0 = 1$. The necessary condition for a subdivision scheme to produce a smooth curve with bounded curvature at P_i is $\lambda_1^2 = \lambda_2 > \lambda_3$ (referring to [12] or [1]). By simple calculation, we can obtain the five eigenvalues of M : $1, \frac{\gamma}{4}, \frac{\gamma}{4}, \alpha - \beta$ and $\alpha + \beta$. Examining the 4-point and cubic B-spline refinement schemes, we can assume that $\alpha > 0$ and $\beta \leq 0$. Then $\alpha - \beta \geq \alpha + \beta$. Hence we choose $\lambda_0 = 1, \lambda_1 = \alpha - \beta, \lambda_2 = \alpha + \beta, \lambda_3 = \frac{\gamma}{4}$ and $\lambda_4 = \frac{\gamma}{4}$. Moreover, we require that $(\alpha - \beta)^2 = \alpha + \beta$ and $\alpha + \beta > \frac{\gamma}{4}$.

We parameterize α and β by introducing δ and letting $\alpha - \beta = \delta$. Then $\alpha + \beta = \delta^2$. Solving for α and β gives $\alpha = (\delta^2 + \delta)/2$ and $\beta = (\delta^2 - \delta)/2$. Moreover, it is derived from $(\alpha - \beta)^2 = \alpha + \beta \leq \alpha - \beta < 1$ and $\alpha - \beta > 0$ that $\delta \in (0, 1)$. In addition, γ should be less than $4\delta^2$. Therefore we let $\gamma = 4c\delta^2$ where $c \in (0, 1)$ is a constant. In general, we just simply let $c = \frac{1}{2}$, which conforms to the cubic B-spline refinement where $\delta = \frac{1}{2}$.

Furthermore, the parameter δ has a geometric meaning, which is described in the following theorem.

Theorem 1. *Let $q(s)$ be a quadratic polynomial curve that interpolates P_{i-1}^k, P_i^k and P_{i+1}^k at $s = -1, 0$ and 1, respectively. Then the new edge points P_{2i+1}^{k+1} and P_{2i-1}^{k+1} defined in (3) lie on curve $q(s)$ at $s = \delta$ and $s = -\delta$, respectively.*

Proof. Let $q(s) = q_0 + q_1 s + q_2 s^2$. Then $q(-1) = P_{i-1}^k$, $q(0) = P_i^k$ and $q(1) = P_{i+1}^k$, from which we get $q_0 = P_i^k$, $q_1 = \frac{P_{i+1}^k - P_{i-1}^k}{2}$, $q_2 = \frac{P_{i-1}^k + P_{i+1}^k}{2} - P_i^k$. Thus $q(\delta) = q_0 + q_1\delta + q_2\delta^2 = \frac{\delta^2 - \delta}{2}P_{i-1}^k + (1 - \delta^2)P_i^k + \frac{\delta^2 + \delta}{2}p_{i+1}^k = \beta P_{i-1}^k + (1 - \alpha - \beta)P_i^k + \alpha P_{i+1}^k$. Similarly, $q(-\delta) = q_0 - q_1\delta + q_2\delta^2 = \alpha P_{i-1}^k + (1 - \alpha - \beta)P_i^k + \beta P_{i+1}^k$. This completes the proof. □

To sum up, we define geometric rules for our interproximate refinement as follows:

(1) New edge point P_{2i+1}^{k+1} is computed based on the following four cases (see Fig. 3).

(1.a) Both P_i^k and P_{i+1}^k are labeled by '*I*':

$$P_{2i+1}^{k+1} = \left(\frac{1}{2} + w_i\right)(P_i^k + P_{i+1}^k) - w_i(P_{i-1}^k + P_{i+2}^k) \tag{5}$$

where w_i is a parameter assigned to edge $P_i^k P_{i+1}^k$.

(1.b) P_i^k is labeled by '*I*' and P_{i+1}^k is labeled by '*A*':

$$P_{2i+1}^{k+1} = \frac{(\delta_i^k)^2 - \delta_i^k}{2} P_{i-1}^k + (1 - (\delta_i^k)^2)P_i^k + \frac{(\delta_i^k)^2 + \delta_i^k}{2} P_{i+1}^k \tag{6}$$

where δ_i^k is a parameter assigned to point P_i^k.

(1.c) P_i^k is labeled by 'A' and P_{i+1}^k is labeled by 'I':

$$P_{2i+1}^{k+1} = \frac{(\delta_{i+1}^k)^2 + \delta_{i+1}^k}{2} P_i^k + (1 - (\delta_{i+1}^k)^2) P_{i+1}^k + \frac{(\delta_{i+1}^k)^2 - \delta_{i+1}^k}{2} P_{i+2}^k \quad (7)$$

where δ_{i+1}^k is a parameter assigned to point P_{i+1}^k.

(1.d) Both P_i^k and P_{i+1}^k are labeled by 'A':

$$P_{2i+1}^{k+1} = \frac{1}{2} P_i^k + \frac{1}{2} P_{i+1}^k. \quad (8)$$

(2) New vertex point P_{2i}^{k+1} is computed based on the following two cases.

(2.a) P_i^k is labeled by 'I':

$$P_{2i}^{k+1} = P_i^k. \quad (9)$$

(2.b) P_i^k is labeled by 'A':

$$P_{2i}^{k+1} = (1 - \gamma_i^k) P_i^k + \gamma_i^k \frac{P_{2i-1}^{k+1} + P_{2i+1}^{k+1}}{2} \quad (10)$$

where $\gamma_i^k = 2 \min\{(\delta_{i-}^k)^2, (\delta_{i+}^k)^2\}$ with

$$\delta_{i-}^k = \begin{cases} \delta_{i-1}^k, & \text{if } P_{i-1}^k \text{ is an 'I' point} \\ \frac{1}{2}, & \text{otherwise} \end{cases}$$

and

$$\delta_{i+}^k = \begin{cases} \delta_{i+1}^k, & \text{if } P_{i+1}^k \text{ is an 'I' point} \\ \frac{1}{2}, & \text{otherwise.} \end{cases}$$

2.4. Interproximate subdivision algorithm

Using the geometric rules as a building block, we can construct our interproximate subdivision algorithm to create a subdivision curve from a given control polygon. The inputs are a control polygon consisting of labeled vertices $\{P_i\}$ and parameters $\{w_i\}$ and $\{\delta_i\}$. Each vertex P_i is labeled by either 'I' or 'A'. For each 'I' vertex P_i, a parameter $\delta_i \in (0, 1)$ is assigned, and for each edge with two end vertices P_i and P_{i+1} both labeled by 'I', a parameter $w_i \in (0, 1/8)$ is assigned. The interproximate subdivision algorithm involves determination of geometric positions of new control points and other information as well. It recursively refines the polygons. The main steps are as follows:

- initialization: Let $P_i^0 = P_i$ and $\delta_i^0 = \delta_i$.
- for ($k = 0$; $k <$ the maximum iteration number; $k++$) {
  - Use the geometric refinement rules described in Eqs. (5)–(10) to compute P_i^{k+1};
  - If P_i^k is an 'I' vertex, label P_{2i}^{k+1} by 'I' and let $\delta_{2i}^{k+1} = \delta_i^k$;
    else label P_{2i}^{k+1} by 'A';
  - Label P_{2i+1}^{k+1} by 'A'.

  }.

Note that for a vertex P_i labeled by 'I', all subsequent vertices $P_{2^k i}^k$, $k = 1, 2, \ldots$ are also labeled by 'I' and $P_{2^k i}^k = P_i$. Thus all the refined polygons pass the position of P_i and so does the limit curve. Particularly, if all the initial control points are 'I' points, they lie on the refined polygons and thus the limit curve as well. On the other hand, if all the initial control points are 'A' points, the algorithm just degenerates to the uniform cubic B-spline refinement.

In our interproximate subdivision scheme, there are two sets of parameters w_i and δ_i. w_i is for edges bounded by two 'I' vertices and δ_i is for vertices labeled by 'I'. These parameters serve as tension parameters that can be used to adjust the tightness of the curve. Specifically, when w_i approaches zero, the curve tends to be pulled to the corresponding edge. When δ_i approaches zero, the curve is pulled towards the corresponding vertex. Figs. 4 and 5 show the effects of w_i and δ_i, respectively. In the interproximate subdivision, we choose 1/8 as a default value for w_i while 1/16 is one popular choice in the classic 4-point scheme. This is because in the classic 4-point scheme w_i occurs in all iterations of refinement and in the interproximate subdivision w_i only occurs in the first iteration of refinement, after which there is no edge bounded by two 'I' vertices. For δ_i, we choose 1/2 as the default value, which corresponds to a cubic B-spline curve as shown in Theorem 3 given in Section 2.5.

Remark 2. In the case that the input control polygon $P_0 P_1 \cdots P_n$ is open, we add two "boundary vertices" P_{-1} and P_{n+1} and label them by 'A'. The subdivision algorithm is applied to this extended polygon. The choice of P_{-1} and P_{n+1} depends on applications. Two common choices are $P_{-1} = P_0$, $P_{n+1} = P_n$ and $P_{-1} = 2P_0 - P_1$, $P_{n+1} = 2P_n - P_{n-1}$.

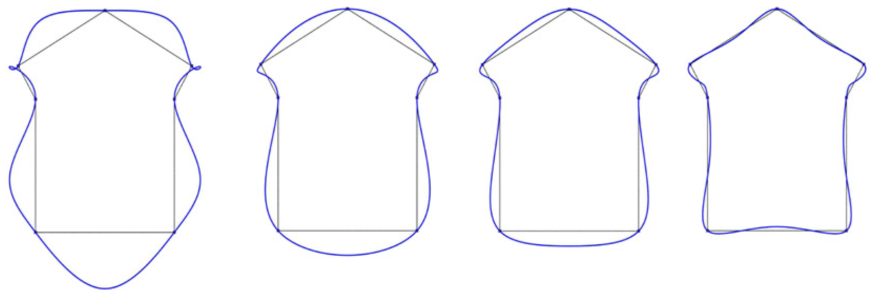


Fig. 4. Interpolation of the vertices of a polygon with $\delta_i = 0.5$ and $w_i = 1/3, 1/8, 1/16, -1/16$ (from left to right).

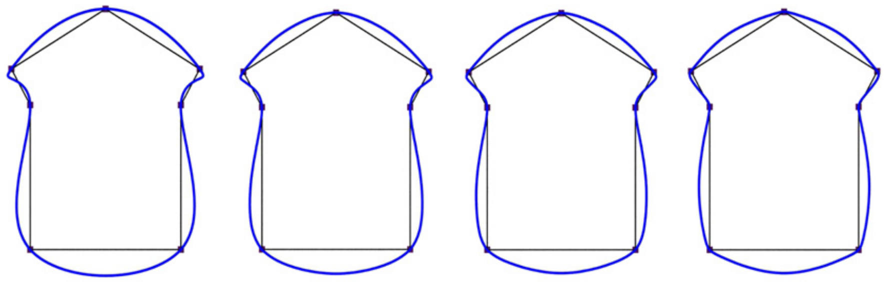


Fig. 5. Interpolation of the vertices of a polygon with $w_i = \frac{1}{8}$ and $\delta_i = 15/32, 11/32, 7/32, 3/32$ (from left to right).

2.5. Continuity analysis

By the interproximate subdivision, each '*I*' vertex will be separated from other '*I*' vertices after several iterations of refinement. The '*A*' vertices between two consecutive '*I*' vertices actually serve as the control points for a cubic *B*-spline curve segment that is a portion of the limit curve. Specifically, suppose $P^k_{2^k i}$ and $P^k_{2^k j}$ are two consecutive '*I*' vertices and the inbetween '*A*' vertices are $P^k_{2^k i+1}, P^k_{2^k i+2}, \ldots, P^k_{2^k j-2}, P^k_{2^k j-1}$. These '*A*' vertices define a cubic *B*-spline curve that is on the limit curve between $P^k_{2^k i}$ and $P^k_{2^k j}$. When the iteration number increases, the number of the inbetween '*A*' vertices increases, and the cubic *B*-spline curve segment extends and eventually the curve segment connects $P^k_{2^k i}$ and $P^k_{2^k j}$ at the two ends. Therefore we can conclude the limit curve is C^2 continuous at any point other than those '*I*' vertices on the limit curve. For the '*I*' vertices, the refinement rules are designed such that the necessary condition for the second order continuity at these points are satisfied, but whether the limit curve is indeed C^2 continuous is still under investigation. In a special case where all $\delta^k_i = 0.5$, the curve is proved to be C^2 continuous due to the following theorem.

Theorem 3. *When all* $\delta^0_i = \frac{1}{2}$, *the limit curve generated by the interproximate subdivision scheme is a cubic B-spline curve.*

Proof. Consider the polygon $\cdots P^1_{2i-1}P^1_{2i}P^1_{2i+1}\cdots$ that is obtained by one iteration of refinement. We construct a new polygon $\cdots Q^1_{2i-1}Q^1_{2i}Q^1_{2i+1}\cdots$ such that Q^1_k corresponds to P^1_k. For each '*A*' vertex P^1_i, we let $Q^1_i = P^1_i$. Obviously, all $P^1_{2i\pm1}$ are '*A*' vertices. If P^1_{2i} is an '*I*' vertex, we let $Q^1_{2i} = \frac{-P^1_{2i-1}+6P^1_{2i}-P^1_{2i+1}}{4}$.

If we apply the cubic *B*-spline refinement scheme to the new polygon $\cdots Q^1_{2i-1}\, Q^1_{2i}Q^1_{2i+1}\cdots$, we generate a limit curve that is a uniform cubic *B*-spline curve. By Theorem 1, it is easy to verify that the new vertices generated by the cubic *B*-spline refinement scheme from $\cdots Q^1_{2i-1}Q^1_{2i}Q^1_{2i+1}\cdots$ are the same as the new vertices generated by the interproximate subdivision scheme from $\cdots P^1_{2i-1}P^1_{2i}P^1_{2i+1}\cdots$ except for those '*I*' vertices P^1_{2i}. By the construction of Q^1_{2i}, we know that the cubic *B*-spline curve interpolates P^1_{2i} and $\{Q^k_{2^k i}\}$ converges to P^1_{2i}. Therefore the limit curve generated by the interproximate subdivision on $\cdots P^1_{2i-1}P^1_{2i}P^1_{2i+1}\cdots$ is the same as the cubic *B*-spline curve generated by the cubic *B*-spline refinement scheme on $\cdots Q^1_{2i-1}Q^1_{2i}Q^1_{2i+1}\cdots$, which completes the proof. □

Experiments indicate that the interproximate subdivision scheme produces curves with better curvature behavior compared to the classic 4-point subdivision scheme that requires all the intermediate vertices generated during the iterations of refinement to lie on the limit curves. Two examples are shown in Fig. 6 where the curves created using the 4-point scheme are depicted in blue and the curves created using the interproximate subdivision are depicted in red. The curvature plots placed on the right of the curves show that the 4-point scheme produces bigger curvature variance than the interproximate subdivision.

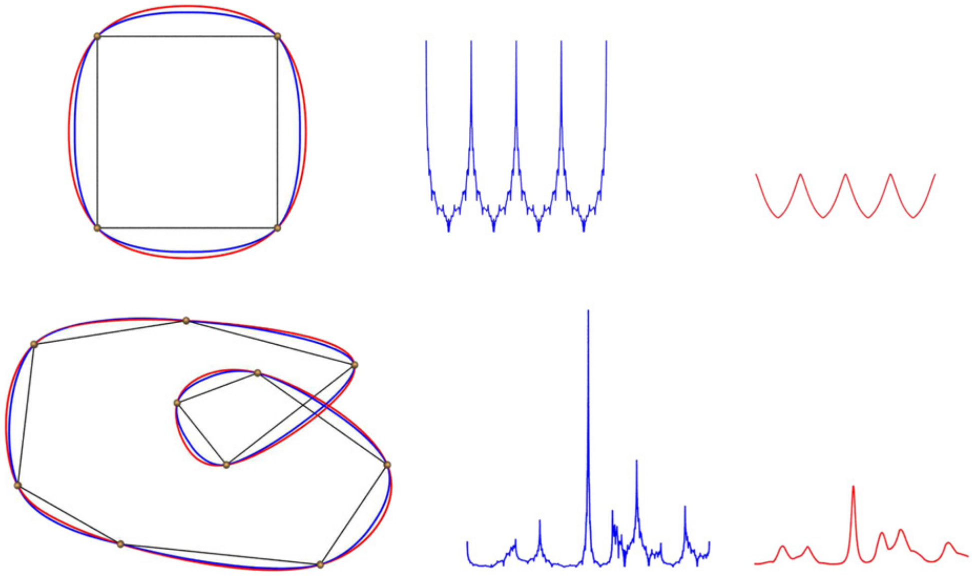


Fig. 6. Curves generated using the 4-point scheme (blue) and the interproximate subdivision (red). Their curvature plots are shown on the middle and right, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3. Two variants of interproximate subdivision in R^2

This section provides two geometrically controlled extensions of the interproximate subdivision in R^2 – convexity preserving and relaxed schemes – by selecting the tension parameter values based on local geometry or changing some vertices from label 'I' to label 'A' during the iterations of refinement.

3.1. Convexity preserving scheme

As an important shape-preserving property, convexity preserving is often required. There has been a considerable amount of research for convexity preserving subdivision algorithm. For example, [13] proposed to choose a global w depending on the initial convex functional data such that the limit function of the 4-point scheme is convex. [6,14] proposed to adapt variable tension parameters locally according to the geometry of the control polygon within the 4-point stencil.

For a given sequence of control points at refinement level k: $P^k = \{P_i^k\}$ where $P_i^k \in R^2$, we denote edge $P_i^k P_{i+1}^k$ by e_i^k. Following [6], we call an edge e_i^k to be convex if P_{i-1}^k and P_{k+2}^k lie in a common half-plane with respect to the line defined by P_i^k and P_{i+1}^k. This definition for a convex edge is locally based, which means a convex edge is not necessarily convex with respect to the entire control polygon. The polygon P^k is said to be convex if every e_i^k in P^k is convex. Note that the convex polygon defined here is equivalent to the strictly convex polygon in [6]. A convex polygon could be closed, open, or even self-intersecting.

Our objective is to modify the interproximate subdivision proposed in the preceding section to make it convexity preserving. The basic idea is that we allow the tension parameters to change from one level of refinement to another and the individual values of the tension parameters are determined locally based on the geometry of the control polygons to preserve the convexity. Note that even the cubic B-spline refinement does not have the convexity preserving property based on our definition of convexity (see Fig. 7 for an example). This suggests that simply adjusting the tension parameters w_i and δ_i^k may not be sufficient to achieve the convexity preservation, which is different from the situation in [6]. Therefore we propose to revise the formula of computing δ_i^k in (10) for 'A' vertex updating to $\gamma_i^k = 4c_i^k \min\{(\delta_{i-}^k)^2, (\delta_{i+}^k)^2\}$ where $c_k^i \in (0, 1)$ is a new parameter assigned to each 'A' vertex. This parameter has actually been suggested earlier in the eigenanalysis of the refinement matrix, where we just set it to $\frac{1}{2}$. Thus we have three sets of shape parameters: w_i, δ_i^k and c_i^k. Now we present a convexity preserving scheme that automatically determines values for them so that the convexity of the control polygons is preserved at each iteration of refinement. Since only situations (1.a)–(1.c) and (2.b) in the proposed geometric rules for the interproximate subdivision involve the shape parameters, in the following we discuss each of them. For the convenience of description, we introduce notation $L_{i,j}^{k,l}$ for the line passing through P_i^k and P_j^l, notation $[L_{i,j}^{k,l}](P)$ for the half-plane containing point P with respect to line $L_{i,j}^{k,l}$, and notation $\overline{[L_{i,j}^{k,l}](P)}$ for the half-plane not containing point P with respect to line $L_{i,j}^{k,l}$. We also define t_i^k to be a line passing through P_i^k and along the direction of $P_{i+1}^k - P_{i-1}^k$, which coincides with the tangent of a parabola interpolating $P_{i-1}^k, P_i^k, P_{i+1}^k$ at P_i^k.

(1) Consider situation (1.a) where the two vertices P_i^0 and P_{i+1}^0 of edge e_i^0 are both 'I' vertices (see Fig. 8). Re-write the formula (5): $P_{2i+1}^1 = m_i + w_i d_i$ where $m_i = \frac{P_i^0 + P_{i+1}^0}{2}$ and $d_i = m_i - \frac{P_{i-1}^0 + P_{i+2}^0}{2}$. Find λ_i and λ_{i+1} such that $m_i + \lambda_i d_i$

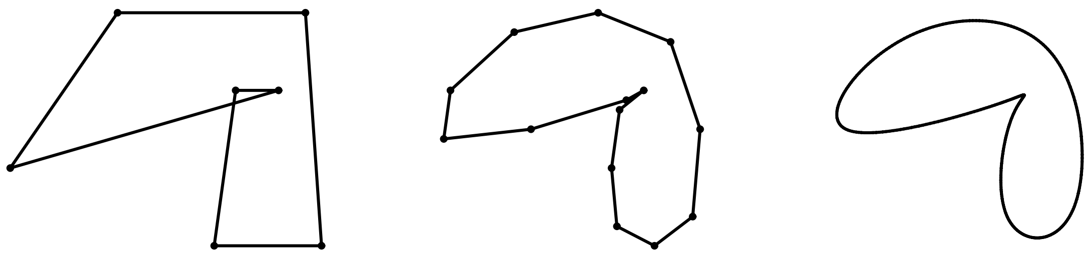


Fig. 7. An example showing that the cubic *B*-spline refinement is not convexity preserving. Left: initial control polygon; middle: refined control polygon after one refinement; right: cubic *B*-spline curve.

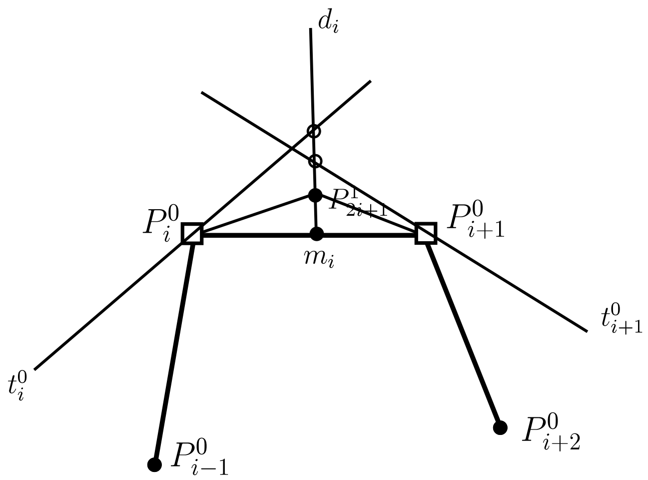


Fig. 8. Determining w_i for convexity preservation.

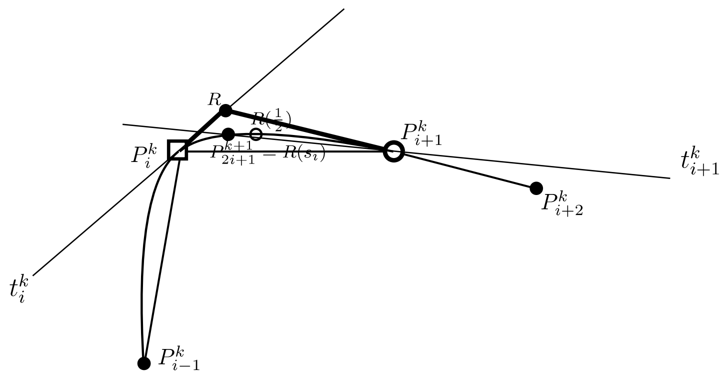


Fig. 9. Determining δ_i^k for convexity preservation.

is on line t_i^0 and $m_i + \lambda_{i+1} d_i$ is on line t_{i+1}^0. We use the values of λ_i and λ_{i+1} to bound w_i. If $\lambda_i, \lambda_{i+1} > 0$, we take $0 < w_i < \min(\lambda_i, \lambda_{i+1})$ in order to ensure that the refined polygon P^1 with P_{2i+1}^1 is convex if P^0 is convex. So we define $\mu_i = \rho \min(\lambda_i, \lambda_{i+1})$ where $\rho \in (0, 1)$ is a user defined constant. We further bound w_i by a global value $W > 0$ specified by user: $w_i = \min(W, \mu_i)$. If $\lambda_i \leq 0$, we ignore its value and simply replace it by $\lambda_i = \frac{W}{\rho}$. Similarly, if $\lambda_{i+1} \leq 0$, we simply let $\lambda_{i+1} = \frac{W}{\rho}$.

(2) Consider situation (1.b) where vertex P_i^k is labeled by 'I' and vertex P_{i+1}^k is labeled by 'A'. Theorem 1 has shown that the new edge point P_{2i+1}^{k+1} is on a quadratic polynomial curve passing through $P_{i-1}^k, P_i^k, P_{i+1}^k$. It is easy to check that the quadratic polynomial curve segment from P_i^k to P_{i+1}^k can be represented as a Bézier curve $R(s)$ with Bézier points P_i^k, $R = P_i^k + \frac{1}{4}(P_{i+1}^k - P_i^k)$ and P_{i+1}^k (see Fig. 9). Find the intersection point $R(s_i)$ between the Bézier curve $R(s)$ and the line t_{i+1}^k. We use s_i to bound δ_i^k so that either P_{2i+1}^{k+1} and P_i^k are on the same half-plane with respect to t_{i+1}^k or P_{2i+1}^{k+1} is on t_{i+1}^k. Recall that $\delta_i^k = \frac{1}{2}$ is our default choice. Therefore if $s_i \in (0, \frac{1}{2}]$, we let $\delta_i^k = s_i$. Otherwise, we ignore s_i and simply let $\delta_i^k = \frac{1}{2}$. Since situation (1.c) and situation (1.b) are symmetric, situation (1.c) can be handled in a likewise manner. For an 'I' vertex P_i^k, if its neighbors P_{i-1}^k and P_{i+1}^k both are type 'A', we obtain two δ_i^k. Then we choose the smaller one.

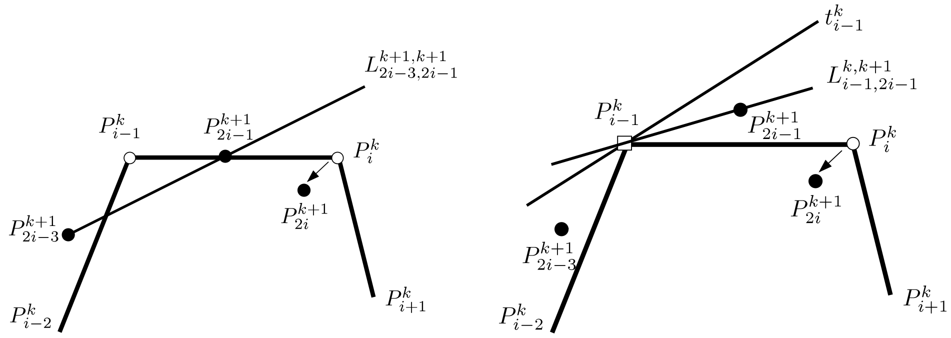


Fig. 10. Determining c_i^k for convexity preservation.

(3) Consider situation (2.b) where vertex P_i^k is labeled by 'A'. Rewrite the formula for P_{2i}^{k+1} : $P_{2i}^{k+1} = P_i^k + c_i^k\Big(4\min\{(\delta_-^k)^2, (\delta_+^k)^2\}(\frac{P_{2i-1}^{k+1}+P_{2i+1}^{k+1}}{2} - P_i^k)\Big)$. We look at edge $P_i^k P_{i-1}^k$ first. If P_{i-1}^k is an 'A' vertex as shown on the left of Fig. 10, find the maximum value c_- of c_i^k such that P_{2i}^{k+1} and P_i^k are on the two sides of line $L_{2i-3,2i-1}^{k+1,k+1}$. If P_{i-1}^k is an 'I' vertex as shown on the right of Fig. 10, find the maximum value c_- of c_i^k such that P_{2i}^{k+1} and P_i^k are on the same side of line $L_{i-1,2i-1}^{k,k+1}$. Similarly, we can find a number c_+ for c_i^k from edge $P_i^k P_{i+1}^k$. Then we choose $c_i^k = \min\{c_+, c_-, \frac{1}{2}\}$.

To show that the above scheme is convex preserving, we prove the following lemmas first.

Lemma 4. *If edge e_i^k is convex in polygon P^k, its new edge point P_{2i+1}^{k+1} is in the region $[L_{i-1,i}^{k,k}](P_{i+1}^k)\cap[L_{i+1,i+2}^{k,k}](P_i^k)\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$.*

Proof. If both P_i^k and P_{i+1}^k are 'A' vertices, P_{2i+1}^{k+1} is on the middle of edge $P_i^k P_{i+1}^k$ and thus the lemma is obviously correct. If at least one of P_i^k and P_{i+1}^k is an 'I' vertex, by the construction of P_{2i+1}^{k+1} and the assumption that P^k is convex, we know that P_{2i+1}^{k+1} is within the region $[t_i^k](P_{i+1}^k)\cap[t_{i+1}^k](P_i^k)\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$, which is contained in $[L_{i-1,i}^{k,k}](P_{i+1}^k)\cap[L_{i+1,i+2}^{k,k}](P_i^k)\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$. □

Lemma 5. *For a convex polygon P^k, edge e_{2i}^{k+1} constructed by the scheme described above in this section is convex in P^{k+1}.*

Proof. We examine e_i^k. There are four cases. Refer to Fig. 11 for the notations.

Case (a). Both P_i^k and P_{i+1}^k are 'I' vertices. The tangent line t_i^k and line $L_{i,i+1}^{k,k}$ divide the plane into four regions: $R_0 = \overline{[t_i^k](P_{i+1}^k)}\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$, $R_1 = \overline{[t_i^k](P_{i+1}^k)}\cap[L_{i,i+1}^{k,k}](P_{i-1}^k)$, $R_2 = [t_i^k](P_{i+1}^k)\cap[L_{i,i+1}^{k,k}](P_{i-1}^k)$ and $R_3 = [t_i^k](P_{i+1}^k)\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$. Since e_{i-1}^k and e_i^k are convex, by construction $P_{2i+1}^{k+1}\in R_3$, $P_{2i-1}^{k+1}\in R_2$, and $P_{2i+2}^{k+1} = P_{i+1}^k \in R_3\cap R_2$. The line $L_{i,2i+1}^{k,k+1}$ containing e_{2i}^{k+1} is in the interior of the cone $R_1\cup R_3$. Thus points P_{2i-1}^{k+1} and P_{2i+2}^{k+1} are in the same half-plane relative to e_{2i}^{k+1}. Therefore e_{2i}^{k+1} is convex.

Case (b). Both P_i^k and P_{i+1}^k are 'A' vertices. Line $L_{2i-1,2i+1}^{k+1,k+1}$ and line $L_{i,i+1}^{k,k}$ divide the plane into four regions: $R_0 = [L_{2i-1,2i+1}^{k+1,k+1}](P_i^k)\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$, $R_1 = [L_{2i-1,2i+1}^{k+1,k+1}](P_i^k)\cap[L_{i,i+1}^{k,k}](P_{i-1}^k)$, $R_2 = [L_{2i-1,2i+1}^{k+1,k+1}](P_{i+1}^k)\cap[L_{i,i+1}^{k,k}](P_{i-1}^k)$ and $R_3 = [L_{2i-1,2i+1}^{k+1,k+1}](P_{i+1}^k)\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$. Since e_{i-1}^k, e_i^k and e_{i+1}^k are convex, by construction and Lemma 4 $P_{2i-1}^{k+1}\in R_1\cap R_2$, $P_{2i+1}^{k+1}\in e_i^k$, $P_{i+1}^k\in R_2\cap R_3$, and $P_{2i+3}^{k+1}\in\overline{[L_{i+1,i+2}^{k,k}](P_i^k)}\cap(R_1\cup R_2)$. The line $L_{2i,2i+1}^{k+1,k+1}$ containing e_{2i}^{k+1} is in the interior of the cone $R_1\cup R_3$. By construction (3) given in this section and Lemma 4, we can find a positive c_{i+1}^k such that $P_{2i+2}^{k+1}\in R_2$. Thus points P_{2i-1}^{k+1} and P_{2i+2}^{k+1} are in the same half-plane relative to e_{2i}^{k+1}. Therefore e_{2i}^{k+1} is convex.

Case (c). P_i^k is an 'I' vertex and P_{i+1}^k is an 'A' vertex. As in case (a), the tangent line t_i^k and line $L_{i,i+1}^{k,k}$ divide the plane into four regions: $R_0 = \overline{[t_i^k](P_{i+1}^k)}\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$, $R_1 = \overline{[t_i^k](P_{i+1}^k)}\cap[L_{i,i+1}^{k,k}](P_{i-1}^k)$, $R_2 = [t_i^k](P_{i+1}^k)\cap[L_{i,i+1}^{k,k}](P_{i-1}^k)$ and $R_3 = [t_i^k](P_{i+1}^k)\cap\overline{[L_{i,i+1}^{k,k}](P_{i-1}^k)}$. The line $L_{i,2i+1}^{k,k+1}$ containing e_{2i}^{k+1} is in the interior of the cone $R_1\cup R_3$ and $P_{2i-1}^{k+1}\in R_2$. Since $P_{i+1}^k\in R_2\cap R_3$, we can find a positive c_{i+1}^k such that $P_{2i+2}^{k+1}\in[L_{i,2i+1}^{k,k+1}](P_{i+1}^k)$ by construction (3) given in this section. Hence points P_{2i-1}^{k+1} and P_{2i+2}^{k+1} are in the same half-plane relative to e_{2i}^{k+1} and e_{2i}^{k+1} is convex.

Case (d). P_i^k is an 'A' vertex and P_{i+1}^k is an 'I' vertex. It is proven from case (a) that edges $P_{2i-1}^{k+1}P_i^k$, $P_i^k P_{2i+1}^{k+1}$, and $P_{2i+1}^{k+1}P_i^k$ are all convex. Line $L_{2i-1,2i+1}^{k+1,k+1}$ and line $L_{i,2i+1}^{k,k+1}$ divide the plane into four regions: $R_0 = [L_{2i-1,2i+1}^{k+1,k+1}](P_i^k)\cap\overline{[L_{i,2i+1}^{k,k+1}](P_{i+1}^k)}$, $R_1 = [L_{2i-1,2i+1}^{k+1,k+1}](P_i^k)\cap[L_{i,2i+1}^{k,k+1}](P_{i+1}^k)$, $R_2 = [L_{2i-1,2i+1}^{k+1,k+1}](P_{i+1}^k)\cap[L_{i,2i+1}^{k,k+1}](P_{i+1}^k)$ and $R_3 = [L_{2i-1,2i+1}^{k+1,k+1}](P_{i+1}^k)\cap\overline{[L_{i,2i+1}^{k,k+1}](P_{i+1}^k)}$. Then the line $L_{2i,2i+1}^{k+1,k+1}$ containing e_{2i}^{k+1} is in the interior of the cone $R_1\cup R_3$, $P_{2i-1}^{k+1}\in R_1\cap R_2$, and $P_{i+1}^k\in R_1\cup R_2$. If $P_{i+1}^k\in[L_{2i,2i+1}^{k+1,k+1}](P_i^k)$, then P_i^k and P_{2i}^{k+1} will be on two sides of line $L_{2i+1,i+1}^{k+1,k}$, which contradicts the construction (3) in this section for P_{2i}^{k+1}. Thus points P_{2i-1}^{k+1} and $P_{2i+2}^{k+1} = P_{i+1}^k$ are in the same half-plane relative to e_{2i}^{k+1}, which implies that e_{2i}^{k+1} is convex. □

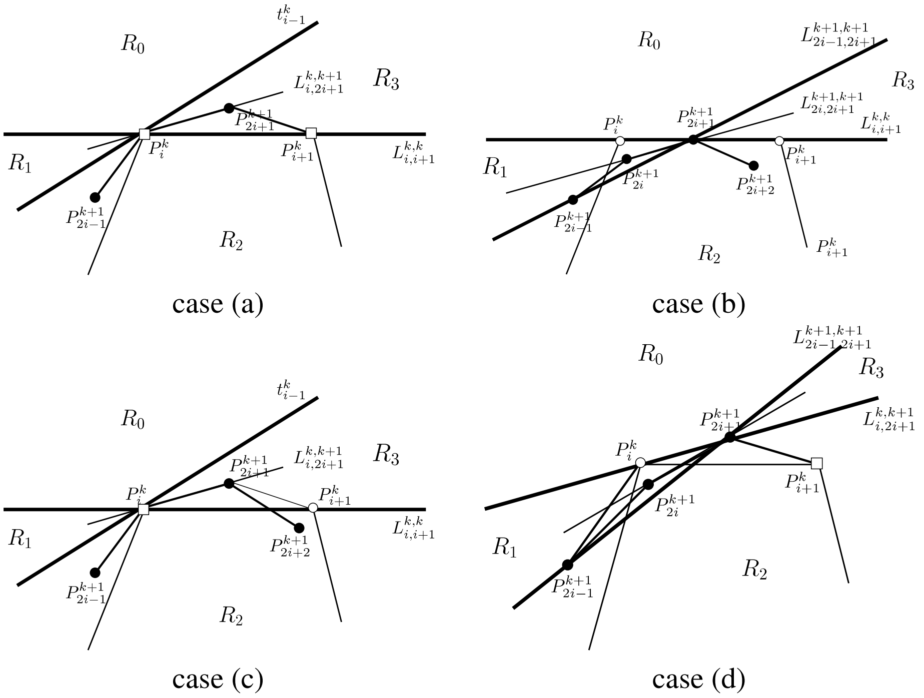


Fig. 11. Notations for proving the convexity of edge e_{2i}^{k+1}.

Similarly, we have

Lemma 6. *For a convex polygon P^k, edge e_{2i-1}^{k+1} constructed by the scheme described above in this section is convex in P^{k+1}.*

Combining Lemmas 5 and 6, we have that for each vertex P_i^k of a convex polygon P^k, its corresponding two edges P_{2i-1}^{k+1} and P_{2i}^{k+1} are both convex in the refined polygon P^{k+1}. This proves the following theorem for the convexity preserving property.

Theorem 7. *Given a convex polygon P^k, the refined control polygon P^{k+1} produced by the scheme described above in this section is convex.*

Fig. 12 shows several examples of our convexity preserving scheme, the geometrically controlled convexity preserving scheme proposed in [6], and the 4-point scheme. The last column of the figure shows the discrete curvature plotting of three limit curves. The solid points in the figure are the initial control points and these points are the only interpolatory points in our scheme. Benefited from the movable control points that are updated by a low-pass filter, our new scheme has both the convexity preserving property and good curvature behavior.

3.1.1. *Experimental analysis of smoothness*

For a non-linear subdivision scheme, it is usually very difficult to check its convergence and smoothness properties theoretically. Here we adopt the experimental method used in [15,16,6] to conduct smoothness analysis of the convexity preserving scheme by numerically checking the Hölder regularity of limit curves. Similar to [6], we define the interproximate subdivision scheme to have Hölder regularity $R_H = l + \alpha_l$ if there exist constant C and $h > 0$ such that

$$\lim_{k\to\infty} l!2^{kl} \max_i |(\Delta^l P^k)_{i+1} - (\Delta^l P^k)_i| \leq C(2^{-k}h)^{\alpha_l}$$

where $(\Delta P^k)_i = P_{i+1}^k - P_i^k$ and $(\Delta^{l+1}P^k)_i = (\Delta(\Delta^l P^k))_i$. A method is suggested in [6] to estimate α_l by computing

$$\alpha_j = -\log_2 \left(\frac{(j+1)!2^{k(j+1)} \max_i |(\Delta^{j+1}P^k)_{i+1} - (\Delta^{j+1}P^k)_i|}{j!2^{kj} \max_i |(\Delta^j P^k)_{i+1} - (\Delta^j P^k)_i|} \right)$$

such that $\alpha_j \approx 1$ for $j = 0, 1, \ldots, l-1$ and $\alpha_l \not\approx 1$. We employ this method to test the convexity preserving interproximate subdivision scheme on about 200K randomly generated control polygons (Fig. 13). Fig. 13 shows the statistics of the experiment result, from which we can see that most of the curves generated have the Hölder regularity between 2.1 and 2.9 and only about 1.5% of the curves generated have the Hölder regularity under 2.0. One example of such outliers is given in Fig. 14, where points P_0, P_1, P_2 are almost collinear and the location of P_0 lies between P_1 and P_2.

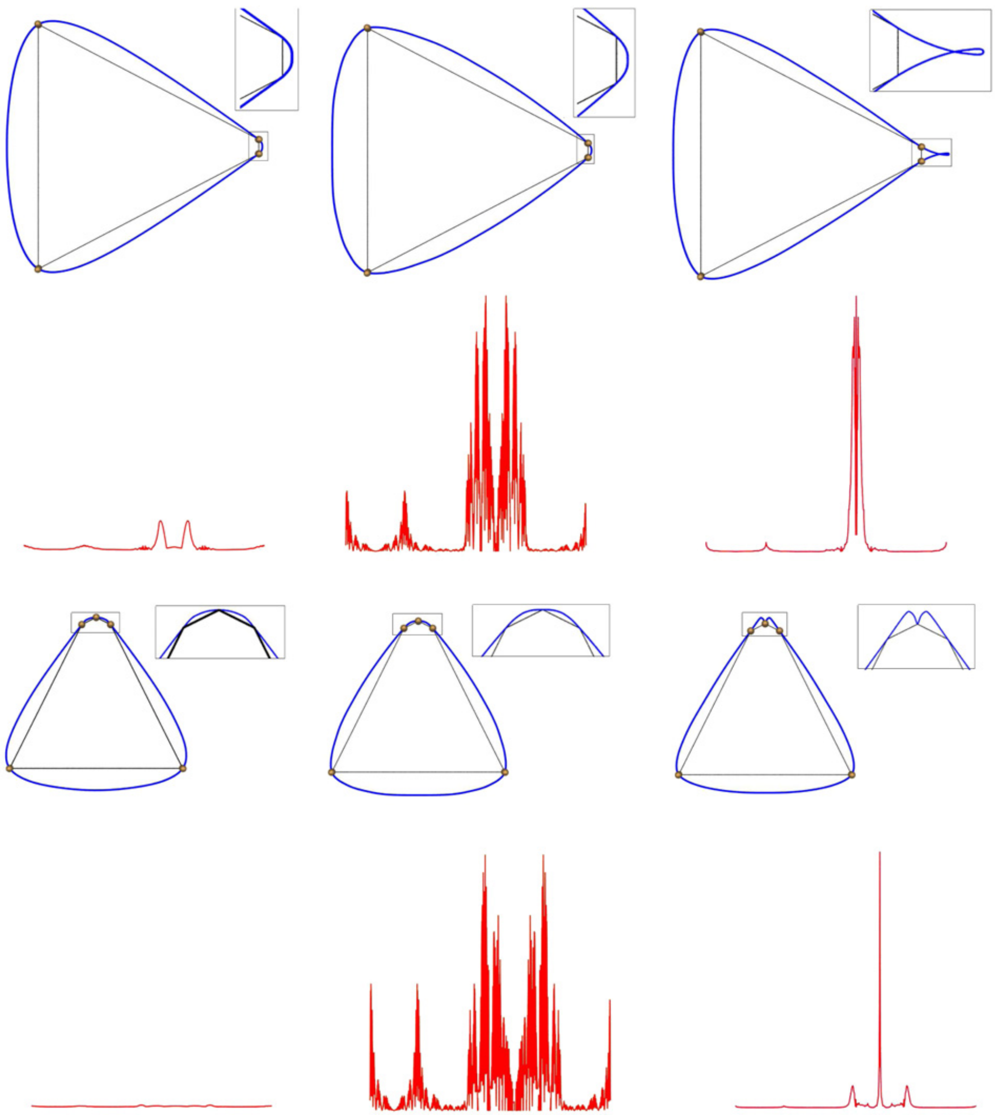


Fig. 12. Comparison with the geometrically controlled convexity-preserving scheme [6] and the 4-point scheme [4]. 1st column: the proposed convexity preserving scheme; 2nd column: the geometrically controlled convexity-preserving scheme; and 3rd column: the 4-point scheme with $w = \frac{1}{16}$. The respective curvature plotting is shown below the curves.

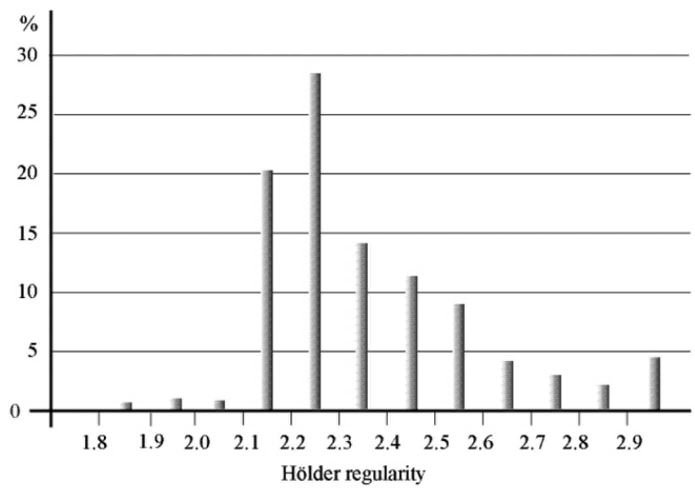


Fig. 13. Experimental Hölder regularity of the limit curves generated by the convexity preserving interproximate subdivision scheme applied to randomly generated control polygons.

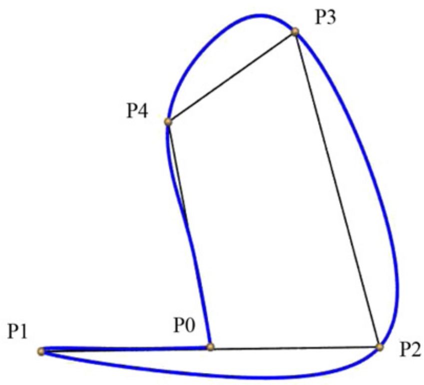


Fig. 14. An example with the Hölder regularity smaller than 2.

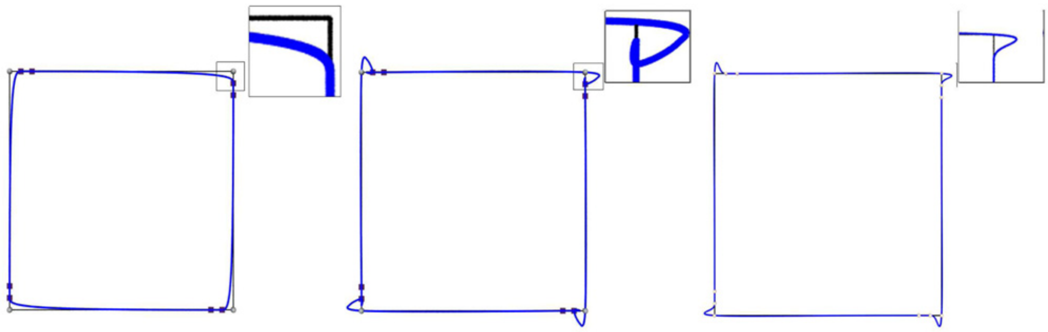


Fig. 15. Comparison of the relaxed scheme (left), the 4-point scheme (middle), and the geometrically controlled convexity preserving scheme [6] (right).

3.2. Relaxed scheme

Note that in some complicated situations where there exist straight edges or very small angles for example, it is difficult to generate visually pleasing interpolating curves that go through all the given control points, even by adapting the tension parameters to the local geometry of the control polygon. Thus it may be necessary to relax some vertices from the interpolation requirement during the iterations of refinement. Here we present a simple relaxation strategy that automatically selects the vertices to be relaxed based on the value of tension parameter δ_i^k. Specifically, for an 'I' vertex the k-level, we compute its tension parameter δ_i^k using the method proposed in the convexity preserving method. If it is smaller than a user defined constant (we choose $\frac{1}{4}$ in our experiment), we change the label of the vertex to 'A'. [8] also presented relaxed schemes, which relax all the control points and thus are actually becoming approximate schemes. Our algorithm only relaxes some of the 'I' vertices. It is also convexity preserving.

Two examples are given in Figs. 15 and 16. The results from the 4-point scheme, the proposed convexity preserving scheme, or the geometrically controlled convexity preserving scheme [6] are also shown for comparison. Enlarged local regions of the curves are placed nearby. Two vertices are relaxed in Fig. 15 and three vertices are relaxed in Fig. 16.

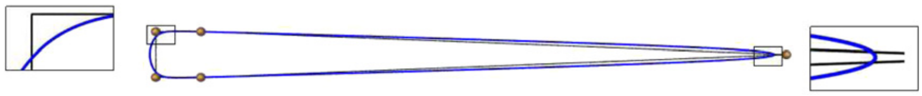
4. Conclusion

This paper presents an interproximate subdivision algorithm that can be used to generate curves interpolating some of the given vertices and approximating the others. The algorithm follows the fashion of the 4-point and cubic B-spline refinement schemes. During the subdivision process, the algorithm refines the polygon by adding new edge points. Different from the 4-point scheme, the interproximate subdivision fixes only the vertices initially specified by the user to be interpolated and all the other vertices are updated by approximate methods such as the cubic B-spline refinement. The interproximate method has applications in situations where some of the data points cannot be measured exactly. Even if all the data points are required to be interpolated, the interproximate method can give smoother interpolatory curve than the 4-point scheme. This is because the interproximate method does not require those edge vertices inserted during the refinement to be interpolated. Future work is to theoretically study whether the limit curve generated by the interproximate subdivision is C^2 continuous everywhere.

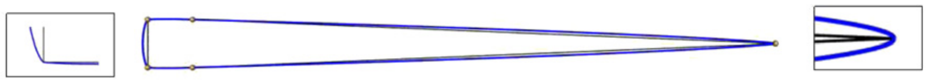
The paper also explores how to adjust the tension parameters provided by the interproximate subdivision scheme or the label types of vertices to improve shape behavior of the limit curve. Particularly, we allow the tension parameters to be variable and present a simple method that automatically adapts the parameters according to the local geometry of the polygons in the refinement process to achieve the convexity preserving property. We also presents a simple way to

(a) 4-point scheme.

(b) The proposed convexity preserving scheme.



(c) The proposed relaxed scheme.



(d) The geometrically controlled convexity preserving scheme [6].

Fig. 16. Comparison of the 4-point scheme, the proposed convexity preserving scheme, the proposed relaxed scheme and the geometrically controlled convexity preserving scheme [6].

determine which '*I*' vertices should be relaxed from their interpolatory requirement in order to improve the curve shape. More sophisticated extensions of the interproximate subdivision warrant further investigation.

Acknowledgments

This work is supported by MOE Tier-2 grant (MOE2011-T2-2-041) of Singapore, the NSF of China (Nos 11031007 and 60903148), SRF for ROCS SE, Chinese Universities Scientific Fund and Chinese Academy of Science (Startup Scientific Research Foundation).

References

[1] J. Warren, H. Weimer, Subdivision Methods for Geometric design, Morgan Kaufmann, 2002.
[2] G. Chaikin, An algorithm for high-speed curve generation, Computer Graphics and Image Processing 3 (1974) 346–349.
[3] E. Cohen, T. Lyche, R. Riesenfeld, Discrete *B*-spline and subdivision techniques in computer aided design and computer graphics, Computer Graphics and Image Processing 14 (1980) 87–111.
[4] N. Dyn, D. Levin, J.A. Gregory, A four-point interpolatory subdivision scheme for curve design, Computer Aided Geometric Design 4 (1987) 257–268.
[5] S. Dubuc, Interpolation through an iterative scheme, Journal of Mathematical Analysis and Applications 114 (1986) 185–204.
[6] M. Marinov, N. Dyn, D. Levin, Geometrically controlled 4-point interpolatory schemes, in: Dodgson, N.A., Floater, M.S., Sabin, M.A. (Eds.), Advances in Multiresolution for Geometric Modelling, New York, NY, USA, 2004, pp. 301–315.
[7] N. Dyn, M. Floater, K. Hormann, Four-point curve subdivision based on iterated chordal and centripetal parameterizations, Computer Aided Geometric Design 26 (2009) 279–286.
[8] U.H. Augsdorfer, N.A. Dodgson, M.A. Sabin, Variations on the four-point subdivision scheme, Computer Aided Geometric Design 27 (2010) 78–95.
[9] F. Cheng, B. Barsky, Interproximation: interpolation and approximation using cubic spline curves, Computer Aided Design 23 (1991) 700–707.
[10] A. Powell, J. Rossignac, Screwbender: smoothing piecewise helical motions, IEEE Computer Graphics and Applications 28 (2008) 56–63.
[11] J. Rossignac, S. Schaefer, *J*-splines, Computer Aided Design 40 (2008) 1024–1032.
[12] D. Doo, M. Sabin, Behaviour of recursive division surfaces near extraordinary points, Computer Aided Design 10 (1978) 356–360.
[13] N. Dyn, D. Levin, D. Liu, Interpolatory convexity-preserving subdivision schemes for curves and surfaces, Computer Aided Geometric Design 9 (1992) 211–216.
[14] N. Dyn, F. Kuijt, D. Levin, R.M.J.V. Damme, Convexity preservation of the four point interpolatory subdivision scheme, Computer Aided Geometric Design 8 (1999) 789–792.
[15] N. Aspert, T. Ebrahimi, P. Vandergheynst, Non-linear subdivision using local spherical coordinates, Computer Aided Geometric Design 20 (2003) 165–187.
[16] F. Kuijt, Convexity preserving interpolation—stationary nonlinear subdivision and splines, Ph.D. Thesis, University of Utah, 1987.