

ITC 2/50 Information Technology and Control Vol. 50 / No. 2 / 2021 pp. 213-223 DOI 10.5755/j01.itc.50.2.25178	An Approximation of Bézier Curves by a Sequence of Circular Arcs	
	Received 2020/01/28	Accepted after revision 2021/04/23
	 http://dx.doi.org/10.5755/j01.itc.50.2.25178	

HOW TO CITE: Nuntawisuttiwong, T., Dejdumrong, N. (2021). An Approximation of Bézier Curves by a Sequence of Circular Arcs. *Information Technology and Control*, 50(2), 213-223. <https://doi.org/10.5755/j01.itc.50.2.25178>

An Approximation of Bézier Curves by a Sequence of Circular Arcs

Taweechai Nuntawisuttiwong

Department of Computer Engineering; King Mongkut's University of Technology Thonburi; Bangkok, Thailand;
e-mail: taweechai.wee@gmail.com

Natasha Dejdumrong

Department of Computer Engineering; King Mongkut's University of Technology Thonburi; Bangkok, Thailand;
e-mail: natasha@cpe.kmutt.ac.th

Corresponding author: taweechai.wee@gmail.com

Some researches have investigated that a Bézier curve can be treated as circular arcs. This work is to propose a new scheme for approximating an arbitrary degree Bézier curve by a sequence of circular arcs. The sequence of circular arcs represents the shape of the given Bézier curve which cannot be expressed using any other algebraic approximation schemes. The technique used for segmentation is to simply investigate the inner angles and the tangent vectors along the corresponding circles. It is obvious that a Bézier curve can be subdivided into the form of subcurves. Hence, a given Bézier curve can be expressed by a sequence of calculated points on the curve corresponding to a parametric variable t . Although the resulting points can be used in the circular arc construction, some duplicate and irrelevant vertices should be removed. Then, the sequence of inner angles are calculated and clustered from a sequence of consecutive pixels. As a result, the output dots are now appropriate to determine the optimal circular path. Finally, a sequence of circular segments of a Bézier curve can be approximated with the pre-defined resolution satisfaction. Furthermore, the result of the circular arc representation is not exceeding a user-specified tolerance. Examples of approximated n^{th} -degree Bézier curves by circular arcs are shown to illustrate efficiency of the new method.

KEYWORDS: Bézier Curves, Circular arc Approximation, Analytic geometric of circle, Arbitrary degree.

1. Introduction

In geometric modeling, there are different kinds of file formats used in design and production processes. In the design process, models are usually designed by Bézier curves and B-spline but such models cannot be directly used in the production process. Besides the

vector graphics format used in design applications, the raster graphics format is another format applied in the production process. Exchanging data between processes, shapes must be converted into arcs or lines for each supported format because arcs and lines can

be used as media for both data formats: vector and raster. Using only line segments, the machine will receive numerous lines of code in production. Reducing lines of code, circular arcs must be used.

In CAD/CAM applications Bézier curves are commonly used because it can model complex curves with shape preservation during control point relocation [4]. In the past, only quadratic and cubic Bézier curves were used to design because of some limitations. However, some technologies are developed to accept arbitrary degree Bézier curves for drawing models e.g. the style spline feature in SolidWorks application [19]. Thus, investigation in the arbitrary degree of Bézier curves is appropriate to represent the model of data exchange between design and production process with modern technologies.

In manufacturing industries, Computerized Numerical Control (CNC) can only support arc and line segments to cut the workpieces. Typically, a nonlinear shape can only be subdivided into a sequence of line segments. The method can be called a linearization process. Linearizing with the high precision of interpolation, it will be dramatically increased by a large number of the line segments. To improve the efficiency of the production of CNC machine, a quadratic Bézier curve can be approximated by a sequence of arc segments [1, 10-12, 14, 20]. The curve fitting using linear and circular-arc interpolation was first presented by L. Piegl [14] in 1986. However, this method could only create a continuous path and errors always occurred at the joints of subcurves. Later, this approximation algorithm was applied to the cases of quadratic Bézier curves [20]. Although convexity and C^1 properties were improved by approximating methods [1, 10-12], these methods are only appropriate for the quadratic Bézier curve. Therefore, the approximation quadratic Bézier curves were used instead of the linearization method.

As regards the cubic Bézier curves representation, subdivision method [15] was invented to improve the efficiency of approximation and to reduce the number of arc segments. The subdivision strategies were applied by bisection method, developed by J. H. Yong et al. [21]. Later, the subdivision algorithm was improved to reduce the number of arcs by A. Riskus [16] in 2013. Besides, a Bézier curve could be described as arc splines by using Hough transform [8]. Nevertheless, the previous circular arc approximation methods

employ recursive methods with a number of factors which cause a high computational time and the techniques to cubic Bézier curves were only restricted.

In curve comparison or image matching, for example pattern recognition, curvature [17] is generally used as a feature for characterizing shapes. The curvature at any point of curve can be defined by the inverse of the radius of its osculating circle. Using curvature as a shape description, different primitives are identified with rotation, scaling, and translation invariance. Comparing a raster image (bit-mapped image) to a vector image by extracting curvature as features [7], the vector-to-raster conversion (rasterization) can be discarded. Avoiding rasterization, vector images can keep various merits such as scalability, smoothness and continuity. Nonetheless, there are few methods used to extract curvatures suitably for simple, fast and robust implementation.

Besides vector graphics formats used in applications, a sequence of points or raster images can be used as an input in industries for example trajectory robot arm. A sequence of points is approximated by arcs for supporting machines in production and movement of robotic automation. An optimal arc spline approximation was presented by G. Maier [9] in 2014. This method computes a SMAP (smooth minimum arc path) to construct the path of given point sequences with user-defined tolerance. Approximating path of robot movement, algorithm for planar movement was proposed in [2]. The path of robot movement was approximated by using three points to define a circle technique. Later in 2017, this technique was improved to apply in spatial movement [3]. Moreover, the velocity of the movement is increased while the size of code can be reduced.

This paper focuses on circular arc approximation of n^{th} -degree Bézier curves. The key parameter used in this study is linear interpolation of the equal-arc length portions (inscribed regular polygon) on the curve segments. Regarding the merit of polygon, circular arcs can be approximated that proved in [22]. Thus, the inscribed regular polygons are treated as circular arcs on Bézier curve segments. The method to approximate Bézier curves by a sequence of arc splines with inscribed regular polygon was proposed in [13]. The cubic and quintic monotonic Bézier curves were approximated. In this research, the algorithm is improved for approximating arbitrary degree Bézier curves.

Section 2 proposes a method of circular arcs approximation of an arbitrary degree Bézier curve by using geometric analysis of a circle. As a result, a set of circular arcs for a Bézier curve approximation can be obtained. Some examples of circular arc approximation of arbitrary degree Bézier curves are shown in Section 3.

2. Circular Arc Approximation for a Bézier Curve

This section presents an approximation method for a Bézier curve construction by circular arcs. Employing geometric analysis, a sequence of points on a circle has been calculated with the same arc length. However, a Bézier curve can be represented in Bernstein basis function by

$$B(t) = \sum_{i=0}^n b_i B_i^n(t),$$

where $\{b_i\}_{i=0}^n$, are the Bézier control points, and $B_i^n(t)$ is the Bernstein polynomial of degree n defined by

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}.$$

Definition 1. (Sampled points on a Bézier curve). Given a Bézier curve of degree n , a sequence of control points p_i , $0 \leq i \leq N$, consists of the points on the curve by substituting each parameter t_i into Bézier curve $B(t)$. Then,

$$p_i = B(t_i) \tag{1}$$

where $t_i = i / N$.

For any arc lengths of $B(t)$ from a to b , point-based methods applied by Simpson's rule [5] is used to estimate the arc length of a Bézier curve as follows:

$$L(B(t)|_{[a,b]}) \approx \frac{1}{6} (|-3p_0 + 4p_1 - p_2| + 4|p_2 - p_0| + |p_0 - 4p_1 + 3p_2|), \tag{2}$$

where $p_i = B(t_i)$, $a = t_0$, $b = t_2$, and $p_i = B((a+b)/2)$.

Let s_i , $0 \leq i \leq N$, be the arc length from p_0 to p_i satisfying $s_0 < s_1 < s_2 < \dots < s_N$, where $s_0 = 0$ and s_N is the length of Bézier curve. For a large number of N , the

value of s_i is gradually increased. Accordingly, s_i can be used as the domain for sampling curves by uniform arc length. As a result, the domain of Bézier curves is changed from parameter t to space s .

Definition 2. (Equal-length sampled points). Given a sequence of Bézier sampled points, denoted by $\{p_i\}_{i=0}^N$, a vertex sequence on such a Bézier curve, denoted by $V = \{v_0, v_1, v_2, \dots, v_M\}$, can be computed from sampling Bézier curve into M equal-length arc portions.

By Equation (2), the arc length of each curve segment from v_0 to v_j is defined by

$$L_j = L|_{v_0}^{v_j},$$

where $0 \leq j \leq M$. Then, the equal-arc length portions must satisfy

$$L_j - L_{j-1} = L_{j+1} - L_j.$$

Any uniform arc vertices v_j can be directly calculated by assigning parametric variables, denoted by t_j . Then,

$$v_j = B(t_j),$$

where i is the index of arc length s_i at point $B(t_i)$ and it is satisfied by the condition $|L_j - s_i| \rightarrow 0$.

In this paper, uniform arc length vertices are used to detect circular arcs. Such a sequence of vertices will be linearly interpolated to construct an inscribed polygon in a curve. If the sides of a polygon are abundant, the polygon is approaching a curve. It can be implied that the error of linear interpolation depends on a number of sampled points. The more sampled points, the less error of linear interpolation.

Lemma 1. A Bézier curve $B(t)$ has an appropriate number of uniform arc length sampled points, denoted by M , that is satisfied the given error of linear interpolation, called tolerance (τ) if

$$M = \sqrt{\frac{|B''(t)|}{8\tau}}, \quad 0 \leq \tilde{t} \leq 1, \tag{3}$$

where \tilde{t} is a parameter.

Proof. A linear interpolating curve $C(t)$ in an interval $[a, b]$ of $B(t)$ is given by

$$C(t) = B(a) + \frac{B(b) - B(a)}{b - a} (t - a). \tag{4}$$

It has an error of interpolation, denoted by $\varepsilon = |\mathbf{B}(t) - \mathbf{C}(t)|$. By Rolle's theorem, it is obtained:

$$\varepsilon = \frac{(t-a)(t-b)}{2} |B''(\tilde{t})|, \quad a \leq \tilde{t} \leq b. \tag{5}$$

In calculus, an error ε can be maximized and simplified in terms of interval width, $h = b - a$. The maximum ε can be considered by τ and defined as follows:

$$\tau = \frac{h^2}{8} |B''(\tilde{t})|. \tag{6}$$

The relationship between width of each segment, denoted by h , and number of sampling points, denoted by M , inversely varies to each other as $M = 1/h$. Then, it can be concluded that

$$\tau = \frac{1}{8M^2} |B''(\tilde{t})|. \tag{7}$$

Therefore,

$$M = \sqrt{\frac{|B''(\tilde{t})|}{8\tau}}.$$

By Lemma 1, the number of sampled points can be evaluated adaptively due to the user defined tolerance. Beside tolerance, shape and curve resolutions are also affected by interpolation error. However, shape and curve resolution depend on a set of control points that can be expressed in term of $\mathbf{B}(t)$. Hence, Lemma 1 can be applied on any shapes and resolutions of Bézier curves.

Definition 3. (Representation of Bézier subcurves). A Bézier curve defined by a sequence of vertices, $V = \{v_0, v_1, v_2, \dots, v_M\}$, can be subdivided into a sequence of subcurves, denoted by $\{q_0, q_1, q_2, \dots, q_{M-1}\}$. Then, q_j , where $0 \leq j \leq M - 1$, is a Bézier subcurve represented by vertices, v_j and v_{j+1} , and edge, denoted by e_j .

By Definition 2, a vertex sequence has equal arc-length portions. Considering a set of edges, denoted by $E = \{e_0, e_1, e_2, \dots, e_{M-1}\}$, an edge e_j , connecting between two vertices v_j and v_{j+1} , can be acted as Bézier subcurves. Therefore, all edges are equal arc length.

After choosing points on a Bézier curve, the sampled points, denoted by $\{v_i\}_{i=0}^M$, divide the curve into

M subcurves. Let subcurve q_j be a part of the Bézier curve $\mathbf{B}(t)$ with vertices v_i and v_{i+1} . The inner angle, a_k , between subcurves, v_{k-1} , and v_k , where $k = 1, 2, \dots, M - 1$ is the cosine of this angle formed by the vectors $\overrightarrow{v_{k-1}v_k}$ and $\overrightarrow{v_kv_{k+1}}$. Consequently, the inner angle a_k can be calculated

$$a_k = \cos^{-1} \left(\frac{\overrightarrow{v_{k-1}v_k} \cdot \overrightarrow{v_kv_{k+1}}}{\| \overrightarrow{v_{k-1}v_k} \| \| \overrightarrow{v_kv_{k+1}} \|} \right). \tag{8}$$

Definition 4. (Regular polygon inscribed in Bézier subcurves). By linear interpolation, a sequence of subcurves, denoted by $\{q_i, q_{i+1}, q_{i+2}, \dots, q_j\}$, has a sequence of edges $\{e_i, e_{i+1}, e_{i+2}, \dots, e_{j-1}\}$, where $1 \leq i < j \leq M$. Such sequence of subcurves has an inscribed regular polygon if $a_{i+1} = a_{i+2} = \dots = a_{j-1}$.

Definition 5. (Bézier incidence matrix). Suppose that $v_0, v_1, v_2, \dots, v_M$ are the vertices and $e_0, e_1, e_2, \dots, e_{M-1}$ are the edges of curve. Then, the incidence matrix with respect to this ordering of E and V is the $M \times (M + 1)$ matrix $Q = [q_{ij}]$, where

$$q_{ij} = \begin{cases} 1, & \text{when } e_i \text{ is incident with } v_j \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Given a sequence of vertices $\{v_0, v_1, v_2, \dots, v_M\}$ and a sequence of edges $\{e_0, e_1, e_2, \dots, e_{M-1}\}$ are corresponding to the Bézier curve portions, an incidence matrix of Bézier curve can be represented by

$$\begin{matrix} & v_0 & v_1 & v_2 & v_3 & \dots & v_{M-1} & v_M \\ e_0 & \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \end{bmatrix} \\ e_1 & \begin{bmatrix} 0 & 1 & 1 & 0 & \dots & 0 & 0 \end{bmatrix} \\ e_2 & \begin{bmatrix} 0 & 0 & 1 & 1 & \dots & 0 & 0 \end{bmatrix} \\ \vdots & \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \end{bmatrix} \\ e_{M-2} & \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} \\ e_{M-1} & \begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{bmatrix} \end{matrix}.$$

Considering the incidence matrix, the adjacent edges share a joint point joining two subcurves. Using the incidence matrix to detect circular arcs, a given sequence of edges will be classified as a circular arc if a regular polygon can be inscribed in a given sequence -- satisfied Definition 4.

Definition 6. (Interior vertices on circular path). Let e_{i-1} , e_i , and e_{i+1} be a sequence of edges on a Bézier curve portion. A pair of edges, e_{i-1} and e_i , is connected by vertex v_i . Another pair of edges, e_i and e_{i+1} is con-

ned by vertex v_{i+1} . Both vertices v_i and v_{i+1} are the interior vertices on a circular path if inner angles a_i and a_{i+1} are equal.

Definition 7. (Incidence matrix Transformation). Let a_i and a_{i+1} be inner angles at vertices v_i and v_{i+1} , respectively. Edges e_i and e_{i+1} will be a circular path if inner angles a_i and a_{i+1} are equal. Then, the incidence matrix will be transformed by combining rows of edges e_i and e_{i+1} to circular path c_k defined by

$$c_{kj} = e_{ij} \vee e_{(i+1)j},$$

where $0 \leq i \leq M$.

Definition 8. (Circular path matrix). The m circular paths of an incidence matrix with $M + 1$ vertices can be defined as the $m \times (M + 1)$ boolean matrix $C = \{c_{ij}\}$, in which the element in the i^{th} row and the j^{th} column is 1 if the i^{th} circular path contains j^{th} vertex; otherwise, c_{ij} is 0.

Demonstrating incidence matrix with interior vertices, let $\{v_0, v_1, v_2, v_3, v_4, v_5\}$ be a sequence of vertices and $\{e_0, e_1, e_2, e_3, e_4\}$ be a sequence of edges in Bézier curve portion. The incidence matrix of such a Bézier curve can be illustrated by

$$Q^{(0)} = \begin{matrix} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \\ e_0 & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\ e_1 & \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \\ e_2 & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \\ e_3 & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \\ e_4 & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}.$$

Assumes a sequence of inner angles to be a set $\{a_1, a_2, a_3, a_4\}$ where $a_1 = a_2$ and $a_3 = a_4$. First, two edges e_0 and e_1 are combined into a circular path c_0 . Then, incidence matrix is transformed as follows:

$$Q^{(1)} = \begin{matrix} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \\ c_0 & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \\ e_2 & \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \\ e_3 & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \\ e_4 & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}.$$

Considering inner angle a_1 and a_2 at vertices v_1 and v_2 respectively, the edge e_2 will be merged with circular path c_0 and incidence matrix is transformed as follows:

$$Q^{(2)} = \begin{matrix} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \\ c_0 & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \\ e_3 & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \\ e_4 & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}.$$

The next step of transforming the incidence matrix, inner angles a_2 and a_3 are considered. An incidence matrix will not be changed because inner angles a_2 and a_3 are not equal. The last pair of inner angles a_3 and a_4 are equal so the incidence matrix will be combined e_3 and e_4 to a new circular path as follows:

$$Q^{(3)} = Q^{(4)} = \begin{matrix} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \\ c_0 & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \\ c_1 & \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix}.$$

As a result of incidence matrix transformation in this demonstration, two circular paths are generated by merging edges with the same inner angles. Therefore, there are two circular arcs in such a Bézier curve portion. The first circular arc contains a sequence of vertices $\{v_0, v_1, v_2, v_3\}$. Other circular arcs contain only a sequence of vertices $\{v_3, v_4, v_5\}$.

Theorem 1. A Bézier curve can be expressed by a sequence of circular arcs, denoted by $\{c_1, c_2, \dots, c_m\}$, where c_k can be constructed by a sequence of edge E_k and $m < M$. Then, each circular arc, the following conditions hold true.

- c_k consists of edges e_{ki} that are incident with the same set of vertices v_{kj} .
- A circular arc c_k is a representative for a sequence of edges E_k if and only if $a_{ki} \approx a_{(k+1)i}$ and $\left| \frac{v_{ki} v_{k(i+1)}}{|v_{ki} v_{k(i+1)}}| - \left| \frac{v_{k(i+1)} v_{k(i+2)}}{|v_{k(i+1)} v_{k(i+2)}}| \right| \rightarrow \varepsilon$.

Proof. Suppose q_{i-1} , q_i , and q_{i+1} are consecutive subcurves with equal arc length. The line segment e_{i-1} , e_i , and e_{i+1} are linearly interpolated in each subcurves. If all line segments have equal length and inner angle are the same, then $\{q_{i-1}, q_i, q_{i+1}\}$ can be represented as circular arc with radius R as demonstrated in Figure 1.

For evaluating $\{q_{i-1}, q_i, q_{i+1}\}$ as a circular arc, all points on these subcurves must be points on a circle with radius R . Considering maximum error ε between e_i and q_i , it can be determined that

$$\varepsilon = |q_i - e_i| = \left| R - R \cos\left(\frac{\theta}{2}\right) \right|.$$

The maximum error can be reduced by bisection subcurves as shown in Figure 2. By bisection of subcurves, new subcurves, $q_i^{(1)}$, are subdivided and new line segments, e_i , are interpolated. Furthermore, the maximum error is reduced into

Figure 1

A sequence of edges as circular arc and its inscribed polygon segment

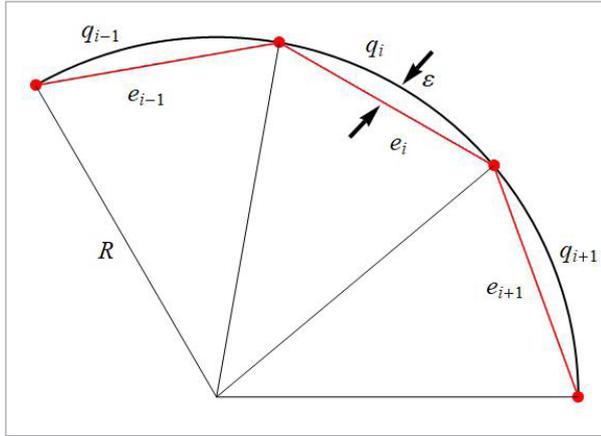
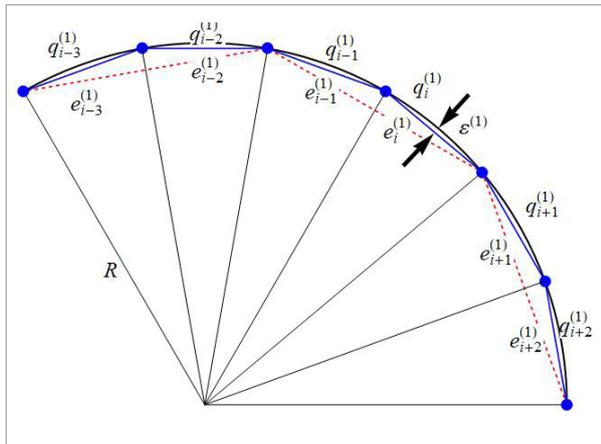


Figure 2

The new sequence of edges with subcurve bisection



$$\varepsilon^{(1)} = |q_i^{(1)} - e_i^{(1)}| = \left| R - R \cos\left(\frac{\theta}{4}\right) \right|.$$

After bisectioning subcurves times, the maximum error can be defined as follows:

$$\varepsilon^{(n)} = |q_i^{(n)} - e_i^{(n)}| = \left| R - R \cos\left(\frac{\theta}{2^{n+1}}\right) \right|. \quad (10)$$

Thus,

$$\lim_{n \rightarrow \infty} \varepsilon^{(n)} = \lim_{n \rightarrow \infty} \left| R - R \cos\left(\frac{\theta}{2^{n+1}}\right) \right| = 0.$$

3. Algorithm and Examples

1. Algorithm for Circular Arcs Approximation

The algorithm uniformly chooses the points on an n^{th} -degree Bézier curve by two techniques by time parameter t and space s . For the first calculation, a great number of generated points, denoted by N , is used to compute an arc length array, $\{s_i\}_{i=0}^N$. On the other hand, the input curve is reformed into subcurves $\{q_i\}_{i=1}^M$, where $M < N$, with equally uniform arc length. These subcurves are represented in form of vertices, $\{v_i\}_{i=0}^M$, and edges, $\{e_i\}_{i=0}^{M-1}$, where v_i is an endpoint of each subcurve and e_i connects two vertices. Then, an n^{th} -degree Bézier curve can be expressed by the incidence matrix Q . Furthermore, inner angles, $\{a_i\}_{i=1}^{M-1}$, of each subcurve are calculated and rounded into an integer. Considering each pair of inner angles a_{i-1} and a_i incidence matrix will be transformed if both inner angles a_{i-1} and a_i are equal. After transforming the incidence matrix for $M-1$ times, a sequence of circular arcs is represented as a row of the incidence matrix. The algorithm is shown in Algorithm 1.

Algorithm 1: Circular Arcs Approximation

Input: The n^{th} -degree Bézier control points $b[0 \dots n]$ with the pre-defined tolerance τ

Output: A sequence of circular arcs $c[0 \dots m][0 \dots M]$

- 1 Sample the Bézier curve to generate points on curve $p[0 \dots N]$; // sample by uniform t for $N+1$ points and N is given by user
- 2 Calculate the arc length $s[0 \dots N]$; // use Eq.(2) to calculate and store arc length cumulatively
- 3 Compute appropriate number of subcurve M ; // satisfied by Lemma 1
- 4 Construct the sequence of vertices $v[0 \dots M]$; // uniformly arc length sample points by using $s[0 \dots N]$
- 5 Calculate and round inner angle $a[1 \dots M-1]$;
- 6 Initial $Q[0 \dots M-1, 0 \dots M]$; // construct the incidence matrix
- 7 $k \leftarrow 0$;
- 8 for $j \leftarrow 0$ to M do
- 9 | $c[k][j] = Q[0][j] \vee Q[1][j]$;
- 10 end
- 11 for $i \leftarrow 2$ to $M-1$ do
- 12 | if $a[i] = a[i-1]$ then
- 13 | | for $j \leftarrow 0$ to M do
- 14 | | | $c[k][j] = c[k][j] \vee Q[i][j]$;
- 15 | | end
- 16 | | else
- 17 | | | $k \leftarrow k+1$;
- 18 | | | for $j \leftarrow 0$ to M do
- 19 | | | | $c[k][j] = Q[i][j]$;
- 20 | | | end
- 21 | | end
- 22 end
- 23 return $c[0 \dots m][0 \dots M]$;

2. Examples

For the convenience of estimation, the algorithm can be implemented with screen coordinate. Considering an initial number of segments N , all distances among adjacent points must be less than a point pixel

to generate the continuous arc length parameter s . The maximum distance between two adjacent points is actually the endpoints of the curve so we must approximate p_1 which is next to p_0 (when p_0 is the first control point \mathbf{b}_0).

Suppose that t_1 is the corresponding parameter p_1 , satisfied $p_1 \approx \mathbf{B}(t_1)$. Then, it is obtained that

$$N = \frac{1}{t_1}.$$

By Newton Raphson's method [18], the parameter t_1 can be approximated by

$$t_1 = t_0 - \frac{(\mathbf{B}(t_0) - p_1)\mathbf{B}'(t_0)}{[(\mathbf{B}(t_0) - p_1)\mathbf{B}'(t_0)]'}, \quad (11)$$

where t_0 is an initial condition given by the user.

Evaluating the similarity between the given original curve and the calculated curve, both curves are scan converted into two sets of points. After performing scan conversion of a given Bézier curve, such a curve is represented by a sequence of points. The maximum distance between two adjacent points must be less than or equal to one pixel on the screen coordinate. Then, all points are rounded and combined all duplicate points. Employing the scan conversion for an approximated curve, Bresenham's circle algorithm [6] is adopted to render points on each circular arc on the Bézier path. The distance error determination of a point on the given curve and a corresponding point on the calculated curve is considered and compared in Table 1.

Example 1. Given a 5th-degree Bézier curve $\mathbf{B}(t)$ with control points $\mathbf{b}_0 = (0, 600)$, $\mathbf{b}_1 = (300, 600)$, $\mathbf{b}_2 = (300, 900)$, $\mathbf{b}_3 = (600, 900)$, $\mathbf{b}_4 = (600, 0)$, and $\mathbf{b}_5 = (900, 0)$ (see Figure 3). A sequence of circular arcs approximates $\mathbf{B}(t)$ with the given tolerance $\tau = 1$. The curve of $\mathbf{B}(t)$ can be approximated by 25 circular arcs (as shown in Figure 4). The maximum error distance is 1.41421 pixels, and the average error distance is 0.26925 pixels.

Example 2. An 8th-degree Bézier curve $\mathbf{B}(t)$ with the control points $\mathbf{b}_0 = (0, 200)$, $\mathbf{b}_1 = (50, 0)$, $\mathbf{b}_2 = (450, 250)$, $\mathbf{b}_3 = (475, 400)$, $\mathbf{b}_4 = (250, 450)$, $\mathbf{b}_5 = (25, 400)$, $\mathbf{b}_6 = (50, 250)$, $\mathbf{b}_7 = (450, 0)$, and $\mathbf{b}_8 = (500, 200)$ (see Figure 5) is considered. Circular arcs approximate $\mathbf{B}(t)$ with the given tolerance $\tau = 1$. The curve of $\mathbf{B}(t)$

Figure 3
A 5th-degree Bézier curve

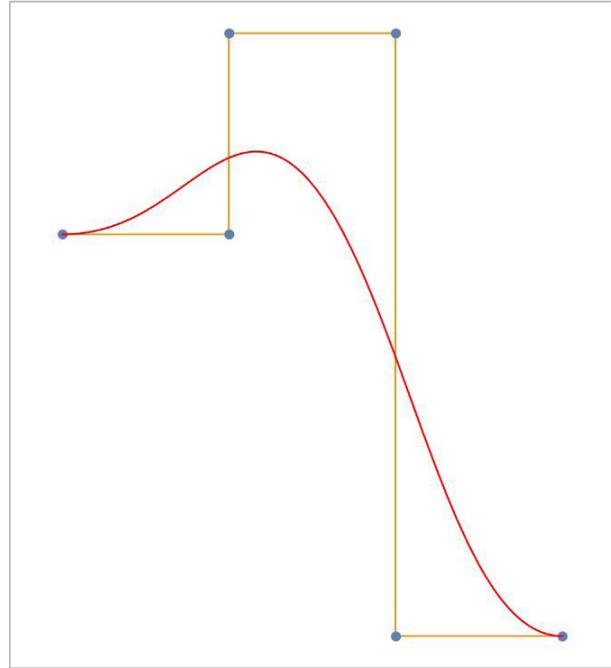
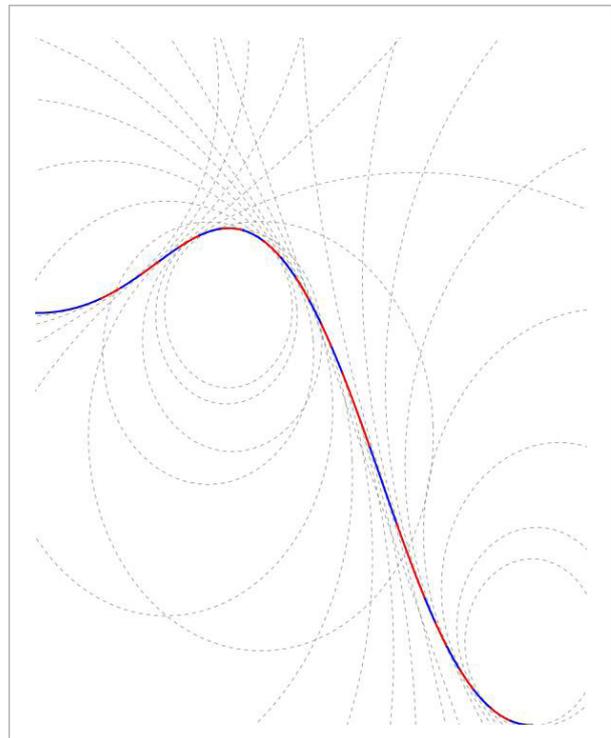


Figure 4
A circular arcs approximation curve



is approximated by 43 circular arcs (see Figure 6). The maximum error distance is 1.41421 pixels, and the average error distance is 0.170438 pixels.

Figure 5

An 8th-degree Bézier curve

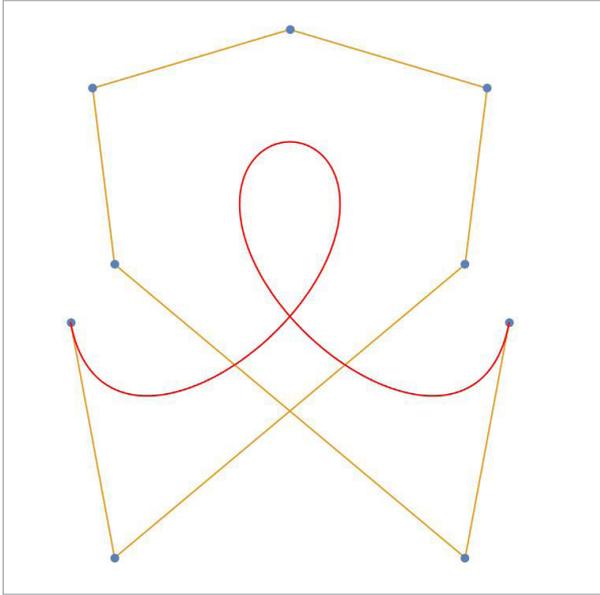
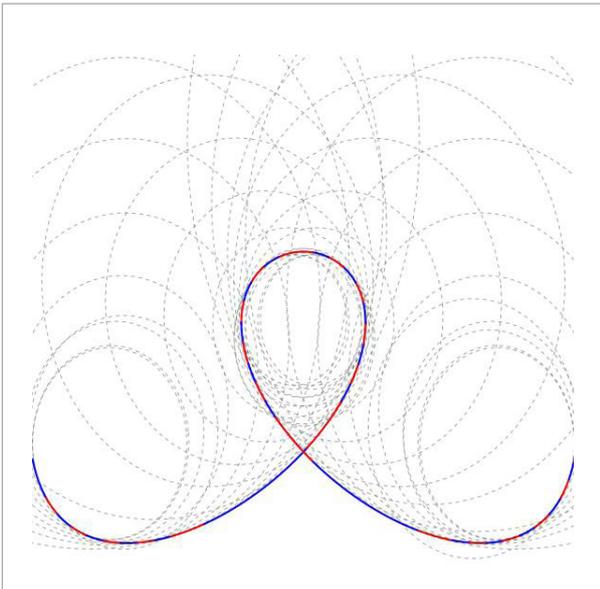


Figure 6

A circular arcs approximation curve



Example 3. A 9th-degree Bézier curve $\mathbf{B}(t)$ with the control points $\mathbf{b}_0 = (175, 107)$, $\mathbf{b}_1 = (220, 35)$, $\mathbf{b}_2 = (114, 0)$, $\mathbf{b}_3 = (27, 18)$, $\mathbf{b}_4 = (33, 126)$, $\mathbf{b}_5 = (146, 142)$,

$\mathbf{b}_6 = (205, 252)$, $\mathbf{b}_7 = (84, 168)$, $\mathbf{b}_8 = (43, 226)$, and $\mathbf{b}_9 = (58, 261)$ (see Figure 7). Circular arcs approximate $\mathbf{B}(t)$ with a given tolerance $\tau = 1$. The curve of $\mathbf{B}(t)$ is generated by 33 circular arcs (see Figure 8). The maximum error distance is 1.0 pixel, in other words, the average error distance is 0.235443 pixels.

Figure 7

A 5th-degree Bézier curve

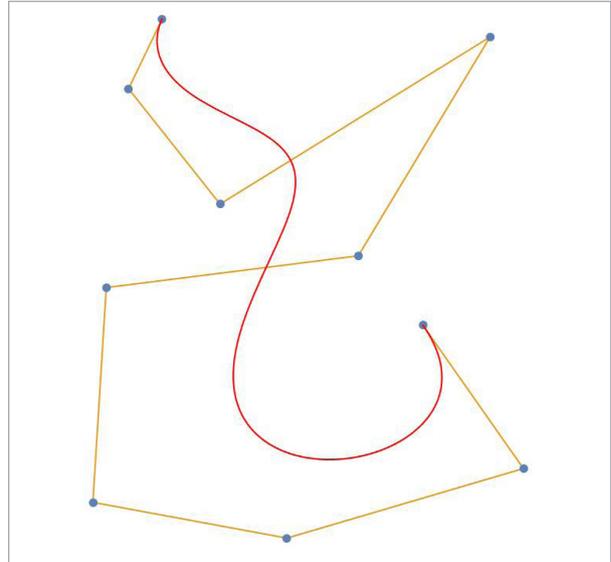
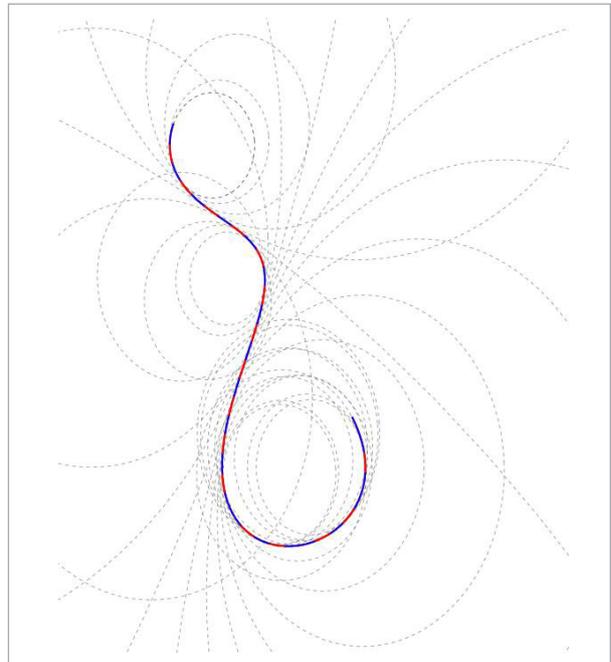


Figure 8

A circular arcs approximation curve



Example 4. When considering an 14th-degree Bézier curve, $\mathbf{B}(t)$ with random control points $\mathbf{b}_0 = (0, 0)$, $\mathbf{b}_1 = (100, 100)$, $\mathbf{b}_2 = (300, 100)$, $\mathbf{b}_3 = (310, 200)$, $\mathbf{b}_4 = (110, 250)$, $\mathbf{b}_5 = (60, 450)$, $\mathbf{b}_6 = (160, 500)$, $\mathbf{b}_7 = (400, 500)$, $\mathbf{b}_8 = (640, 500)$, $\mathbf{b}_9 = (740, 450)$, $\mathbf{b}_{10} = (690, 250)$, $\mathbf{b}_{11} = (490, 200)$, $\mathbf{b}_{12} = (500, 100)$, $\mathbf{b}_{13} = (700, 100)$, and $\mathbf{b}_{14} = (800, 0)$ (see Figure 9), this example will compute circular arcs that approximates $\mathbf{B}(t)$ with the given tolerance $\tau = 1$. The curve of $\mathbf{B}(t)$ is approximated by 47 circular arcs (see Figure 10). The maximum error distance is 1.41421 pixels, and the average error distance is 0.302386 pixels.

When considering the circular arc approximation for Bézier curve with screen coordinate, a major error is found after the rounding process. For the convenience of clustering subcurves, all inner angles a_i are rounded to the nearest multiple of a given value. In addition, the scan conversion process can also increase the error from rounding.

If Bézier curve is approximated by circular arcs, then there may be some unpredicted constraints, for example, the degree of the Bézier curve, the inflection point investigation, and the resolution of the curve. To improve the approximation, the analytic geometry of a circle is applied by classifying the regular polygon segment on each set of subcurves.

Figure 9
An 14th-degree Bézier curve

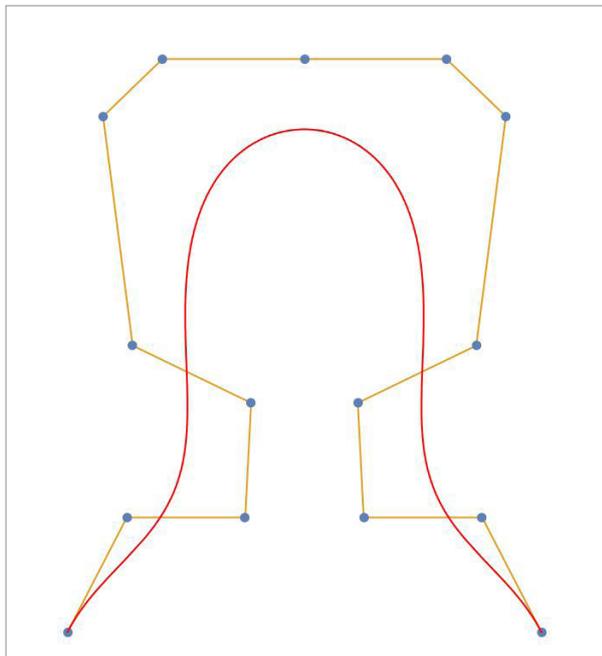


Figure 10
A circular arcs approximation curve

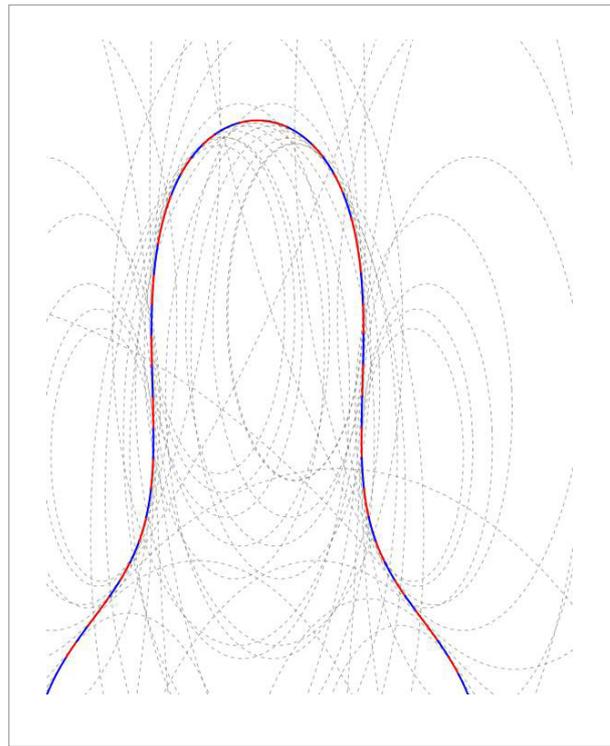


Table 1 shows the maximum and mean errors of the circular arcs approximation using the proposed method. Accordingly, Bézier curves can be approximated with in a user-specified tolerance band.

Table 1
The maximum and mean errors of the circular arc approximation via our proposed algorithm

Type of Curve	Number of Segments	Maximum Error	Average Error
Degree-5	25	1.41421	0.269250
Degree-8	43	1.41421	0.170438
Degree-9	33	1.00000	0.235443
Degree-14	47	1.41421	0.302386

The determination of a Bézier curve can be represented by a sequence of circular arcs on a raster graphic display. Applying analytic geometry of a circle, a sequence of Bézier subcurves, which can be inscribed by a regular polygon, can be classified as a circular arc. The relationship among vertices and

edges can be shown as an incidence matrix. It can be transformed into a simpler form by row elimination with the condition that the inner angles are equal. Hence, a sequence of circular arcs is simplified. As a result, each row of the final transformation matrix represents a circular arc of a given Bézier curve. Ultimately, a given Bézier curve can be expressed in terms of these circular arcs.

4. Conclusions

This paper presents a simple and elegant algorithm used for converting an arbitrary degree of Bézier curve into a collection of consecutive circular arcs.

References

1. Ahn, Y. J., Kim, H. O., Lee, K. Y. G1 Arc Spline Approximation of Quadratic Bézier Curves. *Computer-Aided Design*, 1998, 30(6), 615-620. [https://doi.org/10.1016/S0010-4485\(98\)00016-5](https://doi.org/10.1016/S0010-4485(98)00016-5)
2. Borisov, O. I., Gromov, V. S., Kolyubin, S. A., Pyrkin, A. A., Bobtsov, A. A., Salikhov, V. I., Klyunin, A. O., Petranevsky, I. V. Human-Free Robotic Automation of Industrial Operation. *IECON 2016, 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, 6867-6872. <https://doi.org/10.1109/IECON.2016.7793922>
3. Borisov, O. I., Gromov, V. S., Pyrkin, A. A., Petranevsky, I. V., Klyunin, A. O., Bobtsov, A. A. Arc Approximation Algorithm of Spatial Movements for Industrial Robots. *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, 3429-3434. <https://doi.org/10.1109/IECON.2017.8216581>
4. Farin, G. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, 2002.
5. Floater, M. S., Rasmussen, A. F. Point-Based Method for Estimating the Length of a Parametric Curve. *Journal of Computational and Applied Mathematics*, 2006, 196, 512-522. <https://doi.org/10.1016/j.cam.2005.10.001>
6. Foley, J. D., Dam, A. V., Feiner, S. K. *Computer Graphics, Principles and Practice*, Second Edition in C. Pearson Education Inc., 1996.
7. Hayashi, T., Kiyono, T., Abe, K., Onai, R. Retrieval of 2D Vector Images by Matching Weighted Feature Points. *IEEE International Conference on Image Processing*, 2008, 961-964. <https://doi.org/10.1109/ICIP.2008.4711916>
8. Maier, G. Optimal Arc Spline Approximation. *Computer Aided Geometric Design* 31, 2014, 211-226. <https://doi.org/10.1016/j.cagd.2014.02.011>
9. Kaewsaiha, P., Dejdumrong, N. An Approach to Describe Parametric Curves Using Hough-Based Arc Spline Approximation. *11th International Conference on Computer Graphics, Imaging and Visualization*, 2014, 5-10. <https://doi.org/10.1109/CGiV.2014.14>
10. Meek, D. S., Walton, D. J. Approximating Quadratics NURBS Curves by Arc Splines. *Computer-Aided Design*, 1993, 25, 371-376. [https://doi.org/10.1016/0010-4485\(93\)90032-J](https://doi.org/10.1016/0010-4485(93)90032-J)
11. Meek, D. S., Walton, D. J. Approximating Smooth Planar Curves by Arc Splines. *Journal of Computational and Applied Mathematics*, 1995, 59, 221-231. [https://doi.org/10.1016/0377-0427\(94\)00029-Z](https://doi.org/10.1016/0377-0427(94)00029-Z)
12. Meek, D. S., Walton, D. J. Approximation of a Planar Cubic Bézier Spiral by Circular Arcs. *Journal of Computational and Applied Mathematics*, 1996, 75, 47-56. [https://doi.org/10.1016/S0377-0427\(96\)00056-8](https://doi.org/10.1016/S0377-0427(96)00056-8)
13. Nuntawisuttiwong, T., Dejdumrong, N. An Approach to Bézier Curve Approximation by Circular Arcs. *15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2018. <https://doi.org/10.1109/JCSSE.2018.8457180>

14. Piegl, L. Curve Fitting Algorithm for Rough Cutting. *Computer-Aided Design*, 1986, 18(2), 79-82. [https://doi.org/10.1016/0010-4485\(86\)90154-5](https://doi.org/10.1016/0010-4485(86)90154-5)
15. Rueda, S., Udupa, J. K., Bai, L. Local Curvature Scale: A New Concept of Shape Description. *Medical Imaging 2008: Image Processing*, 2008, 6914. <https://doi.org/10.1117/12.770444>
16. Riskus, A. Approximation of a Cubic Bézier Curve by Circular Arc and Vice Versa. *Informaiton Technology and Control*, 2006, 35(4), 371-378.
17. Riskus, A., Liutkus, G. An Improved Algorithm for the Approximation of a Cubic Bézier and Its Application for Approximating Quadratic Bézier Curve. *Informaiton Technology and Control*, 2013, 42(4), 303-308. <https://doi.org/10.5755/j01.itc.42.4.1707>
18. Schneider, P. J. Phoenix: An Interactive Curve Design System Based on the Automatic Fitting of Hand-Sketched Curves. Master's thesis, University of Washington, 1998.
19. Verma G., Weber M. *Solidworks 2020 Black Book (Colored)*, CAD/CAM/CAE WORKS, 2019.
20. Walton, D. J., Meek, D. S. Approximation of Quadratic Bézier Curve by Arc Splines. *Journal of Computational and Applied Mathematics*, 1994, 54, 107-120. [https://doi.org/10.1016/0377-0427\(94\)90398-0](https://doi.org/10.1016/0377-0427(94)90398-0)
21. Yonga, J. H., Hua, S. M., Sun, J. G. Bisection Algorithms for Approximating Quadratic Bézier Curves by G1 Arc Splines. *Computer-Aided Design*, 2000, 32, 253-260. [https://doi.org/10.1016/S0010-4485\(99\)00100-1](https://doi.org/10.1016/S0010-4485(99)00100-1)
22. Zygmunt, M. Circular Arc Approximation Using Polygons. *Journal of Computational and Applied Mathematics* 322, 2017, 81-85. <https://doi.org/10.1016/j.cam.2017.03.030>

