Research Article

The Approximation of Generalized Log-Aesthetic Curves with $G^2$ Cubic Trigonometric Bézier Function

Diya’ J. Albayari,1,2 R. U. Gobithaasan3,4 and Kenjiro T. Miura5

1Mathematics Department, Arab Evangelical Episcopal Church Council (Ahliyyah & Mutran), Amman 11181, Jordan
2Mathematics Department, Ibn Rushd National Academy, Amman 17126, Jordan
3Special Interest Group on Modelling & Data Analytics, Faculty of Ocean Engineering Technology and Informatics, University Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia
4School of Mathematics Sciences, Universiti Sains Malaysia, Gelugor, Naka-Ku, Hamamatsu, Shizuoka 432-8561, Japan

Correspondence should be addressed to R. U. Gobithaasan; gobithaasan@gmail.com

Received 16 December 2022; Revised 2 May 2023; Accepted 26 July 2023; Published 7 September 2023

One of the requirements of curves in computer-aided design (CAD) is a curve with monotonic curvature profiles. Generalized log-aesthetic curves (GLACs) comprise a family of aesthetic curves which possesses a monotonic curvature profile. However, we cannot directly implement GLAC in CAD systems since it is in the form of a transcendental form. In this paper, we used cubic trigonometric Bézier (T-Bézier) curves with two shape parameters to approximate GLAC with $G^2$ continuity. The final approximation formula inherits the shape parameters of GLAC whereas T-Béziers’ shape parameters are utilized to satisfy $G^2$ constraints. Numerical results indicate that the proposed algorithm is capable of approximating GLAC within the given tolerance in (at least) two iterations.

1. Introduction

In 2010, Gobithaasan and Miura formulated the generalized log-aesthetic curves (GLACs) [1] by extending the formulation of generalized Cornu spiral (GCS) [2], similar to log-aesthetic curves (LACs) [3]. LAC is a curve obtained by a curve synthesis process using linear representation of logarithmic curvature graph (LCG) which involves the third derivative. Due to its capabilities capturing the shape of natural spirals, LCG has been used as shape interrogation tools [4]. The main characteristic of GLAC is that it has an extra curvature shift parameter ($\rho$), which increases the flexibility of the curve. In general, the curvature of GLAC is always monotonic and, hence, it produces fair curves where the family is composed of clothoid, logarithmic spiral, circle involute, Neilsen’s curve, and GCS.

There are two types of GLAC, the first one is called $\rho$-shift GLAC which can be constructed by manipulating the radius of curvature of LAC as follows [1]:

$$\kappa_{\rho-\text{GLAC}}(s) = \begin{cases} 
\frac{1}{e^{\Lambda s} + \nu}, & \alpha = 0, \\
\frac{1}{(\Lambda \alpha s + 1)^{1/\alpha} + \nu}, & \text{otherwise.}
\end{cases}$$

Similarly, the second type of GLAC is called $\kappa$-shift GLAC which can be constructed by manipulating the radius of curvature of LAC as follows [1]:

$$\kappa_{\kappa-\text{GLAC}}(s) = \begin{cases} 
\frac{1}{e^{-\Lambda s} + \nu}, & \alpha = 0, \\
(\Lambda \alpha s + 1)^{-1/\alpha} + \nu, & \text{otherwise.}
\end{cases}$$
Thus, the directional angle of GLAC can be represented as follows:
\[
\theta_{\text{GLAC}}(s) = \theta_0 + \int_0^s \kappa_{\text{GLAC}}(u) \, du,
\]  
and finally, the parametric form of GLAC is as follows:
\[
\mathbf{C}_{\text{GLAC}}(s) = \mathbf{P}_0 + \left( \int_0^s \cos(\theta_{\text{GLAC}}(u)) \, du \right) \mathbf{N}_0 + \left( \int_0^s \sin(\theta_{\text{GLAC}}(u)) \, du \right) \mathbf{N}_1,
\]  
where \( \mathbf{P}_0 \) is the start point, \( \theta_0 \) is the angle between the tangent at the start point measured anticlockwise from the \( x \)-axis, and \( \{\Lambda, \alpha, \nu\} \in \mathbb{R} \) are variables related to the shape of the GLAC segment [1]. Fair curves are essential in CAD/CAM environment. However, we cannot implement GLAC in the CAD system directly due to its transcendental form. Thus, one way to implement this curve for design feat is utilizing approximation to traditional curves, for example, quintic Bézier curves with trigonometric coordinate functions. In 1964, Schoenberg first introduced trigonometric B-splines as a function in space [6]. In 2003, Han introduced piecewise quadratic trigonometric B-splines as a function in space [6]. In 2004, he further proposed cubic trigonometric polynomial curves with \( C^1 \) continuity analogous to the quadratic B-spline curves [7], and in 2004, he further proposed cubic trigonometric polynomial curves with a shape parameter [8]. A class of quartic trigonometric polynomial curves with a shape parameter was presented by Han in 2011 [9].

Similarly, there were attempts to represent traditional Bézier curves with trigonometric coordinate functions. In 1997, Walz introduced trigonometric B-splines as a function in space [6]. In 2003, Han introduced piecewise quadratic trigonometric polynomial curves with \( C^1 \) continuity analogous to the quadratic B-spline curves [7], and in 2004, he further proposed cubic trigonometric polynomial curves with a shape parameter [8]. A class of quartic trigonometric polynomial curves with a shape parameter was presented by Han in 2011 [9].

There are numerous works on the representation of splines with trigonometric functions. In 1964, Schoenberg first introduced trigonometric B-splines as a function in space [6]. In 2003, Han introduced piecewise quadratic trigonometric polynomial curves with \( C^1 \) continuity analogous to the quadratic B-spline curves [7], and in 2004, he further proposed cubic trigonometric polynomial curves with a shape parameter [8]. A class of quartic trigonometric polynomial curves with a shape parameter was presented by Han in 2011 [9].

Similarly, there were attempts to represent traditional Bézier curves with trigonometric coordinate functions. In 1997, Walz introduced trigonometric Lagrange and Berstein polynomials to form Stancu polynomials [10]. A notable work of Han et al. is a cubic trigonometric Bézier (T-Bézier) curve with two shape parameters similar to traditional Bézier curves which were published in 2009 [9]. The advantage of T-Bézier over traditional Bézier curves is twofold: firstly, it can precisely represent circular arcs, cylinders, cones, tori, etc. Thus, CAD implementation becomes an easy task. Secondly, T-Bézier curves are located closer to the control polygon as compared to traditional Bézier curves which are of use during the designing process. The applications of trigonometric splines were the motivation for introducing various types of the trigonometric spline with features suitable for CAD/CAM applications. Readers are referred to [11–16] for further advantages of trigonometric splines in geometric modelling.

It is essential to consider the geometric constraints at the endpoints for developing splines which possess local control. For example, creating a piecewise curve which satisfies second-order geometric continuity \( G^2 \) involves matching the endpoints, its tangent vectors, and curvatures [15, 17–20]. In this paper, we approximate GLAC to cubic T-Bézier curves with \( G^2 \) conditions and derived a stable solution region so that GLAC can be approximated efficiently and directly implemented for the CAD environment.

The rest of the paper is arranged as follows. The next section introduces cubic T-Bézier curves. Section 3 elaborates the formulation to approximate GLAC with cubic T-Bézier curves. Section 4 elaborates on the curvature error measure, and it is followed by Section 5 detailing the solution region of the approximation before laying out the approximation algorithm. Finally, numerical examples are depicted before concluding remarks.

2. Preliminaries

For a given four control points \( P_i \) (\( i = 0, 1, 2, 3 \)) in \( \mathbb{R}^2 \) or \( \mathbb{R}^3 \), Han et al. presented a cubic trigonometric Bézier curve with two real-valued shape parameters \( \lambda, \mu \in [-2, 1] \) as follows [11):
\[
r(t) = \sum_{i=0}^{3} P_i b_i, \quad t \in [0, 1],
\]
where
\[
b_0(t) = \left( 1 - \sin \frac{\pi t}{2} \right) \left( 1 - \lambda \sin \frac{\pi}{2} t \right),
\]
\[
b_1(t) = \sin \frac{\pi}{2} t \left( 1 - \sin \frac{\pi}{2} t \right) \left( 2 + \lambda - \lambda \sin \frac{\pi}{2} t \right),
\]
\[
b_2(t) = \cos \frac{\pi}{2} t \left( 1 - \cos \frac{\pi}{2} t \right) \left( 2 + \mu - \mu \cos \frac{\pi}{2} t \right),
\]
\[
b_3(t) = \left( 1 - \cos \frac{\pi}{2} t \right) \left( 1 - \mu \cos \frac{\pi}{2} t \right),
\]
are the basis functions for the cubic T-Bézier curve whenever \( \lambda, \mu \neq 0 \). To note, when \( \lambda, \mu \in [-2, 1] \), essential properties of traditional Bézier curves are preserved, e.g., non-negativity, partition of unity, monotonicity, and symmetry as proven in [11].

The derivatives of \( r(t) \) at the endpoints are given as follows:
\[
\begin{align*}
r(0) &= P_0, \\
r'(0) &= \frac{\pi}{2} (\lambda + 2) (P_1 - P_0), \\
r''(0) &= \frac{\pi^2}{2} (2\lambda + 1) P_0 - 2 (\lambda + 1) P_1 + P_2, \\
r(1) &= P_3, \\
r'(1) &= \frac{\pi}{2} (\mu + 2) (P_3 - P_2), \\
r''(1) &= \frac{\pi^2}{2} (P_1 - 2 (\mu + 1) P_2 + (2 \mu + 1) P_3).
\end{align*}
\]

3. The Approximation Method

Let the GLAC be in a normalized form defined over \( s \in [0, S] \). We can express a GLAC segment in normalized form by translating it to the origin, then rotate it such that its
initial tangent point resides along the positive \( x \)-axis \( (\theta_0 = 0) \) and finally scale it to make the arc length \( s = 1 \). The rest of this paper employs \( \kappa \)-shift GLAC for cubic T-Bézier approximation. To note the derivation of \( \rho \)-shift approximation [21] is straightforward and similar to the steps shown as follows.

Consider the \( i \)th derivative of \( C_{\text{GLAC}}(s) \) w.r.t. \( s \) evaluated at \( s = a \) denoted by \( D_i^a \), then to approximate GLAC segment using \( G^2 \) cubic T-Bézier curve, we should satisfy the set of equations in equation (9) at the start and endpoints with the shape parameters \( \xi_1 \) and \( \xi_2 \) used at the start point and the shape parameters \( \omega_1 \) and \( \omega_2 \) used at the endpoint.

At the start point, we have the following set of equations:

\[
P_0 = D_0^p, \quad \frac{\pi}{2} (\lambda + 2)(P_1 - P_0) = \xi_1 D_0^1, \quad \frac{\pi^2}{2} ((2\lambda + 1)P_0 - 2(\lambda + 1)P_1 + P_2) = \xi_2 D_0^1 + \xi_1 D_0^2.
\]

After algebraic simplifications, we get the control points at the start point as follows:

\[
P_0 = D_0^p, \quad P_1 = \frac{(2 + \lambda)\pi D_0^0 + 2\xi_1 D_0^1}{(2 + \lambda)\pi}, \quad P_2 = \frac{(2 + \lambda)\pi^2 D_0^0 + 4(1 + \lambda)\pi D_0^0 + 2(2 + \lambda)\xi_2 D_0^1 + 2(2 + \lambda)\xi_1 D_0^2}{(2 + \lambda)\pi^2}.
\]

Similarly, at the endpoint, we have the following set of equations:

\[C_{\text{GLAC}}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} C_{\text{GLAC}}(1),\]

\[C_{\text{GLAC}}'(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} C_{\text{GLAC}}'(1) = \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix},\]

\[C_{\text{GLAC}}''(0) = \begin{pmatrix} 0 \\ (1 + \nu) \end{pmatrix} C_{\text{GLAC}}''(1) = (\Lambda\alpha + 1)^{-1/\alpha + \gamma} \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix},\]

\[C_{\text{GLAC}}''(0) = \begin{pmatrix} 0 \\ (1 + \nu) \end{pmatrix} C_{\text{GLAC}}''(1) = (\Lambda\alpha + 1)^{-1/\alpha + \gamma} \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}.
\]

\[\dot{x}(\tau_1)^{\prime} C_{\text{GLAC}}(a) + \dot{x}(\tau_2)^{\prime} C_{\text{GLAC}}(a),\]

where \( \prime \) is the differentiation with respect to the parameter \( s \).

To guarantee the second-order geometric continuity \( (G^2) \) between \( r(t) \) and \( C_{\text{GLAC}}(u) \) at \( a \), it is required to establish \( C_{\text{GLAC}}'(h(u)) \) parameterization equivalent to \( C_{\text{GLAC}}(u) \) satisfying the following equation [2]:

\[\frac{d^i}{dt^i} r(a) = \frac{d^i}{du^i} C_{\text{GLAC}}(a), \quad i = 0, 1, 2.\]

The accompanying sets of equations are given as follows [2]:

\[r(a) = C_{\text{GLAC}}(a),\]

\[r'(a) = \tau_1 C_{\text{GLAC}}'(a),\]

\[r''(a) = \tau_1 C_{\text{GLAC}}''(a) + \tau_1^2 C_{\text{GLAC}}''(a),\]

where \( \tau_1 \) and \( \tau_2 \) are real-valued shape factors. The second-order parametric continuity, \( C^2 \), is obtained by letting \( \tau_1 = 1 \) and \( \tau_2 = 0 \).

Let \( \kappa_\kappa(s) \) be the curvature function of \( \kappa \)-shift GLAC as stated in equation (2), \( (x, y) \) be the endpoint where \( x = x(1) \) and \( y = y(1) \), and \( \theta \) be the winding angle at the endpoint such that \( \theta = \int_0^1 \kappa(s) ds \). Then, the derivatives of GLAC at the endpoints are given as follows:

\[\frac{d^i}{dt^i} r(a) = \frac{d^i}{du^i} C_{\text{GLAC}}(a), \quad i = 0, 1, 2.\]
\[ P_3 = D_1^0, \]
and
\[ \frac{\pi}{2} (\mu + 2)(P_3 - P_2) = \omega_1 D_1^1, \]
(15)
\[ \frac{\pi^2}{2} \left( P_1 - 2(\mu + 1)P_2 + (2\mu + 1)P_3 \right) = \omega_2 D_1^1 + \omega_2^2 D_2^2, \]
\[ \frac{\pi^2}{2} \left( P_1 - 2(\mu + 1)P_2 + (2\mu + 1)P_3 \right) = \omega_2 D_1^1 + \omega_2^2 D_2^2, \]
\[ P_1 = \frac{(2 + \mu)\pi^2 D_0^0 - 4(1 + \mu)\omega_1 \pi D_1^1 + 2(2 + \mu)\omega_2 D_1^1 + 2(2 + \mu)\omega_1^2 D_2^2}{(2 + \lambda)\pi^2}, \]
(16)
\[ P_2 = \frac{(2 + \mu)\pi^2 D_0^0 - 2\omega_1 D_1^1}{(2 + \mu)\pi}, \]
(17)
\[ P_3 = D_1^0. \]
(18)

For the control points \( P_1 \) and \( P_2 \), each has two expressions at the start and endpoints. Thus, we have four equations in \( x \) and \( y \) coordinate functions to be solved for \( \lambda \) and \( \mu \), leading to the following equations:

\[ \lambda(\xi_1, \omega_1) = \frac{2\pi \xi_1 \sin \theta}{\pi^2 (x \sin \theta - y \cos \theta) - 2\left( (\Lambda \alpha + 1)^{-1/\alpha} + \nu \right) \omega_1^2} - 2, \]
(19)
\[ \mu(\xi_1, \omega_1) = \frac{2\pi \omega_1 \sin \theta}{\pi^2 y - 2(1 + \nu) \xi_1^2} - 2. \]
(20)

**Definition 1.** \( G^2 \) cubic T-Bézier approximation

A \( G^2 \) cubic trigonometric approximation curve of GLAC is defined by the following equation:

\[ \tau(t) = \sum_{i=0}^{3} P_i b_i, \quad t \in [0, 1], \xi_1, \omega_1 \in \delta, \]
(21)

where \( \delta \) is the solution region shown in Section 6, with the definitions of basis functions, \( b_i \), as stated in equation (5), \( \lambda \) and \( \mu \) as stated in equations (19) and (20). By employing equations (12), (13), (17), and (18), the control points, \( P_i \), are defined as follows:

\[
\begin{align*}
P_0 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \\
P_1 &= \begin{pmatrix} x - y \cot(\theta) - \frac{2\omega_1^2 \csc(\theta)\left( (\alpha \Lambda + 1)^{-1/\alpha} + \nu \right)}{\pi^2} \\ 0 \end{pmatrix}, \\
P_2 &= \begin{pmatrix} x - y \cot(\theta) + \frac{2(\nu + 1)\xi_1^2 \cot(\theta)}{\pi^2} \\ 0 \end{pmatrix}, \\
P_3 &= \begin{pmatrix} x \\ y \end{pmatrix}.
\end{align*}
\]
(22)
The free shape parameters for the approximated curve are $\alpha$, $\Lambda$, and $\nu$ which are inherited from GLAC. $\lambda$ and $\mu$ are calculated by searching for suitable values $\xi^*$ and $\omega^*$ satisfying $G^2$ data. Once these shape parameters are given values by the designer, then we would be able to identify their corresponding control points as stated in Definition 1.

**Definition 2.** An alternative to $G^2$ cubic T-Bézier approximation

One can use the equations (14) and (16) instead of equations (13) and (17) to define the control points $P_i$ and $P_{i+1}$

\[
\xi_2(\xi_1, \omega_1) = (v + 1)\xi^* \cos(\theta) - 2(\omega^* \csc(\theta)(\alpha \Lambda + 1)^{-1/\alpha} + \nu) + \pi (3\pi x - 4\xi_1 - 3\pi \nu \cot(\theta)),
\]

\[
\omega_2(\xi_1, \omega_1) = -\frac{1}{2} \csc(\theta)(\frac{2\omega^* \cos(\theta)(\alpha \Lambda + 1)^{-1/\alpha} + \nu)} - 4\pi \omega_1 \sin(\theta) - 4(v + 1)\xi^* \cos(\theta) + 3\nu^2 \gamma.
\]

Since both definitions provide the same results, it is recommended to use $\bar{r}(t)$ in Definition 1 instead of $\bar{r}(t)$ to avoid extra computations in calculating the values of $\xi_2$ and $\omega_2$.

**4. Curvature Error Measure**

It is known that the curvature of GLAC is a smooth monotonic function which guarantees fairness. Thus, we used the curvature error measure as proposed by Cross and Cripps which was used to approximate GCS [2] as follows:

\[
\epsilon_{\text{max}} = \max_{s \in [0,1]} \left| \frac{\kappa_s(s) - \kappa_C(s)}{\max \{\kappa_C(s), 1\}} \right|,
\]

where $\kappa_s(s)$ is the curvature function of the Bézier curve defined in the domain $[0,1]$ and $\kappa_C(s)$ is the curvature function of the GCS. The advantage of this measure is that it can accurately reflect the curvature profile and produce smooth approximation results. However, the disadvantage is that it is computationally costly due to the arc length reparameterization of the Bézier curve. Thus, to reduce the high computational cost, Albayari [22] proposed another curvature error measure as follows:

\[
\tau_{\text{max}} = \max_{t \in [0,1]} \left| \frac{\kappa_t(t) - \kappa_C(t)}{\max \{\kappa_C(t), 1\}} \right|,
\]

where $\kappa_t(t)$ is the curvature function of the Bézier curve and $\kappa_C(t)$ is the curvature function of the GCS, and to calculate the error, one has to sample a sequence of equally spaced parameter values $\{t_i\}_{i=0}^N$. Even though Lu’s curvature error measure takes less computation than Cross and Cripps’s curvature measure, we still must calculate Cross and Cripps’s error measure to exactly identify the quality of an approximation. Hence, one may end up calculating both $\epsilon_{\text{max}}$ and $\tau_{\text{max}}$ for an efficient approximation [23].

In 2016, Albayari et al. proposed a new error measure that can reduce the computations and determine the acceptability of the approximations accurately as follows [21]:

while employing the basis functions, $b_i$, as stated in equation (5), $\lambda$ and $\mu$ as stated in equations (19) and (20). Then, the $G^2$ cubic trigonometric approximation of GLAC is defined as follows:

\[
\bar{r}(t) = \sum_{i=0}^{3} P_i b_i, \quad t \in [0,1], \xi_1, \omega_1 \in \delta,
\]

where

\[
\bar{e}_{\max} = \max_{i=0,\ldots,n} \frac{|\kappa_t(t_i) - \kappa_C(s_i)|}{\max \{\kappa_C(s_i), 1\}},
\]

where $\kappa_t(t_i)$ and $\kappa_C(s_i), i = 0,\ldots,n$, are the corresponding curvatures of the curve and the approximation curve at the same arc length, respectively. Although their measure reduces the computation, and it is very accurate to the exact error but the accuracy of the results depends on the numbers of the selected nodes. Since there is no rule to identify an ideal number of nodes which satisfy the exact error for all cases, and the maximum error may occur in between some of the two nodes. Thus, Lu introduced a new error measure as follows [21]:

\[
\hat{e}_{\text{max}} = \max_{t \in [0,1]} |\kappa_t(s) - \kappa_C(s)|/\max \{\kappa_C(s), 1\},
\]

where $\kappa_t(s)$ is the curvature function of $r(s(t))$ normalized to the domain $[0,1]$ and $\kappa_C(s)$ is the curvature function of the GLAC. By using this error measure, one may obtain a parameterized arc length curvature error measure by interpolating the arc length function of GLAC to avoid unnecessary computation. The default tolerance is set to be 0.05 which guarantees a high-quality approximation as proposed by Cross and Cripps [2].

Similarly, we used the error measure proposed by Lu [21] which is tailored to approximate GLAC as follows:

\[
\hat{e}_{\text{max}} = \max_{t \in [0,1]} |\kappa_{C-GLAC}(s) - \kappa_C(s)|/\max \{\kappa_{C-GLAC}(s), 1\},
\]

where $\kappa_{C-GLAC}(s)$ is the curvature function of $\bar{r}(s(t))$ normalized to the domain $[0,1]$ and $\kappa_{C-GLAC}(s)$ is the curvature function of the GLAC which can be either $\kappa$-shift or $\rho$-shift GLAC.

**5. Solution Region**

The first step in the approximation is to set an initial value to the shape parameters $\xi_1$ and $\omega_1$. A plausible set of initial values for shape parameters are $\xi_1 = \omega_1 = 1$ which satisfies
C² continuity [2]. However, in some cases, these initial values provide unacceptable errors. Thus, it is essential to find a suitable solution region for the shape parameters ξ₁ and ω₁ to meet the tolerance in a shorter period.

Indeed, there is a geometric effect of the two shape parameters ξ₁ and ω₁ on the initial and end tangent vectors. From equations (12), (13), (17), and (18), it is evident that ξ₁ relates to the magnitude of ‖P₁ − P₀‖ and ω₁ relates to the magnitude of ‖P₃ − P₂‖. Since λ, μ > −2, the shape parameters ξ₁ and ω₁ should be positive; otherwise, the tangents will point in the opposite direction. Hence, we set the lower bound of ξ₁ and ω₁ as 0. Additionally, the values of the shape parameters should not be large otherwise the control points will be far away from each other. Since the total arc length of the curve ∫₀¹ ‖V′(s)‖ ds is approximately 1, so using the trapezoid rule, we get the following equation [2]:

\[
1 = \int_0^1 \|r'(s)\| ds \approx \frac{1}{4} \|r'(0)\| + \frac{1}{2} \|r'\left(\frac{1}{2}\right)\| + \frac{1}{4} \|r'(1)\| = \frac{1}{4} \xi_1 + \frac{1}{2} \|r'\left(\frac{1}{2}\right)\| + \frac{1}{4} \omega_1 > \frac{1}{4}(\xi_1 + \omega_1).
\]

6. The Basis Functions’ Properties of G² GLAC T-Bézier Approximated Curves

The proposed G² basis functions for all GLAC approximated curves possess various properties. The following theorems elucidate these properties. Without loss of generality, the case when a = 0 for κ-shift GLAC is discussed in Theorems 3 and 4. However, all other 3 different cases are discussed in detail in [21].

Theorem 3. The proposed G² basis functions of the G² GLAC cubic trigonometric approximated curve satisfy the partition of unity property.

Proof. After algebraic simplification of the basis functions, the approximated curve basis functions stated in equation (5) can be rewritten as follows:

\[
b_{\kappa,0}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = \frac{2\pi [\cos(\xi_1) \sin(\theta) \sin^3(\tau t/2)]}{2\omega_0^4(\cos^4(\xi_1) + \pi^2 [\cos(\theta) - x \sin(\theta)]}
\]

\[
\quad - \frac{4\pi [\cos(\xi_1) \sin(\theta) \sin^2(\tau t/2)]}{2\omega_0^4(\cos^4(\xi_1) + \pi^2 [\cos(\theta) - x \sin(\theta)]}
\]

\[
\quad + \frac{2\pi [\cos(\xi_1) \sin(\theta) \sin(\tau t/2)]}{2\omega_0^4(\cos^4(\xi_1) + \pi^2 [\cos(\theta) - x \sin(\theta)]}
\]

\[
b_{\kappa,1}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = \frac{2\pi [\cos(\xi_1) \sin(\theta) \sin^3(\tau t/2)]}{2\omega_0^4(\cos^4(\xi_1) + \pi^2 [\cos(\theta) - x \sin(\theta)]}
\]

\[
\quad - \frac{4\pi [\cos(\xi_1) \sin(\theta) \sin^2(\tau t/2)]}{2\omega_0^4(\cos^4(\xi_1) + \pi^2 [\cos(\theta) - x \sin(\theta)]}
\]

\[
\quad + \frac{2\pi [\cos(\xi_1) \sin(\theta) \sin(\tau t/2)]}{2\omega_0^4(\cos^4(\xi_1) + \pi^2 [\cos(\theta) - x \sin(\theta)]}
\]

\[
\quad - 2\sin^3\left(\frac{\tau t}{2}\right) + 2\sin^3\left(\frac{\tau t}{2}\right) + 1,
\]

\[
\quad - 2\sin^3\left(\frac{\tau t}{2}\right) + 2\sin^3\left(\frac{\tau t}{2}\right) + 1.
\]
\[
b_{k,2}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = \frac{2\pi \omega_1 \sin(\theta) \cos^3(\pi t/2)}{\pi^2 y - 2(v + 1)\xi_1^2} - \frac{4\pi \omega_1 \sin(\theta) \cos^3(\pi t/2)}{\pi^2 y - 2(v + 1)\xi_1^2} + \frac{2\pi \omega_1 \sin(\theta) \cos(\pi t/2)}{\pi^2 y - 2(v + 1)\xi_1^2} - 2 \cos^3\left(\frac{\pi t}{2}\right) + 2 \cos^2\left(\frac{\pi t}{2}\right).
\]

(34)

\[
b_{k,3}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = \frac{-2\pi \omega_1 \sin(\theta) \cos^3(\pi t/2)}{\pi^2 y - 2(v + 1)\xi_1^2} + \frac{4\pi \omega_1 \sin(\theta) \cos^3(\pi t/2)}{\pi^2 y - 2(v + 1)\xi_1^2} - \frac{2\pi \omega_1 \sin(\theta) \cos(\pi t/2)}{\pi^2 y - 2(v + 1)\xi_1^2} + 2 \cos^3\left(\frac{\pi t}{2}\right) - 3 \cos^2\left(\frac{\pi t}{2}\right) + 1.
\]

(35)

The first four terms of \(b_{k,0}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1))\) stated in equation (32) are the negation terms of the first four terms of \(b_{k,1}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1))\) stated in equation (33). Similarly, for the first four terms of \(b_{k,2}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1))\) stated in equation (34) with the first four terms of \(b_{k,3}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1))\) stated in equation (35), the summation of the basis functions is as follows:

\[
\sum_{i=0}^{3} b_{k,i}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = -\sin^2\left(\frac{\pi t}{2}\right) + 1 - \cos^2\left(\frac{\pi t}{2}\right) + 1 = 2 - \left(\sin^2\left(\frac{\pi t}{2}\right) + \cos^2\left(\frac{\pi t}{2}\right)\right) = 1.
\]

(36)

\[
b_i(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = b_{3-i}(1 - t, \mu(\xi_1, \omega_1), \lambda(\xi_1, \omega_1)).
\]

(37)

Theorem 4. The proposed \(G^2\) basis functions of the \(G^2\) GLAC cubic trigonometric approximated curves satisfy the symmetric property.

Proof. The symmetric property for the \(G^2\) basis functions holds if the following equations hold for \(i = 0, 1, 2, 3\).

\[
b_{k,0}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = \left(\sin\left(\frac{\pi t}{2}\right) - 1\right)^2 \left(\frac{\pi [e^\lambda] \xi_1 \sin(\theta)}{2\omega_1^2 (\lceil e^\lambda \rceil y + 1) + \pi^2 [e^\lambda] (y \cos(\theta) - x \sin(\theta))} + 1\right).
\]

(38)

Similarly, the value of \(b_3(1 - t, \mu(\xi_1, \omega_1), \lambda(\xi_1, \omega_1))\) can be calculated as follows:

\[
b_{k,3}(1 - t, \mu(\xi_1, \omega_1), \lambda(\xi_1, \omega_1)) = 4 \cos^4\left(\frac{1}{4} \pi (t + 1)\right) \left(\frac{\pi [e^\lambda] \xi_1 \sin(\theta)}{2\omega_1^2 (\lceil e^\lambda \rceil y + 1) + \pi^2 [e^\lambda] (y \cos(\theta) - x \sin(\theta))} + 1\right).
\]

(39)

It is evident from equations (38) and (39) that it is sufficient to show \(4 \cos^4((1/4)\pi (t + 1)) = (\sin((1/2) \pi t) - 1)^2\) to prove that \(b_0(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = b_3(1 - t, \mu(\xi_1, \omega_1), \lambda(\xi_1, \omega_1))\).
Based on the trigonometric identities in equations (40) and (41), we can complete the proof by equation (42).

\[
\cos^2\left(\frac{\theta}{2}\right) = \frac{1}{2} (\cos(\theta) + 1), \quad (40)
\]

\[
\cos(x + y) = \cos(x) \cos(y) - \sin(x) \sin(y), \quad (41)
\]

\[
4 \cos^4\left(\frac{1}{4} \pi (t + 1)\right) = 4 \left(\cos^2\left(\frac{1/2 \pi (t + 1)}{2}\right)\right)^2 = 4 \left(\frac{1}{2} \left(\cos\left(\frac{1}{2} \pi (t + 1)\right) + 1\right)\right)^2 = \left(\cos\left(\frac{1}{2} \pi (t + 1)\right) + 1\right)^2 = \left(1 - \sin\left(\frac{1}{2} \pi t\right)\right)^2 = \left(\sin\left(\frac{1}{2} \pi t\right) - 1\right)^2. \quad (42)
\]

To prove the symmetric property for the remaining basis functions, it is required to show that \(b_1(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = b_2(1 - t, \mu(\xi_1, \omega_1), \lambda(\xi_1, \omega_1))\), which recall from equations (5), (19), and (20) that the value of \(b_1(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1))\) is as follows:

\[
b_{\varepsilon, 1}(t, \lambda(\xi_1, \omega_1), \mu(\xi_1, \omega_1)) = -\left(2 \left(\sin\left(\frac{\pi t}{2}\right) - 1\right) \sin\left(\frac{\pi t}{2}\right) \right) \left(\frac{\pi [\varepsilon] \lambda_{\xi_1} \sin(\theta) (\sin(\pi t/2) - 1)}{2 \omega_1^2 ([\varepsilon] \lambda_{\nu} + 1) + \pi^2 [\varepsilon] \lambda_{\lambda} (y \cos(\theta) - x \sin(\theta))} \right) \sin(\pi t/2) \left(2 \omega_1^2 ([\varepsilon] \lambda_{\nu} + 1) + \pi^2 [\varepsilon] \lambda_{\lambda} (y \cos(\theta) - x \sin(\theta)) \right) \right), \quad (43)
\]

where the value of \(b_{\varepsilon, 2}(1 - t, \mu(\xi_1, \omega_1), \lambda(\xi_1, \omega_1))\) can be calculated as follows:

\[
b_{\varepsilon, 2}(1 - t, \mu(\xi_1, \omega_1), \lambda(\xi_1, \omega_1)) = \left(2 \left(1 - \sin\left(\frac{\pi t}{2}\right) \right) \times \sin\left(\frac{\pi t}{2}\right) \right) \left(\frac{\pi [\varepsilon] \lambda_{\xi_1} \sin(\theta) (\sin(\pi t/2) - 1)}{2 \omega_1^2 ([\varepsilon] \lambda_{\nu} + 1) + \pi^2 [\varepsilon] \lambda_{\lambda} (y \cos(\theta) - x \sin(\theta))} \right) \sin(\pi t/2) \left(2 \omega_1^2 ([\varepsilon] \lambda_{\nu} + 1) + \pi^2 [\varepsilon] \lambda_{\lambda} (y \cos(\theta) - x \sin(\theta)) \right) \right). \quad (44)
\]
Multiplying the constant $-1$ by the first parenthesis of equation (43) makes it identically equal to equation (44). Therefore, the symmetric property holds.

**Theorem 5.** The proposed $G^2$ basis functions are non-negative functions.

\[
\begin{align*}
  b_0(t, \lambda(\xi_1, \omega_1)) &= \left(1 - \sin \frac{\pi}{2} t\right)^2 \left(1 - \lambda(\xi_1, \omega_1) \sin \frac{\pi}{2} t\right), \\
  b_1(t, \lambda(\xi_1, \omega_1)) &= \sin \frac{\pi}{2} t \left(1 - \sin \frac{\pi}{2} t\right) \left(2 + \lambda(\xi_1, \omega_1) - \lambda(\xi_1, \omega_1) \sin \frac{\pi}{2} t\right), \\
  b_2(t, \mu(\xi_1, \omega_1)) &= \cos \frac{\pi}{2} t \left(1 - \cos \frac{\pi}{2} t\right) \left(2 + \mu(\xi_1, \omega_1) - \mu(\xi_1, \omega_1) \cos \frac{\pi}{2} t\right), \\
  b_3(t, \mu(\xi_1, \omega_1)) &= \left(1 - \cos \frac{\pi}{2} t\right)^2 \left(1 - \mu(\xi_1, \omega_1) \cos \frac{\pi}{2} t\right).
\end{align*}
\]

**Proof.** The proposed $G^2$ basis functions are constructed by substituting the proposed $G^2$ formulas of the shape parameters $\lambda$ and $\mu$ in Han’s basis functions \[7\] stated in equation (5) as follows:

\[
\begin{align*}
  b_0(\lambda(\xi_1, \omega_1)) &= -\left(\sin \frac{\pi t}{2} - 1\right)^2 \sin \frac{\pi t}{2}, \\
  b_1(\lambda(\xi_1, \omega_1)) &= \left(\sin \frac{\pi t}{2} - 1\right)^2 \sin \frac{\pi t}{2}, \\
  b_2(\mu(\xi_1, \omega_1)) &= 4 \sin^2 \left(\frac{\pi t}{4}\right) \cos \frac{\pi t}{2}, \\
  b_3(\mu(\xi_1, \omega_1)) &= -4 \sin^2 \left(\frac{\pi t}{4}\right) \cos \frac{\pi t}{2}.
\end{align*}
\]

**Theorem 6.** For a given parameter $t$, the proposed $G^2$ basis functions preserve monotonicity property for the corresponding shape parameters $\lambda(\xi_1, \omega_1)$ and $\mu(\xi_1, \omega_1)$ as follows.

1. $b_0(t, \lambda(\xi_1, \omega_1))$ monotonically decreasing for the corresponding shape parameter $\lambda(\xi_1, \omega_1)$.
2. $b_1(t, \lambda(\xi_1, \omega_1))$ monotonically increasing for the corresponding shape parameter $\lambda(\xi_1, \omega_1)$.
3. $b_2(t, \mu(\xi_1, \omega_1))$ monotonically increasing for the corresponding shape parameter $\mu(\xi_1, \omega_1)$.
4. $b_3(t, \mu(\xi_1, \omega_1))$ monotonically decreasing for the corresponding shape parameter $\mu(\xi_1, \omega_1)$.

**Proof.** This can be proved by considering the sign of the first derivatives for the basis functions over the entire domain $t \in [0, 1]$.

From equations in equation (45), the first derivatives of the basis functions with respect to the corresponding shape parameters can be calculated as follows:

\[
\begin{align*}
  b'_0(\lambda(\xi_1, \omega_1)) &= -\left(\sin \frac{\pi t}{2} - 1\right)^2 \sin \frac{\pi t}{2}, \\
  b'_1(\lambda(\xi_1, \omega_1)) &= \left(\sin \frac{\pi t}{2} - 1\right)^2 \sin \frac{\pi t}{2}, \\
  b'_2(\mu(\xi_1, \omega_1)) &= 4 \sin^2 \left(\frac{\pi t}{4}\right) \cos \frac{\pi t}{2}, \\
  b'_3(\mu(\xi_1, \omega_1)) &= -4 \sin^2 \left(\frac{\pi t}{4}\right) \cos \frac{\pi t}{2}.
\end{align*}
\]

The derivatives’ functions $b'_0(\lambda(\xi_1, \omega_1))$ and $b'_2(\mu(\xi_1, \omega_1))$ are negative over the entire domain $t \in [0, 1]$, where the derivatives’ function $b'_1(\lambda(\xi_1, \omega_1))$ and $b'_3(\mu(\xi_1, \omega_1))$ are positive over the entire domain $t \in [0, 1]$ which proves the monotonicity of the proposed basis functions, as mentioned in Theorem 6.

To prove that the approximation curve $r(t)$ is invariant under translation, consider the translation $\mathbf{m} = (x^*, y^*)$, then
For the approximation curve to be invariant, the Cauchy's relation \( \sum_{i=0}^{3} b_i(t) = 1 \), must be satisfied [24], which is proved in Theorem 3.

7. Approximation Algorithm

This section illustrates the overall algorithm to approximate GLAC with \( C^2 \) cubic T–Bézier as shown in Algorithm 1. The numerical method used to search for suitable \( \xi_1 \) and \( \omega_1 \) in the algorithm is the interior point method utilizing the FindMinimum [] command found in Mathematica. Based on numerical results discussed in the next section, the values of \( \xi_1 \) and \( \omega_1 \) is always around (1, 1). This is because the length of normalized GLAC where the parameter \( s \) is equal to 1.

\[
\begin{align*}
\mathbf{r}^* (t) &= \left[ b_0(t) \ b_1(t) \ b_2(t) \ b_3(t) \right] \left[ \begin{array}{c}
P_0^* \\
P_1^* \\
P_2^* \\
P_3^*
\end{array} \right] \\
&= \left\{ \left( \sin \frac{\pi}{2} \left( 1 - \sin \frac{\pi}{2} t \right) \right) \left( 2 + \frac{2\pi \xi_1 \sin \theta}{\pi^2 (x \sin \theta - y \cos \theta - 2((\Lambda \alpha + 1)^{-1/\alpha} + y)\omega_1^2 - 2)} \sin \frac{\pi}{2} t \right) \right. \\
&\quad - \left( \sin \frac{\pi}{2} t \right) \left( 2 + \frac{2\pi \xi_1 \sin \theta}{\pi^2 (x \sin \theta - y \cos \theta - 2((\Lambda \alpha + 1)^{-1/\alpha} + y)\omega_1^2 - 2)} \sin \frac{\pi}{2} t \right) \left( 1 - \cos \frac{\pi}{2} t \right)^2 \left( 1 - \frac{2\pi \omega_1 \sin \theta}{\pi^2 y - 2(1 + y)\xi_1^2} - 2 \cos \frac{\pi}{2} t \right) \left( \frac{2}{\pi^2} + y^* \right) \\
&\quad + \left. \left( \frac{2(\theta + 1)\xi_1^2 \cos \theta}{\pi^2} + x^* \right) \right\} \\
&= \mathbf{r}(t) + \left( b_0(t) + b_1(t) + b_2(t) + b_3(t) \right) (x^*, y^*) = \mathbf{r}(t) + \left( \sum_{i=0}^{3} b_i(t) \right) \mathbf{m}.
\end{align*}
\]

8. Numerical Examples

As stated in the previous section, we may generate LAC with the formulation of GLAC by letting \( y = 0 \) [1]. Table 1 depicts various types of GLAC segments (including LAC) with different values of shape parameters to show the efficiency of the proposed method in approximating the GLAC family with cubic T–Béziers.

Numerous approximation examples of LACs \((y = 0)\) and GLACs are depicted in Table 1. The first 1–5 examples show the results of approximating various GLACs. The next 6–13 examples depict special cases of GLAC and LAC \((y = 0)\). Most cases terminate after the first run which indicates the efficiency of the proposed algorithm. However, two cases (examples 14 and 15) have error of the approximation which...
is large on the first attempt of approximation, where the next iteration itself minimizes the error with tol = 0.05.

Figures 1–3 are related to rows 3, 7, and 8 in Table 1, respectively. The GLAC and its $G^2$ cubic T-Bézier approximation is shown in 1(a) where the curvature error measure appears in 1(b), the graph of the dependent shape parameters $\lambda(\xi_1, \omega_1)$ and $\mu(\xi_1, \omega_1)$ over the solution region $\delta$ are shown in 1(c) and 1(d), respectively.

For the case of example 14, which shown in Figure 4 below, the error was 0.078 obtained with the initial values of $\xi_1 = \omega_1 = 1$, where their corresponding shape parameters are $\lambda = 0.587$ and $\mu = 0.15$. To reduce the number of iterations, we recommended searching around the parameter $s$ where the maximum error occurs. In this example, the maximum error of 0.078 occurs at $s = 0.5054$. By inputting this information during the search with the interior point method, we obtained the desired result on the next run itself where the error is 0.045 and the corresponding shape parameters are $\xi_1 = 0.989$ and $\omega_1 = 0.989$, which led to the calculation of cubic T-Béziers’ shape parameters as $\lambda = 0.471$ and $\mu = -0.194$.

For example 15, the error was 0.063 with the initial values of $\xi_1 = \omega_1 = 1$, and its corresponding shape parameters were $\lambda = 0.034$ and $\mu = 0.43$. The maximum error occurs 0.063 occurs at $s = 0.489$. Similar to example 14, we also obtained the desired result on the next run itself where the error is 0.025 and the corresponding shape parameters are $\xi_1 = 0.984$ and $\omega_1 = 0.988$, which led to the calculation of cubic T-Béziers’ shape parameters as $\lambda = -0.028$ and $\mu = 0.31$.

9. Comparing the Proposed T-Bézier Method with the Standard Bézier Method

In this section, we are comparing the proposed T-Bézier method with the standard Bézier method in approximating several GLAC with various arc lengths.

Table 2 shows the results of approximating several normalized $\kappa$-shift GLAC while Table 3 depicts the results of approximating numerous examples of $\kappa$-shift GLAC with several arc lengths. Table 2 shows that the cubic Bézier approximation method does not reach satisfactory approximations without the need of searching for other values of the shape parameters $(\xi_1, \omega_1)$. As a consequence, this method constantly replaces the control points of the cubic Bézier approximated curve with better positions to provide increased accuracy. However, the proposed cubic trigonometric Bézier approximation method completes most of the examples in Table 2 with the given initial values without any change in values. This, then, indicates that the proposed cubic trigonometric Bézier approximation method is more efficient than the cubic Bézier approximation method. Furthermore, although both of the cubic methods require 1 iteration to reach an acceptable approximation for GLAC $(\alpha, \lambda, \gamma) = (0.7, -0.6, 0.1)$, the proposed cubic trigonometric Bézier approximation method provides an error of 0.008955 which is more accurate when compared to the cubic Bézier approximation method’s error of 0.017444.

For aesthetic curves with various arc lengths, the cubic trigonometric Bézier approximation method needs iteration for more cases compared to the normalized curves.
Table 1: The error in approximation for various types of GLAC tested by the error measure $\hat{\epsilon}_{\max}$.

<table>
<thead>
<tr>
<th>#</th>
<th>$(\alpha, \Lambda, v)$</th>
<th>$\delta$</th>
<th>Final $(\xi_1, \omega_1)$</th>
<th>$\lambda (\xi_1, \omega_1), \mu (\xi_1, \omega_1)$</th>
<th>$\hat{\epsilon}_{\max}$</th>
<th>No of Itrs</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(0.1, -0.3, 0.4)$</td>
<td>$[[0, 1.17] \times (0, 1.36)] [([1.17, 1.47] \times (0, 1.07))]$</td>
<td>(1, 1)</td>
<td>$(0.124, -0.085)$</td>
<td>0.007</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$(0.4, -0.8, -0.3)$</td>
<td>$[[0, 1.36] \times (0, 1.30)] [([1.36, 1.78] \times (0, 0.96))]$</td>
<td>(1, 1)</td>
<td>$(0.434, -0.107)$</td>
<td>0.014</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$(0.8, 0.4, 0.7)$</td>
<td>$[[0, 1.08] \times (0, 1.47)] [([1.08, 1.37] \times (0, 1.16))]$</td>
<td>(1, 1)</td>
<td>$(-0.057, 0.115)$</td>
<td>0.011</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$(3.0, 1.0, 0.4)$</td>
<td>$[[0, 1.11] \times (0, 1.47)] [([1.11, 1.44] \times (0, 1.14))]$</td>
<td>(1, 1)</td>
<td>$(0.017, 0.062)$</td>
<td>0.003</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$(4.0, 0.5, 0)$</td>
<td>$[[0, 1.06] \times (0, 1.58)] [([1.06, 1.44] \times (0, 1.18))]$</td>
<td>(1, 1)</td>
<td>$(0.081, 0.224)$</td>
<td>0.013</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$(2.0, 0.5, 1.5)$</td>
<td>$[[0, 1.05] \times (0, 1.27)] [([1.05, 1.2] \times (0, 1.11))]$</td>
<td>(1, 1)</td>
<td>$(-0.285, 0.04)$</td>
<td>0.042</td>
<td>0 Circle involute</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$(2, -0.2, 0)$</td>
<td>$[[0, 1.18] \times (0, 1.42)] [([1.18, 1.55] \times (0, 1.07))]$</td>
<td>(1, 1)</td>
<td>$(0.176, 0.023)$</td>
<td>0.008</td>
<td>0 Circle involute</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>$(-1.0, 1.1)$</td>
<td>$[[0, 1.10] \times (0, 1.35)] [([1.1, 1.32] \times (0, 1.12))]$</td>
<td>(1, 1)</td>
<td>$(-0.113, -0.039)$</td>
<td>0.008</td>
<td>0 Clothoid</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>$(-1.0, 5.0)$</td>
<td>$[[0, 1.01] \times (0, 1.77)] [([1.01, 1.4] \times (0, 1.31))]$</td>
<td>(1, 1)</td>
<td>$(-0.008, 0.249)$</td>
<td>0.019</td>
<td>0 Clothoid</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$(1, -0.2, -1)$</td>
<td>$[[0, 3.2] \times (0, 1.24)]$</td>
<td>(1, 1)</td>
<td>$(0.643, -0.018)$</td>
<td>0.015</td>
<td>0 Logarithmic spiral</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>$(1.0, 5.0)$</td>
<td>$[[0, 1.05] \times (0, 1.63)] [([1.05, 1.42] \times (0, 1.21))]$</td>
<td>(1, 1)</td>
<td>$(0.061, 0.243)$</td>
<td>0.007</td>
<td>0 Logarithmic spiral</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>$(0, 1.1)$</td>
<td>$[[0, 1.03] \times (0, 1.49)] [([1.03, 1.3] \times (0, 1.2))]$</td>
<td>(1, 1)</td>
<td>$(-0.13, 0.286)$</td>
<td>0.038</td>
<td>0 Neilsen’s spiral</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>$(0, 0.7, 0)$</td>
<td>$[[0, 1.00] \times (0, 1.74)] [([1.00, 1.28] \times (0, 1.28))]$</td>
<td>(1, 1)</td>
<td>$(0.045, 0.331)$</td>
<td>0.006</td>
<td>0 Neilsen’s spiral</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>$(1, -0.5, 0.2)$</td>
<td>$[[0, 1.25] \times (0, 1.25)] [([1.25, 1.55] \times (0, 0.98))]$</td>
<td>(0.989, 0.989)</td>
<td>$(0.471, -0.194)$</td>
<td>0.045</td>
<td>1 Logarithmic spiral</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>$(1, 1.5, 0.5)$</td>
<td>$[[0, 1.00] \times (0, 1.59)] [([1.00, 1.33] \times (0, 1.22))]$</td>
<td>(0.984, 0.988)</td>
<td>$(-0.028, 0.31)$</td>
<td>0.025</td>
<td>1 Logarithmic spiral</td>
<td></td>
</tr>
</tbody>
</table>
However, examples with iterations propose that the trigonometric method provides errors of 0.005753 and 0.015615 for the GLAC \((\alpha, \Lambda, \gamma) = (0,0.5,0.2)\) and GLAC \((\alpha, \Lambda, \gamma) = (2,0.3,−0.4)\), respectively. This is more accurate than the errors of the cubic Bézier method which are 0.019160 and 0.022365.

Table 4 displays the results of several normalized \(\rho\)-shift GLAC approximated curves. Table 5 shows the results of approximating the various \(\rho\)-shift GLAC with different arc lengths.

For the normalized \(\rho\)-shift GLACs, the cubic Bézier approximation method [21] still requires more iterations to reach satisfactory approximations. It is evident that the cubic trigonometric basis functions of the cubic trigonometric Bézier approximation method contribute to an increase in accurate results compared to the Bernstein polynomial basis functions of the cubic Bézier approximation method. The last example of Table 4 is another example which shows that even though the cubic trigonometric method does not reach a satisfactory approximation with the initial values, it still produces a result that is more accurate than the standard cubic Bézier method.

For \(\rho\)-shift GLAC with various arc lengths, the pattern remains the same. The cubic trigonometric method requires less iteration than the cubic Bézier method for most of the examples while it provides more accurate results.

In general, the cubic T-Bézier takes more time than the standard cubic Bézier as more time is needed for computations of the trigonometric basis functions as compared to the Bernstein polynomial basis functions.

Concerning accuracy, as the cubic trigonometric Bézier curve is closer to the control polygon compared to the standard cubic Bézier curve, the cubic trigonometric Bézier curve has an extra degree of freedom. This explains the advantage related to the accuracy of the proposed GLAC cubic trigonometric Bézier approximated curves over cubic Bézier approximated curves.
Figure 2: (a) The GLAC $(\Lambda, \alpha, \nu) = (2, -0.2, 0)$ and its approximation. (b) The curvature error $\varepsilon$. (c) The restricted solution region of $(\xi_1, \omega_1)$ with corresponding $\lambda$ values. (d) The restricted solution region of $(\xi_1, \omega_1)$ with corresponding $\mu$ values.

Figure 3: Continued.
Figure 3: (a) The GLAC $(\Lambda, \alpha, \gamma) = (-1, 0, 1, 1)$ and its approximation. (b) The curvatures error. (c) The restricted solution region of $(\xi_1, \omega_1)$ with corresponding $\lambda$ values. (d) The restricted solution region of $(\xi_1, \omega_1)$ with corresponding $\mu$ values.

Figure 4: Continued.
Figure 4: (a) The GLAC \((\Lambda, \alpha, \nu) = (1, -0.5, 0.2)\) and its approximation at the first iteration. (b) The same GLAC and its approximation at the second iteration. (c) The curvatures of the GLAC \((\Lambda, \alpha, \nu) = (1, -0.5, 0.2)\) and its approximation at the first iteration. (d) The curvatures of the same GLAC and its approximation at the second iteration. (e) The curvature errors of the GLAC \((\Lambda, \alpha, \nu) = (1, -0.5, 0.2)\) and its approximation at the first and second iterations. (f) \(\lambda\) over the restricted solution region. (g) \(\mu\) over the restricted solution region.

Table 2: Approximations of normalized \(\kappa\)-shift GLAC.

<table>
<thead>
<tr>
<th>(\kappa)-GLAC ((\alpha, \Lambda, \nu))</th>
<th>((\xi_1, \omega_1))</th>
<th>Cubic Bézier</th>
<th>Cubic trigonometric Bézier</th>
</tr>
</thead>
<tbody>
<tr>
<td>((-1, 0.5, 1)) ((1, 1)) ((0.985, 1.140))</td>
<td>0 1</td>
<td>0.596104 0.025942</td>
<td>0.3906 0.8125</td>
</tr>
<tr>
<td>((0, 0.5, 0.5)) ((1, 1)) ((0.955, 1.113))</td>
<td>0 1</td>
<td>0.394663 0.008084</td>
<td>0.4063 0.8594</td>
</tr>
<tr>
<td>((0.7, -0.6, 0.1)) ((1, 1)) ((1.242, 0.862))</td>
<td>0 1</td>
<td>0.634895 0.017444</td>
<td>0.5000 0.8594</td>
</tr>
<tr>
<td>((1, 0.1, 0.6)) ((1, 1)) ((1.031, 1.062))</td>
<td>0 1</td>
<td>0.441414 0.017211</td>
<td>0.3750 0.7656</td>
</tr>
<tr>
<td>((2, -0.1, 0.2)) ((1, 1)) ((1.056, 1.007))</td>
<td>0 1</td>
<td>0.34501 0.007432</td>
<td>0.4063 0.7969</td>
</tr>
</tbody>
</table>
A new set of $G^2$ formulas to approximate GLAC with cubic T-Bézier for the two shape parameters $\lambda$ and $\mu$ is derived. These new formulas employ the basis functions of cubic T-Bézier as proposed by Han et al. [11] and the intermediate control points are expressed in the form of GLACs’ shape parameters. A solution region is established to ensure the efficiency of the approximation is at optimum. Based on the numerical results, the proposed algorithm may obtain an approximation of GLAC within 0.05 tolerances either on the first or second iteration. Future work includes approximating GLAC with generalized Trigonometric Bézier curves controlled at curvature extrema.

### Data Availability

The numerical simulation data used to support the findings of this study are included within the article.

### Disclosure

The manuscript was already published as an abstract [25].

### Conflicts of Interest

The authors declare that they have no conflicts of interest.
Authors’ Contributions

D.J.A., R.U.G., and K.T.M. conceptualized the study; D.J.A. and R.U.G. designed the methodology; R.U.G. validated the study; R.U.G. and K.T.M. were responsible for the resources; D.J.A. and R.U.G. wrote the original draft; R.U.G. and K.T.M. reviewed and edited the article; R.U.G. and K.T.M. visualized and supervised the study; R.U.G. was responsible for project administration; R.U.G. and K.T.M. were responsible for funding acquisition. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The authors acknowledge the Arab Evangelical Episcopal Church Council (Ahliyyah and Mutran), Ibn Rushd National Academy, and University Malaysia Terengganu (UMT) for providing facilities to complete this work. This work was partially supported by JST CREST Grant no. JPMJCR1911 and JSPS Grant-in-Aid for Scientific Research (B) Grant no. 19H02048. We acknowledge UMT for providing travel fund to share preliminary outcome of this work at the Geometric Computation and Applications Workshop held in June 2018, at Trinity College, Dublin, Ireland.

References