Title:
Fast G^1 Hermite Interpolation of Euler Bézier/B-spline Spirals

Authors:
Gaku Sasaki, ciga26006@g.nihon-u.ac.jp, Nihon University
Taisei Inoue, inoue.taisei@nihon-u.ac.jp, Nihon University
Norimasa Yoshida, yoshida.norimasa@nihon-u.ac.jp, Nihon University
Takafumi Saito, txsaito@cc.tuat.ac.jp, Tokyo University of Agriculture and Technology

Keywords:
Bézier curves, B-spline curves, Euler spirals, G^1 Hermite interpolation, Monotonically varying curvature

DOI: 10.14733/cadconfP.2026.1-6

Introduction:

An Euler spiral, also known as the Cornu spiral or clothoid, is characterized by curvature that varies linearly with respect to arc length. It is utilized in aesthetic shape design, route planning for robots, and highway road design. In previous studies, Euler spirals have been represented using Fresnel integrals [1] or approximated by Bézier curves via Taylor expansions [2].

Yang proposed a method [5] to approximate Euler spirals using polynomial Bézier or B-spline curves constructed using Euler polygons; this approximation is hereafter referred to as an Euler Bézier/B-spline spiral. Yang's method employs a smoothing process. However, a major limitation of this method is its reliance on iterative calculations to achieve convergence, which makes it significantly computationally expensive. Therefore, in this study, we propose a method for rapidly generating Euler Bézier/B-spline spirals utilizing the properties of Euler spirals and the concept of standard forms.

In addition, to overcome the limitation of Yang's curvature monotonicity condition, which is proven only as a necessary condition, our system integrates the curvature monotonicity evaluation function (CMEF) [4]. By implementing a real-time cross-validation algorithm during our fast G^1 Hermite interpolation process, the system strictly verifies whether the resulting curvature indeed varies monotonically whenever Yang's condition is satisfied.

Euler Polygon and Euler Bézier/B-spline Spirals [5]:

Consider a sequence of $n+1$ control points, $\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$, where $n \geq 3$. Let θ_i denote the angle between the edge vectors $\boldsymbol{P}_i - \boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_{i+1} - \boldsymbol{P}_i$ for $i = 1, \ldots, n-1$. According to Yang[5], given an initial angle θ_1 and an angle increment $\Delta\theta$, an *Euler polygon* is defined as a control polygon satisfying the following two conditions:

$$|\boldsymbol{P}_i - \boldsymbol{P}_{i-1}| = |\boldsymbol{P}_{i+1} - \boldsymbol{P}_i|, \quad i = 1, 2, \ldots, n-1 \tag{2.1}$$

$$\theta_i = \theta_1 + (i-1)\Delta\theta, \quad i = 2, \ldots, n-1 \tag{2.2}$$

Assuming the control polygon is an Euler polygon with $|\theta_i| < \frac{\pi}{2}$, Yang demonstrated that if

$$\operatorname{sign}(\Delta\theta) = \operatorname{sign}(\kappa'(0)) = \operatorname{sign}(\kappa'(1)), \tag{2.3}$$

Proceedings of CAD'26, Athens, Greece, July 6-8, 2026, 1-6

the curvature becomes monotonically varying. Here, $\kappa'(0)$ and $\kappa'(1)$ denote the derivatives of curvature with respect to the parameter at the start and end points of the curve, respectively. Based on this property, which is a necessary condition for monotonically varying curvature, Yang defined an *Euler Bézier spiral* as a Bézier curve.

Similarly, for a cubic uniform B-spline curve, Yang defined an *Euler B-spline spiral* as a curve derived from an Euler polygon with $|\theta_i| < \frac{\pi}{2}$, wherein

$$\operatorname{sign}(\Delta\theta) = \operatorname{sign}(\kappa_1'(0)) = \operatorname{sign}(\kappa_1'(1)) = \operatorname{sign}(\kappa_m'(0)) = \operatorname{sign}(\kappa_m'(1)). \tag{2.4}$$

Here, $\kappa_1'(0)$ and $\kappa_1'(1)$ represent the curvature derivatives at the start and end points of the first segment, respectively, while $\kappa_m'(0)$ and $\kappa_m'(1)$ correspond to the curvature derivatives at the start and end points of the m-th (final) segment.

Yang's Method Based on a Smoothing Process :

This section outlines Yang's method for generating Euler Bézier/B-spline spirals [5]. To generate an Euler Bézier spiral, a set of initial control points $\boldsymbol{P}_i$ $(i = 0, 1, \ldots, n)$ is provided as input. The control points $\boldsymbol{P}_i$ and the angles θ_i are iteratively computed using the following set of equations until the resulting control polygon converges to an Euler polygon. Let $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$ denote the unit tangent vectors at $\boldsymbol{P}_0$ and $\boldsymbol{P}_n$, respectively, and let l_{ave} represent the average edge length of the control polygon. The operator $\mathrm{R}\left(\frac{\pi}{2}\right)$ denotes a counter-clockwise rotation of $\frac{\pi}{2}$ radians. The hat notation (e.g., $\hat{\boldsymbol{P}}$) indicates a control point updated via the following iterative *smoothing process* described below.

$$\hat{\boldsymbol{P}}_1 = \boldsymbol{P}_0 + l_{ave}\boldsymbol{T}_a, \ \hat{\boldsymbol{P}}_{n-1} = \boldsymbol{P}_n - l_{ave}\boldsymbol{T}_b \tag{3.1}$$

$$\hat{\boldsymbol{P}}_i = \frac{\boldsymbol{P}_{i-1} + \boldsymbol{P}_{i+1}}{2} - \tan\frac{\hat{\theta}_i}{2}\mathrm{R}\left(\frac{\pi}{2}\right)\frac{\boldsymbol{P}_{i+1} - \boldsymbol{P}_{i-1}}{2} \tag{3.2}$$

$$\hat{\theta}_i = \frac{1}{3}(\theta_{i-1} + \theta_i + \theta_{i+1}) \tag{3.3}$$

Here, we assume $n \geq 4$ since only two θ_i can be defined for $n = 3$.

The process for generating an Euler B-spline spiral is analogous to that for an Euler Bézier spiral. Unlike the Euler Bézier spiral, however, the endpoints of the B-spline curve do not generally coincide with the first and last control points; consequently, the control points $\hat{\boldsymbol{P}}_0$, $\hat{\boldsymbol{P}}_1$, $\hat{\boldsymbol{P}}_{n-1}$, and $\hat{\boldsymbol{P}}_n$ require a distinct calculation [5].

In Yang's method, since the initial input control points correspond to $n = 3$, degree-elevation to $n = 4$ is performed before executing the smoothing process. Starting with $n = 4$, the control points are iteratively updated using Eq. (3.1)-(3.3) until the Euler polygon conditions Eq. (2.1)-(2.2) are satisfied. Subsequently, the algorithm verifies whether the necessary conditions, Eq. (2.3) or Eq. (2.4), are met. If these conditions are not satisfied, the curve is degree-elevated by one. This process alternates between the smoothing process and the verification step until a valid spiral is obtained or the number of control points reaches the specified maximum.

Efficient Generation of Euler Bézier Spirals:

In this study, adopting the approach outlined in [3], we introduce the concept of a standard form, as illustrated in Fig. 1. Since constructing an Euler Bézier spiral is essentially a G^1 Hermite interpolation problem, the task reduces to finding a curve where the standard form parameters θ_d' and θ_e' match the given angles θ_d and θ_e. Given the target tangential angle θ_d and the degree n, the angle increment $\Delta\theta$ is derived from the Euler polygon condition as follows;

$$\Delta\theta = \frac{2(\theta_d - (n-1)\theta_1)}{(n-1)(n-2)}. \tag{4.1}$$

Consequently, the problem of G^1 Hermite interpolation reduces to adjusting the initial angle θ_1 so that θ'_e coincides with the target θ_e. Note that the uniform edge length (Eq. (2.1)) can be set to any positive value (e.g., 1), as this parameter merely scales the resulting curve. Fig. 2 illustrates an Euler polygon of 31 control points generated by varying θ_1 with $\theta_d = \frac{2}{3}\pi$. Here, the tangent vector at the origin is aligned with the positive x-axis, while the tangent vector at the endpoint is oriented counterclockwise by $\frac{2}{3}\pi$ from the x-axis. As indicated in Eq. (4.1), for a given θ_1, the value of $\Delta\theta$ is uniquely determined by θ_d and n. Fig. 2 illustrates the resulting behavior: the endpoint position moves counterclockwise (i.e., θ_e increases) as θ_1 increases and shifts clockwise as θ_1 decreases. Leveraging this monotonic property, we can efficiently generate an Euler Bézier spiral by optimizing θ_1, thereby eliminating the need for the iterative control point calculations required by Yang's method. The proposed method computes the Euler polygon through the following procedure. Assuming θ_1 is given and $\boldsymbol{P}_0$ is at the origin, we define the tangent vector at $\boldsymbol{P}_0$ as $\boldsymbol{v} = [1, 0]^T$ and set $\boldsymbol{P}_1 = \boldsymbol{P}_0 + \boldsymbol{v}$. We then calculate $\Delta\theta$ via Eq. (4.1) and determine the subsequent control points using:

$$\boldsymbol{P}_i = \boldsymbol{P}_{i-1} + R(\theta_1 + ... + \theta_{i-1})\boldsymbol{v} \quad (i = 2, 3, ..., n). \tag{4.2}$$

The control points generated by this method inherently satisfy the Euler polygon condition, and the resulting θ'_d matches the target θ_d. Although the tangent condition is satisfied, the endpoint angle θ'_e may not initially match the target θ_e. To address this, we optimize θ_1 using an extended bisection search to align θ'_e with θ_e. Specifically, initialized with $\theta_1 = 0$, we first establish a search interval. If the endpoint of the generated Euler polygon lies clockwise relative to the target vector(defined by θ_e), the upper bound of θ_1 is incrementally increased. Once the endpoint crosses to a counterclockwise position, the bisection method is applied to locate the solution. Conversely, if the initial position is counterclockwise, the lower bound of θ_1 is incrementally decreased until the position shifts clockwise, at which point the bisection search is executed. If the resulting curve satisfies the Euler Bézier spiral condition (Eq. (2.3)), an Euler Bézier spiral is generated. If the resulting curve fails to satisfy the condition, n is incremented, and the process is repeated until the Euler Bézier spiral condition is satisfied or the degree reaches a prescribed limit.

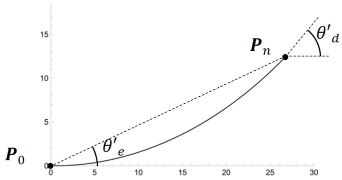
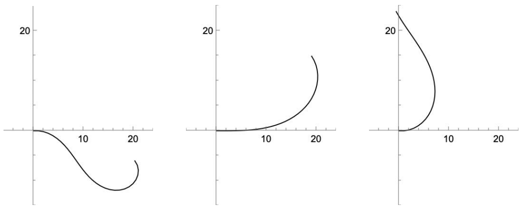


Fig. 1: Standard form



(a) $\theta_1 = -10°$ (b) $\theta_1 = 0°$ (c) $\theta_1 = 10°$

Fig. 2: Variation of the end point $\boldsymbol{P}_n$ with changing θ_1

Efficient Generation of Euler B-spline Spirals:

G^1 Hermite Interpolation of cubic Euler B-spline spirals is analogous to that of Euler Bézier spirals. The task reduces to finding a curve where the input angles θ_d and θ_e match the standard form parameters θ'_d and θ'_e. Note that since the start and end points of a cubic Euler B-spline curve are not $\boldsymbol{P}_0$ and $\boldsymbol{P}_n$, the angles θ_d and θ_e can instead be computed from the Euler polygon by exploiting the properties specific to cubic B-spline curves. Note that the expression for $\Delta\theta$ differs from the Euler Bézier spiral formulation and is defined as follows:

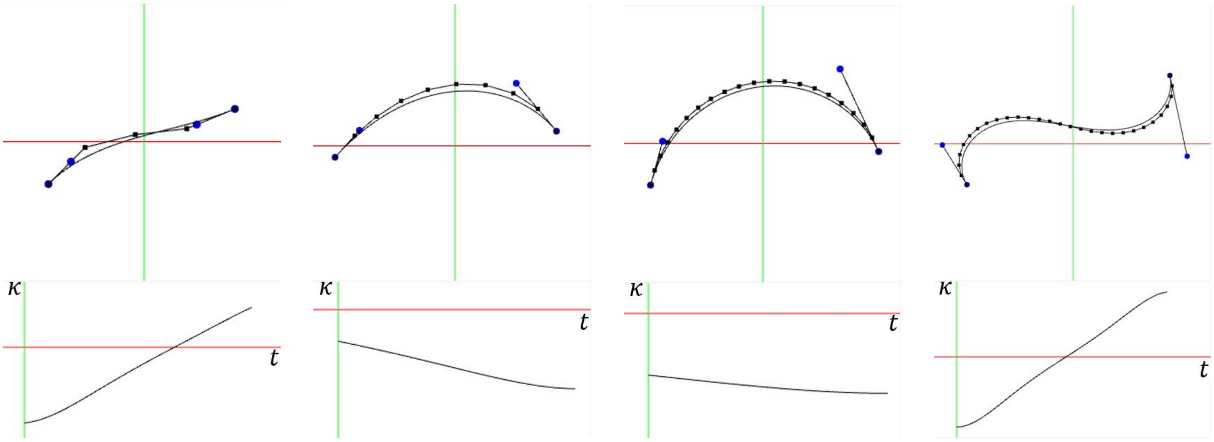
$$\Delta\theta = \frac{2(\theta_d - (n-2)\theta_1)}{(n-2)^2} \tag{5.1}$$

Adopting the same optimization strategy as described in the preceding section, we align θ'_e with θ_e to generate an Euler polygon. If the resulting curve satisfies the Euler B-spline spiral condition (Eq. (2.4)), an Euler B-spline spiral is generated. If the resulting polygon fails to meet the conditions, n is incremented by one, and the process is repeated up to a specified maximum limit.

A distinct advantage of this method lies in how the control points are utilized. Specifically, the control points of the computed Euler polygon are treated as defining a sequence of curve segments, where every consecutive set of four control points defines a single cubic uniform B-spline segment. This property facilitates the approximation of Euler spirals using cubic uniform B-spline curves, effectively accommodating an increased number of control points and a large value of n.

Results and Discussions:

Fig. 3 illustrates Euler Bézier spirals generated using the proposed method for n=4, 9, 20 and 28. In all four cases shown in Fig. 3(a)-(d), the curves generated by Yang's method and the proposed method are identical. The bottom panels of Fig. 3(a)-(d) plot the curvature κ against the parameter t, confirming the monotonically varying curvature.



(a) $n = 4$ (b) $n = 9$ (c) $n = 20$ (d) $n = 28$

Fig. 3: Euler Bézier spirals using the proposed method and their curvature plots

Table 1 compares the computation time required to generate the Euler Bézier spirals shown in Fig. 3, with both approaches implemented in C++. The computational performance was evaluated on a desktop system equipped with an AMD Ryzen 7 8700G (4.2GHz) CPU and 32 GB of DDR5 memory. The values listed for each n correspond to the curves displayed in Fig. 3(a)-(d). As Table 1 indicates, the proposed method exhibits significantly higher computational efficiency than the previous method, particularly as the degree n increases.

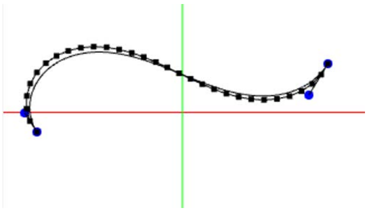
Fig. 4 illustrates examples of different curves generated under the same G^1 Hermite interpolation conditions. Our method successfully generates a spiral with $n = 30$; in contrast, Yang's method reaches a maximum degree of 149 without producing a spiral. The computation time for Fig. 4(a) is 1.263 ms, whereas that of Fig. 4(b) is 2090.517 ms. Consequently, Yang's method becomes numerically unstable and significantly loses its interactivity as the number of control points increases.

Fig. 5 shows Euler B-spline spirals generated using the proposed method for n=5, 10, 21, and 28. In the four cases shown in Fig. 5(a)-(d), the curves generated by Yang's method and the proposed method matched. The bottom panels of Fig. 5(a)-(d) plot the curvature κ against the parameter t.

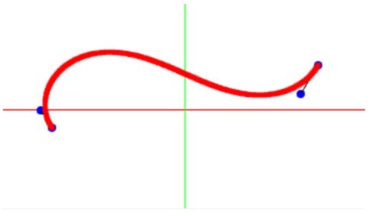
Table 2 presents a comparison of the computation time required to generate the Euler B-spline spirals shown

Table 1: Computation time of Euler Bézier spirals

n	Yang's Method (ms)	Proposed Method (ms)
4	0.076	0.145
9	0.238	0.192
20	7.929	0.515
28	27.347	1.217



(a) Proposed Method($n = 30$)



(b) Yang's Method($n = 149$)

Fig. 4: Examples of different curves

in Fig. 5, where the computation times correspond to the specific curves in Fig. 5(a)-(d). As Table 2 indicates, the proposed method significantly outperforms the previous method, particularly as the number of control points increases. A similar trend was observed for Euler B-spline spirals, as illustrated in Fig. 4.

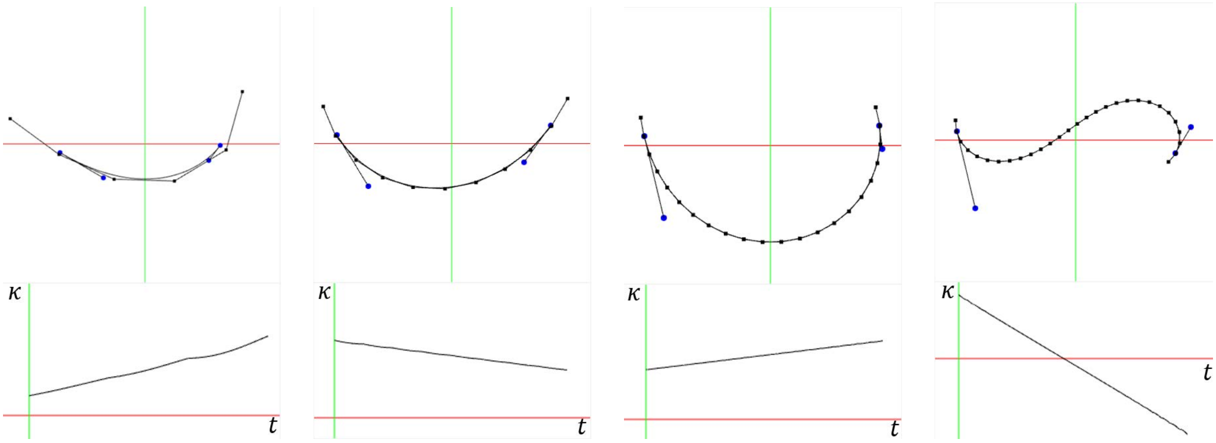
Table 2: Computation time of Euler B-spline spirals

n	Yang's Method (ms)	Proposed Method (ms)
5	0.017	0.021
10	0.156	0.049
21	12.113	0.173
28	31.914	0.295

Yang's method utilizes iterative convergence calculations that necessitate repeated updates to control points; consequently, curve generation becomes computationally intensive and numerically unstable as the number of control points increases. Our method demonstrates superior geometric flexibility: in instances where the target curve is a near-circular arc, our method can successfully generate a spiral that Yang's method is unable to produce.

A further distinction lies in the generated curves. The generation of Euler Bézier/B-spline spirals is based on G^1 Hermite interpolation, which theoretically admits more than two distinct curves that satisfy the given G^1 Hermite condition. While Yang's previous method converges to a single solution dependent on the initial control polygon, the proposed method can generate all the curves satisfying a given G^1 Hermite condition by adding $2k\pi\,(k = \dots, -1, 0, 1, \dots)$ to the tangential angle θ_d.

Finally, regarding curvature properties, only the necessary conditions for Euler Bézier spirals and B-spline spirals have been mathematically established. To ensure validity, we evaluated the results using the CMEF [4]. Specifically, curves were generated under a program configuration that terminated whenever the evaluation based on Yang's necessary conditions did not agree with the CMEF assessment. Within the scope of our experiments, we confirmed that whenever Yang's necessary conditions were satisfied, the resulting curvature profiles varied monotonically. For a strict verification of curvature monotonicity, the CMEF should be used in addition to Yang's necessary conditions.



(a) $n = 5$ (b) $n = 10$ (c) $n = 21$ (d) $n = 28$

Fig. 5: Euler B-spline spirals using the proposed method and their curvature plots

Conclusions:

In this study, we proposed a method for interactively and efficiently controlling Euler Bézier/B-spline spirals based on their geometric properties and standard forms. Regarding the generation of Euler Bézier/B-spline spirals, the proposed method offers greater computational efficiency and numerical stability than Yang's method [5], particularly as the number of control points increases. Furthermore, for Euler Bézier/B-spline spirals that satisfy Yang's necessary conditions, we additionally applied the CMEF to verify that the curvature is indeed monotonically varying. Within the scope of our experiments, we observed no cases in which Yang's conditions were satisfied while the CMEF indicated non-monotonic curvature.

Acknowledgement:

This work was supported by JSPS KAKENHI Grant Number 24K07280.

Gaku Sasaki, https://orcid.org/0009-0006-7815-3535
Taisei Inoue, https://orcid.org/0000-0002-2215-6589
Norimasa Yoshida, https://orcid.org/0000-0001-8889-0949
Takafumi Saito, https://orcid.org/0000-0001-5831-596X

References:

[1] D. S. Meek, R. S. D. Thomas, A guided clothoid spline, Computer Aided Geometric Design, 8, 2, 163-174, 1991. https://doi.org/10.1016/0167-8396(91)90042-A

[2] L. Wang, K. T. Miura, E. Nakamae, T. Yamamoto, T. J. Wang, An approximation approach of the clothoid curve defined in the interval [0,pi/2] and its offset by free-form curves, Computer Aided Geometric Design, Vol.33, No.14, 1049-1058, 2001. https://doi.org/10.1016/S0010-4485(00)00142-1

[3] N.Yoshida and T.Saito, Interactive Aesthetic Curve Segments, The Visual Computer, 22(9-11), 896-905, 2006. https://doi.org/10.1007/s00371-006-0076-5

[4] T.Saito and N.Yoshida, Curvature monotonicity evaluation functions on rational Bézier curves, Computers & Graphics. 114, 219-229, 2023. https://doi.org/10.1016/j.cag.2023.05.019

[5] X.Yang, Euler Bézier spirals and Euler B-spline spirals, Computer Aided Geometric Design, 112, 1-16, 2024. https://doi.org/10.1016/j.cagd.2024.102361