Fast G^1 Hermite Interpolation of Cubic Euler B-spline Spirals

Gaku Sasaki[1], Taisei Inoue[2], Norimasa Yoshida[3] Takafumi Saito[4]

[1]Nihon University, ciga26006@g.nihon-u.ac.jp
[2]Nihon University, inoue.taisei@nihon-u.ac.jp
[3]Nihon University, yoshida.norimasa@nihon-u.ac.jp
[4]Tokyo University of Agriculture and Technology, txsaito@cc.tuat.ac.jp

Corresponding author: N. Yoshida, yoshida.norimasa@nihon-u.ac.jp

Abstract. Euler spirals, characterized by linearly varying curvature with respect to arc length, are fundamental to shape design, route planning, and road design. While traditional methods represent these spirals using Fresnel integrals or Taylor expansions, a recent approach has utilized Euler polygons to construct polynomial Bézier or B-spline approximations. However, this method relies on computationally expensive iterative calculations to ensure convergence. This study proposes an efficient method for rapidly generating Euler B-spline spirals by leveraging their inherent geometric properties alongside the concept of the standard form. Furthermore, we rigorously evaluated the curvature monotonicity, including critical edge cases near circular arcs, using the Curvature Monotonicity Evaluation Function (CMEF). Our results confirmed that whenever Yang's necessary condition is satisfied, the curvature is strictly monotonically varying across all tested cases.

Keywords: Bézier curves, B-spline curves, Euler spirals, G^1 Hermite interpolation, monotonically varying curvature
DOI: https://doi.org/10.14733/cadaps.2027.175-188

1 INTRODUCTION

Euler spirals, also known as Cornu spirals or clothoids, are a fundamental class of curves whose curvature varies linearly with respect to arc length. Due to their aesthetically pleasing curvature profiles and smooth transitions, Euler spirals have been widely utilized in various applications, including aesthetic shape design [4, 5], route planning for robots [10], and highway road design [11].

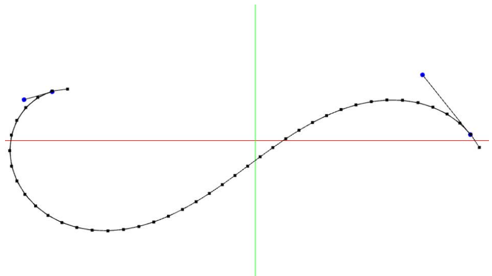
Despite their excellent geometric properties, Euler spirals are defined by Fresnel integrals [6]. Because these integrals lack a closed-form analytical solution, evaluating Euler spirals is computationally expensive. Furthermore, this non-polynomial nature makes it difficult to integrate them directly into standard Computer-Aided Design (CAD) systems, which predominantly rely on polynomial representations such as Bézier and B-spline curves.

To bridge this gap, researchers have studied the approximation of Euler spirals using polynomial curves. Euler Bézier and B-spline spirals proposed by Yang [13] have made it possible to generate Bézier and B-spline curves that closely approximate Euler spirals. However, significant challenges remain. First, generating these curves often requires computationally expensive iterative processes. Second, Yang's methods rely on the necessary conditions to evaluate the curvature monotonicity. Yang asserts that if these necessary conditions are met, the curvature varies monotonically. To investigate this, we extensively tested curvature monotonicity using the Curvature Monotonicity Evaluation Function (CMEF)[8], including critical edge cases near circular arcs.

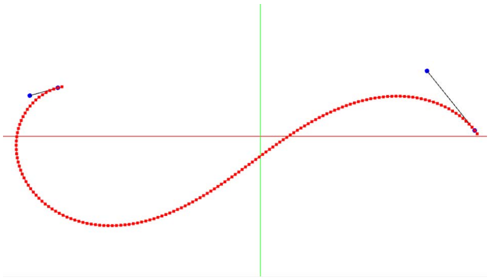
To address these fundamental limitations, this paper proposes an efficient algorithm for generating and rigorously evaluating Euler B-spline spirals. We primarily focus on cubic Euler B-spline spirals because the degree of Euler Bézier spirals can become quite high, and the curvature plots of Euler B-spline spirals are preferable to those of Euler Bézier spirals, as detailed in Sec. 5.1. We developed a faster, more robust generation algorithm that significantly reduces computational cost, which can also be applied to Euler Bézier spirals [9]. Fig. 1 illustrates generated examples of Euler B-spline spirals. Our method successfully produces a valid Euler B-spline spiral in 0.298 ms (Fig. 1(a)). In contrast, Yang's method fails to generate a valid spiral, meaning the curvature does not vary monotonically, even when the number of control points reaches its maximum limit (Fig. 1(b)). Furthermore, the computation time for Yang's method is 2229.130 ms, a delay that severely compromises interactive applications.

The main contributions of this paper are summarized as follows:

- We propose a highly efficient algorithm for generating cubic Euler B-spline spirals, reducing the iterative computational cost compared to Yang's method.
- Our algorithm demonstrates superior stability compared to Yang's method. Given identical G^1 Hermite constraints, our approach successfully converges to a valid Euler B-spline spiral even in scenarios where Yang's method fails. See Fig. 1.
- We rigorously evaluated the curvature monotonicity, including critical edge cases near circular arcs, using the CMEF. Our results confirmed that whenever Yang's necessary condition is satisfied, the curvature is strictly monotonically varying across all tested cases.



(a) Proposed method($n = 42$, 0.298 ms).



(b) Yang's method($n = 149$, 2229.130 ms).

Figure 1: Under identical G^1 Hermite interpolation constraints, the proposed method efficiently generates an Euler B-spline spiral, whereas Yang's method fails to produce a valid spiral and incurs a computation delay that severely hinders interactivity.

2 RELATED WORK

In aesthetic geometric design, such as the creation of "Class A" surfaces in the automotive and product design industries, curves with strictly monotonically varying curvature are highly desirable to ensure premium aesthetic quality [3]. While standard representations like Non-Uniform Rational B-Splines (NURBS) [7] are the industry standard, mathematically enforcing strict curvature monotonicity on these general free-form curves is not easy. Because the curvature derivative of a NURBS curve is a highly complex, non-linear function of its control points [2], designers are often forced to rely on tedious manual adjustments to achieve fair shapes. Euler spirals have attracted significant attention as an ideal alternative, as their curvature inherently varies linearly, and thus strictly monotonically, with respect to their arc length [5].

Integrating traditional Euler spirals into standard polynomial-based CAD systems presents significant challenges due to their reliance on Fresnel integrals. To reduce this cost, various strategies have been proposed. One direction focuses on optimizing the exact numerical computation of these curves; for instance, Bertolazzi et al. [1] proposed a fast algorithm for G^1 Hermite interpolation by reducing the complex non-linear system to a single equation. However, because standard CAD kernels inherently require polynomial or rational formats, approximation methods are frequently employed. One such approach involves alternative geometric representations, such as piecewise clothoid curves [4], which generate smooth spirals without directly relying on the traditional integral forms. Another approach involves approximating Euler spirals using standard free-form polynomial curves. For instance, Wang et al. [11] approximated Euler spirals using polynomial Bézier and B-spline curves by applying Taylor expansion to the integral equations. Similarly, Yoshida et al. [15] employed optimization techniques to represent log-aesthetic curves, including Euler spirals, using rational cubic Bézier curves.

Among these advancements, the recent work by Yang [13] is particularly noteworthy. Yang proposed a framework to generate Euler Bézier spirals and Euler B-spline spirals by introducing a specialized geometric structure termed an "Euler polygon." This approach marked a step forward in integrating Euler spirals into standard CAD environments while satisfying G^1 boundary conditions.

As discussed in the Introduction, Yang's method relies on computationally expensive iterative processes. In this paper, we propose a faster G^1 Hermite interpolation method using the property of an Euler polygon. The detailed mathematical formulation of Yang's method, which serves as the preliminary basis for our proposed method, is introduced in the next section.

3 YANG'S EULER B-SPLINE SPIRALS

Euler B-spline spirals are constructed by employing Euler polygons as their control polygons, which are specially defined polygons with equal-length edges and linearly varying vertex angles. Let $\boldsymbol{P}_0, \boldsymbol{P}_1, \ldots, \boldsymbol{P}_n$ be the control points, where $n+1$ $(n \geq 3)$ is the number of control points. Let θ_i be the signed angle between $\boldsymbol{P}_i - \boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_{i+1} - \boldsymbol{P}_i$. An Euler polygon satisfies the following geometric conditions:

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

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

where $\Delta\theta$ is a constant angle difference. Note that Yang required $|\theta_i| < \frac{\pi}{2}$. Fig. 2 shows an example of an Euler Polygon with $n = 5$. Given an arbitrary control polygon, Yang proposed a smoothing process that transforms the control polygon into an Euler polygon to achieve G^1 Hermite interpolation.

For a uniform cubic B-spline curve to be a spiral, which is characterized by monotonically varying curvature, Yang's necessary condition requires that $\Delta\theta$, $F_0(\theta_1)$, $F_1(\theta_2)$, $F_0(\theta_{n-2})$, and $F_1(\theta_{n-1})$ all have the same sign, where:

$$F_0(\theta) = -3\sin\frac{\theta}{2} + \sin\left(\frac{3\theta}{2} + \Delta\theta\right), \tag{3}$$

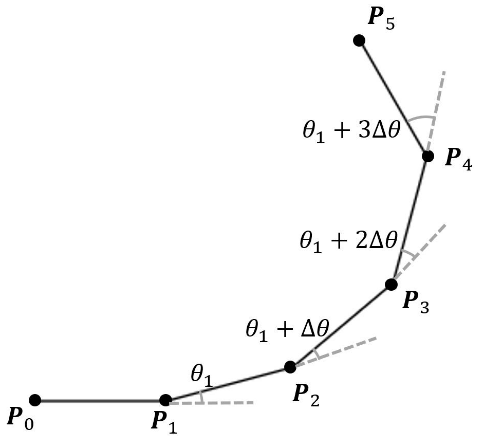


Figure 2: An example of an Euler polygon: $|\boldsymbol{P}_i - \boldsymbol{P}_{i-1}| = |\boldsymbol{P}_{i+1} - \boldsymbol{P}_i|$ and $\theta_i = \theta_1 + (i-1)\Delta\theta$ for $i = 1, 2, \ldots, n-1$.

$$F_1(\theta) = 3\sin\frac{\theta}{2} - \sin\left(\frac{3\theta}{2} - \Delta\theta\right). \tag{4}$$

Yang asserted that if the necessary condition is satisfied, the curvature becomes monotonically varying. Because $F_0(\theta_1)$ and $F_1(\theta_2)$ determine the signs of the first derivatives of the curvature, $\kappa_1'(0)$ and $\kappa_1'(1)$, at the start and end points of the first segment respectively, and $F_0(\theta_{n-2})$ and $F_1(\theta_{n-1})$ determine the signs of the curvature derivatives, $\kappa_m'(0)$ and $\kappa_m'(1)$, at the start and end points of the m-th (final) segment respectively, this condition can be equivalently rewritten as follows:

$$\mathrm{sign}(\Delta\theta) = \mathrm{sign}(\kappa_1'(0)) = \mathrm{sign}(\kappa_1'(1)) = \mathrm{sign}(\kappa_m'(0)) = \mathrm{sign}(\kappa_m'(1)). \tag{5}$$

Yang defines curves satisfying the necessary condition as Euler B-spline spirals.

The process for generating a cubic Euler B-spline spiral is analogous to that for an Euler Bézier spiral [9]. Specifically, to generate an Euler polygon that satisfies the G^1 Hermite conditions, the smoothing process iteratively updates the tangential angles and control points. Eq. (6) calculates the updated angles $\hat{\theta}_i$, which iteratively guides the tangential angles to approach a constant difference $\Delta\theta$. Subsequently, using these updated angles and the current control points, Eq. (7) generates the new control points $\hat{\boldsymbol{P}}_i$, driving the control polygon to gradually converge toward equal edge lengths.

$$\hat{\theta}_i = \frac{1}{3}(\theta_{i-1} + \theta_i + \theta_{i+1}) \tag{6}$$

$$\hat{\boldsymbol{P}}_i = \frac{\boldsymbol{P}_{i-1} + \boldsymbol{P}_{i+1}}{2} - \tan\frac{\hat{\theta}_i}{2}\mathrm{R}\left(\frac{\pi}{2}\right)\frac{\boldsymbol{P}_{i+1} - \boldsymbol{P}_{i-1}}{2} \tag{7}$$

Here, we assume $n \geq 4$ since only two θ_i can be defined for $n = 3$. Unlike Bézier curves, the endpoints of a uniform cubic B-spline curve do not interpolate its first and last control points. Consequently, the calculation of the boundary control points $\hat{\boldsymbol{P}}_0$, $\hat{\boldsymbol{P}}_1$, $\hat{\boldsymbol{P}}_{n-1}$, and $\hat{\boldsymbol{P}}_n$ requires a distinct formulation during the smoothing process [13].

Subsequently, the algorithm verifies whether the necessary curvature monotonicity condition (Eq. ((5)) is met. If this condition is not satisfied, the curve is degree-elevated by one. This procedure alternates between the smoothing process and the verification step until a valid spiral is obtained or the number of control points reaches a specified maximum.

4 EFFICIENT GENERATION OF EULER B-SPLINE SPIRALS

In this section, we propose a method capable of generating Euler B-spline spirals faster than Yang's method. Similarly to Yang's method, the proposed method represents the shape as a single uniform cubic B-spline curve defined by $n+1$ control points, which inherently consists of $n-2$ curve segments.

4.1 Standard Form of Euler B-spline Spirals

In G^1 Hermite interpolation, a curve is generated to pass through two given endpoints, $\boldsymbol{P}_a$ and $\boldsymbol{P}_b$, while satisfying their corresponding unit tangents, $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$. As illustrated in Fig. 3, θ_e is the angle formed by $\boldsymbol{T}_a$ and the vector $\boldsymbol{P}_b - \boldsymbol{P}_a$, and θ_d is the angle formed by $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$. Our approach utilizes an Euler polygon in standard form, defined by $\boldsymbol{P}_0 = [0\ 0]^{\mathrm{T}}$ and $\boldsymbol{P}_1 = [1\ 0]^{\mathrm{T}}$. An Euler polygon in standard form, shown in Fig. 2, can be uniquely constructed if n, θ_1, and $\Delta\theta$ are given. $\boldsymbol{P}_i$ $(i = 2, ..., n)$ are computed by

$$\boldsymbol{P}_i = \boldsymbol{P}_{i-1} + R(\theta_1 + \cdots + \theta_{i-1})\boldsymbol{v} \quad (i = 2, 3, \ldots, n), \tag{8}$$

where $R(\theta)$ denotes the 2D rotation matrix by angle θ and $\boldsymbol{v} = [1\ 0]^{\mathrm{T}}$. As we will show, two angles θ'_d and θ'_e are computed from a curve generated by an Euler polygon. Following [14], G^1 Hermite interpolation is equivalent to finding an Euler polygon where $\theta_d = \theta'_d$ and $\theta_e = \theta'_e$.

In G^1 Hermite interpolation, we first determine a standard-form Euler polygon that matches the angles θ_d and θ_e. As will be described shortly, $\Delta\theta$ can be computed from n, θ_d, and θ_1. Once found, we apply an appropriate similarity transformation so that the polygon aligns with the given endpoints and their tangents.

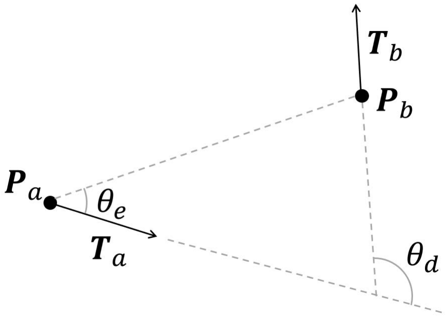


Figure 3: G^1 Hermite conditions defined by endpoints $\boldsymbol{P}_a, \boldsymbol{P}_b$ and unit tangents $\boldsymbol{T}_a, \boldsymbol{T}_b$, which are characterized by angles θ_d and θ_e.

Let θ'_d be the angle between the initial and final tangent vectors of a curve generated from an Euler polygon (Fig. 4). Let θ'_e be the angle between the initial tangent vector and the vector directed from the starting point to the endpoint of the curve (Fig. 5). We describe how θ'_d and θ'_e are computed from an Euler polygon. Let $\boldsymbol{P}_0, \boldsymbol{P}_1, \boldsymbol{P}_2$, and $\boldsymbol{P}_3$ be the control points of a uniform cubic B-spline curve segment. The curve $\boldsymbol{Q}(t)$ is

$$\boldsymbol{Q}(t) = \frac{(1-t)^3}{6}\boldsymbol{P}_0 + \frac{3t^3 - 6t^2 + 4}{6}\boldsymbol{P}_1 + \frac{-3t^3 + 3t^2 + 3t + 1}{6}\boldsymbol{P}_2 + \frac{t^3}{6}\boldsymbol{P}_3. \tag{9}$$

Substituting $t = 0$ and $t = 1$ yields the start and end points of this curve segment, respectively:

$$\boldsymbol{Q}(0) = \frac{1}{6}(\boldsymbol{P}_0 + 4\boldsymbol{P}_1 + \boldsymbol{P}_2), \qquad \boldsymbol{Q}(1) = \frac{1}{6}(\boldsymbol{P}_1 + 4\boldsymbol{P}_2 + \boldsymbol{P}_3).$$

Differentiating Eq. (9) with respect to t yields

$$\dot{\boldsymbol{Q}}(t) = -\frac{1}{2}(1-t)^2\boldsymbol{P}_0 + \frac{1}{2}(3t^2-4t)\boldsymbol{P}_1 + \frac{1}{2}(-3t^2+2t+1)\boldsymbol{P}_2 + \frac{t^2}{2}\boldsymbol{P}_3.$$

Substituting $t=0$ and $t=1$ into the derivative gives the tangent vectors at the start and end points of the segment, respectively:

$$\dot{\boldsymbol{Q}}(0) = \frac{1}{2}(\boldsymbol{P}_2 - \boldsymbol{P}_0), \qquad \dot{\boldsymbol{Q}}(1) = \frac{1}{2}(\boldsymbol{P}_3 - \boldsymbol{P}_1).$$

θ'_d is computed as the angle formed by the initial tangent vector $\boldsymbol{P}_2 - \boldsymbol{P}_0$ and the final tangent vector $\boldsymbol{P}_n - \boldsymbol{P}_{n-2}$. See Fig. 4. Similarly, θ'_e is computed as the angle formed by $\boldsymbol{P}_2 - \boldsymbol{P}_0$ and $\boldsymbol{P}_e - \boldsymbol{P}_s$. See Fig. 5.

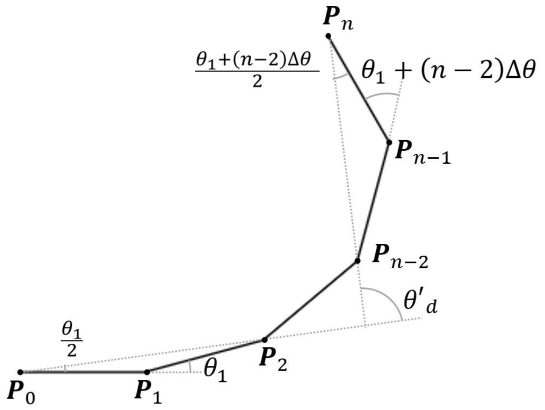


Figure 4: θ'_d is the angle between the initial tangent vector $\boldsymbol{P}_2 - \boldsymbol{P}_0$ and the final tangent vector $\boldsymbol{P}_n - \boldsymbol{P}_{n-2}$.

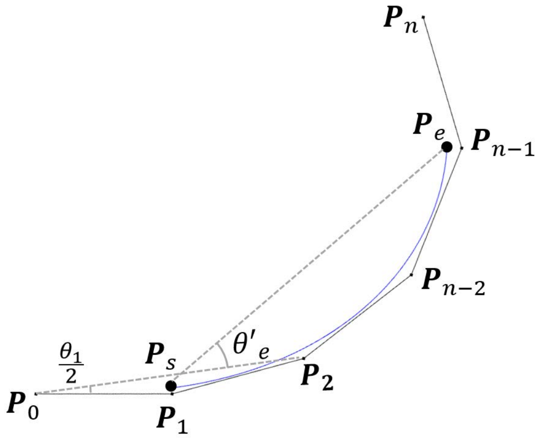


Figure 5: θ'_e denotes the angle between the initial tangent vector $\boldsymbol{P}_2 - \boldsymbol{P}_0$ and the chord vector $\boldsymbol{P}_e - \boldsymbol{P}_s$ connecting the start and end points.

From Fig. 4, θ'_d is:

$$\theta'_d = (n-1)\theta_1 + \frac{(n-1)(n-2)}{2}\Delta\theta - \frac{\theta_1}{2} - \frac{\theta_1 + (n-2)\Delta\theta}{2}. \tag{10}$$

Solving Eq. (10) with respect to $\Delta\theta$ yields:

$$\Delta\theta = \frac{2(\theta'_d - (n-2)\theta_1)}{(n-2)^2}. \tag{11}$$

Using $\Delta\theta$ as defined in Eq. (11), the condition $\theta_d = \theta'_d$ is always satisfied. Consequently, the problem of G^1 Hermite interpolation reduces to finding a value for θ_1 such that $\theta_e = \theta'_e$.

4.2 Algorithm for G^1 Hermite Interpolation

The proposed G^1 Hermite interpolation by Euler B-spline spirals consists of the following four steps:

1. Set $n=3$. Compute θ_d and θ_e from the given two endpoints $\boldsymbol{P}_a$ and $\boldsymbol{P}_b$, and their respective tangent vectors $\boldsymbol{T}_a$ and $\boldsymbol{T}_b$.

2. Compute $\Delta\theta$ using Eq. (11) by initially setting $\theta_1 = 0$.

3. Perform an extended bisection search shown in Algorithm 1 to find the value of θ_1 such that θ'_e equals θ_e.

4. Evaluate whether the curvature of the curve defined by the control points is monotonically varying using Yang's condition (Eq. (5)). If Yang's condition is met, we verify the curvature monotonicity using the CMEF. If the monotonicity is confirmed, generate an Euler B-spline spiral by applying an appropriate similarity transformation. If Yang's condition is not met, increment n by one and repeat the process from Step 2 until a valid Euler B-spline spiral is produced or n reaches a specified maximum.

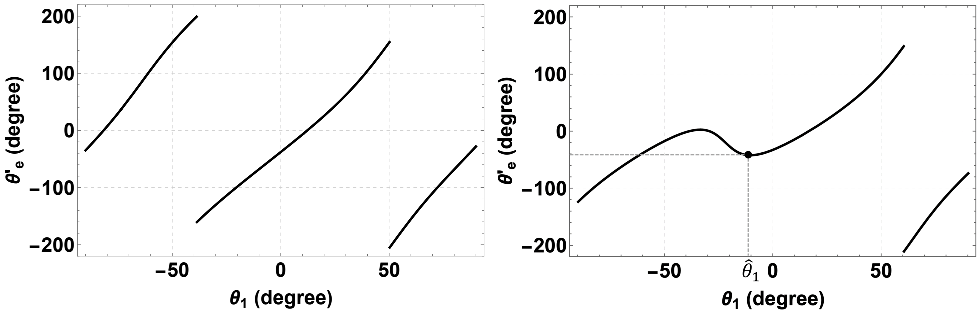
Algorithm 1 Extended Bisection Search for θ_1 with Dynamic Interval Expansion

Require: Initial guess θ_1, target angle θ_e, tolerance ϵ, step size Δ_{step}
Ensure: Optimal θ_1 that satisfies $|\theta'_e - \theta_e| \leq \epsilon$, or Failure

$\theta_{\min} \leftarrow \theta_1$, $\theta_{\max} \leftarrow \theta_1$
bracketed $\leftarrow$ False
$\Delta_{\text{prev}} \leftarrow \{\}$
while maximum iterations not reached **do**
    $\theta_1 \leftarrow (\theta_{\min} + \theta_{\max})/2$
    Compute Euler polygon using θ_1 and compute θ'_e
    $\Delta \leftarrow \theta_e - \theta'_e$
    if $|\Delta| \leq \epsilon$ **then**
        return θ_1 ▷ Converged
    end if
    if sign(Δ_{prev}) $\neq$ sign(Δ) **then**
        bracketed $\leftarrow$ True
    end if
    if not bracketed **then**
        ▷ Phase 1: Expand search interval
        if $\Delta > 0$ **then**
            $\theta_{\max} \leftarrow \theta_{\max} + \Delta_{\text{step}}$
        else
            $\theta_{\min} \leftarrow \theta_{\min} - \Delta_{\text{step}}$
        end if
    else
        ▷ Phase 2: Standard bisection search
        if $\Delta > 0$ **then**
            $\theta_{\min} \leftarrow \theta_1$
        else
            $\theta_{\max} \leftarrow \theta_1$
        end if
    end if
    $\Delta_{\text{prev}} \leftarrow \Delta$
end while
return Failure ▷ Target not found

Note that in Step 1, the value of n can be set arbitrarily, provided $n \geq 3$. For instance, by setting $n = 20$, more linear curvature plots can be obtained. In Step 3, we employ the extended bisection search described in Algorithm 1 to find the optimal value of θ_1 such that $\theta'_e = \theta_e$. As illustrated in Fig. 6(a), θ'_e typically varies monotonically with respect to θ_1. Due to this predictable relationship, the extended bisection search effectively locates the θ_1 value that satisfies $\theta_e = \theta'_e$. However, Fig. 6(b) illustrates an instance of non-monotonic behavior where Algorithm 1 fails. Such behavior generally occurs when $|\theta_d|$ is large.

To address cases where Algorithm 1 fails with an initial guess of $\theta_1 = 0°$, we employ the following heuristic. Suppose we are given $\theta_d = -420°$ and $\theta_e = -100°$. Referring to Fig. 6(b), Algorithm 1 might yield $\theta_1 = \hat{\theta}_1$ and $\theta'_e = -40°$, resulting in $\theta_e - \theta'_e < 0$. Let $\hat{\theta}_e = \theta'_e = -40°$. Because $\hat{\theta}_1$ is negative, we incrementally decrease θ_1 (e.g., in steps of $3°$). If a slight decrease causes $\theta'_e - \hat{\theta}_e$ to become positive, we continue to incrementally decrease θ_1 until $\theta'_e - \hat{\theta}_e$ becomes negative. Conversely, if $\hat{\theta}_1$ is positive, we incrementally increase θ_1 instead. Finally, we use this adjusted θ_1 as a new initial guess for Algorithm 1 to find the precise θ_1 such that $\theta'_e = \theta_e$.



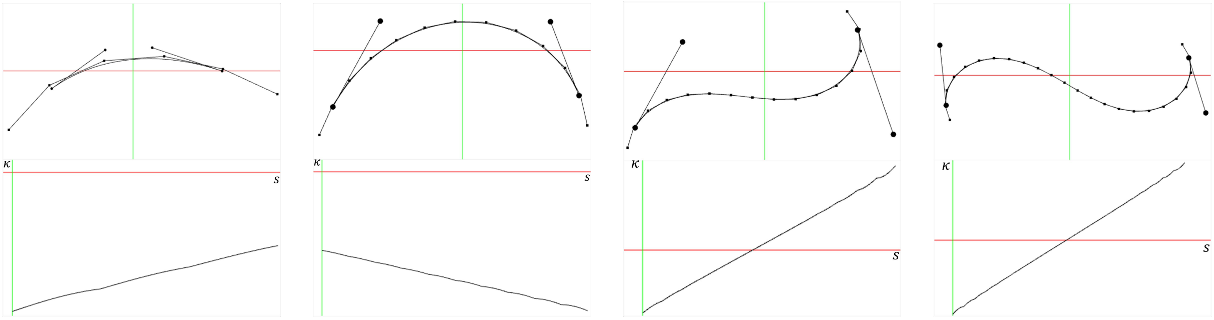
(a) Ideal monotonic behavior ($\theta_d = -120°$). (b) Non-monotonic behavior ($\theta_d = -420°$).

Figure 6: Variation of θ'_e depending on θ_1 $(n = 20)$.

5 RESULTS

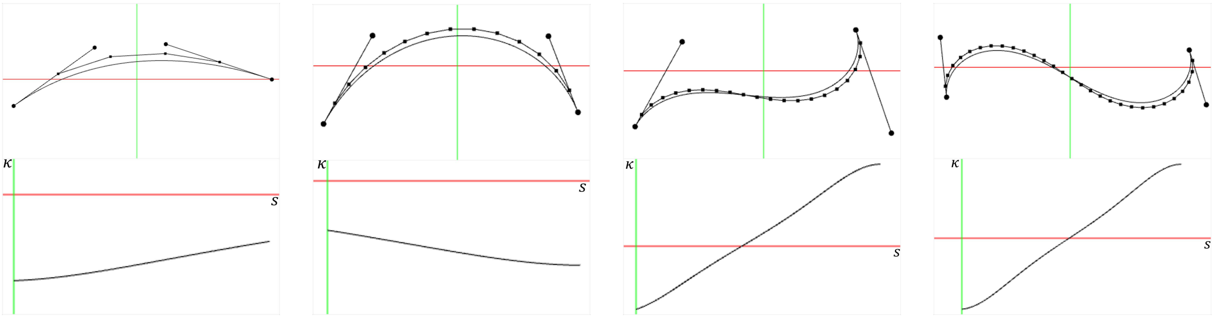
5.1 Comparison with Yang's Euler B-spline Spirals

Fig. 7 illustrates Euler B-spline spirals generated using the proposed method for $n = 5, 12, 15$, and 23. For comparison, we also implemented Yang's smoothing process, setting the edge lengths of the Euler polygon to be equal within a tolerance of 10^{-6}. We confirmed that the curves generated by both Yang's method and the proposed method are identical in all four cases shown in Fig. 7, under the same control point distance criterion ($< 10^{-5}$). The bottom panels of Fig. 7 plot the curvature κ against the arc length s. Note that both κ and s are scaled to fit in the plotting area. The curvature profile tends toward linearity as n increases. Fig. 8 shows Euler Bézier spirals [9] and their corresponding curvature plots. The plots exhibit non-linear behavior near the endpoints ($t = 0$ and $t = 1$), even for large values of n. Consequently, we conclude that cubic Euler B-spline spirals are preferable; they maintain a low polynomial degree while offering a curvature profile that is more linear than that of Euler Bézier spirals. Table 1 compares the computation times required to generate the Euler B-spline spirals shown in Fig. 7 using the proposed method versus Yang's smoothing process. The computational performance was evaluated on a desktop system equipped with an AMD Ryzen 7 8700G (4.2 GHz) CPU and 32 GB of DDR5 memory. As the results indicate, the proposed method exhibits significantly higher computational efficiency than Yang's approach, particularly as n increases. Moreover, as demonstrated in Fig. 1, there are cases where Yang's method fails to produce valid Euler spirals, whereas our proposed method succeeds.



(a) $n = 5$ (b) $n = 12$ (c) $n = 15$ (d) $n = 23$

Figure 7: Euler B-spline spirals using the proposed method and their curvature plots for $n = 5, 12, 15$, and 23.



(a) $n = 5$ (b) $n = 14$ (c) $n = 21$ (d) $n = 30$

Figure 8: Euler Bézier spirals using the proposed method and their curvature plots for $n = 5, 14, 21$, and 30. Even for large n, the curvature profile exhibits non-linear behavior near the start and endpoints.

Table 1: Computation time of Euler B-spline spirals.

n	Yang's method (ms)	Proposed method (ms)
5	0.019	0.014
12	0.438	0.062
15	1.228	0.114
23	10.500	0.166

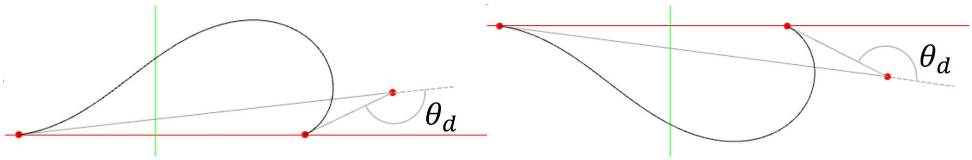


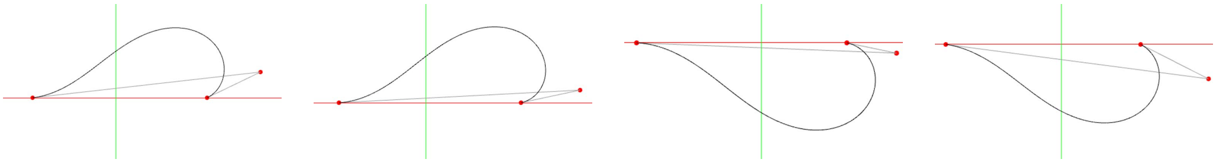
Figure 9: θ_d in two distinct configurations resulting from the use of the $\arctan$ function, which returns the value in the interval $(-\pi, \pi]$.

5.2 Mitigating Popping Effects

When generating Euler B-spline spirals using the proposed method interactively, a phenomenon termed the "popping effect" can occur. To clearly illustrate this popping effect, we employ a curve control method based on G^1 Hermite interpolation specified by three points like the one in [14]. Fig. 9 illustrates θ_d in two distinct configurations resulting from the use of the $\arctan$ function (specifically atan2), which returns values in the interval $(-\pi, \pi]$.

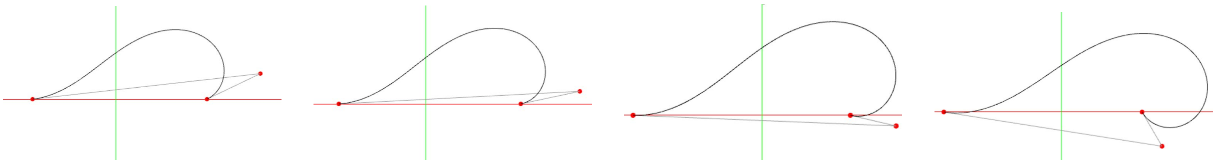
Fig. 10 illustrates a sequence of generated Euler B-spline spirals as θ_d varies. As the input points are continuously adjusted, the curve initially transitions smoothly from $\theta_d = -160°$ in Fig. 10(a) to $-170°$ in Fig. 10(b). However, as the manipulation continues, the calculated angle abruptly jumps to $170°$ in Fig. 10(c), and then proceeds to $160°$ in Fig. 10(d). This sudden jump between Fig. 10(b) and (c) occurs because the underlying angle calculation crosses the wrapping boundary at $\pm 180°$. Consequently, this transition from $-170°$ to $170°$ represents a numerical discontinuity in θ_d, even though the physical manipulation simply crosses $-180°$. During interactive curve generation, such discontinuous angular changes cause the resulting curve's shape to abruptly shift, creating a disruptive popping effect.

To resolve this popping issue, we modify the generation of an Euler B-spline spiral by tracking θ_d to ensure it remains continuous with its previously calculated values. Fig. 11 shows the result of this adjustment, where θ_d smoothly transitions through $-160°$ in Fig. 11(a), $-170°$ in Fig. 11(b), $-190°$ in Fig. 11(c), and $-230°$ in Fig. 11(d). By maintaining this numerical continuity, the popping effect is successfully suppressed.



(a) $\theta_d = -160°$ (b) $\theta_d = -170°$ (c) $\theta_d = 170°$ (d) $\theta_d = 160°$

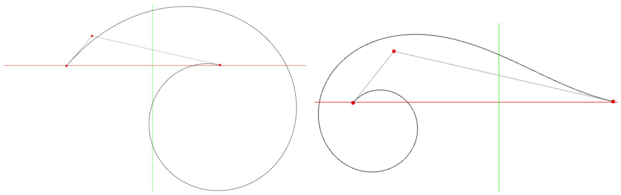
Figure 10: Popping effects of Euler B-spline spirals caused by abrupt change in θ_d.



(a) $\theta_d = -160°$ (b) $\theta_d = -170°$ (c) $\theta_d = -190°$ (d) $\theta_d = -230°$

Figure 11: Popping effects mitigated by making θ_d continuous.

Nevertheless, enforcing the continuity in θ_d does not completely eliminate the popping phenomenon. As illustrated in Fig. 12, a popping effect can still occur. It should be noted that this secondary popping is not ubiquitous; rather, it is triggered when θ_d becomes large, corresponding to complex spiral shapes. As an illustrative example, Fig. 12(a) shows a curve generated at $\theta_d = -422^\circ$. A slight manipulation of the control points updates this angle to $\theta_d = -424^\circ$, as shown in Fig. 12(b). Despite this small, continuous change in the parameter, the resulting curve abruptly jumps. Note that this popping effect is related to the non-monotonic behavior of θ'_e shown in Fig. 6.



(a) $\theta_d = -422^\circ$ (b) $\theta_d = -424^\circ$

Figure 12: Example of a popping effect observed despite a continuous θ_d. This phenomenon stems from the non-monotonic behavior shown in Fig. 6.

5.3 Curvature Monotonicity Evaluation Using the CMEF

To confirm that curvature monotonicity holds when Yang's necessary condition (Eq. (5)) is satisfied, we apply the Curvature Monotonicity Evaluation Function (CMEF) proposed by Saito et al. [8] for cubic Euler B-spline spirals.

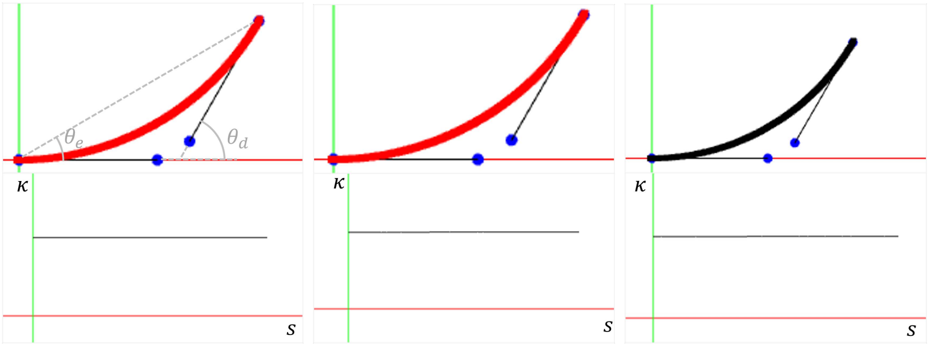
During our extensive interactive generation process, no discrepancies were observed between the evaluation results of Yang's necessary condition and the CMEF. In all cases where Yang's condition was satisfied, the CMEF simultaneously confirmed strict curvature monotonicity. Furthermore, we conducted identical tests on Euler Bézier spirals and observed no discrepancies.

For instance, given $\theta_d = 60^\circ$ and $\theta_e = 30^\circ$ as shown in Fig. 3, the only solution with monotonically varying curvature is a circular arc, which cannot be represented exactly by polynomial curves. Fig. 13(a) illustrates the resulting curve under these conditions. Both Yang's necessary condition and the CMEF classify the curvature as non-monotonic; therefore, the curve is plotted in red. If we slightly decrease θ_e to 29.981° (Fig. 13(b)), the curve deviates from a perfect circle, yet it still fails to achieve strict curvature monotonicity under either evaluation method. When θ_e is decreased slightly further to 29.980° (Fig. 13(c)), creating a near-circular arc, both Yang's necessary condition and the CMEF confirm that the curvature is monotonically varying, and the curve is consequently drawn in black. Throughout these subtle parameter transitions near the circular limit, the evaluations of Yang's necessary condition and the CMEF aligned perfectly.

6 CREATING ILLUSTRATIONS USING EULER B-SPLINE SPIRALS

We created an interactive illustration tool implemented using the proposed Euler B-spline spirals, inspired by the Pen tool in Adobe Illustrator. This tool allows users to interactively manipulate curves using anchors and tangent handles, enabling the intuitive creation of illustrations with guaranteed G^1 continuity.

Fig. 14 demonstrates a bird illustration recreated using our proposed Euler B-spline spirals, inspired by the design originally presented by Yan et al. [12]. The curvature comb for the section highlighted in red in Fig. 14 is illustrated in Fig. 15. Similarly, Fig. 16 shows a shoe illustration crafted entirely from Euler B-spline spirals. The curvature comb for the corresponding red-highlighted portion in Fig. 16 is displayed in Fig. 17.



(a) $\theta_e = 30°$, $n = 149$. (b) $\theta_e = 29.981°$, $n = 149$. (c) $\theta_e = 29.980°$, $n = 143$.

Figure 13: Evaluation of curvature monotonicity for B-spline curves near a circular arc ($\theta_d = 60°$). The evaluations of Yang's necessary condition and the CMEF aligned perfectly.

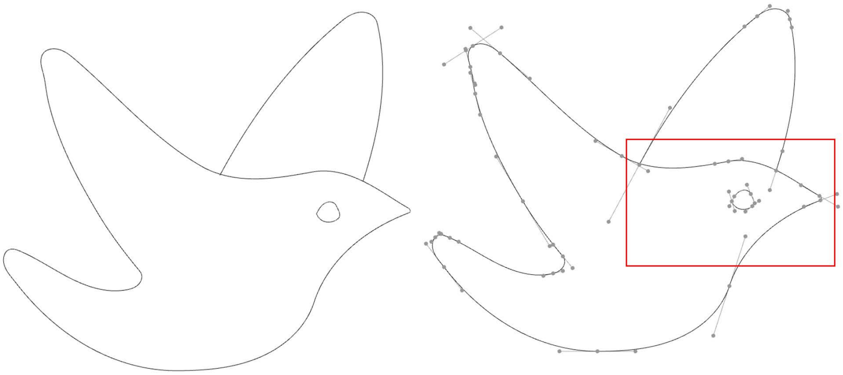


Figure 14: A bird created using Euler B-spline spirals.

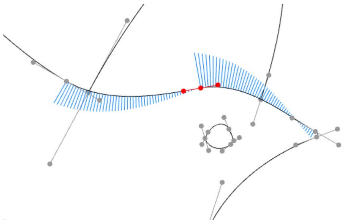


Figure 15: Curvature comb of the bird in the area shown in red.

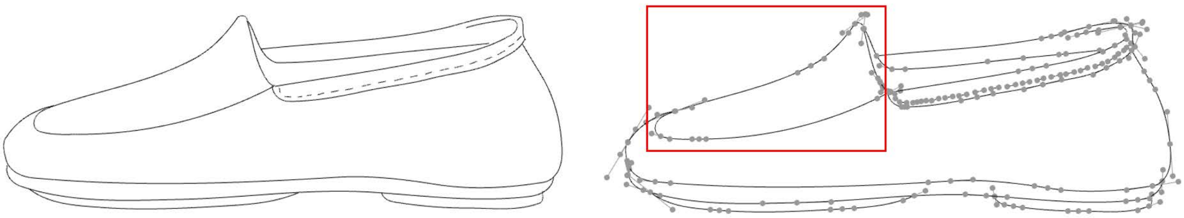


Figure 16: A shoe created using Euler B-spline spirals.

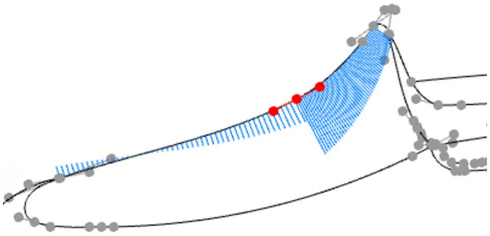


Figure 17: Curvature comb of the shoe in the area shown in red.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an efficient and robust algorithm for generating and rigorously evaluating Euler B-spline spirals. Our approach is significantly faster than Yang's method and successfully generates valid spirals in instances where Yang's method fails. Furthermore, we mitigated the popping effect by resolving the numerical discontinuity associated with the angular parameter θ_d, while also identifying complex edge cases where secondary popping occurs even when this parameter is continuous.

To confirm curvature monotonicity, we employed the Curvature Monotonicity Evaluation Function (CMEF). Through extensive experimentation, including critical edge cases that closely approximate circular arcs, we demonstrated complete agreement between Yang's necessary condition and the CMEF. To validate the practical utility of our approach, we developed an interactive illustration tool, confirming that the proposed Euler B-spline spirals can be effectively utilized to design aesthetically pleasing shapes.

Despite these significant advancements, several avenues for future research remain. First, while we confirmed complete empirical agreement between Yang's condition and the CMEF, establishing a rigorous mathematical proof of this equivalence remains an important goal. Second, although we mitigated the primary wrapping discontinuity, developing an interactive manipulation framework entirely free from secondary popping effects is highly desirable. Third, formulating a quantitative metric to evaluate the linearity of these curvature profiles is an area for future work. Finally, we aim to extend the proposed generation framework to broader engineering and industrial applications where smooth spirals are in high demand, such as highway geometric design, robotic route planning, and trajectory generation for autonomous vehicles.

ACKNOWLEDGEMENTS

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] Bertolazzi, E.; Frego, M.: G^1 fitting with clothoids. Mathematical Methods in the Applied Sciences, 38(5), 881–897, 2015. http://doi.org/10.1002/mma.3114.

[2] Farin, G.: Curves and Surfaces for CAGD: A Practical Guide (5th ed.). Morgan Kaufuman, 2002.

[3] Farin, G.: Class A Bézier curves. Computer Aided Geometric Design, 23(7), 573–581, 2006. http://doi.org/10.1016/j.cagd.2006.03.004.

[4] Havemann, S.; Edelsbruner, J.; Wagner, P.; Fellner, D.: Curvature-controlled curve editing using piecewise clothoid curves. Computers & Graphics, 37, 764–773, 2013. http://doi.org/10.1016/j.cag.2013.05.017.

[5] Levien, R.L.: From Spiral to Spline: Optimal Techniques in Interactive Curve Design. Ph.D. thesis, University of California, Berkeley, 2009.

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

[7] Piegl, L.; Tiller, W.: The NURBS Book. Springer, 2nd ed., 1997. http://doi.org/10.1007/978-3-642-59223-2.

[8] Saito, T.; Yoshida, N.: Curvature monotonicity evaluation functions on rational Bézier curves. Computers & Graphics, 114, 219–229, 2023. http://doi.org/10.1016/j.cag.2023.05.019.

[9] Sasaki, G.; Inoue, T.; Yoshida, N.; Saito, T.: Fast G^1 Hermite Interpolation of Euler Bézier/B-spline Spirals. In Proceedings of the CAD'26 Conference, 2026. http://doi.org/10.14733/cadconfP.2026.1-6.

[10] Scheuer, A.; Fraichard, T.: Continuous-curvature path planning for car-like vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1997.

[11] Wang, L.Z.; Miura, K.T.; Nakamae, E.; Yamamoto, T.; Wang, T.J.: An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves. Computer-Aided Design, 33, 1049–1058, 2001. http://doi.org/10.1016/S0010-4485(00)00142-1.

[12] Yan, Z.; Schiller, S.; Wilensky, G.; Carr, N.; Schaefer, S.: κ-curves: Interpolation at local maximum curvature. ACM Transactions on Graphics (TOG), 36(4), 1–7, 2017. http://doi.org/10.1145/3072959.3073692.

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

[14] Yoshida, N.; Saito, T.: Interactive aesthetic curve segments. The Visual Computer, 22(9-11), 896–905, 2006. http://doi.org/10.1007/s00371-006-0076-5.

[15] Yoshida, N.; Saito, T.: Quasi-aesthetic curves in rational cubic Bézier forms. Computer-Aided Design & Applications, 4(1-4), 477–486, 2007. http://doi.org/10.1080/16864360.2007.10738567.