

Smooth Interpolating Curves with Local Control and Monotone Alternating Curvature

Alexandre Binninger^{ID}

Olga Sorkine-Hornung^{ID}

ETH Zurich, Switzerland

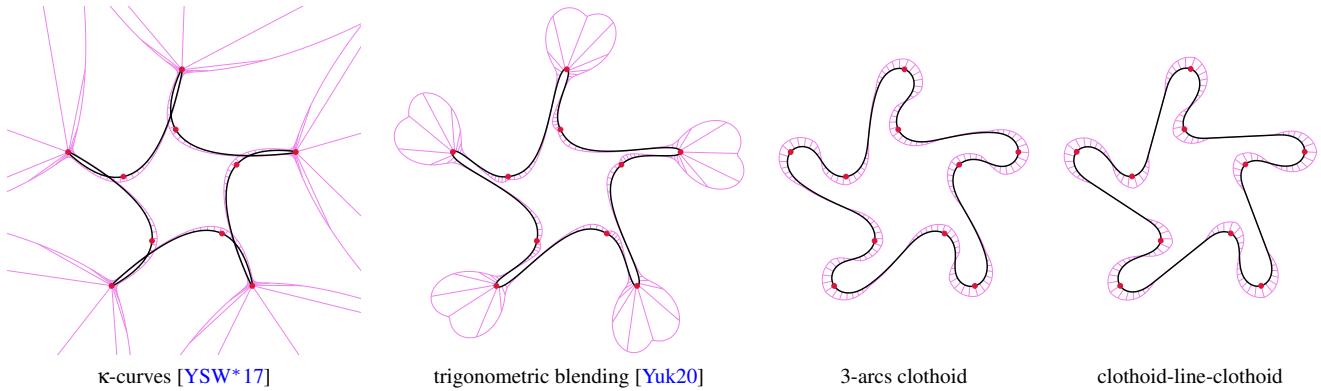


Figure 1: Interpolating curves generated from the same control points (shown in red) using κ -curves [YSW* 17], trigonometric blending [Yuk20], our 3-arcs clothoids method and our clothoid-line-clothoid method. Our approach guarantees G^2 continuity, has bounded local support and provides curvature monotonicity between control points of opposite curvature sign. The curvature normals are visualized with purple lines.

Abstract

We propose a method for the construction of a planar curve based on piecewise clothoids and straight lines that intuitively interpolates a given sequence of control points. Our method has several desirable properties that are not simultaneously fulfilled by previous approaches: Our interpolating curves are C^2 continuous, their computation does not rely on global optimization and has local support, enabling fast evaluation for interactive modeling. Further, the sign of the curvature at control points is consistent with the control polygon; the curvature attains its extrema at control points and is monotone between consecutive control points of opposite curvature signs. In addition, we can ensure that the curve has self-intersections only when the control polygon also self-intersects between the same control points. For more fine-grained control, the user can specify the desired curvature and tangent values at certain control points, though it is not required by our method. Our local optimization can lead to discontinuity w.r.t. the locations of control points, although the problem is limited by its locality. We demonstrate the utility of our approach in generating various curves and provide a comparison with the state of the art.

CCS Concepts

- Computing methodologies → Parametric curve and surface models;

1. Introduction

Curve modeling is a fundamental field of computer graphics and is at the cornerstone of computer-aided geometric design [FF02], spanning over domains as diverse as sketching, shape representation, drawing, shape completion, curve fitting and animation. One major distinction in curve modeling lies between approximation

and interpolation, the latter requiring the curve to pass through the control points, which is desirable in many applications. Our work focuses on the interpolation of a sequence of planar points (a control polygon) without parameterization constraints.

There are infinitely many curves passing through a set of points, thus the interpolation problem is by nature ill posed. In practice,

various mathematical properties and guarantees are desired in applications, enabling characterization of different interpolation methods and construction of new ones. One such major property is *continuity*, which measures the smoothness of a curve. If a curve is parameterized such that its derivatives up to order n exist and are continuous, we say that the curve is C^n . A curve is said to be G^n geometrically continuous if its arc length reparameterization is C^n . Second order geometric continuity is generally considered necessary for producing fair planar curves. Another important property is locality: if moving a control point modifies the curve shape within a bounded number of neighboring control points, the curve is said to have *local support*. Preventing unintended *self-intersections* or *cusps* is another component of shape fairness. Levien [Lev09] discusses such properties in detail.

There is no conclusive evidence that accumulating mathematical properties is directly related to designing fair and good looking curves. The fairness of a curve remains a poorly mathematically defined concept [LS09], but there is a strong indication based on psychological science that curvature variation plays a major role in our perception of a shape [Att74, Zia16]. This advocates for curves where curvature extrema are located at control points, implying monotonicity of curvature between two consecutive control points.

The curves we produce are formed by stitching clothoids and straight lines in a G^2 manner. We start by locally estimating the tangent and curvature objectives at each control point and then locally modify the curvature to enable the transition between two consecutive control points. With regards to the aforementioned properties, our contribution consists in producing a curve parameterized by arc-length with guarantees that have not previously been ensured simultaneously, to our knowledge: G^2 -continuity, local support of 6 or 8 points, monotone curvature between control points with alternate curvature sign, curvature extrema at control points and no self-intersection if the control polygon does not present a self-intersection between the respective control points. Moreover, our method also permits further modeling options, since the user can manually set curvature and tangent constraints at control points. Yet, solving local optimization problems and enforcing the consistency of the curvature sign with the control polygon brings one major drawback: the lack of continuity with respect to the locations of control points. We compare our approach to the state of the art and discuss its advantages and limitations. Our code is publicly available [here](#) to foster further research in this field.

2. Related work

A substantial body of work on interpolating curve construction exists. We only present a subset of this rich domain and refer the reader to [HL93] and [CDH^{*}07] for a more detailed presentation.

Interpolation methods. Catmull-Rom splines [CR74] provide a G^2 interpolation method based on local Lagrange interpolations blended via Bézier curves and B-splines. Catmull-Rom splines with centripetal parameterization do not produce cusps or self-intersections within the curve segments [YSK09]. Interpolating n points via G^2 cubic splines [FF02] can be done by solving a linear system to find $n+2$ control points of a corresponding B-spline. Contrary to our work, these methods take as input the parameter values t_i such that

the resulting curve α interpolates the control points p_i on these values, i.e., $\alpha(t_i) = p_i$. Our method does not require such additional constraints. Note that the distinction between C^2 and G^2 continuity is irrelevant without parameterization constraints, since any G^2 curve has a C^2 parameterization by definition.

Several methods use blending functions without parameterization constraints. Linear functions can be used to blend circular interpolations of triplets of consecutive control points [Wen96], but the result is only G^1 continuous. Trigonometric blending functions raise the continuity to G^2 [SNV00]. These curves reproduce exact circles if four consecutive points are lying on a circle, but are prone to self-intersections, cusps and unbounded distance to the control polygon. Using trigonometric blending directly on the tangent angles estimated by circle fitting at control points [SLY05] produces cusps-free curves. A scheme based on quadratic and trigonometric functions blending local conic interpolation produces curves compatible with nonuniform rational basis splines [SZ09]. Trigonometric blending of hybrid circular-elliptical interpolations [Yuk20] produces curves without cusps or self-intersections between two consecutive control points, but does not prevent self-intersections between different curve segments. In addition, these curves are G^2 continuous and have bounded distance to the control polygon. Polynomial blending functions with vanishing derivatives at the boundary produce curves of arbitrarily high order of continuity [Pob13]. All such blending methods provide continuity with local support and are easy to compute, but fail to ensure piecewise curvature monotonicity and curvature extrema at control points.

Methods producing curves with curvature extrema at control points usually rely on global optimization techniques and do not have local support. Havemann et al. [HEWF13] propose a global iterative algorithm based on piecewise clothoids. They estimate curvature based on circle interpolation of neighbouring points and use it to add a new point between each control point. The curvature information is therefore globally propagated, and this method does not provide local support. Like in our method, the control polygon should be sparse in order to offer intuitive user control and avoid artifacts. Another proposal are κ -curves [YSW^{*}17]: these are piecewise-quadratic curves based on an iterative optimization process to set Bézier control points, which produces G^2 splines except at inflection points. Miura et al. [MGS^{*}21] extend the concept of κ -curves to $\epsilon\kappa$ -curves by using cubic instead of quadratic Bézier curves. This provides user control over the magnitude of the curvature extrema at the input points and can reduce the gap in curvature at inflection points. Yan et al. [YSS19] can additionally reproduce circles and ellipses by using rational quadratic Bézier interpolation primitives and the additional degree of freedom to minimize eccentricity of interpolating conics. A similar optimization process to κ -curves can be used on log-aesthetics interpolating curves [WGS^{*}20] to produce G^2 curves. Bézier curves can also produce splines with monotone curvature under certain conditions [Far06]. In contrast to this type of techniques, our method has both local support and curvature monotonicity between consecutive control points when the control polygon implies curvature of alternating sign.

Clothoids. Clothoids are curves whose curvature varies linearly with arc length. They present particularly beneficial fairness properties for transition curves, which justifies their use in a large va-

riety of domains [Lev09], ranging from mechanical engineering of road design [MJF15] to shape completion [KFP03] or sketching [MS09, BLP10]. Fitting a G^2 curve using piecewise clothoids is possible using global optimization with nonlinear programming [Sto82]. Bertolazzi and Frego [BF18a] obtain a clothoid-based G^2 spline by building a nonlinear system with two degrees of freedom. With these additional degrees of freedom, they globally optimize over a set of nine problems including boundary conditions, minimization of length, and integrated curvature. Similar to other global optimization methods, this approach lacks local support.

Our method is based on the evaluation of tangents and curvatures at control points and fitting a piecewise clothoid according to these conditions. Interpolating two given points with assigned tangent vectors is always possible [WM09, BF15] and is referred to as the Hermite G^1 interpolation problem. Finding a clothoid curve to join two points with tangent and curvature constraints is not necessarily feasible, but the set of points that allows a Hermite G^2 transition via monotone piecewise clothoids has been explicitly determined [MW92]. Without the curvature monotonicity condition, it is always possible to find a 3-arcs clothoid transition that solves the Hermite G^2 problem [BF18b].

Our method relies on several research tasks around clothoids. Finding the shortest transition clothoid from a point with second order constraints to a line with a target angle is explicitly derived in [VMCU16] and is at the foundation of our clothoid-line-clothoid transition method. We also rely on the detection of intersections between clothoids [BBF20] to prevent self-intersections that are inconsistent with the control polygon. Clothoids formulation can be framed within Fresnel integrals, which are transcendental functions [ASR66], but extensive research has been conducted to ease their computation. Fresnel integrals can be approximated via rational functions [Hea85] or s-power series [SRC03], which are much easier to compute. A comparison between the explicit formulas based on Taylor expansion and an alternative method based on the explicit Euler method can be found in [VMCU16].

3. Background

We set the notation and review the necessary concepts related to clothoids. We denote by $p_i \in \mathbb{R}^2$ the n input points. The curve's tangent at p_i is denoted as T_i and the curvature as K_i . We can represent T_i by the angle θ_i it makes with the x -axis. The *control polygon* is the set of segments generated by the ordered control points, i.e., the union of segments $[p_i, p_{i+1}]$.

3.1. Clothoids

Clothoids are curves whose curvature variation is linear w.r.t. arc length. Perusing the notation in [BF15, BF18b, BF18a], if $\alpha(t)$ is a curve parameterized by arc length, then α is a clothoid if its curvature is of the form

$$K(t) = \kappa' t + \kappa_0, \quad (1)$$

where κ' and κ_0 are constants. Let $T(t) = (\cos \theta(t), \sin \theta(t))^T$ be the tangent vector field of the curve. Since the derivative of the

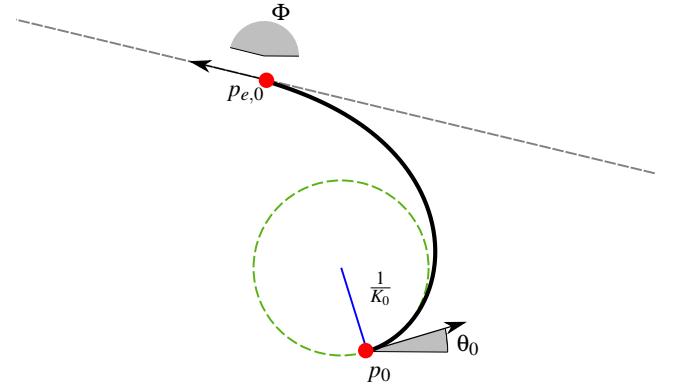


Figure 2: Clothoid-line transition problem: find a clothoid starting at point p_0 with starting tangent angle θ_0 and curvature K_0 that makes a G^2 -smooth transition to a straight line of target angle Φ . Note that the target is the angle Φ , not a precise spatial positioning of the line. The G^2 condition means that the curvature of the clothoid at the endpoint $p_{e,0}$ must be zero.

tangent's slope angle is the curvature, $\theta'(t) = K(t)$, we can write

$$T(t) = \begin{pmatrix} \cos(\frac{\kappa'}{2}t^2 + \kappa_0 t + \theta_0) \\ \sin(\frac{\kappa'}{2}t^2 + \kappa_0 t + \theta_0) \end{pmatrix},$$

and the expression of the curve is recovered as

$$\alpha(t) = \begin{pmatrix} x_0 + \int_0^t \cos(\frac{\kappa'}{2}u^2 + \kappa_0 u + \theta_0) du \\ y_0 + \int_0^t \sin(\frac{\kappa'}{2}u^2 + \kappa_0 u + \theta_0) du \end{pmatrix}. \quad (2)$$

Clothoids are often studied within the frame of Fresnel integrals

$$C(t) = \int_0^t \cos\left(\frac{\pi}{2}u^2\right) du \text{ and } S(t) = \int_0^t \sin\left(\frac{\pi}{2}u^2\right) du.$$

3-arcs clothoids. Given boundary conditions (p_0, θ_0, K_0) and (p_1, θ_1, K_1) for the start and end points, it is always possible to find a transition made of three pieces of clothoids connected in a G^2 manner that matches these conditions [BF18b]. We refer to this as the *3-arcs clothoid transition*.

3.2. Clothoid shell

Part of our method is based on finding a smooth clothoid-line-clothoid (CLC) transition. For this purpose, we need to compute the shortest clothoid that makes a G^2 transition between a starting point with boundary conditions (p_0, θ_0, K_0) and a straight line, such that the tangent angle with the x -axis of the clothoid at the transition point to the line is Φ . See Fig. 2 for an illustration. This problem is studied in [VMCU16], and we summarize the results here. Assume w.l.o.g. that $K_0 > 0$. Let L be the length of the clothoid. A G^2 transition to a straight line implies that the end curvature vanishes: $K(L) = 0$, and from the linear expression of clothoid curvature (Eq. (1)) we get $\kappa'L = -\kappa_0$. The condition $\theta(L) = \Phi$ implies $\Phi = \frac{\kappa'}{2}L^2 + \kappa_0 L + \theta_0$, i.e., $L = 2(\Phi - \theta_0)/\kappa_0$.

The end point of the clothoid can be found by substituting the expression for L in Eq. (2). Given the relations between L , κ_0 and

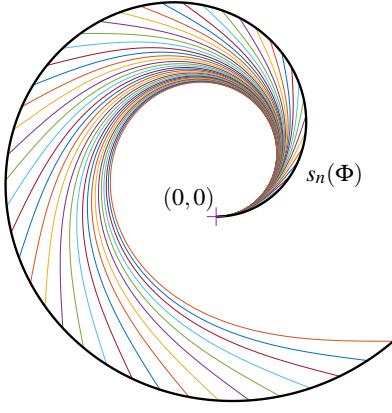


Figure 3: Normal clothoid shell (black curve) for $\Phi \in [0, 2\pi]$. The colored curves are the shortest clothoids emanating from the starting condition at the origin, namely $(p_0, \theta_0, K_0) = ((0,0)^T, 0, 1)$ and connecting with G^2 smoothness to a straight line of slope angle Φ .

κ' , and substituting the variable u in the integral in Eq. (2) with a parameter $s \in [0, 1]$, such that $u = Ls$, one gets

$$\int_0^L \cos\left(\frac{\kappa'}{2}u^2 + \kappa_0 u + \theta_0\right) du = \\ \frac{2(\Phi - \theta_0)}{\kappa_0} \int_0^1 \cos\left((\Phi - \theta_0)(-s^2 + 2s) + \theta_0\right) ds.$$

The same derivation applies for the second Fresnel integral. Therefore, the end point $p_{e,i}(\Phi)$ found for a clothoid-line transition with starting conditions (p_i, θ_i, K_i) and target angle $\Phi \in [\theta_i, \theta_i + 2\pi]$ is

$$p_{e,i}(\Phi) = p_i + \frac{2(\Phi - \theta_i)}{K_i} \left(\int_0^1 \cos\left((\Phi - \theta_i)(-u^2 + 2u) + \theta_i\right) du \right. \\ \left. \int_0^1 \sin\left((\Phi - \theta_i)(-u^2 + 2u) + \theta_i\right) du \right) \quad (3)$$

The following function $s_n : [0, 2\pi] \rightarrow \mathbb{R}^2$ is termed the *normal clothoid shell*, see Fig. 3:

$$s_n(\Phi) = 2\Phi \left(\int_0^1 \cos(\Phi(-u^2 + 2u)) du \right. \\ \left. \int_0^1 \sin(\Phi(-u^2 + 2u)) du \right). \quad (4)$$

It maps the angle Φ to the end point of the shortest clothoid starting at the origin with tangent slope angle 0 and curvature 1 and G^2 -smoothly transitioning to a straight line of slope angle Φ . Note that $p_{e,i}(\Phi)$ can be obtained for $\Phi \in [\theta_i, \theta_i + 2\pi]$ using the normal clothoid shell $s_n(\Phi)$ via a translation by p_i , scaling by $1/K_i$, rotation R_{θ_i} by angle θ_i and shifting by $-\theta_i$:

$$p_{e,i}(\Phi) = p_i + \frac{1}{K_i} R_{\theta_i}(s_n(\Phi - \theta_i)).$$

3.3. Clothoid-line-clothoid (CLC) transition

The clothoid shell provides us with a clothoid-line transition, namely a G^2 interpolation of a starting point with prescribed tangent and curvature values and end point tangent and vanishing curvature conditions, but not the end point *location*. For our curve interpolation

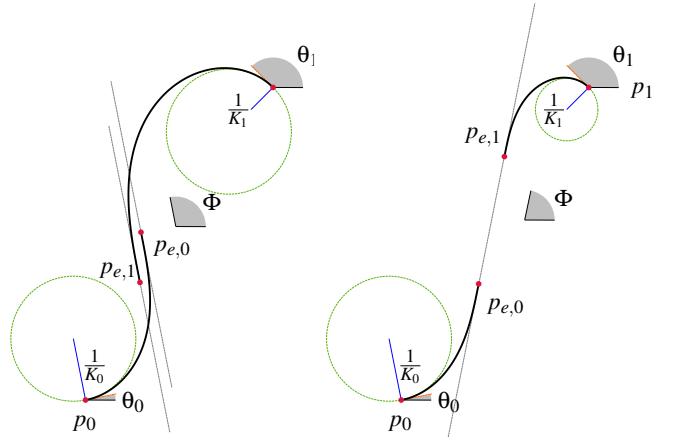


Figure 4: A CLC-transition is not guaranteed to exist. On the left, we show a case where the boundary conditions at the endpoints prevent finding a single straight line to which two clothoid pieces could connect G^2 -smoothly. On the right we show a feasible CLC configuration.

task, we are interested in smoothly connecting control points by pieces of clothoids and straight lines, hence we are seeking a G^2 clothoid-line-clothoid (CLC) transition, where the clothoid pieces connect smoothly to the same straight line. A smooth CLC transition is not guaranteed to exist, see Fig. 4. However, we can prove the following statement:

Proposition 3.1 Let p_0, p_1 be two distinct points and θ_0 and θ_1 two angles. There exist curvature bounds K_0, K_1 , such that for any $\kappa_0 \geq K_0, \kappa_1 \geq K_1$, there exists an angle Φ such that the line segment $[p_{e,0}(\Phi), p_{e,1}(\Phi + \pi)]$ makes an angle Φ with the x -axis, where the starting conditions for the two clothoids are $(p_0, \theta_0, \kappa_0)$ and $(p_1, \theta_1, \kappa_1)$, respectively.

Proof See appendix. \square

Hence by prescribing sufficiently high curvature at control points, Proposition 3.1 ensures the existence of a transition consisting of a clothoid from p_0 to $p_{e,0}(\Phi)$, a straight line from $p_{e,0}(\Phi)$ to $p_{e,1}(\Phi + \pi)$, and a clothoid from $p_{e,1}(\Phi + \pi)$ to p_1 . Note that we prove the existence of an angle Φ that is equal to the angle of the tangent at $p_{e,0}$ and to the angle of the tangent at $p_{e,1}$ plus π , that is to say, the existence of a CLC transition where the end points of the clothoids are not in a reversed order on the transition line. This transition can be as close as desired to the straight segment $[p_i, p_{i+1}]$ by taking sufficiently high curvatures, as shown below.

Proposition 3.2 Let p_0, p_1 be two distinct points and θ_0, θ_1 two angles. If the absolute value of κ_0 and κ_1 simultaneously tends to infinity, then the G^2 -smooth CLC transition between boundary conditions $(p_0, \theta_0, \kappa_0)$ and $(p_1, \theta_1, \kappa_1)$ uniformly converges to the segment $[p_0, p_1]$.

Proof For all $\Phi \in [0, 2\pi]$, applying Jensen's inequality, we have

$$\begin{aligned} \|p_{e,i}(\Phi + \theta_i) - p_i\| &= \frac{1}{|\kappa_i|} \|R_{\theta_i}(s_n(\Phi))\| \\ &\leq \frac{2\Phi}{|\kappa_i|} \sqrt{\int_0^1 (\cos^2(\Phi(-u^2 + 2u)) + \sin^2(\Phi(-u^2 + 2u))) du} \\ &= \frac{2\Phi}{|\kappa_i|} \leq \frac{4\pi}{|\kappa_i|}. \end{aligned}$$

If $|\kappa_i|$ tends to infinity, then $p_{e,i}(\Phi + \theta_i)$ converges to p_i for all Φ . Hence the end points of the clothoid parts of the CLC transition, $p_{e,0}$ and $p_{e,1}$, converge to p_0 and p_1 , whereby the clothoid pieces become increasingly shorter, and the line $[p_{e,0}, p_{e,1}]$ converges to $[p_0, p_1]$. \square

4. Method

We propose two interpolating methods. The first method is based on finding a transition composed of 3-arcs clothoids [BF18b], and the second one is based on G^2 clothoid-line-clothoid transition. Both rely on a local estimation of tangent and curvature values at control points (Sec. 4.1) followed by a curvature refinement procedure (Sec. 4.2) to achieve the desired curve properties. Our method is summarized in Fig. 5 and Algorithm 1. We provide the code in the supplemental material.

Algorithm 1: G^2 interpolating curve

```

Data:  $p_i \in \mathbb{R}^2$  for  $i \in [1, n]$ 
Result: curve  $C = \bigcup_i C_i$ ,  $C_i$  interpolates between  $p_i$  and  $p_{i+1}$ 
for  $i \in [1, n]$  do
     $s_i \leftarrow \text{CurvatureSign}(p_{i-1}, p_i, p_{i+1})$ 
     $I_i \leftarrow \text{HybridCircleEllipse}(p_{i-1}, p_i, p_{i+1})$ 
     $\theta_i \leftarrow \text{EstimateTangent}(I_i)$ 
end
for  $i \in [1, n]$  do
     $G_i \leftarrow \text{G1Clothoid}(p_i, \theta_i, p_{i+1}, \theta_{i+1})$ 
     $K_i \leftarrow \text{EstimateCurvature}(s_i, I_i, G_i)$ 
end
for  $i \in [1, n]$  do
     $|\hat{K}_i| \leftarrow$ 
     $\max(|K_i|, |\text{Refine}((p_{i-1}, \theta_{i-1}, K_{i-1}), (p_i, \theta_i, K_i))|)$ 
     $|\hat{K}_i| \leftarrow$ 
     $\max(|\hat{K}_i|, |\text{Refine}((p_i, \theta_i, K_i), (p_{i+1}, \theta_{i+1}, K_{i+1}))|)$ 
end
for  $i \in [1, n]$  do
     $C_i \leftarrow \text{ClothoidFitting}((p_i, \theta_i, \hat{K}_i), (p_{i+1}, \theta_{i+1}, \hat{K}_{i+1}))$ 
end

```

4.1. Tangent and curvature estimation at control point

Estimation of the tangent t_i and curvature K_i at a control point p_i is a key step in our method. This estimation must be done using a bounded number of consecutive points to keep local support. We start by applying the approach in [Yuk20] that locally fits circles or ellipses, relying on only 3 consecutive points, p_{i-1}, p_i, p_{i+1} . If the three points are collinear, the resulting interpolation is a line

segment; the curvature at p_i is then estimated to be 0 and the tangent is set to $t_i = (p_i - p_{i-1})/\|p_i - p_{i-1}\|$. In a general configuration, circles are uniquely determined by 3 points. For ellipses, the method adds the constraint that p_i should be on one of the two axes of the ellipse (*primary axis*), one of the neighboring control points p_{i-1} or p_{i+1} should be on the other axis (*secondary axis*), and the remaining point on the other side of the primary axis. These added constraints define the ellipse uniquely. The choice of whether to fit a circle or an ellipse depends on the angles corresponding to the circular arcs $\widehat{p_{i-1}p_i}$ and $\widehat{p_ip_{i+1}}$. We fit an ellipse if one of the angles is greater than $\pi/2$, and a circle otherwise. We refer to section 4 of [Yuk20] for a more detailed explanation.

We compute the tangent at a control point p_i from the local interpolation described above. It is possible to also take the curvature of the circle or ellipse interpolant as the initial guess, but such curvature values can lead to sub-optimal results. For example, as shown in Fig. 11, the local ellipse fitting can produce unnecessarily high curvature that hurts the curve's appearance. We propose using an alternative curvature estimation method via G^1 clothoid fitting. The curvature should be oriented in a consistent manner with the control polygon, i.e., the sign of K_i should be equal to the sign of the principal angle formed by vectors $p_i - p_{i-1}$ and $p_{i+1} - p_i$ (see Fig. 6). Since the G^1 Hermite interpolation is a good approximation of the G^2 Hermite interpolation problem [BF18b], we perform two G^1 clothoid fittings with endpoint locations and tangents constraints $(p_{i-1}, t_{i-1}, p_i, t_i)$ and $(p_i, t_i, p_{i+1}, t_{i+1})$ [BF15]. Let \tilde{K}_{i-1} and \tilde{K}_{i+1} be the resulting curvature of the G^1 fitting at p_i . We set the target curvature at p_i as the average $K_i = (\tilde{K}_{i-1} + \tilde{K}_{i+1})/2$. In case one of the two curvatures $\tilde{K}_{i-1}, \tilde{K}_{i+1}$ has opposite sign to the control polygon, we instead set K_i to be equal to the curvature estimated from the local circle/ellipse interpolation, whose sign is always consistent with the control polygon.

4.2. Curvature refinement

As we discuss in the following, the curvature estimated in the previous section does not necessarily allow for a solution that has monotone curvature (3-arcs clothoid) or even exists (CLC). Global optimization of the curvature prescribed at control points would remove the strong locality property. Fortunately, as we saw, starting from a certain threshold, increasing the prescribed curvature maintains the existence of a CLC-transition. For each point p_i , it is therefore possible to find the smallest curvatures that satisfy the desired properties for the segments (p_{i-1}, p_i) and (p_i, p_{i+1}) , and choose the curvature with the largest absolute value. In practice, we can always find the threshold curvatures for points with curvature of alternate signs, but this can fail for consecutive curvatures of same sign. Solutions to this issue are studied in Sec. 5.1. We describe our approach in more detail in Sections 4.2.1 and 4.2.2.

4.2.1. 3-arcs clothoid transition

Given two boundary conditions (p_i, θ_i, K_i) and $(p_{i+1}, \theta_{i+1}, K_{i+1})$, it is always possible to find a curve composed of 3 clothoids with a G^2 connection interpolating between these conditions [BF18b]. However the resulting curve does not necessarily possess a monotone curvature profile between two control points of opposite curvature

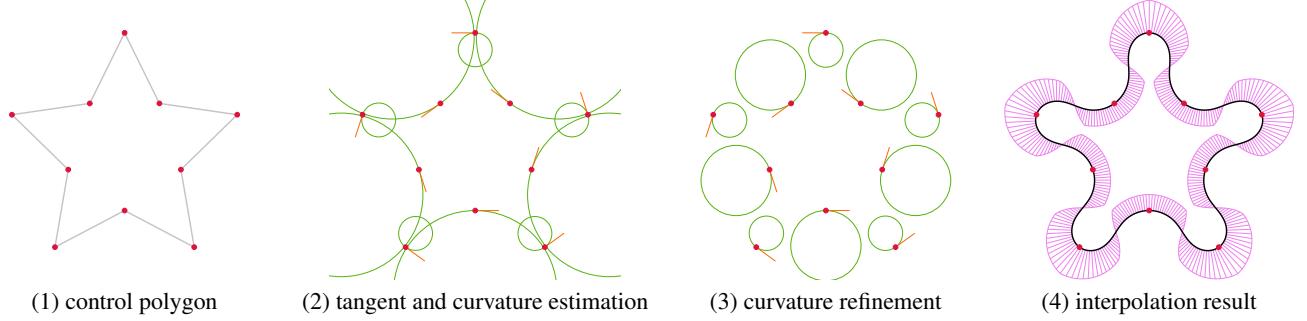


Figure 5: Summary of our method. (1) We start with an ordered set of points p_i . (2) We then estimate the tangent slope angle θ_i and curvature K_i at each control point (Sec. 4.1). (3) We apply our refinement method to increase the curvatures at control points so that a G^2 transition becomes feasible (Sec. 4.2). (4) Finally, we fit a transition curve between each pair of control points that matches the tangent and curvature conditions.

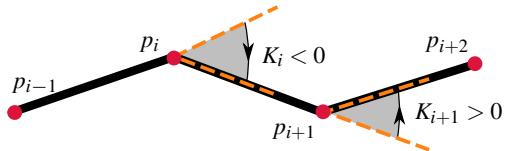


Figure 6: The sign of the curvature at control points is given by the control polygon.

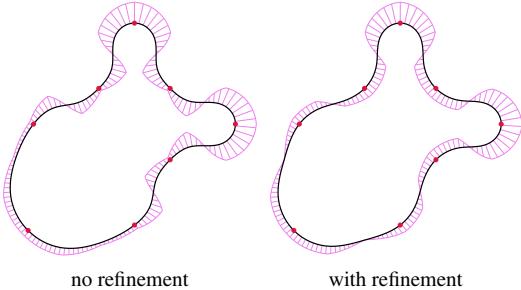


Figure 7: The 3-arcs clothoid transition without curvature refinement (left) and with curvature refinement (right). With our refinement, the curvature is monotone between control points. In this example, the initial curvatures were estimated using the hybrid circle-ellipse fitting of [Yuk20].

sign (see Fig. 7). Intuitively, if the curvature at control points is set sufficiently high, the curve has a monotone curvature profile.

For consecutive points with curvature of the same sign, there is no obvious way to deal with the curvature at control points without resorting to global optimization. If the curvature at a control point is too high, the curvature profile of the resulting curve can exhibit opposite sign to compensate. If the curvature is set too low, then the curvature extrema are not reached at control points. To avoid the latter problem and for consistency with the alternate sign case, we opt to increase the curvature if it does not reach its maximum at control points, even though the curvature can shift sign between two control points.

The increase of curvature is computed as follows. First, we initialize the target curvatures \hat{K}_i with the estimated curvature K_i . For pairs of consecutive points p_i, p_{i+1} , we solve a G^2 Hermite problem with a 3-arc clothoid with constraints (p_i, θ_i, K_i) and $(p_{i+1}, \theta_{i+1}, K_{i+1})$ as described in [BF18b]. If the resulting curve does not attain curvature extrema at control points, we jointly increase the curvatures K_i and K_{i+1} . The choice of the increase function is discussed in Sec. 4.2.3. Then, \hat{K}_i is set to be the largest (in terms of absolute value) of the curvature increases for both problems p_{i-1}, p_i and p_i, p_{i+1} . Since \hat{K}_i only depends on K_{i-1}, K_i and K_{i+1} , curvature refinement somewhat increases the local support but maintains bounded locality. We analyze the local support in detail in Sec. 5. The algorithm is summarized in Algorithm 2.

Algorithm 2: Curvature refinement method

Data: (p_i, θ_i, K_i)
Result: Curvatures \hat{K}_i such that the transition p_{i-1} to p_i and from p_i to p_{i+1} is valid

$$\hat{K} \leftarrow K$$

$$\text{for each } (p_i, \theta_i, K_i), (p_{i+1}, \theta_{i+1}, K_{i+1}) \text{ do}$$

$$| \quad (\kappa_i, \kappa_{i+1}) \leftarrow (K_i, K_{i+1})$$

$$| \quad \text{repeat}$$

$$| | \quad \text{increase}(\kappa_i, \kappa_{i+1})$$

$$| \quad \text{until } \text{IsValidTransition}((p_i, \theta_i, \kappa_i), (p_{i+1}, \theta_{i+1}, \kappa_{i+1}))$$

$$| \quad \text{FindSmallestValidCurvatures}(\kappa_i, \kappa_{i+1})$$

$$| | \quad |\hat{K}_i| \leftarrow \max(|\hat{K}_i|, |\kappa_i|)$$

$$| | \quad |\hat{K}_{i+1}| \leftarrow \max(|\hat{K}_{i+1}|, |\kappa_{i+1}|)$$

$$\text{end}$$

4.2.2. Clothoid-line-clothoid transition

Given two boundary conditions (p_i, θ_i, K_i) and $(p_{i+1}, \theta_{i+1}, K_{i+1})$, a G^2 CLC is not guaranteed to exist. If such a transition does exist, its curvature varies linearly from K_i to 0 (first clothoid), continues as constant 0 (straight line), and finally goes linearly from 0 to K_{i+1} (second clothoid). With such a curvature profile, a CLC transition attains curvature extrema at control points and curvature monotonicity between control points of opposite curvature sign. If the straight line portion has vanishing length, the result is a 2-arcs clothoid.

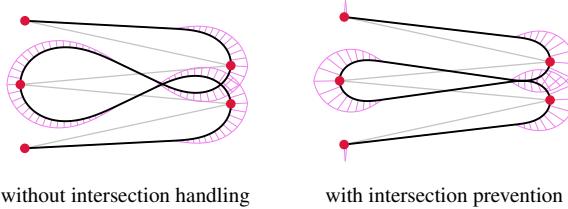


Figure 8: CLC transition with intersection, and with further refinement of the curvature to prevent intersections that do not occur within the control polygon (shown in grey).

The angle Φ entirely determines the parameters of both clothoids of the transition curve (Sec. 3.2). Computing the CLC transition amounts to finding Φ such that the vector $v(\Phi) = p_{e,i+1}(\Phi + \pi) - p_{e,i}(\Phi)$ makes an angle Φ with the x -axis. We use a binary search method for this purpose and find the value of $\Phi \in [\theta_i, \theta_i + \pi]$ that gives the best alignment of $v(\Phi)$ with $T(\Phi) = (\cos(\Phi), \sin(\Phi))^\top$. Alignment is measured as $\|v(\Phi) \times T(\Phi)\|$. We reject values of Φ for which $v(\Phi) \cdot T(\Phi)$ is negative, as it means that the points are aligned in the wrong order.

The increase of curvature is analogous to Sec. 4.2.1. The target curvatures \hat{K}_i are initialized with the estimated curvatures K_i . Then, for both conditions $(p_i, \theta_i, K_i), (p_{i+1}, \theta_{i+1}, K_{i+1})$, we find Φ that achieves the best alignment. If the alignment error is above a certain threshold or if $v(\Phi) \cdot T(\Phi) < 0$, we jointly increase K_i and K_{i+1} and set \hat{K}_i to be the maximum (in terms of absolute value) found for both problems. The locality is affected in the same way as for the 3-arcs curves.

Intersections. Self-intersections are often seen as undesirable yet unavoidable in the general case. We claim that our method can produce curves with intersections consistent with those of the control polygon, i.e., the piece of curve C_i does not intersect with C_j if the straight segment (p_i, p_{i+1}) does not intersect with (p_j, p_{j+1}) . According to Proposition 3.2, we can approximate the control polygon by taking arbitrarily high curvature. Therefore, we can use the same principle as the previous sections to increase the curvature if necessary. First, we check whether the control polygon self-intersects. If not, we apply Algorithm 2 where the function *InvalidTransition* returns *true* if the two curves intersect. These two curves are composed of straight lines and clothoids, for which the intersection detection can be performed via curve segmentation [BBF20].

Intersection detection and locality are opposite targets if we apply intersection detection on the whole curve. It is however possible to restrict the intersection detection and the refinement to adjacent curves to keep the local support property. We show in Fig. 8 the result of our curvature refinement via intersection detection.

4.2.3. Increase function for the curvature

We consider two strategies when increasing the curvature at control points: linear and a max-linear increase. Both rely on an increase parameter $\tau > 1$. The linear increase function scales both curvatures K_i and K_{i+1} by the parameter τ . This leads to an undifferentiated treatment of the two curvatures. In contrast, the max-linear increase

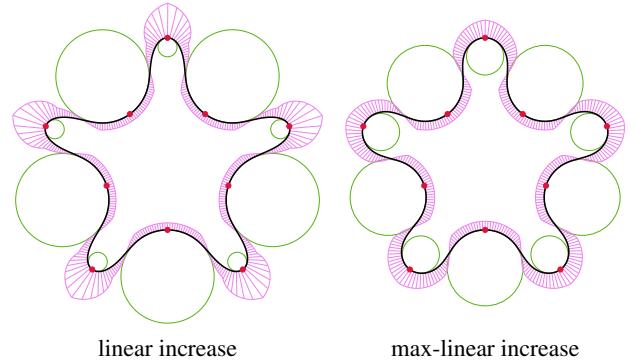


Figure 9: The linear and the max-linear increase functions give different results. We show their effect in this example using 3-arcs clothoid transition, where the control point curvatures are initialized by circle-ellipse fitting. The osculating circles are shown in green, and the scaled curvature normals in purple.

prioritizes increasing the smaller curvature. Assume $|K_i| < |K_{i+1}|$, and let $\tau_l = |K_{i+1}|/|K_i|$. The max-linear increase is defined as

$$\text{Inc}_{ML}(K_i, K_{i+1}) = \begin{cases} (\tau K_i, K_{i+1}) & \text{if } \tau < \tau_l \\ \frac{\tau}{\tau_l} (\text{sign}(K_i)|K_{i+1}|, K_{i+1}) & \text{otherwise} \end{cases}$$

Since we want the increase to be only as high as necessary, the modification of τ has to undergo two phases. First, τ is initialized to 1, which is equivalent to no increase of curvature. If the resulting curve is not valid, in the sense defined for each respective method in Sections 4.2.1 and 4.2.2, we add to τ a positive step ϵ_t . We double the step $\epsilon_{t+1} = 2\epsilon_t$ until we reach a coefficient τ_{\max} high enough for a valid transition. Once (K_i, K_{i+1}) is high enough for a valid transition, we know that τ must be in the range $[\tau_{\max} - \epsilon_t, \tau_{\max}]$. We can then operate a binary search on this interval.

The different results given by the increase functions are showcased in Fig. 9. While linear increase maintains proportionality between curvature profiles, the max-linear strategy produces more homogeneous curvatures among the control points.

4.3. Interactive input of tangents and curvatures

Our method is based on an estimation of tangents and curvatures at the control points. We can offer further modeling options by allowing the user to input a constrained tangent and curvature at certain control points. Constrained curvatures are not modified during the refinement and therefore improve the locality of the method. However, the user choices, in extreme cases, could prevent us from fulfilling the desired set of mathematical properties of the curve. We show examples in Fig. 10.

5. Results

We implement our technique in C++ using libigl [JP*18] for interacting with the curve and libhedral [V*17] for visualization. Clothoid computations for solving the G^1 problem [BF15], the 3-arcs problem [BF18b], and intersections detection [BBF20] employ a dedi-

Property	Trigonometric blending [Yuk20]	κ -curves [YSW*17]	$\varepsilon\kappa$ -curves [MGS*21]	Clothoid spline [BF18a]	3-arcs clothoid transition	CLC transition
Smoothness	G^2	mostly G^2	bounded G^1 gaps	G^2	G^2	G^2
Continuity w.r.t. p_i	yes	yes	yes	no	no	curvature estimation
Local support	4 points	no	no	no	6 or 8 points	6 or 8 points
Global optimization	no	yes	yes	yes	no	no
Self-intersections			no intersections within a curve segment			consistent w/control polygon
Monotone curvature	no	yes	yes	yes	yes	yes
Curvature extrema at p_i	no	yes	yes	yes	alternate curvature	yes
Collinearity	yes	hard	hard	hard	yes	hard
Roundness	locally	no, but [YSW19] can globally		globally	locally	no
User control	no	no	curvature profile	9 objectives	directly on tangents and curvatures	

Table 1: Comparison of curvature-continuous splines formulations; the criteria are taken from [Yuk20]. Our method is listed in the last two columns, distinguishing between the two options for transition curves: 3-arcs clothoids and the CLC transition. Collinearity refers to reproducing a straight line when three consecutive control points are collinear. Roundness refers to reproducing a circle or part of it.

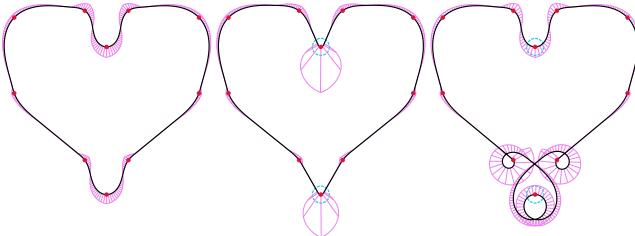


Figure 10: Allowing the user to input curvature and tangent constraints offers additional modeling possibilities. We show the reference shape (left), the shape with user-modified higher curvature at top and bottom (middle), and a result where input curvature at the bottom tip is set inconsistently with the sign of the control polygon (right). The results are produced with CLC transition, G^1 curvature estimate and max-linear increase function.

cated library [BF21]. Our method has a few variations, and the user can tweak the result as follows:

1. Choose either 3-arcs clothoids or CLC-transitions for the transition curves between consecutive control points.
2. Initial curvature estimation via circle-ellipse local interpolation [Yuk20] or G^1 -clothoid fitting.
3. Linear or max-linear increase function for curvature refinement.

We present our results with all the possible combinations and compare with other works in Fig. 11. In the following, we discuss and compare our method with the state of the art; Table 1 summarizes the comparison using the criteria employed in [Yuk20].

5.1. Discussion, comparisons and limitations

Smoothness. Our method produces curves with G^2 smoothness, an important property for fairness. The method does not require parameter values as input and produces arc length parameterized

curves, hence no reparameterization is required to ensure G^2 . The trigonometric blending functions in [Yuk20] and the clothoid splines [BF18a] achieve G^2 continuity, while κ -curves [YSW*17] may show G^1 discontinuity at inflection points. The $\varepsilon\kappa$ -curves [MGS*21] have an additional degree of freedom that can be tuned to reduce the discontinuity at inflection points. Our method produces G^2 curves, but we do not control the curvature slope, which can in certain cases give an impression of curvature discontinuity in practice (see Fig. 11). The user can manually increase the curvature at a neighboring control point to prevent it, but we do not propose an automatic correction method.

Local support. Locality characterizes what part of the curve is influenced by moving a control point. With trigonometric blending [Yuk20], each curve is obtained through the blending of two arcs interpolating between three consecutive points. Therefore, each curve segment between two control points has a support of 4 consecutive points. Catmull-Rom splines [CR74] provide local support with 6 points. All previous techniques that enforce curvature extrema at control points rely on global optimization and do not have local support: κ -curves [YSW*17] and $\varepsilon\kappa$ -curves [MGS*21] optimize the whole shape to place Bézier control points, and Bertolazzi and Frego [BF18a] propose clothoid splines with curvature continuity, which only leaves two degrees of freedom, exploited to optimize a selection of optimization objectives.

The locality of our curves depends on the curvature estimation method. The curve segment C_i interpolating between p_i and p_{i+1} is determined by the values of (θ_i, \hat{K}_i) and $(\theta_{i+1}, \hat{K}_{i+1})$. The curvature \hat{K}_i depends on K_{i-1} . If we estimate curvature solely via the hybrid ellipse-circle method [Yuk20], then K_{i-1} depends at most on p_{i-2} . If we use the more advanced G^1 Hermite fitting [BF15] to perform the estimation, then K_{i-1} depends on t_{i-2} which depends on p_{i-3} . Likewise, \hat{K}_{i+1} either depends on p_{i+3} or also on p_{i+4} . Therefore our local support relies on either 6 or 8 consecutive control points, depending on the choice of the curvature estimation method. Our optional curvature refinement based on intersection detection (see

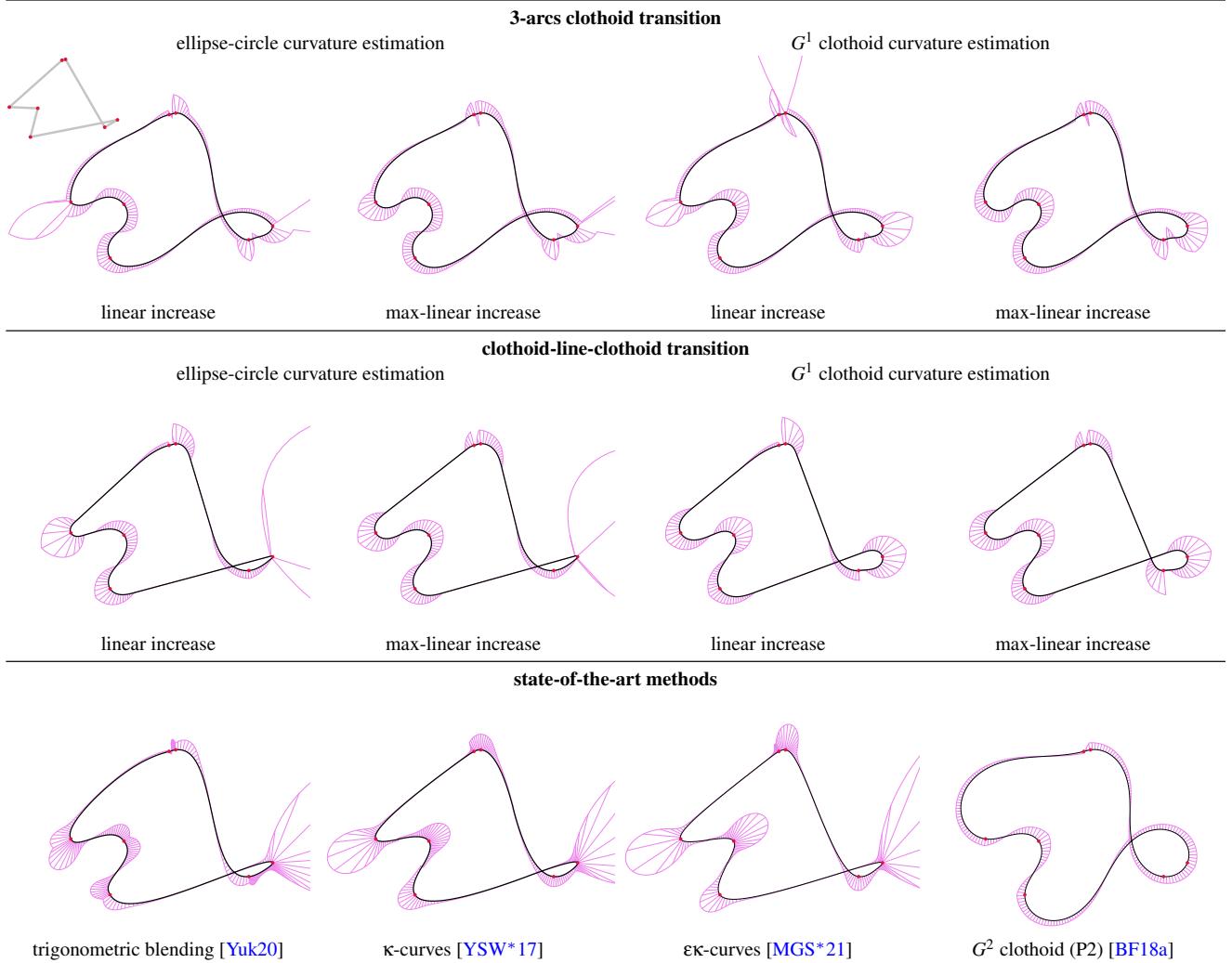


Figure 11: We show all the variants of our method and compare it with the state of the art. All curves are generated from the same set of control points, taken from [Yuk20]. The control polygon is shown in the inset (top left). The top row demonstrates our formulation for the 3-arcs clothoid transition, and the middle row shows CLC transition, each with the different options in curvature estimation and increase functions for curvature refinement. The $\varepsilon\kappa$ -curves [MGS*21] are produced with the cubic method, $\alpha = 0.85$. The purple lines represent the scaled curvature normals.

Sec. 4.2.2) is performed up to a certain number of consecutive curve segments and can therefore impact locality. The supplemental video demonstrates the effects of moving control points and the local support in practice.

Curvature profile. Curvature is a core object of our work. We separate two cases: the monotonicity of curvature between two control points whose curvature signs are opposite, and the local extremum of curvature at control points. Blending methods [Yuk20] cannot achieve these goals. Their local interpolations pass through the control points but with different tangents and curvatures. Blending these interpolations results in a necessary curvature increase near the control points, which can be observed as two bumps in the curvature comb (see e.g. Fig. 1). In contrast, κ -curves [YSW*17] and $\varepsilon\kappa$ -

curves [MGS*21] achieve the curvature goals precisely, at the cost of finite support. By elevating the degree of quadratic Bézier segments to cubic, $\varepsilon\kappa$ -curves propose an additional parameter to tweak the curvature profile and reduce the discontinuity by increasing the curvature at control points. G^2 clothoid splines [BF18b] achieve the best curvature profile, since monotonicity is ensured even when some consecutive control points have curvature of the same sign.

Both our methods provide monotonicity of curvature between two control points of opposite curvature sign, see e.g. Fig. 1. However, the 3-arcs clothoid method does not ensure that the maximum of curvature is reached at control points because the curvature sign can flip between two control points of the same sign. This problem does not show up with the CLC transition because its curvature profile always follows a specific pattern: linear ramp from \hat{K}_i to 0,

constant 0, and another ramp from 0 to \hat{K}_{i+1} . Even though our method can work solely based on the automatic estimation of the geometric features at control points, we also offer to manually set the curvature and tangents at certain control points, which variegates the possibilities in curvature profiles (see the accompanying videos). Other methods with parameters [MGS^{*}21, BF18b] can also vary their results, but such a variation must apply to the whole shape.

Intersections and distance to control polygon. All methods shown in Fig. 11 avoid self-intersections within the individual curve segments between two successive control points. However, our CLC transition method can use curvature refinement so that the resulting curve is as close as possible to the control polygon (Proposition 3.2). As shown in Sec. 4.2.2, we can therefore produce curves that do not self-intersect at all if the self-intersection is not present in the control polygon, which is a stronger property.

Another point developed in [Yuk20] concerns the distance to the control polygon. Trigonometric blending based on circular interpolation [SNV00, SLY05] produces curves than can be arbitrarily far from the control polygon, while Bézier or elliptic interpolations [SZ09, Yuk20] form curves with bounded distance. We do not provide such bounds for our methods. However, we note that for the CLC method, increasing the curvature at both control points shortens the distance from the control polygon segment to the CLC curve. It is therefore possible to bound the distance from the control polygon as much as desired. Contrary to intersection detection, such a curvature refinement does not affect our local support size.

Lines and circles. Having curvature extrema at control points appears to be in contradiction with collinearity and roundness properties. Locally, these properties imply that if several consecutive control points are on a line or a circle, the curve reproduces this line or circle segment. Blending methods excel at this property when blending the appropriate interpolants [Yuk20, SNV00], but the constant curvature along a line or a circle conflicts with attaining a curvature extrema at control points. For κ -curves [YSW^{*}17], $\epsilon\kappa$ -curves [MGS^{*}21] and CLC transitions, even global roundness fails: when all the control points are set on a circle, the resulting curve is not a circle (Fig. 13). Yan et al. [YSS19] fix this issue on κ -curves by using a rational quadratic Bézier interpolation base. Clothoid splines [BF18b] provide global roundness, and our 3-arcs clothoid method has local roundness, since 3-arcs clothoids do not enforce curvature maximum at control points. This method requires 6 consecutive points set on a circular to produce a circle arc, while [Yuk20] only requires 4. Our CLC method fails if three or more consecutive control points are collinear, since the curvature refinement cannot operate on vanishing curvature. In this case, we can either break G^2 continuity or use a 3-arcs clothoid transition, losing the placement of curvature extrema at control points. We show in Fig. 12 how our method compares to $\epsilon\kappa$ -curves [MGS^{*}21] and trigonometric blending [Yuk20]. All methods fail to simultaneously provide maximum curvature at control points and linearity, because these two objectives are contradictory.

Invalid CLC transitions. We prove in Proposition 3.1 that if two clothoid shells have curvatures greater than a certain threshold, then there exists a CLC transition. This is a weaker statement than the

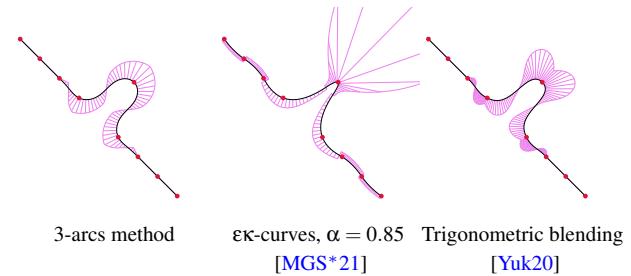


Figure 12: Linearity and curvature extrema at control points are conflicting objectives. The 3-arcs clothoid transitions have local collinearity property, but the curvature extrema do not necessarily occur at control points (left). The $\epsilon\kappa$ -curves [MGS^{*}21] attain curvature extrema at control points, but the curve shape is wavy and does not reproduce a straight line (middle). Trigonometric blending [Yuk20] ensures linearity but fails to provide curvature extrema at control points (right).

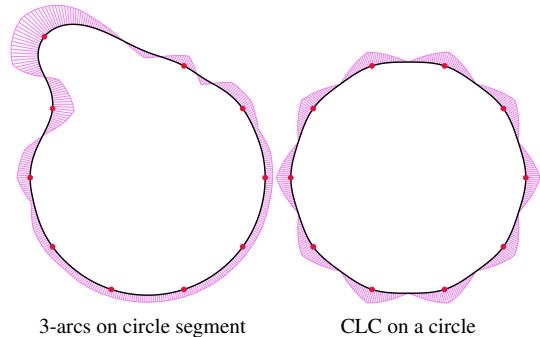


Figure 13: Circles can be challenging for our method. The 3-arcs clothoid transitions have local collinearity and roundness properties, but the curvature extrema do not necessarily occur at control points (left). CLC transitions do not reproduce circles (right).

affirmation that if a curvature pair (K_0, K_1) allows a valid CLC transition, then any pair (K'_0, K'_1) such that $K'_i \geq K_i$, $i = 0, 1$, also allows it. We believe it to be true for points of alternating curvature sign (see appendix). Unfortunately, in rare cases, this does not hold for control points with curvature of same sign, as show in Fig. 14. It is possible to fix the problem in several ways: the user can always try to manually set the curvature; we can apply a second round of curvature refinement at the cost of locality; finally, we can use the 3-arcs clothoid transition at the cost of losing curvature extrema at control points. In practice, the latter solution seems to keep the curvature extremum property because the failure of the CLC transition means that the curvature conditions are too low, which is opposite to the change of sign of the curvature in the 3-arcs transition, as explained in Sec. 4.2.1.

Number of control points. Our method places curvature extrema at control points, implying that control points are positioned at salient features of the shape and should generally be sparse. Increasing the number of control points produces either wavier curves (3-arcs method) or high localized curvature and straight lines (CLC method).

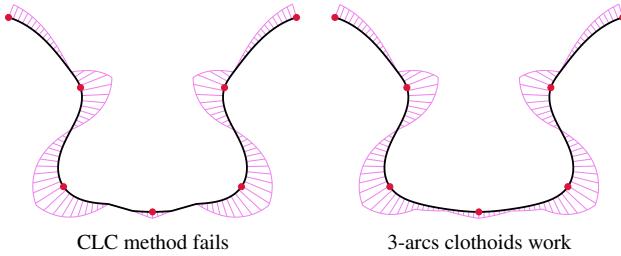


Figure 14: Curvature refinement for consecutive points of the same curvature sign can lead to failure of G^2 CLC transition (left). Detecting such cases and using 3-arcs clothoid transition solves the problem (right).

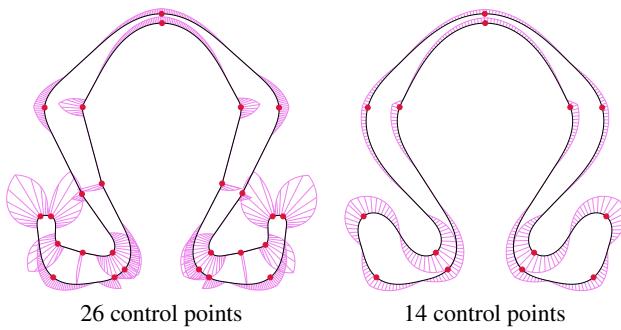


Figure 15: Clothoid-line-clothoid transition on the Ω shape using varying number of control points (taken from [BF21]).

Our method works best with points whose curvature is of alternate sign, in contrast to κ -curves, which have discontinuities at inflection points. As shown in Fig. 15, using more control points does not necessarily produce a fairer shape.

Continuity with respect to control points. Continuity w.r.t. the control point locations is desirable, as it makes the modeling results more predictable. For methods with local support, any discontinuity w.r.t. moving a point p_i is restricted to the neighborhood of p_i . Trigonometric blending [Yuk20] and κ -curves [YSW*17, MGS*21] have general continuity w.r.t. control points, while clothoid splines [BBF20] and our 3-arcs clothoid methods do not, as our experiments reveal. The case of CLC transitions is dependent on the curvature estimation technique. With G^1 clothoid estimation, there is no general continuity for two reasons: we use hybrid interpolation if the curvature estimation via G^1 fitting is not consistent with the sign of the control polygon, leading to an abrupt change in some cases; additionally, the curvature estimation abruptly changes when the sign of the angles of the control polygon changes, which happens when a control point crosses a segment of the control polygon. However, CLC transitions with circle-ellipse based curvature estimation are continuous w.r.t. control points, except when they are collinear. We stress that discontinuity w.r.t. control points is mitigated by the local support. Moreover, any modeling method that involves nonlinear optimization does not guarantee such continuity, since there can be multiple optima, but the collection of desirable properties generally leads to intuitive user experience overall.

5.2. Computation

According to our experiments on a 2.6 GHz 6-Core Intel Core i7, our method takes 2.3 seconds on 1000 control points and 100 evaluations per control. Thanks to the local support, the runtime is linear in the number of control points. Changing a control point modifies up to 8 piecewise clothoid transitions, hence there is no need for complete recomputation, and modeling by moving one control point can be performed in constant time. Since clothoids are based on Fresnel integrals, which are transcendental functions, they are expensive to compute. Contrary to trigonometric functions, they are not periodic and therefore prone to numerical errors for evaluation of large quantities. However, Fresnel integrals are popular and their evaluation is well studied [SRC03, VMCU16]. The accompanying videos show the realtime performance of our method, and our interactive demo code is attached to the submission as well.

6. Conclusion

We presented a method for planar curve interpolation that unites several properties that previously could not be achieved within one framework: G^2 smoothness, strong locality, curvature sign consistent with the control polygon, curvature extrema at control points and curvature monotonicity for alternating curvature sign, and self-intersections entirely consistent with the control polygon. Previously, guarantees on curvature monotonicity and extrema could only be achieved using global optimization in a G^2 framework. The local support offered by our method leads to more efficient computation and intuitive control. Our key insight is that the curvature estimations at control points can be minimally increased in a coordinated, but localized manner to achieve the desired curve properties.

Several improvements on performance and quality could be attempted in future work. We would like to find a formula for the clothoid shell intersection to obtain a better understanding of self-intersection scenarios and more efficient computation. It would be interesting to explore additional curvature refinement approaches and derive lower bounds on the feasible curvatures at control points. We did not modify the tangent estimations in this work, but tangents do have a significant influence on the final curve shape, and their optimization could be explored.

Perhaps surprisingly, there are still many unanswered questions to explore on the topic of smooth curve interpolation despite the maturity of this research field. The quantification of *aesthetics* remains elusive, although subjectively clothoids seem appealing to the eye. Clothoids have linear curvature profiles, and it would be interesting to explore other profiles, such as polynomial curvature. Another avenue for future work is the generalization to curves in 3D and tensor product surfaces.

Acknowledgements

We thank the reviewers for their remarks. We also thank Ilya Baran for the insightful discussions and inspiration. This work was supported in part by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 101003104, ERC CoG MYCLOTH).

References

- [ASR66] ABRAMOWITZ M., STEGUN I. A., ROMAIN J. E.: Handbook of mathematical functions, with formulas, graphs, and mathematical tables. *Physics Today* 19, 1 (1966), 120–121. doi:[10.1063/1.3047921](https://doi.org/10.1063/1.3047921). 3
- [Att74] ATTNEAVE F.: Some informational aspects of visual perception. *Psychological Review* 61, 3 (1974), 183–193. doi:[10.1037/h0054663](https://doi.org/10.1037/h0054663). 2
- [BBF20] BERTOLAZZI E., BEVILACQUA P., FREGO M.: Efficient intersection between splines of clothoids. *Mathematics and Computers in Simulation* 176 (2020), 57–72. Applied Scientific Computing XV: Innovative Modelling and Simulation in Sciences. doi:<https://doi.org/10.1016/j.matcom.2019.10.001>. 3, 7, 11
- [BF15] BERTOLAZZI E., FREGO M.: G^1 fitting with clothoids. *Mathematical Methods in the Applied Sciences* 38, 5 (2015), 881–897. 3, 5, 7, 8
- [BF18a] BERTOLAZZI E., FREGO M.: Interpolating clothoid splines with curvature continuity. *Mathematical Methods in the Applied Sciences* 41, 4 (Mar. 2018), 1723–1737. doi:[10.1002/mma.4700](https://doi.org/10.1002/mma.4700). 3, 8, 9
- [BF18b] BERTOLAZZI E., FREGO M.: On the G^2 Hermite interpolation problem with clothoids. *Journal of Computational and Applied Mathematics* 341 (2018), 99–116. doi:[10.1016/j.cam.2018.03.029](https://doi.org/10.1016/j.cam.2018.03.029). 3, 5, 6, 7, 9, 10
- [BF21] BERTOLAZZI E., FREGO M.: G^1 and G^2 fitting with clothoids, spline of clothoids, circle arc and biarc. <https://github.com/Clothoids/>, 2021. Accessed: 2022-03-28. 8, 11
- [BLP10] BARAN I., LEHTINEN J., POPOVIĆ J.: Sketching clothoid splines using shortest paths. *Computer Graphics Forum* 29, 2 (2010), 655–664. 3
- [CDH*07] CHEUTET V., DANIEL M., HAHMANN S., LA GRECA R., LEON J.-C., MACULET R., MENEGAUX D., SAUVAGE B.: Constraint modeling for curves and surfaces in cagd: a survey. *International Journal of Shape Modeling* 13, 02 (2007), 159–199. doi:[10.1142/S0218654307000993](https://doi.org/10.1142/S0218654307000993). 2
- [CR74] CATMULL E., ROM R.: A class of local interpolating splines. In *Computer aided geometric design*. Elsevier, 1974, pp. 317–326. 2, 8
- [Far06] FARIN G.: Class A Bézier curves. *Computer Aided Geometric Design* 23, 7 (2006), 573–581. doi:[10.1016/j.cagd.2006.03.004](https://doi.org/10.1016/j.cagd.2006.03.004). 2
- [FF02] FARIN G. E., FARIN G.: *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002. 1, 2
- [Hea85] HEALD M. A.: Rational approximations for the fresnel integrals. *Mathematics of Computation* 44, 170 (1985), 459–461. 3
- [HEWF13] HAVEMANN S., EDELSBRUNNER J., WAGNER P., FELLNER W.-D.: Curvature-controlled curve editing using piecewise clothoid curves. *Computers & Graphics* 37, 6 (2013), 764–773. 2
- [HL93] HOSCHEK J., LASSER D.: *Fundamentals of computer aided geometric design*. AK Peters, Ltd, 1993. 2
- [JP*18] JACOBSON A., PANIZZO D., ET AL.: libigl: A simple C++ geometry processing library. <https://libigl.github.io/>, 2018. 7
- [KFP03] KIMIA B., FRANKEL I., POPESCU A.-M.: Euler spiral for shape completion. *International Journal of Computer Vision* 54 (08 2003), 159–182. doi:[10.1023/A:1023713602895](https://doi.org/10.1023/A:1023713602895). 3
- [Lev09] LEVIEN R. L.: *From Spiral to Spline: Optimal Techniques in Interactive Curve Design*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2009. URL: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-162.html>. 2, 3
- [LS09] LEVIEN R., SÉQUIN C. H.: Interpolating splines: Which is the fairest of them all? *Computer-Aided Design and Applications* 6, 1 (2009), 91–102. 2
- [MGS*21] MIURA K. T., GOBITHAASAN R. U., SALVI P., WANG D., SEKINE T., USUKI S., INOGUCHI J.-I., KAJIWARA K.: $\epsilon\kappa$ -curves: controlled local curvature extrema. *The Visual Computer* (2021). doi:[10.1007/s00371-021-02149-8](https://doi.org/10.1007/s00371-021-02149-8). 2, 8, 9, 10, 11
- [MJF15] MARZBANI H., JAZAR R. N., FARD M.: Better road design using clothoids. In *Sustainable Automotive Technologies 2014* (Cham, 2015), Denbratt I., Subic A., Wellnitz J., (Eds.), Springer International Publishing, pp. 25–40. 3
- [MS09] MCCRAE J., SINGH K.: Sketching piecewise clothoid curves. *Computers & Graphics* 33, 4 (2009), 452–461. 3
- [MW92] MEEK D. S., WALTON D. J.: Clothoid spline transition spirals. *Mathematics of Computation* 59 (1992), 117–133. 3
- [Pob13] POBEGAILO A.: Interpolating rational Bézier spline curves with local shape control. *International Journal of Computer Graphics & Animation* 3 (10 2013), 1–14. doi:[10.5121/ijcga.2013.3401](https://doi.org/10.5121/ijcga.2013.3401). 2
- [SLY05] SÉQUIN C. H., LEE K., YEN J.: Fair, G^2 -and C^2 -continuous circle splines for the interpolation of sparse data points. *Computer-Aided Design* 37, 2 (2005), 201–211. 2, 10
- [SNV00] SZILVÁSI-NAGY M., VENDEL T. P.: Generating curves and swept surfaces by blended circles. *Computer Aided Geometric Design* 17, 2 (2000), 197–206. 2, 10
- [SRC03] SÁNCHEZ-REYES J., CHACÓN J. M.: Polynomial approximation to clothoids via s-power series. *Comput. Aided Des.* 35 (2003), 1305–1313. 3, 11
- [Sto82] STOER J.: Curve fitting with clothoidal splines. *Journal of research of the national Bureau of standards* 87, 4 (1982), 317. 3
- [SZ09] SUN C., ZHAO H.: Generating fair, C^2 continuous splines by blending conics. *Computers & Graphics* 33, 2 (2009), 173–180. 2, 10
- [V*17] VAXMAN A., ET AL.: libheda: geometric processing and optimization of polygonal meshes. <https://github.com/avaxman/libheda>, 2017. 7
- [VMCU16] VÁZQUEZ-MÉNDEZ M., CASAL URCERA G.: The clothoid computation: A simple and efficient numerical algorithm. *Journal of Surveying Engineering* 142 (08 2016), 04016005:1–9. doi:[10.1061/\(ASCE\)SU.1943-5428.0000177](https://doi.org/10.1061/(ASCE)SU.1943-5428.0000177). 3, 11
- [Wen96] WENZ H.-J.: Interpolation of curve data by blended generalized circles. *Computer Aided Geometric Design* 13, 8 (1996), 673–680. doi:[10.1016/0167-8396\(95\)00054-2](https://doi.org/10.1016/0167-8396(95)00054-2)
- [WGS*20] WANG D., GOBITHAASAN R., SEKINE T., USUKI S., MIURA K.: Interpolation of point sequences with extremum of curvature by log-aesthetic curves with G^2 continuity. *Computer-Aided Design and Applications* 18 (07 2020), 399–410. doi:[10.14733/cadaps.2021.399-410.2](https://doi.org/10.14733/cadaps.2021.399-410.2)
- [WM09] WALTON D., MEEK D.: G^1 interpolation with a single Cornu spiral segment. *Journal of Computational and Applied Mathematics* 223, 1 (2009), 86–96. doi:<https://doi.org/10.1016/j.cam.2007.12.022>. 3
- [YSK09] YUKSEL C., SCHAEFER S., KEYSER J.: On the parameterization of Catmull-Rom curves. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling* (New York, NY, USA, 2009), ACM, pp. 47–53. doi:[10.1145/1629255.1629262](https://doi.org/10.1145/1629255.1629262). 2
- [YSS19] YAN Z., SCHILLER S., SCHAEFER S.: Circle reproduction with interpolatory curves at local maximal curvature points. *Comput. Aided Geom. Des.* 72 (2019), 98–110. 2, 8, 10
- [YSW*17] YAN Z., SCHILLER S., WILENSKY G., CARR N., SCHAEFER S.: K-curves: Interpolation at local maximum curvature. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–7. 1, 2, 8, 9, 10, 11
- [Yuk20] YUKSEL C.: A class of C^2 interpolating splines. *ACM Transactions on Graphics* 39, 5 (jul 2020), 160:1–160:14. doi:[10.1145/3400301](https://doi.org/10.1145/3400301). 1, 2, 5, 6, 8, 9, 10, 11, 14
- [Zia16] ZIATDINOV R.: Visual perception, quantity of information function and the concept of the quantity of information continuous splines. *Scientific Visualization* 8 (03 2016), 168–178. 2

Appendix A: Proof of Proposition 3.1

The equation of the normal clothoid shell function is given by

$$s_n : \Phi \in [0, 2\pi] \rightarrow 2\Phi \left(\int_0^1 \cos(\Phi(-u^2 + 2u)) du \right) \quad (5)$$

We defined the clothoid shell of the point p_i with tangent angle condition θ_i and curvature κ_i through the normal shell clothoid s_n via the formula

$$s_i(\Phi) = p_i + \frac{1}{\kappa_i} R(\theta_i) s_n(\Phi - \theta_i).$$

This formula is valid for $\Phi \in [\theta_i, 2\pi + \theta_i]$, but the function can be extended to $\Phi \in \mathbb{R}$ by taking the angle corresponding to the angle in $[\theta_i, 2\pi + \theta_i]$. Therefore s_i is discontinuous on \mathbb{R} at $\theta_i + 2k\pi$ with $k \in \mathbb{Z}$.

Proposition 3.1. Let p_0, p_1 be two distinct points and θ_0 and θ_1 two angles. There exist curvature bounds K_0, K_1 , such that for any $\kappa_0 \geq K_0, \kappa_1 \geq K_1$, there exists an angle Φ such that the line segment $[s_0(\Phi), s_1(\Phi + \pi)]$ makes an angle Φ with the x -axis, where the starting conditions for the two clothoids are $(p_0, \theta_0, \kappa_0)$ and $(p_1, \theta_1, \kappa_1)$, respectively.

Proof. Let s_0 and s_1 be the clothoid shells related to points $(p_0, \theta_0, \kappa_0)$ and $(p_1, \theta_1, \kappa_1)$, respectively. Let Σ_0 and Σ_1 be the traces of curves s_0 and s_1 and $L_{\Sigma_0, \Sigma_1} = \bigcup_{p \in \Sigma_0, q \in \Sigma_1} [p, q]$ be the set of segments whose two end points are in Σ_0 and Σ_1 , respectively. Let $[\Phi_a, \Phi_b]$ be the set of angles between the x -axis and the lines $[p, q] \in L_{\Sigma_0, \Sigma_1}$. This set is simply connected and can be noted as an interval because Σ_0 and Σ_1 are simply connected.

We know that for all Φ in $[0, 2\pi]$, $\|s_i(\Phi) - p_i\| \leq \frac{4\pi}{\kappa_i}$. Therefore, when both curvatures κ_0 and κ_1 converge towards $+\infty$, s_0 and s_1 converge uniformly towards p_0 and p_1 and the set $[\Phi_a, \Phi_b]$ is actually converging to the singleton $\{\Phi_{0,1}\}$, which is the angle formed by the x -axis and the segment $[p_0, p_1]$. We define $v(\Phi) := s_1(\Phi - \pi) - s_0(\Phi)$. Proving Proposition 3.1 is equivalent to proving that there exists Φ such that $v(\Phi)$ makes an angle Φ with the x -axis. We distinguish between several cases:

- If $\Phi_{0,1} = \theta_0 = \theta_1 - \pi$, then $v(\theta_0) = s_1(\theta_1) - s_0(\theta_0) = p_1 - p_0$ makes an angle θ_0 with the x -axis.
- Otherwise, we recall that s_i is continuous except on the set of points $\{\theta_i + 2k\pi \mid k \in \mathbb{Z}\}$. We have again two cases to consider:
 - If $\Phi_{0,1}$ is not equal to $\theta_1 - \pi$ or θ_0 , then there exists an $\epsilon > 0$ such that there are K_0 and K_1 , such that for $\kappa_0 > K_0, \kappa_1 > K_1$, $[\Phi_a, \Phi_b] \subset [\Phi_{0,1} - \epsilon, \Phi_{0,1} + \epsilon]$ and both $\theta_1 - \pi$ and θ_0 are not in $[\Phi_a, \Phi_b]$. Therefore, both s_0 and $s_1 \circ \tau_{-\pi}$ are continuous on such an interval $[\Phi_a, \Phi_b]$ ($\tau_{-\pi}$ is a shift by $-\pi$). By continuity, there is $\Phi \in [\Phi_a, \Phi_b]$ such that $v(\Phi)$ makes an angle Φ with the x -axis.
 - We suppose that $\Phi_{0,1} = \theta_0$ but $\Phi_{0,1} \neq \theta_1 - \pi$ (by symmetry of the problem, the case in which $\Phi_{0,1} = \theta_1 - \pi$ and $\Phi_{0,1} \neq \theta_0$ is similar). Without loss of generality, since the situation is the same up to a rotation and translation, we can consider $\theta_0 = 0 = \Phi_{0,1}$ and $p_0 = (0, 0)$. In this setting, p_1 has necessarily a positive x -coordinate and vanishing y -coordinate.

We consider $\epsilon > 0$ such that $\epsilon < \theta_1 \pm \pi$, where $\theta_1 \pm \pi$ is the value of the angle $\theta_1 + \pi$ in $[0, 2\pi]$. Let K_0, K_1 be such that $\forall \kappa_0 > K_0, \kappa_1 > K_1, [\Phi_a, \Phi_b] \subset [-\epsilon, \epsilon]$. We cannot conclude as before because we do not have continuity of s_0 at 0. We must prove that the angle with the x -axis of $v(\Phi) = s_1(\Phi \pm \pi) - s_0(\Phi)$ is positive for $\Phi \in [0, \epsilon]$. To do so, we prove that $s_1(\pi)$ has a y -coordinate (noted $s_1(\pi)_y$) that is always positive. Indeed, we know that if $\theta_1 \in [0, \pi]$, then

$$s_1(\pi)_y = \frac{2(\pi - \theta_1)}{\kappa_1} \int_0^1 \sin((\pi - \theta_1)(-u^2 + 2u) + \theta_1) du > 0$$

and if $\theta_1 \in (\pi, 2\pi)$,

$$s_1(\pi)_y = \frac{2(3\pi - \theta_1)}{\kappa_1} \int_0^1 \sin((3\pi - \theta_1)(-u^2 + 2u) + \theta_1) du > 0$$

- If $\theta_1 \in [0, \pi]$: since $u \in [0, 1] \rightarrow -u^2 + 2u$ is an increasing function with values in $[0, 1]$, we know that $\forall \theta_1 \in [0, \pi], \forall u \in [0, 1], 0 \leq \theta_1 \leq (\pi - \theta_1)(-u^2 + 2u) + \theta_1 \leq \pi$. This means that, $\forall \theta_1 \in [0, \pi], \forall u \in [0, 1], \sin((\pi - \theta_1)(-u^2 + 2u) + \theta_1) \geq 0$. Since the sine function is not constantly equal to 0 on $[\theta_1, \pi]$, this shows that $\int_0^1 \sin((\pi - \theta_1)(-u^2 + 2u) + \theta_1) du > 0$.
- If $\theta_1 \in (\pi, 2\pi)$, let us take $\Theta_1 = \theta_1 - \pi \in (0, \pi)$. We want to prove that $\int_0^1 \sin((3\pi - \theta_1)(-u^2 + 2u) + \theta_1) du > 0$, i.e., $\int_0^1 \sin((2\pi - \Theta_1)(-u^2 + 2u) + \Theta_1 + \pi) du > 0$, i.e., that

$$\int_0^1 \sin((2\pi - \Theta_1)(-u^2 + 2u) + \Theta_1) du < 0 \quad (6)$$

We therefore show that:

$$\begin{aligned} & \int_0^1 \sin((2\pi - \Theta_1)(-u^2 + 2u) + \Theta_1) du \\ &= \int_0^1 \sin(-(2\pi - \Theta_1)(u^2 - 2u + 1) + 2\pi - \Theta_1 + \Theta_1) du \\ &= - \int_0^1 \sin((2\pi - \Theta_1)(1-u)^2) du \\ &= - \int_0^1 \sin((2\pi - \Theta_1)u^2) du \\ &= - \frac{\int_0^{\sqrt{2\pi - \Theta_1}} \sin(x^2) dx}{\sqrt{2\pi - \Theta_1}} \\ &= - \frac{S(\sqrt{2\pi - \Theta_1})}{\sqrt{2\pi - \Theta_1}}, \end{aligned}$$

where S is the Fresnel integral $S : t \rightarrow \int_0^t \sin(x^2) dx$. It is known that S is positive on $\mathbb{R}_{>0}^+$, therefore we have proved the statement in (6).

Hence the angle ψ between the x -axis and $v(0) = s_1(\pi) - s_0(0)$ is positive. This angle ψ is smaller than ϵ since $[\Phi_a, \Phi_b] \subset [-\epsilon, \epsilon]$ and $s_1 \circ \tau_{-\pi}$ is continuous on $[0, \epsilon]$. By continuity, there exists $\Phi \in [0, \psi] \subset [0, \epsilon]$ such that $v(\Phi) = s_1(\Phi - \pi) - s_0(\Phi)$ makes an angle Φ with the x -axis. \square

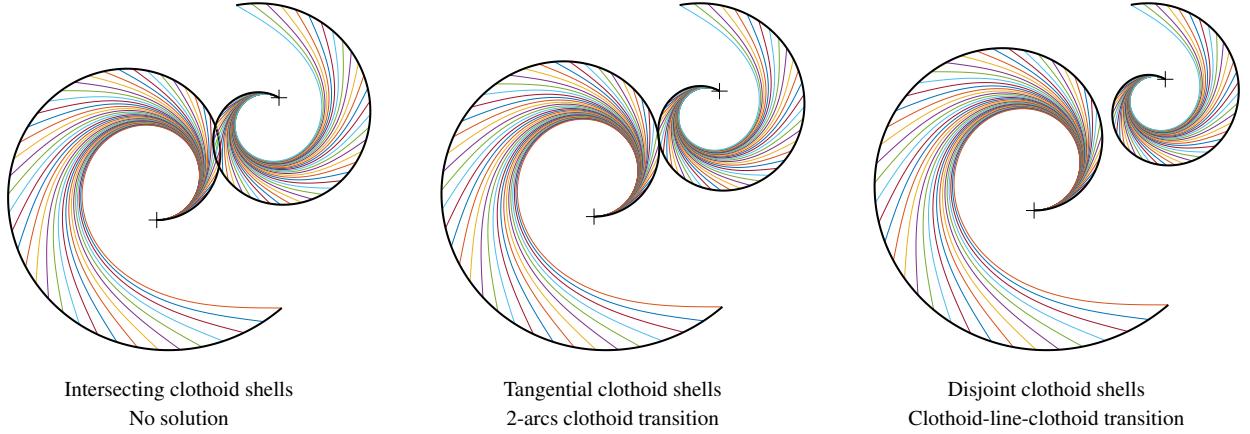


Figure 16: Clothoid shells (black curves) for clothoid-line-clothoid (CLC) transition between points with given curvatures of opposite signs. $p_0 = (0,0)$, $\theta_0 = 0$, $\kappa_0 = 1$, $p_1 = (3,3)$, $\theta_1 = -\frac{\pi}{6}$ and κ_1 is equal to 1.5 (left), 1.655 (middle) and 2.0 (right). The function s_1 has been produced with positive curvature: we are looking for the point $p_{e,1} = s_1(\Phi)$ that makes $v(\Phi)$ have an angle Φ with the x-axis. But the second part of the CLC transition goes from $p_{e,1}$ to p_1 and therefore has curvature $-\kappa_1$ at p_1 . Colored curves represent the shortest clothoids joining (p, θ, K) and a line of slope angle Φ .

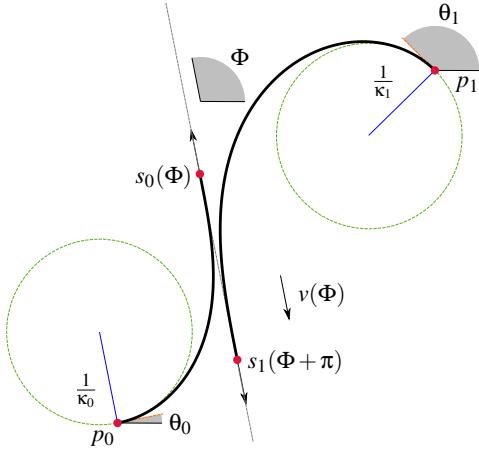


Figure 17: In this example, the order of the point is reversed on the transition line. This case is not considered as valid for Proposition 3.1 because $v(\Phi)$ makes an angle $\Phi + \pi$ with the x-axis.

Interpretations of Proposition 3.1. In our method, the tangent is computed based on a hybrid circular-elliptical interpolation of three successive points p_{i-1} , p_i and p_{i+1} [Yuk20]. Therefore, the angles

θ_0 and θ_1 are always such that the vector $p_1 - p_0$ is making a positive angle smaller than $\pi/2$ with the line (p_0, θ_0) (respectively, the vector $p_0 - p_1$ and the line $(p_1, \pi + \theta_1)$). This situation is illustrated in Figure 16. We can see in this figure that three cases can happen:

1. The clothoid shells intersect twice. The angle with the x-axis of $v(\Phi)$ therefore makes a whole round and is never equal to Φ .
2. The clothoid shells are tangent. The unique intersection point p is such that there exists Φ such that $s_0(\Phi) = s_1(\Phi + \pi) = p$. Therefore, the line part of the CLC transition is vanishes, and we have a 2-arcs clothoid transition.
3. The clothoid shells are disjoint. Let ψ be the angle that $v(\Phi)$ makes with the x-axis. In this situation, the difference $\psi - \Phi$ is first positive around $\Phi = 0$ and then negative around $\Phi = \pi$. Therefore there is a Φ such that $\psi - \Phi = 0$.

Note that in the case of same-sign curvature, the situation can be different. It is possible to find K_0, K_1 such that there exists a CLC transition from (p_0, θ_0, K_0) to (p_1, θ_1, K_1) , but not every $\kappa_0 > K_0$ and $\kappa_1 > K_1$ makes the existence of a CLC transition persist (Sec. 5.1).

It is possible that the only Φ such that $s_0(\Phi)$ and $s_1(\Phi + \pi)$ end on the same tangent line is not suitable for our CLC transition. In Fig. 17, the order of the points is reversed. However, $v(\Phi)$ makes an angle $\Phi + \pi$ with the x-axis. This case is not considered as valid in our proof, and hence further increase of the curvature is required. Therefore, Proposition 3.1 states that there exists a *valid* CLC transition, without eluding the case of reverse order of the points on the transition line.