

Title:

**Intrinsically Defined Planar Curves based on Explicit B-spline Curvature Functions**

Authors:

Norimasa Yoshida, norimasa@acm.org, Nihon Univeristy

Takafumi Saito, txsaito@cc.tuat.ac.jp, Tokyo University of Agriculture and Technology

Keywords:

Intrinsically defined curves, Explicit B-spline curvature function,  $G^1$  and  $G^2$  Hermite interpolation, Curvature comb

DOI: 10.14733/cadconfP.2021.51-55

Introduction:

In the design of aesthetic objects, controlling the curvature variation of a curve segment is an important task. Freeform curves, such as Bézier curves or B-spline curves, are widely used in many CAD systems. Most of the work has been done for maintaining the curvature monotonicity, such as [5]. Controlling the curvature variation of freeform curves, however, is not easy due to the complexity of curvature equation.

This paper introduces an intrinsically defined planar curve based on an explicit polynomial B-spline curve and its  $G^1$  and  $G^2$  Hermite interpolation method, which is an extension of our previous work [8]. The advantage of using explicit B-spline curvature functions instead of explicit Bézier curvature functions is that a wider variety of curvature variation can be represented by increasing the number of segments and modifying the knots. Especially the curvature plot shown in Fig. 3(d) cannot be represented in previous methods [8, 6]. Thus the proposed method can match a wider variety of  $G^2$  Hermite conditions. In the proposed approach, the curvature plot is specified in terms of an explicit B-spline curve and the explicit B-spline curve is integrated to generate a curve segment such that given  $G^1$  or  $G^2$  Hermite interpolation conditions are satisfied. As an application of the proposed curve, we show a method of controlling the curvature variation interactively by modifying *control curvatures* shown on the curvature comb satisfying  $G^1$  or  $G^2$  Hermite conditions.

Curves based on Explicit B-spline Curvature Functions:

This section describes an intrinsically defined curves based on an explicit polynomial B-spline curvature function. To simplify the description, we consider a curve in the standard form where the starting point and its tangent are the origin and  $[0 \ 1]^T$ , respectively. A curve in a general position can be obtained by performing an appropriate transformation to the curve in the standard form.

Let  $s$  be the arc length and  $\kappa(s)$  be the curvature function. The tangential angle  $\theta(s)$  is computed by

$$\theta(s) = \int_0^s \kappa(u) du. \quad (2.1)$$

The curve position  $\mathbf{P}(s)$  is

$$\mathbf{P}(s) = \begin{bmatrix} \int_0^s \cos(\theta(u)) du \\ \int_0^s \sin(\theta(u)) du \end{bmatrix}. \quad (2.2)$$

Eqn. (2.1), (2.2) are standard equations in differential geometry. If we use the curvature function of log-aesthetic curves [7] as  $\kappa(s)$ , Eqn. (2.2) immediately becomes the equation of log-aesthetic curves.

An explicit polynomial B-spline curve of degree  $n$  with control points  $p_i (i = 1, 2, \dots, m)$  and knots  $t_1, t_2, \dots, t_{m+n-1}$  in polar form [2, 4] is a curve whose  $x$  coordinate is the parameter  $t$  and  $y$  coordinate is

$$p(t) = \sum_{i=1}^m B_i^n(t) p_i \quad (t \in [t_n, t_m]) \quad (2.3)$$

Here,  $B_i^n(t)$  is a B-spline basis function. In explicit B-spline curves,  $t$  coordinate  $c_i$  for each control point  $p_i$ , known as Greville abscissa, is the average of  $n$  consecutive knots [4]:

$$c_i = \frac{t_i + t_{i+1} + \dots + t_{i+n-1}}{n}. \quad (2.4)$$

Let  $s_t$  be the arc length of a curve segment and  $\kappa_i (i = 1, 2, \dots, m)$  be *control curvatures*. Without loss of generality, we assume  $t_n = 0$  and  $t_m = 1$  to make the parameter range of an explicit B-spline curve  $[0, 1]$ . The curvature function  $\kappa(s)$  in terms of arc length  $s$  in explicit polynomial B-spline form is

$$\kappa(s) = \sum_{i=1}^m B_i^n(s/s_t) \kappa_i \quad (s \in [0, s_t]). \quad (2.5)$$

Plugging Eqn. (2.5) into Eqn. (2.1) and then equation (2.1) into Eqn. (2.2), curve points in the standard form can be computed. Note that to compute Eqn. (2.1), numerical integration is not necessary. This integration can be computed by converting an explicit B-spline curve to explicit Bézier curves using knot insertion and using the characteristics of explicit Bézier curves [4, 8].

In an explicit B-spline curvature function, detecting an inflection point and curvature extrema is straightforward. To detect an inflection point, we check if the derivative of  $\kappa(s)$  with respect to  $s$  becomes 0 using Bézier clipping [3]. Similarly, to detect curvature extrema, we need to check if  $\kappa(s)$  becomes 0 using Bézier clipping. For a low degree curvature function, we can directly compute such points.

### $G^1$ and $G^2$ Hermite Interpolation Method:

In  $G^1$  Hermite interpolation, a starting point  $\mathbf{P}_s$  and an endpoint  $\mathbf{P}_e$  with their unit tangent vectors  $\mathbf{t}_s$  and  $\mathbf{t}_e$  are given. In  $G^2$  Hermite interpolation, control curvatures  $\kappa_s$  and  $\kappa_e$  at  $\mathbf{P}_s$  and  $\mathbf{P}_e$  are also given. In case of  $G^2$  Hermite interpolation, the first and the last control curvatures are set to  $\kappa_1 = \kappa_s$  and  $\kappa_m = \kappa_e$ .

In  $G^1$  or  $G^2$  Hermite conditions, we use the standard form where  $\mathbf{P}_s$  is placed at the origin and  $\mathbf{t}_s = [1 \ 0]^T$ . Let  $\theta_d$  be the angle formed by  $\mathbf{t}_s$  and  $\mathbf{t}_e$ . Thus  $\mathbf{t}_e = [\cos(\theta_d) \ \sin(\theta_d)]^T$ . Given the degree  $n$ , the number control curvatures  $m$  and  $\kappa_i (i = 1, \dots, m)$ , the arc length  $s_t$  of the curve segment is given by

$$s_t = \frac{\theta_d}{\int_0^1 (\sum_{i=1}^m B_i^n(t) \kappa_i) dt}. \quad (2.6)$$

In  $G^1/G^2$  Hermite interpolation in the standard form, the generated curve must satisfy the following conditions:

$$\begin{aligned} (a) \quad & \mathbf{P}(0) = [0 \ 0]^T, \quad (b) \quad \mathbf{P}(s_t) = [0 \ 1]^T, \\ (c) \quad & [\cos(\theta(0)) \ \sin(\theta(0))]^T = \mathbf{t}_s, \quad (d) \quad [\cos(\theta(s_t)) \ \sin(\theta(s_t))]^T = \mathbf{t}_e. \end{aligned} \quad (2.7)$$

Conditions (a), (b) are always satisfied by using Eqn. (2.1) and (2.2), respectively. Condition (d) is always satisfied by using the arc length  $s_t$  computed by Eqn. (2.6). The remaining condition (c) is satisfied by an optimization that minimizes  $|\mathbf{P}(s_t) - \mathbf{P}_e|$ .

Fig. 1 shows typical cases how the control curvatures are computed. In  $G^1$  Hermite interpolation, we typically use  $\kappa_1$  and  $\kappa_m$  as optimization parameters and  $\kappa_2, \dots, \kappa_{m-1}$  are either user-specified or computed by some kind of interpolation, such as linear interpolation. In  $G^2$  Hermite interpolation,  $\kappa_1 (= \kappa_s)$  and  $\kappa_m (= \kappa_e)$  are given in addition to  $G^1$  Hermite conditions. Therefore, we typically use  $\kappa_2$  and  $\kappa_{m-1}$  as optimization parameters and compute  $\kappa_3, \dots, \kappa_{m-2}$  similarly by some kind of interpolation or specify them.

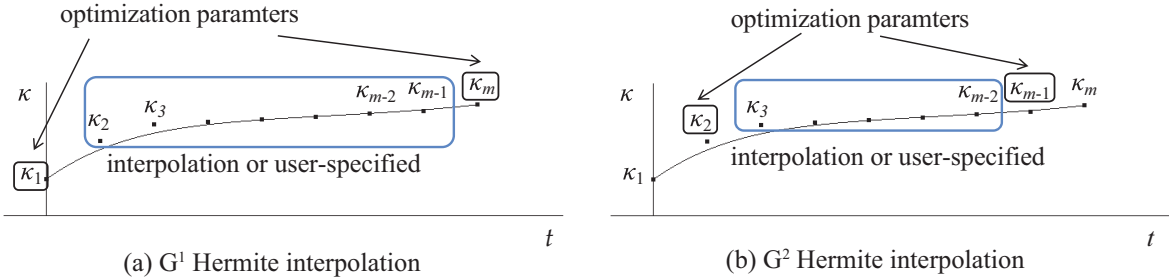


Fig. 1: Typical  $G^1$  and  $G^2$  Hermite interpolations

In our implementation, we use an interpolation using a parameter  $p$  such that the curvature plot changes depending on the value of  $p$ . Fig. 2 shows examples of typical  $G^1$  Hermite interpolation where  $\kappa_1$  and  $\kappa_6$  are used as optimization parameters.  $\kappa_2, \dots$  and  $\kappa_5$  are equal to  $\kappa_6$  when  $p = -1$ , linearly interpolated when  $p = 0$ , and equal to  $\kappa_1$  when  $p = 1$ . When  $p$  takes a value between between -1 and 0, control curvatures are smoothly interpolated.

For typical  $G^1$  Hermite interpolation, we set  $i_1 = 1$  and  $i_m = m$ . The monotonicity of curvature is always guaranteed if  $p \in [-1, 1]$ , though the curve may include an inflection point. For typical  $G^2$  Hermite interpolation, we set  $i_1 = 1$  and  $i_m = m - 1$ .

### Results:

Fig. 2 shows examples of generated curves and their curvature plots for typical  $G^1$  Hermite conditions. Note that  $\kappa$  values of curvature plots are scaled so that the curvature plot fits in the graph. In Fig.2(a)-(c), the number of control curvatures is 6 and the knot vector is  $[0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1]$  but the value of  $p$  is different. In the curvature plot, black circles represent control curvatures  $\kappa_1, \dots, \kappa_6$  from left to right. Tick marks in  $t$ -axis represent knots  $t_i$ . The rightmost tick mark is placed at  $t = 1$ . Since we use Bézier end conditions, the multiplicity of knots at  $t = 0$  and  $t = 1$  are set to  $n$ . By modifying the value of  $p$ , the curve shape and the curvature plot can be variously controlled for the same  $G^1$  Hermite condition guaranteeing the monotonicity of curvature. Depending on  $G^1$  Hermite conditions, the generated curve may include an inflection point.

In  $G^2$  Hermite interpolation, we are given  $\kappa_s$  and  $\kappa_e$  in addition to  $G^1$  Hermite conditions. Let  $\theta_e$  be the angle between  $\mathbf{P}_e - \mathbf{P}_s$  and  $\mathbf{t}_s$ . Fig. 3 shows examples of  $G^2$  Hermite interpolation for  $\theta_e = \frac{1}{9}\pi$ ,  $\theta_d = \frac{1}{3}\pi$  and  $|\mathbf{P}_e - \mathbf{P}_s| = 1$ . Fig. 3(a) shows the region of  $(\kappa_s, \kappa_e)$  where curves with monotonically varying curvature can be generated. See [1, 8] for details. In Fig. 3 (b),(c),(d),(e),  $(\kappa_s, \kappa_e)$  are the points denoted as **a**, **b**, **c** and **d** in Fig. 3(a), respectively. The degree and the number of control curvatures are 3 and 8, respectively. The control curvatures and the knots are set manually to make the curvature monotonically varying. Based on these results and other experiments we have performed, proposed curves

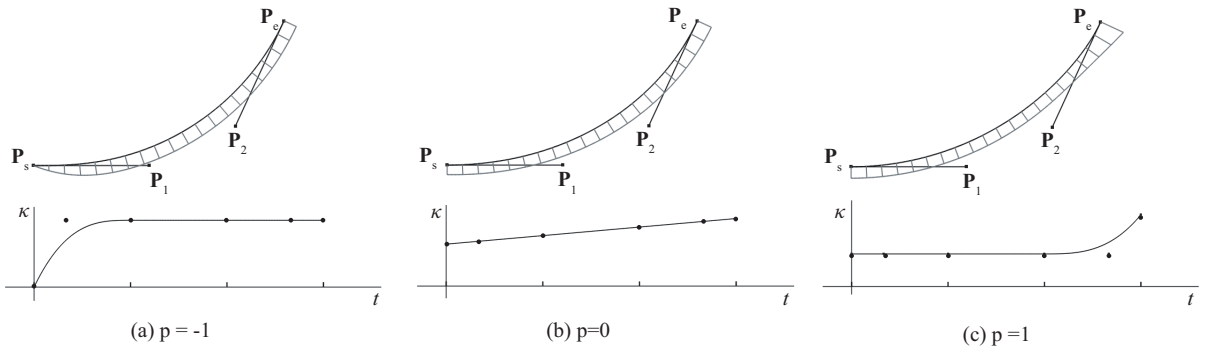


Fig. 2:  $G^1$  Hermite interpolation using explicit cubic B-spline curves

can match a wide variety of  $G^2$  Hermite conditions by appropriately specifying the number of control points, control curvatures and a knot vector. Especially for the pairs of  $(\kappa_s, \kappa_e)$  denoted as **a**, **b**, **c**,  $G^2$  Hermite interpolation with monotonically varying curvature is impossible by the previous method[8].

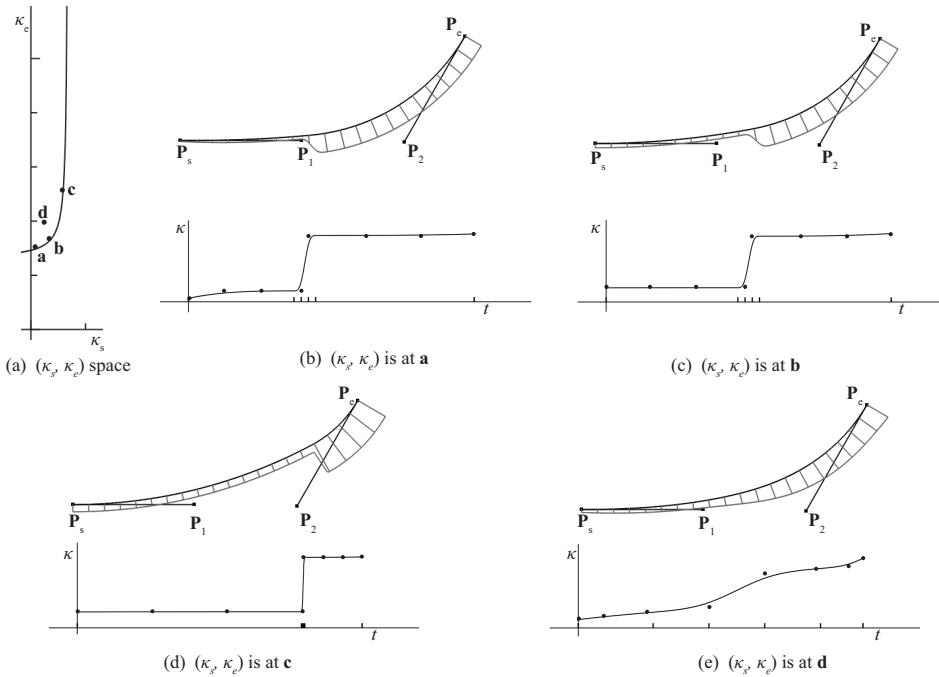


Fig. 3: Examples of  $G^2$  Hermite interpolation

Fig. 4 (a) shows a curve segment and its curvature plot satisfying given  $G^1$  Hermite conditions. On the curvature comb, control curvatures are plotted as black dots in the normal direction of the curve. In Fig. 4 (b), control curvature  $\kappa_4$  indicated by an arrow is moved. To control the curvature variation, the control curvature can be interactively moved in the normal direction of the curve. To modify  $\kappa_4$ , the scaled (signed) difference between the original position of  $\kappa_4$  and the moved position in the normal direction of the curve is added to the original  $\kappa_4$ . After  $\kappa_4$  is modified, an optimization is performed to

satisfy the given  $G^1$  Hermite conditions. In this example,  $\kappa_1$  and  $\kappa_6$  are used for optimization parameters. In Fig. 4 (c), control curvature  $\kappa_5$  indicated by an arrow is further moved. Note that large movement of a control curvature may generate an undesirable curve shape including inflection points and curvature extrema. Controlling a control curvature can also be done in the curvature plot.

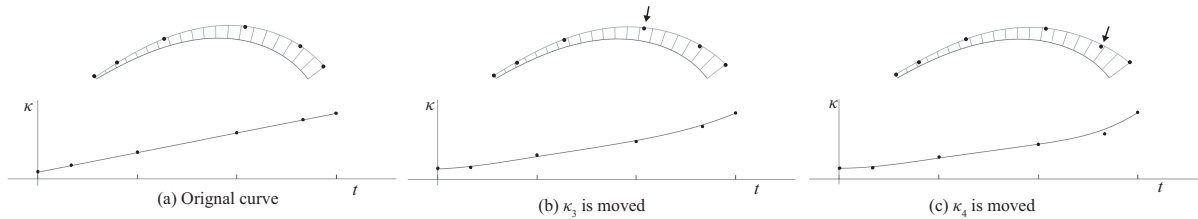


Fig. 4: Direct control of a control curvature on the curvature comb

### Conclusions:

In this paper, we proposed intrinsically defined curves based on explicit B-spline curvature functions. In the proposed method, two of the control curvatures are used as an optimization for fitting  $G^1$  or  $G^2$  Hermite interpolations and other control curvatures (except for two control curvatures specified in case of  $G^2$  Hermite interpolation) are either interpolated or user-specified. Given  $G^1$  or  $G^2$  Hermite interpolation conditions, the proposed curve has a wider variety of curvature variation than our previous work[8] due to the use of an explicit B-spline curvature function. We also proposed a method for interactively controlling the curvature variation by moving the control curvature shown on the curvature comb or curvature plot. Future work includes a method for covering all the viable region of  $G^2$  Hermite interpolation and the extension to space curves

### References:

- [1] Dietz, D. A.; Piper, B.: Interpolation with cubic spirals, Computer Aided Geometric Design, 21(2), 2004, 165-180. <https://doi.org/10.1016/j.cagd.2003.09.002>
- [2] Ramshaw, L.: Blossoms are polar forms, Computer Aided Geometric Design, 6, 1989, 323-358. [https://doi.org/10.1016/0167-8396\(89\)90032-0](https://doi.org/10.1016/0167-8396(89)90032-0)
- [3] Sederberg, T. W.; Nishita, T.: Curve intersection using Bézier clipping, Computer-Aided Design, 22(9), 1990, 538-549. [https://doi.org/10.1016/0010-4485\(90\)90039-F](https://doi.org/10.1016/0010-4485(90)90039-F)
- [4] Sederberg, T. W.: Computer Aided Geometric Design, BYU Course Notes, 2017. <http://hdl.lib.byu.edu/1877/2822>
- [5] Wang, Y.; Zhao, B.; Zhang, L.; Xu, J.; Wang, K.; Wang, S.: Designing fair curves using monotone curvature pieces. Computer Aided Geometric Design, 21(5), 2004, 515-527. <https://doi.org/10.1016/j.cagd.2004.04.001>
- [6] Watanabe, Y.; Saito, T.; Kuroda, M.: A method for generation of curves from specified curvature profile(in japanese). In Graphics and CAD Workshop of Information Processing Society of Japan, 7-12, 1997.
- [7] Yoshida, N.; Saito, T.: Interactive Aesthetic Curve Segments, The Visual Computer, 22(9-11), 2006, 896-905. <http://dx.doi.org/10.1007/s00371-006-0076-5>
- [8] Yoshida, N.; Saito, T.: Planar Curves based on Explicit Bézier Curvature Functions, Computer Aided Design and Applications, 17(1), 2020, 77-87. <https://doi.org/10.14733/cadaps.2020.77-87>