Continuity-Enhancing Degree Elevation and Splits

CEM YUKSEL, University of Utah, USA

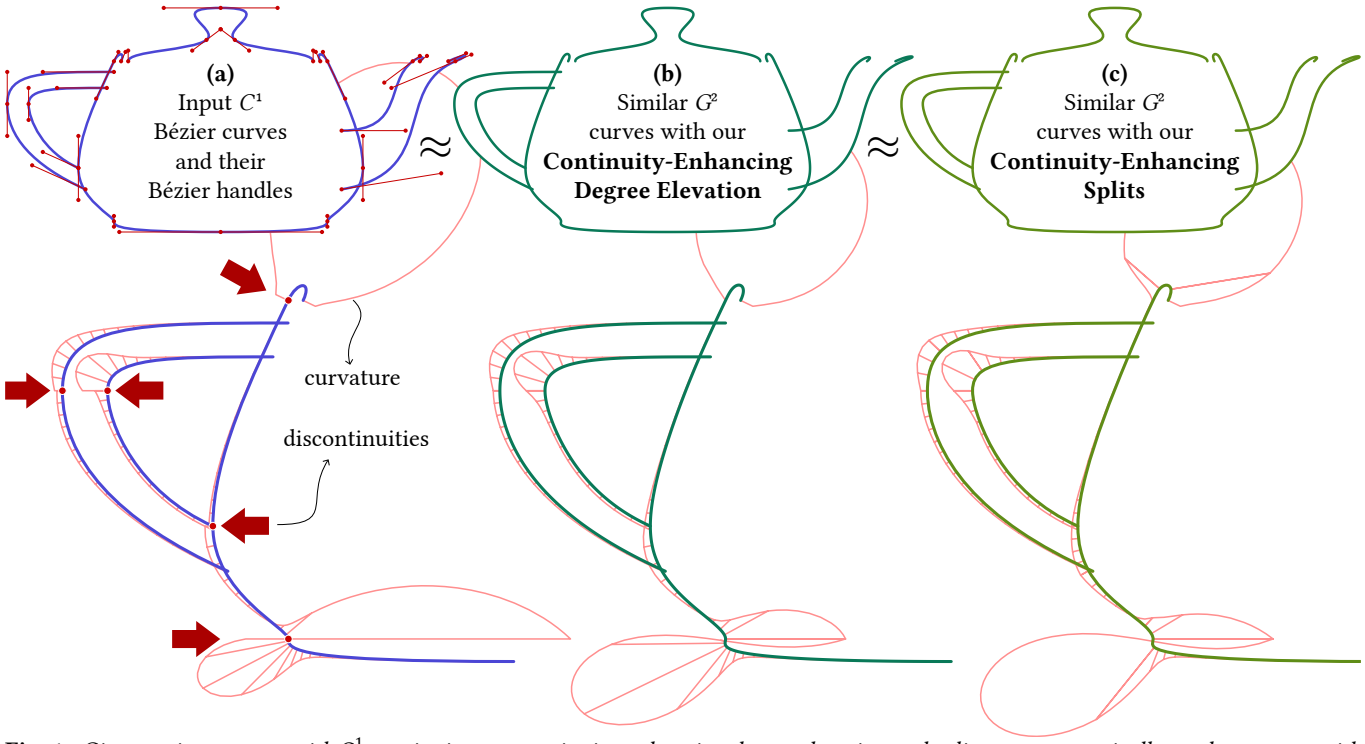


Fig. 1. *Given an input curve with C^1 continuity, our continuity-enhancing degree elevation and splits can automatically produce curves with similar shapes but with C^2 or G^2 continuity. In this example,* **(a)** *the piecewise cubic curves of the Utah Teapot model are converted to* **(b)** *piecewise quintic splines and* **(c)** *piecewise cubic splines, both with a similar shape but with G^2 continuity. The bottom row visualizes the (signed square root of the) curvature, highlighting the discontinuities of the input curves, which are entirely eliminated with either of our two methods.*

Cubic polynomial curve modeling with Bézier handles is ubiquitous, but it provides no practical mechanism for achieving curvature continuity. We present two distinct solutions for this problem. The first one is continuity-enhancing degree elevation that offers a general solution to achieve any level of continuity by converting the given curve to a polynomial of a higher degree. Our second solution is continuity-enhancing splits, which is specific to cubic curves and achieves curvature continuity by splitting the curve pieces but maintaining the piecewise cubic polynomial form. Both of these solutions utilize a local optimization process with a closed-form solution, achieving continuity enhancement with a constant computation overhead per piece. We also explain how to incorporate linearity constraints to seamlessly form linear curve pieces, when desired. Our solutions are effective in extending the popular curve modeling interface with Bézier handles to splines with curvature (or higher) continuity. Furthermore, we show that our solutions can also be used for defining new interpolating curve formulations with desirable properties, and they can be used with higher-dimensional curves or surfaces.

CCS Concepts: • **Computing methodologies** → **Parametric curve and surface models**.

ACM Reference Format:
Cem Yuksel. 2026. Continuity-Enhancing Degree Elevation and Splits. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3799902.3811063

Author's Contact Information: Cem Yuksel, University of Utah, Salt Lake City, Utah, USA, cem@cemyuksel.com.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/26/07
https://doi.org/10.1145/3799902.3811063

1 Introduction

Splines (i.e. polynomial curves) have numerous applications in computer graphics and beyond. In particular, piecewise cubic (degree 3) splines with Bézier handles (Fig. 1) are ubiquitous in virtually all applications that offer user-specified curve editing. Typically, curve editing interfaces allow "locking" the Bézier handles such that the *intermediate control points* at the ends of these handles and the *interpolated control point* between them are on a line. This simple restriction ensures that the resulting piecewise cubic curve has parametric C^1 or geometric G^1 (i.e., direction) continuity.

Achieving C^2 or G^2 (i.e., curvature) continuity, however, has been a major challenge. Though piecewise cubic curves can have C^2 continuity (e.g., B-Splines), curve formulations with C^2 continuity do not allow defining them using the familiar Bézier handles of C^1 piecewise cubic splines.

In this paper, we present two distinct solutions for producing a curve with parametric C^2 or geometric G^2 continuity (or higher) from a given piecewise cubic spline with C^1 continuity:

- **Continuity-Enhancing Degree Elevation:** Our first solution applies a degree elevation to form additional control points, which are then placed automatically to improve the curve's continuity. This is a general solution for input polynomials of any degree to achieve continuity of any desired level.
- **Continuity-Enhancing Splits:** Our second solution produces a piecewise cubic spline with curvature continuity by splitting each piece of a given C^1 cubic curve into three parts. The first and last parts ensure curvature continuity with the neighboring pieces, and the middle part connects the two end parts with C^2 continuity.

In both solutions, the additional control points are automatically placed to ensure improved continuity where neighboring pieces join by solving a local optimization problem with a closed-form solution, which involves a minor constant computation overhead per curve piece. They both allow, for the first time to our knowledge, defining curves with curvature continuity using familiar Bézier handles–the most popular method for curve editing. The resulting curves have similar shapes to the input C^1 curves defined by the given Bézier handles (Fig. 1). This makes them easy to control for anyone familiar with this common interface.

Using our continuity-enhancing degree elevation, any arbitrary level of continuity can be achieved utilizing the same interface, though curves with C^5 or higher continuity are prone to forming unintended wiggles, indicating the practical limits of this approach.

In addition, we present methods for preserving linear portions of a given input curve during continuity enhancement, which is an important practical consideration. Furthermore, we show that our methods can be used for defining new types of interpolating curves with desirable properties, and they can be used for higher-dimensional curves or surfaces.

2 Background

There are too many curve formulations in computer graphics, computer-aided geometric design, and mathematics, more broadly, to provide an exhaustive summary in this paper [Farin 2002; Hoschek and Lasser 1993]. Therefore, we briefly discuss them at a high level and specifically mention only a few, mainly concentrating on recent work and curves commonly used in practice today.

Most curve formulations fall into one of these two groups: *interpolating curves* and *approximating curves*. Interpolating curves, as the name suggests, go through each control point. This allows precisely specifying some positions on the curve as control points. They differ in how the curves interpolate these control points. In particular, most interpolating curves with C^2 continuity are prone to forming undesirable visually-significant features in the form of cusps and self-intersections between the interpolated control points (e.g. C^2 Catmull-Rom curves [Catmull and Rom 1974]). More recent interpolating curve formulations prevent such problems by placing curvature maxima at the control points by solving a global optimization problem [Yan et al. 2017] or blending interpolating curve pieces [Hu and Lai 2025; Jiang and Chen 2025; Yuksel 2020]. Regardless of how the interpolation is formulated, interpolating curves do not offer a mechanism to easily control the shape of the interpolating regions, which can only be done by adding intermediate control points. Therefore, even simple tasks like creating a linear segment might be entirely impossible [Wang et al. 2024; Yan et al. 2019, 2017], require precisely placing at least four consecutive control points on a line [Hu and Lai 2025; Jiang and Chen 2025; Yuksel 2020], or can only be formed away from control points [Binninger and Sorkine-Hornung 2022; Wang et al. 2020].

Approximating curves, such as B-Splines, can easily achieve C^2 continuity using cubic polynomials and without introducing unintended cusps and self-intersections. Yet, precisely controlling approximating curves is even a bigger challenge, since they do not go through the control points.

Bézier curves do not neatly fall into either of these two categories, since they interpolate the first and the last control points, but not the *intermediate control points* between them. They can be considered a generic formulation, since any polynomial curve can be represented in a Bézier form. Most importantly, they allow precisely controlling not only the positions of the interpolated points but also the shape of the interpolating pieces via the intermediate control points. The pervasive application of the Bézier interface in virtually all software that offers curve editing can be attributed to this property. In particular, piecewise cubic Bézier curves are omnipresent, since the two intermediate control points they have allow independently controlling the tangent of the curve on either side of each curve piece with the minimum number of intermediate control points. Cubic polynomials are also the lowest degree required for forming 3D curves, as quadratics (degree 2) are always planar.

Achieving C^1 (or G^1) continuity with the cubic Bézier control points is trivial: one simply needs to linearly place the intermediate control points immediately before and after an interpolated control point. In most curve editing software, when the user modifies one of the intermediate control points, the position of the interpolated control point is preserved, and the other intermediate control point on its other side is automatically moved to maintain linearity.

This popular interface, however, does not offer any help for specifying curves with C^2 (or G^2) continuity. This is because achieving C^2 continuity where two curve pieces join at an interpolated control point not only depends on the intermediate control points immediately neighboring the interpolated control point, but also on the other two intermediate control points near the other ends of the two pieces. Thus, achieving C^2 continuity at an interpolated control point is impossible without overriding the user-specified curve tangents elsewhere, except for special cases.

The methods we introduce in this paper offer solutions to this problem. They allow using the intuitive piecewise cubic Bézier control point interface while automatically ensuring C^2 continuity for the generated curves.

3 Continuity Enhancement

Our solutions for continuity enhancement are based on modifying the control points of a given input curve to achieve the desired level of continuity. Therefore, before we discuss our solutions, in this section, we introduce the underlying optimization problem they utilize to minimize the modification of the input curve while satisfying the necessary continuity conditions.

3.1 Parametric Continuity Conditions

A Bézier curve of degree n can be considered to have C^∞ continuity between its two endpoints, since all derivatives are continuous, though they are zero beyond the n^{th} derivative. Therefore, the continuity for a piecewise Bézier curve of degree n depends solely on the continuity at the interpolated control points where two curve pieces join. More specifically, to achieve C^k continuity, all derivatives up to (and including) k must be the same for each pair of consecutive pieces where they join at an interpolated control point.

Let $\mathbf{B}(t)$ represent the Bernstein polynomial of degree n for a curve piece using Bézier control points $\mathbf{b}_i$ where $i \in \{0, 1, \cdots, n\}$, using a normalized parameter $t \in [0, 1]$ for this piece, such that

$$\mathbf{B}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i}\, t^i\, \mathbf{b}_i \,, \quad \text{where} \quad \binom{n}{i} = \frac{n!}{i!(n-i)!} \,. \tag{1}$$

In a piecewise polynomial curve, we can represent the previous piece with a similar notation using control points $\mathbf{b}_i$ where $i \in \{-n, \cdots, -1, 0\}$. Note that the interpolated control point $\mathbf{b}_0$ is shared by the two pieces, which ensures C^0 continuity. To achieve C^k continuity, all derivatives up to k on either side of $\mathbf{b}_0$ must match. Let $\mathbf{b}_\oplus^{(k)}$ and $\mathbf{b}_\ominus^{(k)}$ represent the k^{th} derivatives on either side with respect to their normalized parameters, such that

$$\mathbf{b}_\ominus^{(k)} = \frac{n!}{(n-k)!}\mathbf{D}_{-k} \qquad \text{and} \qquad \mathbf{b}_\oplus^{(k)} = \frac{n!}{(n-k)!}\mathbf{D}_{k} \,, \tag{2}$$

where

$$\mathbf{D}_{-k} = \sum_{i=0}^{k} (-1)^i \binom{k}{i} \mathbf{b}_{-i} \quad \text{and} \quad \mathbf{D}_{k} = \sum_{i=0}^{k} (-1)^{i+k} \binom{k}{i} \mathbf{b}_{i} \,. \tag{3}$$

We can write the condition for C^k continuity as

$$\mathbf{D}_{-i} = \alpha^i \mathbf{D}_i \,, \qquad \forall\, i \le k \,. \tag{4}$$

where α is the relative parameter speeds of the two curve pieces that join at $\mathbf{b}_0$ such that

$$\alpha = \frac{d_{-1}}{d_1} \,, \qquad \text{using} \qquad d_i = \|\mathbf{D}_i\| \,. \tag{5}$$

Notice that $\mathbf{D}_k$ depends on the first k intermediate control points of the curve piece and $\mathbf{D}_{-k}$ depends on the last k intermediate control points of the previous piece. Therefore, we can ensure C^k continuity at $\mathbf{b}_0$ by locally adjusting the k intermediate control points on either side of it, such that we match the derivatives on either side, as we explain below.

3.2 Parametric Continuity Enhancement

If a given curve lacks the desired level of continuity at an interpolated control point $\mathbf{b}_0$, we must modify the intermediate control points around it to satisfy the continuity conditions. This means changing the shape of the curve, and, unfortunately, it is unavoidable. Therefore, our goal is to minimize the change in the curve's control points, while matching the derivatives on either side of $\mathbf{b}_0$.

We process each derivative up to k in order. Starting from a curve with C^{k-1} continuity, where $k \ge 1$, it is sufficient to only modify intermediate control points $\mathbf{b}_k$ and $\mathbf{b}_{-k}$ to match the k^{th} derivatives without altering the lower order derivatives. This ensures that our modification of $\mathbf{b}_k$ and $\mathbf{b}_{-k}$ does not alter the lower order derivatives. Let $\mathbf{b}_k^*$ and $\mathbf{b}_{-k}^*$ represent the new positions for these control points, respectively. The C^k continuity condition for the modified curve is

$$\mathbf{D}_{-k} + (-1)^k \left(\mathbf{b}_{-k}^* - \mathbf{b}_{-k}\right) = \alpha^k \left(\mathbf{D}_k + \mathbf{b}_k^* - \mathbf{b}_k\right) \,, \tag{6}$$

By rearranging the terms in this equation, we can write one of the modified control points based on the other one, such that

$$\mathbf{b}_{-k}^* = (-1)^k \alpha^k \left(\mathbf{D}_k + \mathbf{b}_k^* - \mathbf{b}_k\right) - (-1)^k\, \mathbf{D}_{-k} + \mathbf{b}_{-k} \,. \tag{7}$$

To minimize the change on the curve due to continuity enhancement, we would like the modified control points to be as close as possible to the original ones, minimizing a loss function of the form

$$\mathcal{L} = \|\mathbf{b}_k^* - \mathbf{b}_k\|^2 + \|\mathbf{b}_{-k}^* - \mathbf{b}_{-k}\|^2 \,. \tag{8}$$

To derive a closed-form solution for the local optimization problem that minimizes this loss function, we introduce

$$\mathbf{b}_{-k}^+ = \frac{1}{\alpha^k}\mathbf{D}_{-k} - \mathbf{D}_k + \mathbf{b}_k \,. \tag{9}$$

This allows writing the second term of the loss function as

$$\begin{aligned} \mathbf{b}_{-k}^* - \mathbf{b}_{-k} &= (-1)^k \alpha^k \left(\mathbf{D}_k + \mathbf{b}_k^* - \mathbf{b}_k\right) - (-1)^k \mathbf{D}_{-k} + \cancel{\mathbf{b}_{-k}} - \cancel{\mathbf{b}_{-k}} \\ &= (-1)^k \alpha^k \left(\mathbf{b}_k^* - \mathbf{b}_{-k}^+\right) \,, \end{aligned} \tag{10}$$

resulting in

$$\mathcal{L} = \|\mathbf{b}_k^* - \mathbf{b}_k\|^2 + \alpha^{2k} \|\mathbf{b}_k^* - \mathbf{b}_{-k}^+\|^2 \,. \tag{11}$$

Note that here the only unknown is $\mathbf{b}_k^*$, and its position that minimizes the loss function is between $\mathbf{b}_k$ and $\mathbf{b}_{-k}^+$. More specifically, the loss function is minimized when

$$\mathbf{b}_k^* = \frac{1}{1+\alpha^{2k}}\mathbf{b}_k + \frac{\alpha^{2k}}{1+\alpha^{2k}}\mathbf{b}_{-k}^+ \,. \tag{12}$$

The other modified control point $\mathbf{b}_{-k}^*$ can then be calculated using Eq. 7 or 10. Combining these equations, we can write the updated control points as

$$\mathbf{b}_{-k}^* = \mathbf{b}_{-k} + (-1)^k \Delta\mathbf{b}_{-k} \qquad \text{and} \qquad \mathbf{b}_k^* = \mathbf{b}_k + \Delta\mathbf{b}_k \,, \tag{13}$$

where

$$\Delta\mathbf{b}_{-k} = \frac{\alpha^k\, \mathbf{D}_k - \mathbf{D}_{-k}}{1+\alpha^{2k}} \qquad \text{and} \qquad \Delta\mathbf{b}_k = \alpha^k \frac{\mathbf{D}_{-k} - \alpha^k\, \mathbf{D}_k}{1+\alpha^{2k}} \tag{14}$$

See the supplemental document Section S1 for the derivation steps and reformulations to avoid the singularities in Section S2.

This formulation allows modifying the intermediate control points around the interpolated control point $\mathbf{b}_0$ until the desired continuity is achieved by applying minimal modification to the intermediate control points.

3.3 Geometric Continuity Enhancement

Above, we discussed how to enhance the parametric (i.e., C^k) continuity. In many applications, however, geometric (i.e., G^k) continuity is sufficient. Parametric continuity also implies geometric continuity. However, since geometric continuity does not require the parametric derivatives to match, it is less restrictive and can be achieved with smaller modifications to existing control points.

More specifically, geometric continuity conditions [DeRose and Barsky 1985] up to G^4 can be written as

$$\mathbf{b}_{\ominus}^{(1)} = \alpha\, \mathbf{b}_{\oplus}^{(1)} \tag{15}$$

$$\mathbf{b}_{\ominus}^{(2)} = \alpha^2\, \mathbf{b}_{\oplus}^{(2)} + \beta\, \mathbf{b}_{\oplus}^{(1)} \tag{16}$$

$$\mathbf{b}_{\ominus}^{(3)} = \alpha^3\, \mathbf{b}_{\oplus}^{(3)} + 3\alpha\beta\, \mathbf{b}_{\oplus}^{(2)} + \gamma\, \mathbf{b}_{\oplus}^{(1)} \tag{17}$$

$$\mathbf{b}_{\ominus}^{(4)} = \alpha^4\, \mathbf{b}_{\oplus}^{(4)} + 6\alpha^2\beta\, \mathbf{b}_{\oplus}^{(3)} + (4\alpha\gamma + 3\beta^2)\mathbf{b}_{\oplus}^{(2)} + \delta\, \mathbf{b}_{\oplus}^{(1)} \tag{18}$$

where $\alpha > 0$, β, γ, and δ are scalar factors due to reparameterization of the two curve pieces. Since we ignore parameterization for geometric continuity, they can be chosen arbitrarily. Note that in the special case of $\beta = \gamma = \delta = 0$, we also get parametric continuity.

Notice that the G^1 condition in Eq. 15 is equivalent to the C^1 condition in Eq. 4, since we do not require $\alpha = 1$ for parametric continuity, but only require all other factors to be zero.

The G^2 continuity condition in Eq. 16 using modified control points $\mathbf{b}_2^*$ and $\mathbf{b}_{-2}^*$ can be simplified as

$$\mathbf{b}_{-2}^* = \mathbf{b}_{-1} + \alpha^2(\mathbf{b}_2^* - \mathbf{b}_1) + \beta'\mathbf{D}_1\,, \text{ where} \tag{19}$$

$$\beta' = \frac{1}{n-1}\beta - \alpha^2 - \alpha\,. \tag{20}$$

See the supplemental document Section S3 for the derivations of these and other geometric continuity conditions. Using this equation, we can write our loss function from Eq. 8 as

$$\mathcal{L} = \|\mathbf{b}_2^* - \mathbf{b}_2\|^2 + \alpha^4\|\mathbf{b}_2^* - \mathbf{b}_{-2}^+\|^2\,, \text{ where} \tag{21}$$

$$\mathbf{b}_{-2}^+ = \mathbf{b}_1 + \frac{1}{\alpha^2}(\mathbf{b}_{-2} - \mathbf{b}_{-1} - \beta'\mathbf{D}_1)\,. \tag{22}$$

For any β', $\mathbf{b}_2^*$ that minimizes this loss function can be found, as before, using Eq. 12. The value of β' that minimizes the loss produces the shortest distance between $\mathbf{b}_2$ and $\mathbf{b}_{-2}^+$; therefore, we set

$$\beta' = \arg\min_{\beta'} \left\|\mathbf{b}_2 - \mathbf{b}_{-2}^+\right\|^2\,. \tag{23}$$

We can solve this by setting $\partial\left\|\mathbf{b}_2 - \mathbf{b}_{-2}^+\right\|^2/\partial\beta' = 0$, resulting in

$$\beta' = -\frac{\mathbf{D}_1}{d_1^2}\cdot\Big(\alpha^2(\mathbf{b}_2 - \mathbf{b}_1) - (\mathbf{b}_{-2} - \mathbf{b}_{-1})\Big)\,. \tag{24}$$

Then, we can find $\mathbf{b}_2^*$ using Eq. 12 and finally $\mathbf{b}_{-2}^*$ using Eq. 10 or 19.

For G^3 continuity, the additional free variable γ can be calculated similarly, starting with a simplified form of Eq. 17, assuming modifications to only $\mathbf{b}_3$ and $\mathbf{b}_{-3}$, such that

$$\mathbf{b}_{-3}^* = \mathbf{b}_{-2} - \alpha^3(\mathbf{b}_3^* - \mathbf{b}_2) + \beta''\mathbf{D}_2 + \gamma'\mathbf{D}_1\,, \text{ where} \tag{25}$$

$$\beta'' = -\frac{3}{n-2}\alpha\beta + 2\alpha^3 + 2\alpha^2\,, \text{ and} \tag{26}$$

$$\gamma' = -\frac{1}{(n-1)(n-2)}\gamma + \frac{2}{n-1}\beta + \alpha^3 - \alpha\,. \tag{27}$$

In this case, since α and β are known (starting with a curve with G^2 continuity), following similar steps as above, we just need to solve for γ' (or γ) that minimizes the loss function

$$\mathcal{L} = \|\mathbf{b}_3^* - \mathbf{b}_3\|^2 + \alpha^6\|\mathbf{b}_3^* - \mathbf{b}_{-3}^+\|^2\,, \text{ where} \tag{28}$$

$$\mathbf{b}_{-3}^+ = \mathbf{b}_2 + \frac{1}{\alpha^3}\left(\mathbf{b}_{-2} - \mathbf{b}_{-3} + \beta''\mathbf{D}_2 + \gamma'\mathbf{D}_1\right)\,, \text{ such that} \tag{29}$$

$$\gamma' = \arg\min_{\gamma'} \left\|\mathbf{b}_3 - \mathbf{b}_{-3}^+\right\|^2\,, \text{ resulting in} \tag{30}$$

$$\gamma' = \frac{\mathbf{D}_1}{d_1^2}\cdot\Big(\alpha^3(\mathbf{b}_3 - \mathbf{b}_2) + (\mathbf{b}_{-3} - \mathbf{b}_{-2}) - \beta''\mathbf{D}_2\Big)\,. \tag{31}$$

Similarly, for G^4 continuity, assuming modifications to only $\mathbf{b}_4$ and $\mathbf{b}_{-4}$, Eq. 18 simplifies as

$$\mathbf{b}_{-4}^* = \mathbf{b}_{-3} + \alpha^4(\mathbf{b}_4^* - \mathbf{b}_3) + \beta'''\mathbf{D}_3 + \gamma''\mathbf{D}_2 + \delta'\mathbf{D}_1\,, \text{ where} \tag{32}$$

$$\beta''' = \frac{6\alpha^2\beta}{n-3} - 3\alpha^3 - 3\alpha^4\,, \tag{33}$$

$$\gamma'' = \frac{4\alpha\gamma + 3\beta^2}{(n-2)(n-3)} - \frac{9\alpha\beta}{n-2} + 3\alpha^2 - 3\alpha^4\,, \text{ and} \tag{34}$$

$$\delta' = \frac{\delta}{(n-1)(n-2)(n-3)} - \frac{3\gamma}{(n-1)(n-2)} + \frac{3\beta}{n-1} - \alpha - \alpha^4\,. \tag{35}$$

Following similar steps, we solve for δ' that minimizes

$$\mathcal{L} = \|\mathbf{b}_4^* - \mathbf{b}_4\|^2 + \alpha^8\|\mathbf{b}_4^* - \mathbf{b}_{-4}^+\|^2\,, \text{ where} \tag{36}$$

$$\mathbf{b}_{-4}^+ = \mathbf{b}_3 + \frac{1}{\alpha^4}\left(\mathbf{b}_{-4} - \mathbf{b}_{-3} + \beta'''\mathbf{D}_3 + \gamma''\mathbf{D}_2 + \delta'\mathbf{D}_1\right)\,, \text{ and} \tag{37}$$

$$\delta' = \arg\min_{\delta'} \left\|\mathbf{b}_4 - \mathbf{b}_{-4}^+\right\|^2\,, \text{ resulting in} \tag{38}$$

$$\delta' = \frac{\mathbf{D}_1}{d_1^2}\cdot\Big(\alpha^4(\mathbf{b}_4 - \mathbf{b}_3) - (\mathbf{b}_{-4} - \mathbf{b}_{-3}) - \beta'''\mathbf{D}_3 - \gamma''\mathbf{D}_2\Big)\,. \tag{39}$$

Higher levels of geometric continuity can be accomplished using the same procedure, but we conclude our derivations here with G^4, as higher levels of continuity are seldom needed.

4 Continuity-Enhancing Degree Elevation

Our continuity-enhancing degree elevation is a general solution to achieve C^k or G^k continuity for any $k \geq 1$. Given a piecewise Bézier curve, the continuity enhancement process described above can be applied to the intermediate control points surrounding each interpolated control point. This can be handled independently for each interpolated control point to achieve up to C^k or G^k continuity, as long as the curve's polynomial degree is at least $2k + 1$. If the curve's polynomial degree n is lower (i.e., $n < 2k + 1$), however, we can no longer adjust the continuity around each interpolated control point independently, because adjusting the derivatives on either side of a curve piece would involve modifying a shared set of intermediate control points.

Therefore, we begin with an elevation to degree n', such that $n' \geq 2k + 1$. This provides sufficient degrees of freedom to control the continuity on either side of each curve piece independently up to C^k or G^k. Then, we can modify the intermediate control points on either side of each interpolated control point to achieve C^k or G^k continuity.

Any spline of degree n can be written as an identical spline using a polynomial of a higher degree [Cohen et al. 1985]. More specifically, given a curve $\mathbf{B}(t)$ of degree n, we can write it as a polynomial of degree $n+1$, using $\mathbf{B}(t) = (1-t)\,\mathbf{B}(t)+t\,\mathbf{B}(t)$. See supplemental document Section S4 for degree elevation formulas. The degree elevation process can be repeated until the desired degree is achieved.

Note that, for our continuity enhancement, it is possible to apply degree elevation beyond the minimum polynomial degree needed. The advantage of this is that, using a higher polynomial degree, the weight of each control point decreases. As a result, our modifications for continuity enhancement result in a smaller deformation of the input curve's shape.

Our continuity-enhancing degree elevation achieves the goal of producing a curve with the desired level of continuity, at the cost of increasing the polynomial degree. While the degree elevation corresponds to negligible computational overhead, not all existing applications and file formats support polynomials of degrees higher than cubics.

5 Continuity-Enhancing Splits

Our second solution for continuity enhancement is specific to cubic curves. Given a piecewise cubic spline with C^1 continuity, we produce another piecewise cubic curve with C^2 or G^2 continuity. This is achieved by splitting each piece of the input curve into three parts, thereby tripling the number of pieces, but the cubic polynomial degree is maintained.

For each piece with control points $\mathbf{b}_0$, $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$, we begin with constructing two alternative curve pieces. The first one ensures C^2 or G^2 continuity with the previous piece at $\mathbf{b}_0$ by modifying $\mathbf{b}_2$ to $\mathbf{b}_2^*$, using the formulation described in Sec. 3. Obviously, this modified curve with the set of control points $\{\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2^*, \mathbf{b}_3\}$ only achieves C^2 or G^2 continuity with the previous piece and does not enhance the continuity with the next piece. In fact, it even breaks the existing C^1 continuity with the next piece.

Therefore, we also construct a second set of control points $\{\mathbf{b}_0, \mathbf{b}_1^*, \mathbf{b}_2, \mathbf{b}_3\}$ by modifying $\mathbf{b}_1$ to achieve C^2 or G^2 continuity with the next piece at $\mathbf{b}_3$, using the same approach. Similarly, this second set of control points is not useful for improving (or even preserving) the continuity with the previous piece.

To form our final C^2 or G^2 curve, we take the beginning part of the curve defined by the first set and the ending part of the curve defined by the second set. Then, we join them with C^2 continuity using a new set of control points, forming the middle part between these two end parts. This results in replacing each piece of the original curve with three cubic pieces, as shown in Fig. 2.

Let $\mathbf{p}_i$, $\mathbf{q}_i$, and $\mathbf{r}_i$ with $i \in \{0, 1, 2, 3\}$ represent the resulting curve control points forming the beginning, middle, and ending of the output pieces, respectively. We construct $\mathbf{p}_i$ and $\mathbf{r}_i$ starting with a Bézier subdivision formulation at normalized parameter values λ and $1-\lambda$, where $\lambda \in (0, \frac{1}{2})$ is an adjustable parameter. The resulting control points for the first part are

$$\mathbf{p}_0 = \mathbf{b}_0 \tag{40}$$

$$\mathbf{p}_1 = (1-\lambda)\,\mathbf{b}_0 + \lambda\,\mathbf{b}_1 \tag{41}$$

$$\mathbf{p}_2 = (1-\lambda)^2\,\mathbf{b}_0 + 2(1-\lambda)\lambda\,\mathbf{b}_1 + \lambda^2\,\mathbf{b}_2^* \tag{42}$$

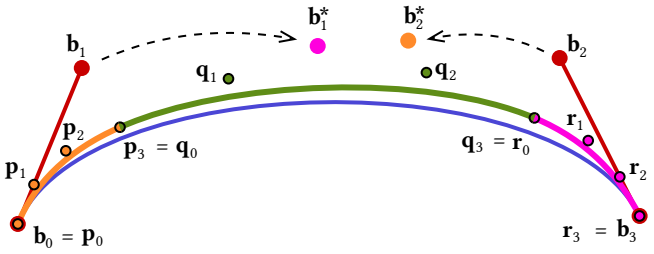


Fig. 2. *An example of continuity-enhancing splits for an input curve segment (shown in blue) that is split into 3 pieces with C^2 continuity.*

and for the last part are

$$\mathbf{r}_1 = \lambda^2\,\mathbf{b}_1^* + 2(1-\lambda)\lambda\,\mathbf{b}_2 + (1-\lambda)^2\,\mathbf{b}_3 \tag{43}$$

$$\mathbf{r}_2 = \lambda\,\mathbf{b}_2 + (1-\lambda)\,\mathbf{b}_3 \tag{44}$$

$$\mathbf{r}_3 = \mathbf{b}_3\ . \tag{45}$$

The control points $\mathbf{q}_i$ are chosen to ensure that the middle part connects the two end parts with C^2 continuity.

We get C^0 continuity by setting $\mathbf{p}_3 = \mathbf{q}_0$ and $\mathbf{r}_0 = \mathbf{q}_3$. To achieve C^1 continuity, we must satisfy

$$\frac{1}{\lambda}(\mathbf{q}_0 - \mathbf{p}_2) = \frac{1}{1-2\lambda}(\mathbf{q}_1 - \mathbf{q}_0) \tag{46}$$

$$\frac{1}{\lambda}(\mathbf{q}_3 - \mathbf{r}_1) = \frac{1}{1-2\lambda}(\mathbf{q}_2 - \mathbf{q}_3)\ . \tag{47}$$

Then, to achieve C^2 continuity, we must also satisfy

$$\frac{1}{\lambda^2}(\mathbf{p}_1 - 2\mathbf{p}_2 + \mathbf{q}_0) = \frac{1}{(1-2\lambda)^2}(\mathbf{q}_0 - 2\mathbf{q}_1 + \mathbf{q}_2) \tag{48}$$

$$\frac{1}{\lambda^2}(\mathbf{q}_3 - 2\mathbf{r}_1 + \mathbf{r}_2) = \frac{1}{(1-2\lambda)^2}(\mathbf{q}_1 - 2\mathbf{q}_2 + \mathbf{q}_3)\ . \tag{49}$$

By combining these four equations, we can find the formulations for the remaining control points as

$$\mathbf{q}_0 = \frac{2+\lambda^2-4\lambda}{1-\lambda}\mathbf{p}_2 - (1-2\lambda)\mathbf{p}_1 + \lambda\mathbf{r}_1 - \frac{\lambda(1-2\lambda)}{1-\lambda}\mathbf{r}_2 \tag{50}$$

$$\mathbf{q}_1 = \frac{1-\lambda}{\lambda}\mathbf{q}_0 - \frac{1-2\lambda}{\lambda}\mathbf{p}_2 \tag{51}$$

$$\mathbf{q}_2 = \frac{1-\lambda}{\lambda}\mathbf{q}_3 - \frac{1-2\lambda}{\lambda}\mathbf{r}_1 \tag{52}$$

$$\mathbf{q}_3 = \frac{2+\lambda^2-4\lambda}{1-\lambda}\mathbf{r}_1 - (1-2\lambda)\mathbf{r}_2 + \lambda\mathbf{p}_2 - \frac{\lambda(1-2\lambda)}{1-\lambda}\mathbf{p}_1\ . \tag{53}$$

The derivations of the formulas for $\mathbf{q}_i$ are included in the supplemental document Section S5.

Any value $\lambda \in (0, \frac{1}{2})$ produces a piecewise cubic curve with C^2 or G^2 continuity. As λ approaches zero, the shape of the middle part approaches the input piece, since each $\mathbf{q}_i$ approaches $\mathbf{b}_i$, and the two end parts vanish. Though the resulting curve has C^2 continuity for any $\lambda > 0$, when λ is close to zero, the resulting curve can include steep changes in curvature near the interpolated control points of the original curve. Conversely, when λ approaches $\frac{1}{2}$, the middle part vanishes, forming a sharp curvature change in the middle where the two end parts come together, and the resulting curve shape has maximum deviation from the input curve. The parameter value $\lambda = \frac{1}{4}$ in the middle of the range results in curves that follow the

shape of the original curve relatively closely without producing too short end parts with too steep curvature changes. Based on our observation, $\lambda = \frac{1}{4}$ also appears to produce curve shapes that are close to the ones formed by our continuity-enhancing degree elevation (Sec. 4); therefore, we use it as our default parameter.

6 Linear Pieces

One important practical problem with our continuity enhancement approach is that linear pieces of an input cubic curve are not guaranteed to maintain their linear shapes. This makes it difficult to form linear segments. Even when all Bézier control points of a piece are perfectly placed on a line, continuity enhancement with the neighboring pieces is likely to bend such originally linear pieces.

This is because the optimization problem we solve is designed to minimize the overall change in all control point positions, and it contains no constraints for maintaining linearity. Preserving linearity cannot be accomplished by an additional term in the loss function, because any violation of linearity is unacceptable when the input control points are placed on a line.

Fortunately, this can be easily rectified by introducing a linearity constraint to the optimization problem discussed in Sec. 3. Linear segments are formed when all control points of a piece are on a line. Therefore, if we restrict the modified control points $\mathbf{b}_k^*$ and $\mathbf{b}_{-k}^*$ around an interpolated control point $\mathbf{b}_0$ to be on a line defined by $\mathbf{b}_1$ and $\mathbf{b}_{-1}$, we can achieve perfectly linear pieces.

Note that for a curve with C^1 continuity, $\mathbf{b}_{-1}$, $\mathbf{b}_0$, and $\mathbf{b}_1$ are guaranteed to be on a line. To constrain the modified control points to be on the same line, we can represent them using

$$\mathbf{b}_k^* = \mathbf{b}_0 + u_k\,\hat{\mathbf{d}}\,, \qquad \text{where} \qquad \hat{\mathbf{d}} = \frac{\mathbf{b}_1 - \mathbf{b}_{-1}}{\|\mathbf{b}_1 - \mathbf{b}_{-1}\|} \tag{54}$$

with a free scalar value u_k, accounting for the remaining degree of freedom for placing $\mathbf{b}_k^*$. Plugging this into the loss function, we get

$$\mathcal{L} = \|\mathbf{b}_0 + u_k\hat{\mathbf{d}} - \mathbf{b}_k\|^2 + \alpha^{2k}\|\mathbf{b}_0 + u_k\hat{\mathbf{d}} - \mathbf{b}_{-k}^+\|^2 \tag{55}$$

We can find u_k that minimizes the loss function by taking the derivative of the loss function with respect to u_k and setting it to zero, resulting in a closed-form solution

$$u_k = \hat{\mathbf{d}} \cdot (\Delta\mathbf{b}_k + \mathbf{b}_k - \mathbf{b}_0)\,. \tag{56}$$

Using this in Eq. 54, we can find $\mathbf{b}_k^*$. Then, $\mathbf{b}_{-k}^*$ can be calculated from Eq. 7, or, more robustly, using a symmetrical formulation

$$\mathbf{b}_k^* = \mathbf{b}_0 + u_{-k}\,\hat{\mathbf{d}}, \quad \text{where} \tag{57}$$

$$u_{-k} = \hat{\mathbf{d}} \cdot \Big((-1)^k\Delta\mathbf{b}_{-k} + \mathbf{b}_{-k} - \mathbf{b}_0\Big)\,. \tag{58}$$

The derivation is included in the supplemental document Section S6.

This constrained optimization solution can be used with either of our solutions to form curves with C^2 or G^2 continuity from C^1 curves defined by Bézier handles, or with higher levels of continuity using our degree elevation process. It can easily form linear segments, as expected. However, because of the linearity constraint, continuity enhancement may result in a larger deviation from the input curve, as compared to the optimization without this constraint.

We can mitigate this extra deviation by invoking the linearity constraint only when the input C^1 curve is close to a linear shape. Let $\rho_i \in [0,1]$ be a measure of the curve's nonlinearity near an interpolated control point $\mathbf{b}_i$. We can evaluate two solutions for the intermediate control point $\mathbf{b}_{i+k}^*$ to achieve C^k or G^k continuity: one with and one without the linearity constraint. Then, we can *blend* (i.e., linearly interpolate) these two solutions using ρ_i. This approach allows gradually introducing the linearity constraint with continuous modifications to the shape of the given curve, as the user alters the input control points, and the linearity constraint is incorporated only when needed, based on the nonlinearity measure.

This nonlinearity measure can be defined in various ways, as long as it is zero when either of the two pieces joining at an interpolated control point is linear. In our implementation, we measure nonlinearity based on the *normalized total curvature*, K, of a curve piece as a unitless measure of the curve's total bending, defined as

$$K = \int_0^1 |\kappa(t)|\,\|\mathbf{B}'(t)\|\,\mathrm{d}t = \int_0^1 \frac{\|\mathbf{B}''(t)\times\mathbf{B}'(t)\|}{\|\mathbf{B}'(t)\|^2}\mathrm{d}t\,, \tag{59}$$

where $\kappa(t)$ is curvature, $\mathbf{B}'(t) = \partial\mathbf{B}(t)/\partial t$, and $\mathbf{B}''(t) = \partial^2\mathbf{B}(t)/\partial t^2$. We approximate K using numerical quadrature with the trapezoidal rule (for symmetry) using two segments, which involves computing the curvature at $t \in \{0, \frac{1}{2}, 1\}$. Then, the nonlinearity measure is assigned as $\rho_i = \max(1, \min(K_i^+, K_i^-))$, where K_i^+ and K_i^- are the approximate normalized total curvature of the two curve pieces that join at the interpolated control point $\mathbf{b}_i$.

As intended, blending the linearity constraint using this formulation allows producing perfectly linear segments while avoiding the extra alteration of the curve due to the linearity constraint when the input pieces are not close to a linear shape.

7 Evaluation and Discussion

We implemented our continuity-enhancing degree elevation and split methods using JavaScript within a single HTML file, included as a supplemental document. All cubic curves, including our curves with C^2 and G^2 continuity, are rendered natively using SVG format. Since SVG does not support higher-order polynomials, they are rendered as highly tessellated polylines, as commonly handled. Unless otherwise specified, all results are generated using quintic (i.e., $n = 5$) curves with our degree elevation examples, $\lambda = \frac{1}{4}$ with our continuity-enhancing splits, and no linearity constraints.

7.1 Shape Matching

As can be seen in Fig. 1, our curves with G^2 continuity can closely match the shape of a given cubic C^1 curve, while eliminating all curvature discontinuities that appear at every interpolated control point. Fig. 3 shows all three curves overlaid on top of each other to highlight the minor differences.

Such differences can be reduced using polynomials of a higher degree or using a smaller λ parameter for our splits. This is demonstrated in Fig. 4. Naturally, as the curve shape gets closer to the input curve, the modified region to ensure curvature continuity shrinks, resulting in relatively sharper curvature changes.

Our curves with C^2 continuity also match the input shape relatively closely in most cases, but they can also deviate significantly, as they must respect the input parameterization. This is particularly visible when the second derivative flips along the curve's tangent direction, as can be seen in Fig. 5. Again, using polynomials of a higher degree or a smaller λ parameter mitigates this issue.

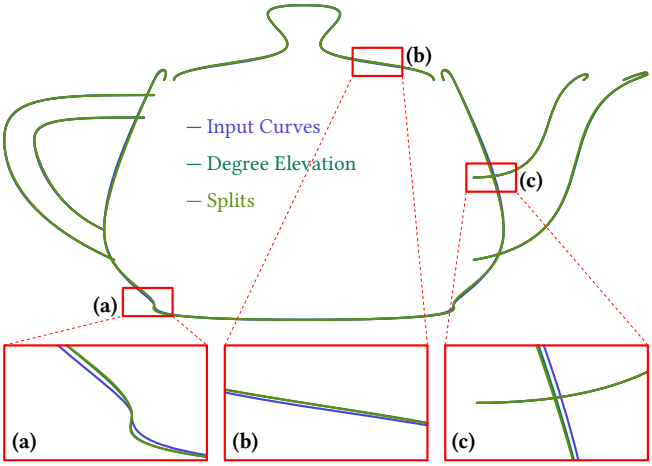


Fig. 3. *Input C^1 curves, our G^2 curves with continuity-enhancing degree elevation, and our G^2 curves with continuity-enhancing splits overlayed on top of each other. Our two G^2 curves are extremely close to each other with minor differences from the input curves.*

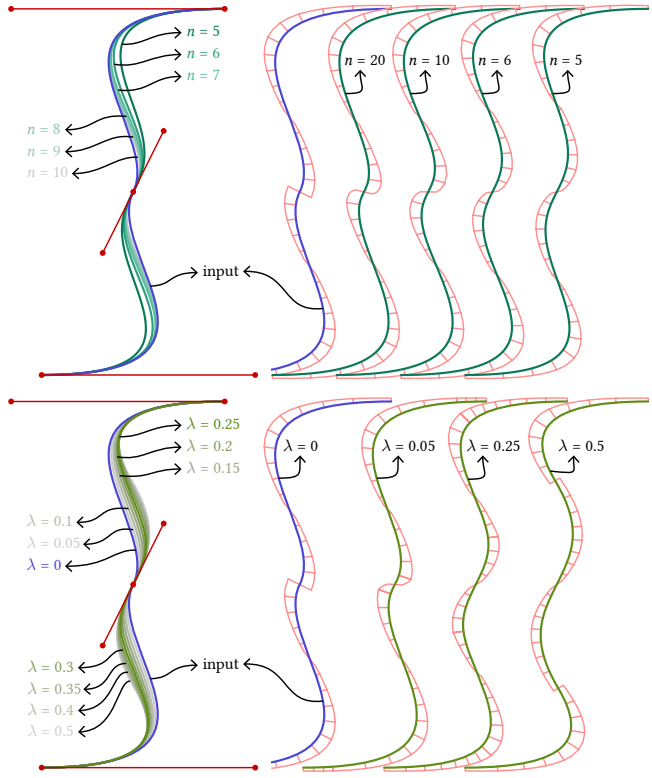


Fig. 4. *G^2 curves with our continuity-enhancing (top) degree elevation of different polynomial degrees and (bottom) splits using different λ parameters, generated from an input curve with a curvature flip on its center. With degree elevation, higher polynomial degrees produce closer shapes to the input with sharper curvature changes near the center. With splits, smaller λ values produce closer shapes to the input with sharper curvature changes near the center. With $\lambda = 0$ the output curve is identical to the input. Using larger λ values increases the shape difference and moves the sharp curvature change towards the center of each piece. Using $\lambda = 0$ or $\lambda = 0.5$, the output curve is no longer G^2.*

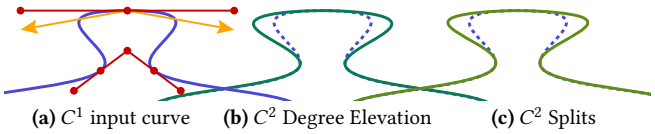


Fig. 5. *The shapes of our C^2 curves can significantly deviate from the input curves when the second derivative flips along the tangent direction, such as the tip of the Teapot's lid. The orange arrows show the second derivative of the curve on either side of the tip, flipping along the tangent direction, while the curvature on both sides is identical.*

Nonetheless, our goal is not to match the shape of an input curve exactly. After all, continuity enhancement *must* modify the shape of the curve. Yet, how closely our curves follow the shape of the input curve is one strong evidence for how easy it is to control our C^2 and G^2 curves using the same familiar Bézier handles.

7.2 Curve Properties

The resulting C^2 and G^2 curves we generate are Bézier curves with either a higher degree or more pieces. Therefore, they have all the curve properties of Bézier curves. However, if we consider the resulting splines as curves that are solely specified by the input control points, we lose some desirable properties of C^1 Bézier curves.

First and foremost, the resulting curves do not have an exact subdivision property. Subdividing a piece of one of our C^2 or G^2 curves can slightly change the shape of the curve, if the new curve is generated from scratch using the new interpolated control points and their tangents. This is because the resulting Bézier handles at the subdivision point do not have sufficient information to perfectly reproduce the C^2 or G^2 curve before the subdivision.

Also, curves after continuity enhancement are not guaranteed to remain within the convex hull of the input control points. Though they typically remain within the convex hull, continuity enhancement can break the convex hull property. Yet, if desired, it is possible to constrain the optimization to always find modified control points within the convex hull, thereby ensuring that the convex hull property is maintained (see supplemental document Section S7).

Most importantly, continuity enhancement extends the local support of each control point, as demonstrated in Fig. 6. Bézier handles that control a C^1 curve only control the shape of the two immediately connected curve pieces. Our continuity enhancement extends this to the next piece on either side, so each control point impacts the shapes of four curve pieces. While the shape changes of these two additional pieces are much smaller in comparison, they are nonetheless impacted. Note that additional levels of continuity enhancement with higher degree polynomials do not extend the locality any further.

7.3 Linearity Constraint

The importance of our linearity constraint is demonstrated in Fig. 7. Without linearity constraint (Fig. 7a), even when the control points are placed perfectly linearly, and so the input curve is linear, our continuity enhancement can bend the curve. Using our linearity constraint resolves this (Fig. 7b) at the cost of introducing a larger shape deformation to the neighboring pieces.

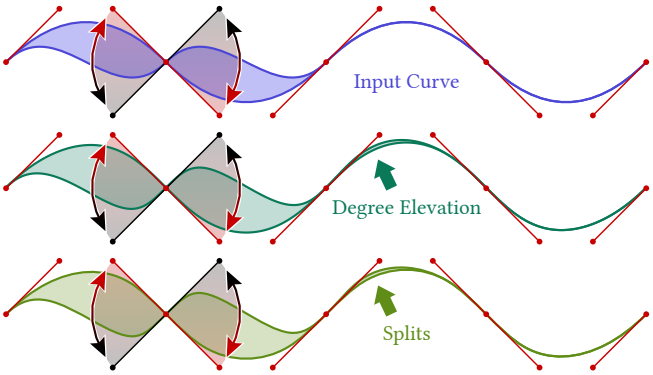


Fig. 6. *Our continuity-enhancing degree elevation and splits extend the local support of the control points. In this example, as the highlighted Bézier handles move up and down, only two immediately connected pieces of the input C^1 curve deform within the shaded area, and its other pieces remain unchanged. Our G^2 curves, however, also slightly deform the neighboring pieces, as highlighted with arrows.*

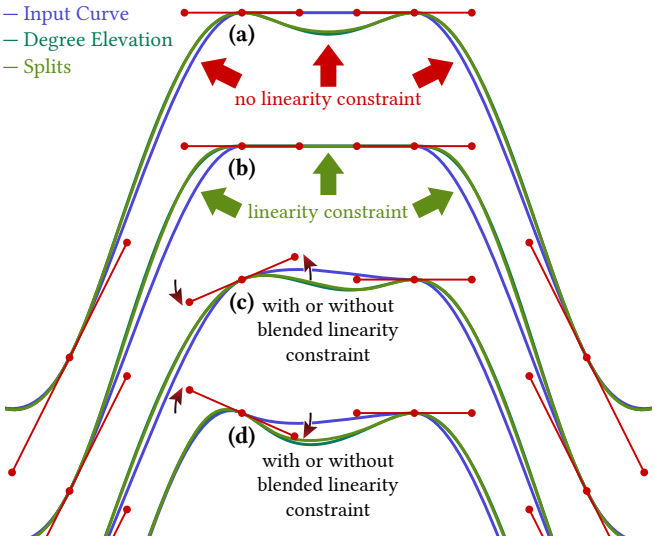


Fig. 7. *Linearity constraint:* **(a)** *without using a linearity constraint, the output G^2 curves can bend even when the input curve piece is perfectly linear with linearly placed control points.* **(b)** *Using the linearity constraint ensures that linear shapes are preserved, at the cost of deforming the shape further for the neighboring pieces.* **(c-d)** *If the linearity constraint is blended using our nonlinearity measure, when the normalized total curvature is large enough, we get the same result with or without the blended linearity constraint.*

Our blending formulation with our nonlinearity measure gradually removes the impact of the linearity constraint, and eliminates it entirely when the input curve is sufficiently nonlinear (Fig. 7c-d). However, notice that the middle part of the curve bends down in both Fig. 7c and Fig. 7d, which can be unintuitive. In comparison, the input C^1 curve (blue) bends up in Fig. 7c but down in Fig. 7d, more faithfully following the new orientation of the Bézier handle.

7.4 Higher Levels of Continuity

Our continuity-enhancing degree elevation allows achieving higher levels of continuity than C^2 and G^2. This is demonstrated in Fig. 8. Notice that G^2 and C^2 examples fix the curvature discontinuities of the C^1 input, but the curvature change remains discontinuous. This is fixed with G^3 and C^3. G^4 and C^4 examples provide an extra level of continuity, but they also increase the degrees of freedom in the curve representation, using polynomials of degree 9. In this example, we see reasonable shapes up to C^{12}, beyond which the curvature becomes unstable, and at C^{15}, while the target continuity is achieved, the shape drastically deviates from the given input.

In general, such instabilities can appear much earlier. Though we can theoretically achieve an arbitrary level of continuity, as the user alters the Bézier handles, unexpected shape deformations may arise beyond G^4 and C^4, which is why they are not practically useful.

7.5 Interpolating Curves

Our continuity-enhancing degree elevation can also be used for constructing interpolating curves. We can begin with a polyline that interpolates a given set of control points with C^0 continuity. Then, an interpolating curve can be formed by a degree elevation followed by continuity enhancement.

Degree elevation for a line piece that interpolates $\mathbf{c}_0$ and $\mathbf{c}_1$ results in a cubic Bézier curve with linearly placed control points

$$\mathbf{b}_0 = \mathbf{c}_0\,, \quad \mathbf{b}_1 = \frac{2}{3}\mathbf{c}_0 + \frac{1}{3}\mathbf{c}_1\,, \quad \mathbf{b}_2 = \frac{1}{3}\mathbf{c}_0 + \frac{2}{3}\mathbf{c}_1\,, \quad \mathbf{b}_3 = \mathbf{c}_1\,. \tag{60}$$

Before continuity enhancement, we must parameterize the curve, since we cannot infer it from the given C^1 continuity. Common methods for this are uniform, chordal, and centripetal parameterizations, similar to Catmull-Rom splines [Yuksel et al. 2011]. Then, we can apply continuity enhancement to C^1, as explained in Sec. 3.2.

The resulting curves are identical to C^1 Catmull-Rom splines with uniform parameterization (except for how the curve ends are handled). Using other parameterizations, the curve shape differs from Catmull-Rom splines, but does not appear to have any desirable properties over Catmull-Rom splines with the same parametrization.

For higher levels of continuity, we can begin with any C^1 interpolating spline. For example, C^1 Catmull-Rom curves with centripetal parameterization have unique properties, such as guaranteed self-intersection-free interpolation [Yuksel et al. 2011], which make them highly desirable. C^2 Catmull-Rom curves, however, can easily form cusps and self-intersections, even with centripetal parameterization. On the other hand, if we apply either of our solutions to get C^2 or G^2 curves from C^1 centripetal Catmull-Rom splines, we maintain these desirable properties: the tangents still point towards the next control point, and no cusps/self-intersections can be formed.

Another advantage over Catmull-Rom curves with higher levels of continuity is the local support of each control point. As noted earlier, the control points of our curves only impact the shapes of 4 curve pieces (see Fig. 6). This is true for any level of continuity. With the Catmull-Rom formulation, however, locality grows with each level of continuity (e.g., C^3 control points shape 6 pieces).

Fig. 9 shows a comparison involving various interpolating curves.

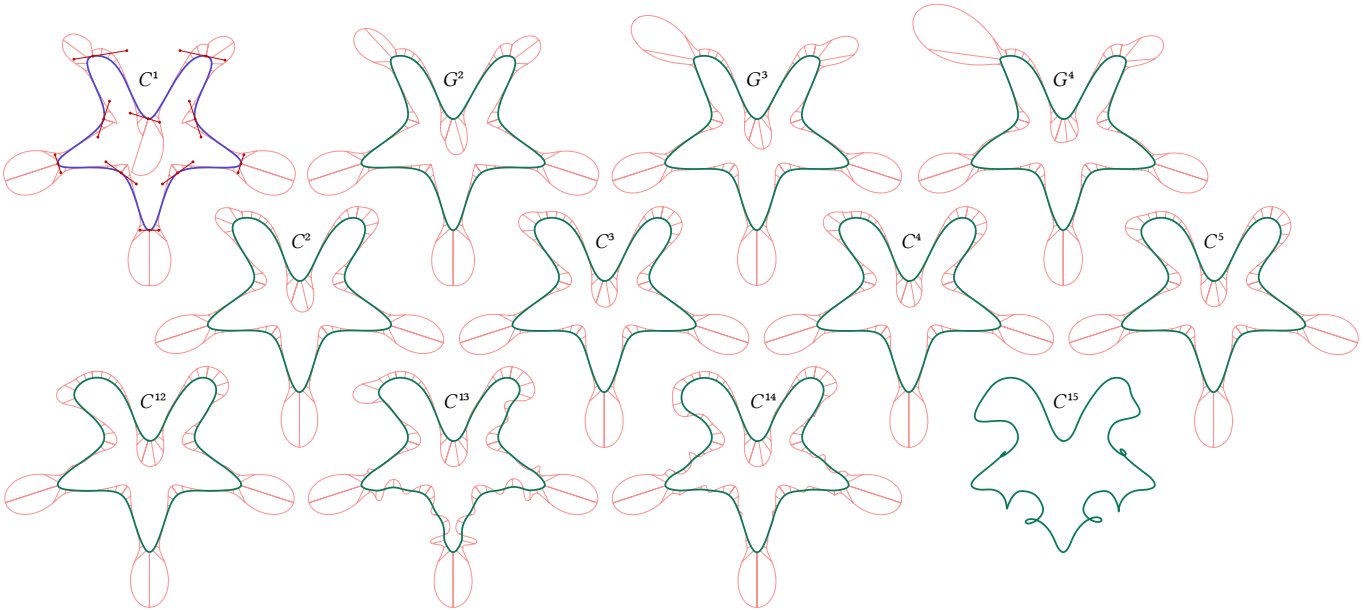
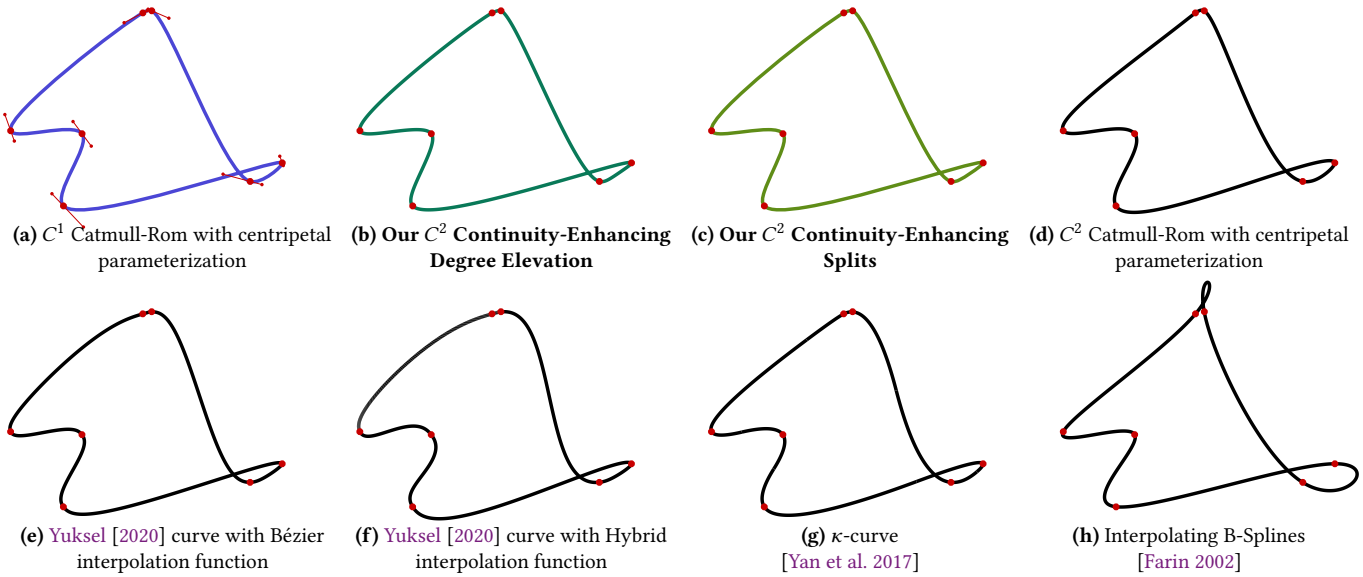


Fig. 8. *Continuity-enhancing degree elevation to improve a C^1 input model with curvature discontinuities. The polynomial degrees are elevated to $n = 2k + 1$ to achieve G^k or C^k continuity. Notice that G^2 and C^2 models fix the curvature discontinuities, and G^3 and C^3 models eliminate the discontinuities in curvature change, particularly visible near the centers of the stars. This model produces stable results till C^{12}. Results with C^{13} and C^{14} have more frequent curvature variations, though the input shape is mostly maintained. Our result with C^{15} fails to preserve the input shape.*



(a) C^1 Catmull-Rom with centripetal parameterization

(b) Our C^2 Continuity-Enhancing Degree Elevation

(c) Our C^2 Continuity-Enhancing Splits

(d) C^2 Catmull-Rom with centripetal parameterization

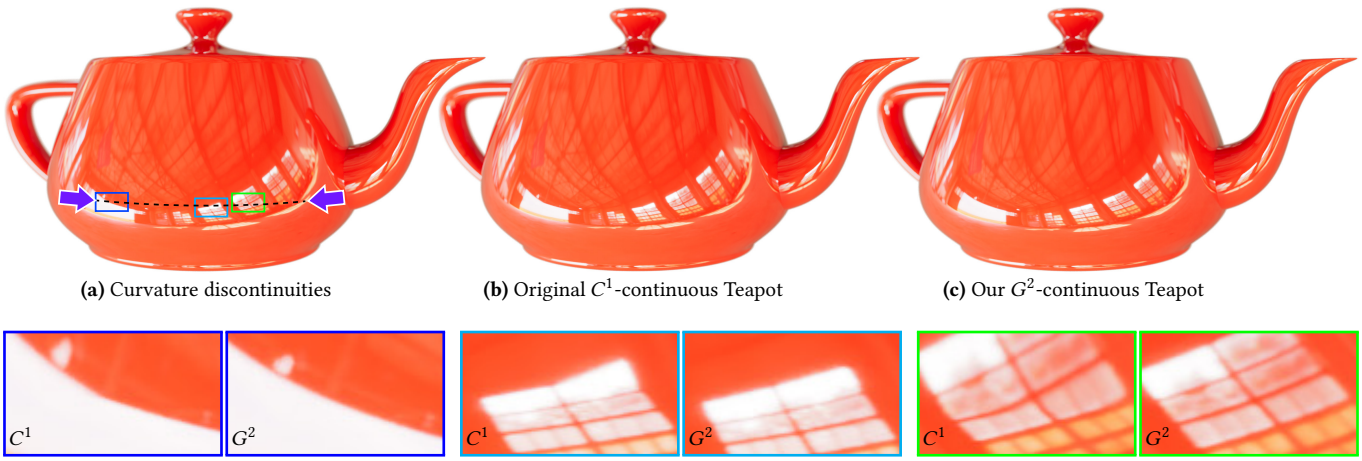
(e) Yuksel [2020] curve with Bézier interpolation function

(f) Yuksel [2020] curve with Hybrid interpolation function

(g) κ-curve [Yan et al. 2017]

(h) Interpolating B-Splines [Farin 2002]

Fig. 9. *Interpolating curves generated from the same control points using different formulations:* **(a)** *C^1 Catmull-Rom curve that automatically generates the Bézier handles, which is also used as the input to our methods,* **(b)** *our C^2 continuity-enhancing degree elevation, closely following the shape of the input curve,* **(c)** *our C^2 continuity-enhancing splits with almost identical shape,* **(d)** *C^2 Catmull-Rom curves with centripetal parameterization, which can form self-intersecting interpolation,* **(e-f)** *Yuksel [2020] curves with Bézier and Hybrid interpolation functions, which are not polynomial curves,* **(g)** *κ-curve [Yan et al. 2017], which is mostly G^2, limited to 2D, and requires solving a global optimization problem, and* **(h)** *interpolating B-Splines [Farin 2002], which are prone to self-intersecting interpolation while also requiring a global optimization problem.*



(a) Curvature discontinuities **(b)** Original C^1-continuous Teapot **(c)** Our G^2-continuous Teapot

Fig. 10. *Utah Teapot model is formed by cubic Bézier patches connected with C^1 continuity. We can turn it into a G^2-continuous surface with minimal changes to its shape. Teapots here are rendered as polygonal meshes. Our G^2-continuous Teapot is generated using our continuity-enhancing degree elevation. Our continuity-enhancing splits produce visually identical results. The insets highlight curvature discontinuities fixed by G^2.*

7.6 Higher Dimensions and Surfaces

Our continuity enhancement formulations are not limited to 2D curves. Therefore, they can be used in any dimension.

In addition, we can also use them for enhancing the continuity of surfaces defined by Bézier patches. Similar to cubic curves with Bézier handles, achieving curvature continuity with Bézier patches is practically infeasible. That is why surface formulations based on B-splines (such as NURBS) are favored over them. With our solutions, however, we can build surfaces with curvature continuity starting from a C^1 surface.

A good example of this is the Utah Teapot model in Fig. 10. The curvature discontinuities of the original model are visible in reflections (see the supplemental video), which are fixed by the G^2-continuous Teapot produced by our continuity-enhancing degree elevation.

8 Conclusion and Future Work

We have presented two solutions for enhancing the continuity of given C^1 splines. Our methods allow, for the first time to our knowledge, the ability to model C^2 or G^2 curves using Bézier handles, which is by far the most common interface for curve editing. Our continuity-enhancing splits can achieve this without even increasing the polynomial degree. We have also demonstrated that, if higher-degree polynomials are permitted, our continuity-enhancing degree elevation remains effective for utilizing the same interface to achieve even higher levels of continuity. In addition, we have discussed how our methods can be used for building interpolating curves with desirable properties, curves in higher dimensions, and even surfaces.

Future work can improve upon the formulations we have described. In particular, alternative nonlinearity measures or new ways of enforcing linear pieces can be developed. Moreover, shape is far more important than parametric measures for some applications, so alternative loss functions that are agnostic to parameterization can be integrated into our methods. Furthermore, for achieving higher levels of continuity, new optimizations that alter multiple control points at once might outperform our greedy solution of enhancing the continuity one level at a time.

References

Alexandre Binninger and Olga Sorkine-Hornung. 2022. Smooth Interpolating Curves with Local Control and Monotone Alternating Curvature. *Computer Graphics Forum* 41, 5 (2022), 25–38. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14600 doi:10.1111/cgf.14600

Edwin Catmull and Raphael Rom. 1974. A Class of Local Interpolating Splines. In *Computer Aided Geometric Design*. Academic Press, 317–326. doi:10.1016/B978-0-12-079050-0.50020-5

Elaine Cohen, Tom Lyche, and Larry L. Schumaker. 1985. Algorithms for degree-raising of splines. *ACM Trans. Graph.* 4, 3 (July 1985), 171–181. doi:10.1145/282957.282962

Tony DeRose and Brian Barsky. 1985. An intuitive approach to geometric continuity for parametric curves and surfaces. In *Proceedings of the Graphics Interface '85* (Montréal, Québec, Canada) *(GI 1985)*. Canadian Information Processing Society, Toronto, Ontario, Canada, 343–351. doi:10.20380/GI1985.49

Gerald Farin. 2002. *Curves and Surfaces for CAGD: A Practical Guide* (5th ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Josef Hoschek and Dieter Lasser. 1993. *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, Ltd.

Tsung-Wei Hu and Ming-Jun Lai. 2025. Construction of interpolating space curves with arbitrary degree of geometric continuity. *Sampling Theory, Signal Processing, and Data Analysis* 23, 2 (06 Aug 2025), 18. doi:10.1007/s43670-025-00108-1

Bowen Jiang and Renjie Chen. 2025. G2 Interpolating Spline with Local Maximum Curvature. *ACM Trans. Graph.* 44, 6, Article 229 (Dec. 2025), 13 pages. doi:10.1145/3763316

Dan Wang, R.U. Gobithaasan, Tadatoshi Sekine, Shin Usuki, and Kenjiro T. Miura. 2020. Interpolation of point sequences with extremum of curvature by log-aesthetic curves with G2 continuity. *Computer-Aided Design and Applications* 18, 2 (2020), 399 – 410. doi:10.14733/cadaps.2021.399-410

Zhihao Wang, Juan Cao, Tuan Guan, Zhonggui Chen, and Yongjie Jessica Zhang. 2024. pκ-Curves: Interpolatory curves with curvature approximating a parabola. *Computer Aided Geometric Design* 111 (2024), 102330. doi:10.1016/j.cagd.2024.102330

Zhipei Yan, Stephen Schiller, and Scott Schaefer. 2019. Circle reproduction with interpolatory curves at local maximal curvature points. *Computer Aided Geometric Design* 72 (2019), 98–110. doi:10.1016/j.cagd.2019.06.002

Zhipei Yan, Stephen Schiller, Gregg Wilensky, Nathan Carr, and Scott Schaefer. 2017. κ-curves: Interpolation at Local Maximum Curvature. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)* 36, 4, Article 129 (2017), 7 pages. doi:10.1145/3072959.3073692

Cem Yuksel. 2020. A Class of C2 Interpolating Splines. *ACM Transactions on Graphics* 39, 5, Article 160 (jul 2020), 14 pages. doi:10.1145/3400301

Cem Yuksel, Scott Schaefer, and John Keyser. 2011. Parameterization and Applications of Catmull-Rom Curves. *Computer Aided Design* 43, 7 (2011), 747–755. doi:10.1016/j.cad.2010.08.008