146: Curvature-sensitive splines and design with basic curves

Kęstutis Karčiauskas [a] Jörg Peters [b]
[a] *Vilnius University*
[b] *University of Florida*

Abstract

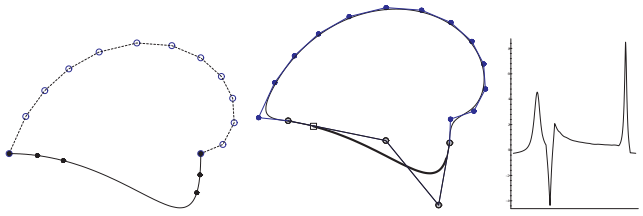
This paper presents a non-uniform cubic C^2 spline framework that unifies three scenarios for incorporating data from basic curves, such as spirals and conics. In the first scenario, no parameterization of the basic curves is available, only well-spaced samples; in the second, a parameterization is available but cannot be used directly in a spline framework; only in the third scenario can pieces of basic curves be exactly re-represented and included into the spline. In all three cases the output is a cubic C^2 spline suitable for standard CAD downstream processing. A key challenge in constructing the spline is to cope with transitions in the presence of strongly differing curvatures. Here we introduce a new form of curvature-sensitive averaging.

Key words: Curvature-sensitive splines, design with basic shapes, profile curves

1. Motivation

To satisfy functional or aesthetic criteria, designed objects often have to exactly match prescribed data, such as a series of positions, or incorporating a *basic shape*, a template curve, typically of pleasing curvature, such as a circle arc. A major challenge is to smoothly transition between these sites, in particularly when, by relative placement or intrinsically, the data imply strongly differing curvatures. A typical artifact, affecting overall quality, are oscillations. Even slight oscillations in the curvature are visible when a planar curve is used as the profile curve of a surface of revolution or as the starting point of extrusion (see Fig. 18).

In this paper, we consider three complementary scenarios of incorporating data from basic curve segments into a standard cubic C^2 spline. In the first scenario no parameterization of the underlying basic shape is available, only sufficiently dense samples (see Section 7 for a discussion of 'sufficient density'). For this first scenario, we propose a new tool, the csr-spline. The csr-spline uses curvature-sensitive non-linear averaging to trade-off data fitting and distribution of curvature. In a second scenario, the parameterization of a basic curve is available but cannot be used directly in the spline modeling framework. In this scenario, the C^2 spline closely approximates the basic shape. For example, a designer can closely mimic a transcendental curve piece, such as an arc of an Archimedian spiral. In the third scenario, the basic shape can be explicitly (re-)represented as a cubic spline. That is, basic curve segments, such as circle arcs, other conic sections or general rational cubics, are directly incorporated into the C^2 spline. This approach refines and expands the 'design with basic shapes' of [KP11a].



(a) spiral+cubic input (b) New unifying C^2 spline (c) curvature

Fig. 1. Unifying C^2 spline. (a) Samples of an Archimedian spiral (top) and a polynomial cubic (bottom) (b) Control points of the unifying spline: the cubic (thick) is exactly replicated. (c) Curvature plot. Here and in all following such figures, a □ square on the curve (b) marks the beginning of the counterclockwise curve traversal for which the curvature is plotted. The curvature of the spiral-part is between the two large spikes.

All three solutions to incorporating data from basic curves can be integrated into a single cubic C^2 spline. This unification is demonstrated in Fig. 1 where blue control points (and certain scalars β) are those of a csr-spline and the gray control points (and scalars) come from a segmented polynomial (free-form) cubic.

Overview. Section 2 briefly reviews and compares some classic and some recent curve interpolation techniques. Section 3 introduces two tools: how to represent non-uniform cubic C^2

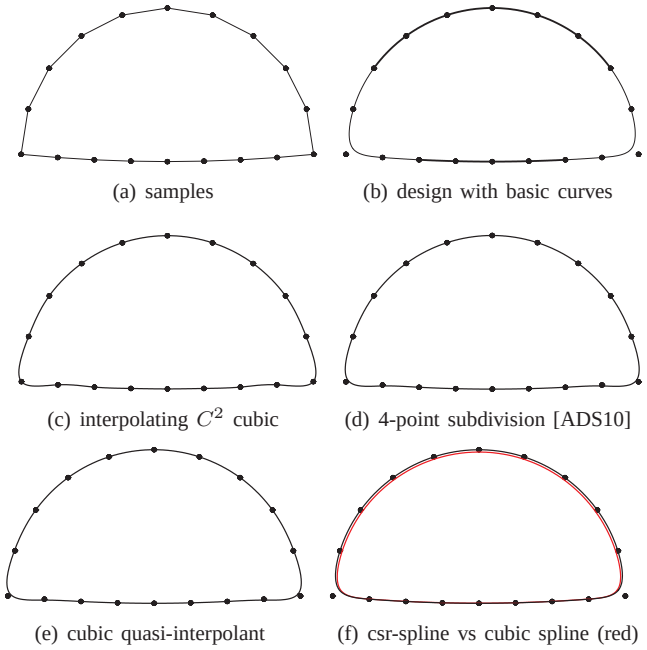
Email address:
jorg@cise.ufl.edu,Kestutis.Karciauskas@mif.vu.lt
(Jörg Peters).

splines equivalently as piecemeal G^2 splines and reversing this relation, and how to determine the u-parameters corresponding to designer markings on a parametric basic curve. The three sections, Section 4, Section 5 and Section 6, explain the C^2 constructions for each of the three scenarios: cubic splines derived from sampled data, cubic splines derived from non-cubic parameterizations, and cubic splines locally replicating basic shapes. The curvature and convexity of these curves are put under the microscope by displaying surfaces of revolution for which the curves serve as profiles. Section 7 discusses the need for sufficient density of samples to make interpolation meaningful.

2. Interpolation of samples and curves

Interpolating curve constructions are often tested by their ability to reproduce, from samples, such basic shapes as conics. But they are rarely judged by their ability to smoothly join segments of the shapes into an ensemble. Yet this is a major challenge for design as illustrated by Fig. 2. Especially where curvatures differ strongly, it is difficult to avoid overshoot and curvature oscillation. The uniform point samples in Fig. 2a imply a strong change of curvature where the two basic shape samples meet up. Here we measure 'discrete curvature', i.e., the curvature at a point obtained by fitting a local quadratic and evaluating it at the middle interpolation point. The classic C^2 cubic spline interpolant in Fig. 2c visibly oscillates near this transition. Even moderately dense sampling (cf. example Fig. 19d) or removing corner points from interpolation (Fig. 19g) fails to remove the oscillations. (However, classic cubic interpolation serves well for reconstruction where the data are much denser and noisy.) By relaxing the interpolation requirements, the subdivision scheme [ADS10], shown in Fig. 2d, does slightly better, but still exhibits oscillations. The underlying problem is the scheme's lack of response to the very different curvatures implied by the point samples. The standard cubic quasi-interpolant of Fig. 2e fares no better. Only the new curvature-sensitive, relaxed cubic C^2 csr-spline of Section 4 does well (Fig. 2f). The csr-spline is obtained by local curvature averaging; and while this spline generally does not interpolate the input points, it approximates the locations well where the discrete curvature changes moderately. Curvature averaging complements design with basic shapes of [KP11a] (see Fig. 2b). In *design with basic shapes* the designer chooses and places pieces of basic shapes. Instead of sampling the input curves, the designer only marks points on the basic shapes. An algorithm, with knowledge of the underlying dictionary of basic shapes, returns a control polygon. The designer can then freely modify this polygon and an algorithm converts it back into a smooth spline that preserves the input shapes where the data have not been modified.

We focus on smoothness and changes of curvature in the transitions. Input data need not be connected, can be continuous, or even C^1. In all cases, the unified construction returns non-uniform C^2 spline curves with CAD downstream compatibility. This output and the broadly applicable construction are in contrast, and complement, specialized constructions such as [WM90,WM05,GMW09] or more compute-intensive approach like L^1 optimization [Lav04], or other energy minimization.



(a) samples (b) design with basic curves (c) interpolating C^2 cubic (d) 4-point subdivision [ADS10] (e) cubic quasi-interpolant (f) csr-spline vs cubic spline (red)

Fig. 2. Semi-circle and almost flat parabola are sampled uniformly (a). (b) Design reproducing basic shapes (thick). (c) The C^2 cubic interpolating spline oscillates. (d) Relaxed C^2 4-point subdivision [ADS10], with the recommended $\alpha = 0.634$, oscillates. (e) The cubic $(-1, 8, -1)/6$ quasi-interpolant oscillates akin to (d). (f) The new cubic C^2 csr-splines of Section 4 (black) matches data more closely than the C^2 cubic approximating spline (red: the C^2 cubic approximating spline interprets the data points as B-spline control points).

3. Tools: cubic G^2 splines and choice of parameters

In this section, we briefly review two basic techniques used for all three constructions, corresponding to the scenarios.

3.1. *Non-uniform cubic C^2 splines as piecemeal G^2 splines*

When working with cubic curves, we freely switch between their representation as C^2 spline with non-uniform, strictly increasing sequence of single knots $\{t_i\}$ and their representation as individual polynomial pieces, each in their own unit domain $[0..1]$. Let f and g be two consecutive pieces of a C^2 spline, each parameterized over $[0..1]$. Then f and g join with special *geometric continuity* G^2 at their common point $g(0) = f(1)$, namely

$$g'(0) = \beta f'(1)\ ,\ g''(0) = \beta^2 f''(1), \quad \beta_i := \frac{t_{i+1} - t_i}{t_i - t_{i-1}}. \quad (1)$$

The ith piece of a C^2 cubic spline with control point sequence $\{\mathbf{c}_i\}$ and knots $\{t_i\}$ can be expressed in *Bernstein-Bézier form*

$$\sum_{j=0}^{3} \mathbf{b}_j^i B_j, \qquad B_j(u) := \binom{3}{j}(1-u)^{3-j}u^j,$$

with coefficients

$$\begin{aligned}
\mathbf{b}_1^i &:= (1-e_i)\mathbf{c}_i + e_i\mathbf{c}_{i+1},\\
\mathbf{b}_2^i &:= \tilde{e}_i\mathbf{c}_i + (1-\tilde{e}_i)\mathbf{c}_{i+1},\\
\mathbf{b}_0^i = \mathbf{b}_3^{i-1} &:= (1-x_i)\mathbf{b}_2^{i-1} + x_i\mathbf{b}_1^i,\\
e_i &:= \frac{1}{1+\beta_i+\beta_i\beta_{i+1}},\ \tilde{e}_i := \frac{\beta_i\beta_{i+1}}{1+\beta_i+\beta_i\beta_{i+1}},\\
x_i &:= \frac{1}{1+\beta_i}.
\end{aligned} \tag{2}$$

Note that x_i, e_i and $\tilde{e}_i$ are in $[0\,..\,1]$ and that $1-\tilde{e}_i > e_i$. We can reverse the relation to obtain the B-spline control points $\mathbf{c}_i$ from the Bernstein-Bézier coefficients $\mathbf{b}_j^i$ via

$$-\beta_i\mathbf{b}_1^{i-1} + (1+\beta_i)\mathbf{b}_2^{i-1} = \mathbf{c}_i = \Big(1+\frac{1}{\beta_i}\Big)\mathbf{b}_1^i - \frac{1}{\beta_i}\mathbf{b}_2^i. \tag{3}$$

That is $\mathbf{c}_i$ is the intersection of two lines through consecutive pairs of 'interior' coefficients $\mathbf{b}_1^j$, $\mathbf{b}_2^j$.

3.2. *Determining parameters corresponding to samples*

In Section 5 and Section 6, the designer marks points on parametric basic curves in lieu of manipulating control structures. A practical implementation for marking, to avoid introducing noise, is to use a slider interface to move the points on the curve, rather than asking the designer to free-hand place the points. An algorithm then has to convert the marked points to their pre-images, i.e., their parameter values. The density of such marked points is typically much lower than samples generated by a scanning device for reverse engineering, and, as explained, is not noisy.

Since the circle and its affine transformations to ellipses are ubiquitous and can be dealt with explicitly, they deserve special attention when determining parameters. While the unit circle's trigonometric parameterization $C(\alpha) := (\cos\alpha, \sin\alpha)$ is convenient for mathematical treatment, inclusion into a free-form modeling environment requires a rational representation. Rather than the C^2 homogeneous representation of degree 6 [BP97] or the G^2 representation of degree 3 [KP11b], we will use the classic symmetric rational quadratic presentation $\mathbf{f}(u)$, $u \in [0,1]$ (see e.g. [Far02, Chap 12]). Given samples $\mathbf{q}_i$ on a circle, we can find parameters u_i such that $\mathbf{f}(u_i) = \mathbf{q}_i = C(\alpha_i)$ either by conversion of the angles α_i or by numerical least-squares fit as explained below.

For a circle arc with opening angle α, we assume without loss of generality that $\mathbf{f}(0) := (1,0)$, $\mathbf{f}(1) := C(\alpha)$. Then $\mathbf{q}_i = C(\alpha_i)$, $0 < \alpha_i < \alpha$ is solved by

$$u_i := \begin{cases} \dfrac{\sin(\alpha_i+\frac{\alpha}{2}) - \sin\frac{\alpha}{2} - \sin\alpha_i}{2(\cos\frac{\alpha}{2}-1)(\sin\frac{\alpha}{2}+\sin\alpha_i)} & \text{if } \alpha_i < \pi,\\ \dfrac{1+\cos(\alpha_i+\frac{\alpha}{2}) - \cos\frac{\alpha}{2} - \cos\alpha_i}{2(\cos\frac{\alpha}{2}-1)(\cos\frac{\alpha}{2}+\cos\alpha_i)}, & \text{else,}\end{cases} \tag{4}$$

where the case-distinction prevents division by zero (symbolic calculation shows equality of the top and bottom expressions).

Alternatively, the numerical solution of

$$\frac{d}{du}\|\mathbf{f}(u) - \mathbf{q}_i\|^2 = 0, \quad u \in [0,1], \tag{5}$$

is well-defined. This numerical least-squares fit can also be applied to any other basic curve's parameterization $\mathbf{f}(u)$, $u \in [0,1]$ and even when the points $\mathbf{q}_i$ do not exactly lie on $\mathbf{f}$.

4. Curvature-sensitive non-linear averaging

In this section, we construct our cubic C^2 spline via its G^2 form with parameters β_i as in (1) from reasonably dense samples $\{\mathbf{q}_i\}$ (see Section 7 for a discussion of density) assumed to be from underlying basic curve segments. By default, we associate chordal-spaced parameters $\{t_i\}$ with the samples. In particular, we do not assume prior knowledge of the parameterization of the segments. The spline will depend on curvature and will stay very close to the samples $\{\mathbf{q}_i\}$, but need not interpolate. Since we relax the interpolation requirements, we call the construction a *curvature-sensitive relaxed spline*, short csr-spline. Its shape and its curvature are near-indistinguishable from those of the cubic splines constructed in Section 5.

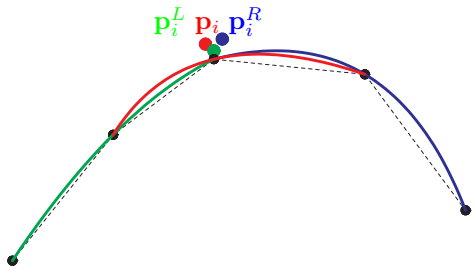


Fig. 3. Curvature-sensitive averaging of data from three quadratic interpolants: $\mathbf{f}_{i-1}$ (green) $\mathbf{f}_i$ (red) and $\mathbf{f}_{i+1}$ (blue).

We now specify the construction of the csr-spline. Let $\mathbf{f}_i$ be a quadratic polynomial interpolating the sample points $\mathbf{q}_{i-1}$ at t_{i-1}, $\mathbf{q}_i$ at t_i and $\mathbf{q}_{i+1}$ at t_{i+1}. The control point $\mathbf{c}_i$ corresponding to a sample point $\mathbf{q}_i$ is defined by averaging the data from the three interpolants $\mathbf{f}_{i-1}$, $\mathbf{f}_i$, $\mathbf{f}_{i+1}$ as illustrated in Fig. 3. Representing each $\mathbf{f}_i$ in BB-form of degree 3 and applying (3), we obtain their coefficients corresponding to t_i as

$$\begin{aligned}
\mathbf{p}_i &:= -\frac{\beta_i^2}{3(\beta_i+1)}\mathbf{q}_{i-1} + \frac{(\beta_i+1)^2}{3\beta_i}\mathbf{q}_i - \frac{1}{3\beta_i(\beta_i+1)}\mathbf{q}_{i+1}\\
\mathbf{p}_i^L &:= a_{-2}\mathbf{q}_{i-2} + a_{-1}\mathbf{q}_{i-1} + (1-a_{-2}-a_{-1})\mathbf{q}_i,\\
\mathbf{p}_i^R &:= a_2\mathbf{q}_{i+2} + a_1\mathbf{q}_{i+1} + (1-a_2-a_1)\mathbf{q}_i,\\
a_{-2} &:= \frac{-\beta_{i-1}^2}{3(\beta_{i-1}+1)}, a_{-1} := \frac{\beta_{i-1}}{3},\\
a_2 &:= \frac{-1}{3\beta_{i+1}(\beta_{i+1}+1)}, a_1 := \frac{1}{3\beta_{i+1}}.
\end{aligned}$$

Then the coefficient of the interpolant is

$$\begin{aligned}\mathbf{c}_i :=& (1-\nu_i)\big(\mu_i \mathbf{p}_i^L + (1-\mu_i)\mathbf{p}_i\big) \\ &+ \nu_i\big((1-\mu_{i+1})\mathbf{p}_i^R + \mu_{i+1}\mathbf{p}_i\big).\end{aligned} \tag{6}$$

$$\mu_i := \begin{cases} \dfrac{\kappa_i}{\kappa_{i-1}+\kappa_i} & \text{if } \kappa_{i-1}+\kappa_i > \epsilon; \\ \dfrac{1}{2} & else, \end{cases}$$

$$\nu_i := \begin{cases} \dfrac{\kappa_{i-1}}{\kappa_{i-1}+\kappa_{i+1}} & \text{if } \kappa_{i-1}+\kappa_{i+1} > \epsilon; \\ \dfrac{1}{2} & else, \end{cases}$$

where κ_i is the absolute value of the curvature of $\mathbf{f}_i$ at $\mathbf{q}_i$ and $\epsilon > 0$ is a fixed tolerance by which we avoid numerical instability when the denominator is close to or equals zero. Alternatively, we can add ϵ to the curvatures κ_j, e.g., set $\nu_i := \frac{\kappa_{i-1}+\epsilon}{\kappa_{i-1}+\kappa_{i+1}+2\epsilon}$, and can then drop the case distinction. When all $\beta_j = 1$, then the formula for $\mathbf{p}_i$ becomes that for cubic quasi-interpolation, $(-1, 8, -1)/6$, which, by itself, without curvature averaging, results in oscillations (cf. Fig. 2e).

The rationale underlying (6) is to negotiate high curvature differences by weighing the data from one side with the curvature of the other. This gives more weight to data exhibiting the lower curvature and thereby counteracts the tendency to overshoot. In particular, this helps to transition from highly-curved to almost linear segments (see Fig. 2). When the samples are so fine that the local quadratics yield a sufficiently good approximation of both position and curvature of an underlying smooth convex basic curve, we have consistently observed that the csr-spline is locally convex.

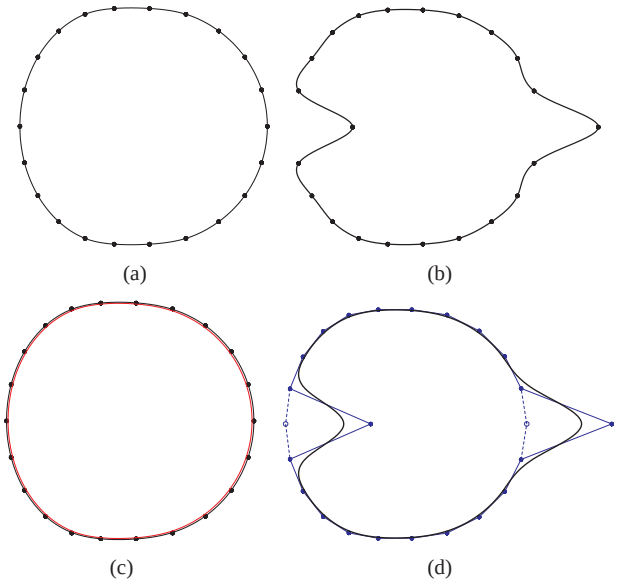


Fig. 4. Joining a half-circle to a half-ellipse. (a,b) classic cubic C^2 interpolation; (c,d) curvature-sensitive csr-spline.

Fig. 4 complements Fig. 2 in contrasting the csr-spline with the classic cubic interpolating spline. While the classic spline (a) copes well with the input, moving the interpolation points (b) generates unexpected oscillations in position and curvature. The csr-spline (c) does well both on the input and retains the overall quality under subsequent modification.

Data-dependent corrections as in (6) can be applied to subdivision schemes [MDL05]. But there the corrections complicate the analysis and increase cost since they need to be applied at each step. For splines, the data-dependent correction is performed once.

5. Approximating spirals and other curves

In this section, we aim to closely represent spirals and other parametric curve segments within the same cubic C^2 spline framework. Given the parameterization $\mathbf{f}(u)$, $u \in [0,1]$, of the ith segment of the basic curve defined by $[u_i, u_{i+1}]$, we approximate it by quintic polynomial pieces with Bézier coefficients $\mathbf{b}_j^i$, $j = 0, \ldots, 5$:

$$\begin{aligned} \mathbf{b}_0^i &:= \mathbf{f}(u_i), \\ \mathbf{b}_1^i &:= \mathbf{b}_0^i + \frac{1}{5}\mathbf{f}'(u_i), \\ \mathbf{b}_2^i &:= \mathbf{b}_0^i + \frac{2}{5}\mathbf{f}'(u_i) + \frac{1}{20}\mathbf{f}''(u_i), \\ \mathbf{b}_3^i &:= \mathbf{b}_5^i - \frac{2}{5}\mathbf{f}'(u_{i+1}) + \frac{1}{20}\mathbf{f}''(u_{i+1}), \\ \mathbf{b}_4^i &:= \mathbf{b}_5^i - \frac{1}{5}\mathbf{f}'(u_{i+1}), \\ \mathbf{b}_5^i &:= \mathbf{f}(u_{i+1}). \end{aligned} \tag{7}$$

This quintic approximation is often very close to $\mathbf{f}(u)$ not only in distance but also in curvature. For example, a segmentation of a unit-sized Archimedian spiral, Fig. 7, deviates in position by less than 10^{-8} and in curvature by about 10^{-5}. In many applications these quintics can therefore stand in for an original transcendental curve.

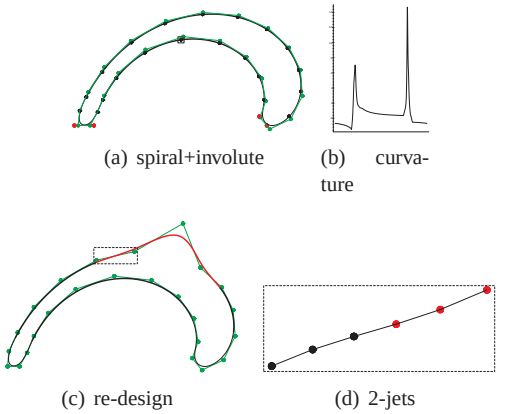
We replace the quintic polynomial in turn by our C^2 cubic splines (in G^2 form), converting their 2-jets $\{\mathbf{b}_0^i, \mathbf{b}_1^i, \mathbf{b}_2^i\}$ to 2-jets $\{\bar{\mathbf{b}}_0^i, \bar{\mathbf{b}}_1^i, \bar{\mathbf{b}}_2^i\}$ in degree 3 form:

$$\begin{aligned} \bar{\mathbf{b}}_0^i &:= \mathbf{b}_0^i, \;\; \bar{\mathbf{b}}_1^i := -\frac{2}{3}\mathbf{b}_0^i + \frac{5}{3}\mathbf{b}_1^i, \\ \bar{\mathbf{b}}_2^i &:= \mathbf{b}_0^i - \frac{10}{3}\mathbf{b}_1^i + \frac{10}{3}\mathbf{b}_2^i. \end{aligned} \tag{8}$$

We then compute the spline control points $\mathbf{c}_i$ by formula (3). At either end, the control polygon of each basic segment is augmented by the curve end-points with geometric continuity scalars $\beta := 1$. Where end-points of two basic curve segments are in close proximity they can be averaged into one point (see Fig. 6a).

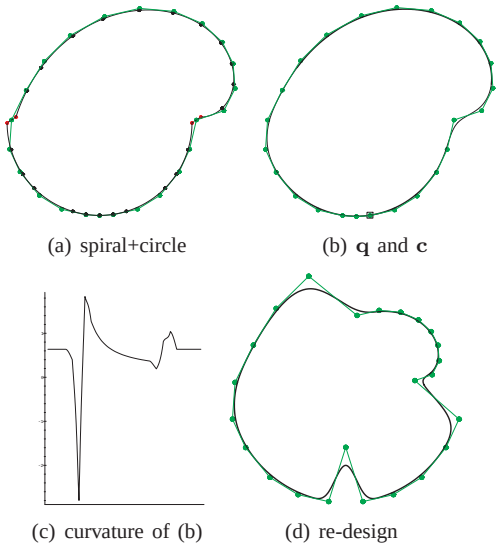
While a rational spline can not reproduce the original transcendental curve, all our measurements show our spline to have just twice the error of the quintic approximation (7). Applying the construction to the homogeneous presentation of a rational cubic curve yields the same, exact result as the construction of Section 6.

To assess our construction, we consider three challenges: joining an involute to a spiral (Fig. 5), joining a circle to a spiral (Fig. 6) and joining a line segment to a spiral (Fig. 7). In all three experiments, the transcendental basic curve segmen, restricted to $[u_2, u_{m-2}]$ ($u_0 = 0$ and $u_m = 1$) is closely approximated by the cubic spline (8). (While our cubic spline



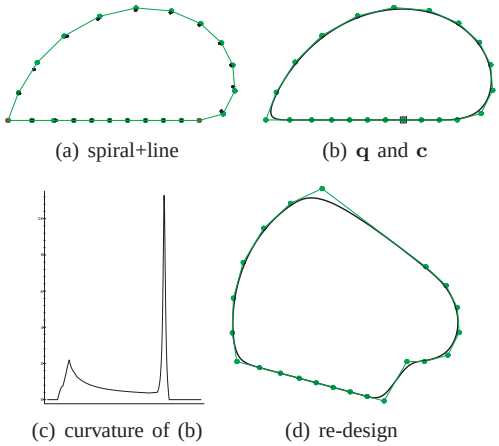
(a) spiral+involute (b) curvature (c) re-design (d) 2-jets

Fig. 5. Joining an Archimedian spiral and a circle involute. (a) Marked points $\mathbf{q}$ (black), spline and control polygon with coefficients $\mathbf{c}$ (green) and (b) its curvature; (c) a design variant: The enlargement (d) shows preservation of the (black) 2-jets at the border to the re-design (red).



(a) spiral+circle (b) $\mathbf{q}$ and $\mathbf{c}$ (c) curvature of (b) (d) re-design

Fig. 6. Joining an Archimedian spiral and a circle. End-points of basic curves that are in close proximity are averaged into one point. (a) Marked points and spline control polygon; (b) spline and (c) its curvature; (d) a design variant.



(a) spiral+line (b) $\mathbf{q}$ and $\mathbf{c}$ (c) curvature of (b) (d) re-design

Fig. 7. Joining an Archimedian spiral and a line segment. (a) Marked points and spline control polygon; (b) spline and (c) its curvature; (d) a design variant.

approximation will often suffice, we can alternatively, on the interval $[u_2, u_{m-2}]$, replace the cubics by the quintics (7) or by the original parameterization. The resulting hybrid curve remains curvature continuous.)

For subsequent modeling, we may switch between manipulating the approximations of the basic curves – via the cubic control polygon as illustrated in Fig. 5c – and sampling the underlying basic curves. Moving one control point $\mathbf{c}_i$ affects four segments (red) but not the outermost (quintic) 2-jets of (7) shown as black disks in Fig. 5d.

6. Exact representation of rational cubic curves as G^2 splines

In this third scenario, we exactly convert any homogeneous parameterization of degree at most 3 of a rational curve $\mathbf{f}(u)$, $u \in [0,1]$, to our C^2 spline representation. Rational curves of degree 3 include for example conics and circles. The designer starts by marking $m-1$ points $\mathbf{q}_i$ on $\mathbf{f}$ inducing a partition $\{0 < u_i < 1\}$, $i = 1, \ldots, m-1$, determined by (4) or (5). The sequence is completed by setting $u_0 := 0$, $u_m := 1$. The u_i split $\mathbf{f}$ into segments joining with geometric continuity (1) and $\beta_i := \frac{u_{i+1}-u_i}{u_i-u_{i-1}}$. Using formula (3), we obtain, coordinate-wise in homogeneous space, the de Boor control points $\mathbf{c}_i$ of the corresponding rational cubic C^2 spline. Specifically, let $\mathbf{b}_j$, $j = 0, \ldots, 3$ be the Bézier coefficients of the homogeneous representation $\mathbf{f}^h$. Then the spline coefficient $\mathbf{c}_i$ corresponding to the junction of the intervals $[u_{i-1}, u_i]$, $[u_i, u_{i+1}]$ is given by

$$\begin{aligned} \mathbf{c}_i &:= \sum_{j=0}^{3} d_j \mathbf{b}_j, \qquad (9)\\ d_0 &:= (1-u_{i-1})(1-u_i)(1-u_{i+1}),\\ d_2 &:= u_{i-1}u_i + u_{i-1}u_{i+1} + u_i u_{i+1} - 3u_{i-1}u_i u_{i+1},\\ d_3 &:= u_{i-1}u_i u_{i+1}, \quad d_1 := 1 - d_0 - d_2 - d_3. \end{aligned}$$

Conversely, given the coefficients $\mathbf{c}_i$, (2) reproduces the original cubic restricted to $[u_2, u_{m-2}]$. To extend the construction to

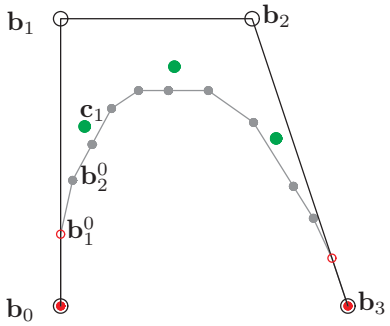
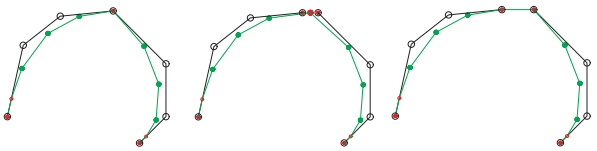


Fig. 8. Basic cubic in refined, hence locally editable, B-spline form extending to both ends.

an endpoint of an open curve, e.g. the intervals $[0, u_1]$ and $[u_1, u_2]$ (or, symetrically, at the other end), we include as B-spline control points, also the Bézier points (cf. Fig. 8)

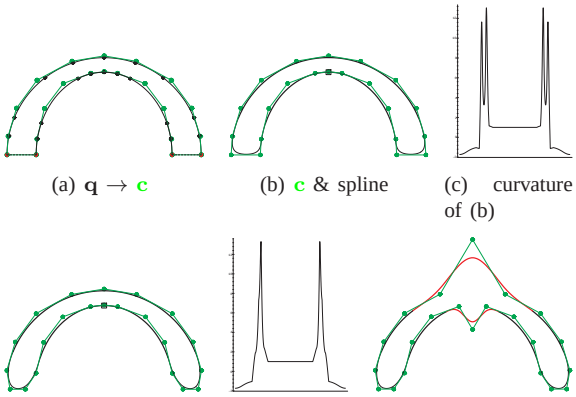
$$\mathbf{c}_0 := \mathbf{b}_0^0, \quad \mathbf{c}_1 := \mathbf{b}_1^0, \text{ and, by (3), } \mathbf{b}_2^0 := \frac{\mathbf{c}_1 + \beta_1 \mathbf{b}_1^0}{1+\beta_1}. \quad (10)$$

The B-spline representation is used to join adjacent basic segments into one composite curvature continuous curve. As illustrated in Fig. 9, there are three cases for the endpoints of two adjacent basic segments: they (a) agree, (b) are sufficiently close to be averaged into one point, (c) are apart. In cases (a), (b) the point becomes part of the B-spline polygon, in (c) both points and the segment in-between are part of the B-spline polygon. In all three cases, we choose $\beta := 1$ for the new control point(s).



(a) joint point (b) averaged point (c) new edge

Fig. 9. Merging basic shape control polygons.



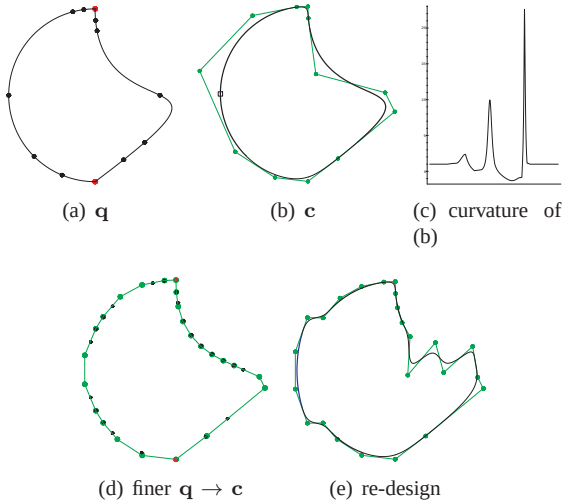
(a) $\mathbf{q} \to \mathbf{c}$ (b) $\mathbf{c}$ & spline (c) curvature of (b)

(d) rounded ends (e) curvature of (d) (f) re-design

Fig. 10. Smoothly interpolating a semi-circle and a semi-ellipse in a design with basic curves. (a) Sample points $\mathbf{q}_i$ (black) chosen by the designer and corresponding cubic B-spline control polygon with coefficients $\mathbf{c}$ (larger, green) augmented by end points (red). (b) Cubic C^2 spline and (c) its curvature. (d) Taking advantage of the spline representation, the control polygon is rounded yielding a new spline with curvature (e). (f) A design variant.

In Fig. 10 two non-intersecting basic curves, a semi-circle and a semi-ellipse, are joined. The designer marks points $\mathbf{q}_i$ on the basic curves according to design intentions, e.g. more densely in regions that are to be modified as in Fig. 10f. Then (9) yields the B-spline control points $\mathbf{c}_i$. With the (red) endpoints retained to connect the basic curves, this yields the C^2 spline shown in Fig. 10b. By converting the basic curves to splines, the designer can now manipulate the ensemble with built-in C^2 continuity, by moving the control points as in Fig. 10f.

In Fig. 11 a semi-circle is joined to a polynomial cubic curve. The cubic C^2 spline is derived as in the previous example. While standard knot insertion allows refinement, denser marking yields a more natural interface to locally increase the degrees of freedom. We can increase the number of samples as in Fig. 11d to be able to create Fig. 11e, where the control points have been scaled with respect to the circle center (blue section).

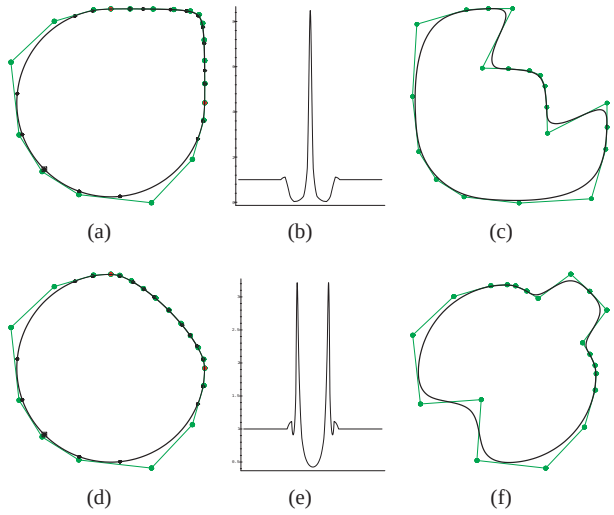


(a) $\mathbf{q}$ (b) $\mathbf{c}$ (c) curvature of (b)

(d) finer $\mathbf{q} \to \mathbf{c}$ (e) re-design

Fig. 11. A semi-circle is joined to a polynomial cubic curve. Degrees of freedom added by denser sampling. (a) Sample points $\mathbf{q}$ on basic curves, (b) cubic C^2 spline and (c) its curvature. (d) Locally refined sampling of the basic curves and the derived control polygon enabling variant (e).

Conversely a circle arc with a large opening angle can be modeled using just enough samples to prevent negative weights.



(a) (b) (c)

(d) (e) (f)

Fig. 12. Circle + hyperbola, circle + ellipse. (a, d): marked points, derived control polygon and cubic C^2 spline; (b, e): curvatures; (c, f): re-design.

In Fig. 12 one quarter of a circle is replaced by a hyperbolic arc in Fig. 12a, respectively by an elliptic arc in Fig. 12d. Denser marking is used in lieu of knot insertion. While controlling shape with control polygons representing large opening angles can be difficult, especially in the presence of very differently-sized control edges, marking unequally-spaced points poses no challenge to a designer. The non-negative curvature plots Fig. 12b,e hint at Proposition 1, retention of convexity.

A planar rational cubic curve with positive weights and a convex control polygon is convex. We can guarantee positive weights, say for circle arcs, by sufficiently splitting the arc; and we can refine such control polygons to assure convexity. The next proposition extends this property to transitions between convex curve segments (see Fig. 9).

Proposition 1 (convexity preservation) *If the control polygons of two basic curve pieces, joined at the endpoints or joined by an edge, form a convex polygon then the curve generated by* (9) *is locally convex.*
Lemma 1 does not require strong convexity – adjacent segments can lie on a line.

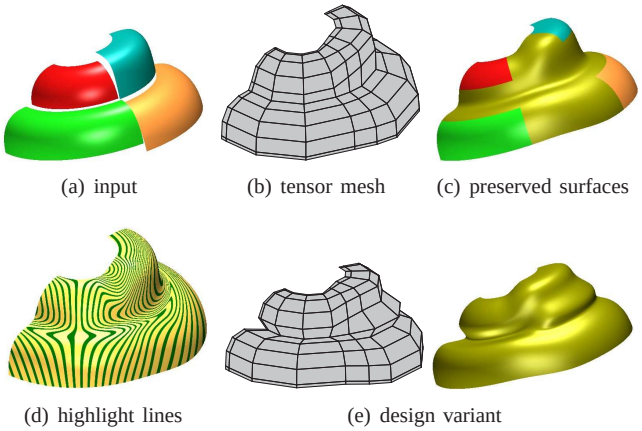
Proof By assumption the joint breakpoint sequence $\{u_i\}$ is strictly increasing and therefore $\beta_i > 0$. By construction (2) and its reverse (3), local convexity of the Bézier control polygon and the B-spline control polygon are equivalent: they correspond to switching between the vertex and the half-space formulation of convexity, where each half-space is defined by a line $\overline{\mathbf{b}_1^i \mathbf{b}_2^i}$. Therefore convexity of the joint polygon implies convex Bézier control polygons. Since the weights of the Bézier form can be assumed to be positive, the curve is locally convex. The proof of case (b), averaged end-points, follows from (c), added edge, since the polgon obtained by merging two points of a convex polygon is also convex. |||

Tensor-product surfaces. Tensoring the construction of the B-spline polygon yields a bi-cubic C^2 spline surface in B-spline form. Away from any boundary, the interior Bézier control points $\mathbf{b}_{rs}^{ij}$ of $\sum_{r=0}^{3}\sum_{s=0}^{3}\mathbf{b}_{rs}^{ij}B_rB_s$ yield by formula (3) the bi-cubic spline control points $\mathbf{c}_{ij}$,

$$\begin{aligned}\mathbf{c}_{ij} := \dot\beta_i\ddot\beta_j\mathbf{b}_{11}^{i-1,j-1} - \dot\beta_i(1+\ddot\beta_j)\mathbf{b}_{12}^{i-1,j-1} \\ -(1+\dot\beta_i)\ddot\beta_j\mathbf{b}_{21}^{i-1,j-1} + (1+\dot\beta_i)(1+\ddot\beta_j)\mathbf{b}_{22}^{i-1,j-1},\end{aligned} \tag{11}$$

where $\dot\beta_i$ and $\ddot\beta_j$ are the geometric continuity scalars in either variable.

For a concrete example, see Fig. 13. Applying the tensored construction rules to the basic shapes of Fig. 13a, stitched together by averaging close B-spline control points, yields the control net Fig. 13b and the surface Fig. 13c that the highlight lines of Fig. 13d declare curvature continuous. The control net can be freely modified via its control net to yield shape variants.

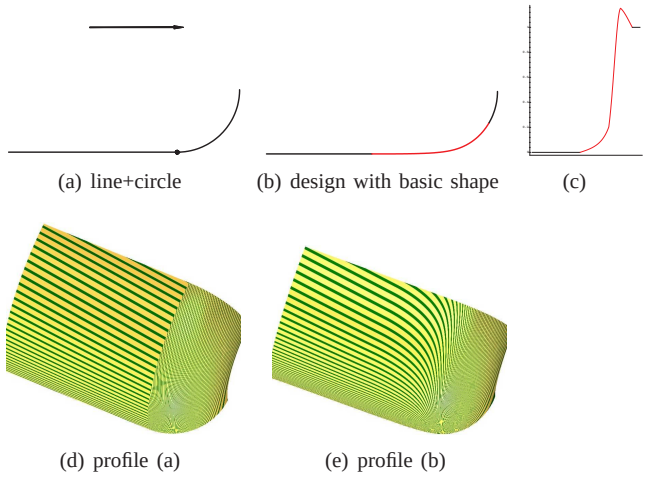


(a) input (b) tensor mesh (c) preserved surfaces

(d) highlight lines (e) design variant

Fig. 13. Tensored construction. (a) Input basic surfaces of revolution with profile circle (red, torus), parabola (green), ellipse (gold), hyperbola (cyan). (b) control net; (c) preserved parts of basic surfaces, blended together; (d) highlight shading of (c); (e,f) design variant.

Testing as profile curves We test the spline constructions as profile curves of surfaces of revolution. In this role, even small flaws in the transition between the basic shapes become visible. We consider the profiles in Fig. 14. Rotating the line segment attached to the quarter-circle about the indicated axis (arrow) yields a surface of revolution that exhibits kinks in the highlight lines Fig. 14d. We could exploit specific reasoning to improve the situation, but we are looking for a general method.



(a) line+circle (b) design with basic shape (c)

(d) profile (a) (e) profile (b)

Fig. 14. Curvature change. (a) Input: a quarter-circle touching a line segment and marked points; (b) cubic C^2 spline of Section 6 and (c) its curvature. (d),(e) surfaces of revolution from (a),(b).

Our construction yields the profile shown in Fig. 14b with its curvature displayed in Fig. 14c. Despite some overshoot, the curvature is positive and the corresponding surface of revolution in Fig. 14e has a much-improved distribution of highlight lines.

7. Density of interpolation samples

Sparse data sets can be useful to explain fundamental facts. The minimal input polygon of Fig. 15 serves nicely to illustrate the classic trade-off between interpolation and convexity for smooth curves, motivating the relaxation of interpolation constraints for csr-splines. (The underlying challenge is surely familiar to practitioners, even though it can be obscured in more substantial examples.) But configurations such as in Fig. 16a

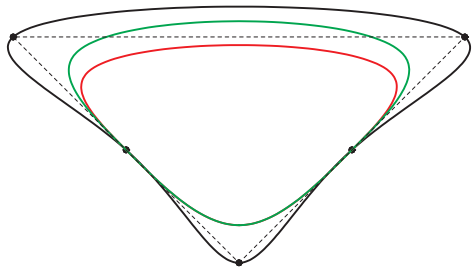
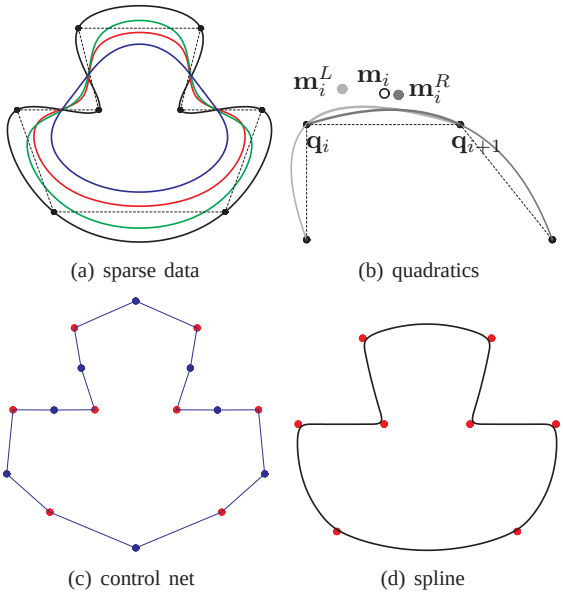


Fig. 15. Fundamental trade-off between (incompatibility of) interpolation, convexity and smoothness. Classic C^2 cubic interpolation (black); csr-spline (green); C^2 cubic spline using data as control points (red).

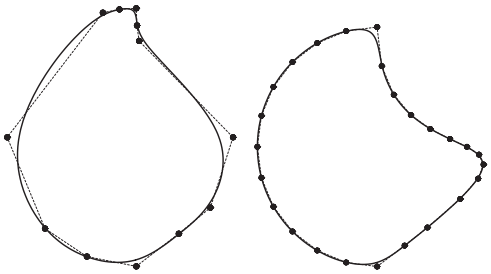
are too sparse to recognize any intent. Such data allow for many interpretations. Curves using the data as B-spline control points increasingly ignore the data as the degree increases. Classic spline interpolants overshoot. It is therefore not surprising that the csr-splines of Section 4 require well-spaced samples, such



(a) sparse data (b) quadratics

(c) control net (d) spline

Fig. 16. Sparse data set (a) of 8 black points has a wide range of interpretations: Classic C^2 cubic interpolation (black); csr-spline (green); cubic C^2 spline (red) and C^6 spline of degree 7 (blue) using data as control points. (b) Quadratic design: interpolant $\mathbf{f}_i$ defining $\mathbf{m}_i^L$ (light grey) and $\mathbf{f}_{i+1}$ defining $\mathbf{m}_i^R$ with average $\mathbf{m}_i$ (circle). (c) derived quadratic control polygon and (d) curve generated by the algorithm of Section 6.



(a) (b)

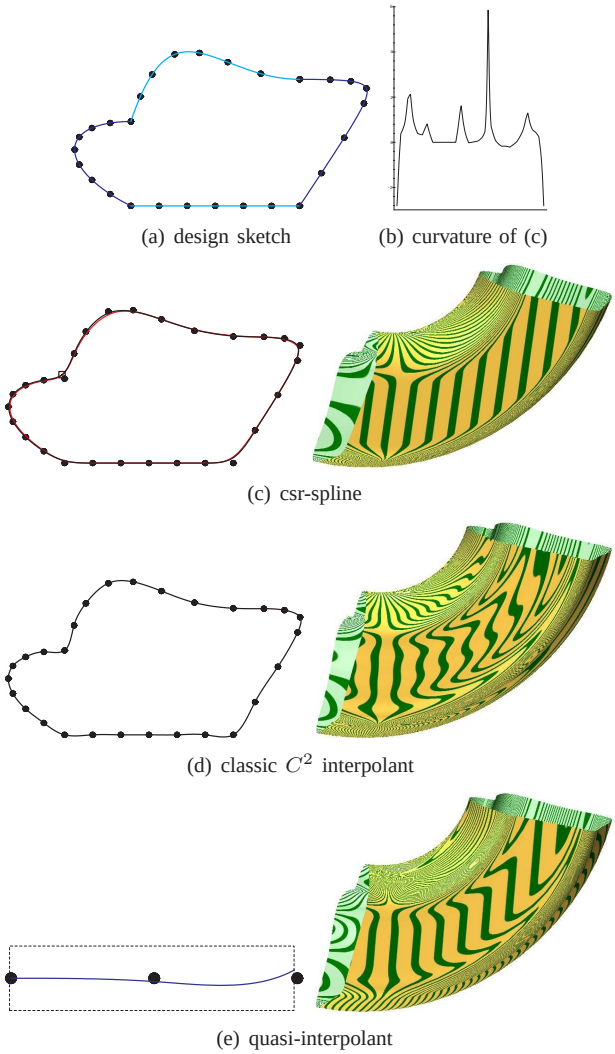
Fig. 17. The csr-spline responding to denser sampling.

as in Fig. 19f or Fig. 19i, to be able to closely replicate underlying basic curves. Note however that moderate density of mark points suffices for design and that this density is far lower than in reverse engineering, where a spline has to fit noisy data. Design with basic shapes can do with sparse mark points as illustrated in Fig. 19a-c (see also Fig. 11a). Applying the csr-spline to such coarse input does not reproduce the shape well, as shown in Fig. 17a; finer sampling does, Fig. 17b.

One way to help the casual designer generate basic shapes from sparse data is the following construction, outlined in Fig. 16b, of quadratic (basic) Bézier curves with coefficients $\{\mathbf{q}_i, \mathbf{m}_i, \mathbf{q}_{i+1}\}$ and shape handles $\mathbf{m}_i$. Here the input polygon $\mathbf{q}_i$ with parameters t_i and β_i are interpolated by quadratic polynomials $\mathbf{f}_i$. The middle Bézier coefficient of $\mathbf{f}_i$ restricted to $[t_i, t_{i+1}]$ is

$$\mathbf{m}_i^L := \frac{1}{2}\Big(-\frac{\beta_i^2}{1+\beta_i}\mathbf{q}_{i-1} + (1+\beta_i)\mathbf{q}_i + \frac{1}{1+\beta_i}\mathbf{q}_{i+1}\Big)$$

and, by symmetry, the middle coefficient of $\mathbf{f}_{i+1}$ is



(a) design sketch (b) curvature of (c)

(c) csr-spline

(d) classic C^2 interpolant

(e) quasi-interpolant

Fig. 18. (a) The designer sketches a line (cyan), a hyperbola (blue), a cubic segment (cyan) and an ellipse (blue) and marks 7 to 9 points on each. (c) The csr-spline (black; its curvature in (b)) approximates the points more closely than the C^2 cubic spline (red) that uses the points as control points. (d) The classic cubic C^2 interpolating spline oscillates. (e) The classic cubic C^2 quasi-interpolating spline oscillates (only enlargement shown).

$$\mathbf{m}_i^R := \frac{1}{2}\Big(\frac{-1}{\beta_{i+1}^2+\beta_{i+1}}\mathbf{q}_{i+2} + \frac{1+\beta_{i+1}}{\beta_{i+1}}\mathbf{q}_{i+1} + \frac{\beta_{i+1}}{1+\beta_{i+1}}\mathbf{q}_i\Big).$$

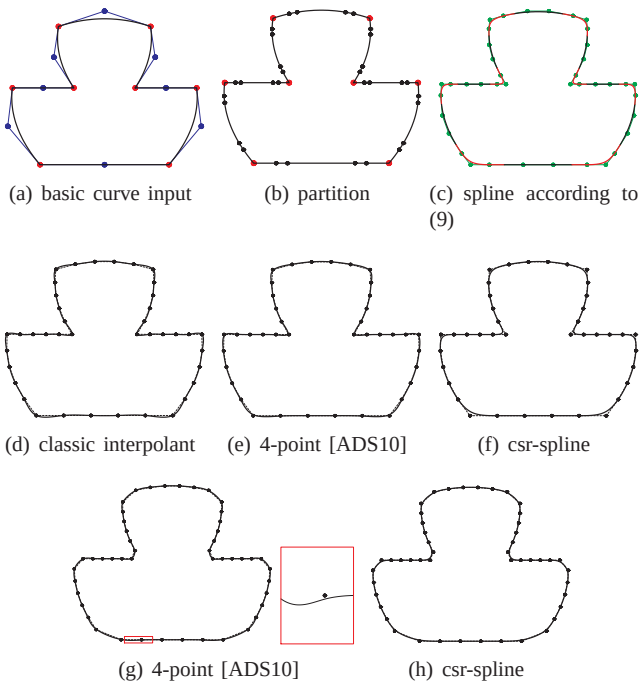
Then, with μ_i as in (6),

$$\mathbf{m}_i := \mu_{i+1}\mathbf{m}_i^L + (1-\mu_{i+1})\mathbf{m}_i^R \qquad (12)$$

typically yields a good first shape that is easy to further adjust via $\mathbf{m}_i$ (see Fig. 19a).

8. Conclusion

We unified three scenarios for incorporating data from basic curves, such as spirals and conics, creating, in each case, a non-uniform cubic C^2 spline. This spline either closely approximates the data, as points or 2-jets, or it interpolates the data by replicating basic curve segments in C^2 cubic spline form and rounding off the transitions. This combined, standard representation is suitable for standard CAD downstream processing.



(a) basic curve input (b) partition (c) spline according to (9)

(d) classic interpolant (e) 4-point [ADS10] (f) csr-spline

(g) 4-point [ADS10] (h) csr-spline

Fig. 19. Comparing scenarios: (a,b,c) Third scenario: coarse input of quadratic curve segments in Bernstein-Bézier form. (a) Only the middle control points (blue) are added to the interpolation points (red) of Fig. 16. (b) The designer marks off rounding areas with points. (c) Control points (9) and the cubic C^2 spline. Parts of the basic curve segments (black) are exactly retained. (d,e,f) low density samples. While (e) looks slightly better than (d), its oscillations are still visible without magnification. (g,h) with the corner points removed, (g) oscillates (although less than the classic interpolant to the same data), the csr-spline (h) does not.

The unified spline is intended for design. In reverse engineering, samples are typically denser and they are noisy. There the classic cubic interpolation or its variants do well, whereas they fail for design, due to oscillation. By contrast, we saw in the scenario of Section 6 that local convexity is preserved by our construction when convex control polyhedra are joined to form a convex polyhedron. This explains why the second scenario spline approximating convex transcendental curves, such as Archimedian spirals, exhibit positive curvature. Indeed, also the csr-construction, based on samples from convex shapes, consistently exhibits convexity.

Acknowledgments.

Work supported in part by NSF Grant CCF-1117695.

References

[ADS10] Ursula H. Augsdörfer, Neil A. Dodgson, and Malcolm A. Sabin. Variations on the four-point subdivision scheme. *Computer Aided Geometric Design*, 27(1):78–95, 2010.

[BP97] Claudia Bangert and Hartmut Prautzsch. Circle and sphere as rational splines. *Neural, Parallel and Scientific Computations*, (5):153–162, 1997.

[Far02] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press, San Diego, 2002.

[GMW09] Tim N. T. Goodman, Dereck S. Meek, and Desmond J. Walton. An involute spiral that matches G^2 hermite data in the plane. *Computer Aided Geometric Design*, 26(7):733–756, 2009.

[KP11a] K. Karčiauskas and J. Peters. Rational bi-cubic G^2 splines for design with basic shapes. *Computer graphics forum*, 30(5):1389–1395, 2011.

[KP11b] K. Karčiauskas and J. Peters. Rational G^2 splines. *Graphical Models*, 23(5):286–295, 2011.

[Lav04] John E. Lavery. Shape-preserving approximation of multiscale univariate data by cubic L1 spline fits. *Computer Aided Geometric Design*, 21(1):43–64, 2004.

[MDL05] M. Marinov, N. Dyn, and D. Levin. Geometrically controlled 4-point interpolatory schemes. In M.A. Sabin N.A. Dodgson, M.S. Floater, editor, *Advances in Multiresolution for Geometric Modelling*, pages 301–318. Springer-Verlag, 2005.

[WM90] D. J. Walton and D. S. Meek. Clothoidal splines. *Computers and Graphics*, 14(1):95–100, 1990.

[WM05] D. J. Walton and D. S. Meek. A controlled clothoid spline. *Computers and Graphics*, 29(3):353–363, June 2005.