# Approximating Curves and Their Offsets using Biarcs and Pythagorean Hodograph Quintics

Zbyněk Šír[1], Robert Feichtinger and Bert Jüttler

Johannes Kepler University, Institute of Applied Geometry,
Altenberger Str. 69, 4040 Linz, Austria.

## Abstract

This paper compares two techniques for the approximation of the offsets to a given planar curve. The two methods are based on approximate conversion of the planar curve into circular splines and Pythagorean hodograph (PH) splines, respectively. The circular splines are obtained using a novel variant of biarc interpolation, while the PH splines are constructed via Hermite interpolation of $C^1$ boundary data.

We analyze the approximation order of both conversion procedures. As a new result, the $C^1$ Hermite interpolation with PH quintics is shown to have approximation order 4 with respect to the original curve, and 3 with respect to its offsets. In addition, we study the resulting data volume, both for the original curve and for its offsets. It is shown that PH splines outperform the circular splines for increasing accuracy, due to the higher approximation order.

Keywords: Pythagorean Hodograph curves, G-code, Biarc, CNC machining, approximation order

## 1 Introduction

Curves with simple closed form descriptions of their parametric speed and arc–length are useful for numerically controlled (NC) machining, since they greatly facilitate the control of the tool along a curved trajectory with constant (or user–defined) speed. They are closely related to curves that provide a simple exact representation of their offset curves which allow, e.g., to take the tool radius easily into account while designing the path of a cutting tool.

Traditional techniques of NC tool path description rely mostly on the so–called G–code, which uses piecewise linear and circular curve segments. Clearly, this curve description is inherited by its offsets. Biarc interpolation is one of the main techniques for converting general curves into G-code [4, 7, 9, 12, 13, 18, 19].

The interesting class of Pythagorean Hodograph (PH) curves, see [2] and the references cited therein, may serve as an alternative description, which is capable of producing tool paths with higher smoothness. These curves, distinguished by having a polynomial arc length function and rational offset curves, provide a mathematically elegant solution to the problems occurring in NC machining. Various aspects of applications to NC machining were studied by Farouki and his co-authors [3, 16].

[1]Corresponding author. E-mail: `zbynek.sir@jku.at`, phone/fax: +43 (0) 732-2468-9178 / -9142, Homepage: `www.ag.jku.at`.

Due to the special algebraic properties of PH curves, all constructions, such as interpolation or approximation, which are linear in the case of standard spline curves become *nonlinear*. Hence, the use of local techniques seems to be most appropriate. Various constructions of planar PH curves matching given Hermite type boundary data were developed, see e.g. [5, 8, 14, 17].

The approximate conversion of a planar curve into G-code or PH form produces *simultaneously* an approximation of the *original curve* and of the *system of offsets*. As an alternative, single offset curves may be approximated directly, see, e.g., [1, 6, 15].

The first part of the paper is devoted to *biarc interpolation*. We recall basic facts and describe a novel variant, which is able to achieve "arc spline precision", i.e., it reproduces spline curves which are composed of circular arcs. In addition, we analyze the approximation order of the biarc interpolation, both with respect to the curves and to its offsets.

In the second part, which deals with quintic *Pythagorean hodograph curves*, we address the construction of PH quintic spline curves. As a new result, we show that these curves have approximation order 4 with respect to the given curve, and 3 with respect to its offsets.

Finally, we compare both techniques with respect to the data volume produced by them. It is shown that PH curves perform better than biarcs, if high accuracy is desired.

## 2 Construction of biarc splines

We review the problem of biarc interpolation of $G^1$ boundary data. We present the main results in a simple geometric way which naturally leads to a novel variant of the biarc construction. Then we apply the biarc interpolation for designing an algorithm converting an arbitrary $G^1$ continuous curve into a biarc spline. We investigate the approximation order of this method and the precision of the corresponding offset approximation.

### 2.1 Biarc interpolation

**Definition 1** *The two circular arcs $A_0$, $A_1$ are said to form a biarc interpolating given oriented $G^1$ data, represented by end points $\mathbf{P}_0$, $\mathbf{P}_1$ and unit tangent vectors $\mathbf{U}_0$, $\mathbf{U}_1$ (see Fig. 1) if and only if the two circular arcs share one common end point $\mathbf{J}$ called joint and satisfy the following properties:*

1. *The arc $A_0$ has the end points $\mathbf{P}_0$ and $\mathbf{J}$, and $\mathbf{U}_0$ is tangent*

*to $A_0$ with orientation corresponding to a parameterization of $A_0$ from $\mathbf{P}_0$ to $\mathbf{J}$.*

2. *The arc $A_1$ has the end points $\mathbf{J}$ and $\mathbf{P}_1$ and $\mathbf{U}_1$ is tangent to $A_1$ with orientation corresponding to a parameterization of $A_1$ from $\mathbf{J}$ to $\mathbf{P}_1$.*

3. *The two arcs have a common unit tangent vector at $\mathbf{J}$, with orientation corresponding to a parameterization of $A_0$ from $\mathbf{P}_0$ to $\mathbf{J}$ and of $A_1$ from $\mathbf{J}$ to $\mathbf{P}_1$.*

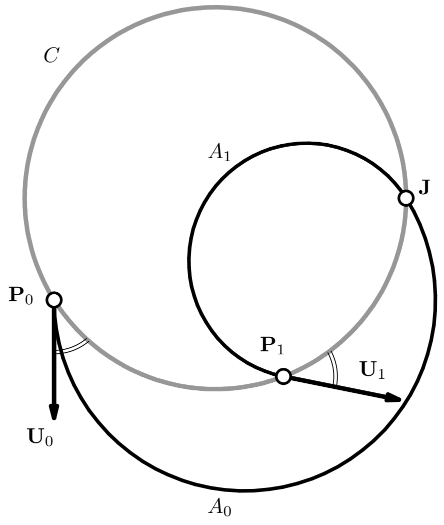

Figure 1: A biarc (black) and the joint circle (grey).

Consequently an interpolating biarc represents a $G^1$ smooth path from $(\mathbf{P}_0, \mathbf{U}_0)$ to $(\mathbf{P}_1, \mathbf{U}_1)$.

As a well-known fact [9], there is a one-dimensional parametric system of interpolating biarcs to general planar data, and the locus of all possible joints $\mathbf{J}$ is a circle passing through $\mathbf{P}_0$ and $\mathbf{P}_1$. We give a simple proof of this fact, which will yield a geometric insight that may be useful for later constructions.

**Proposition 2** *Consider the family of biarcs interpolating given oriented $G^1$ data $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{U}_0$, $\mathbf{U}_1$. Then the locus of all possible joints $\mathbf{J}$ is the circle $C$ passing through the points $\mathbf{P}_0$, $\mathbf{P}_1$ and having the same oriented angles with the vectors $\mathbf{U}_0$ and $\mathbf{U}_1$.*

**Proof.** For any data there is precisely one circle $C$ passing through the points $\mathbf{P}_0, \mathbf{P}_1$ and having the same (oriented) angles with the vectors $\mathbf{U}_0$ and $\mathbf{U}_1$, see Figure 1. It is obtained as the trajectory of the point $\mathbf{P}_0$ under the unique rotation transforming the data $\mathbf{P}_0, \mathbf{U}_0$ into the data $\mathbf{P}_1, \mathbf{U}_1$. If the vectors $\mathbf{U}_0, \mathbf{U}_1$ are parallel, then $C$ degenerates into the straight line passing through $\mathbf{P}_0, \mathbf{P}_1$.

For any point $\mathbf{J}$ in the plane there is precisely one arc $A_0$ satisfying property (1) of Def. 1 and one arc $A_1$ satisfying the second property[2]. These two arcs have a common tangent at $\mathbf{J}$ if and only if $\mathbf{J} \in C$. In fact, if $\mathbf{J} \in C$, then the angle between the circle $C$ and the arc $A_0$ at the point $\mathbf{J}$ is the same as the angle between the circle $C$ and the vector $\mathbf{U}_0$. Similarly the angle between $C$ and the arc $A_1$ at the point $\mathbf{J}$ is equal to the angle between the circle $C$ and the vector $\mathbf{U}_1$. Since the circle $C$ has the same angle with $\mathbf{U}_0$ and $\mathbf{U}_1$, the two arcs $A_0$, $A_1$ join with $G^1$ continuity at $\mathbf{J}$. The orientations coincide also, since one of the arcs $A_0$, $A_1$ will be inside and one outside the circle $C$. Suppose, on the other hand, that a $G^1$ interpolating biarc is constructed. The circle passing through the points $\mathbf{P}_0$, $\mathbf{J}$ and $\mathbf{P}_1$ must have the same angle with both circular arcs $A_0$, $A_1$ and therefore also with the vectors $\mathbf{U}_0$, $\mathbf{U}_1$. It is hence identical with the circle $C$ and $\mathbf{J} \in C$. □

[2]An exception occurs when $\mathbf{J} = \mathbf{P}_0 + t\mathbf{U}_0$ or $\mathbf{J} = \mathbf{P}_1 - t\mathbf{U}_1$ ($\mathbf{J}$ lies on the line given by the point $\mathbf{P}_i$ and the direction vector $\mathbf{U}_i$, where $i$ is 0 or 1). If $t > 0$ then the arc $A_i$ degenerates into a linear segment. If $t = 0$ then it degenerates into a point. Finally if $t < 0$ then it degenerates into a "line segment passing through infinity". In fact the natural framework for biarc interpolation is the Möbius plane which contains one point at infinity.

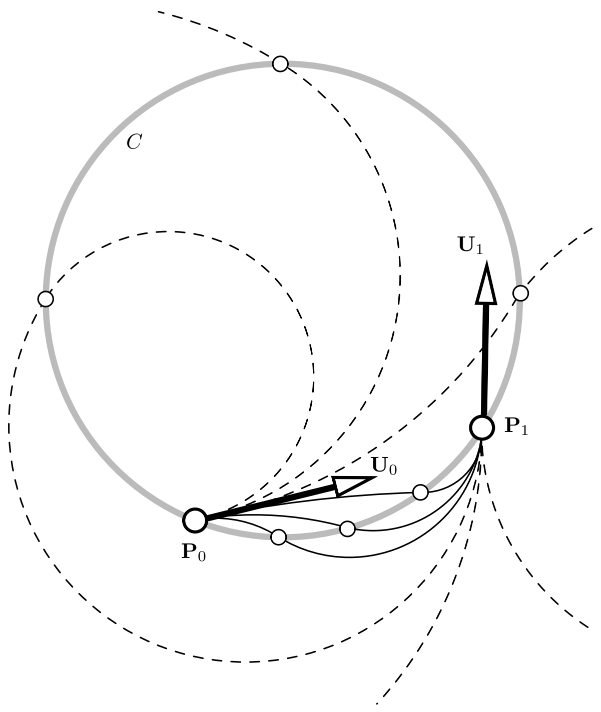

Figure 2: The system of biarcs interpolating given $G^1$ data. The solid biarcs represent the useful solutions.

To illustrate this result, Figure 2 shows the system of biarc interpolants to given $G^1$ data, along with the circle $C$.

**Remark 3** The system of biarcs is rational, since the circle $C$ can be rationally parameterized. Consequently, there is no need of using trigonometric functions in the description of this system.

Various biarc interpolation schemes were proposed in the rich literature on biarcs, which are distinguished by the choice of the joint $\mathbf{J}$. Among the most important ones are the "equal chord" biarc and the "parallel tangent" biarc. The former one is constructed so that the two segments $\mathbf{P}_0\mathbf{J}$ and $\mathbf{J}\mathbf{P}_1$ have equal lengths, while the latter one ensures that the tangent at the point $\mathbf{J}$ is parallel to the segment $\mathbf{P}_0\mathbf{P}_1$.

We propose a new choice of the joint $\mathbf{J}$ which is based on the following simple observation. Suppose, that the $G^1$ data are taken from a $C^1$ continuous curve $\mathbf{c}(t)$ - see Fig. 3:

$$\begin{array}{ll} \mathbf{P}_0 = \mathbf{c}(t_0), \quad \mathbf{U}_0 = \frac{\mathbf{c}'(t_0)}{||\mathbf{c}'(t_0)||} \quad \text{and} \\ \mathbf{P}_1 = \mathbf{c}(t_1), \quad \mathbf{U}_1 = \frac{\mathbf{c}'(t_1)}{||\mathbf{c}'(t_1)||}. \end{array} \tag{1}$$

Then – by the construction of the circle $C$ – the two vectors $\mathbf{U}_0$, $\mathbf{U}_1$ are both pointing outside or inside the circle $C$. By using a continuity argument we conclude that there exists at least one value $t_J \in (t_0, t_1)$ such that $\mathbf{c}(t_J) \in C$. We suggest to choose this point (or one of them, if more than one exist) as the joint $\mathbf{J}$ determining the interpolating $G^1$ biarc.

As observed in our numerical experiments, see Section 2.3, this choice typically produces a biarc which is closer to the original curve $\mathbf{c}(t)$ then those produced by other methods (see Table 2 and Fig. 3). The additional point $\mathbf{J}$ taken from the curve allows the biarc to better follow the shape of the curve, therefore reducing the error.

We summarize the biarc construction in the following algorithm.

**Algorithm 4** *Procedure* `Biarc`$(\mathbf{c}(t), [t_0, t_1])$
***Input:*** *A continuous curve $\mathbf{c}(t)$ defined in a closed interval $t \in [t_0, t_1]$, having a right derivative at $t_0$ and a left derivative at $t_1$.*
***Output:*** *Biarc composed of two arcs $A_0$, $A_1$.*

1. *Compute the boundary data using* (1)*.*
2. *Find the center $\mathbf{S}$ of the circle $C$, e.g., by intersecting the bisectors of $\mathbf{P}_0\mathbf{P}_1$ and of $(\mathbf{P}_0 + \mathbf{U}_0)(\mathbf{P}_1 + \mathbf{U}_1)$. Compute the implicit equation $F(x, y) = 0$ of the circle $C$.*
3. *Find a solution $t_J \in (t_0, t_1)$ of the equation $F(\mathbf{c}(t)) = 0$. If more then one solution is available in the open interval $(t_0, t_1)$, take the middle one.*
4. *Define $\mathbf{J} = \mathbf{c}(t_J)$ and find the unique arcs $A_0, A_1$ satisfying properties (1), (2) of Definition 1.*

**Remark 5** If the curve $\mathbf{c}(t)$ is an biarc, then the output of Algorithm 4 is again this biarc.

**Remark 6** The third step requires to locate an existing root of a function within a given interval. A robust implementation requires some care, since additional roots at the two boundaries of the interval exist. We combine a binary search with the regula falsi method.

## 2.2 Approximate Conversion into Biarc splines

Algorithm 4 can be used to formulate an efficient conversion procedure of arbitrary (piecewise) $G^1$ continuous planar curves into arc splines. In the sequel we present a simple non-adaptive algorithm for the conversion of $C^1$ continuous curves.

**Algorithm 7** *Procedure* `BiarcSpline`$(\mathbf{c}(t), \epsilon)$
***Input:*** *A $C^1$ curve $\mathbf{c}(t)$, $t \in [0, 1]$; prescribed error $\epsilon$.*
***Output:*** *Biarc spline $\mathbf{b}^n = \{\mathbf{b}^n_1, \ldots, \mathbf{b}^n_n\}$ approximating $\mathbf{c}(t)$.*

1. *Set $n = 1$.*
2. *Construct the sequence of Biarc interpolants:*
$$\mathbf{b}^n_i = \texttt{Biarc}\left(\mathtt{c(t)}, [\tfrac{\mathtt{i-1}}{\mathtt{n}}, \tfrac{\mathtt{i}}{\mathtt{n}}]\right) \text{ for } i = 1 \ldots n$$
*and collect them to form the biarc spline $\mathbf{b}^n$.*
3. *Evaluate the distance of the curve $\mathbf{c}(t), t \in [0, 1]$ from the spline $\mathbf{b}^n$ If the distance is greater than $\epsilon$ then set $n = 2n$ and GOTO (2). Otherwise STOP.*

**Remark 8** In practice one will use an adaptive version of this algorithm. Only those parts of the curve with errors larger than the given tolerance would be subdivided.

Also, the algorithm always produces curves with step–size $h = 2^{-n}$. Clearly, other step–sizes may be used. We restrict ourselves to these step–sizes, since they allow a particularly simple interpretation of the approximation order: when halving the segments, the errors are multiplied (roughly) by $2^{-d}$, where $d$ is the approximation order.

**Remark 9** A good estimate of the error can be obtained by sampling. Alternatively, if $\mathbf{c}(t)$ is a NURBS curve, then upper bounds on the error can be obtained by exploiting the convex hull property of Bernstein–Bézier representations, similar to the method used in [1].

## 2.3 Example

We apply the procedure `BiarcSpline` to a Bézier curve $\mathbf{c}(t)$ of degree 7. Figure 4 shows the original curve (grey) and the Biarc spline curves for $n = 4$ and $n = 8$. All arc end points lie on the curve.

The spline is globally $G^1$. For each circular arc, one end point matches the curve with $G^1$ precision (hollow dots on the Figure) and one with $G^0$ precision (grey dots on the Figure). In addition, the figure shows the offsets to the original curve and to the biarc splines at distance 2. In the latter case, the offset is again a biarc spline curve.

The biarc spline was generated for $n = 1, 2, \ldots, 2^{14}$. For each value of $n$, Table 1 reports the error and its improvement, as compared to the previous value of $n$ (ratio of two successive errors). Similarly, the error produced by approximating the offset at distance 2 is shown.

Note that the improvement ratio tends to $8 = 2^3$, which indicates that the approximation order of the method is 3. In the case of the offset, the improvement ratio tends to $4 = 2^2$, which indicates that the approximation order is 2.

In order to compare our method with the existing ones, we have also constructed biarc interpolants using the equal chord and parallel tangent methods – see Table 2. While in the limit the errors produced by the three methods are more less the same, our method achieves smaller errors for most cases. For each value of $n$, the smallest, second smallest, and the biggest error have been marked by ❶, ② and ③, respectively.

## 2.4 Discussion

According to [7] the interpolation using ”equal chord” or ”parallel tangent” biarcs has approximation order 3. The same is valid

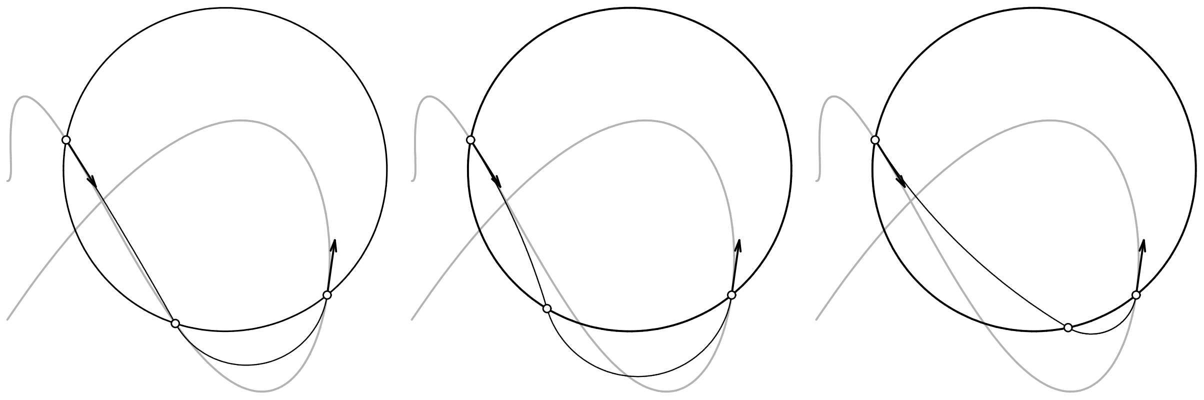

Figure 3: Biarc constructed to the data taken from a smooth curve (grey line) by our new method (left), by the "equal chord" method (middle) and by the "parallel tangent" method (right).

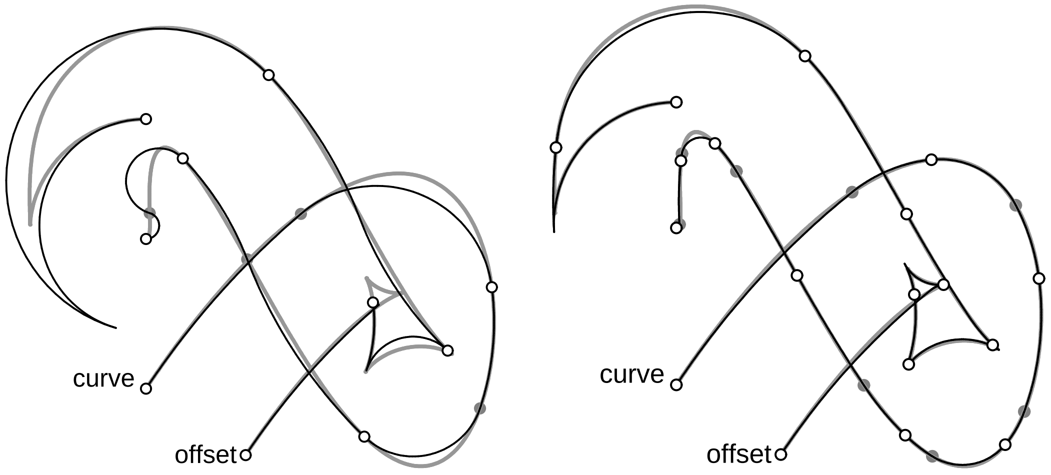

Figure 4: A continuous curve (grey line) converted into a biarc spline (black lines) composed of $4$ (left) and $8$ (right) parts. The associated approximation of the offset at distance $2$ is also shown.

for our construction, i.e., the distance of the curve $\mathbf{c}(t)$ from the spline $\mathbf{b}^n$ behaves as $\mathcal{O}(\frac{1}{n^3})$.

Offsets to the constructed biarc spline yield approximations to the offsets of the original curve. In this case, however, the approximation order is only 2 - see Section 2.3. These results about approximation orders can be proved using Taylor expansions, similarly to the proof of Proposition 16. For the sake of brevity, these proofs are omitted here.

Our method has following advantages, as compared to standard methods:

- This biarc conversion is in fact an *arc conversion*. All end points of circular arcs lie on the curve. Consequently, it is obvious which arc matches which part of the curve. For other biarc constructions, this correspondence has to be established, for example by intersecting the curve with the normal to the biarc at the joint $\mathbf{J}$. The partition of the curve is useful for an efficient evaluation of the one sided Hausdorff distance between the curve and the biarc

$$\max_{t\in[t_0,t_1]} \min_{\mathbf{p}\in A_0\cup A_1} ||\mathbf{c}(t)-\mathbf{p}||,$$

  which can be estimated via sampling points on the curve and evaluating their distance from the center of the corresponding arc – see Figure 5. In the case of a piecewise rational rational curve $\mathbf{c}(t)$, upper bounds on the Hausdorff distance can be generated by analyzing the Bernstein–Bézier representations of the compositions $F_i\circ\mathbf{c}$, where $F_i$ denotes the implicit representation of the $i$–th circular arc $A_i$.

- If the number of segments is sufficiently large, then the construction reproduces arc splines, i.e., it has *arc spline precision*.

- The construction is invariant under the group of *Möbius transformations*, which includes all Euclidean similarities and reflections with respect to circles.

- According to our numerical experiments, due to the additional joints lying on the curve, the constructed biarcs seem to follow better the shape of the original curve and the error is smaller – see Example 2.3 and Table 2.

Clearly, as a slight disadvantage to other methods, the construction of each biarc segment requires the numerical solution of a single univariate equation. Also, the generalization of this technique to the 3D case is not obvious.

Table 1: Example for Biarc conversion: Errors.

| | Curve | | Offset at distance 2 | |
|---|---|---|---|---|
| Parts | Error | Ratio | Error | Ratio |
| 1 | 4.97 | | 3.22 | |
| 2 | 2.23 | 2.233 | 2.83 | 1.136 |
| 4 | $3.98\,10^{-1}$ | 5.594 | 2.35 | 1.202 |
| 8 | $9.85\,10^{-2}$ | 4.042 | $6.74\,10^{-1}$ | 3.494 |
| 16 | $8.73\,10^{-3}$ | 11.29 | $4.66\,10^{-1}$ | 1.446 |
| 32 | $1.15\,10^{-3}$ | 7.614 | $2.36\,10^{-1}$ | 1.975 |
| 64 | $1.88\,10^{-4}$ | 6.084 | $5.36\,10^{-2}$ | 4.400 |
| 128 | $3.33\,10^{-5}$ | 5.656 | $3.17\,10^{-2}$ | 1.690 |
| 256 | $4.46\,10^{-6}$ | 7.468 | $7.89\,10^{-3}$ | 4.023 |
| 512 | $5.56\,10^{-7}$ | 8.018 | $2.11\,10^{-3}$ | 3.743 |
| 1024 | $6.90\,10^{-8}$ | 8.061 | $5.33\,10^{-4}$ | 3.951 |
| 2048 | $8.62\,10^{-9}$ | 8.009 | $1.36\,10^{-4}$ | 3.929 |
| 4096 | $1.08\,10^{-9}$ | 7.955 | $3.44\,10^{-5}$ | 3.951 |
| 8192 | $1.36\,10^{-10}$ | 7.939 | $8.64\,10^{-6}$ | 3.977 |
| 16384 | $1.71\,10^{-11}$ | 7.970 | $2.17\,10^{-6}$ | 3.989 |

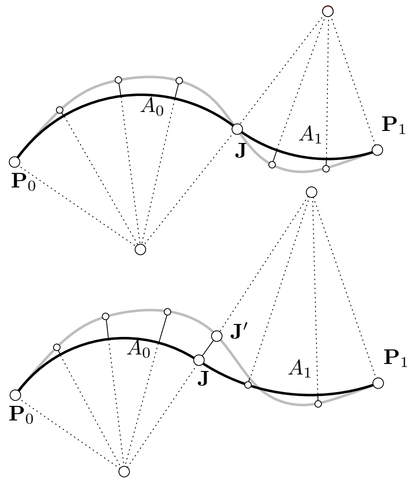

Figure 5: Evaluating the one–sided Hausdorff distance between the curve (grey) and the biarc (black) constructed by our method (top) and other methods (bottom).

Table 2: Comparison of different Biarc constructions.

| | Error | | |
|---|---|---|---|
| Parts | our method | equal chords | par. tangents |
| 1 | ❶ 4.97 | ② 5.04 | ③ 5.31 |
| 2 | ❶ 2.23 | ③ 4.73 | ② 3.46 |
| 4 | ② $3.98\,10^{-1}$ | ❶ $3.24\,10^{-1}$ | ③ $6.61\,10^{-1}$ |
| 8 | ❶ $9.85\,10^{-2}$ | ② $2.51\,10^{-1}$ | ③ $3.83\,10^{-1}$ |
| 16 | ❶ $8.73\,10^{-3}$ | ③ $6.19\,10^{-2}$ | ② $1.01\,10^{-2}$ |
| 32 | ❶ $1.15\,10^{-3}$ | ③ $4.95\,10^{-3}$ | ② $3.58\,10^{-3}$ |
| 64 | ❶ $1.88\,10^{-4}$ | ② $5.32\,10^{-4}$ | ③ $6.98\,10^{-4}$ |
| 128 | ❶ $3.33\,10^{-5}$ | ③ $5.57\,10^{-5}$ | ② $5.06\,10^{-5}$ |
| 256 | ❶ $4.46\,10^{-6}$ | ③ $5.77\,10^{-6}$ | ② $5.60\,10^{-6}$ |
| 512 | ❶ $5.56\,10^{-7}$ | ③ $6.27\,10^{-7}$ | ② $6.22\,10^{-7}$ |
| 1024 | ❶ $6.90\,10^{-8}$ | ② $7.36\,10^{-8}$ | ③ $7.38\,10^{-8}$ |
| 2048 | ❶ $8.62\,10^{-9}$ | ③ $8.88\,10^{-9}$ | ② $8.80\,10^{-9}$ |
| 4096 | ❶ $1.08\,10^{-9}$ | ② $1.09\,10^{-9}$ | ③ $1.11\,10^{-9}$ |
| 8192 | ❶ $1.36\,10^{-10}$ | ② $1.37\,10^{-10}$ | ③ $1.38\,10^{-10}$ |
| 16384 | ❶ $1.71\,10^{-11}$ | ❶ $1.71\,10^{-11}$ | ③ $1.72\,10^{-11}$ |

**Remark 10** No (bi–) arc interpolation scheme can achieve a better approximation order than 3, since even the fitting of the best arc to a shrinking segment of a $C^3$ curve has this approximation order. In order to prove this observation (see also [7]), we consider a segment $x \in [0, h]$, where $h \to 0$, of a graph of a function $y = f(x)$ which is given by its Taylor expansion. We assume – without loss of generality – that it touches the $x$–axis at $x = 0$, i.e.,

$$y = f(x) = f_2 \tfrac{x^2}{2} + f_3 \tfrac{x^3}{6} + \dots$$

with constant coefficients $f_2, f_3 \in \mathbb{R}$. Inflections at $x = 0$ are excluded: $f_2 \neq 0$. The family of circles $F_h = 0$ which are defined by the quadratic polynomials[3]

$$F_h(x, y) = x^2 + y^2 - 2x\, m(h) - 2\, y\, n(h) + a(h),$$

approximates with order three iff for any $t \in [0, 1]$ it holds that $F_h(th, f(th)) = \mathcal{O}(h^3)$. Indeed, the signed distance $d_h(x, y)$ of a point $(x, y)$ to the circle (which is negative for points within the circle) satisfies $F_h = (d_h + R_h)^2 - R_h^2$, where $R_h$ is the radius. Consequently, if $R_h = \mathcal{O}(1)$, then

$$\forall k \in \mathbb{Z}_+ : d_h(x, y) = \mathcal{O}(h^k) \iff F_h(x, y) = \mathcal{O}(h^k).$$

After replacing $m, n, a$ by Taylor expansions with respect to $h$ one gets

$$F_h(th, f(th)) = a_0 + (a_1 - 2tm_0)h + \\ +(\tfrac{1}{2}a_2 - 2m_1 t + (1 - f_2\, n_0)t^2)h^2 + \dots,$$

with the derivatives $z_i = (\mathrm{d}/\mathrm{d}h)^i z(h)\big|_{h=0}$ for $z \in \{m, n, a\}$. By comparing the coefficients with $\mathcal{O}(h^3)$ we obtain $a_0 = a_1 = a_2 = m_0 = m_1 = 0$ and $n_0 = 1/f_2$, and therefore the limit is the osculating circle of the graph at $x = 0$. Under these necessary conditions,

$$F_h(th, f(th)) = (\tfrac{1}{6}a_3 - m_2 t - n_1 f_2 t^2 - \tfrac{1}{3}\,(f_3/f_2)\, t^3)h^3 + \dots.$$

which never behaves as $\mathcal{O}(h^4)$ at general points, except for vertices of the curve, where $f_3 = 0$.

## 3 Construction of PH quintic splines

We give an algorithmic construction of PH quintic interpolants to $C^1$ boundary data, using one of the four solutions discussed in [8]. This choice is justified later by studying the approximation order. Then we apply the PH interpolation to the conversion of arbitrary $C^1$ continuous curves into PH quintic splines. We investigate the approximation order of this conversion, the required data volume and the precision of the corresponding offset approximation.

### 3.1 $C^1$ Hermite interpolation by PH quintics

The following algorithm is based on results from [8].

[3]Its center and radius are $(m, n)$ and $R = \sqrt{m^2 + n^2 - a^2}$.

**Algorithm 11** *Procedure* `PHQuintic`$(\mathbf{P}_0, \mathbf{V}_0, \mathbf{P}_1, \mathbf{V}_1)$
***Input:*** *End points $\mathbf{P}_0$, $\mathbf{P}_1$ and end point derivatives (velocity vectors) $\mathbf{V}_0, \mathbf{V}_1$. All these data are considered as complex numbers, by identifying the plane with the Argand diagram.*
***Output:*** *PH quintic $\mathbf{p}(\tau)$ defined over the interval $[0,1]$ and interpolating the input.*

1. *Transform the data to a certain canonical position,*

$$\tilde{\mathbf{V}}_0 = \frac{\mathbf{V}_0}{\mathbf{P}_1 - \mathbf{P}_0}, \qquad \tilde{\mathbf{V}}_1 = \frac{\mathbf{V}_1}{\mathbf{P}_1 - \mathbf{P}_0}.$$

2. *Compute the control points of the so–called preimage:*

$$\mathbf{w}_0 = \sqrt[+]{\tilde{\mathbf{V}}_0}, \qquad \mathbf{w}_2 = \sqrt[+]{\tilde{\mathbf{V}}_1}.$$

$$\mathbf{w}_1 = \frac{-3(\mathbf{w}_0+\mathbf{w}_2) + \sqrt[+]{120-15(\tilde{\mathbf{V}}_0+\tilde{\mathbf{V}}_1)+10\mathbf{w}_0\mathbf{w}_2}}{4},$$

*where $\sqrt[+]{\ }$ denotes square root with the positive real part.*

3. *Compute the control points of the hodograph (i.e., the first derivative vector) and transform it back to the original position:*

$$\begin{aligned}
\mathbf{h}_0 &= \mathbf{w}_0^2(\mathbf{P}_1 - \mathbf{P}_0)\\
\mathbf{h}_1 &= \mathbf{w}_0\mathbf{w}_1(\mathbf{P}_1 - \mathbf{P}_0)\\
\mathbf{h}_2 &= (\frac{2}{3}\mathbf{w}_1^2 + \frac{1}{3}\mathbf{w}_0\mathbf{w}_2)(\mathbf{P}_1 - \mathbf{P}_0)\\
\mathbf{h}_3 &= \mathbf{w}_1\mathbf{w}_3(\mathbf{P}_1 - \mathbf{P}_0)\\
\mathbf{h}_4 &= \mathbf{w}_2^2(\mathbf{P}_1 - \mathbf{P}_0).
\end{aligned}$$

4. *Compute the control points of the PH interpolant,*

$$\mathbf{p}_0 = \mathbf{P}_0, \quad \mathbf{p}_i = \mathbf{p}_{i-1} + \frac{1}{5}\mathbf{h}_{i-1} \textit{ for } i = 1\ldots5,$$

*and return the PH curve in Bernstein-Bézier representation*

$$\mathbf{p}(\tau) = \sum_{i=0}^{5} \mathbf{p}_i \binom{5}{i} \tau^i (1-\tau)^{5-i}.$$

**Remark 12** It can be verified by a direct computation that the curve $\mathbf{p}(\tau)$ interpolates the input data and that it is a PH curve, i.e., its parametric speed is a (possibly piecewise) polynomial:

$$||\mathbf{p}'(\tau)|| = ||\mathbf{w}(\tau)||^2|\mathbf{P}_1 - \mathbf{P}_0|,$$

where

$$\mathbf{w}(\tau) = \mathbf{w}_0(1-\tau)^2 + 2\mathbf{w}_1\tau(1-\tau) + \mathbf{w}_2\tau^2$$

is the so–called preimage.

**Remark 13** Algorithm `PHQuintic` fails for some rare cases of singular data. First of all the start point $\mathbf{P}_0$ and the end point $\mathbf{P}_1$ must be different because of the division in the step 1. Next, the function $\sqrt[+]{\ }$ is not defined on the line $\mathbb{R}_0^- = \{\lambda + 0\mathbf{i} : \lambda \in (-\infty, 0]\}$. In order to compute $\mathbf{w}_0$ and $\mathbf{w}_2$ it is therefore necessary that the input tangent vectors $\mathbf{V}_0$ and $\mathbf{V}_1$ are non-zero and that they are not opposite to the difference vector $\mathbf{P}_1 - \mathbf{P}_0$. Finally, we need

$$120-15(\tilde{\mathbf{V}}_0+\tilde{\mathbf{V}}_1)+10\mathbf{w}_0\mathbf{w}_2 \notin \mathbb{R}_0^-. \tag{2}$$

Note, that the vector $\mathbf{w}_0\mathbf{w}_2 = \sqrt[+]{\tilde{\mathbf{V}}_0\tilde{\mathbf{V}}_1}$ bisects the angle between $\tilde{\mathbf{V}}_0$ and $\tilde{\mathbf{V}}_1$ and its length is equal to the geometric average of the lengths of $\tilde{\mathbf{V}}_0$ and $\tilde{\mathbf{V}}_1$. Only input tangent vectors having a certain rare symmetry with respect to the difference vector $\mathbf{P}_1 - \mathbf{P}_0$ and at the same time being much longer $\mathbf{P}_1 - \mathbf{P}_0$ may violate condition (2). Various sufficient conditions can be determined for practical purposes in order to satisfy (2), e.g.,

$$||\mathbf{V}_i|| \le 3||\mathbf{P}_1 - \mathbf{P}_0||, \quad i = 0, 1.$$

## 3.2 Conversion into PH splines

Algorithm 11 can be used to formulate an efficient conversion procedure of arbitrary (piecewise) $G^1$ continuous planar curves into PH splines. The simplest non-adaptive algorithm for $C^1$ continuous curves is as follows:

**Algorithm 14** *Procedure* `PHSpline`$(\mathbf{c}(t), \epsilon)$
***Input:*** *A $C^1$ curve $\mathbf{c}(t)$, $t \in [0,1]$; prescribed error $\epsilon$.*
***Output:*** *PH spline $\mathbf{p}^n = \{\mathbf{p}_1^n, \ldots, \mathbf{p}_n^n\}$ approximating $\mathbf{c}(t)$.*

1. *Set $n = 1$.*

2. *Generate the sequence of PH interpolants:*

$$\mathbf{p}_i^n = \texttt{PHQuintic}\left(\mathbf{c}(\tfrac{\mathrm{i-1}}{\mathrm{n}}), \tfrac{1}{\mathrm{n}}\mathbf{c}'(\tfrac{\mathrm{i-1}}{\mathrm{n}}), \mathbf{c}(\tfrac{\mathrm{i}}{\mathrm{n}}), \tfrac{1}{\mathrm{n}}\mathbf{c}'(\tfrac{\mathrm{i}}{\mathrm{n}})\right)$$

*for $i = 1 \ldots n$. After a linear reparameterization join them into a spline $\mathbf{p}^n$,*

$$\mathbf{p}^n(t) = \mathbf{p}_i(nt - i + 1), t \in [\tfrac{i-1}{n}, \tfrac{i}{n}], i = 1, \ldots, n.$$

*defined over the interval $[0,1]$.*

3. *Evaluate the parametric distance between $\mathbf{c}(t)$ and $\mathbf{p}^n(t)$*

$$\begin{aligned}
&\max_{t\in[0,1]} ||\mathbf{c}(t) - \mathbf{p}^n(t)||\\
&\quad = \max_{\substack{\tau\in[0,1]\\ i=1,\ldots,n}} \left\|\mathbf{c}(\tfrac{\tau+i-1}{n}) - \mathbf{p}_i^n(\tau)\right\|.
\end{aligned}$$

*If it is greater than $\epsilon$ then set $n = 2n$ and GOTO (2). Otherwise STOP.*

The error can again be evaluated by sampling, or it can be bounded using control polygons.

**Remark 15** If the curve $\mathbf{c}(t)$ is regular and $C^1$, then for sufficiently large values of $n$ all requirements of Algorithm 11 (see Remark 13) will be satisfied for all $i$.

## 3.3 Example

We apply the procedure `PHQuintic` to the same curve $\mathbf{c}(t)$ as in Section 2.3. The PH spline was constructed for $n = 1, 2, 4, \ldots 2^{14}$ - see Fig. 6. At each step we give the sampled error and its improvement compared to the previous step (ratio

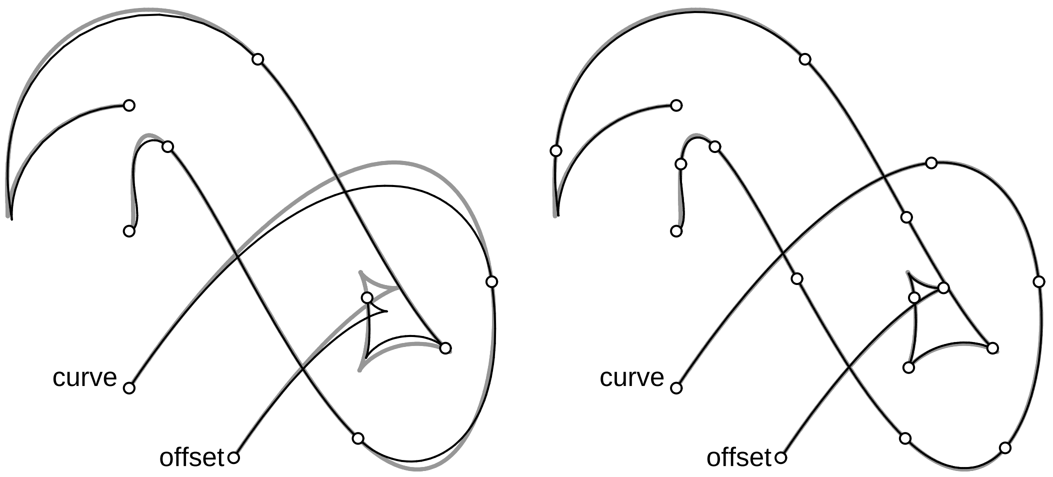

Figure 6: A polynomial curve (grey line) converted into a PH spline (black lines) composed of $4$ (left) and $8$ (right) parts. The associated approximation of the offset at distance 2 is also shown.

Table 3: Example for PH conversion: Errors

| | Curve | | Offset at distance 2 | |
|---|---|---|---|---|
| Parts | Error | Ratio | Error | Ratio |
| 1 | $9.09$ | | $9.32$ | |
| 2 | $2.65$ | 3.430 | $3.96$ | 2.354 |
| 4 | $4.57\ 10^{-1}$ | 5.798 | $1.11$ | 3.557 |
| 8 | $5.62\ 10^{-2}$ | 8.138 | $6.30\ 10^{-1}$ | 1.768 |
| 16 | $1.16\ 10^{-2}$ | 4.860 | $2.23\ 10^{-1}$ | 2.830 |
| 32 | $1.80\ 10^{-3}$ | 6.427 | $4.11\ 10^{-2}$ | 5.416 |
| 64 | $1.95\ 10^{-4}$ | 9.204 | $2.21\ 10^{-2}$ | 1.857 |
| 128 | $1.45\ 10^{-5}$ | 13.51 | $3.59\ 10^{-3}$ | 6.164 |
| 256 | $9.30\ 10^{-7}$ | 15.55 | $4.45\ 10^{-4}$ | 8.075 |
| 512 | $5.93\ 10^{-8}$ | 15.69 | $5.87\ 10^{-5}$ | 7.572 |
| 1024 | $3.71\ 10^{-9}$ | 15.96 | $7.40\ 10^{-6}$ | 7.935 |
| 2048 | $2.32\ 10^{-10}$ | 15.98 | $9.24\ 10^{-7}$ | 8.011 |
| 4096 | $1.45\ 10^{-11}$ | 15.99 | $1.16\ 10^{-7}$ | 7.989 |
| 8192 | $9.07\ 10^{-13}$ | 16.00 | $1.45\ 10^{-8}$ | 7.998 |
| 16384 | $5.67\ 10^{-14}$ | 16.00 | $1.81\ 10^{-9}$ | 8.001 |

of two successive errors). The same quantities are computed for the approximation of the offset at distance 2. Table 3 reports these numbers.

Note that the improvement ratio of the curve approximation tends to $16 = 2^4$, while the improvement ratio of the offset approximation tends to $8 = 2^3$. This indicates that the approximation order of the curve approximation by procedure `PHSpline` is $4$, while the approximation order of the offset approximation is only 3.

## 3.4 Discussion

The following proposition describes the asymptotical behavior of the procedure `PHSpline` for $n \to \infty$.

**Proposition 16** *If the curve $\mathbf{c}(t)$ is regular and $C^\infty$ on the parameter domain $t \in [0,1]$, then the approximation error satisfies*

$$\max_{t\in[0,1]} ||\mathbf{c}(t) - \mathbf{p}^n(t)|| = \mathcal{O}(\frac{1}{n^4}), \tag{3}$$

*i.e., the approximation order of the quintic PH spline curve is equal to four.*

**Proof.** We want to study the distance of interpolants $\mathbf{p}_i^n(t)$ from the corresponding segment of the curve $\mathbf{c}(t)$ for $n \to \infty$. For this purpose let us extend the definition of $\mathbf{p}_i^n(t)$ by introducing

$$\mathbf{p}(s,\tau,h) := \texttt{PHQuintic}\,(\mathbf{c}(\mathtt{s}), \mathtt{h}\mathbf{c}'(\mathtt{s}), \mathbf{c}(\mathtt{s}+\mathtt{h}), \mathtt{h}\mathbf{c}'(\mathtt{s}+\mathtt{h})),$$

which represents the PH interpolant (with parameter $\tau$) to the data taken from the segment $\{\mathbf{c}(s + h\tau), \tau \in [0,1]\}$.

Strictly speaking, the function $\mathbf{p}(s,\tau,h)$ is defined only when $s$ and $s + h$ are within $[0,1]$ and $h$ is positive. Moreover the algorithm `PHQuintic` may fail for some values - see Remark 13. However, because the curve $\mathbf{c}$ is $C^\infty$ on $[0,1]$, it can be extended to a $C^\infty$ curve on a larger open interval. Moreover, similar to Remark 15, there exists some $h_0 > 0$ such that $\mathbf{p}(s,\tau,h)$ is well–defined for $[0,1]\times[0,1]\times(0,h_0]$. Finally it can be continuously extended to $h = 0$ by setting $\mathbf{p}(s,\tau,0) = \mathbf{c}(s)$.

**Step 1.** We analyze the behavior of the error

$$\max_{\tau\in[0,1]} ||\mathbf{c}(s+\tau h) - \mathbf{p}(s,\tau,h)||$$

for $h \to 0$, when $s$ is fixed. The given $C^\infty$ curve $\mathbf{c}(t)$ has a Taylor expansion

$$\mathbf{c}(t) = \sum_{i=0}^{k} \frac{x_i}{i!}(t-s)^i + \mathbf{i}\sum_{i=0}^{k}\frac{y_i}{i!}(t-s)^i + \mathcal{O}(t-s)^{k+1}$$

of arbitrary order $k$ at the point $t = s$, with certain real coefficients $x_i = x_i(s)$ and $y_i = y_i(s)$. Since the entire construction is invariant with respect to translations, rotations and scalings, we may – in order to simplify the computations – suppose that $x_0 = y_0 = 0$, $x_1 = 1$ and $y_1 = 0$. Geometrically, we assume that the point $\mathbf{c}(s)$ is at the origin and the tangent vector at this point is $1 + 0\mathbf{i}$. Clearly, the transformations to and from these local coordinates are $C^\infty$ with respect to $s$.

The input boundary data (see step (2) of Procedure `PHSpline`) depend in a $C^\infty$ way on $h$ and the Taylor polynomials of arbitrary order $k$ at $h = 0$ can be computed[4] directly, leading to

$$\mathbf{P}_0 = \mathbf{c}(s) = 0, \quad \mathbf{V}_0 = h\mathbf{c}'(s) = h,$$

$$\mathbf{P}_1 = \mathbf{c}(s+h) = (h + \tfrac{x_2}{2}h^2 + \tfrac{x_3}{6}h^3 + \dots) + \mathbf{i}(\tfrac{y_2}{2}h^2 + \tfrac{y_3}{6}h^3 + \dots)$$

and

$$\mathbf{V}_1 = h\mathbf{c}'(s+h) = (h + x_2h^2 + \tfrac{x_3}{2}h^3 + \dots) + \mathbf{i}(y_2h^2 + \tfrac{y_3}{2}h^3 + \dots).$$

Now we analyze the procedure `PHQuintic` and consider each computed quantity as a function of $h$. At each step we verify that the obtained quantity is again $C^\infty$ with respect to $h$ and we compute the Taylor polynomials at $h = 0$. For instance, in step 1 we get

$$\begin{aligned}\tilde{\mathbf{V}}_0 = \frac{\mathbf{V}_0}{\mathbf{P}_1 - \mathbf{P}_0} = &\left(1 - \tfrac{x_2}{2}h + \tfrac{3x_2^2 - 2x_3 - 3y_2^2}{12}h^2 + \dots\right) + \\ &+\mathbf{i}\left(-\tfrac{y_2}{2}h + \tfrac{3x_2y_2 - y_3}{6}h^2 + \dots\right), \\ \tilde{\mathbf{V}}_1 = \frac{\mathbf{V}_1}{\mathbf{P}_1 - \mathbf{P}_0} = &\left(1 + \tfrac{x_2}{2}h + \tfrac{4x_3 + 3y_2^2 - 3x_2^2}{12}h^2 + \dots\right) + \\ &+\mathbf{i}\left(-\tfrac{y_2}{2}h + \tfrac{2y_3 - 3x_2y_2}{6}h^2 + \dots\right).\end{aligned}$$

The remainder of the algorithm involves only products, additions and (complex) square roots. Products and additions preserve the continuity and modify the Taylor polynomials in a straightforward way. The only technical difficulty is related to the square root with positive real part (step 2 of Algorithm 10), since it is $C^\infty$ only out of the half–line $\mathbb{R}_0^- = \{\lambda + 0\mathbf{i} : \lambda \in (-\infty, 0]\}$. Therefore we must verify that its arguments $\tilde{\mathbf{V}}_0$, $\tilde{\mathbf{V}}_1$ and $120 - 15(\mathbf{w}_0^2 + \mathbf{w}_2^2) + 10\mathbf{w}_0\mathbf{w}_2$ have a limit for $h \to 0$ within $\mathbb{C} - \mathbb{R}_0^-$. Looking at the previous Taylor expansions we see that

$$\tilde{\mathbf{V}}_0, \tilde{\mathbf{V}}_1 \to 1 + 0\mathbf{i} \quad \text{for} \quad h \to 0.$$

Also, by a direct computation,

$$120 - 15(\mathbf{w}_0^2 + \mathbf{w}_2^2) + 10\mathbf{w}_0\mathbf{w}_2 \to 100 + 0\mathbf{i}.$$

Therefore also the preimage control points $\mathbf{w}_i$ and the control points of the resulting PH curve $\mathbf{p}_i$ depend $C^\infty$ on $h$. The corresponding Taylor polynomials are listed in the Table 4.

Eventually we arrive at the expansion of the PH interpolant $\mathbf{p}(s, \tau, h)$ and compare it with the corresponding segment of the original curve:

$$\begin{aligned}&\mathbf{c}(s + \tau h) - \mathbf{p}(s, \tau, h) = \tfrac{1}{96}\tau^2(1 - \tau)^2 \cdot \\ &[\,\left(-6\,y_2y_3 + 6\,x_2x_3 - 4\,x_4 - 3\,{x_2}^3 + 9\,x_2{y_2}^2\right)h^4 \\ &\quad + \left(6\,y_2x_3 + 6\,x_2y_3 - 9\,y_2{x_2}^2 - 4\,y_4 + 3\,{y_2}^3\right)\mathbf{i}h^4\,] + \dots.\end{aligned}$$

All coefficients up to $h^3$ vanish and thus, for fixed values of $s$ and $\tau$, the difference behaves as $\mathcal{O}(h^4)$. Moreover, according to Taylor's theorem, for any values of $h, s, \tau$ there exists $h^* = h^*(h, s, \tau) \in (0, h)$ such that

$$\mathbf{c}(s + \tau h) - \mathbf{p}(s, \tau, h) = h^4 \left.\frac{\partial^4}{\partial h^4}[\mathbf{c}(s + \tau h) - \mathbf{p}(s, \tau, h)]\right|_{h = h^*}.$$

[4]We used the computer algebra system Maple 9.

**Step 2.** As shown in Step 1, the function $\mathbf{c}(s + \tau h) - \mathbf{p}(s, \tau, h)$ is $C^\infty$ with respect to $h$. Clearly it is also continuous and differentiable with respect to $s$ and $\tau$. Therefore

$$\frac{\partial^4}{\partial h^4}[\mathbf{c}(s + \tau h) - \mathbf{p}(s, \tau, h)]$$

is continuous on the compact (i.e., closed and bounded) domain $[0, 1] \times [0, 1] \times [0, h_0]$ and its absolute value can therefore be bounded by some constant $K > 0$, which does no longer depend on $s$ and $\tau$. This implies

$$||\mathbf{c}(s + \tau h) - \mathbf{p}(s, \tau, h)|| < h^4 K$$

for all $s \in [0, 1]$, $\tau \in [0, 1]$ and $h \in [0, h_0]$. Since $h = 1/n$, this completes the proof of (3). $\square$

Since PH curves possess polynomial speed functions, they have rational offsets. We can therefore approximate the offsets of the original curve $\mathbf{c}(t)$ by the offsets of $\mathbf{p}(t)$. Using again the Taylor expansion technique it can be shown that error of the offset approximation behaves as $\mathcal{O}(\frac{1}{n^3})$.

**Remark 17** Both for interpolation by biarcs and PH quintics, the approximation order of the offset construction is one less than that of the the corresponding curve construction. The offsets at distance $d$ are obtained from

$$\mathbf{o}_d(t) = \mathbf{c}'(t) + d\frac{\mathbf{c}'(t)^\perp}{||\mathbf{c}'(t)||},$$

where the derivative vector $\mathbf{c}'(t)$ has the same approximation order as the curve $\mathbf{c}(t)$, but the unit tangent vector $\frac{\mathbf{c}'(t)}{||\mathbf{c}'(t)||}$ has a reduced approximation order. In fact, numerators and denominator of the unit tangent vectors of the approximating curve segments behave as $\mathcal{O}(h)$ and they approximate the original quantities with certain accuracy $\mathcal{O}(h^4)$ and $\mathcal{O}(h^3)$ for PH curves and biarcs, respectively. After dividing the Taylor expansion (and removing the singularity at $h = 0$), the order of accuracy is reduced by 1.

**Remark 18** Any piecewise $C^\infty$ curve (such as NURBS), can be split into $C^\infty$ segments.

**Remark 19** Procedure `PHQuintic` is based on the first of the four PH interpolants described in [8] and labeled $(++)$, $(+-)$, $(-+)$ and $(--)$. The two signs correspond to the choice of complex roots with positive or negative real part for $\mathbf{w}_0, \mathbf{w}_2$ in step (2) of the procedure `PHQuintic`; this label can be therefore re-written as $(\text{sgn}(\Re(\mathbf{w}_0), \text{sgn}(\Re(\mathbf{w}_2))$. In contrast to the good behavior and approximation order of the interpolant labeled $(+, +)$, the remaining three curves exhibit an undesired asymptotical behavior. It can be proved, that for these three solutions the error (3) behaves only as $\mathcal{O}(\frac{1}{n})$. Using other interpolants than $(+, +)$ would therefore produce global splines converging much slower to the original curve.

Moreover, from the leading terms of the Taylor expansions, one can deduce the *limit shapes* of the interpolants, see Figure 7. We will skip the the technical details of how these shapes were obtained and will rather focus on their meaning. Once more, for any fixed end point $s$ we consider the interpolant $\mathbf{p}(s, \tau, h)$, constructed by taking always the solution $(+, +)$. Additionally, for each $h$ we translate, rotate and scale the interpolant such that its end points are $0+0\mathbf{i}$ and $1+0\mathbf{i}$. The curves obtained in such way converge for $h \to 0$ to some limit shape which, for all $s$, is an affine transformation of the first shape of Figure 7 (in fact only the $\mathbf{i}$ axis must be scaled depending on the curvature of $\mathbf{c}$ at the point $s$). Note, that in this case the limit shape is a parabola although the interpolants are quintic.

Using other solutions instead of $(+,+)$ would lead to different limit shapes (Figure 7) which are truly quintic curves. While the limit shape of the interpolant $(+,+)$ is smooth, the remaining three limit shapes involve one or two cusps (note that their hodographs pass through the origin).

This fact is extremely important for applications: In any algorithm based on subdivision, the solution $(++)$ must be used. Interpolation using the other solutions would produce curves with an increasing number of small loops.

## 4 Comparison and concluding remarks

Based on our theoretical results, we were able to design a new variant of biarc conversion algorithm and to justify and analyze the PH conversion algorithm. Both algorithms can be used in general for piecewise $G^1$ continuous curves. We will conclude by comparing both algorithms from the point of view of the computational effort and of the required data volume.

The construction of each biarc segment requires a numerical root finding within a given interval (in addition to basic operations). In the case of PH quintic, three complex square roots must be computed, which corresponds to evaluating 6 real square roots. The computational effort for one PH or biarc segment seems to be therefore approximately the same.

Also the required data volume is roughly the same for biarc and PH spline having the same number $n$ of segments. In practice, the piecewise rational Bézier representation would be used both for biarc and PH splines. A biarc can be represented as two rational quadratic curves, requiring $2 \times 3 \times 3 = 18$ real coefficients. A PH quintic curve can be considered as a rational quintic, requiring $3 \times 6 = 18$ coefficients. The total number of the coefficients for a spline is thus approximately the same in both cases and equals $18n$. In this counting we suppose that the arcs and quintics are represented as non-connected rational curves with arbitrary weights (not in standard form). Of course, in both cases, various more compact representations are also available. The data volume would change only by constant factor.

Both methods are distinguished by the number of segments required for converting a given curve with prescribed precision. Due to the higher approximation order in the PH case, the same precision will be achieved using a smaller number of PH segments, compared to biarcs segments.

Offsets of a PH quintic are rational curves of degree 9, requiring therefore $3 \times 10 = 30$ coefficients. Offsets of a biarc are also biarcs, requiring therefore again 18 coefficients. However, if the prescribed error is sufficiently small, then due to the higher approximation order the data volume required for offset approximation is smaller when using PH curves.

Based on given numerical examples, we display in Figure 8 the data volume required for achieving prescribed errors of the curve approximation (upper figure) and of the offset approximation (lower figure). For sufficiently small errors, the PH splines require considerably less data then the biarc splines (note the logarithmic scales on both axes). Consequently, for certain applications requiring high accuracy, PH quintics may be more appropriate than biarcs.

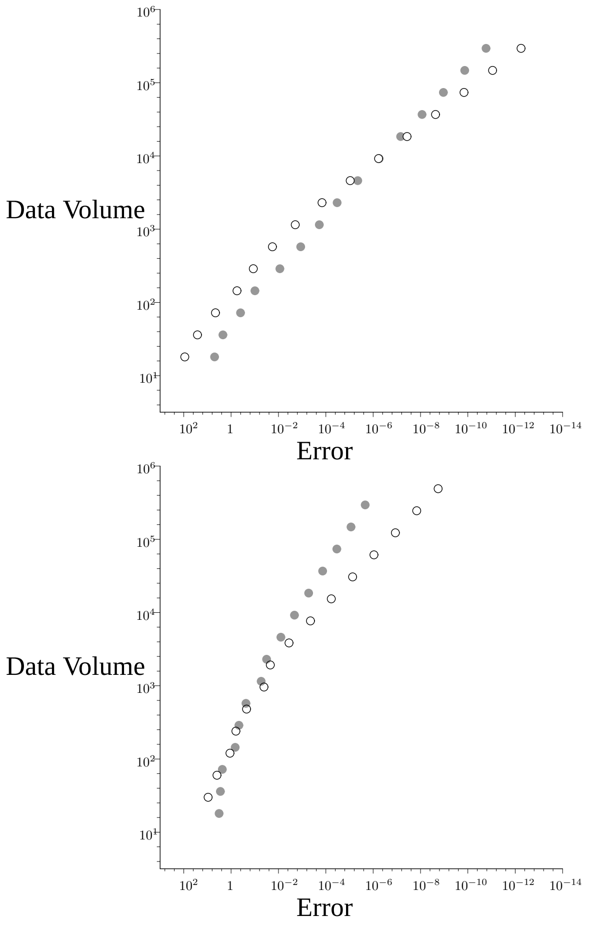

Figure 8: Data volume (top: curve, bottom: offsets) needed for achieving prescribed error when applying the procedure `BiarcSpline` (grey dots) and the procedure `PHSpline` (hollow dots).

**Acknowledgment** The authors were supported through grant P17387-N12 of the Austrian Science Fund (FWF). We thank to all referees for their comments which helped us to improve the manuscript.

## References

[1] G. Elber, I.-K. Lee and M.-S. Kim (1998), Comparing Offset Curve Approximation Methods. IEEE Comp. Graphics and Appl. 17, 62–71.

[2] R.T. Farouki (2002), Pythagorean hodograph curves, in G. Farin, J. Hoschek and M.-S. Kim (eds.), Handbook of Computer Aided Geometric Design, North-Holland, Amsterdam, 405–427.

[3] R.T. Farouki, J. Manjunathaiah, D. Nichlas, G.F. Yuan and S. Jee (1998), Variable-feedrate CNC interpolators for constant material removal rates along Pythagorean-hodograph curves. Comp.-Aided Design 30 (8), 631-640.

[4] M. Held and J. Eibl (2005), Biarc approximation of polygons within asymmetric tolerance bands. Computer-Aided Design 37(4): 357-371.

Table 4: Taylor expansions of the control points occurring in the procedure `PHQuintic`.

| | |
|---|---|
| $\mathbf{w}_0$ | $[1 - \frac{1}{4}\,x_2h + \left(-\frac{1}{12}\,x_3 + \frac{3}{32}\,{x_2}^2 - \frac{3}{32}\,{y_2}^2\right)h^2 + \left(-\frac{1}{48}\,x_4 + \frac{1}{16}\,x_2x_3 - \frac{1}{16}\,y_2y_3 - \frac{5}{128}\,{x_2}^3 + \frac{15}{128}\,x_2{y_2}^2\right)h^3 + \ldots]$<br>$+\mathbf{i}[-\frac{1}{4}\,y_2h + \left(\frac{3}{16}\,y_2x_2 - \frac{1}{12}\,y_3\right)h^2 + \left(\frac{1}{16}\,y_2x_3 - \frac{15}{128}\,y_2{x_2}^2 + \frac{5}{128}\,{y_2}^3 + \frac{1}{16}\,x_2y_3 - \frac{1}{48}\,y_4\right)h^3 + \ldots]$ |
| $\mathbf{w}_1$ | $[1 + \left(-\frac{1}{12}\,x_3 + \frac{1}{32}\,{x_2}^2 - \frac{1}{32}\,{y_2}^2\right)h^2 + \left(-\frac{1}{24}\,x_4 + \frac{7}{96}\,x_2x_3 - \frac{7}{96}\,y_2y_3 - \frac{1}{32}\,{x_2}^3 + \frac{3}{32}\,x_2{y_2}^2\right)h^3 + \ldots]$<br>$+\mathbf{i}[\left(\frac{1}{16}\,y_2x_2 - \frac{1}{12}\,y_3\right)h^2 + \left(-\frac{1}{24}\,y_4 + \frac{7}{96}\,y_2x_3 - \frac{3}{32}\,y_2{x_2}^2 + \frac{7}{96}\,x_2y_3 + \frac{1}{32}\,{y_2}^3\right)h^3 + \ldots]$ |
| $\mathbf{w}_2$ | $[1 + \frac{1}{4}\,x_2h + \left(\frac{1}{6}\,x_3 - \frac{5}{32}\,{x_2}^2 + \frac{5}{32}\,{y_2}^2\right)h^2 + \left(\frac{1}{16}\,x_4 - \frac{1}{6}\,x_2x_3 + \frac{1}{6}\,y_2y_3 + \frac{13}{128}\,{x_2}^3 - \frac{39}{128}\,x_2{y_2}^2\right)h^3 + \ldots]$<br>$+\mathbf{i}[\frac{1}{4}\,y_2h + \left(-\frac{5}{16}\,y_2x_2 + \frac{1}{6}\,y_3\right)h^2 + \left(-\frac{1}{6}\,y_2x_3 + \frac{39}{128}\,y_2{x_2}^2 - \frac{13}{128}\,{y_2}^3 - \frac{1}{6}\,x_2y_3 + \frac{1}{16}\,y_4\right)h^3 + \ldots]$ |
| $\mathbf{p}_0$ | $0 + 0\mathbf{i}$ |
| $\mathbf{p}_1$ | $\frac{1}{5}h + 0\mathbf{i}$ |
| $\mathbf{p}_2$ | $[\frac{2}{5}h + \frac{1}{20}\,x_2h^2 + \left(-\frac{1}{240}\,x_4 + \frac{1}{160}\,x_2x_3 - \frac{1}{320}\,{x_2}^3 + \frac{3}{320}\,x_2{y_2}^2 - \frac{1}{160}\,y_2y_3\right)h^4 + \ldots]$<br>$+\mathbf{i}[\frac{1}{20}\,y_2h^2 + \left(-\frac{1}{240}\,y_4 + \frac{1}{160}\,x_2y_3 + \frac{1}{160}\,y_2x_3 - \frac{3}{320}\,y_2{x_2}^2 + \frac{1}{320}\,{y_2}^3\right)h^4 + \ldots]$ |
| $\mathbf{p}_3$ | $[\frac{3}{5}h + \frac{3}{20}\,x_2h^2 + \frac{1}{60}\,x_3h^3 + \left(-\frac{1}{240}\,x_4 + \frac{1}{160}\,x_2x_3 - \frac{1}{320}\,{x_2}^3 + \frac{3}{320}\,x_2{y_2}^2 - \frac{1}{160}\,y_2y_3\right)h^4 + \ldots]$<br>$+\mathbf{i}[\frac{3}{20}\,y_2h^2 + \frac{1}{60}\,y_3h^3 + \left(-\frac{1}{240}\,y_4 + \frac{1}{160}\,x_2y_3 + \frac{1}{160}\,y_2x_3 - \frac{3}{320}\,y_2{x_2}^2 + \frac{1}{320}\,{y_2}^3\right)h^4 + \ldots]$ |
| $\mathbf{p}_4$ | $[\frac{4}{5}h + \frac{3}{10}\,x_2h^2 + \frac{1}{15}\,x_3h^3 + \frac{1}{120}\,x_4h^4 + \ldots]$<br>$+\mathbf{i}[\frac{3}{10}\,y_2h^2 + \frac{1}{15}\,y_3h^3 + \frac{1}{120}\,y_4h^4 + \ldots]$ |
| $\mathbf{p}_5$ | $[h + \frac{1}{2}\,x_2h^2 + \frac{1}{6}\,x_3h^3 + \frac{1}{24}\,x_4h^4 + \ldots]$<br>$+\mathbf{i}[\frac{1}{2}\,y_2h^2 + \frac{1}{6}\,y_3h^3 + \frac{1}{24}\,y_4h^4 + \ldots]$ |

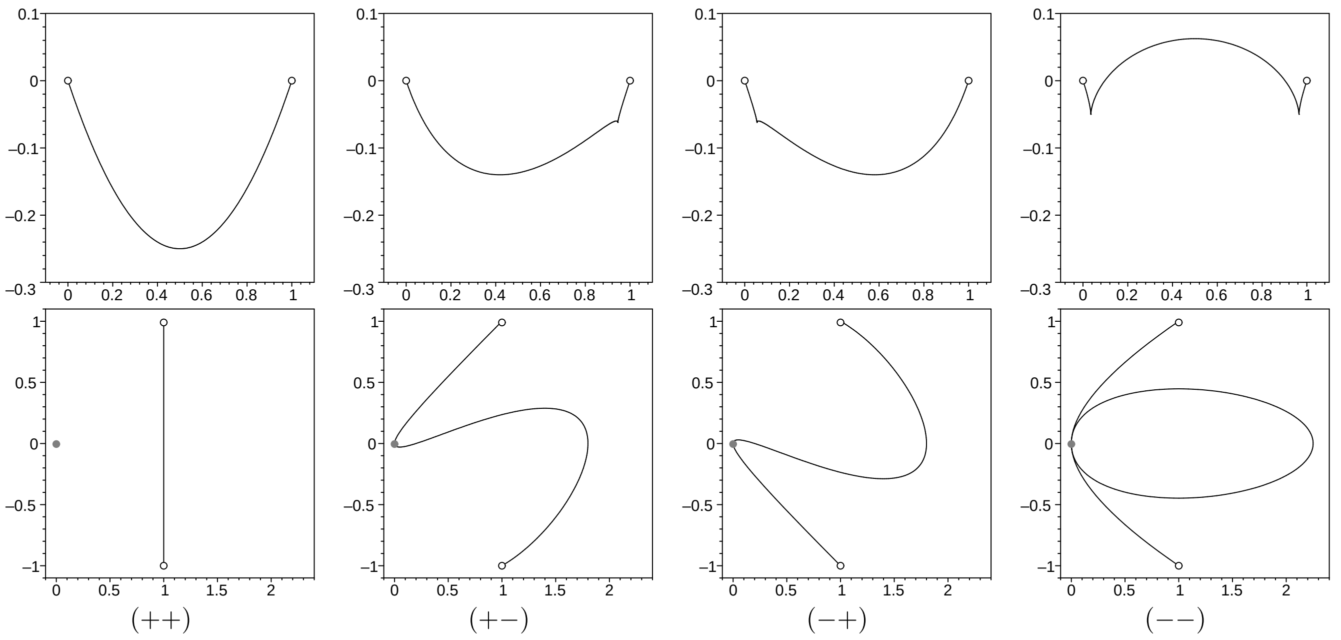

Figure 7: Limit shapes of the four PH quintic $C^1$ interpolants (top row) and of their hodographs (bottom row).

[5] B. Jüttler (2001), Hermite interpolation by Pythagorean hodograph curves of degree seven. Math. Comp. 70, 1089–1111.

[6] T. Maekawa (1999), An overview of offset curves and surfaces, Computer-Aided Design 31, pp. 165–173.

[7] D.S. Meek and D. J. Walton (1995), Approximating smooth planar curves by arc splines, J. Comput. Appl. Math. 59, pp. 221–231.

[8] H.P. Moon, R.T. Farouki and H.I. Choi (2001), Construction and shape analysis of PH quintic Hermite interpolants, Comp. Aided Geom. Design 18, 93-115.

[9] A. W. Nutbourne and R. R. Martin (1988), Differential geometry applied to curve and surface design, Vol. 1, Foundations. Ellis Horwood Ltd., Chichester; Halsted Press, New York.

[10] H. Park (2004), Error-bounded biarc approximation of planar curves Computer-Aided Design 36, pp. 1241-1251.

[11] L. A. Piegl and W. Tiller (2002), Biarc approximation of NURBS curves, Computer-Aided Design 34, pp. 807–814.

[12] J. F. Poliakoff, Y.-K. Wong and P. D. Thomas (1998), An analysis of biarc algorithms for 2-D curves. in Mathematical methods for curves and surfaces II (Lillehammer, 1997), Vanderbilt Univ. Press, Nashville, TN, pp. 401–408.

[13] M. A. Sabin (1976), The use of circular arcs to form curves interpolated through empirical data points. Rep. VTO/MS/164, British Aircraft Corporation.

[14] Z. Š´ır and B. Jüttler (2005), Constructing acceleration continuous tool paths using pythagorean hodograph curves. Mech. Mach. Theory. 40(11), 1258-1272.

[15] Y. F. Sun, A. Y. C. Nee and K. S. Lee (2004), Modifying free-formed NURBS curves and surfaces for offsetting without local self-intersection, Computer-Aided Design 36, pp. 1161–1169.

[16] Y.-F. Tsai, R.T. Farouki and B. Feldman (2001), Performance analysis of CNC interpolators for time-dependent feedrates along PH curves, Comput. Aided Geom. Design 18, no. 3, 245–265.

[17] D.S. Meek and D.J. Walton (1997), Geometric Hermite interpolation with Tschirnhausen cubics, Journal of Computational and Applied Mathematics 81, 299-309.

[18] X. Yang (2002), Effi cient circular arc interpolation based on active tolerance control, Computer Aided Design 34, pp. 1037–1046.

[19] J.-H. Yong, S.-M. Hu and J.-G. Sun (2000), Bisection algorithms for approximating quadratic B´ezier curves by $G^1$ arc splines, Computer-Aided Design 32, pp. 253–260.