Typographers, programmers and mathematicians, or the case of an æsthetically pleasing interpolation

Bogusław Jackowski

Abstract

The reason for preparing this report is that the author has been convinced for many years that John D. Hobby's algorithm for connecting Bézier segments, implemented by Donald E. Knuth in METAFONT and later transferred by Hobby to METAPOST, based on the notion of a "mock curvature", is a genuine pearl which deserves both proper acknowledgement and a far wider awareness of its existence. Of course, one can find nearly all the necessary details in the relevant papers by Hobby and in the METAFONT source, but, needless to say, it is not easy to dig through the publications. The present paper provides a full mathematical description of Hobby's interpolation algorithm, discusses its advantages and disadvantages (in particular, its instability) and compares Hobby's approach with a few selected simpler approaches.

1 Introduction

The most popular curves in computer graphics applications are based on polynomials of degrees 2 and 3. These include Bézier curves and B-splines of orders 2 and 3. Bernstein polynomials (cf. [3], [6, p. 14] and [8]) allow Bézier curves to be generalized to degree n as defined by the formula

$$\mathbf{B}(t) \overset{\text{def}}{=} \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^{n-i} \mathbf{B}_i, \tag{1}$$

where $\mathbf{B}_i$, $i = 0, 1, \ldots, n$ are points in a k-dimensional space.

In practical applications, $k = 2$ or $k = 3$. The present paper discusses the case $k = 2$, $n = 3$, i.e., planar Bézier curves; such curves are used in the PostScript language (including PostScript fonts), in page description formats such as SVG or PDF and in graphic design programs such as CorelDRAW, Adobe Illustrator and the (free software) Inkscape program. In such applications, the main problem is interpolation, i.e., connecting Bézier curves in an æsthetically pleasing manner. A single Bézier arc is capable of expressing relatively few shapes, whereas combining multiple Bézier arcs provides more possibilities.

It might seem that discussing the notion of "æsthetic pleasingness" or "beauty" with respect to a strictly mathematical problem makes no sense. However, the physiology of the human eye provides a valuable insight: the eye surprisingly easily identifies straight lines and circles in a muddle of lines (cf. figure 1). Therefore, it seems reasonable to assume that perhaps for this reason we prefer curves changing the direction uniformly (circles) or not changing the direction at all (straight lines).

Mathematicians use the notion of curvature to investigate changes of curve direction.

A method for connecting Bézier segments that minimizes curvature change has been proposed by J. R. Manning [7]. It turns out, however, that preserving exact curvature at junction points is computationally complex and not necessarily needed. J. D. Hobby's paper [2] presents an interesting solution, substantially reducing the computational complexity of this process due to a smart curvature approximation. The solution has been implemented by D. E. Knuth in the METAFONT program [6] which is basically meant for creating fonts. The same algorithm has been transferred later by Hobby to METAPOST (a modification of METAFONT which generates PostScript output).

This paper aims at investigating Hobby's method, discusses its advantages and disadvantages, and compares it with a few other interpolation methods. Hobby's approach presents an interesting case which illustrates how important a mathematician can be as a link between

This paper was presented at the BachoTEX 2010 conference, and originally published in Polish in *Acta Poligraphica* 1/2013. Translated and reprinted by permission.

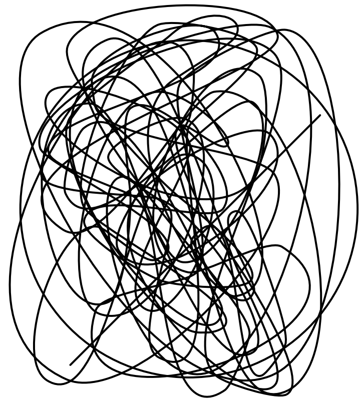

Figure 1: Exercise: find the segment of a straight line and a circle.

an artist, e.g., a typographer who creates fonts, and a programmer, who prepares relevant computer tools. A mathematician creates reality models (here: “smoothness” models) and equips programmers with a theoretical basis for creating appropriate tools.

It is assumed that readers have a basic knowledge of mathematics, for example, they are aware that the derivative of a function $f(t)$ at a point t has something to do with a limit $\lim_{\Delta t\to 0}\bigl(f(t+\Delta t)-f(t)\bigr)/\Delta t$. On the other hand, although professional mathematicians would find them unnecessary, the details of the derivation of almost all formulas have been presented here, to minimize the prerequisite mathematics. Overall, the author finds Hobby's idea for connecting Bézier curves a real gem that deserves a careful and more widely accessible description than the ones presented in [2, 5].

2 Determinants

Let us start by recalling the notion of a determinant: a second order determinant, i.e., the determinant for a pair of planar vectors, $\mathbf{v}_1=(x_1,y_1)$ and $\mathbf{v}_2=(x_2,y_2)$, is defined as

$$\det(\mathbf{v}_1,\mathbf{v}_2)\equiv\begin{vmatrix}x_1 & x_2\\ y_1 & y_2\end{vmatrix}\stackrel{\text{def}}{=}x_1y_2-x_2y_1. \tag{2}$$

The following determinant property will be useful later in this paper: it is easy to check with direct calculations for planar determinants that

$$\det(\mathbf{v}_1+p\mathbf{v}_2,\mathbf{v}_2)=\det(\mathbf{v}_1,\mathbf{v}_2+q\mathbf{v}_1)=\det(\mathbf{v}_1,\mathbf{v}_2), \tag{3}$$

where p and q are arbitrary real numbers. The geometric interpretation of a determinant will be useful as well: the absolute value of a planar determinant represents the area of the parallelogram spanned on the vectors in question and its sign depends on the orientation of the vectors — for the counterclockwise orientation of the vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ the respective determinant is positive.

A small digression: the above interpretation is, in general, so important that the outstanding Russian mathematician Vladimir I. Arnold in his article on the teaching of mathematics [1] emphasizes: *The determinant of a matrix is an (oriented) volume of the parallelepiped whose edges are its columns. If the students are told this secret (which is carefully hidden in a pure algebraic education), then the whole theory of determinants becomes easy to understand. If determinants are defined otherwise, then any sensible person will forever hate all determinants.*

3 Differentiating Bernstein polynomials

The derivative of a parametrically defined curve is determined by differentiating components of the vector defining the curve. Let us imagine a vehicle (point) whose location at time t is defined by $(x(t),y(t))$; then $(x'(t),y'(t))$ simply defines the direction and value of the velocity of this vehicle and $(x''(t),y''(t))$ defines the direction and value of vehicle's acceleration.

Let us now return to formula (1) in its general form in order to determine the value of the first and second derivative of the function $\mathbf{B}(t)$ at points $t=0$ and $t=1$, i.e., at the nodes

$\mathbf{B}(0) = \mathbf{B}_0$ and $\mathbf{B}(1) = \mathbf{B}_n$. Differentiating the polynomial once and twice produces results of the form

$$\begin{aligned} \mathbf{B}'(t) &= n(1-t)^{n-1}(\mathbf{B}_1 - \mathbf{B}_0) + \mathbf{V}(t), \\ \mathbf{B}''(t) &= (n-1)n(1-t)^{n-2}\big((\mathbf{B}_0 - \mathbf{B}_1) + (\mathbf{B}_2 - \mathbf{B}_1)\big) + \mathbf{W}(t), \end{aligned} \tag{4}$$

where $\mathbf{V}(t)$ and $\mathbf{W}(t)$ are certain polynomials (the exact formulas are unimportant here) such that $\mathbf{V}(0) = \mathbf{W}(0) = \mathbf{V}(1) = \mathbf{W}(1) = 0$, hence

$$\mathbf{B}'(0) = n(\mathbf{B}_1 - \mathbf{B}_0), \quad \mathbf{B}''(0) = (n-1)n\big((\mathbf{B}_0 - \mathbf{B}_1) + (\mathbf{B}_2 - \mathbf{B}_1)\big). \tag{5}$$

For $t = 1$, similar relations can be found:

$$\mathbf{B}'(1) = n(\mathbf{B}_n - \mathbf{B}_{n-1}), \quad \mathbf{B}''(1) = (n-1)n\big((\mathbf{B}_{n-2} - \mathbf{B}_{n-1}) + (\mathbf{B}_n - \mathbf{B}_{n-1})\big). \tag{6}$$

Note that defining "normalized" derivatives as

$$\bar{\mathbf{B}}'(t) \equiv \big(\bar{B}'_x(t), \bar{B}'_y(t)\big) \stackrel{\text{def}}{=} \frac{1}{n}\mathbf{B}'(t), \quad \bar{\mathbf{B}}''(t) \equiv \big(\bar{B}''_x(t), \bar{B}''_y(t)\big) \stackrel{\text{def}}{=} \frac{1}{(n-1)n}\mathbf{B}''(t), \tag{7}$$

we can rewrite dependencies (5) and (6) as

$$\bar{\mathbf{B}}'(0) = \mathbf{B}_1 - \mathbf{B}_0, \quad \bar{\mathbf{B}}''(0) = (\mathbf{B}_0 - \mathbf{B}_1) + (\mathbf{B}_2 - \mathbf{B}_1), \tag{8}$$

$$\bar{\mathbf{B}}'(1) = \mathbf{B}_n - \mathbf{B}_{n-1}, \quad \bar{\mathbf{B}}''(1) = (\mathbf{B}_{n-2} - \mathbf{B}_{n-1}) + (\mathbf{B}_n - \mathbf{B}_{n-1}). \tag{9}$$

The geometric interpretation, shown in figure 2, should help to understand the formulas (8) and (9), especially the expression for the second derivative. For example, vectors $\bar{\mathbf{B}}''(0)$ and $\bar{\mathbf{B}}''(1)$ shown in figure 2, indicate the direction of the centripetal force acting on a fictive point passenger of the point vehicle (mentioned at the beginning of this section) at the endpoints $\mathbf{B}_0$ and $\mathbf{B}_n$, respectively.

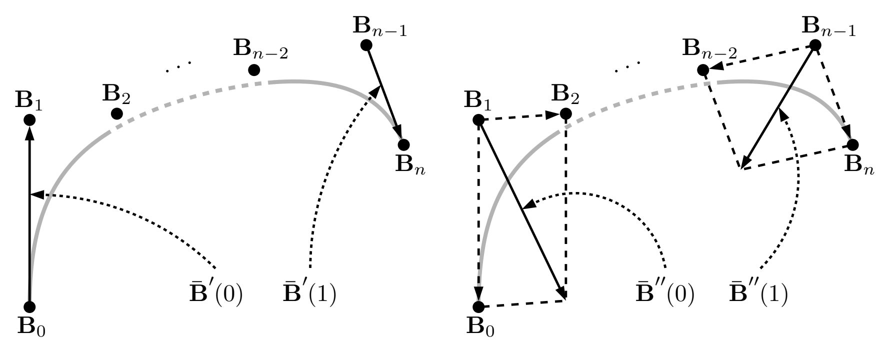

Figure 2: Geometric interpretation of the first and second derivatives of a Bernstein polynomial.

4 Curvature

In what follows, the precise definition of the curvature of a planar curve will be needed. For the purpose of this paper, the most commonly accepted definition which is simple and is (the author believes) the most natural one will be used. According to this definition, the curvature of a flat curve is simply the measure of the change of curve direction; more precisely, the change of the angle counted with respect to the curve length for an infinitesimal change of the "time" parameter — see figure 3.

Applying elementary transformations of the curvature formula presented in figure 3, one obtains

$$\begin{aligned} \kappa(t) &= \lim_{\Delta t \to 0} \frac{\Delta\varphi}{\Delta t}\,\frac{1}{\Delta s/\Delta t} = \lim_{\Delta t \to 0} \frac{\Delta\varphi}{\Delta t}\,\frac{1}{\sqrt{(\Delta x/\Delta t)^2 + (\Delta y/\Delta t)^2}} = \\ &= \frac{d\varphi}{dt}\,\frac{1}{\sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}}. \end{aligned} \tag{10}$$

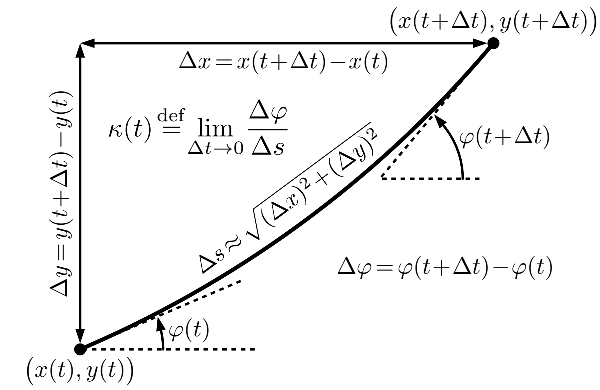

Figure 3: Definition of the curvature $\kappa(t)$ of a planar curve defined parametrically as $(x(t), y(t))$; for purposes of consistency with the definition of a determinant (2), counterclockwise angles will be considered positive (cf. figure 4).

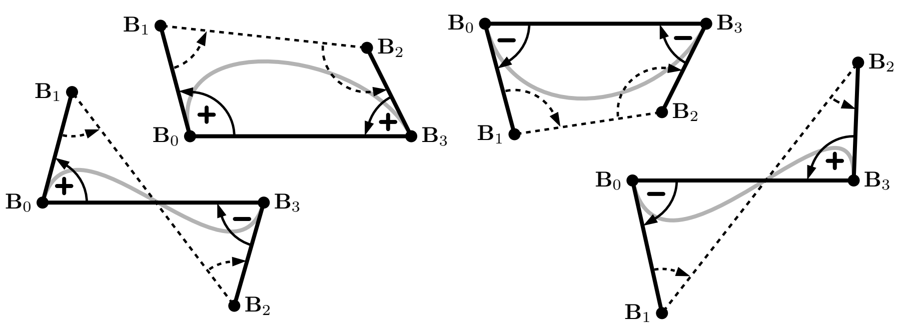

Figure 4: Convention of determining of the angle $\sphericalangle\mathbf{B}_a\mathbf{B}_b\mathbf{B}_c$ with absolute value less than π: the sign of the directed angle (from the arm $\mathbf{B}_b\mathbf{B}_a$ to the arm $\mathbf{B}_b\mathbf{B}_c$) at the vertex $\mathbf{B}_b$ will be considered consistent with the sign of the determinant $\det(\mathbf{B}_a - \mathbf{B}_b, \mathbf{B}_c - \mathbf{B}_b)$; arrows denote the assumed orientation of the angles, gray lines mark the shape of the Bézier arcs defined by the respective quadrilaterals $\mathbf{B}_0\mathbf{B}_1\mathbf{B}_2\mathbf{B}_3$, according to equation (1) for $k = 2$ and $n = 3$.

Using a more convenient notation already used in the previous point, namely, $x'(t) \equiv \frac{dx}{dt}$, $y'(t) \equiv \frac{dy}{dt}$, formula $\varphi(t) = \arctan\Big(\frac{y'(t)}{x'(t)}\Big)$, and applying basic differential calculus identities, one obtains

$$\begin{aligned}\frac{d\varphi}{dt} \equiv \varphi'(t) &= \left(\arctan\left(\frac{y'(t)}{x'(t)}\right)\right)' = \frac{1}{1+\left(\frac{y'(t)}{x'(t)}\right)^2}\left(\frac{y'(t)}{x'(t)}\right)' \\ &= \frac{1}{1+\left(\frac{y'(t)}{x'(t)}\right)^2}\,\frac{x'(t)y''(t)-y'(t)x''(t)}{x'(t)^2} \\ &= \frac{x'(t)y''(t)-y'(t)x''(t)}{x'(t)^2+y'(t)^2}\,. \end{aligned} \tag{11}$$

Observe that the numerator of formula (11) is nothing other than the determinant, namely, $\det\big((x'(t), y'(t)), (x''(t), y''(t))\big)$, which explains our interest in determinants. We will come back to determinants later.

Combining equations (10) and (11) eventually yields

$$\kappa(t) = \frac{x'(t)y''(t)-y'(t)x''(t)}{\big(x'(t)^2+y'(t)^2\big)^{3/2}}\,. \tag{12}$$

The above derivation is far from complete mathematical precision but making it more rigorous does not seem difficult (e.g., it suffices to take arccot instead of arctan in order to handle the case $x'(t) = 0$).

The resulting formula (12) can be assigned a simple geometrical meaning.

The simplest case is a straight line defined by $z(t) = (a_x t + b_x, a_y t + b_y)$, where a_x, b_x, a_y and b_y are real numbers such that $a_x^2 + a_y^2 \neq 0$. Obviously, it has curvature equal to zero which is no surprise. Consider thus a less trivial example, namely, a circle of radius R with its center in the origin of the coordinate system, oriented counterclockwise: $z(t) = \big(R\cos(t), R\sin(t)\big)$, $0 \le t < 2\pi$. Obviously, $z'(t) = \big(-R\sin(t), R\cos(t)\big)$ and $z''(t) = \big(-R\cos(t), -R\sin(t)\big)$; from formula (12), it can immediately be derived that

$$\kappa = \frac{R^2}{\big(R^2\big)^{3/2}} = \frac{1}{|R|} = \text{const.}$$

Note that changing the curve orientation to clockwise, $z(t) = \big(R\cos(t), -R\sin(t)\big)$, results in the change of the sign of curvature:

$$\kappa = -\frac{1}{|R|}\,.$$

This example explains why the entity κ^{-1} is called the *radius of curvature* and why clockwise planar curves are called *negatively oriented* while counterclockwise planar curves are called *positively oriented*.

The sign of curvature is, obviously, related to the accepted convention for determining the sign of an angle (cf. figures 3 and 4). The curvature is constant in the case of a straight line and a circle. In general, it is a local quantity. Positive curvature means that the curve (locally) turns to the left, negative curvature that it turns to the right. The points at which the curvature changes sign are called inflection points.

4.1 Curvature of Bernstein polynomials

The formula for curvature for Bernstein polynomials (12) expressed with "normalized" derivatives (7) takes the form

$$\kappa(t) = \frac{B'_x(t)B''_y(t) - B'_y(t)B''_x(t)}{\big(B'_x(t)^2 + B'_y(t)^2\big)^{3/2}} = \frac{n-1}{n}\,\frac{\bar{B}'_x(t)\bar{B}''_y(t) - \bar{B}'_y(t)\bar{B}''_x(t)}{\big(\bar{B}'_x(t)^2 + \bar{B}'_y(t)^2\big)^{3/2}}\,. \tag{13}$$

Hence, taking into account formulas (2), (8) and (9), it is easy to determine the curvature for $t = 0$ and $t = 1$

$$\begin{aligned} \kappa(0) &= \frac{n-1}{n}\,\frac{\det\big((\mathbf{B}_1 - \mathbf{B}_0), (\mathbf{B}_2 - \mathbf{B}_1) + (\mathbf{B}_0 - \mathbf{B}_1)\big)}{|\mathbf{B}_1 - \mathbf{B}_0|^3}\,,\\ \kappa(1) &= \frac{n-1}{n}\,\frac{\det\big((\mathbf{B}_n - \mathbf{B}_{n-1}), (\mathbf{B}_{n-2} - \mathbf{B}_{n-1}) + (\mathbf{B}_n - \mathbf{B}_{n-1})\big)}{|\mathbf{B}_{n-1} - \mathbf{B}_n|^3}\,, \end{aligned} \tag{14}$$

where the notation $|\mathbf{V}|$ means the length of a vector $\mathbf{V}$. Next, using property (3), one obtains

$$\begin{aligned} \kappa(0) &= \frac{n-1}{n}\,\frac{\det\big((\mathbf{B}_1 - \mathbf{B}_0), (\mathbf{B}_2 - \mathbf{B}_1)\big)}{|\mathbf{B}_1 - \mathbf{B}_0|^3}\,,\\ \kappa(1) &= \frac{n-1}{n}\,\frac{\det\big((\mathbf{B}_n - \mathbf{B}_{n-1}), (\mathbf{B}_{n-2} - \mathbf{B}_{n-1})\big)}{|\mathbf{B}_{n-1} - \mathbf{B}_n|^3}\,. \end{aligned} \tag{15}$$

Let $h_2^{0,1}$ be the distance (directed) of the point $\mathbf{B}_2$ to the straight line connecting the points $\mathbf{B}_0$ and $\mathbf{B}_1$ and, similarly, let $h_{n-2}^{n-1,n}$ be the distance of the point $\mathbf{B}_{n-2}$ to the straight line connecting the points $\mathbf{B}_{n-1}$ and $\mathbf{B}_n$; note that $h_2^{0,1}$ and $h_{n-2}^{n-1,n}$ have the same signs as the determinants $\det\big((\mathbf{B}_1 - \mathbf{B}_0), (\mathbf{B}_2 - \mathbf{B}_1)\big)$ and $\det\big((\mathbf{B}_n - \mathbf{B}_{n-1}), (\mathbf{B}_{n-2} - \mathbf{B}_{n-1})\big)$, respectively — cf. figure 5: h_a corresponds to $h_2^{0,1}$ and h_b corresponds to $h_{n-2}^{n-1,n}$, i.e., $h_1^{2,3}$.

In other words, if the point $\mathbf{B}_2$ is located to the right of the straight line directed from $\mathbf{B}_0$ to $\mathbf{B}_1$, then $h_2^{0,1} < 0$; if it is located to the left, then $h_2^{0,1} > 0$. Similarly, if the point $\mathbf{B}_{n-2}$

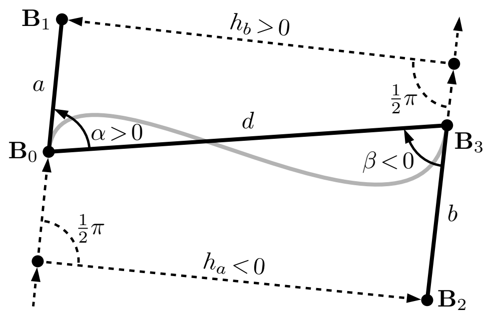

Figure 5: Geometrical interpretation of formulas (17) for a Bézier arc ($n = 3$); the curvature at the points $\mathbf{B}_0$ and $\mathbf{B}_3$ is given by (18).

is located to the right of the straight line directed from $\mathbf{B}_{n-1}$ to $\mathbf{B}_n$, then $h_{n-2}^{n-1,n} < 0$; if it is located to the left, then $h_{n-2}^{n-1,n} > 0$.

As was said in section 2, the absolute value of a determinant of a pair of vectors on a plane is equal to the area of the relevant parallelogram, thus

$$\begin{aligned} \det\bigl((\mathbf{B}_1 - \mathbf{B}_0), (\mathbf{B}_2 - \mathbf{B}_1)\bigr) &= h_2^{0,1}|\mathbf{B}_1 - \mathbf{B}_0|, \\ \det\bigl((\mathbf{B}_{n-1} - \mathbf{B}_n), (\mathbf{B}_{n-2} - \mathbf{B}_{n-1})\bigr) &= h_{n-2}^{n,n-1}|\mathbf{B}_{n-1} - \mathbf{B}_n|. \end{aligned} \tag{16}$$

Hence eventually

$$\kappa(0) = \frac{n-1}{n}\,\frac{h_2^{0,1}}{|\mathbf{B}_1 - \mathbf{B}_0|^2}, \quad \kappa(1) = \frac{n-1}{n}\,\frac{h_{n-2}^{n,n-1}}{|\mathbf{B}_{n-1} - \mathbf{B}_n|^2}. \tag{17}$$

The derivation of formulas (17) completes our considerations for the general $(n+1)$-node case.

4.2 Curvature of Bézier arcs

Now we can proceed to deriving formulas for the curvature of Bézier arcs, i.e., for the 4-node case. The resulting formulas will be used for the smooth connection of arcs.

As was already mentioned, curvature sign depends on the convention for determining the sign of an angle. In the following, we will determine the sign of an angle (less than π) in a $\mathbf{B}_0\mathbf{B}_1\mathbf{B}_2\mathbf{B}_3$ quadrilateral according to the convention accepted so far, i.e., we will consider each angle as directed from the previous to the next edge (see figures 3 and 4). Let us examine the situation presented in figure 5: obviously $\alpha > 0$ whereas $\beta < 0$. According to the definition of the directed distance introduced in section 4.1, $h_a < 0$ and $h_b > 0$.

Using the same notation as in figure 5, the curvature formulas for the nodes (endpoints) can be expressed as follows (regardless of the vertex configuration of the relevant quadrilateral):

$$\kappa(0) = \frac{2}{3}\,\frac{h_a}{a^2}, \quad \kappa(1) = \frac{2}{3}\,\frac{h_b}{b^2}. \tag{18}$$

Now, we will eliminate the entities h_a and h_b from formulas (18) in order to express curvature as a function of length (which is always non-negative) of the handles (a and b), the chord (d), and the (directed) angles α and β.

It follows from figure 6 that $c = d - b\sin(\beta)\cot(\alpha) - b\cos(\beta)$ and $h_a = -c\sin(\alpha)$, hence

$$h_a = -\left(d\sin(\alpha) - b\sin(\beta)\cos(\alpha) - b\sin(\alpha)\cos(\beta)\right) = b\sin(\alpha+\beta) - d\sin(\alpha). \tag{19}$$

Similarly

$$h_b = a\sin(\alpha+\beta) - d\sin(\beta). \tag{20}$$

Making use of formulas (19) and (20), we arrive eventually at the expression for Bézier curvature at the endpoints, expressed as a function of length of the handles and angles between the respective handles and the chord

$$\kappa(0) = \frac{2}{3}\,\frac{b\sin(\alpha+\beta) - d\sin(\alpha)}{a^2}, \quad \kappa(1) = \frac{2}{3}\,\frac{a\sin(\alpha+\beta) - d\sin(\beta)}{b^2}. \tag{21}$$

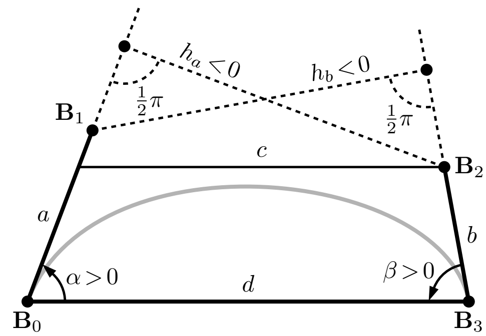

Figure 6: Auxiliary sketch for deriving formula (21) which describes the relationship between curvature and the relevant angles (here α and β) and the length of the handles and the chord (a, b and d, respectively).

Let us emphasize once again that the resulting formulas are invariant with respect to the vertex configuration of the quadrilateral $\mathbf{B}_0\mathbf{B}_1\mathbf{B}_2\mathbf{B}_3$, i.e., they do not depend on the signs of the angles α and β (cf. figure 4).

4.3 Curvature peculiarities

As figure 7 shows, the mathematical notion of curvature usually reflects what the human eye can see; curves with slightly changing curvature are likely to be called smooth or "neatly formed". If curvature changes rapidly, then we are usually able to point to the place where such a change occurs without mathematical considerations.

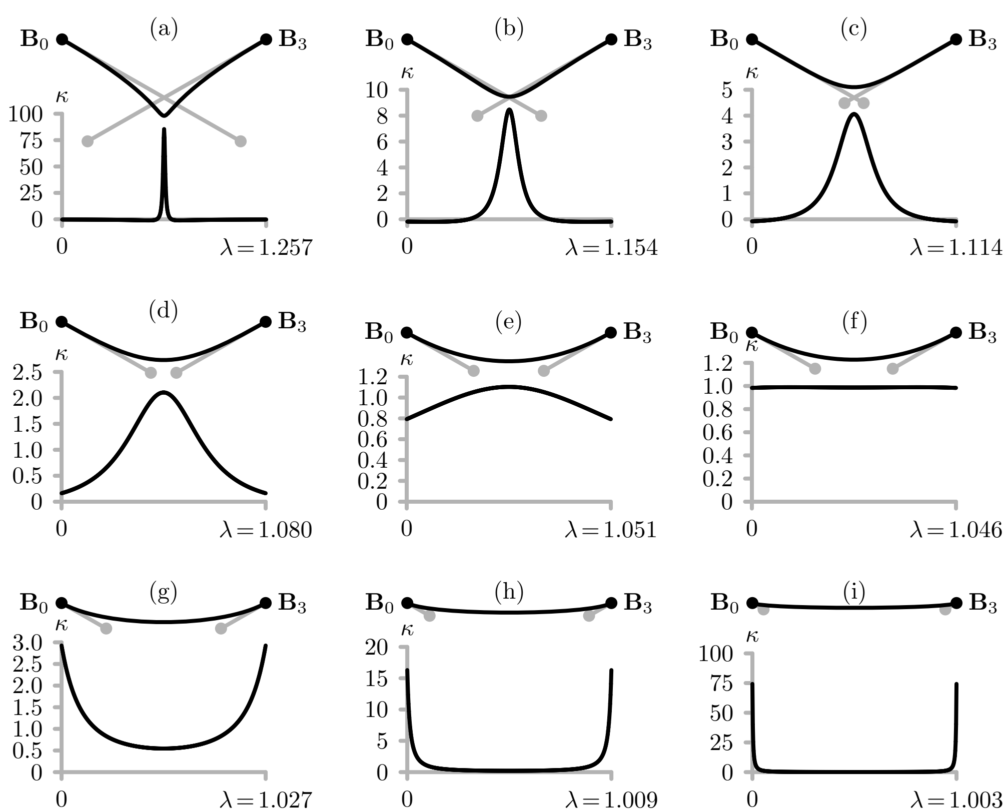

Figure 7: Curvature κ of the Bézier arc having a unit chord ($|\mathbf{B}_0 - \mathbf{B}_3| = 1$); case (f) results from applying formula (29).

It follows from equation (21), however, that curvature can achieve large values when the respective handles are short. It can even approach infinity if control points are very close to the respective endpoints. Even worse, infinite curvature can be imperceptible to the human eye, as figure 7 shows: for cases (a)–(c) the change of curvature is clearly visible in the middle of the diagram; cases (d)–(f) are perceived as fragments of a circle or "circle-like" shapes; cases (g)–(i) illustrate a transformation from a Bézier arc to a chord as the control nodes approach the respective endpoints; in the latter case, paradoxically, the curvature near the endpoints suddenly increases despite the eye not noticing it.

5 Bézier arc and a circle

Let us assume that for given angles α and β, $-\pi < \alpha, \beta \le \pi$, and for chord length $d \neq 0$, the length of the handles can be calculated from the following formulas

$$a = \frac{1}{3} d \frac{\rho(\alpha, \beta)}{\tau_a}, \quad b = \frac{1}{3} d \frac{\sigma(\alpha, \beta)}{\tau_b}, \tag{22}$$

where $\rho(\alpha, \beta)$ and $\sigma(\alpha, \beta)$ are certain functions, not necessarily given explicitly (we will discuss them in a moment) and τ_a and τ_b are given real (positive) numbers; we will call these numbers *tensions*.

In the METAFONT and METAPOST programs, tensions can be specified explicitly using the '`tension`' operator ([6, pp. 129–132 and p. 136, ex. 14.15]). Usually, in practical applications, $\tau_a = \tau_b = 1$ which is the default tension value in METAFONT and METAPOST.

Let us assume for a moment that $\tau_a = \tau_b = 1$; moreover, assume that $d = 1$ (in other words, adjust units in such a way that $d = 1$). Then, the first derivative vectors at the endpoints ($t = 0$ and $t = 1$; cf. (5) and (6)) have the length $\rho(\alpha, \beta)$ and $\sigma(\alpha, \beta)$, respectively. This explains why the functions ρ and σ are called *velocity functions*.

In applications under consideration, it is natural to assume a basic symmetry property: an exchange of variables should return a geometrically congruent (mirrored) figure, precisely

$$\rho(\alpha, \beta) = \sigma(\beta, \alpha). \tag{23}$$

Therefore, we are actually dealing with one function, but it is more convenient to distinguish between the velocity functions at the points $t = 0$ and $t = 1$.

In the METAFONT and METAPOST programs, the velocity function is defined with a relatively complex heuristic formula (due to Hobby):

$$\rho(\alpha, \beta) = \sigma(\beta, \alpha) = \frac{2 + \sqrt{2}(\sin\alpha - \frac{1}{16}\sin\beta)(\sin\beta - \frac{1}{16}\sin\alpha)(\cos\alpha - \cos\beta)}{\left(1 + \frac{1}{2}(\sqrt{5} - 1)\cos\alpha + \frac{1}{2}(3 - \sqrt{5})\cos\beta\right)}, \tag{24}$$

substantiated in his paper [2]. An alternative form is also given (defined for $0 < |\alpha| \le \beta < \pi$), which supposedly works better in asymmetrical cases but is far more complex computationally and more difficult to analyze theoretically:

$$\rho(\alpha, \beta) = f(\alpha, \beta) + \gamma(\beta)\sin\left(\psi_\beta\left(\frac{\alpha}{\beta}\right)\right), \quad \sigma(\alpha, \beta) = f(\alpha, \beta) - \gamma(\beta)\sin\left(\psi_\beta\left(\frac{\alpha}{\beta}\right)\right), \tag{25}$$

where

$$f(\alpha, \beta) = \frac{m\mu^2 + \mu + 2n}{\mu + n\cos(\nu) + n}, \quad m = 0.2678306, \quad n = 0.2638750,$$

$$\mu = (\beta - \alpha)\left(\frac{\beta - \alpha}{2\beta}\right)^{1.402539}, \qquad \nu = \frac{\alpha + \beta}{2}\left(\frac{2\beta}{\alpha + \beta}\right)^{0.7539063}, \tag{26}$$

$$\gamma(\beta) = \frac{1.17}{\pi}\beta - 0.15\sin(2\beta), \quad \psi_\beta(x) = \pi\left(x + (x^2 - 1)\left(\left(0.32 - \frac{\beta}{2\pi}\right)x + 0.5 - \frac{\beta}{2\pi}\right)\right).$$

Knuth, in his collection of essays entitled *Digital Typography* [4], devotes some attention to the choice of functions ρ and σ; he proposes, among others, the following formula

$$\rho(\alpha,\beta)=\sigma(\beta,\alpha)=\frac{2\sin(\beta)}{\left(1+\cos\left(\dfrac{\alpha+\beta}{2}\right)\right)\sin\left(\dfrac{\alpha+\beta}{2}\right)} \tag{27}$$

and suggests how to improve it.

A formula of this kind, but much simpler, was first proposed by J. R. Manning in his already cited paper [7] (Hobby's formula (24) is in fact a sophisticated modification of Manning's formula):

$$\rho(\alpha,\beta)=\sigma(\beta,\alpha)=\frac{2}{1+c\cos(\beta)+(1-c)\cos(\alpha)}\,. \tag{28}$$

Manning suggests setting $c=\frac{2}{3}$. The idea behind Manning's formula stems from a simple observation: if $\alpha=\beta$, then the formula $\rho(\alpha)=2/\bigl(1+\cos(\alpha)\bigr)$ provides a good approximation of a circle by a Bézier arc. Namely, if both (symmetric) handles of the Bézier arc $\mathbf{B}(t)$ have the length given by

$$\rho(\alpha)=\frac{d}{3}\,\frac{2}{1+\cos(\alpha)}\,, \tag{29}$$

where d is the length of the chord of the arc and α is the angle between handles and the chord, then the point $\mathbf{B}\bigl(\frac{1}{2}\bigr)$ coincides with the center of the segment of a circle going through the points $\mathbf{B}(0)$ and $\mathbf{B}(1)$ and tangent to the Bézier arc at these points. Such a Bézier arc, especially for small angles, is visually indistinguishable from a circle — more information on the precision of such an approximation can be found in [4]; cf. also figure 7f.

All in all, we may note that the velocity function is intended to be a generalization, heuristic of course, to two parameters of the function $\rho(\alpha)$ guaranteeing a good approximation (equation (29)) of a circle by a Bézier arc. This remains, of course, related to our primary goal — striving to obtain a possibly smooth joining of Bézier arcs. Bézier arcs computed using velocity functions as defined above are expected to imitate circles optically which, because of their constant curvature, are supposed to be an ideal candidate for an "æsthetic model". This observation provides a clue for our final results.

6 Smooth joining of Bézier arcs — J. D. Hobby's method

Finally, we have all the necessary tools and measures necessary to face the following task: given a series of $n+1$ points on a plane $\mathbf{P}_0$, $\mathbf{P}_1,\ldots,$ $\mathbf{P}_n$, $n>1$, connect these points with Bézier arcs in such a way that the result forms a possibly "elegant" (smooth, neat) curve.

The *sine qua non* condition is obvious: at a connection point the curve should not change its direction abruptly, meaning that the handles at the connection points must be collinear. (Collinearity of handles is a weaker condition than equality of derivatives; the equality of derivatives, as defined by formulas (5) and (6), implies the equality of the lengths of adjacent handles.)

This condition does not guarantee a unique solution. We might impose an additional condition that the curvature at a junction point is the same on both sides of this point. However, this leads to a system of trigonometrical equations that is hard to solve.

In such cases, mathematicians and physicist often replace a function with its linear approximation (not always justifiably); in this particular case, the sine function would be replaced by a linear function of the argument and the relevant velocity function — with a function identically equal to 1.

$$\sin(\alpha)\approx\alpha,\quad \rho(\alpha,\beta)\approx\sigma(\alpha,\beta)\approx 1. \tag{30}$$

Such an approximation can be justified mathematically for small values of arguments (with respect to their absolute values) for (24) and (28) (formulas (25) and (27) are undefined for $\alpha=\beta=0$), but it will also be used for values significantly different from zero. Therefore, we can consider (30) to be a heuristic simplifying assumption.

Applying relation (30) to the formula for Bézier curvature at endpoints, derived from combining formulas (21) and (22), yields

$$\begin{aligned}\kappa(0) &= \frac{2}{3}\,\frac{\frac{1}{3}d\frac{\sigma(\alpha,\beta)}{\tau_b}\sin(\alpha+\beta) - d\sin(\alpha)}{\left(\frac{1}{3}d\frac{\rho(\alpha,\beta)}{\tau_a}\right)^2} = \frac{2\tau_b^{-1}\sigma(\alpha,\beta)\sin(\alpha+\beta) - 6\sin(\alpha)}{d\tau_a^{-2}\rho^2(\alpha,\beta)},\\ \kappa(1) &= \frac{2}{3}\,\frac{\frac{1}{3}d\frac{\rho(\alpha,\beta)}{\tau_a}\sin(\alpha+\beta) - d\sin(\beta)}{\left(\frac{1}{3}d\frac{\sigma(\alpha,\beta)}{\tau_b}\right)^2} = \frac{2\tau_a^{-1}\rho(\alpha,\beta)\sin(\alpha+\beta) - 6\sin(\beta)}{d\tau_b^{-2}\sigma^2(\alpha,\beta)},\end{aligned} \tag{31}$$

and we have

$$\kappa(0) \approx \bar\kappa(0) \stackrel{\text{def}}{=} \frac{2\tau_b^{-1}(\alpha+\beta) - 6\alpha}{d\tau_a^{-2}}, \quad \kappa(1) \approx \bar\kappa(1) \stackrel{\text{def}}{=} \frac{2\tau_a^{-1}(\alpha+\beta) - 6\beta}{d\tau_b^{-2}}. \tag{32}$$

The function $\bar\kappa$ defined above is the *mock curvature* introduced by Hobby in [2], mentioned by Knuth in [6] and discussed in detail in [5, § 274–277]. It constitutes a key to joining Bézier arcs smoothly: first we solve a set of linear equations, derived from (32), obtaining the values of angles between chords and the respective handles, and then we compute the length of the relevant handles from (22).

6.1 Equations

At the junctions of Bézier arcs, we require that mock curvature be preserved, bringing the task to solving a set of linear equations. The formulation of these equations consists of a series of tedious but elementary calculations.

First, let us adopt the notation and conventions presented in figure 8.

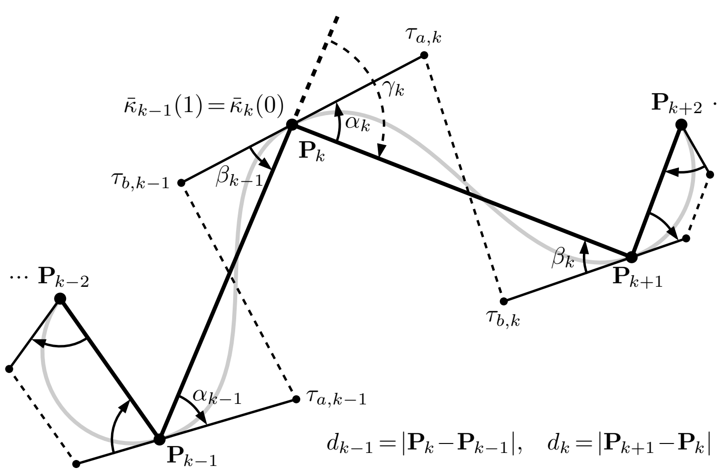

Figure 8: Notation assumed for formulating linear equations (38); index k refers (unlike in [5]) to the quantities related to the Bézier arc based on the endpoints $\mathbf{P}_k$ and $\mathbf{P}_{k+1}$.

Second, observe that the collinearity of handles at junction points implies that

$$\alpha_k + \beta_{k-1} + \gamma_k = 0 \ \text{ for } \ k = 1, 2, \ldots, n-1, \tag{33}$$

where γ_k is the turning angle of the broken line $\mathbf{P}_0\mathbf{P}_1\ldots\mathbf{P}_n$ at the point $\mathbf{P}_k$.

Third, observe moreover that the preservation of the mock curvature at the point $\mathbf{P}_k$ results in a simple linear equation

$$\bar\kappa_{k-1}(1) \equiv \frac{2\tau_{a,k-1}^{-1}(\alpha_{k-1}+\beta_{k-1}) - 6\beta_{k-1}}{d_{k-1}\tau_{b,k-1}^{-2}} = \frac{2\tau_{b,k}^{-1}(\alpha_k+\beta_k) - 6\alpha_k}{d_k\,\tau_{a,k}^{-2}} \equiv \bar\kappa_k(0). \tag{34}$$

By making use of (33), we can eliminate β_{k-1} and β_k from (34) obtaining

$$\frac{\tau_{a,k-1}^{-1}(\alpha_{k-1}-\alpha_k-\gamma_k)-3(-\alpha_k-\gamma_k)}{d_{k-1}\tau_{b,k-1}^{-2}}-\frac{\tau_{b,k}^{-1}(\alpha_k-\alpha_{k+1}-\gamma_{k+1})-3\alpha_k}{d_k\tau_{a,k}^{-2}}=0, \tag{35}$$

i.e.,

$$\begin{aligned}\frac{\tau_{a,k-1}^{-1}}{d_{k-1}\tau_{b,k-1}^{-2}}\alpha_{k-1}+\left(\frac{3-\tau_{a,k-1}^{-1}}{d_{k-1}\tau_{b,k-1}^{-2}}+\frac{3-\tau_{b,k}^{-1}}{d_k\tau_{a,k}^{-2}}\right)\alpha_k+\frac{\tau_{b,k}^{-1}}{d_k\tau_{a,k}^{-2}}\alpha_{k+1}\\=-\frac{3-\tau_{a,k-1}^{-1}}{d_{k-1}\tau_{b,k-1}^{-2}}\gamma_k-\frac{\tau_{b,k}^{-1}}{d_k\tau_{a,k}^{-2}}\gamma_{k+1}.\end{aligned} \tag{36}$$

Now, by introducing one-letter symbols for known values (coefficients)

$$\begin{gathered}A_k\stackrel{\text{def}}{=}\frac{\tau_{a,k-1}^{-1}}{d_{k-1}\tau_{b,k-1}^{-2}},\quad B_k\stackrel{\text{def}}{=}\frac{3-\tau_{a,k-1}^{-1}}{d_{k-1}\tau_{b,k-1}^{-2}},\quad C_k\stackrel{\text{def}}{=}\frac{3-\tau_{b,k}^{-1}}{d_k\tau_{a,k}^{-2}},\\D_k\stackrel{\text{def}}{=}\frac{\tau_{b,k}^{-1}}{d_k\tau_{a,k}^{-2}},\quad E_k\stackrel{\text{def}}{=}-B_k\gamma_k-D_k\gamma_{k+1},\end{gathered} \tag{37}$$

we obtain a set of $n-1$ linear equations with $n+1$ unknowns:

$$\begin{aligned}A_1\alpha_0+(B_1+C_1)\alpha_1+D_1\alpha_2&=E_1,\\A_2\alpha_1+(B_2+C_2)\alpha_2+D_2\alpha_3&=E_2,\\A_3\alpha_2+(B_3+C_3)\alpha_3+D_3\alpha_4&=E_3,\\&\cdots\\A_{n-1}\alpha_{n-2}+(B_{n-1}+C_{n-1})\alpha_{n-1}+D_{n-1}\alpha_n&=E_{n-1}.\end{aligned} \tag{38}$$

Thus, two more equations are needed.

If a closed curve is to be obtained, then $\alpha_n=\alpha_0$ and the problem reduces to n unknowns. The missing equation can be obtained by assuming that mock curvature is preserved at the point $\mathbf{P}_0=\mathbf{P}_n$,

$$A_0\alpha_{n-1}+(B_0+C_0)\alpha_0+D_0\alpha_1=E_0. \tag{39}$$

If an open curve is to be obtained, then either the angles α_0 and α_n should be given explicitly which reduces the number of unknowns by 2, or another condition must be found.

Before we proceed to this issue, let us discuss a purely technical detail; the angle that we want to find is, in fact, β_{n-1} because α_n is located outside the broken line $\mathbf{P}_0\mathbf{P}_1\ldots\mathbf{P}_n$. It is, however, more convenient (because of the symmetry of formulas), to use the unknown α_n; therefore, we accept an artificial condition

$$\gamma_n=0. \tag{40}$$

This means that relation (33) for the angles β_0 and β_{n-1}, in the case of an open curve, takes the form

$$\beta_0=-\alpha_1-\gamma_1,\quad \beta_{n-1}=-\alpha_n. \tag{41}$$

We may also assume that mock curvature at the endpoints is given explicitly or implicitly or, alternatively, we may impose the requirement of the equality of mock curvature at the endpoints and at the respective neighbouring nodes, namely, $\bar\kappa_0(0)=\bar\kappa_0(1)$ and $\bar\kappa_{n-1}(0)=\bar\kappa_{n-1}(1)$. The latter condition can be generalized in a natural way as follows:

$$\bar\kappa_0(0)=\omega_0\bar\kappa_0(1),\quad \omega_n\bar\kappa_{n-1}(0)=\bar\kappa_{n-1}(1),\quad \omega_0,\omega_n\ge 0. \tag{42}$$

It is exactly this idea that has been used in the METAFONT program (there is, however, no possibility of explicitly setting the mock curvature at a given point). The additional parameters ω_0 and ω_n are called *curls*.

Both METAFONT and METAPOST allow to explicitly set curls at the endpoints of a path using the '`curl`' operator ([6, pp. 128ff.]); by default, curl=1. Conditions (42) can be transformed to

$$\begin{aligned} \frac{2\tau_{b,0}^{-1}(\alpha_0-\alpha_1-\gamma_1)-6\alpha_0}{d_0\tau_{a,0}^{-2}} &= \omega_0\frac{2\tau_{a,0}^{-1}(\alpha_0-\alpha_1-\gamma_1)+6(\alpha_1+\gamma_1)}{d_0\tau_{b,0}^{-2}}, \\ \omega_n\frac{2\tau_{b,n-1}^{-1}(\alpha_{n-1}-\alpha_n)-6\alpha_{n-1}}{d_{n-1}\tau_{a,n-1}^{-2}} &= \frac{2\tau_{a,n-1}^{-1}(\alpha_{n-1}-\alpha_n)+6\alpha_n}{d_{n-1}\tau_{b,n-1}^{-2}}. \end{aligned} \tag{43}$$

by using the definition of mock curvature (32) to develop $\bar\kappa_0(0)$, $\bar\kappa_0(1)$, $\bar\kappa_{n-1}(0)$, and $\bar\kappa_{n-1}(1)$.

Using relation (41) to order equations (43) yields

$$\begin{aligned} \left(\frac{\tau_{b,0}^{-1}}{\tau_{a,0}^{-2}}-\frac{3}{\tau_{a,0}^{-2}}-\frac{\tau_{a,0}^{-1}\omega_0}{\tau_{b,0}^{-2}}\right)\alpha_0-\left(\frac{\tau_{b,0}^{-1}}{\tau_{a,0}^{-2}}-\frac{\tau_{a,0}^{-1}\omega_0}{\tau_{b,0}^{-2}}+\frac{3\omega_0}{\tau_{b,0}^{-2}}\right)\alpha_1 &= \left(\frac{\tau_{b,0}^{-1}}{\tau_{a,0}^{-2}}-\frac{\tau_{a,0}^{-1}\omega_0}{\tau_{b,0}^{-2}}+\frac{3\omega_0}{\tau_{b,0}^{-2}}\right)\gamma_1, \\ \left(\frac{\tau_{b,n-1}^{-1}\omega_n}{\tau_{a,n-1}^{-2}}-\frac{3\omega_n}{\tau_{a,n-1}^{-2}}-\frac{\tau_{a,n-1}^{-1}}{\tau_{b,n-1}^{-2}}\right)\alpha_{n-1}-\left(\frac{\omega_n\tau_{b,n-1}^{-1}}{\tau_{a,n-1}^{-2}}-\frac{\tau_{a,n-1}^{-1}}{\tau_{b,n-1}^{-2}}+\frac{3}{\tau_{b,n-1}^{-2}}\right)\alpha_n &= 0, \end{aligned} \tag{44}$$

By multiplying the first equation (44) by $-\tau_{a,0}^{-2}$, and the second by $-\tau_{b,n-1}^{-2}$ (thanks to which the obtained formulas are easier to compare with those given in [5]), we finally obtain the two missing equations:

$$\begin{aligned} C_0\alpha_0+D_0\alpha_1 &= E_0, \\ A_n\alpha_{n-1}+B_n\alpha_n &= 0, \end{aligned} \tag{45}$$

where

$$\begin{aligned} &C_0=\omega_0\frac{\tau_{a,0}^{-3}}{\tau_{b,0}^{-2}}+3-\tau_{b,0}^{-1}, \quad D_0=\omega_0\frac{\tau_{a,0}^{-2}}{\tau_{b,0}^{-2}}\big(3-\tau_{a,0}^{-1}\big)+\tau_{b,0}^{-1}, \quad E_0=-D_0\gamma_1, \\ &A_n=\omega_n\frac{\tau_{b,n-1}^{-2}}{\tau_{a,n-1}^{-2}}\big(3-\tau_{b,n-1}^{-1}\big)+\tau_{a,n-1}^{-1}, \quad B_n=\omega_n\frac{\tau_{b,n-1}^{-3}}{\tau_{a,n-1}^{-2}}+3-\tau_{a,n-1}^{-1}. \end{aligned} \tag{46}$$

One may of course ask about the solvability of the set of linear equations thus formulated. In this way, we go back to determinants as those are exactly the determinants (of degree equal to the number of unknowns) that mathematicians use for examining this problem; columns (or rows) of coefficients of the set of equations constitute relevant "vectors". If a determinant for a given set of equations is non-zero, then there exists a unique solution.

We will not go into mathematically advanced analysis of this issue (readers interested in details, please refer to [2]). We content ourselves by adducing without proof the theorem given in [2] (see also [5, § 276]): if for $0\le k\le n-1$ we assume $\tau_{a,k}\ge\frac{3}{4}$, $\tau_{b,k}\ge\frac{3}{4}$ (a limitation built into METAFONT and METAPOST), then the equation sets for both a closed path (38)+(39) and an open path (38)+(45) have unique solutions and, in general, any disturbance introduced at a given node, caused, e.g., by the change of a node location, disappears exponentially as the distance from this node grows (more precisely: the change of conditions at node k results in an angle change at node j proportional to $2^{-|k-j|}$). It should be noted, however, that there are cases for which a tiny perturbation at a certain node may cause the global change of the shape of the resulting curve (see subsection 6.4, figures 10 and 11).

6.2 Two-point case

So far we have assumed that $n>1$. The discussion can be extended to the $n=1$ case, i.e., to the two-point case, but this requires additional assumptions.

If $n=1$ and we are dealing with a closed curve, then it is reasonable to assume that this is simply a degenerate case, i.e., $\mathbf{P}_0=\mathbf{P}_1$ and, moreover, lengths of both handles are equal to zero — cf. equation (22). Note that the METAFONT path expression '`(0,0)..(1,0)..cycle`' corresponds, in fact, to the case $n=2$: $\mathbf{P}_0=(0,0)$, $\mathbf{P}_1=(1,0)$, $\mathbf{P}_2=(0,0)=\mathbf{P}_0$.

If $n=1$ and we are dealing with an open curve, then the degenerate case $\mathbf{P}_0=\mathbf{P}_1$ can be treated as above. In other cases, the set of equations reduces to two equations (45). In this case, $E_0=0$ (because $\gamma_1=0$ by (40)) which means that $\alpha_0=\alpha_1=0$, provided the relevant matrix determinant is non-zero. However, for default values of coefficients defining curl and tension, i.e., 1, we have $C_0=D_0=A_1=B_1=3$, thus the determinant equals 0 which means that the set of equations (45) has infinitely many solutions and therefore again an additional assumption is needed. The simplest way would be to assume that if no angles are given explicitly, then $\alpha_0 = \alpha_1 = 0$.

If the angle α_0 is given, then α_1 can be computed from the second equation of system (45); and vice versa, if the angle α_1 is given, then α_0 can be computed from the first equation of system (45); the asymmetry is due to the asymmetry of the curl boundary conditions (equations (45) and (46)) and to the fact that curl and angle cannot both be specified at a given endpoint.

At this stage, we have all the angles in question either given explicitly or calculated. The only remaining task is to calculate the length of the handles from formula (22).

6.3 Operator '`atleast`'

The above discussion explains how all the METAFONT path constructors work except for the '`tension atleast`' operator ([6, p. 129, 132, 136, and ex. 14.15]) which is a variant of the '`tension`' operator (section 5, equation (22)). It is a primitive operator in the METAFONT and METAPOST engines but, in principle, it could be implemented using METAFONT/METAPOST macros. The recipe is as follows: first skip the '`atleast`' operator (modifier), i.e., use the standard '`tension`' operator thereby reducing the problem to a known issue.

Figure 9 illustrates the further procedure. Assume that we have computed all the handles by using the algorithm described above. Then we check whether the nodes defining the relevant Bézier segments form a concave quadrilateral (case (d) in figure 9). If so, then the extension of one of the handles bisects the other — the handle being bisected should be shortened by moving its endpoint to the intersection point (which, on the one hand, results in increasing tension and, on the other hand, introduces a discontinuity of mock curvature). More precisely, the length of the shortened handle ($\bar{a}$ in figure 9) can be determined using the law of sines. In such a way, inflection points can be avoided. Sometimes, however, surprising results are obtained, namely, when the "shortening handle" is very short.

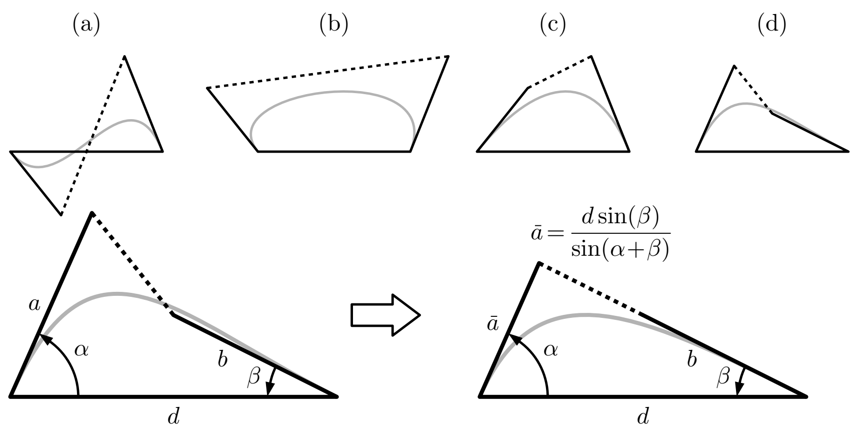

Figure 9: Implementation of the '`atleast`' operator: only the configuration of the control points presented in figure (d) is to be taken into account; (a), (b) and (c) configurations are ignored.

Let us note that the '`tension atleast`' operator is a little like increasing the tension just enough to avoid concave quadrilaterals such as shown in figure 9d, except that increasing the tension might affect the direction of angles α and β.

6.4 Instability of the interpolation algorithm — the straight angle issue

One of the crucial steps of Hobby's algorithm is determining angles γ_k between subsequent segments of the broken line $\mathbf{P}_0\mathbf{P}_1 \ldots \mathbf{P}_n$ — see figure 8 and equation (33). If the broken line turns by an angle significantly different from the straight angle (its absolute is much less than π), then the algorithm behaves stably, i.e., small changes in the location of the points $\mathbf{P}_0$, $\mathbf{P}_1, \ldots, \mathbf{P}_n$ yield small changes in the shape of the resulting curve. Unfortunately, an essential problem arises for angles close to the straight angle. This problem is not in the least specific to METAPOST and METAFONT. It seems to be an intrinsic quandary in the realm of discrete graphics (in general, numerical methods), as small perturbations are unavoidable there due to rounding errors.

In the case of METAFONT and METAPOST, it is a convention adopted by the authors that is the immediate source of problems, namely, both programs operate on angles less than or equal (with respect to the absolute value) to π. If a temporary value occurs which does not meet this limitation, then it is reduced to the appropriate numerical range by adding or subtracting a multiple of 2π. For example, an angle $\pi+\varepsilon$, where $\pi>\varepsilon>0$, will be replaced by the angle $\pi+\varepsilon-2\pi=-\pi+\varepsilon$. If $\varepsilon\approx 0$, then unstable behavior can be expected (we omit here the issue of the precision of the computer representation of the number π).

In order to see how instability can manifest, let us consider a trivial example, namely, a cyclic path built from two nodes, $\mathbf{P}_0$ and $\mathbf{P}_1$; as mentioned in subsection 6.2, it is convenient to regard this example as a three-point case, i.e., to apply the interpolation algorithm to the points $\mathbf{P}_0$, $\mathbf{P}_1$ and $\mathbf{P}_2$, where $\mathbf{P}_0=\mathbf{P}_2$ which means that the respective broken line turns by the straight angle at the points $\mathbf{P}_0=\mathbf{P}_2$ and $\mathbf{P}_2$. The question is: to the right or to the left?

For this case, $\alpha_0=\alpha_2$ and equations (38) and (39) boil down to the system of two linear equation with two unknowns

$$\begin{aligned}(B_0+C_0)\alpha_0+(A_0+D_0)\alpha_1&=E_0,\\(A_1+D_1)\alpha_0+(B_1+C_1)\alpha_1&=E_1.\end{aligned}\tag{47}$$

Coefficients on the left side of system (47) are given by simple formulas, namely

$$A_0=A_1=D_0=D_1=\frac{1}{d},\quad B_0=B_1=C_0=C_1=\frac{2}{d},\tag{48}$$

where $d=|\mathbf{P}_1-\mathbf{P}_0|$, provided the default values of the curl and tension parameters are assumed. Coefficients E_0 and E_1 depend on the turning angles γ_0 and γ_1 of the broken line $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2$ (cf. equation (33))

$$E_0=-\frac{2\gamma_0+\gamma_1}{d},\quad E_1=-\frac{\gamma_0+2\gamma_1}{d}.\tag{49}$$

The above conditions can be rewritten as

$$4\alpha_0+2\alpha_1=-(2\gamma_0+\gamma_1),\quad 2\alpha_0+4\alpha_1=-(\gamma_0+2\gamma_1).\tag{50}$$

If we assume turning to the left, i.e., $\gamma_0\approx\pi$ and $\gamma_1\approx\pi$, then $\alpha_0\approx\frac{1}{2}\pi$ and $\alpha_1\approx\frac{1}{2}\pi$; but if we assume turning to the right, i.e., $\gamma_0\approx-\pi$ and $\gamma_1\approx-\pi$, then $\alpha_0\approx-\frac{1}{2}\pi$ and $\alpha_1\approx-\frac{1}{2}\pi$. In other words, using the angle $\approx\pi$ instead of $\approx-\pi$ reverses the orientation of the resulting path. Theoretically, two other cases might occur: $\gamma_0\approx-\pi$, $\gamma_1\approx\pi$ and $\gamma_0\approx\pi$, $\gamma_1\approx-\pi$, which would produce figure-eight shaped curves. It turns out, however, that the METAFONT and METAPOST path expression '`P0..P1..cycle`' usually creates positively oriented nearly circular paths, occasionally creates negatively oriented nearly circular paths (see figure 10), but never figure-eight shaped curves; however, if the direction at one node is specified explicitly, figure-eight shapes may occur: the expression '`(0,0){down}..(100,0)..cycle`' creates an oval path while the expression '`(0,0){up}..(100,0)..cycle`' creates an figure-eight shape.

If the broken line "turns back", i.e., turns by a nearly straight angle, the algorithm implemented in METAFONT and METAPOST tries to detect this situation and, in most cases, chooses the positive angle (see [5, § 454]). Sometimes, however, despite the precautions taken, the algorithm fails to decide properly whether the angle in question is a positively oriented nearly straight angle, or a distorted by rounding errors negatively oriented (on purpose) nearly straight angle.

Based on the foregoing considerations, it is not difficult to predict that if input data contains several (nearly) collinear adjacent points, instability is bound to emerge.

Without a detailed analysis, we present one more example, a little bit less trivial: we apply the interpolation algorithm to creating a cyclic path (nota bene, cyclicality is not crucial) for input data consisting of five nearly collinear points, arranged horizontally. It turns out that shifting one of the nodes (vertically) by 2^{-16}, i.e., by the smallest non-zero value accepted by

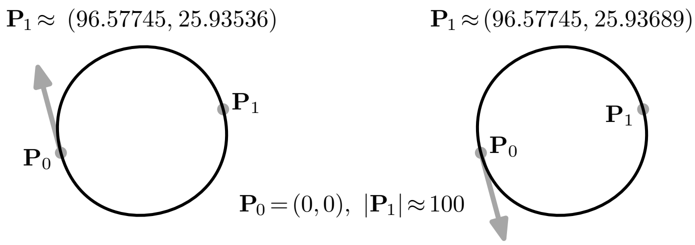

Figure 10: A very small location change of the point $\mathbf{P}_1$ reverses the orientation of the resulting curve (gray arrow); the curve on the left side of the figure was created with the following METAFONT/METAPOST path expression '`(0,0) .. ((100,0) rotated 15.03189) .. cycle`', the curve on the right side with the slightly modified path expression '`(0,0) .. ((100,0) rotated (15.03189+.0003)) .. cycle`'.

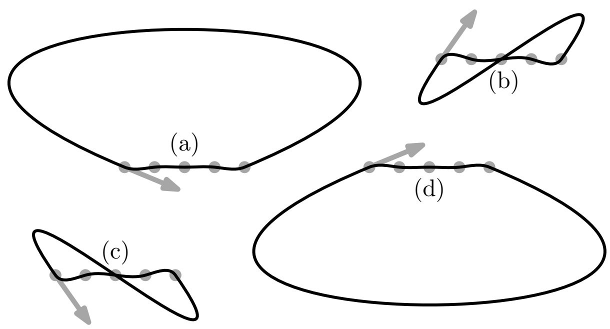

Figure 11: Even a minimal location change of a single point can result not only in the change of orientation, but also in a change of the shape of the resulting curve. In case (a), points are distributed uniformly, i.e., $\mathbf{P}_i = (20i, 0)$, for $i = 0, 1, \ldots, 4$; in case (b), $\mathbf{P}_1$ is shifted up by 2^{-16}, in case (c), $\mathbf{P}_4$ is shifted down by 2^{-16}; in case (d), $\mathbf{P}_0$ is shifted down by 2^{-16}; in all cases, the METAFONT/METAPOST path expression '`P0..P1..P2..P3..P4..cycle`' was used.

METAFONT and "canonical" implementations of METAPOST (without double precision), can change not only the orientation of the resulting curve, but also its shape — see figure 11.

Although the problem cannot, by its nature, be cured, it is rather innocuous in practice. Nevertheless, as suggested by John D. Hobby (in private communication), the following fixes to METAFONT/METAPOST implementations can be proposed:

- One natural idea is that users should avoid approaching this situation. METAFONT and METAPOST already have internal variables that can be used to enable and disable certain error messages and an appropriate error message could be helpful.
- Since the discontinuities are caused by adding or subtracting 2π from the γ_k angles, users should perhaps have a way of specifying whether or not to do this.

6.5 Other technical details

It is clear that the implementation of such a complex algorithm as the one described above abounds in technical challenges. Not all of them can be discussed in a paper like this one. Information on specific implementation-related solutions, such as, for example, the fact that METAFONT imposes a limit ≤ 12 (for default tension values) on Hobby's velocity function (24) or that the curl value can be altered in some cases, can be found in [5] (§ 116 and § 296, respectively). A word of warning is needed, however: the program source is not an easy read.

7 Comparison of selected interpolation methods

The interpolation algorithm described in section 6 works well in most applications, except for the (rarely encountered in practice) cases of instability. This fact is well-known to METAFONT and METAPOST users. It does not mean, however, that the algorithm is recommended in every situation. Sometimes it is better to use other algorithms and this issue will be briefly surveyed at the end of our discussion.

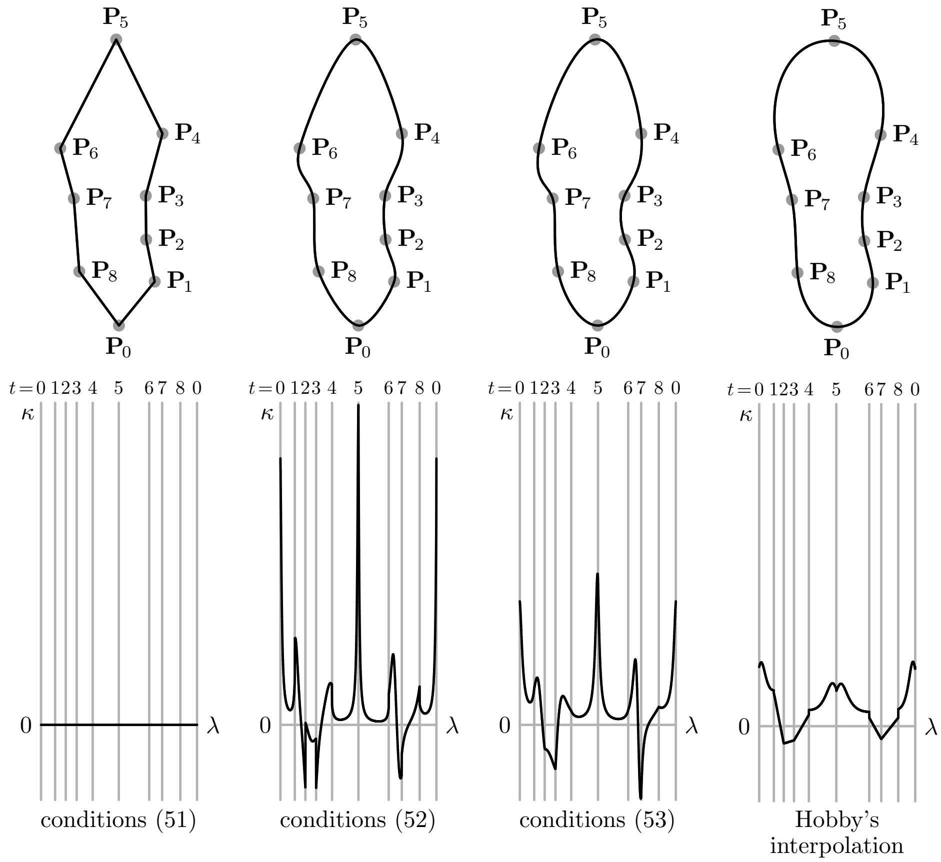

Figure 12: Comparison of interpolation methods for a closed curve; curvature scale κ is common for all the cases; parameter λ, as in figure 7, refers to the length of a path.

Let a series of points $\mathbf{P}_0$, $\mathbf{P}_1, \ldots, \mathbf{P}_n$, $n > 1$, be given as above and let $\mathbf{P}_k$, $\mathbf{P}_k^a$, $\mathbf{P}_{k+1}^b$, and $\mathbf{P}_{k+1}$ (a — after, b — before) be the nodes of a Bézier arc based on the chord $\mathbf{P}_k\mathbf{P}_{k+1}$.

The easiest way to connect the points is to use a broken line:

$$\mathbf{P}_k^a = \mathbf{P}_k + \frac{1}{3}(\mathbf{P}_{k+1} - \mathbf{P}_k), \quad \mathbf{P}_k^b = \mathbf{P}_k + \frac{1}{3}(\mathbf{P}_{k-1} - \mathbf{P}_k). \tag{51}$$

METAFONT and METAPOST users do not need to calculate the handles explicitly. Such a connection of points can be obtained by using the macro '`--`', which expands to the following path expression '`{curl 1} .. {curl 1}`'. A similar optical effect could be achieved by superimposing conditions $\mathbf{P}_k^a = \mathbf{P}_k$, $\mathbf{P}_k^b = \mathbf{P}_k$; the macro '`---`', which expands to '`... tension infinity ...`', yields a similar result, i.e., $\mathbf{P}_k^a \approx \mathbf{P}_k$, $\mathbf{P}_k^b \approx \mathbf{P}_k$, but the latter two methods, seemingly equivalent, can yield perceptibly different curves.

In order to smooth corners, setting the first derivative explicitly (cf. equation (5)), e.g.,

$$\mathbf{P}_k - \mathbf{P}_k^b = \mathbf{P}_k^a - \mathbf{P}_k = \frac{1}{3}\frac{(\mathbf{P}_{k+1} - \mathbf{P}_{k-1})}{2}, \tag{52}$$

may yield fairly satisfactory results. This is a local method, i.e., a change of coordinates of point $\mathbf{P}_k$ has an impact on the resulting curve only at points $\mathbf{P}_{k-2}$, $\mathbf{P}_{k-1}$, $\mathbf{P}_{k+1}$, and $\mathbf{P}_{k+2}$.

A non-local method, far less complex than the method described in section 6, involves the assumption of the equality (continuity) of the first and second derivatives at junction points (see formulas (5) and (6)):

$$\begin{gathered}\mathbf{P}_k - \mathbf{P}_k^b = \mathbf{P}_k^a - \mathbf{P}_k\,,\\ (\mathbf{P}_{k-1}^a - \mathbf{P}_k^b) + (\mathbf{P}_k - \mathbf{P}_k^b) = (\mathbf{P}_k - \mathbf{P}_k^a) + (\mathbf{P}_{k+1}^b - \mathbf{P}_k^a).\end{gathered} \tag{53}$$

As already mentioned, Hobby's interpolation usually produces satisfactory results. Figure 12, an example excerpted from [2] and [7], provides a convincing argument. Let us note that the continuity of the first and second derivative (conditions (53)) implies the continuity of curvature (equation (12)) while Hobby's interpolation does not guarantee the continuity of curvature (figure 12, $t = 1$, 4, 6, 8).

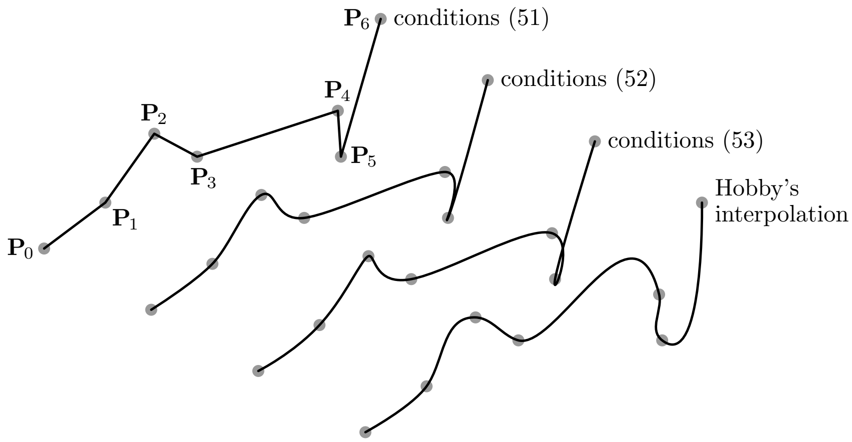

Figure 13: Comparison of interpolation methods for a sample 7-point set of data; an additional boundary condition $\mathbf{P}_0 = \mathbf{P}_0^a$, $\mathbf{P}_6 = \mathbf{P}_6^b$ has been imposed in each case except the broken line interpolation.

Hobby's interpolation exhibits the smallest curvature fluctuation and produces the most regular optical curves in comparison with the other methods. In particular, curves obtained by applying condition (53) do not look as smooth. It turns out that curvature discontinuity is nearly imperceptible unless it is accompanied by a noticeable change of curve direction. (cf. remarks in subsection 4.3). And vice versa, even though curvature is constant for a broken line, namely, equal to zero except nodes where its value is undefined, the eye immediately catches these points because curve direction is not preserved there.

A typical task for which Hobby's method cannot be recommended is the visualisation of empirical data — METAFONT and METAPOST generate too "rotund" shapes (figure 13). Condition (53) also produces hardly acceptable results in this particular case — the diagram of a function should rather not reveal loops. In such cases, the best approach seems to be using as simple a method as possible, e.g., (51) or (52).

Of course, there are many variants of the methods briefly reviewed in this section. We have not discussed them here at length not because they are not worthy of being applied. On the contrary, it is good to know that the unquestionably excellent Hobby's interpolation algorithm can sometimes successfully be replaced by a simpler one.

8 Acknowledgements

The author very gratefully thanks Alan Hoenig, Daniel Luecking, and John Hobby for reviewing an earlier version of this paper and suggesting several significant improvements, as well as correcting drafting errors.

References

[1] Arnold, Vladimir I., *On teaching mathematics*, 1997, `http://pauli.uni-muenster.de/~munsteg/arnold.html`, accessed 14.04.2013.

[2] Hobby, John. D., “Smooth, Easy to Compute Interpolating Splines”, *Discrete and Computational Geometry*, 1986, vol. 1(2), `ftp://db.stanford.edu/pub/cstr/reports/cs/tr/85/1047/CS-TR-85-1047.pdf`, accessed 14.04.2013.

[3] Joy, Kenneth I., *Bernstein Polynomials*, 2000, `http://www.idav.ucdavis.edu/education/CAGDNotes/CAGDNotes/Bernstein-Polynomials.pdf`, accessed 14.04.2013.

[4] Knuth, Donald E., *Digital Typography*, CSLI Publications, Stanford, California, 1999.

[5] Knuth, Donald E., *METAFONT: The Program, Computers & Typesetting, vol. D*, Addison-Wesley, Reading, Massachusetts, 1986.

[6] Knuth, Donald E., *The METAFONTbook, Computers & Typesetting, vol. C*, Addison-Wesley, Reading, Massachusetts, 1986.

[7] Manning, J. R., “Continuity Conditions for Spline Curves”, *Computer Journal*, 1974, vol. 17(2), p. 181–186, `http://comjnl.oxfordjournals.org/content/17/2/181.full.pdf`, accessed 14.04.2013.

[8] `http://en.wikipedia.org/wiki/Bernstein_polynomial`, accessed 03.06.2013.

◇ Bogusław Jackowski
GUST
Gdańsk, Poland
b_jackowski (at) gust dot org dot pl