Journal for Geometry and Graphics
Volume 9 (2005), No. 1, 67–75.

Blending curves

Albert Wiltsche

Institute of Geometry, Graz University of Technology
Kopernikusgasse 24, A-8010 Graz, Austria
email: wiltsche@tugraz.at

Abstract. Two arbitrarily given curves $k_1(t)$ and $k_2(t)$ are blended to a third curve $b(t)$ so that b joins k_1 and k_2 in given points A_1 and B_2 C^l- and C^m-continuously, respectively. In order to meet this objective we use polynomial functions $\alpha_{lm}(t)$ for the blending process. The Casteljau algorithm for curves is used in a special way to build the blended curve $b(t)$. Furthermore we can use our construction to generate interpolating spline curves.

Key Words: Spline curves, Hermite interpolation, interpolation

MSC 2000: 68U05

1. Introduction

Our aim is to find a simple method to join two curves $k_1(t)$ and $k_2(t)$ by a curve $b(t)$ as shown in Fig. 1. For both curves k_1 and k_2 we first choose a parameter interval $[t_0, t_1]$ which choice affects the final curve b. Then we assume $t_0 = 0$ and $t_1 = 1$ to simplify the notation which can always be achieved by a simple parameter transformation. The curve $b(t)$ shall start with the parameter value $t_0 = 0$ at the point $k_1(0) = A_1$ on k_1 and end with the parameter value $t_1 = 1$ at the point $k_2(1) = B_2$ on k_2. So we have

$$\begin{aligned} b(0) &= k_1(0) = A_1, \qquad k_1(1) = A_2, \\ b(1) &= k_2(1) = B_2, \qquad k_2(0) = B_1. \end{aligned}$$

For $b(t)$ we require

$$\begin{aligned} \frac{d^i}{(dt)^i}\, b(0) &= \frac{d^i}{(dt)^i}\, k_1(0), \quad i = 0, \dots, l, \\ \frac{d^i}{(dt)^i}\, b(1) &= \frac{d^i}{(dt)^i}\, k_2(1), \quad i = 0, \dots, m. \end{aligned} \tag{1}$$

ISSN 1433-8157/$ 2.50

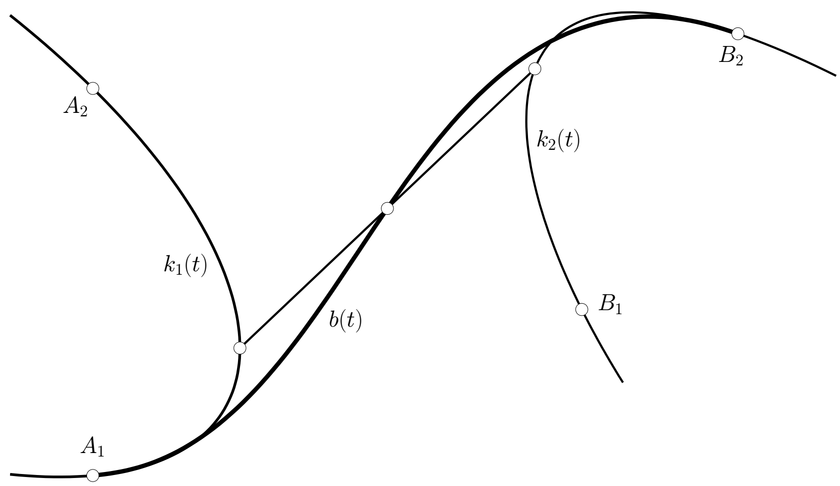

Figure 1: Two curves $k_1(t)$ and $k_2(t)$ are linearly blended to a third curve $b(t)$

2. Linear blending

Our construction shall be a linear blending process between the curves $k_1(t)$ and $k_2(t)$ (see Fig. 1). Therefore we will use a function $\alpha_{lm}(t)$ and the following setup to find a suitable curve $b(t)$:

$$b(t) = \alpha_{lm}(t) \cdot k_1(t) + [1 - \alpha_{lm}(t)] \cdot k_2(t), \quad t \in [0..1] \subset \mathbb{R}. \tag{2}$$

The function $\alpha_{lm}(t)$ has to be chosen so that (1) holds.[1]

First we calculate the derivatives of the curve $b(t)$

$$\begin{aligned}
b(t) &= \alpha_{lm}(t) \cdot [k_1(t) - k_2(t)] + k_2(t) \\
\dot{b}(t) &= \dot{\alpha}_{lm}(t) \cdot [k_1(t) - k_2(t)] + \alpha_{lm}(t) \cdot [\dot{k}_1(t) - \dot{k}_2(t)] + \dot{k}_2(t) \\
\ddot{b}(t) &= \ddot{\alpha}_{lm}(t) \cdot [k_1(t) - k_2(t)] + 2\dot{\alpha}_{lm}(t) \cdot [\dot{k}_1(t) - \dot{k}_2(t)] + \\
&+ \alpha_{lm}(t) \cdot [\ddot{k}_1(t) - \ddot{k}_2(t)] + \ddot{k}_2(t) \\
&\vdots \\
\frac{d^j}{(dt)^j} b(t) &= \sum_{i=0}^{j} \binom{j}{i} \frac{d^{j-i}}{(dt)^{j-i}} \alpha_{lm}(t) \cdot \left[\frac{d^i}{(dt)^i} k_1(t) - \frac{d^i}{(dt)^i} k_2(t) \right] + \frac{d^j}{(dt)^j} k_2(t).
\end{aligned} \tag{3}$$

If we compare (3) with (1) we find the following conditions for $\alpha_{lm}(t)$:

$$\begin{aligned}
\alpha_{lm}(0) &= 1 \\
\alpha_{lm}(1) &= 0 \\
\frac{d^j}{(dt)^j}\alpha_{lm}(0) &= 0, \quad 1 \le j \le l, \\
\frac{d^j}{(dt)^j}\alpha_{lm}(1) &= 0, \quad 1 \le j \le m.
\end{aligned} \tag{4}$$

Remark 1 In order to keep the construction simple we will use polynomial functions $\alpha_{lm}(t)$. Instead of that one can also use rational or transcendent functions as it is shown for instance

[1] The subscripts l and m of α_{lm} indicate the order of continuity of b, k_1 and b, k_2 at the respective points A_1 and B_2.

in [5], [10] and [11]. In [5], [10], and [12] blending functions are also used to blend two surfaces to a third one. Szilvási [11] even applies a special variant of the Coons' method in order to blend four surfaces to a fifth one.

Remark 2 In (4) we see the input data of a Hermite interpolation problem to the real parameter values $t_0 = 0 < t_1 = 1$ (see [6, p. 15], [7, pp. 4–11]). Therefore we know that there exists exactly one polynomial function $\alpha_{lm}(t)$ with polynomial degree

$$\deg\ \alpha_{lm}(t) \leq l + m + 1$$

satisfying (4).

Although there exists a Hermite interpolation formula to determine $\alpha_{lm}(t)$ (see [1]) we will use the Bernstein-polynomials

$$\left.\begin{array}{c} B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i \\ i, n \in \mathbb{N}_0,\ n \geq i,\ t \in \mathbb{R},\ B_0^0 := 1 \end{array}\right\} \ldots \text{ Bernstein-polynomials} \tag{5}$$

The Bernstein-polynomials form a basis of the vector space of all polynomials of order $\leq n$. Hence we can write $\alpha_{lm}(t)$ in the form

$$\alpha_{lm}(t) = \sum_{i=0}^{l+m+1} \lambda_i \cdot B_i^{l+m+1}(t), \quad \lambda_i \in \mathbb{R}. \tag{6}$$

Due to the well known properties of the Bernstein-polynomials and their derivatives at $t = 0$ and $t = 1$ we obtain by straight forward computation

$$\begin{aligned} \lambda_0 = \ \ldots\ &= \lambda_l = 1, \\ \lambda_{l+1} = \ \ldots\ &= \lambda_{l+m+1} = 0. \end{aligned}$$

Hence the unique polynomial that solves (4) can be written in the form

$$\alpha_{lm}(t) = \sum_{i=0}^{l} B_i^{l+m+1}(t) \tag{7}$$

We summarize in

Theorem 1 *Given two parametric curves k_1, k_2 with*

$$k_1(t) \neq k_2(t),\ t \in \mathbb{R},\ k_1 \in C^{n_1},\ k_2 \in C^{n_2},\ n_1, n_2 \in \mathbb{N}_0$$

and

$$k_1(0) = A_1,\ k_1(1) = A_2, \quad k_2(0) = B_1,\ k_2(1) = B_2, \quad A_1 \neq B_1,\ A_2 \neq B_2.$$

Then there exists exactly one polynomial function $\alpha_{lm}(t)$ with $\deg \alpha_{lm}(t) \leq l+m+1$, so that for the curve

$$b(t) = \alpha_{lm}(t) \cdot k_1(t) + [1 - \alpha_{lm}(t)] \cdot k_2(t)$$

the conditions (1) hold.

$\alpha_{lm}(t)$ can be written in the form

$$\alpha_{lm}(t) = \sum_{i=0}^{l} B_i^{l+m+1}(t), \quad 0 \leq l \leq n_1, \quad 0 \leq m \leq n_2.$$

For the special case that the two curves k_1 and k_2 have common points $A_1 = B_1$ and $A_2 = B_2$ and common derivatives at this points we state the following theorem:

Theorem 2 *Given two parametric curves k_1, k_2*

$$k_1(t) \neq k_2(t),\ t \in \mathbb{R},\ k_1 \in C^{n_1}, \quad k_2 \in C^{n_2},\ n_1, n_2 \in \mathbb{N}_0$$

with

$$k_1(0) = A_1 = B_1 = k_2(0) \ \wedge\ k_1(1) = A_2 = B_2 = k_2(1)$$

and

$$\begin{aligned} \frac{d^i}{(dt)^i} k_1(0) = \frac{d^i}{(dt)^i} k_2(0), &\quad 1 \leq i \leq l_1 \ \textit{ and } \ 1 \leq l_1 < l \leq n_1 \\ \frac{d^i}{(dt)^i} k_1(1) = \frac{d^i}{(dt)^i} k_2(1), &\quad 1 \leq i \leq m_1 \ \textit{ and } \ 1 \leq m_1 < m \leq n_2 \end{aligned} \tag{8}$$

Then there exists exactly one polynomial function

$$\alpha_{\lambda\mu}(t) := \alpha_{l-l_1-1,m-m_1-1}(t) \ \textit{ and } \ \deg \alpha_{\lambda\mu}(t) \leq l - l_1 + m - m_1 - 1 = \lambda + \mu + 1,$$

so that for the curve

$$b(t) \ = \ \alpha_{\lambda\mu}(t) \cdot k_1(t) + [1 - \alpha_{\lambda\mu}(t)] \cdot k_2(t)$$

the conditions (1) hold. $\alpha_{\lambda\mu}(t)$ can be written in the form

$$\alpha_{\lambda\mu}(t) = \sum_{i=0}^{\lambda} B_i^{\lambda+\mu+1}(t), \quad \lambda = l - l_1 - 1, \quad \mu = m - m_1 - 1. \tag{9}$$

Proof: If we compare (8) with (3) we find the following conditions for $\alpha_{\lambda\mu}(t)$:

$$\begin{aligned} \alpha_{\lambda\mu}(0) &= 1, \\ \alpha_{\lambda\mu}(1) &= 0, \\ \frac{d^j}{(dt)^j}\alpha_{\lambda\mu}(0) &= 0, \quad 1 \leq j \leq l - l_1 - 1, \\ \frac{d^j}{(dt)^j}\alpha_{\lambda\mu}(1) &= 0, \quad 1 \leq j \leq m - m_1 - 1. \end{aligned} \tag{10}$$

The conditions (10) are again the input data of a Hermite interpolation problem. So we know that there exists exactly one polynomial function with

$$\deg \ \alpha_{\lambda\mu}(t) \leq l - l_1 + m - m_1 - 1 = \lambda + \mu + 1$$

which solves our problem. Analogously to Theorem 1 the polynomial $\alpha_{\lambda\mu}(t)$ can be expressed in the form (9) with the help of the Bernstein polynomials. □

Remark 3 With the linear parameter transformation

$$t(\tau) = \frac{\tau - \tau_0}{\tau_1 - \tau_0}, \quad \tau_0 \leq \tau \leq \tau_1$$

one can adapt an arbitrary parameter interval $[\tau_0, \tau_1]$ to $[0, 1]$.

Remark 4 The curve $b(t)$ lies within the convex hull of the the two curve segments $k_1(t)$, $k_2(t), t \in [t_0, t_1]$ and its construction is affinely invariant.

Remark 5 In the neighborhood of the points $b(0)$ and $b(1)$ the curve b sticks closely to k_1 and k_2, respectively. So you can imagine its shape before constructing the curve $b(t)$.

Remark 6 One can use line segments as tangents and circles of curvature at the points A_1 and B_2 to generate C^1- and C^2-continuous blending curves. Fig. 2 shows our blending construction between two line segments and a circle. The blending function a_{lm} must be of order $l + m + 1 = 1 + 2 + 1 = 4$. So the blending curves b_1, b_2 are of order six.

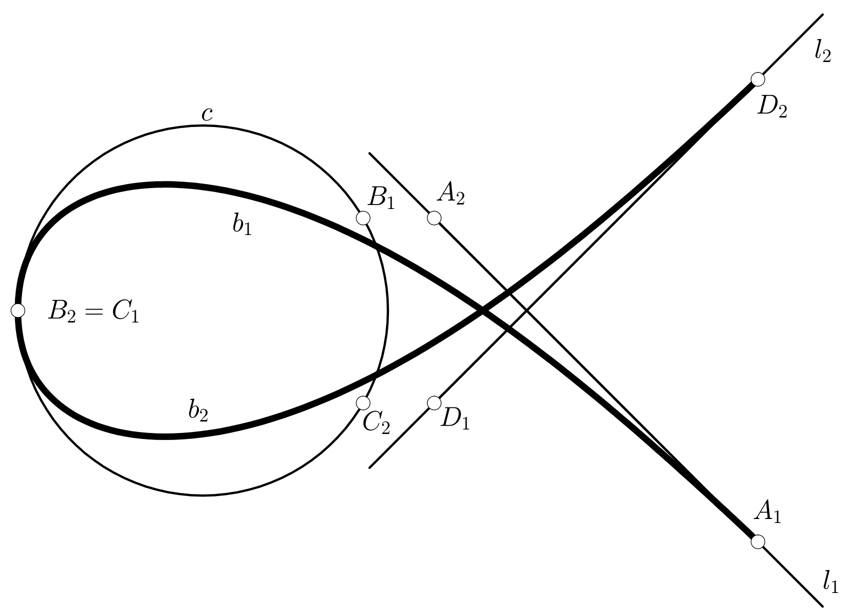

Figure 2: Two blending curves b_1, b_2 joining a circle c with the line segments l_1, l_2. The curves b_1, b_2 are C^1-continuous at A_1 and D_2 and C^2-continuous at $B_2 = C_1$.

Remark 7 With our construction we can also build blending curves which connect space curves.

Remark 8 The advantage of using polynomial blending functions α_{lm} is that the blending curve b is polynomial if the curves k_1 and k_2 are so. A disadvantage of the uniquely defined blending functions α_{lm} is the lack of design parameters.

If we want to have more scope of design in the construction of blending curves we have to use polynomial functions with degree $> l + m + 1$. If we prescribe a degree $l + m + 1 + d$ then a simple consideration shows that every possible polynomial function $\bar{\alpha}_{lm}$ satisfying (4) can be written in the form

$$\bar{\alpha}_{lm}(t) = \sum_{i=0}^{l} B_i^n(t) + \sum_{i=1}^{d} \lambda_i \cdot B_{l+i}^n(t), \tag{11}$$
$$n = l + m + 1 + d, \quad d \in \mathbb{N}, \quad \lambda_i \in \mathbb{R}.$$

This endows the user with d design parameters $\lambda_1, \ldots, \lambda_d$. The blending curve b can now be described by

$$\begin{aligned} b(t) &= \bar{\alpha}_{lm}(t)k_1 + [1-\bar{\alpha}_{lm}(t)]\cdot k_2(t) = \\ &= \sum_{i=0}^{l} B_i^n \cdot k_1(t) + \sum_{i=1}^{d} B_{l+i}^n \cdot [\lambda_i \cdot k_1(t) + (1-\lambda_i)\cdot k_2(t)] + \sum_{i=l+d+1}^{n} B_i^n \cdot k_2(t) \end{aligned} \quad (12)$$

Remark 9 In order to guarantee that the blending curve b lies within the convex hull of the two curve segments $k_1(t), k_2(t)$, $t \in [t_0, t_1]$, one has to choose $0 \leq \lambda_i \leq 1$.

Fig. 3 shows the uniquely defined cubic blending function $\alpha_{11}(t) = 2t^3 - 3t^2 + 1$ and two quartic blending functions for C^1-continuously blending at $t = 0$ and $t = 1$. The quartic functions $\bar{\alpha}_{lm}$ depend on one design parameter λ: $\bar{\alpha}_{11}(t) = B_0^4(t) + B_1^4(t) + \lambda \cdot B_2^4(t)$. In the figure we set $\lambda = 0.2$ and $\lambda = 0.9$.

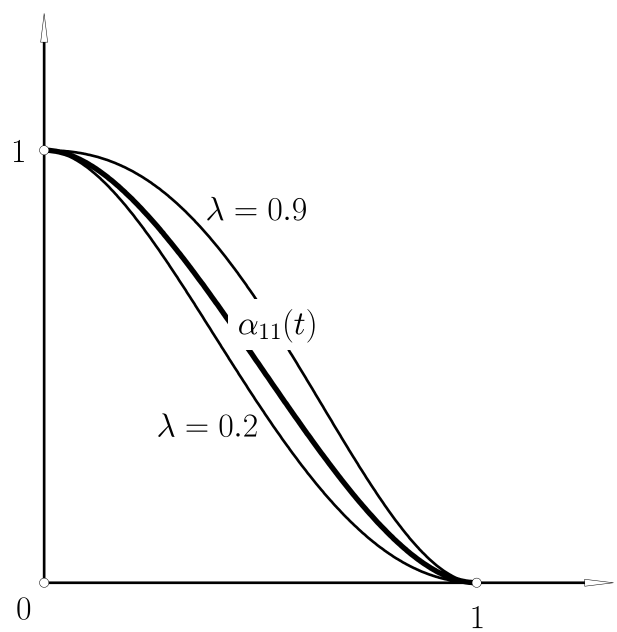

Figure 3: The uniquely defined blending function $\alpha_{11}(t)$ and two quartic blending functions with the design parameters $\lambda = 0.2$ and $\lambda = 0.9$ for C^1-continuous blending at $t = 0$ and $t = 1$

3. Algorithm

1. If we consider the representations of the uniquely defined function $\alpha_{lm}(t)$ in eq. (7) and the blending curve b in eq. (2) we see that we can evaluate the curve point $b(t)$ at one fixed parameter value t with the help of the *Casteljau*-algorithm (see for instance [4] or [9]). As shown in Fig. 4 we construct the point $b(t)$ as the curve point of a $(l+m+1)^{\text{th}}$-order Bézier curve to the Bézier points

$$k_1(t) = P_0 = \ldots = P_l \quad \text{and} \quad P_{l+1} = \ \ldots = P_{l+m+1} = k_2(t)$$

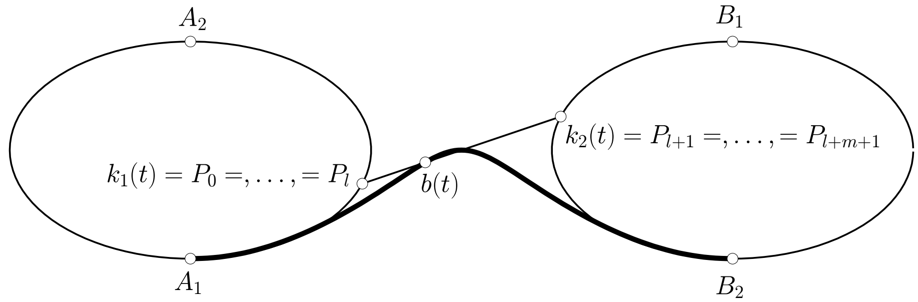

Figure 4: The construction of a curve point $b(t)$ for a fixed parameter value t with the help of the Casteljau-algorithm

2. If we want to use design parameters for the construction of our blending curve b we can apply the Casteljau-algorithm again (see eq. (12)). The curve point $b(t)$ for one fixed parameter value t can be evaluated as the curve point of a $(l+m+1+d)^{\text{th}}$-order Bézier curve to the Bézier points

$$\begin{aligned} P_0 = \ldots = P_l &= k_1(t), \\ P_{l+i} &= \lambda_i \cdot k_1(t) + [1 - \lambda_i] \cdot k_2(t), \quad \lambda_i \in \mathbb{R},\ i = 1, \ldots, d, \\ P_{l+d+1} = \ldots = P_{l+m+1+d} &= k_2(t) \end{aligned}$$

If we choose $0 \leq \lambda_i \leq 1$ then the Bézier point P_{l+i} lies in between the straight line segment $\overline{P_1 P_{l+m+1+d}}$.

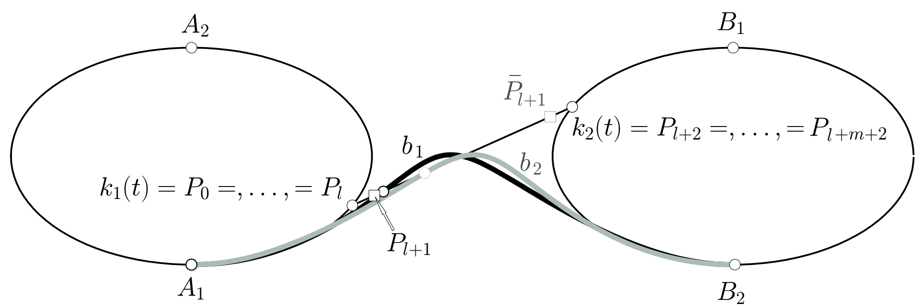

Figure 5: Two alternative curves b_1, b_2 to the blending curve b of Fig. 4 ($\deg b_i = \deg b + 1$, b_2 is drawn grey). The additional design parameter is $\lambda = 0.9$ for b_1 and $\lambda = 0.1$ for b_2. The corresponding Bézier points $P_{l+1}, \bar{P}_{l+1}$ are marked by quadrangles

4. Examples

We can use our construction and Theorem 2 to build spline curves which interpolate given points $P_0, \ldots, P_k$ at corresponding parameter values $t_0, \ldots, t_k$.

1. A well known interpolant is the Overhauser spline (see [2, 3, 4, 8]). It uses parabolas k_i and k_{i+1} which interpolate the point triples $(P_{i-1},\ P_i,\ P_{i+1})$ and $(P_i,\ P_{i+1},\ P_{i+2})$, respectively. Then the curves k_i and k_{i+1} are blended by the linear function $\alpha_{00}(t) = \dfrac{t - t_i}{t_{i+1} - t_i}$ to b_i (see Fig. 6).
2. With the help of the polynomial $\alpha_{11}(t)$ we can improve Example 1 to build a C^2-continuous interpolating spline curve of order 5. Our spline curve depends only on a

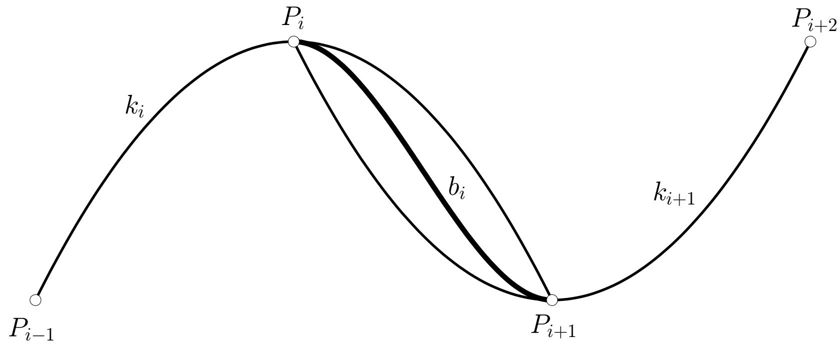

Figure 6: The C^1-continuous Overhauser spline

few points. Changing one point P_i influences only the spline curve between P_{i-2} to P_{i+2} (see Fig. 7).

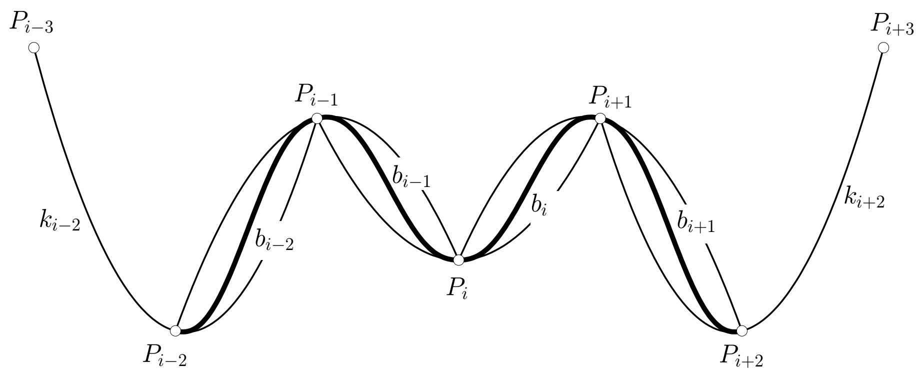

Figure 7: Parabolas blended to a C^2-continuous spline

5. Conclusion

In this paper we have shown the construction of polynomial functions $\alpha_{lm}(t)$ aimed to blend two arbitrarily given parametric curves $k_1(t)$ and $k_2(t)$ to a third curve $b(t)$. It turned out that the function $\alpha_{lm}(t)$ can be described in a very simple form by means of Bernstein-polynomials. This fact enables us to use the de Casteljau-algorithm to generate the points of the blending curve $b(t)$. Furthermore we can use our construction to build interpolating spline curves.

References

[1] I. Beresin, N. Shidkow: *Numerische Methoden 1.* Hochschulbücher für Mathematik, Bd. 70, VEB Deutscher Verlag der Wissenschaften, Berlin 1970.

[2] J. Brewer, D. Anderson: *Visual interaction with Overhauser curves and surfaces.* Computer Graphics **11**(2), 132–137 (1977).

[3] W. Degen: *The Shape of the Overhauser Spline.* Computing Suppl. **10**, 117–128 (1995).

[4] G. Farin: *Curves and Surfaces for CAGD.* 5^{th} ed., Morgan Kaufmann Publishers, San Francisco 2001.

[5] E. Hartmann: *Parametric G^n blending of curves and surfaces.* Visual Computer **17**, 1–13 (2001).

[6] R.J.Y. McLeod, M.L. Baart: *Geometry and interpolation of curves and surfaces.* University Press, Cambridge 1998.

[7] G. Nuernberger: *Approximation by Spline Functions.* Springer Verlag, 1989.

[8] A. Overhauser: *Analytic definition of curves and surfaces by parabolic blending.* Technical Report, Ford Motor Company, 1968.

[9] H. Prautzsch, W. Boehm, M. Paluszny: *Bézier and B-Spline Techniques.* Springer Verlag, 2002.

[10] M. Szilvásy-Nagy, T.P. Vendel: *Generating curves and swept surfaces by blended circles.* Comput.-Aided Geom. Design **17**, 197–206 (2000).

[11] M. Szilvásy-Nagy: *Construction of flexible blending surfaces.* Third Croatian Congress of Mathematics, preprint, Split 2004.

[12] A. Wiltsche: *A polynomial tool for blending surfaces.* Grazer Mathematische Berichte, to appear (2005).

Received October 30, 2004; final form May 2, 2005