An Interpolation Subspline Scheme Related to B-Spline Techniques

O. Röschel, Institute of Geometry, TU Graz, Austria
e-mail: roeschel@geometrie.tu-graz.ac.at

Abstract

We construct (integral) interpolationg subspline curves for given data points and knot vector. The algorithm is very close to B-spline approximation. The idea is to blend interpolating Lagrangian splines by B-spline techniques. All is connected in affinely invariant way with the control points and the knot vector. We are able to show, that our scheme produces high quality subsplines, which include known procedures like Overhauser or quintic interpolation schemes. In addition we may sweep to B-splines and return in a very lucid way. Examples show the power of the method. The given procedure allows generalisations to rational subsplines and to tensor product interpolating surfaces.

0. Introduction

The paper deals with the following interpolation problem: $n+1$ data points $\{\vec{p}_0, \ldots \vec{p}_n\}$ and knots $u_0 < u_1 < \ldots < u_n$ be given. Determine a subspline $\vec{k}(t)$ $(t \in [u_0, u_n])$ of class C^m with $\vec{k}(u_i) = \vec{p}_i$ for all $i = 0, \ldots, n$. Although solutions are well-known, we will add a new idea to construct such subsplines in a way related to B-spline techniques.

In the last years many new attempts have been made to join the well-known algorithms of freeform curves (B-Splines...) and interpolation schemes. In [1] recently socalled X-splines of class C^2 have been considered, which consist of (non integral) rational segments of degree 5. These (sub)splines allow interpolation and approximation as well. [1] have shown, that these X-splines are close to (but not identical with) B-splines.

The idea of the following is to use Lagrangian interpolation for certain sets of the data points $\{\vec{p}_i, \ldots \vec{p}_{i+k}\}$ and corresponding knots. The resulting curves denoted by $\vec{l}_i(t)$ then are blended by normalized B-Spline basic functions $N_{i,k}(t)$ of degree k-1 and give the interpolant $\vec{k}(t)$. As $\vec{l}_i(u_m) = \vec{p}_m$ for $i = m-k, \ldots, m$, for the corresponding subspline we have $\vec{k}(u_m) = \vec{p}_m$. This idea is close to the paper [3] - there Bézier and Lagrange techniques are linked.

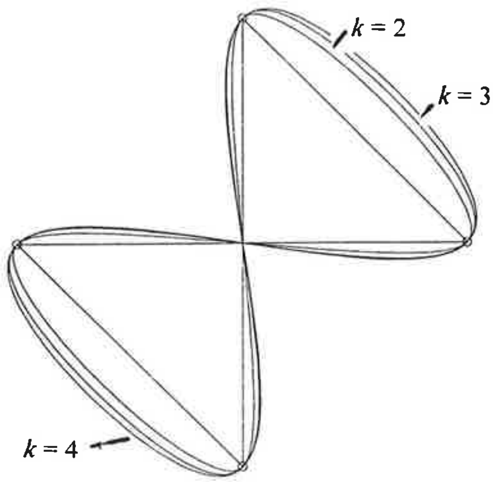

Figure 1: Closed interpolants for the cases $k = 2, 3, 4$

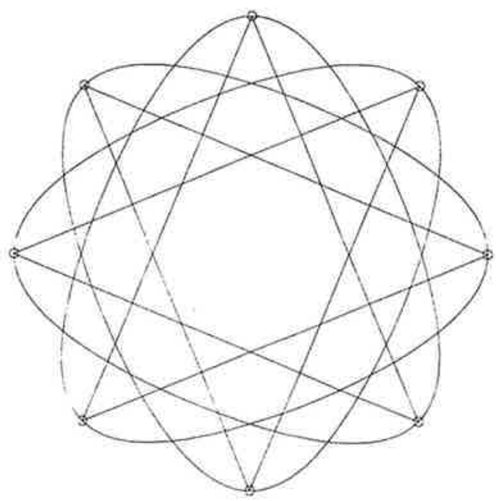

Figure 2: Closed interpolant for $k = 4$

We will be able to proof, that this subspline in general is of class C^{k-1}. The segments are integral rational curves of degree 2*k*-1. Furthermore, our subspline is connected in affinely invariant way with the set of points $\{\vec{p}_0, \dots \vec{p}_n\}$, if the knot vector $(u_0, \dots, u_n)$ does not change. It is easily possible to sweep to and from B-splines.

1. Lagrangian interpolants

In affine space $A_d(R)$ a set of points $\{\vec{p}_0, \dots \vec{p}_n\}$ and knots $(u_0 < u_1 < \dots < u_n)$ are given. We fix an integer *k*. Then we are able to compute the unique Lagrangian interpolant $\vec{l}_i(t)$ of degree *k*, which interpolates the points $\{\vec{p}_i, \dots \vec{p}_{i+k}\}$ at parameters $(u_i, \dots, u_{i+k})$. We determine these interpolants for *i*=0, ..., *n-k*. With the Lagrangian polynomials of degree *k*

$$(1.1) \quad L_j^k(t) := [\prod_{\substack{i=0,\dots,k \\ i \neq j}} (t - u_i)] / [\prod_{\substack{i=0,\dots,k \\ i \neq j}} (u_j - u_i)]$$

for $j = 0, \dots, k$ we define

$$(1.2) \quad \vec{l}_i(t) = \sum_{j=0}^{k} L_{i+j}^k(t)\, \vec{p}_{i+j} \quad \text{for } i = 0, \dots, n-k$$

and for $i := m-k, \dots, m$ we have

$$(1.3) \quad \vec{l}_i(u_m) = \vec{p}_m .$$

2. Definition of the interpolant

We want to blend these interpolant curves (1.2) by B-spline techniques. As usual we define normalized B-spline basic functions via the recursion formula: For $i = 0, \dots, n+k-1$:

$$(2.1) \quad 1)\ N_{i,1}(t) := \begin{cases} 1 & \text{for } t \in [u_i, u_{i+1}) \\ 0 & \text{else} \end{cases}$$

$$2)\ N_{i,j}(t) := \frac{t - u_i}{u_{i+j-1} - u_i} N_{i,j-1}(t) + \frac{u_{i+j} - t}{u_{i+j} - u_{i+1}} N_{i+1,j-1}(t)$$

for $j = 1, \dots, k$ with the additional definitions $\frac{0}{0} := 0$ and $N_{n-k,1}(u_{n-k+1}) := 1$. These notations follow [6]. Now we define a (sub)spline curve by

$$(2.2) \quad \vec{k}(t) := \sum_{i=0}^{n-k} \vec{l}_i(t)\, N_{i,k}(t)$$

for $t \in [u_{k-1}, u_{n-k+1}]$. Like in the case of B-spline curves for $t \in [u_m, u_{m+1})$ we have

$$\vec{k}(t) := \sum_{i=m-k+1}^{m} \vec{l}_i(t)\, N_{i,k}(t) .$$

3. Properties of the subspline curve (2.2)

a) For $m = k-1, \dots, n-k$ we have:

$$\vec{k}(u_m) := \sum_{i=0}^{n-k} \vec{l}_i(u_m)\, N_{i,k}(u_m) =$$

$$= \sum_{i=m-k+1}^{m} \vec{l}_i(u_m)\, N_{i,k}(u_m) = |\text{use (1.3)}| =$$

$$= \sum_{i=m-k+1}^{m} \vec{p}_m\, N_{i,k}(u_m) = |\textit{partition of unity}| =$$

$$= \vec{p}_m .$$

Thus our subspline $\vec{k}(t)$ (2.2) with (1.2) **interpolates the points** $\{\vec{p}_{k-1}, \dots, \vec{p}_{n-k}\}$ **at the parameters** $(u_{k-1}, \dots, u_{n-k})$.

b) Now we state the following

Theorem 1: *The subspline* $\vec{k}(t)$ $(t \in [u_{k-1}, u_{n-k+1}])$ (2.2) *with* (1.3) *is of class* C^{k-1}.

Remarks: 1) The class C^{k-1} is surprising. If we put $\vec{l}_i(t) := \vec{p}_i$ for all *i*=0, ...,*n* we gain a B-spline curve with control points $\{\vec{p}_0, \dots \vec{p}_n\}$ (of course the curve does not interpolate the given points any more). This B-spline curve is of class C^{k-2} at our values $(u_k, \dots, u_{n-k-1})$.

2) The segments of our subspline are integral rational curves of degree *k*+*k*-1 = 2 *k* - 1. Thus curves with even degree are not generated by our procedure.

Proof of theorem 1:

The only problems of discontinuity may occour at the values $t = u_m$ ($m = k, \dots, n-k-1$), where segments of the subspline are linked. We use well-known properties of B-spline basic functions of degree *k*-1 (see [6]):

We consider two adjacent segments of the subspline belonging to values $t \in [u_{m-1}, u_m)$ and $t \in [u_m, u_{m+1})$. We compute basic functions for the interval

$t \in [u_{m-1}, u_m)$: The corresponding interpolating curve segment then is given by

$$(3.1) \qquad \vec{k}(t) := \sum_{i=0}^{n-k} \vec{l}_i(t)\; N_{i,k}(t)\,.$$

We denote basic functions for the adjacent interval $t \in [u_m, u_{m+1})$ by $N^*_{m-i,j}(t)$ - the corresponding curve segment then is represented by

$$(3.2) \qquad \vec{k}^*(t) := \sum_{i=0}^{n-k} \vec{l}_i(t)\; N^*_{i,k}(t)\,.$$

Both curve segments are rationally parametrized by t. At $t = u_m$ we may determine derivatives of (3.1) and (3.2) up to order k-1. Using chain rule we gain

$$(3.3) \qquad \left.\frac{d^j \vec{k}}{(dt)^j}\right|_{u_m} = \sum_{\nu=0}^{j} A(\nu,j) \sum_{i=0}^{n-k} \left.\frac{d^{j-\nu}\vec{l}_i}{(dt)^{j-\nu}}\right|_{u_m} \left.\frac{d^\nu N_{i,k}}{(dt)^\nu}\right|_{u_m}$$

and

$$(3.4) \qquad \left.\frac{d^j \vec{k}^*}{(dt)^j}\right|_{u_m} = \sum_{\nu=0}^{j} A(\nu,j) \sum_{i=0}^{n-k} \left.\frac{d^{j-\nu}\vec{l}_i}{(dt)^{j-\nu}}\right|_{u_m} \left.\frac{d^\nu N^*_{i,k}}{(dt)^\nu}\right|_{u_m}$$

for j = 0, ..., k-1. There $A(\nu, j)$ denote integers not depending on t or the knot vector. For example we have $A(0,j) = A(j,j) = 1$.

We know, that B-spline curves are of class C^{k-2} at $t = u_m$. Therefore we have

$$(3.5) \qquad \left.\frac{d^\nu N_{i,k}}{(dt)^\nu}\right|_{u_m} = \left.\frac{d^\nu N^*_{i,k}}{(dt)^\nu}\right|_{u_m}$$

for all $\nu = 0,\ldots,k-2$ and i = 0, ..., n-k. Thus

$$(3.6) \qquad \left.\frac{d^j \vec{k}}{(dt)^j}\right|_{u_m} = \left.\frac{d^j \vec{k}^*}{(dt)^j}\right|_{u_m} \qquad \text{for } j = 0, \ldots, k\text{-}2.$$

But for our subspline we gain more: We have

$$\left.\frac{d^{k-1}\vec{k}}{(dt)^{k-1}}\right|_{u_m} =$$

$$(3.7) \qquad = \sum_{\nu=0}^{k-2} A(\nu,k-1) \sum_{i=0}^{n-k} \left.\frac{d^{k-1-\nu}\vec{l}_i}{(dt)^{k-1-\nu}}\right|_{u_m} \left.\frac{d^\nu N_{i,k}}{(dt)^\nu}\right|_{u_m} + \sum_{i=0}^{n-k} \vec{l}_i(u_m) \left.\frac{d^{k-1} N_{i,k}}{(dt)^{k-1}}\right|_{u_m}.$$

(3.5) shows, that the first sum in (3.7) is the same as for $\left.\frac{d^{k-1}\vec{k}^*}{(dt)^{k-1}}\right|_{u_m}$. Now we discuss the remaining term $\sum_{i=0}^{n-k} \vec{l}_i(u_m) \left.\frac{d^\nu N_{i,k}}{(dt)^\nu}\right|_{u_m}$ of (3.7): If we put $\vec{b}_i := \vec{l}_i(u_m)$, this term can be interpreted as the (k-1)-th derivative of the B-spline curve of degree k-1 belonging to the control points $\{\vec{b}_i := \vec{l}_i(u_m),\ i = 0,\ldots,n-k\}$ and the given knot vector $(u_0, \ldots, u_n)$. These derivatives may be determined via the following recursion formula (see [6], p. 170):

$$\sum_{i=0}^{n-k} \vec{l}_i(u_m) \left.\frac{d^\nu N_{i,k}}{(dt)^\nu}\right|_{u_m} = (k-1)! \sum_{i=0}^{n-k} \vec{c}[i,k-1]\; N_{i,1}(u_m)$$

with

$$(3.8) \qquad \vec{c}[i,0] := \vec{l}_i(u_m), \qquad i = 0,\ldots,n-k \qquad \text{and}$$
$$\vec{c}[i,j] = \frac{\vec{c}[i,j-1] - \vec{c}[i-1,j-1]}{u_{i+k-j} - u_i} \qquad \text{for } j \geq 1.$$

We know, that we have $N_{m-1,1}(u_m) = 1$ - all other $N_{i,1}(u_m)$ are zero. Therefore we may write

$$(3.9) \qquad \sum_{i=0}^{n-k} \vec{l}_i(u_m) \left.\frac{d^\nu N_{i,k}}{(dt)^\nu}\right|_{u_m} = (k-1)!\; \vec{c}[m-1,k-1]\,.$$

(3.8) shows, that we need $\vec{c}[m-k,0],\ldots,\vec{c}[m-1,0]$ in order to determine the only interesting vector $\vec{c}[m-1,k-1]$. For all these vectors we have $\vec{c}[m-k,0] = \ldots = \vec{c}[m-1,0] = \vec{p}_{m-1}$ (see (1.3)). The recursion formula (3.8) then produces $\vec{c}[m-1,k-1] = \vec{o}$. The second term in the sum (3.7) therefore vanishes. An analogous discussion may be done for the second curve. Therefore we have

$$(3.10) \qquad \left.\frac{d^{k-1}\vec{k}}{(dt)^{k-1}}\right|_{u_m} = \left.\frac{d^{k-1}\vec{k}^*}{(dt)^{k-1}}\right|_{u_m}.$$

Together with (3.6) this finsihes the proof of theorem 1. □

Therefore for the case k = 2 we gain a subspline of class C^1. Comparing with [2] we see that we have got the Overhauser spline scheme. In the case k = 3 we get a subspline of class C^2 - its segments are quintic integral curves. Our method of generation therefore gives the well-known quintic C^2 spline interpolants (see [9] and [6]). In general one may follow well-known algorithms to generate interpolating subsplines of class C^{k-1} (see [6]). There exist solutions of this problem with lower degree segments than that represented here. **But with our method we do not have to solve systems of equations to compute the spline!** Like we will see in chapter 4 our procedure is a more constructive one. Some of our examples shall illustrate the case k = 4. There we have class C^3 of the spline scheme. Figure 1 illustrates interpolation curves with increasing k = 2, 3, 4. The continuity then is of class C^1, C^2 and C^3, resp. The curve becomes "rounder" if k increaeses.

4. An algorithm to generate points on the subspline

<u>**Algorithm to generate points on the subspline (2.2)**</u>

Input: Data points $\{\vec{p}_0,\dots \vec{p}_n\}$ and knots $(u_0 < u_1 < \dots < u_n)$, fixed integer k, parameter value $t \in [u_{k-1}, u_{n-k}]$.

<u>**Output:**</u> <u>Point $\vec{k}(t)$ on the interpolating subspline (2.2)</u>

1) Determine m with $t \in [u_m, u_{m+1})$. If $t = u_{n-k}$ then m:=n-k-1.

2) **FOR** $\nu := 0$ **TO** k-1 **DO**

BEGIN (* Aitken's algorithm *)

FOR i:=0 **TO** k **DO** $\vec{q}_i := \vec{p}_{m-\nu+i}$

FOR j:= 0 **TO** k **DO**

FOR i:=0 **TO** k-j **DO**

$$\vec{q}_i := \frac{u_{m-\nu+i+j} - t}{u_{m-\nu+i+j} - u_{m-\nu+i}}\vec{q}_i + \frac{t - u_{m-\nu+i}}{u_{m-\nu+i+j} - u_{m-\nu+i}}\vec{q}_{i+1}$$

$\vec{r}_\nu := \vec{q}_0$

END.

3) **FOR** j:= 1 **TO** k-1 **DO**(* Algorithm of Cox- DeBoor for B-splines *)

FOR i:=0 **TO** k-j-1 **DO**

$$\vec{r}_i := \frac{t - u_{m-i}}{u_{m+k-i-j} - u_{m-i}}\vec{r}_i + \frac{u_{m+k-i-j} - t}{u_{m+k-i-j} - u_{m-i}}\vec{r}_{i+1}$$

4) $\vec{k}(t) := \vec{r}_0$.

It is easy to give such an algorithm. We just have to combine an algorithm producing Lagrangian interpolants (we use an algorithm presented in [4], pp. 67, called Aitken's algorithm) with Cox-DeBoor algorithm for B-spline curves. In a memory optimized version we get the boxed algorithm presented above.

Both algorithms (Aitken and Cox-DeBoor) are affinely invariant. Therefore we have

Theorem 2: *The subspline (2.2) with (1.3) is connected in affinely invariant way with the data points* $\{\vec{p}_0,\dots \vec{p}_n\}$ *and the knots* $(u_0 < u_1 < \dots < u_n)$. *The subspline interpolates data points* $\{\vec{p}_{k-1},\dots,\vec{p}_{n-k}\}$ *at values* $u_{k-1},\dots,u_{n-k}$.

5. Remarks and Examples

a) Like in the case of B-splines we may generate open or closed versions of these subsplines.

<u>Closed interpolation</u>: In this case we just have to go in the round to gain further input data (as usual in case of B-splines).

<u>Open interpolation</u>: Like in the case of B-splines we may count the first and the last interpolation point k-1 times and add knots to the knot vector.

Figures 3 and 4 show some examples for k = 4 with chord lenght parametrisation for the knot vector. In figure 3 the version following chapter 4 (without multiple choice of control points at the ends) is shown. In figure 4 an open interpolant are drawn. Figures 1 and 2 show closed interpolants with uniform knot vector.

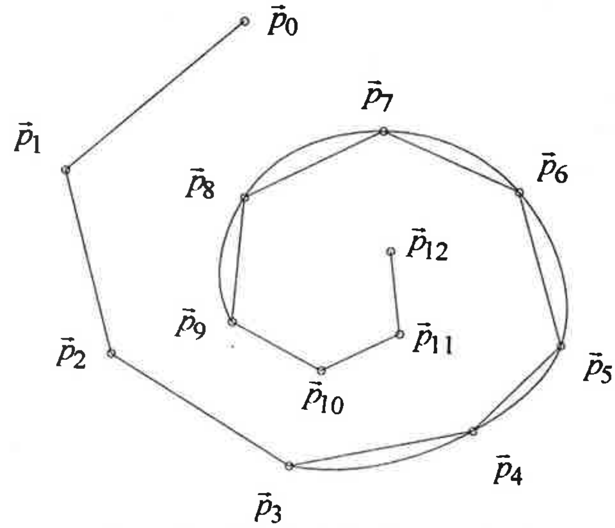

Figure 3: Interpolants for the case $k = 4$.

b) **We may sweep to B-spline curves of degree k-1:** This may be done by putting $\vec{l}_i(t) := \vec{p}_{i+k-1}$ for certain $i := m, \dots, m^*$. An index shift is necessary: it helps to avoid undesired loops of the subspline. The resulting subspline curve then is of class C^{k-2}. Figure 5 shows the situation for the case k = 4. There the whole B-spline and a "sweep curve" are shown. The first part of the sweep curve consists of integral segments of degree 7 (interpolation), the second of B-spline cubics.

If we want to sweep from B-splines belonging to basic points $\{\vec{p}_0,\dots,\vec{p}_m\}$ to our interpolants for points $\{\vec{p}_{m+1},\dots,\vec{p}_n\}$ we put $\vec{l}_i(t) := \vec{p}_i$ for i := 0, ..., m-k. For

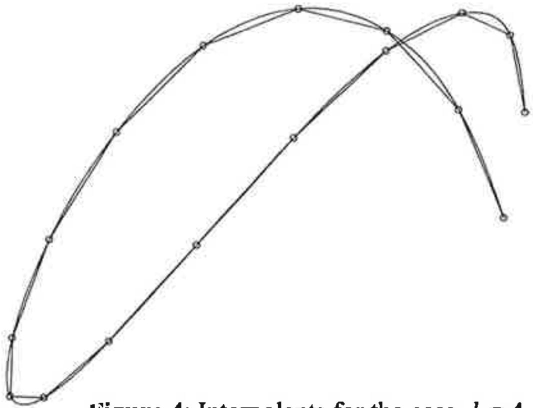

Figure 4: Interpolants for the case $k = 4$.

the following $\vec{l}_i(t)$ $(i > m-k)$ we proceed as given in 2. Figure 5 gives an example.

In both cases the whole curve then is of class C^{k-2} - the same class as we have for our B-splines.

c) All may be done in a rational version of these curves. The influence of the weights of the interpolated points shall be discussed in a further paper.

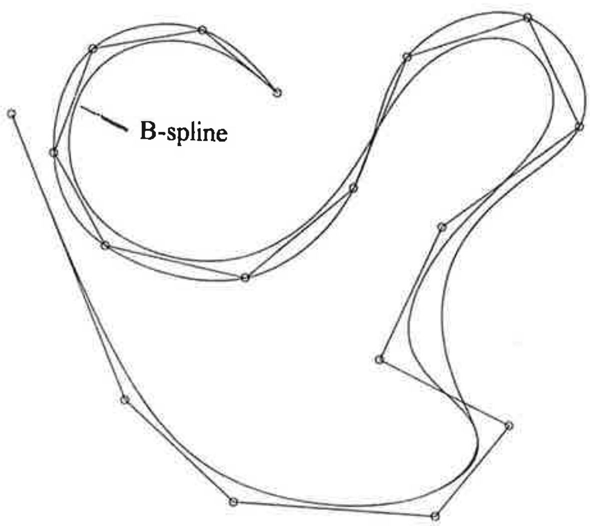

Figure 5: Sweeping of interpolants and B-spline curves for the case $k = 4$.

6. Interpolating subspline surfaces.

If a two-dimensional (rectangular) array of data points and the corresponding array of knots are given, our procedure may be used to generate tensor product interpolants. This is an extension of tensor product B-spline surfaces. The resulting surface patch is of class C^{k-1}. Again our interpolants may be swept to B-spline surfaces like we have seen in the case of curves. Then continuity is reduced to class C^{k-2}. But this shall be demonstrated in a separate paper.

7. Conclusions

The paper has shown the construction of an interpolating subspline scheme, which is connected in affine invariant way with data points and knots. It is gained by combining Lagrangian interpolation with B-Spline techniques. It may be implemented in a very lucid way by combining two well-known algorithms used for solving interpolation and approximation problems (algorithms of Aitken and Cox-DeBoor). We gained integral segments of a subspline scheme, which may be blended to B-spline curves. We generated a powerful tool, which allows interpolation and B-spline approximation at the same time. We do not need to solve any equation to gain theses interpolants. The price we had to pay for it is a high degree of the segments of our curves. But the possibility to use well-known algorithms to generate the interpolant justifies the use of this new method, if the degree of the interpolants is no problem. Rational versions and extensions to tensorproduct schemes can be handled by similar algorithms.

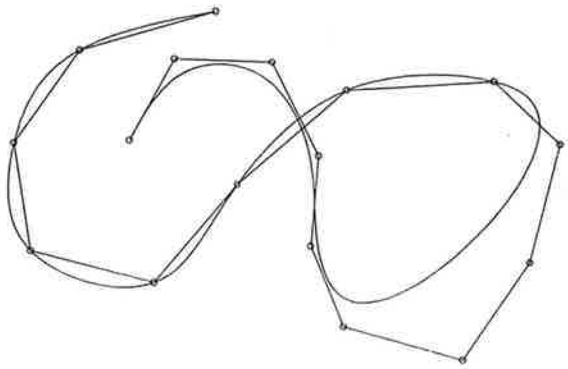

Figure 6: Further Examples of interpolating subsplines for $k = 4$.

References

[1] Blanc, C. - Schlick, Ch.: X-Splines: A Spline Model Designed for the End-User. Proc. SIGGRAPH 95 Conf., 377 - 386 (1995).

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

[3] Farin, G. - Barry, P.J.: Link between Bézier and Lagrange curve and surface schemes. Computer Aided Design 18, 525 - 528 (1986).

[4] Farin, G.: Curves and Surfaces for Computer Aided Geometric Design. 2nd. edn., Academic Press, Boston 1990.

[5] Hagen, H.: Geometric Spline Curves. Comp. Aid. Geom. Design 2, 223 - 227 (1985).

[6] Hoschek, J. - Lasser, D.: Grundlagen der geometrischen Datenverarbeitung. 2nd edn, Teubner, Stuttgart 1992.

[7] Pottmann, H.: Interpolierende Kurven und Flächen mit geometrischer Stetigkeit höherer Ordnung. Proceedings Austrographics'88 (1988).

[8] Pottmann, H.; Smooth curves under Tension. Computer Aided Design 22, 241 - 245 (1990).

[9] Späth, H.: Spline Algorithmen zur Konstruktion glatter Kurven und Flächen. 3rd edn. Oldenburg 1983.

[10]Stachel, H.: Parallel Projections in Multidimensional Space. Proc. Compugraphics 91 (ed. H.P.Santo), 119 - 127 (1991).

For further literature on this field the reader is referred to the references of the cited books and papers.

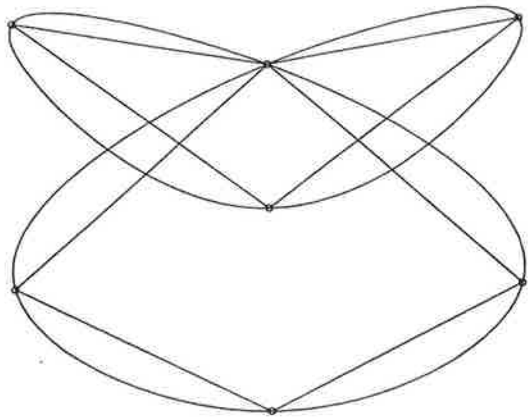

Figure 7: Further Example of interpolating subsplines for $k = 4$.