Tempo Modifications and Spline Functions
Daniel Muzzulini 1993

I Modifications of ~~the~~ time flow and related tempo modifications

are going to investigate

Let G_0: $E_1<E_2<\ldots<E_r$ be a fixed sequence of symbolic onsettimes and $e_0(E_i)$, $i=0,\ldots,r$ a physical reference-interpretation of the E_i. We ~~will study~~ modifications of the reference-interpretation, which are given in form of differences of physical onsettimes z_i as follows:

$$E_i \;\to\; e_0(E_i) \;\to\; e(E_i) = e_0(E_i) + z_i \quad (*)_i$$
$$i=0\ldots r \text{ and } z_0=0$$

It is our task to describe such modifications in terms of reciprocal tempo functions $R_0(E)$ and $R(E)$ with respect to e_0 and e. In order to get welldefined (reciprocal) tempo functions, we could interpolate the discrete functions of onsettimes $e_0(E_i)$ and $e(E_i)$ as appropriate differentiable functions defined on **R** and explore (the reciprocal of) their first derivation. ve
In practice, however, we start with a given tempo function T_0 and with the data of "local translations" $z_0=0$, $z_1,\ldots,z_r$ and we are looking for a modified tempo function T such that the conditions $(*)_i$ are satisfied. In the subsequent we request that $1/T_0=R_0$ and $1/T=R$ are spline functions of degree n defined on **R** and $R(E)=R_0(E)$ if E is not in (E_0,E_r). This implies $e(E) = e_0(E)$ for $E \leq E_0$ and $e(E) = e_0(E) + z_r$ for $E \geq E_r$.
If we requested from T and T_0 (instead of R and R_0) to be spline functions of degree n, the occuring integrals of rational functions in general could not be calculated explicitely. And yet in the case of $n = 1$, where explicite integration is possible, we would have to solve numerically a non-linear equation for every interval $[E_j,E_{j+1}]$ in order to determine T piecewise.
If R and R_0 are spline functions, all problems considered here can be solved by solving linear equation systems. Then the related time flows e and e_0 are also spline functions and can be evaluated by evaluating polynomials - no numerical integration is hereby needed.

Lemma (a space of spline functions)
Let $M(n,G)$ be the set of spline functions of degree n with the sequence of knots G: $x_0<x_1<\ldots<x_m$ $(m>n)$, which are identical zero on the complement of the interval $[x_1,x_m)$, i. e.

identical to zero

$M=\{\ g: R \to R /$
g is n-1 times continuously differentiable,
$g|_{[x_i,x_{i+1})}=p_i|_{[x_i,x_{i+1})}$, p_i=polynomial of degree $\leq n$, $i=0,\dots,m-1$,
$g(x)=0$ if $x \in R\setminus[x_0,x_m)\}$.

Then M(n,G) is a vector space of dimension m-n.

Proof:

(1) M(n,G) is a vector space:
Since M(n,G) is a subset of the space of all functions on R, M(n,G) is a vector space if linear combinations of elements of M(n,G) are also elements of M(n,G). This is clear.

(2) dim(M(n,G))=m-n
The space N(n,G) of all piecewise polynomial functions of degree $\leq$ n with the knot sequence G : $x_0<x_1<\dots<x_m$, which disappear outside of $[x_0,x_n)$, has dimension m(n+1) and M(n,G) is a subspace of it. (We have m independent polynomials of maximal degree n, that means m(n+1) independent coefficients for the polynomials, which form an element g of N(n,G)).
Let us consider the linear independent conditions on a g in N(n,G) that guarantee that g is also in M(n,G). These conditions are the following:

$p_0^{(k)}(x_0)=0=p_{m-1}^{(k)}(x_m)$, $k=0,\dots,n-1$ at the bounderies of G boundaries

$p_i^{(k)}(x_{i+1})=p_{i+1}^{(k)}(x_{i+1})$, $k=0,\dots,n-1$, $i=0,\dots,m-2$ on the inner knots.

The number n(m+1) of these conditions is equal to the rank of the linear system S for the "unknown" coefficients a_{ij} of the m polynomials p_i, which form g and therefore

$$\dim(M(n,G))= \dim(N(n,G))-\mathrm{rank}(S)=m(n+1)-n(m+1)=m-n .$$

To solve our problem we are looking for difference functions of reciprocal tempo functions, which belong to a suitable space of the form M(n,G) of the lemma. Supposing that the reciprocal reference tempo function R_0 is an arbitrary spline function of degree n, it is guaranteed that also R is a spline function of degree n.

Define $f(E) := e(E) - e_0(E)$, then the conditions $(*)_i$ for e are equivalent to the following conditions $(**)_i$ for f:

$$f(E_i)= z_i , \quad i=0,\dots,r \qquad (**)_i$$

From differentiation we get $1/T(E) = 1/T_0(E) + f'(E)$ or with $R(E) := e'(E) = 1/T(E)$ and $R_0(E)=e_0'(E)$, the reciprocal tempofunctions, $R(E) = R_0(E) +$

$f'(E)$. We request that $g := f' = R - R_0$ is in a suitable $M(n,G)$ and $\{E_0 = x_0, E_1, \ldots, E_r = x_m\}$ are elements of G such that $(**)_i$ are fulfilled. $(**)_i$ become now conditions for the definite integrals of g on the r intervals $[E_0, E_i]$:

$$\int_{E0}^{Ei} g(t)dt = z_i - z_0 = z_i$$

respectively on the intervals $[E_i, E_{i+1}]$:

$$\int_{Ei}^{Ei+1} g(t)dt = z_{i+1} - z_i.$$

We must choose G (dependently of n) such that these conditions can be formulated as r indepent linear equations in the coefficients of the polynomials, which form g. With aid of the lemma we get the following result

<u>Theorem</u>
If we add n arbitrary new knots in the inner of the given knot sequence G_0: $E_0 < E_1 < \ldots < E_r$ to define G ($E_0 = X_0 < X_1 < \ldots < X_{r+n-1} < X_{r+n} = E_r$), our interpolation integration problem $(**)_i$, $i = 0, \ldots, r$ is uniquely solvable with $g = f'$ in $M(n,G)$.

Remarks:
(1) $1/T$ is a spline function of degree n, if $1/T_0$ is a spline function of degree n. (The knot sequence of $1/T$ is G plus the knots of $1/T_0$ not belonging to G.)
(2) The corresponding time functions e_0 and e are in this case spline functions of degree n+1.
(3) The values $e(X_j)$, X_j in $G \backslash G_0$ are determined by the interpolation!
(4) For arbitrary n+r+1 knots not containing G_0 our problem is not always solvable. If we ~~liked~~ wanted to work with equidistant knots, we would be probably constrained to take more than n+r+1 knots to get a solution - with loss ~~lost~~ of its uniqueness! (Nürnberger, Theorem 3. 7, p. 109, gives a complete characterization of those knot sequences, for which the so-called Hermite interpolation problem has a unique solution. Our problem is of this type for the function f.)

I.1 Algorithms for n=1

Let G: $E_0<E_1<...>E_k<P<E_{k+1}<...<E_r$ be given. We are looking for a polygon g on the knot sequence G with $g(E_o)=0=g(E_r)$ and $I_j := \int_{E_j}^{E_{j+1}} g(t)dt = z_{j+1} - z_j$ $(z_0=0)$.

g is uniquely determined by the values on G.

We define

$$
\begin{aligned}
u_j &= g(E_j) && j = 0,...,k \\
w &= g(P) && \\
v_j &= g(E_j) && j = k+1,...,r \\
H_j &= E_{j+1}-E_j && j = 0,...,r-1
\end{aligned}
$$

The conditions for the definite integrals I_j are the following:

$$
\begin{aligned}
I_j &= \frac{1}{2} H_j(u_j + u_{j+1}) \quad j = 0,...,k \\
I_k &= \frac{1}{2}(P - E_k)(u_k + w) + \frac{1}{2}(E_{k+1} - P)(w + v_{k+1}) \qquad j = k \\
I_j &= \frac{1}{2} H_j(u_j + u_{j+1}) \quad j = k+1,...,r
\end{aligned}
$$

The u_j and v_j can be determined recursively upwards and downwards:

$$
\begin{aligned}
u_0 &= 0, \qquad u_j = \frac{2I_{j-1}}{H_{j-1}} - u_{j-1}, \qquad 1 \le j \le k \\
v_r &= 0, \qquad v_j = \frac{2I_j}{H_j} - v_{j+1}, \qquad r \ge j \ge k+1
\end{aligned}
$$

With the values u_k and v_{k+1} we can determine w:

$$
w = \frac{2I_k - (P - E_k)u_k - (E_{k+1} - P)v_{k+1}}{H_k}
$$

Variation of P inside of an interval $[E_k,E_{k+1}]$ shows that w is a linear function of P with gradient $m = \frac{v_{k+1} - u_k}{H_k}$ trough the points (E_k,v_k) and (E_{k+1},u_{k+1}), if we continue the above recursions one step (see *Figure 1*). /hr

If we take especially $P = \frac{E_k + E_{k+1}}{2}$ in the middle of I_k, we obtain

$$
w = \frac{2I_k}{H_k} - \frac{u_k + v_{k+1}}{2} = \frac{u_{k+1} + v_k}{2}.
$$

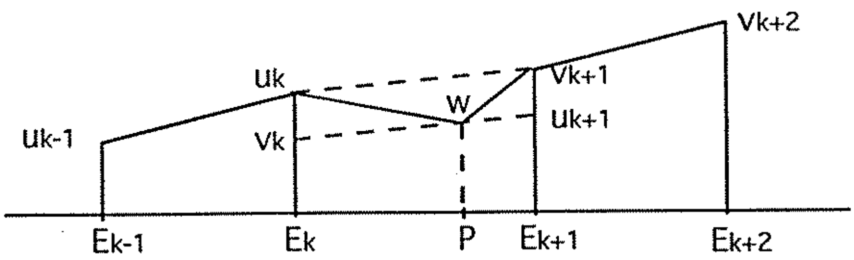


Figure 1. w is a linear function of P inside of $[E_k,E_{k+1}]$

The example in *Figure 2* shows that the shape of the solution polygon depends strongly on the selection of the interval $[E_k,E_{k+1}]$, which contains the extra knot P. To the knots $E_0,\dots,E_5$ we have drawn the solutions to all extra knots $P_j=\frac{E_j+E_{j+1}}{2}$ for $4 \geq j \geq 0$. Below there is shown the unique corresponding solution for step functions (splines of degree 0, without extra knots). The latter is everywhere ≥ 0 in contrast to the the former, which all have two sign changes in (E_0,E_5). greatly

The calculation of all possible solutions to the extra knots in the middle of the subintervals $[E_k,E_{k+1}]$ needs only few operations (3r multiplication/divisions and 4r additions to get the edges of the polygons), since we must only calculate all u_k, v_k, $k= 0,\dots,r$ and for every extra knot P_k an arithmetic mean.

It seems to be useful to determine the adequate solution after calculation of all of them interactively or/and with appropriate topological criterions as for instance criteria

$$g(P)^2+\sum_{j=0}^{r} g(E_j)^2 = \text{minimal}.$$

If g_j are the solutions with extra knot P_j in $[E_j,E_{j+1}]$, then every linear combination $\sum_{j=0}^{r}\alpha_j g_j$ with $\sum_{j=0}^{r}\alpha_j = 1$ is a solution with the knot sequence $E_0<P_0<E_1<P_1<E_2<\dots<P_{r-1}<E_r$, particularly the arithmetic mean $\frac{1}{r}\sum_{j=0}^{r} g_j$.

Figure 3 shows some solutions of this kind with more than the minimal number of extra knots from the example of *Figure 2*.

I. 2 Algorithms for n=2

There are two ways of placing the two extra knots P_1, P_2 needed now:

(a) G: $E_0<E_1<...<E_k<P_1<P_2<E_{k+1}<...<E_r$ $(r \geq 1)$

(b) G: $E_0<E_1<...<E_k<P_1<E_{k+1}<...<E_l<P_2<E_{l+1}<...<E_r$ $(k+1 \leq l, r \geq 2)$

(a) As for n = 1 we can compute the restrictions of the solution g on the interval $[E_j,E_{j+1}]$ for $j \neq k$ recursively upwards and downwards.

$0 \leq j < k$:

$$p_j(t):= \sum_{i=0}^{2} a_{ji}(t - E_j)^i = g(t) \text{ for } t \text{ in } [E_j, E_{j+1}]$$

We have $p_0(E_0) = 0$ and $p_0'(E_0) = 0$ since g is in M(2,G).
Provided that we know the coefficients of p_{j-1} we can determine

$$p_j(E_j) = p_{j-1}(E_j) \text{ and } p_j'(E_j) = p_{j-1}'(E_j).$$

Thus we obtain

$$a_{j0} = p_j(E_j)$$

$$a_{j1} = p_j'(E_j)$$

$$a_{j2} = \frac{3(z_{j+1} - z_j)}{H_j^3} - a_{j0}H_j - \frac{1}{2}a_{j1}H_j^2 \text{ (from } z_{j+1} - z_j = \int_{E_j}^{E_{j+1}} p_j(t)dt)$$

$r-1 \geq j \geq k+1$:

With $q_j(t):= \sum_{i=0}^{2} b_{ji}(t - E_{j+1})^i = g(t)$ for t in $[E_j, E_{j+1}]$ we can proceed as for $j < k$ by descending recursion. (Developping q_j at E_j forces to solve a linear system with three unknowns. We can get this representation of q_j from the above with help of $q_j''(E_j)$ without solving linear systems in order to obtain an uniform representation of the solution.)

We write the remaining three polynomials s_0, s_1, s_2 within $[E_k, E_{k+1}]$ in the form $s_j(t) = \sum_{i=0}^{2} c_{ji}(t - P_j)^i$ with $P_0 := E_k$ and $P_3 := E_{k+1}$. Further we set $L_j:= P_{j+1} - P_j$, $j = 0,...,2$. Then the coefficients of the s_j can be determined by solving the following linear system:

$$
\begin{pmatrix}
1 & & & & & & & & \\
 & 1 & & & & & & & \\
-1 & -L_0 & 1 & & & & -L_0^2 & & \\
 & -1 & & 1 & & & -2L_0 & & \\
 & & -1 & -L_1 & 1 & & & -L_1^2 & \\
 & & & -1 & & 1 & & -2L_1 & \\
 & & & & -1 & -L_2 & & & -L_2^2 \\
 & & & & & -1 & & & -2L_2 \\
L_0 & \frac{L_0^2}{2} & L_1 & \frac{L_1^2}{2} & L_2 & \frac{L_2^2}{2} & \frac{L_0^3}{3} & \frac{L_1^3}{3} & \frac{L_2^3}{3}
\end{pmatrix}
\begin{pmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \\ c_{20} \\ c_{21} \\ c_{02} \\ c_{12} \\ c_{22} \end{pmatrix}
=
\begin{pmatrix} p_{k-1}(E_k) \\ p'_{k-1}(E_k) \\ 0 \\ 0 \\ 0 \\ 0 \\ -q_{k+1}(E_{k+1}) \\ -q'_{k+1}(E_{k+1}) \\ z_{k+1}-z_k \end{pmatrix}
$$

in such a way

We have arranged the unknown coefficients suchway that it is easy to transform the system into a system with ~~upper~~ triangular matrix.

lower

We can proceed in this manner also for $n > 2$ if the n extra knots are in the same interval $[E_k,E_{k+1}]$.

(b) For $j < k$ and for $j \geq l+1$ we can proceed as in (a). In order to determine the remaining polynomials we have to solve a linear system with 3(l-k+3) equations. Renaming the knot sequence $E_k<P_1<E_{k+1}<\ldots<E_l<P_2<E_{l+1}$ to $x_0<x_1<\ldots<x_m$, $m= l-k+3$, and with $h_j = x_{j+1} - x_j$, $j=0,\ldots,m-1$, we get the system shown in *Figure 4* for the coefficients of the polynomials $s_j(t) = \sum_{i=0}^{2} c_{ji}(t - x_j)^i$. It can be reduced to a subsystem for the quadratic coefficients, of degree 2

?

which is nearly lower triangular. Having solved this reduced system, the linear and constant coefficients are obtained by forward insertion. See *Figure 5*.

I. 3 Algorithms for $n \geq 3$

As already mentioned the placing of all extra knots in the same interval is unproblematic for computation. As the example in *Figure 6* shows, this way of placing the extra knots need not correspond to the intuitively imagined shape of the solution already for small n! In all other cases we

have to solve linear systems with $(n+1)(l-k+n+1)$ equations, when $[E_k, E_{l+1}]$ is the smallest subintervall of the knot sequence $E_0<E_1<\ldots<E_r$, which contains all extra knots in order to obtain the polynomials within $[E_k, E_{l+1}]$. (The polynomials outside of $[E_k, E_{l+1}]$ can be determined recursively as for $n=2$.)
The part of the system matrix, which corresponds to the smoothness conditions, is thereby always of the same form, whereas the part corresponding to the integral conditions varies according to the distribution of the extra knots.

Figure 6 shows some solutions of the same interpolation integration problem for $n = 0,\ldots,3$. The placing of all three extra knots equidistantly in the same interval [0,4] for $n = 3$ gives a solution with an oscillation, which obscures the conception of the modifications as a ritardando. (This is already the case for $n = 2$, if both extra knots are equidistantly taken in [0, 4].) It also can be seen that the maximal elongation from 0 becomes larger for larger n. It is probably not very reasonable to use spline functions of degree > 2 to solve our tempo modification problems.

II B-splines

Given a spline function by its piecewise polynomials contains a lot of information: In the case of M(n,G) we need m(n+1) coefficients to describe an element in a space of dimension m-n.
The representation of spline functions as linear combinations of B-splines, which are spline functions with minimal support, is very common because of its numerical stability.

Theorem

M(n,G) has a uniquely determined basis of B-splines B_j^n, j=0,..., m-n-1.
The B_j^n are completely characterized by the following properties (B1)-(B3):

$$B_j^n(t) = 0 \quad t \text{ in } \mathbf{R} \setminus (x_j, x_{j+n+1}) \tag{B1}$$

$$B_j^n(t) > 0 \quad t \text{ in } (x_j, x_{j+n+1}) \tag{B2}$$

$$\int_{x_j}^{x_{j+n+1}} B_j^n(t)dt = 1 \tag{B3}$$

For a proof of existence and uniqueness of splines fulfilling (B1)-(B3) see for instance Nürnberger, Theorem 2.2., p. 96.

Remarks

(0) Some examples of B-splines are shown in *Figure 7*.

(1) The one dimensional vector space spanned by one of the B_j^n is itself of the type M(n,G) for G=G_j : $x_j<x_{j+1}<\dots x_{j+n+1}$.

(2) Provided the existence of B-splines, it is easy to see that they have *minimal* support:
Take to the contrary a spline function s of degree n with support $[x_j,x_k]$, $k<j+n+1$, i.e. s(t)=0 for t in $\mathbf{R}\setminus(x_j,x_k)$. This implies s in $M(n,G_j)$ since $s^{(k)}(x_{j+n+1}) = 0$, k = 0,...,n-1 and therefore $s=\alpha B_j^n$. B_j^n is strictly positive on (x_j,x_{j+n+1}) particularly in (x_k,x_{j+n+1}). This implies $\alpha = 0$ and supp(s)=Ø.

(3) Because of dim(M(n,G)) = m-n = #$\{B_j^n | j=0,\dots,m-n-1\}$ it is sufficient to show that the B_j^n are linear independent in order to demonstrate that they form a basis of M(n,G):
proof

$$\sum_{i=0}^{m-n-1} \alpha_j B_j^n(t) = 0 \Rightarrow \alpha_j = 0, j = 0,\ldots,m-n-1$$

Take t in (x_0,x_1). This implies $\alpha_0 = 0$, since the B_j^n for $j \geq 1$ disappear on $[x_0,x_1]$. The same argument now for t in (x_1,x_2) implies $\alpha_1 = 0$, etc.

(4) The B-splines B_j^n can be determined using divided differences (Nürnberger, Theorem 2. 9, p. 99) or by the following recurrence relation (Nürnberger, Theorem 2. 11, p. 101) due to de Boor (1972) and Cox (1972):

$$B_j^n(t) = \frac{n+1}{n}\left[\frac{t - x_j}{x_{j+n+1} - x_j} B_j^{n-1}(t) + \frac{x_{j+n+1} - t}{x_{j+n+1} - x_j} B_{j+1}^{n-1}(t)\right]$$

(5) The often used normalized B-splines $N_j^n = \frac{x_{j+n+1} - x_j}{n+1} B_j^n$, $j = 0,\ldots,m-n-1$ also form a basis of M(n,G) (basis transformation with diagonal matrix) and can be determined similarly (Nürnberger, Theorem 2. 14, p. 103).

(6) For equidistant knots $x_{j+1} - x_j = h$, $j = 0,\ldots,m-1$, B_{j+k}^n can be obtained from B_j^n by translation along the x-axis:

$$B_j^n(t) = B_{j+k}^n(t - kh)$$

(7) Let G* be a knot sequence, which is a refinement of G. Then M(n,G) is a subspace of M(n,G*). The problem is then to calculate in an efficient way the B-spline coefficients of a spline s in M(n,G) related to G* from the coefficients of s related to G. There are algorithms for the insertion of one extra knot (Boehm, 1980) and for the insertion of several extra knots at a time (Cohen et al., 1980) described in Dierckx, p. 16 f. These knot insertion algorithms can be used in order to find the B-spline representation of the sum of spline functions with different knot sequences (determine the coefficients of the functions to be added for the union of the knot sequences and add corresponding coefficients). If we describe reciprocal tempo functions and their modifications in the basis of B-splines, we are often in ~~the situation~~ able to add splines with different knot sequences.
Using Taylor polynomials instead of B-spline representations, a refinement of the knot sequence forces us to evaluate the piecewise polynomials and their derivatives related to G for all extra knots in order to obtain the piecewise representation of a spline related to G*.

We are often in a position to

(8) There are also formulas (de Boor, Lyche&Schumaker,1972/1976) for computing the B-spline representations of the derivatives and of the indefinite integral of a spline function from its B-spline coefficients, see Nürnberger, Theorems 2.17 / 2. 18, p. 104 f.

(9) The solution of spline interpolation problems in the B-spline representation leads to bend matrices for the coefficients α_j of a spline s(t) = $\sum\alpha_j B_j^n(t)$. For an interpolation point τ in (x_k,x_{k+1}) there are only n+1 indices j (j = k-n,...,k) with τ in supp(B_j^n) and if $\tau = x_k$ for only n indices j (j=k-n,...,k-1) we have $B_j^n(\tau) \neq 0$.

II.1. Synchronization and B-splines

To synchronize the reciprocal tempo function R to a reference reciprocal tempo function R_0 on a symbolic time interval [A,B], their difference function $g = R - R_0$ must fulfill the condition

$$\int_A^B g(t)dt = 0 \qquad \text{(Syn)}$$

We consider as previously only modification functions g in M(n,G) for an appropriate knot sequence G : $A=E_0<E_1<...<E_m=B$ with m > n+1. (m ≤ n+1 only allows g = 0)

Lemma

The set Syn(n,G) of all spline functions in M(n,G), which fulfill (Syn) can be described as follows:

$$\mathrm{Syn}(n,G) = \{g = \sum_{j=0}^{m-n-1}\alpha_j B_j^n(t) \mid \sum_{j=0}^{m-n-1}\alpha_j = 0\}$$

Syn(n,G) is a vector space of dimension m-n-1 (isomorphic to M(n+1, G)).

This follows immediately from the property (B2) of B-splines and from the linearity of integration. (g in Syn(n,G) further implies $f(t) := \int_A^t g(s)ds$ in M(n+1,G).)

II. 1. 1 Interactive definition of synchronizing tempo modifications

There are several possibilities of doing this:

(1) Input: $\alpha_0,\ldots,\alpha_k,\ 0 \le k \le m\text{-}n\text{-}2$

Output: $g_k = \sum_{j=0}^{k} \alpha_j B_j^n - (\sum_{j=0}^{k} \alpha_j) B_{k+1}^n)$

The system always returns the spline in Syn(n,G) with minimal support. The visualization of the g_k helps to ~~controle~~ control further input. Finally we set $g := g_{m-n-2}$.

The sequence of the B-spline coefficients reflect the shape of a spline function in a rough manner: or: reflect ~~approximately~~ roughly the shape of a spline function

The number of sign changes in the sequence (α_j) is less or equal to the number of sign changes of $g = \sum \alpha_j B_j^n$ (Nürnberger, Corollary 2. 22, p. 107).

In the basis of the normalized B-splines N_j^n we can use the corresponding coefficients β_j to approximate a spline function with the aid of the so-called B-polygon defined on the knot sequence

$$x_j := \frac{1}{n+1} \sum_{i=j}^{j+n+1} E_i$$ by the edges (x_j, β_j) (completed with (A, 0) and (B, 0))

(cf de Boor, p. 87f)

Instead of defining the $\alpha_0,\ldots,\alpha_k$, the user therefore could enter the B-polygon up to β_k, and the system returns the B-polygon of g_k on $[A, E_{k+n+2}]$ or a representation of g_k.

B-polygons are used in CAD, because they have shape perserving properties: positivity, monotony, convexity, concavity in relation to the approximated spline function (de Boor, p. 86).

(2) In M(n,G) it is possible to define a spline successively (via "mouse klicking"):

Because of the strong smoothness condition at the boundary E_0,the restriction of a spline function g in M(n,G) on a ~~subintervall~~ subinterval $[E_0,E_k]$, $k \le m\text{-}n\text{-}1$, is uniquely determined by the values $g(E_1),\ldots,g(E_k)$.

We can therefore determine the g_k in Syn(n,G) from above directly, if the user enters the values $g(E_1),\ldots,g(E_{k+1})$. The coefficients $\alpha_0,\ldots,\alpha_k$ are obtained by solving the following linear system by forward elimination:

$$
\begin{bmatrix}
B_0^n(E_1) & & & & & & \\
B_0^n(E_2) & B_1^n(E_2) & & & & & \\
\vdots & & \ddots & & & & \\
B_0^n(E_n) & \cdots & B_{n-1}^n(E_n) & & & & \\
 & B_1^n(E_{n+1}) & \cdots & B_n^n(E_{n+1}) & & & \\
 & & \ddots & & \ddots & & \\
 & & & B_{k-n+1}^n(E_{k+1}) & \cdots & & B_k^n(E_{k+1})
\end{bmatrix}
\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \\ \\ \alpha_k \end{pmatrix}
=
\begin{pmatrix} g(E_1) \\ g(E_2) \\ \vdots \\ \\ \\ g(E_{k+1}) \end{pmatrix}
$$

Then $g_k = \sum_{j=0}^{k} \alpha_j B_j^n + \gamma_k B_{k+1}^n$ with $\gamma_k = -\sum_{j=0}^{k} \alpha_j$.
Figure 8 illustrates the method for n=1.

The method must be altered, if we go away from previously fixed knots. In order to determine the coefficients $\alpha_0,\dots,\alpha_k$, γ_k the knots $E_{k+2},\dots,E_{k+n+2}$ must be known in advance, although the restrictions of g_k on $[E_0,E_{k+1}]$ is uniquely determined by the values of g_k on the knots $E_0,\dots,E_{k+1}$. In other words, some of the α_j are influenced by modifications in the knot subsequence $E_{k+2},\dots,E_{k+n+2}$ (namely for $j = k-n-1,\dots,k$), since the corresponding B-splines are altered.

(3) A method with variable knot sequence

Input: $(E_1, g(E_1)),\dots,(E_{k+1},g(E_{k+1}))$ $(A=E_0<E_1<\dots E_{k+1}<B)$

Output: g_k in $Syn(n,G_k)$ on the knot sequence G_k:
$E_0<E_1<\dots<E_{k+1}=D_0<D_1<\dots<D_{n+1}\le B$ with $D_j = E_{k+1} + jh$
and $h = \min\{\frac{E_{k+1}-E_0}{k+1}, \frac{B-E_{k+1}}{k+1}\}$.

After input of new $(E_j,g(E_j))$, $j > k+1$, the automatically defined $D_1,\dots,D_{n+1}$ must be replaced. As mentioned above, ~~there~~ must be computed new B-splines after each new input. It seems to be more efficient in this case to calculate explicit~~e~~ representations of the restrictions of g_k to the intervals $[E_j,E_{j+1}]$. The corresponding Taylor polynomials up to $[E_k,E_{k+1}]$ can be obtained recursively, and they need not be altered after new input.

? Please ask

For the remaining polynomials on the equidistant knot sequence $D_0 < .. < D_{n+1}$ there is to solve a linear system of the same type as described for n=2 in Chapter I. 2 (a) (g_k must fulfill the integral condition $\int_{D_0}^{D_{n+1}} g_k(t)dt =$

has to ~~must~~ be solved

$- \int_{E_0}^{E_{k+1}} g_k(t)dt$). The occuring (sparse) system matrix depends only on n and h, and it can be inverted or factorized (using LR-decomposition) for a given n as a function of h.

References

Carl de Boor: *Splinefunktionen*, Birkhäuser, Basel 1990 (Z GS 5353)

Paul Dierckx: *Curve and Surface Fitting with Splines*, Oxford University Press, New York 1993 (Z GS 30034)

Günther Nürnberger: *Approximation by Spline Functions*, Springer, Berlin-Heidelberg 1989 (Z GS 2511)

Helmuth Späth: *Eindimensionale Spline-Interpolations-Algorithmen*, Olden- bourg, München 1990 (Z GS 4902)

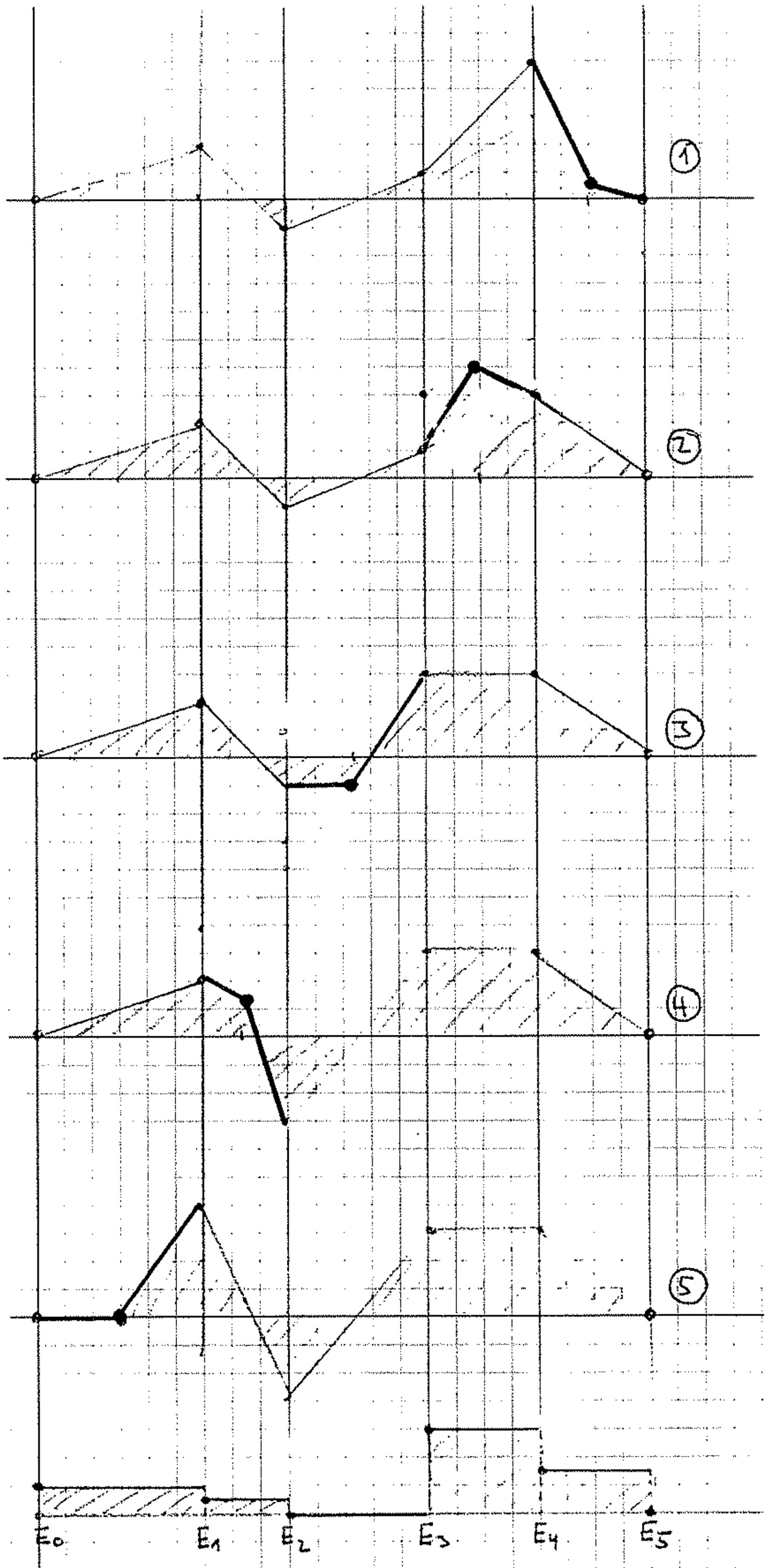


Figure 2

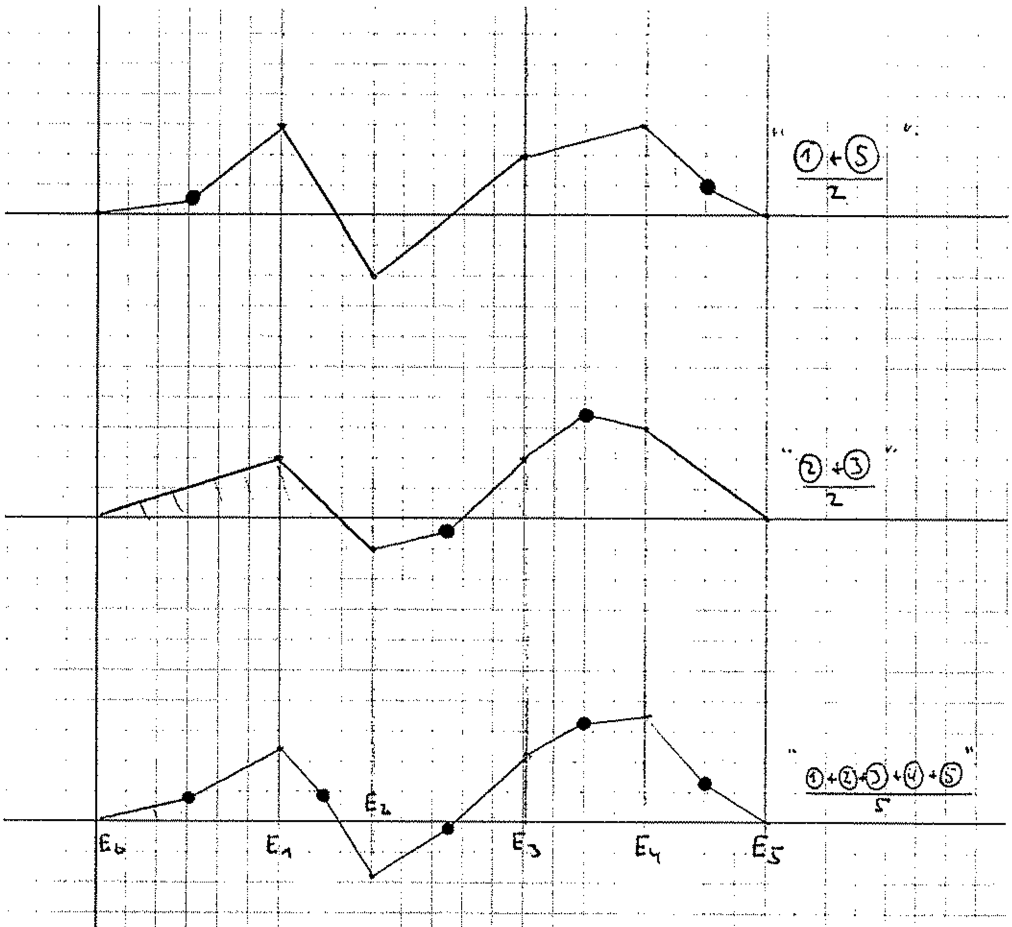


Figure 3. More than the minimal number of extra knots.
(example of Fig. 2)

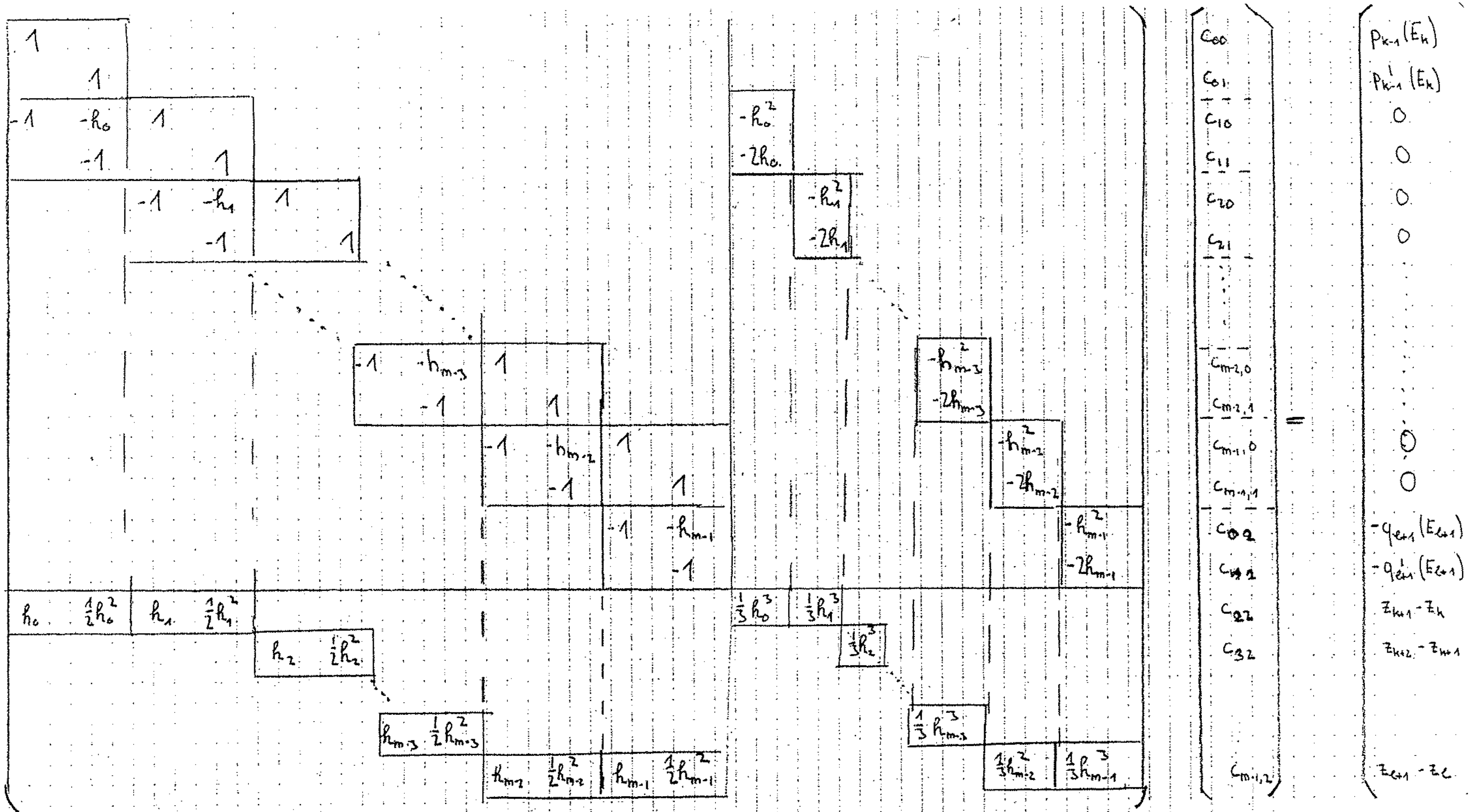
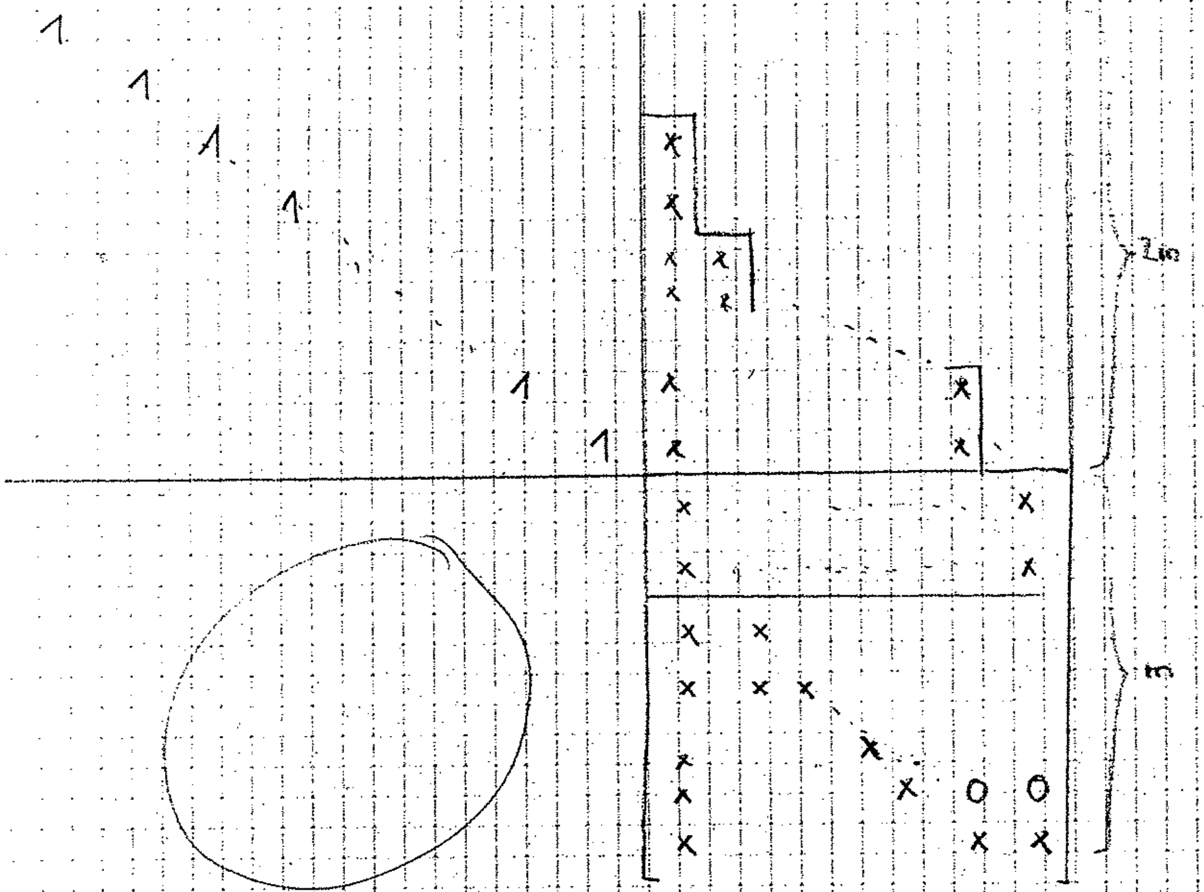


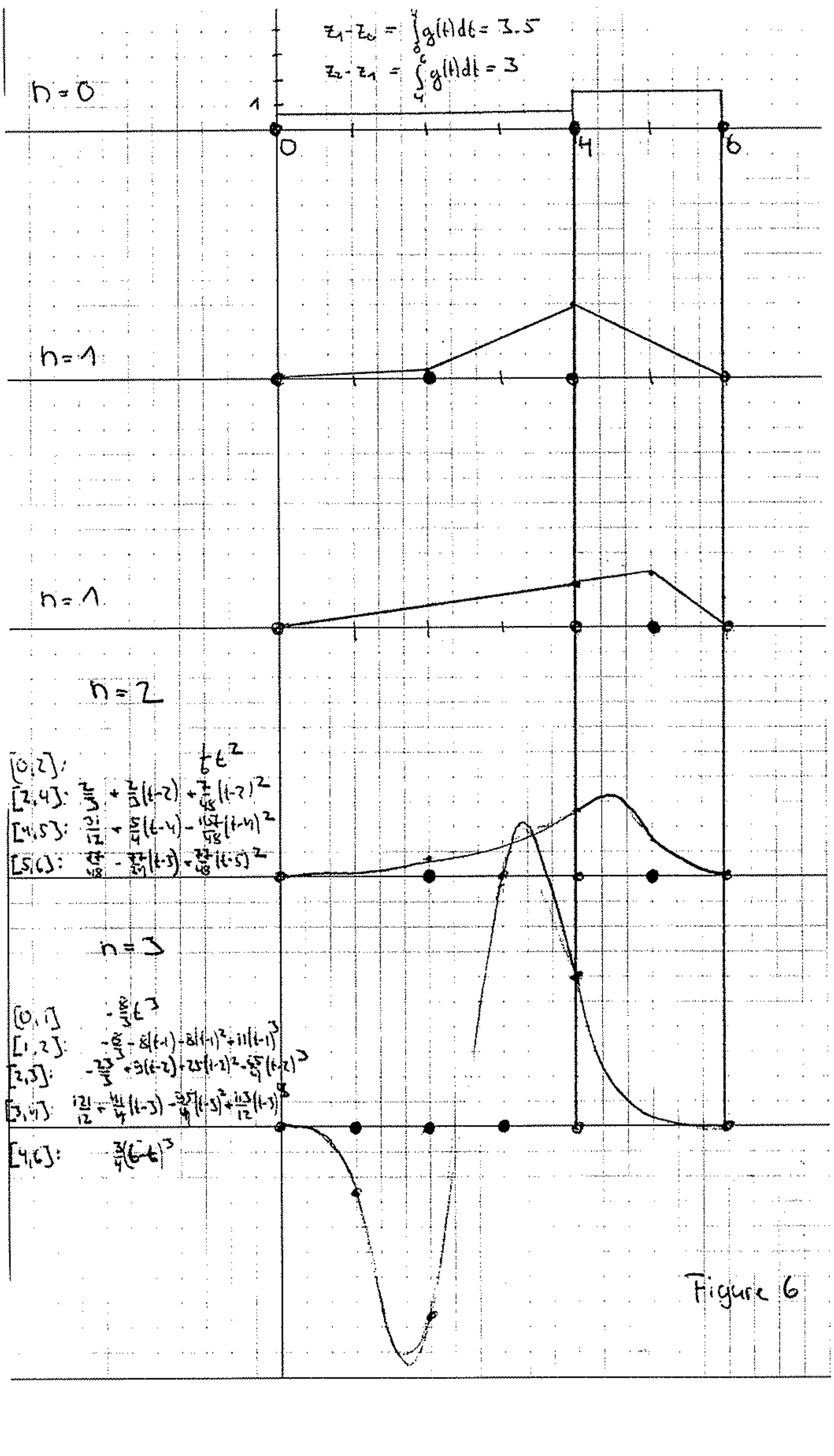
Figure 4



$$\begin{pmatrix} I & B \\ O & C \end{pmatrix}$$

$$C\begin{pmatrix} c_{0z} \\ c_{1z} \\ \vdots \\ c_{m-1,z} \end{pmatrix} = \begin{pmatrix} x \\ x \\ \vdots \\ x \end{pmatrix} \Longrightarrow \begin{pmatrix} x & & \\ \vdots & \ddots & \\ x & \cdots & x \end{pmatrix} \begin{pmatrix} c_{0z} \\ \vdots \\ c_{m-1,z} \end{pmatrix} = \begin{pmatrix} x \\ \vdots \\ x \end{pmatrix}$$

Figure 5. Reduction of the system matrix of Figure 4.



$z_1 - z_0 = \int_0^4 g(t)\,dt = 3.5$
$z_2 - z_1 = \int_4^6 g(t)\,dt = 3$
n = 0
1
0
4
6
n = 1
n = 1
n = 2
[0,2]: $\frac{1}{6}t^2$
[2,4]: $\frac{2}{3} + \frac{2}{3}(t-2) + \frac{7}{48}(t-2)^2$
[4,5]: $\frac{31}{12} + \frac{5}{4}(t-4) - \frac{157}{48}(t-4)^2$
[5,6]: $\frac{27}{48} - \frac{27}{24}(t-5) + \frac{27}{48}(t-5)^2$
n = 3
[0,1] $-\frac{8}{3}t^3$
[1,2]: $-\frac{8}{3} - 8(t-1) - 8(t-1)^2 + 11(t-1)^3$
[2,3]: $-\frac{23}{3} + 3(t-2) + 25(t-2)^2 - \frac{55}{4}(t-2)^3$
[3,4]: $\frac{121}{12} + \frac{41}{4}(t-3) - \frac{327}{4}(t-3)^2 + \frac{113}{12}(t-3)^3$
[4,6]: $\frac{3}{4}(6-t)^3$

Figure 6

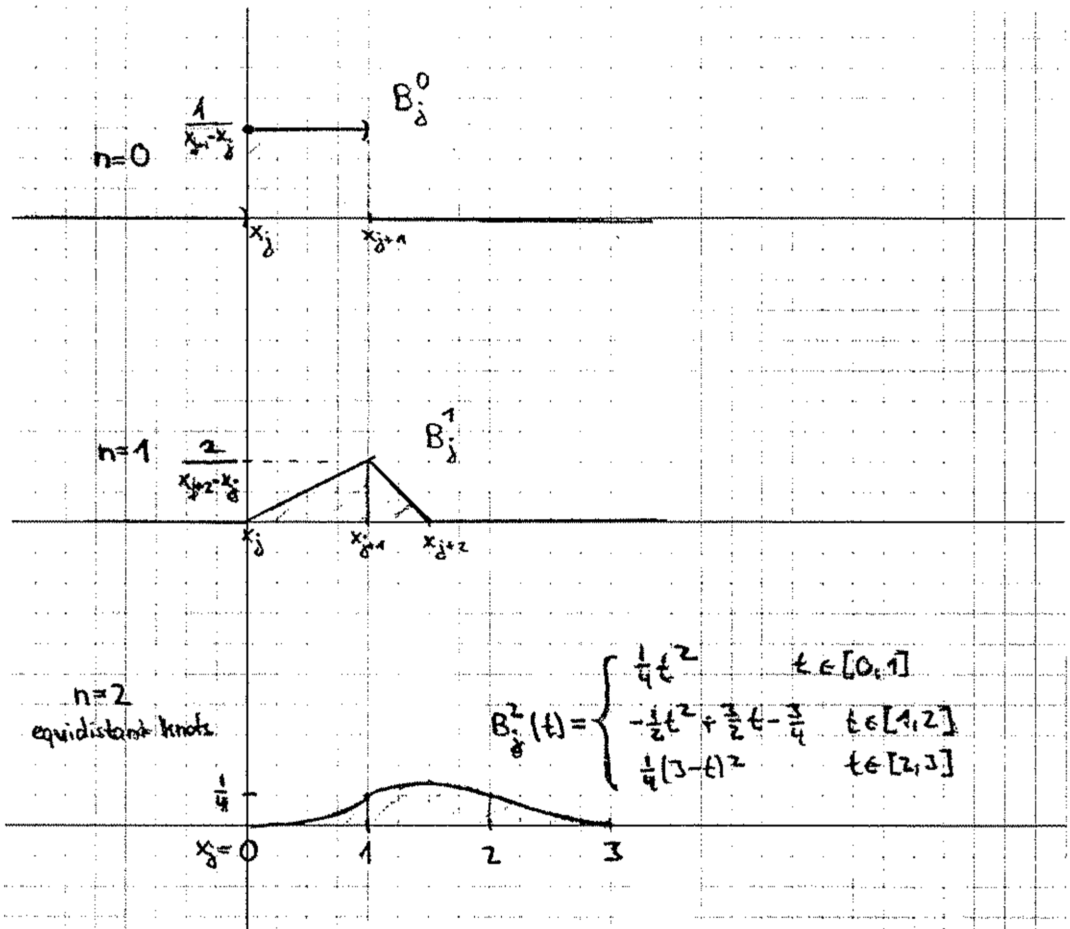


Figure 7. Some examples of B-splines

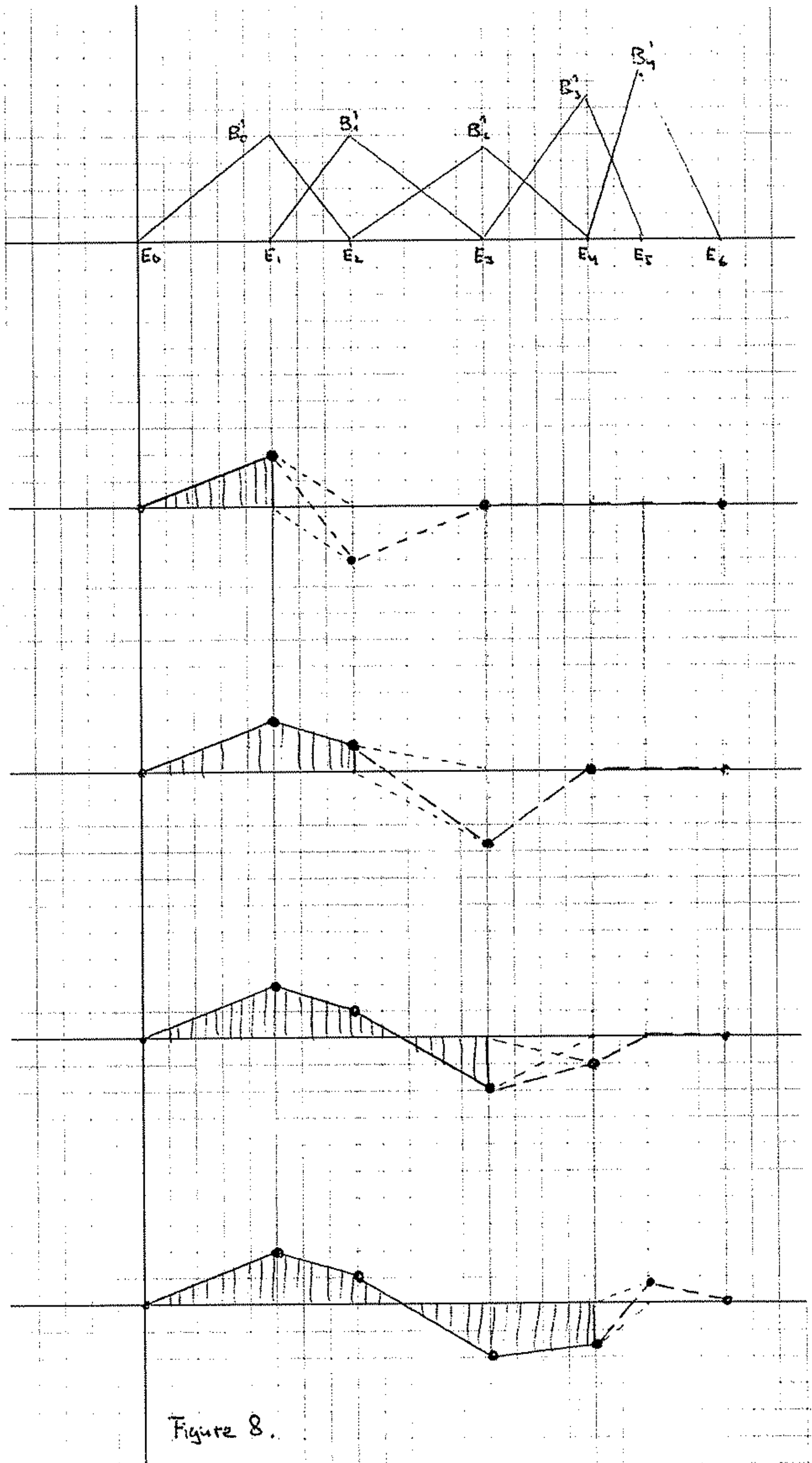
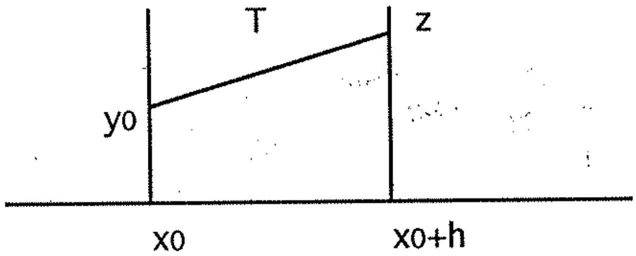


Figure 8.

Polgonale Tempofunktionen

1. Problem



$$T(x) = y_0 + \frac{z-y_0}{h}(x-x_0) \qquad x \text{ in } [x_0,x_0+h]$$

Gesucht z (z>0), so dass $\int_{x_0}^{x_0+h} \frac{1}{T(x)}dx = I$ für einen vorgegebenen Wert I.

Definiere

$$J(z) := \frac{h}{z-y_0}[\ln(z) - \ln(y_0)] \qquad \text{für } z \neq y_0$$

$$J(y_0) := \frac{h}{y_0}$$

Für das gesuchte z muss gelten $J(z) = I$.
Es gilt $J(z) \to 0$ für $z \to \infty$ und $J(z) \to \infty$ für $z \to 0$ und J ist stetig. Unser Problem hat deshalb für alle $I > 0$ eine Lösung.

Verfahren zur numerischen Bestimmung von z:

1) Falls $I = \frac{h}{y_0}$, dann ist $z = y_0$ die gesuchte Lösung.
2) Andernfalls wird die Nullstelle $\zeta \neq y_0$ von

$$f(z) := (z-y_0)I - (\ln(z) - \ln(y_0))h$$

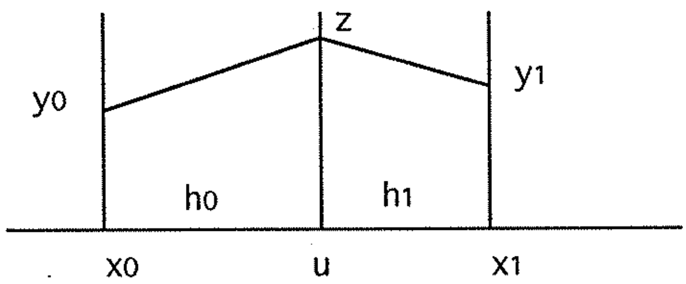
mit dem Newton-Verfahren

$$z_{i+1} = z_i - \frac{f(z_i)}{f'(z_i)}$$

berechnet. Dann ist ζ die gesuchte Lösung mit $J(\zeta) = I$
Zur Wahl des Startwertes z_0:
Es ist $f'(z) = I - \frac{h}{z}$ und $f''(z) = \frac{h}{z^2}$.
Die Funktion f ist also linksgekrümmt und hat ihr Minimum bei $z=\frac{h}{I}$.

Ist $y_0 < \frac{h}{I}$, dann konvergiert das Newtonverfahren für alle Startwerte $z_0 > \frac{h}{I}$ gegen die Nullstelle ζ. (Wegen der Linkskrümmung von f ist $z_1 > \zeta$, und die weiteren Folgeglieder bilden eine monoton fallende Folge).
Ist $y_0 > \frac{h}{I}$, dann konvergiert das Newtonverfahren für alle Startwerte $z_0 < \frac{h}{I}$ mit $f(z_0) > 0$ gegen ζ. (Die z_i bilden dann wegen der Linkskrümmung von f eine monoton wachsende Folge). z_0 muss also nur genügnd klein gewählt werden. (Beginne versuchsweise mit $z_0 = 0.5\,\frac{h}{I}$ und verkleinere falls nötig bis $f(z_0) > 0$, um einen geeigneten Startwert zu finden.)

2. Problem



$$T(x) = y_0 + \frac{z-y0}{h0}(x-x_0) \qquad x \text{ in } [x_0,u]$$
$$T(x) = y_1 + \frac{y1-z}{h}(x-x_1) \qquad x \text{ in } [u,x_1]$$

Gesucht z (z>0), so dass $\int_{x0}^{x1} \frac{1}{T(x)}dx = I$ für einen vorgegebenen Wert I.

Definiere

$$g(z) := h_0\frac{\ln(z) - \ln(y0)}{z - y0} + h_1\frac{\ln(z) - \ln(y1)}{z - y1} - I \qquad \text{für } y_0 \neq z \neq y_1,$$

Falls $y_0 \neq y_1$:

$$g(y_0) := \frac{h0}{y0} + h_1\frac{\ln(y0) - \ln(y1)}{y0 - y1} - I$$
$$g(y_1) := h_0\frac{\ln(y1) - \ln(y0)}{y1 - y0} + \frac{h1}{y1} - I$$

Falls $y_0 = y_1$:

$$g(y_0) := \frac{h0}{y0} + \frac{h1}{y1} - I$$

Die Funktion g ist stetig, g(z) -> ∞ für z -> 0 und g(z) -> -I für z -> ∞. Ferner ist g streng monoton fallend (Die im Ausdruck für g auftretenden Quotienten können als Sekantensteigungen zur Logarithmusfunktion interpretiert werden). g hat also genau eine Nullstelle ζ. Sie kann mit dem Bisektionsverfahren (Einschliessen der Nullstelle durch fortgesetzte Intervallhalbierung) bestimmt werden. Das Auffinden geeigneter Startwerte ist wegen der Monotonie von g einfach. $z = \zeta$ löst unser Problem.
Die Funktion g ist auch an den Stellen y_0 und y_1 differenzierbar, so dass im Prinzip auch mit dem Newtonverfahren gearbeitet werden könnte. Es scheint mir schwierig, einen geeigneten Starwert anzugeben, so dass das Verfahren mit Sicherheit konvergiert. (Ein Kandidat für einen Startwert ergibt sich möglicherweise aus der Lösung der Aufgabe mit polygonalen reziproken Tempofunktionen:

$I = \frac{1}{2}h_0(\frac{1}{y_0} + \frac{1}{z_0}) + \frac{1}{2}h_1(\frac{1}{z_0} + \frac{1}{y_1})$ ergibt $z_0 = \frac{h_0 + h_1}{2I - \frac{h_0}{y_0} - \frac{h_1}{y_1}}$.

Falls sich $z_0 > 0$ ergibt, ist jedenfalls $z_0 > \zeta$.)

Das 1.Problem kann natürlich für die Funktion g(z) = J(z) - I analog mit Bisektion gelöst werden. Das Bisektionsverfahren konvergiert allerdings langsamer als das Newtonverfahren.

Es ist zu bemerken, dass unsere Aufgaben für beliebige I > 0 immer (eindeutig) lösbar sind. Das Problem negativer Tempi kann hier im Unterschied zur analogen Problemstellung bei polygonalen reziproken Tempofunktionen nicht auftreten. Der Rechenaufwand ist allerdings durch die benötigten iterativen Verfahren zur Nullstellenbestimmung erheblich grösser. Ferner sind bei polygonalem Tempo die in meinem Paper untersuchten Zeitflussmodifikationen $e(E_i) \rightarrow e(E_i) + z_i$ nicht unabhängig von einer zugrundeliegenden Referenzinterpretation beschreibbar. Bei Änderung der zugrundeliegenden Referenztempofunktion müssen die den obigen Zeitflussmodifikationen entsprechenden Tempomodifikationen neu berechnet werden.