Optimal approximate conversion of spline curves and spline approximation of offset curves

J Hoschek and N Wissel*

The paper introduces an effective method for approximate spline conversion. The method uses mainly parameter transformations and nonlinear optimization techniques. Geometric continuity conditions are used as parameter invariant spline conditions. For geometric continuity of order 1, 2, 3, 4, algorithms are introduced for approximative reducing of the polynomial degree of a given spline segment (and splitting into as few spline segments as possible) or elevating the polynomial degree (and merging as many spline segments as possible). The method is extended to spline approximation of offset curves (and splitting into as few new spline segements as possible).

computer-aided geometric design, curves, algorithms, optimization, parametrization, geometric continuity, approximation

Most computer-aided design systems for free-form curves and surfaces modelling use parametric polynomial representation with different polynomial bases and maximum polynomial degrees. Therefore there is a need for communication, and the exchange of data between different systems, using an effective method for approximate conversion of spline representation. Conversion from one polynomial base to another can be achieved by direct matrix multiplication whenever the number and degrees of polynomial terms in both representations are equal. In this case any loss of accuracy stems from numerical noise. If two systems do not allow for the same maximum polynomial degrees, then approximate conversions of high order functions into low order functions (reducing combined with splitting spline segments) and perhaps vice versa (elevating and merging spline segments) are inevitable. This causes approximation errors, which must be minimized. Dannenberg and Nowacki[1] first introduced an approach that uses an error estimate due to de Boor[2] and an application of this estimate due to Hölzle[3], while Hoschek[4] has proposed a conversion method using geometric continuity of order 1 and order 2 and parameter optimization. In this paper this method is extended to geometric continuity of order 3 and 4 (i.e. reduction to polynomial degree 3, 5, 7, and 9). Furthermore, a more effective nonlinear optimization algorithm and a spline-splitting algorithm are introduced, which leads to a small number of spline segments for degree reducing. While Hoschek[4] used the cyclic coordinate ascent as the optimization algorithm, which is simple to use but leads only to local minima[5], now optimization algorithms are used that evaluate the global minimum with help of approximation of the gradients of the objective function. Thus difficult situations with an extreme variation of curvature (see Figure 2) or with loops (see Figure 3) can also be approximated. Furthermore, the procedure is used for merging more than one spline segment to one segment and for spline approximation of offset curves. The approximation process can be extended to approximate spline conversion of surfaces.

Fachbereich Mathematik, Technische Hochschule Darmstadt, 6100 Darmstadt, Schlossgartenstr. 7, FRG
* Daimler-Benz AG, Abt. A1DK, D-7031 Sindelfingen, FRG

GEOMETRIC CONTINUITY CONDITIONS

The key idea of the proposed method is to use the parametrization as a design parameter: the shape of an approximation curve of a set of points will be changed if the parameter values of the points are changed during the approximation process. If it is required to change parametrization during the design process, spline conditions must be used that are invariant to the parametrization. Osculating conditions can be used that are well known in differential geometry[6,7] as contact of order k of two curves or surfaces. In computer-aided geometric design these conditions are denoted as geometric continuity of order k (GC^k continuity).

Two curves X and Y have GC^k continuity if the following conditions hold at a common point of X and Y

$k = 1$:

$$X^{I} = r_1 Y^{I}$$

$k = 2$:

$$X^{II} = r_1^2 Y^{II} + r_2 Y^{I} \text{ (conditions for } k = 1)$$

 0010–4485/88/080475–09 $03.00

$k = 3$:

$$X^{III} = r_1^3 Y^{III} + 3r_1 r_2 Y^{II} + r_3 Y^{I} \text{ (conditions for } k = 1, 2)$$

$k = 4$:

$$X^{IV} = r_1^4 Y^{IV} + 6r_1^2 r_2 Y^{III} + (3r_2^2 + 4r_1 r_3) Y^{II} + r_4 Y^{I}$$

$$\text{(conditions for } k = 1, 2, 3) \quad (1)$$

with arbitrarily chosen parameters r_i. These conditions can be developed by using an osculating algebraic curve of degree k or out of the first k terms of the Taylor expansion. The geometric continuity conditions are also used by Barsky to develop Beta-splines[8]. Barsky and de Rose deduced the conditions (1) with help of the chain rule[9]. The geometric splines introduced by Boehm[10] or Hagen[11] use special cases of condition (1) as continuity conditions. Nielson[12] first used generalization of continuity conditions defined as differentiability.

A geometric interpretation of the geometric continuity can be given as follows. If two curves have geometric continuity of order two at a common point, they have the same curvature at the common point; if two curves have geometric continuity of order 3 at a common point, they have the same torsion at the common point for space curves or a common cubic osculating parabola for plane curves.

For the algorithms developed here, Bézier techniques will be used. Other methods, such as B-spline techniques, follow easily by base transformations. The results can be interpreted as a Bézier representation of Beta-splines.

The given curve (spline segment) X may have the parametric representation

$$X = \sum_{i=0}^{n} \mathbf{V}_i B_i^n(t) \qquad t \in [0, 1] \quad (2)$$

with Bernstein polynomials $B_i^n(t)$ of degree n and $\mathbf{V}_i$ as given Bézier points. The required curve Y may be a Bézier curve of degree $m (m < n)$ and may have the parametric representation

$$Y = \sum_{i=0}^{m} \mathbf{W}_i B_i^m(\tau) \qquad \tau \in [0, 1] \quad (3)$$

with unknown Bézier points $\mathbf{W}_i$. m is bound by $m > 2k + 1$, with k as the order of the continuity conditions. To transform the conditions (1) by (2) into boundary conditions for the points $X(0) = Y(0)$, $X(1) = Y(1)$, the following abbreviations are introduced

$$\omega_1 = \frac{m(n-1)}{n(m-1)}$$

$$\omega_2 = \frac{m^2(n-1)(n-2)}{n^2(m-1)(m-2)} \quad (4)$$

$$\omega_3 = \frac{m^3(n-1)(n-2)(n-3)}{n^3(m-1)(m-2)(m-3)}$$

After some calculations the following conditions are obtained for the unknown Bézier points $\mathbf{W}_i$

$k = 1$:

$$\begin{aligned} \mathbf{W}_0 &= \mathbf{V}_0 \\ \mathbf{W}_1 &= \mathbf{V}_0 + (\mathbf{V}_1 - \mathbf{V}_0)\lambda_1 \end{aligned} \quad (5)$$

$$\begin{aligned} \mathbf{W}_m &= \mathbf{V}_n \\ \mathbf{W}_{m-1} &= \mathbf{V}_n + (\mathbf{V}_{n-1} - \mathbf{V}_n)\mu_1 \end{aligned}$$

$k = 2$:

$$\begin{aligned} \mathbf{W}_2 &= \mathbf{V}_0 + (\mathbf{V}_2 - \mathbf{V}_1)\lambda_1^2\omega_1 \\ &\quad + (\mathbf{V}_1 - \mathbf{V}_0)\lambda_2 \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbf{W}_{m-2} &= \mathbf{V}_n + (\mathbf{V}_{n-2} - \mathbf{V}_{n-1})\mu_1^2\omega_1 \qquad \text{(with (5))} \\ &\quad + (\mathbf{V}_{n-1} - \mathbf{V}_n)\mu_2 \end{aligned}$$

(for corresponding relations for $k = 3$, $k = 4$ see Appendix 1).

APPROXIMATION ERROR FOR DEGREE REDUCING

The goal is to approximate the given Bézier curve X by a Bézier curve Y optimally, where optimally means minimizing the square error sum. The position error will be measured at $s + 1$ points $\mathbf{P}_i$ of the given Bézier curve X of degree $n (s > n$, for example $s = 2n)$ with the (equidistant) parameter values t_i

$$\mathbf{P}_i = X(t_i) = X\left(\frac{i}{s}\right) \qquad (i = 0, \ldots, s),\ \{i\} =: I \quad (7)$$

If these parameter values are inserted into the required Bézier curve Y, then the following are obtained as error vectors

$$\boldsymbol{\delta}_i = \mathbf{P}_i - Y(t_i) \quad (8)$$

and as square error sum

$$\delta = \sum_{i=0}^{s} \boldsymbol{\delta}_i^2 \quad (9)$$

For the different continuity conditions, the error vectors are determined by (with conditions (4)–(6))

$k = 1$ and $m = 3$:

$$\boldsymbol{\delta}_i = \mathbf{R}_i - (\mathbf{V}_1 - \mathbf{V}_0)\lambda_1 B_1^3(t_i) - (\mathbf{V}_{n-1} - \mathbf{V}_n)\mu_1 B_2^3(t_i) \quad (10)$$

with

$$\mathbf{R}_i = \mathbf{P}_i - \mathbf{V}_0(B_0^3(t_i) + B_1^3(t_i)) - \mathbf{V}_n(B_2^3 + B_3^3(t_i))$$

$k = 2$ and $m = 5$:

$$\begin{aligned} \boldsymbol{\delta}_i = \mathbf{R}_i &- (\mathbf{V}_1 - \mathbf{V}_0)(\lambda_1 B_1^5(t_i) + \lambda_2 B_2^5(t_i)) \\ &- (\mathbf{V}_2 - \mathbf{V}_1)\lambda_1^2\omega_1 B_2^5(t_i) \\ &- (\mathbf{V}_{n-1} - \mathbf{V}_n)(\mu_1 B_4^5(t_i) + \mu_2 B_3^5(t_i)) \\ &- (\mathbf{V}_{n-2} - \mathbf{V}_{n-1})\mu_1^2\omega_1 B_3^5(t_i) \end{aligned} \quad (11)$$

$$\begin{aligned} \mathbf{R}_i = \mathbf{P}_i &- \mathbf{V}_0(B_0^5(t_i) + B_1^5(t_i) + B_2^5(t_i)) \\ &- \mathbf{V}_n(B_3^5(t_i) + B_4^5(t_i) + B_5^5(t_i)) \end{aligned}$$

(for corresponding relations for $k=3$, $k=4$, see Appendix 2).

In equations (10) and (11) only the case $m = 2k + 1$ is considered. If $m > 2k + 1$ the undetermined inner Bézier points $\mathbf{W}_i$ are additional unknown variables.

OPTIMIZATION ALGORITHM

As only the case $m = 2k + 1$ will be considered, therefore there are $2k$ unknowns $\lambda_1, \ldots, \lambda_k$, $\mu_1, \ldots, \mu_k$ with λ_i, $\mu_i > 0$, hence the permissible domain is out of $\mathcal{R}^{2k}$. During the optimization process, the square error sum δ in equation (9) has to be minimized. The received square error sum depends on parametrization, hence in general the error vectors $\boldsymbol{\delta}_i$ are not orthogonal to the approximation curve Y. To obtain error vectors $\boldsymbol{\delta}_i$ approximately normal to the approximation curve $Y(t)$, a parameter correction is used as proposed by Hoschek[4]. During this parameter correction, the perpendiculars from the given points $\mathbf{P}_i$ to the approximation curve $Y(t)$ are approximated by a set of parameter correction.

A numerical optimization algorithm will be used that approximates the gradients of the objective function and leads to a global minimum (see Jakob[13] and the procedures Globex and Extrem). It is obvious that any other global-working numerical optimization algorithm can be used.

The optimization works in the following steps.

- (0) Choose number s of points $\mathbf{P}_i$, e.g. $s = 2n$,
  error ε_0, limit L,
  degree m of approximation curve,
  continuity condition k with $m = 2k + 1$.
- (1) Compute $\mathbf{P}_i = X(t_i)$ at $t_i = \frac{i}{s}$, $i = 0(1)s$;
  Set index $J = 1$.
- (2) Compute $\lambda_1, \ldots, \lambda_k$, $\mu_1, \ldots, \mu_k$ (λ_i, $\mu_i > 0$) with help of a global nonlinear numerical approximating algorithm;
  From (1) and (3) follows the approximation curve $Y(t)$.
- (3) For $i = 0(1)s$
  Find an improved parametrization t_i^* of $\mathbf{P}_i$ using Hoschek[4];
  Find deviation $\varepsilon_1 = \max|\mathbf{P}_i - Y(t_i)|$;
  Set $t_i = t_i^*$
- (4) Set $J = J + 1$.
- (5) If $\varepsilon_1 < \varepsilon_0$ then Stop;
  else if $j < L$ then goto 2;
  else split the given curve into two segments and goto 1 (and follow the algorithm for both segments).

SPLITTING ALGORITHM

The degree reduction is finished if the maximum error is $\varepsilon_1 = \max_{i \in I} |\boldsymbol{\delta}_i| < \varepsilon_0$ with ε_0 as given error tolerance.

If $\varepsilon_1 > \varepsilon_0$ the given curve must be split in more segments. For splitting use the following strategy. A curve of polynomial degree p can have at most $2p - 4$ inflection points, including imaginary ones. To obtain generic spline curves, suppose that a cubic curve has no more than one inflection point, a quintic curve has no more than three inflection points, or a spline curve of degree p has no more than $p - 2$ inflection points. If a given curve $X(t)$ of degree n is approximated by a curve $Y(t)$ of degree m, therefore in a first step the given curve is split in segments with at most $m - 2$ inflection points. Then each of these segments will be approximated by one curve $Y(t)$ of degree m. If the given curve $X(t)$ has less than $m - 2$ inflection points in the first step no splitting will be carried out. If the error ε_1 of one of these segments exceeds the given error tolerance ε_0 this segment will be split again at the parameter value $t = 1/2$. For splitting the spline segments the de Casteljau algorithm is used, and the new spline segments are transposed with geometric continuity of order k.

EXAMPLES

The examples use the following scheme. the given curve and the approximation curve are plotted one above the other, the given curve and the corresponding Bézier polygon (Bézier points marked by triangles) are drawn by broken lines, and the approximation curve and the corresponding Bézier polygon (Bézier points marked by boxes) are drawn by full lines. The boundary points of the spline segments are marked by boxes with crosses. Figure 1 contains a curve of degree 19 and its approximation of degree 9 and geometric continuity $k = 4$ ($\varepsilon_1 = 0.0067$). The approximation of the whole curve with only one Bézier curve of degree 7 and geometric continuity $k = 3$ would lead to $\varepsilon_1 = 0.035$; if $\varepsilon_0 = 0.008$ and $k = 3$ were chosen for the approximation, two segments would be necessary. If the same curve is approximated by Bézier spline curves of degree 5 and geometric continuity $k = 2$ and a given error tolerance $\varepsilon = 0.002$, three segments are needed. To give an impression of the goodness of fit of the approximation curve out of Figure 1, Figure 2 represents the curvature κ of the given curve and the approximation curve. It can be seen that the curvature of the given curve does not differ too much from the curvature of the approximation curve. Figure 3 contains an approximation of a given Bézier curve of degree 30 and geometric continuity $k = 4$ (degree of approximation curve 9) by seven segments ($\varepsilon_1 = 0.097$). The regions with an extreme variation of the curvature are approximated by small spline segments. A further example is shown in Figure 4: the given Bézier curve of degree 19 has a loop and is approximated by two Bézier segments of degree 9 and geometric continuity $k = 4$ ($\varepsilon_1 = 0.0044$). Figure 5 demonstrates that even for fast-changing curvature the approximating curve has nearly the same curvature as the given curve.

MERGING OF SPLINE SEGMENTS AND DEGREE ELEVATING

The introduced methods can also be used for

- (a) merging spline segments, where the given spline segments and the merged spline segment have the same degree ($m = n$)

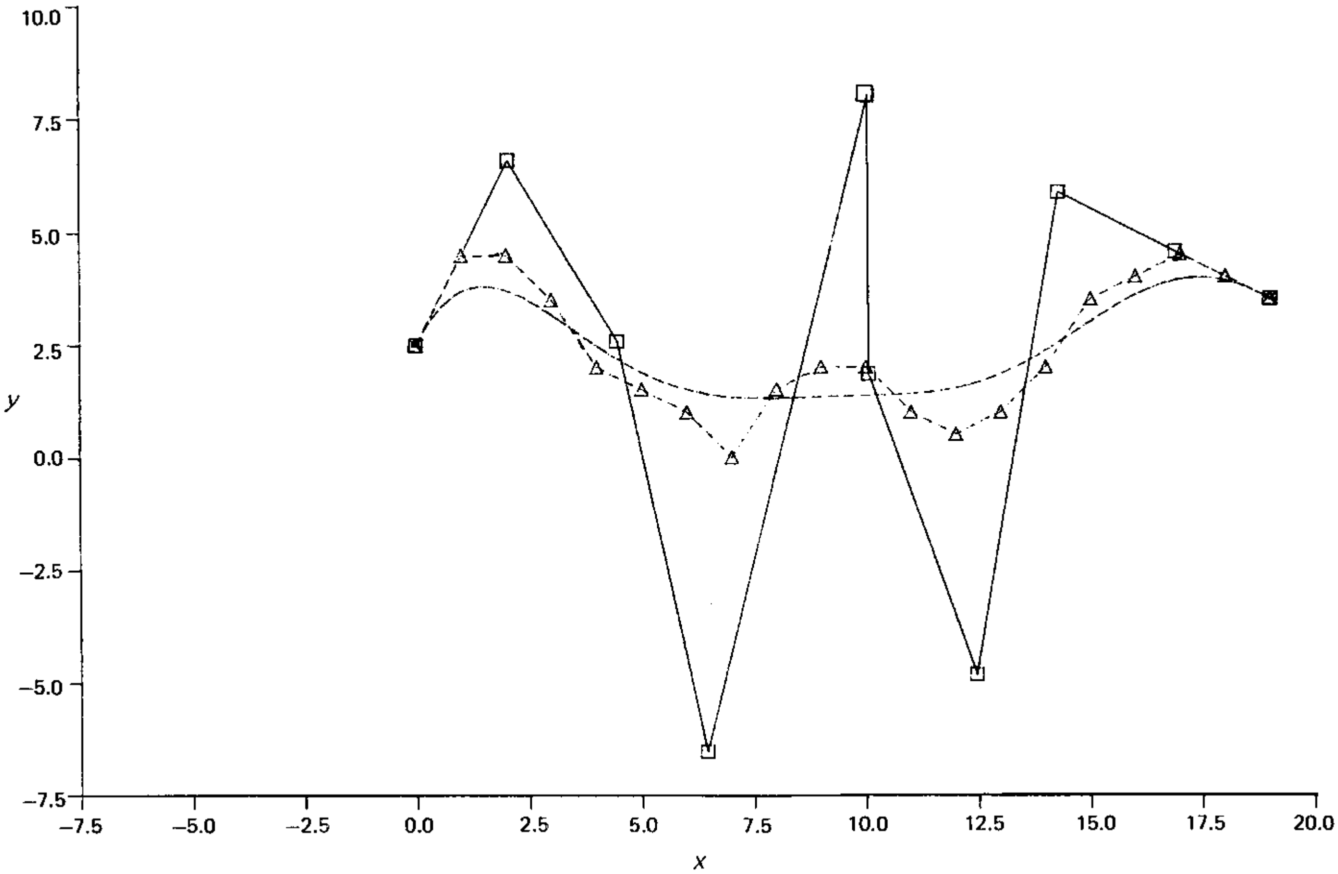

Figure 1. Bézier curve of degree 19 approximated by one Bézier curve of degree 9 and geometric continuity of order 4 (—— = given quantities, - - - = approximations)

- (b) merging spline segments directly combined with degree elevation or degree reduction where the given spline segments have degree n and the new spline segment has degree m with $m \neq n$

It is important to tell the difference between the two approximation processes (a) and (b) with following precise degree elevation. In approximation process (b) for degree elevation the higher continuity conditions at the boundary points lead to another (better) result than process (a) and following precise degree elevation.

All conditions developed in conditions (5)–(6) and the section 'Approximation error for degree reducing' can be used in the same way. Figure 6 contains an example of merging combined with degree elevation: four Bézier spline curves of degree 3 are merged to one Bézier curve of degree 5 and continuity condition with $k = 2$. As error value is obtained $\varepsilon_1 = 0.0016$.

SPLINE APPROXIMATION OF OFFSET CURVES

Now the results are transferred to spline approximation of offset curves. Suppose that the given curve is a Bézier curve of degree n and has the parametric representation $X = X(t)$, then the corresponding offset curve X_d at (oriented) distance d along the unit normal vector $\mathbf{n}(t)$ is given by[14–16]

$$X_d(t) = X(t) + \mathbf{n}(t)d \qquad (12)$$

For plane curves the normal vector $\mathbf{n}$ has the representation

$$\mathbf{n}(t) = \frac{(-\dot{y}(t), \dot{x}(t)}{(\dot{x}^2 + \dot{y}^2)^{1/2}} \qquad (13)$$

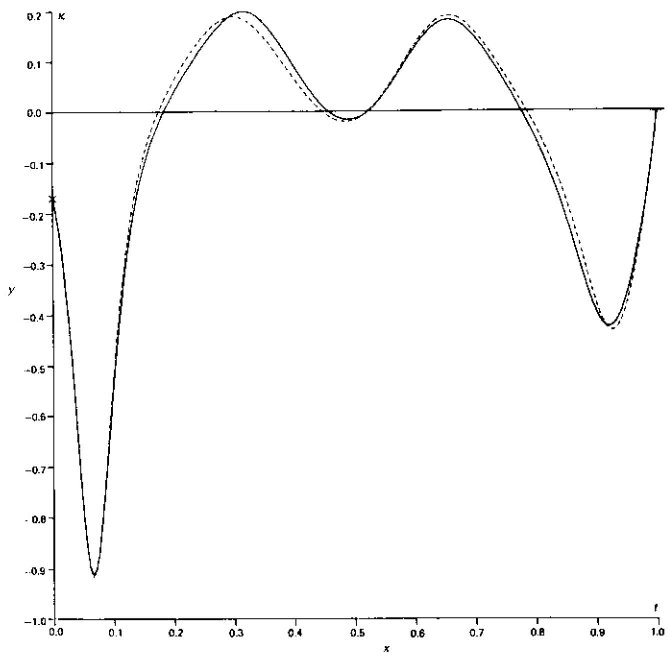

Figure 2. Curvature κ of given (- - -) and approximating (——) curves shown in Figure 1

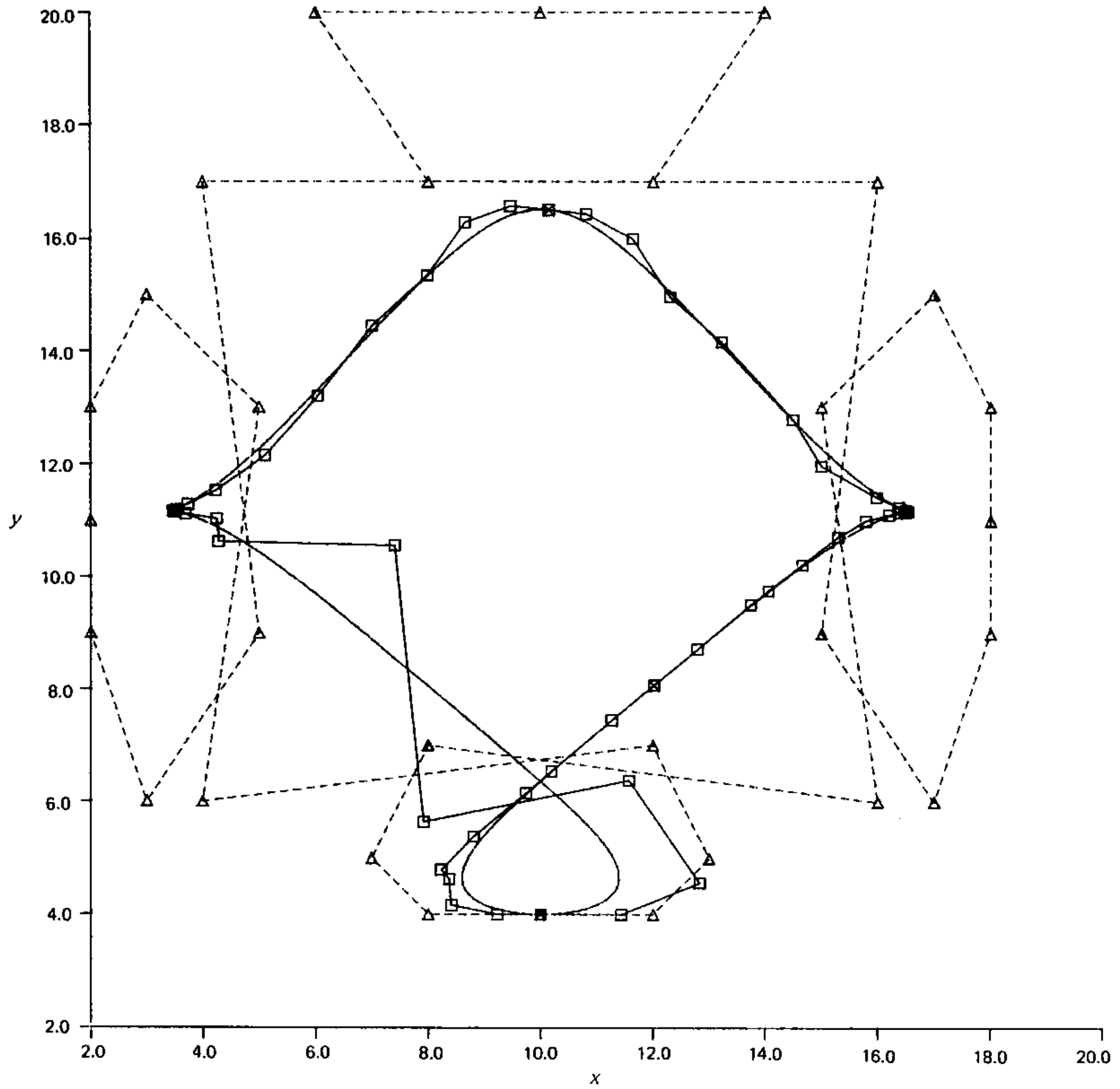

Figure 3. Closed Bézier curve of degree 30 and approximating Bézier spline segments of degree 9 ($k = 4$) (--- = given quantities, —— = approximations)

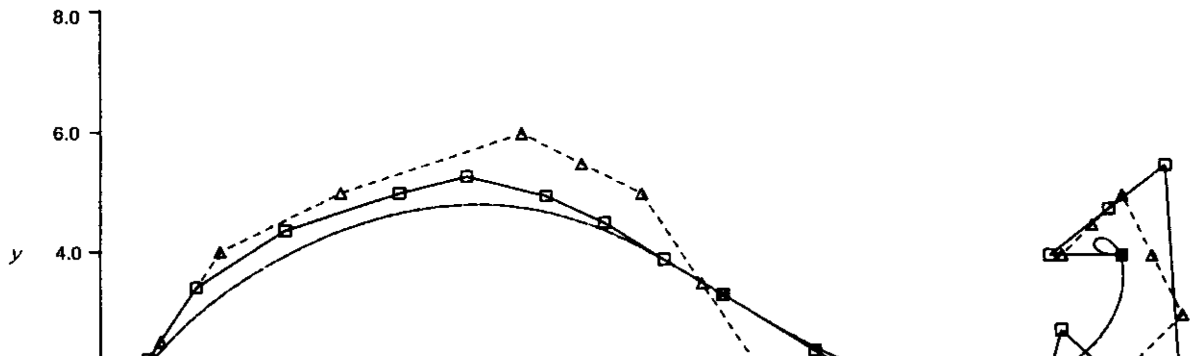

Figure 4. Bézier curve of degree 19 with one loop and approximation by two Bézier segments of degree 9 and geometric continuity $k = 4$ (--- = given quantities, —— = approximations)

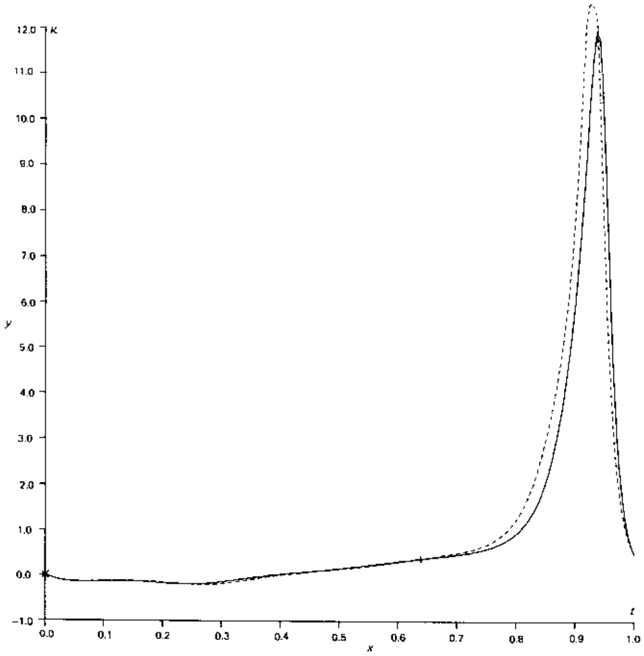

Figure 5. Curvature κ of given (– – –) and approximating (——) curves shown in Figure 4

The normal vector (13) specifies a unique side of X on which the offset d is performed, and the opposite side can be determined by adopting a negative offset magnitude d.

To obtain the boundary conditions for GC^k approximation insert X_d out of condition (12) in condition (1) instead of X. After corresponding evaluations, conditions are obtained analogously to condition (7). The new conditions can be transformed out of condition (7) if the following are exchanged (for $k = 1$, $k = 2$, $k = 3$)

$$\omega_1 \text{ by } \boldsymbol{\Omega}_1 = \omega_1 \frac{1}{1+\kappa(0)d} (\text{at } X_d(0))$$

$$M_1 = \omega_1 \frac{1}{1+\kappa(1)d} (\text{at } X_d(1))$$

$$\omega_2 \text{ by } \boldsymbol{\Omega}_2 = \omega_2 \frac{1}{(1+\kappa(0)d)^2} (\text{at } X_d(0))$$

$$M_2 = \omega_2 \frac{1}{(1+\kappa(1)d)^2} (\text{at } X_d(1))$$

additionally, the factors of $\lambda_1{}^3$ resp. $\mu_1{}^3$ must be completed by addends

$$\boldsymbol{\Omega}_2 \frac{\kappa'(0)d}{(n-2)(1+\kappa(0)d)} \text{ resp. } M_2 \frac{\kappa'(1)d}{(n-2)(1+\kappa(1)d)}$$

with $\kappa(i)$ $(i = 0, 1)$ as curvature in the boundary points and $\kappa'(i)$ as the derivatives of $\kappa(i)$. Otherwise all algorithms can be used as described in conditions (5)–(6) and in the section 'Approximation error for degree reducing'.

Figure 7 shows the Bézier spline approximation of a given Bézier curve of degree 5 (the middle curve) and the approximation of two offset curves of degree 5 and geometric continuity of order $k = 2$. The lower curve contains two segments ($\varepsilon_1 = 0.0094$), while the upper curve (with cusps) contains seven segments ($\varepsilon_1 = 0.0070$). If $k = 1$ the upper curve can be approximated by four segments ($\varepsilon_1 = 0.0027$), and if $k = 3$ four segments are obtained with $\varepsilon_1 = 0.0082$. The curve with the cusps can be approximated by 10 spline segments of degree 3 ($k = 1$). In some applications it is necessary to cancel the part of the offset curve with the cusps. Then algorithms can be used as described by Hoschek[4].

APPENDIX 1

Conditions for the unknown Bézier points $\mathbf{W}_i$ for $k = 3$, $k = 4$:

$k = 3$:

$$\begin{aligned}\mathbf{W}_3 = \mathbf{V}_0 &+ (\mathbf{V}_3 - \mathbf{V}_2)\lambda_1^3\omega_2 \\ &+ (\mathbf{V}_2 - \mathbf{V}_1)\left[\lambda_1^3\left(-2\omega_2 + 3\omega_1^2\frac{m-1}{m-2}\right)\right. \\ &\qquad + \lambda_1^2\omega_1\left(3 - 6\frac{m-1}{m-2}\right) \\ &\qquad \left. + 3\lambda_1\lambda_2\omega_1\frac{m-1}{m-2}\right] \\ &+ (\mathbf{V}_1 - \mathbf{V}_0)\lambda_3 \qquad \text{(with conditions (5) and (6)}\end{aligned} \tag{14}$$

$$\begin{aligned}\mathbf{W}_{m-3} = \mathbf{V}_n &+ (\mathbf{V}_{n-3} - \mathbf{V}_{n-2})\mu_1^3\omega_2 \\ &+ (\mathbf{V}_{n-2} - \mathbf{V}_{n-1}) \\ &\qquad \left[\mu_1^3\left(-2\omega_2 + 3\omega_1^2\frac{m-1}{m-2}\right)\right. \\ &\qquad + \mu_1^2\omega_1\left(3 - 6\frac{m-1}{m-2}\right) \\ &\qquad \left. + 3\mu_1\mu_2\omega_1\frac{m-1}{m-2}\right] \\ &+ (\mathbf{V}_{n-1} - \mathbf{V}_n)\mu_3\end{aligned}$$

$k = 4$:

$$\begin{aligned}\mathbf{W}_4 = \mathbf{V}_0 &+ (\mathbf{V}_4 - \mathbf{V}_3)\lambda_1^4\omega_3 \\ &+ (\mathbf{V}_3 - \mathbf{V}_2)\lambda_1^4 a_{40} + \lambda_1^3 a_{30} + \lambda_1^2\lambda_2 a_{21}\} \\ &+ (\mathbf{V}_2 - \mathbf{V}_1) \\ &\qquad \{\lambda_1^4 b_{400} + \lambda_1^3 b_{300} + \lambda_1^2 b_{200} + \lambda_2^2 b_{020} \\ &\qquad + \lambda_1^2\lambda_2 b_{210} + \lambda_1\lambda_2 b_{110} + \lambda_1\lambda_3 b_{101}\} \\ &+ (\mathbf{V}_1 - \mathbf{V}_0)\lambda_4 \qquad \text{(with conditions (5), (6), and (14))}\end{aligned} \tag{15}$$

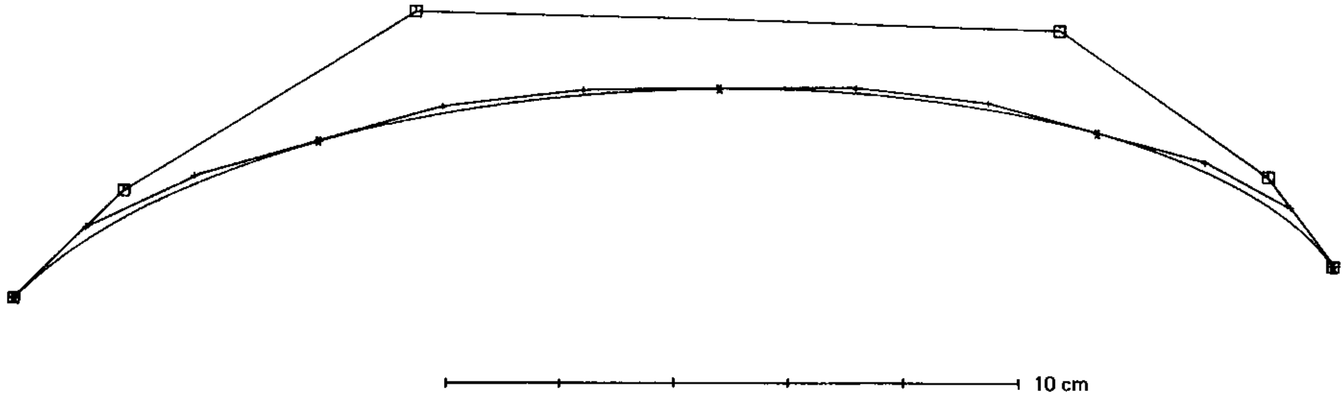

Figure 6.Four Bézier spline curves of degree 3 are merged to one Bézier curve of degree 5 (k = 2). Bézier polygons of given curves and approximation curve are marked.

$$\mathbf{W}_{m-4} = \mathbf{V}_n + (\mathbf{V}_{n-4} - \mathbf{V}_{n-3})\mu_1^4\omega_3$$
$$+ (\mathbf{V}_{n-3} - \mathbf{V}_{n-2})$$
$$\{\mu_1^4 a_{40} + \mu_1^3 a_{30} + \mu_1^2\mu_2 a_{21}\}$$
$$+ (\mathbf{V}_{n-2} - \mathbf{V}_{n-1})$$
$$\{\mu_1^4 b_{400} + \mu_1^3 b_{300} + \mu_1^2 b_{200} + \mu_2^2 b_{020}$$
$$+ \mu_1^2\mu_2 b_{210} + \mu_1\mu_2 b_{110} + \mu_1\mu_3 b_{101}\}$$
$$+ (\mathbf{V}_{n-1} - \mathbf{V}_n)\mu_4$$

The quantities λ_i, μ_i are arbitrarily chosen new parameters (which are abbreviations for the coefficients generated during calculation) and the coefficients a_{ij}, b_{ijk} are determined by

$$a_{40} = 6\frac{m-1}{m-3}\omega_1\omega_2 - 3\omega_3$$

$$a_{30} = \left(4 - 12\frac{m-1}{m-3}\right)\omega_2$$

$$a_{21} = 6\frac{m-1}{m-3}\omega_2$$

$$b_{400} = 3\omega_3 - 16\frac{m-1}{m-3}\omega_1\omega_2 + 15\frac{(m-1)^2}{(m-2)(m-3)}\omega_1^3$$

$$b_{300} = 12\frac{m-1}{m-2}\omega_1^2 + 24\frac{m-1}{m-3}\omega_2 - 8\omega_2$$
$$- 36\frac{(m-1)^2}{(m-2)(m-3)}\omega_1^2$$

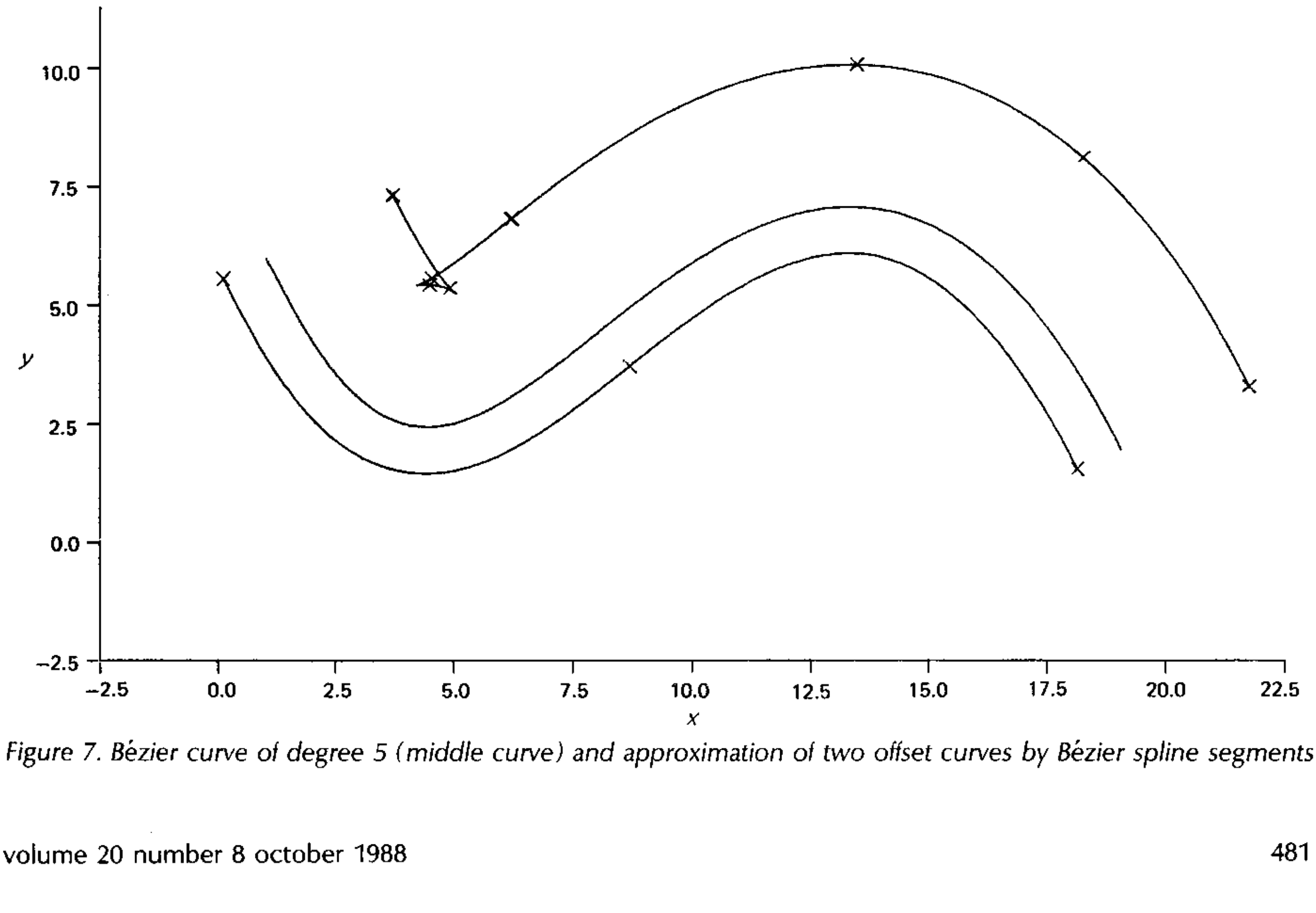

Figure 7. Bézier curve of degree 5 (middle curve) and approximation of two offset curves by Bézier spline segments

$$b_{200} = \omega_1\left(6 + 12\frac{m-1}{m-3} - 24\frac{m-1}{m-2} + 12\frac{(m-1)^2}{(m-2)(m-3)}\right)$$

$$b_{020} = 3\frac{(m-1)^2}{(m-2)(m-3)}\omega_1$$

$$b_{210} = 18\frac{(m-1)^2}{(m-2)(m-3)}\omega_1^2 - 12\frac{m-1}{m-3}\omega_2$$

$$b_{110} = 12\frac{m-1}{m-2}\omega_1 - 12\frac{m-1}{m-3}\omega_1 - 12\frac{(m-1)^2}{(m-2)(m-3)}\omega_1$$

$$b_{101} = 4\frac{m-1}{m-3}\omega_1$$

APPENDIX 2

Error vectors for continuity conditions with $k = 3$, $k = 4$:

$k = 3$ and $m = 7$:

$$\begin{aligned}
\boldsymbol{\delta}_i = \mathbf{R}_i &- (\mathbf{V}_1 - \mathbf{V}_0)(\lambda_1 B_1^7(t_i) + \lambda_2 B_2^7(t_i) + \lambda_3 B_3^7(t_i)) \\
&- (\mathbf{V}_2 - \mathbf{V}_1)\Bigg(\lambda_1^2\omega_1 B_2^7(t_i) \\
&+ \Bigg(\lambda_1^3\left(3\omega_1^2\frac{m-1}{m-2} - 2\omega_2\right) \\
&+ \lambda_1^2\omega_1\left(3 - 6\frac{m-1}{m-2}\right) \\
&+ 3\lambda_1\lambda_2\omega_1\frac{m-1}{m-2}\Bigg)B_3^7(t_i)\Bigg) \\
&- (\mathbf{V}_3 - \mathbf{V}_2)\lambda_1^3\omega_2 B_3^7 \qquad (16)\\
&- (\mathbf{V}_{n-1} - \mathbf{V}_n)(\mu_1 B_6^7(t_i) + \mu_2 B_5^7(t_i) + \mu_3 B_4^7(t_i)) \\
&- (\mathbf{V}_{n-2} - \mathbf{V}_{n-1})\Bigg(\mu_1^2\omega_1 B_5^7(t_i) \\
&+ \Bigg(\mu_1^3\left(3\omega_1^2\frac{m-1}{m-2} - 2\omega_2\right) \\
&+ \mu_1^2\omega_1\left(3 - 6\frac{m-1}{m-2}\right) \\
&+ 3\mu_1\mu_2\omega_1\frac{m-1}{m-2}\Bigg)B_4^7(t_i)\Bigg) \\
&- (\mathbf{V}_{n-3} - \mathbf{V}_{n-2})\mu_1^3\omega_2 B_4^7(t_i)
\end{aligned}$$

with

$$\begin{aligned}
\mathbf{R}_i = \mathbf{P}_i &- \mathbf{V}_0(B_0^7(t_i) + B_1^7(t_i) + B_2^7(t_i) + B_3^7(t_i)) \\
&- \mathbf{V}_n(B_4^7(t_i) + B_5^7(t_i) + B_6^7(t_i) + B_7^7(t_i))
\end{aligned}$$

$k = 4$ and $m = 9$:

$$\begin{aligned}
\boldsymbol{\delta}_i = \mathbf{R}_i &- (\mathbf{V}_1 - \mathbf{V}_0)(\lambda_1 B_1^9(t_i) + \lambda_2 B_2^9(t_i) + \lambda_3 B_3^9(t_i) \\
&+ \lambda_4 B_4^9(t_i)) - (\mathbf{V}_2 - \mathbf{V}_1)\Bigg(\lambda_1^2\omega_1 B_2^9(t_i) \\
&+ \Bigg(\lambda_1^3\left(3\omega_1^2\frac{m-1}{m-2} - 2\omega_2\right) \\
&+ \lambda_1^2\omega_1\left(3 - 6\frac{m-1}{m-2}\right) + 3\lambda_1\lambda_2\omega_1\frac{m-1}{m-2}\Bigg) \\
&B_3^9(t_i) + (\lambda_1^4 b_{400} + \lambda_1^3 b_{300} + \lambda_1^2 b_{200} + \lambda_2^2 b_{020} \qquad (17)\\
&+ \lambda_1^2\lambda_2 b_{210} + \lambda_1\lambda_2 b_{110} + \lambda_1\lambda_3 b_{101})B_4^9(t_i)\Bigg) \\
&- (\mathbf{V}_3 - \mathbf{V}_2)(\lambda_1^3\omega_2 B_3^9(t_i) \\
&+ (\lambda_1^4 a_{40} + \lambda_1^3 a_{30} + \lambda_1^2\lambda_2 a_{21})B_4^9(t_i)) \\
&- (\mathbf{V}_4 - \mathbf{V}_3)\lambda_1^4\omega_3 B_4^9(t_i) - (\mathbf{V}_{n-1} - \mathbf{V}_n)(\mu_1 B_8^9(t_i) \\
&+ \mu_2 B_7^9(t_i) + \mu_3 B_6^9(t_i) + \mu_4 B_5^9(t_i)) \\
&- (\mathbf{V}_{n-2} - \mathbf{V}_{n-1})\Bigg(\mu_1^2\omega_1 B_7^9(t_i) \\
&+ \Bigg(\mu_1^3\left(3\omega_1^2\frac{m-1}{m-2} - 2\omega_2\right) \\
&+ \mu_1^2\omega_1\left(3 - 6\frac{m-1}{m-2}\right) \\
&+ 3\mu_1\mu_2\omega_1\frac{m-1}{m-2}\Bigg)B_6^9(t_i)\Bigg) \\
&+ (\mu_1^4 b_{400} + \mu_1^3 b_{300} + \mu_1^2 b_{200} + \mu_2^2 b_{020} \\
&+ \mu_1^2\mu_2 b_{210} + \mu_1\mu_2 b_{110} + \mu_1\mu_3 b_{101})B_5^9(t_i) \\
&- (\mathbf{V}_{n-3} - \mathbf{V}_{n-2})(\mu_1^3\omega_2 B_6^9(t_i) \\
&+ (\mu_1^4 a_{40} + \mu_1^3 a_{30} + \mu_1^2\mu_2 a_{21})B_5^9(t_i)) \\
&- (\mathbf{V}_{n-4} - \mathbf{V}_{n-3})\mu_1^4\omega_3 B_5^9(t_i)
\end{aligned}$$

with

$$\begin{aligned}
\mathbf{R}_i = \mathbf{P}_i &- \mathbf{V}_0(B_0^9(t_i) + B_1^9(t_i) + B_2^9(t_i) + B_3^9(t_i) + B_4^9(t_i)) \\
&- \mathbf{V}_n(B_5^9(t_i) + B_6^9(t_i) + B_7^9(t_i) + B_8^7(t_i) + B_9^9(t_i))
\end{aligned}$$