Journal of Computational Mathematics, Vol.26, No.1, 2008, 69–75.

REAL ROOT ISOLATION OF SPLINE FUNCTIONS*

Renhong Wang and Jinming Wu
Institute of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
Email: renhong@dlut.edu.cn, wujm97@yahoo.com.cn

Abstract

In this paper, we propose an algorithm for isolating real roots of a given univariate spline function, which is based on the use of Descartes' rule of signs and de Casteljau algorithm. Numerical examples illustrate the flexibility and effectiveness of the algorithm.

Mathematics subject classification: 65D07, 14Q05.
Key words: Real root isolation, Univariate spline, Descartes' rule of signs, de Casteljau algorithm.

1. Introduction

The relationship between the number of real roots of a univariate spline and the sequence of its B-spline coefficients has been studied by de Boor [1] and Goodman [5], which provides a new bounds on the number of real roots of the spline function. However, the specific distribution of real roots of a given univariate spline based on its signs and sizes of B-spline coefficients have not been investigated. The specific distribution can provide a good selection of initial approximations to all of its real roots in order to get started for iterative methods.

In 1989, Grandine [6] proposed a method for finding all real roots of a spline function based on the interval Newton method. It is primarily based on iteratively dividing the domain into segments that contain a zero, by using estimates for the derivatives of the spline function based on knot insertion. However, if we know the isolating intervals of a given spline function, then it will greatly reduce the computational cost for finding all of its real roots.

It is well known that there are several algorithms for polynomial real root isolation based on the use of Descartes' rule of signs, such as Uspensky's algorithm (see [2, 7] and references therein). It can be regarded as a preconditioned process for computing all the real roots of a given polynomial.

In this paper, we propose an algorithm for computing a sequence of disjoint intervals such that each of them contains exactly one real root of a given univariate spline, which is primarily based on the use of Descartes' rule of signs with its B-spline coefficients and de Casteljau algorithm. Numerical examples are also provided to illustrate the flexibility of the proposed algorithm.

2. Preliminaries

We begin by defining the class of spline functions of interest [8, 9]. Take integers $m, n \geq 0$ and a non-decreasing sequence $t = (t_0, t_1, \cdots, t_{m+n+1})$ with $t_i < t_{i+n+1},\ i = 0, 1, \cdots, m$. For

* Received October 11, 2006 / Revised version received April 4, 2007 / Accepted June 6, 2007 /

$i = 0, 1, \cdots, m$, let $N_{i,n}(x)$ denote the B-spline of degree n with knots $t_i, \cdots, t_{i+n+1}$. For a constant sequence $c = (c_0, \cdots, c_m)$, we let

$$s(x) = \sum_{i=0}^{m} c_i N_{i,n}(x), \quad t_0 < x < t_{m+n+1}. \tag{2.1}$$

In [5], Goodman proved that the bounds on the number of real roots of the spline function

$$z(s) \le S(c) \tag{2.2}$$

under the following condition

$$Condition(c,t): \ \forall\, x \in (t_0, t_{m+n+1}),\ \exists\, i, \text{ s.t. } t_i < x < t_{i+n+1} \text{ and } c_i \ne 0, \tag{2.3}$$

where $z(s)$ denotes the number of real roots of the spline function $s(x)$, and $S(c)$ denotes the number of sign variations in the sequence c.

Obviously, $Condition(c,t)$ implies that $s(x)$ cannot vanish on any nontrivial interval in (t_0, t_{m+n+1}).

Let us first recall Descartes' rule of signs [7]:

Theorem 2.1. (Descartes' rule of signs) *Let $P(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial in $\mathbb{R}[x]$. If we denote by S(a) the number of sign variations in the sequence $a = (a_0, a_1, \cdots, a_n)$, and $pos(P)$ the number of positive real roots of $P(x)$ counted with multiplicities, then $pos(P) \le S(a)$, and $pos(P) - S(a)$ is even.*

We remark that Descartes' rule of signs gives the exact number of roots if and only if there is one or no sign variation.

Note that the following direct consequences of sign variations: for any real number sequence $b = (b_i, \cdots, b_j)$, if $b_i b_j > 0$, then b has an even number of sign variations. Moreover, if $b_i b_j < 0$, then b has an odd number of sign variations.

Throughout this paper, we assume $c_j = 0$ when $j < 0$ and $j > m$. For a spline function $s(x)$ defined by (2.1), we have

$$s_i(x) = s(x)\mid_{[t_i, t_{i+1}]} = \sum_{j=i-n}^{i} c_j N_{j,n}(x) \in \mathbf{P}_n,$$

where $\mathbf{P}_n$ denotes the set of all univariate polynomials with real coefficients and degree not exceeding n. Therefore, it can be written in Bézier form:

$$s_i(x) = \sum_{j=0}^{n} b_{i,j} B_{j,n}(t), \quad t \in [0,1] \tag{2.4}$$

under the coordinate transformation

$$t = \frac{x - t_i}{t_{i+1} - t_i}, \quad x \in [t_i, t_{i+1}], \tag{2.5}$$

where $B_{j,n}(t) = C_n^j\, t^j (1-t)^{n-j}$ is the Bernstein polynomial.

Recall that the Bézier curve $s_i(x)$ defined by (2.4) enjoys the variation diminishing property [4]: the curve has no more intersections with any line other than the polygon

$$P_i = \{(\frac{j}{n}, b_{i,j})\}_{j=0}^{n},$$

which provides a rudimentary upper bound estimates on the number of real roots of the Bézier curve.

In order to determine the real roots of $s_i(x)$ in the interval $x \in (t_i, t_{i+1})$, it follows from Descartes' rule of signs that

Lemma 2.1. *If we denote by $z(s_i)$ the number of real roots of $s_i(x)$ in the interval (t_i, t_{i+1}), and $b_i = (b_{i,0}, b_{i,1}, \cdots, b_{i,n})$, then $z(s_i) \le S(b_i)$. Moreover, if $S(b_i) = 0$ or $S(b_i) = 1$, then $s_i(x)$ has no root or only one root in the interval $x \in (t_i, t_{i+1})$.*

Proof. Since

$$s_i(x) = \sum_{j=0}^{n} b_{i,j} B_{j,n}(t) = (1-t)^n \sum_{j=0}^{n} d_{i,j} y^j,$$

where

$$y = \frac{t}{1-t}, \quad t \in (0,1), \quad d_{i,j} = C_n^j \ b_{i,j},$$

it follows from Descartes' rule of signs that $z(s_i) \le S(b_i)$. If $S(b_i) = 0$ or $S(b_i) = 1$, then $s_i(x)$ has no root or only one root in the interval $y \in (0, \infty)$. That is, open interval $y \in (0, \infty)$ is equivalent to $x \in (t_i, t_{i+1})$.

If $S(b_i) \ge 2$, then we have to further isolate the real roots of $s_i(x)$ in the interval $x \in (t_i, t_{i+1})$, i.e., computing a sequence of disjoint intervals such that each of them contains exactly one real root of $s_i(x)$. We introduce the following algorithm based on the use of both Descartes' rule of signs and subdivision algorithm of a given Bézier curve.

It is well known that de Casteljau algorithm is the most fundamental algorithm in the field of curve and surface design [4].

de Casteljau algorithm

Given

$$s(x) = \sum_{i=0}^{n} b_i B_{i,n}(x), \quad x \in [0,1]$$

and specify $u \in [0,1]$.

Set:

$$b_i^r(u) = (1-u) b_i^{r-1}(u) + u b_{i+1}^{r-1}(u) \quad \begin{cases} r = 1, \cdots, n \\ i = 0, \cdots, n-r \end{cases} \tag{2.6}$$

and $b_i^0(u) = b_i$. Then $b_0^n(u)$ is the function value $s(u)$ of Bézier curve $s(x)$ on u.

Meanwhile, the coefficients

$$b_l^1(u) = (b_0^0(u), b_0^1(u), \cdots, b_0^n(u)), \quad b_r^1(u) = (b_0^n(u), b_1^{n-1}(u), \cdots, b_n^0(u))$$

are called the Bézier coordinates with respect to the Bézier curves $s_l(x)$ and $s_r(x)$ defined on the intervals $[0, u]$ and $[u, 1]$, respectively.

Now, we introduce an algorithm for isolating real roots of Bézier curve $s_i(x)$ based on de Casteljau algorithm together with Lemma 2.1.

Algorithm 2.1. ALGORITHM FOR ISOLATING REAL ROOTS OF BÉZIER CURVES

Given

$$s_i(x) = \sum_{j=0}^{n} b_{i,j} B_{j,n}(t), \quad x \in (t_i, t_{i+1})$$

under the coordinate transformation (2.5), and let $b_i = (b_{i,0}, b_{i,1}, \cdots, b_{i,n})$. Here, u is simply chosen to be $\frac{1}{2}$ in de Casteljau algorithm.

1. If $S(b_i) = 0$ or $S(b_i) = 1$, then $s_i(x)$ has no root or only one root in the interval $x \in (t_i, t_{i+1})$.
2. If $S(b_i) \geq 2$, then we can obtain $b_{i,l}(u) = (b_{i,0}^0(u), b_{i,0}^1(u), \cdots, b_{i,0}^n(u))$, and $b_{i,r}(u) = (b_{i,0}^n(u), b_{i,1}^{n-1}(u), \cdots, b_{i,n}^0(u))$ by using de Casteljau algorithm. If $b_{i,0}^n(u) = 0$, then $x = \frac{t_i + t_{i+1}}{2}$ is a root of $s_i(x)$. If $S(b_{i,l}(u)) = 0$ or $S(b_{i,l}(u)) = 1$, then $s_i(x)$ has no root or only one root in the interval $x \in (t_i, \frac{t_i+t_{i+1}}{2})$. If $s(b_{i,l}(u)) \geq 2$, then $s_{i,l}(x)$ is subjected to the same subdivision. Similarly, if $S(b_{i,r}(u)) = 0$ or $S(b_{i,r}(u)) = 1$, then $s_i(x)$ has no root or only one root in the interval $x \in (\frac{t_i+t_{i+1}}{2}, t_{i+1})$. If $s(b_{i,r}(u)) \geq 2$, then $s_{i,r}(x)$ is subjected to the same subdivision.

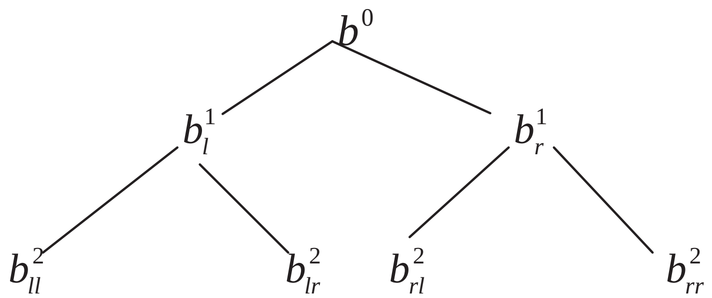

Fig. 2.1. Bézier curve real root isolation.

The implementation of Algorithm 2.1 can be represented in a form of the binary tree (see Fig. 2.1).

Since the subdivision algorithm converges quadratically [3], the algorithm will terminate rapidly, i.e., the binary tree is finite.

Remark 2.1. It can be verified that the proposed algorithm is faster than the Sturm algorithm for isolating the real roots of a given polynomial.

3. Spline Real Root Isolation

In this section, an algorithm is introduced to isolate all real roots of a given spline function. Let

$$s(x) = \sum_{i=0}^{m} c_i N_{i,n}(x), \ t_0 < x < t_{m+n+1}, \quad c = (c_0, \cdots, c_m).$$

The problem of spline real root isolation is to compute all the isolating intervals such that each of them contains exactly one root of the spline $s(x)$.

If the two adjacent non-zero coefficients c_{i_1} and c_{i_2} $(i_1 < i_2)$ have distinct signs, then the interval (t_{i_2}, t_{i_1+n+1}) is called a *feasible isolating interval*. Here, we assume that $(t_{i_2}, t_{i_1+n+1}) = \emptyset$ when $i_2 - i_1 > n$. Hence, the feasible isolating intervals of $s(x)$ are defined by

$$I = \bigcup_{\substack{sign(c_{i_1} c_{i_2})=-1, \\ c_{i_1+1}=\cdots=c_{i_2-1}=0}} (t_{i_2}, t_{i_1+n+1}), \tag{3.1}$$

where $sign(\cdot)$ denotes the sign function.

The feasible isolating intervals generally can be written in the union form of disjoint maximal open intervals.

Example 3.1. If we let $t = \{i\}_{i=0}^{8}$ be an integer sequence and $s(x) = \sum_{i=0}^{5} c_i N_{i,2}(x)$, where $c_0 > 0, c_1 > 0, c_2 < 0, c_3 = 0, c_4 > 0$, and $c_5 > 0$, then the feasible isolating intervals of $s(x)$ are $(2, 4)$ and $(4, 5)$.

Proposition 3.1. *With the above notations, let* $c(k) = (c_{k-n}, c_{k-n+1}, \cdots, c_k)$. *If*

$$sign(s(t_k)s(t_{k+1})) = 1, \quad S(c(k)) \leq 1,$$

then $s(x)$ *has no root in the interval* $[t_k, t_{k+1}]$. *If*

$$sign(s(t_k)s(t_{k+1})) = -1, \quad S(c(k)) \leq 2,$$

then (t_k, t_{k+1}) *is an isolating interval of* $s(x)$. *In other cases, we have to further isolate the real roots of* $s(x)$ *in the interval* $[t_k, t_{k+1}]$.

For any maximal interval (t_i, t_j) of the feasible isolating intervals, we denote by $S(c_{ij})$ the sign variations in the sequence c_{ij}, where $c_{ij} = (c_{i-n}, c_{i-n+1}, \cdots, c_{j-1})$. Moreover, if we compute $b_{ij} = (s(t_i), s(t_{i+1}), \cdots, s(t_j))$, then we have the following result.

Proposition 3.2. *With the above notations,*

$$\sum S(c_{ij}) = S(c), \quad S(b_{ij}) \leq S(c_{ij}).$$

Moreover, if $S(b_{ij}) = S(c_{ij})$, *then all the real roots of* $s(x)$ *on the interval* (t_i, t_j) *lie in the intervals* $(t_k, t_{k+1}), i \leq k \leq j$ *if and only if* $sign(s(t_k)s(t_{k+1})) = -1$.

In particular, if $s(t_k) = 0, i \leq k \leq j$, then $x = t_k$ is a root of $s(x)$ and $[x_k, x_k]$ is an isolating interval of $s(x)$. Propositions 3.1 and 3.2 are referred to the isolating intervals pre-computed process.

Example 3.2. We continue to consider Example 3.1. Suppose $c = (3, 1, -4, 0, 1, 1)$. For the interval $(2, 4)$, we can compute $b_{24} = (2, -\frac{3}{2}, -2)$ and $c_{24} = (3, 1, -4, 0)$, then $(2, 3)$ is an isolating interval for $s(x)$ from Proposition 3.2. Similarly, we conclude that $(4, 5)$ is an isolating interval for $s(x)$.

With the above preparations, we propose an algorithm for computing all the isolating intervals of a given spline function.

Algorithm 3.1. ALGORITHM FOR ISOLATING REAL ROOTS OF SPLINE FUNCTIONS

Input:

$$s(x) = \sum_{i=0}^{m} c_i N_{i,n}(x), \quad x \in (t_0, t_m)$$

(i.e., given degree n, a knot sequence $(t_0, t_1, \cdots, t_{m+n+1})$, and a coefficient sequence $(c_0, c_1, \cdots, c_m)$).

Output: A sequence of isolating intervals of $s(x)$.

Step 1. Determine the feasible isolating intervals (3.1) of $s(x)$.

Step 2. Actualize the isolating intervals pre-computed process.

Step 3. Further isolate real roots of $s(x)$ in the required intervals using Algorithm 2.1.

If an univariate spline $s(x)$ does not satisfy the $Condition(c,t)$, then we can simply split it into pieces which satisfy the $Condition(c,t)$ as done in [5]. Applying Algorithm 3.1 to each piece, we can easily isolate all real roots of the given spline.

4. Examples

In this section, several examples are provided to illustrate the flexibility of the proposed algorithm for isolating the real roots of given spline functions.

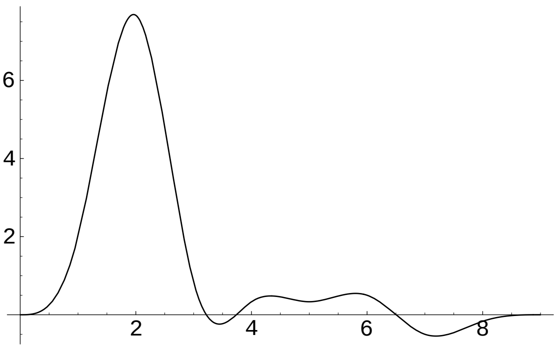

Fig. 4.1. Spline function $s(x)$ for Example 4.1.

Example 4.1. Given degree $n = 3$, a knot sequence $t = \{i\}_{i=0}^{9}$, and a coefficient sequence $c = (12, -2, 1, 0, 1, -1)$. Here, the spline function $s(x)$ is the combination of uniform B-splines as demonstrated in Fig. 4.1.

Step 1. $(1, 5)$ and $(5, 8)$ are the feasible isolating intervals.

Step 2. For the interval $(5, 8)$, we have $b_{58} = (\frac{1}{3}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{6})$ and $c_{58} = (1, 0, 1, -1, 0, 0)$. Hence, $(6, 7)$ is an isolating interval of $s(x)$ from Proposition 3.2. Moreover, since $sign(s(1)s(2)) = 1$ and $S(c(1)) = 1$, it follows from Proposition 3.1 that $s(x)$ has no root in the interval $[1, 2]$. Similarly, we conclude that $s(x)$ has no root in the interval $[4, 5]$. Therefore, the real roots of $s(x)$ in the intervals $(2, 4)$ required to be further isolated.

Step 3. It follows from Algorithm 2.1 that $(3, \frac{7}{2})$ and $(\frac{7}{2}, 4)$ are the isolating intervals of $s(x)$.

Hence, the isolating intervals of $s(x)$ are $(3, \frac{7}{2})$, $(\frac{7}{2}, 4)$, and $(6, 7)$.

Example 4.2. Given degree $n = 3$, a knot sequence $t = (0, 0, 0, 0, 1, 1, 1, 1)$, and a coefficient sequence $c = (1, -2, 2, -2)$. Here, the spline function $s(x)$ is reduced to a Bézier curve on $(0, 1)$.

The isolating interval of $s(x)$ is $(0, \frac{1}{2})$ from Algorithm 2.1.

5. Conclusion and Outlook

An algorithm for isolating real roots of a given spline function is proposed in this work. It is primarily based on the Descartes' rule of signs and de Casteljau algorithm. The above examples indicate that the algorithm is effective, especially for the cases that the spline function has many zeros in the sequence of its B-spline coefficients.

Our ultimate aim is to compute the real piecewise algebraic varieties [10], i.e., the set of common real zeros of multivariate splines. For example, if we can isolate the common real roots of several bivariate splines based on the use of the sign variations of their coefficients and other additional information, then it is significant to the computation of real piecewise algebraic curves. That is to say, we wish to determine a sequence of disjoint region (simplex, hyperrectangle) such that each of them contains exactly one real root of piecewise algebraic varieties. This remains to be our future work.

Acknowledgments. This project is supported by the National Natural Science Foundation of China (Project Nos. 60373093 and 60533060). We are very grateful to the referees for their careful reading of the manuscript and many valuable suggestions.

References

[1] C. de Boor, The multiplicity of a spline zero, *Ann. Numer. Math.*, **4** (1997), 229-238.

[2] G.E. Collins and A.G. Akritas, Polynomial real root isolation using Descarte's rule of signs, in: SYMSAC, 1976, pp. 272-275.

[3] W. Dahmen, Subdivision algorithm converge quadratically, *J. Comput. Appl. Math.*, **16** (1986), 145-158.

[4] G. Farin, Curves and Surfaces for Computer Aided Geometry Design: A Pratical Guide, Academic Press, 1997.

[5] T.N.T. Goodman, New bounds on the zeros of spline functions, *J. Approx. Theory*, **76**:1 (1994), 123-130.

[6] T.A. Grandine, Computing zeros of spline functions, *Comput. Aided Geom. Design*, **6** (1989), 129-136.

[7] F. Rouillier and P. Zimmermann, Efficient isolation of polynomial's real roots, *J. Comput. Appl. Math.*, **162** (2004), 33-50.

[8] L.L. Schumaker, Spline Functions: Basic Theory, Wiley, New York, 1981.

[9] R.H. Wang, Numerical Approximation, Higher Education Press, Beijing, 1999.

[10] R.H. Wang, Multivariate Spline Functions and Their Applications, Science Press/Kluwer Pub., Beijing/New York, 2001.