Fuzzy System Representation of the Spline Interpolation for Differentiable functions

Byung Soo Moon
Korea Atomic Energy Research Institute
P.O. Box105, Taeduk Science Town, Taejon, Korea 305-600
Tel:+82-42-868-2980, Fax:+82-42-868-8916
E-mail:bsmoon@nanum.kaeri.re.kr

Abstract

An algorithm for representing the cubic spline interpolation of differentiable functions by a fuzzy system is presented in this paper. The cubic B-spline functions which form a basis for the interpolation function are used as the fuzzy sets for input fuzzification. The ordinal number of the coefficient c_{kl} in the list of the coefficient c_{ij}'s as sorted in increasing order, is taken to be the output fuzzy set number in the (k, l)th entry of the fuzzy rule table. Spike functions are used for the output fuzzy sets, with c_{ij}'s as support boudaries after they are sorted. An algorithm to compute the support boundaries explicitly without solving the matrix equation involved is included, along with a few properties of the fuzzy rule matrix for the designed fuzzy system.

Keywords: Fuzzy System; Function Representation; Spline Interpolation; B-splines; Fuzzy Sets; Properties of Fuzzy Rule Tables

1. Introduction

It is well known that fuzzy systems can be used to approximate continuous functions on a compact set within arbitrary accuracy. Castro and Delgado[1] show that for a continuous function $f(x)$ on a compact set and an $\epsilon > 0$, there exists a fuzzy system that approximates $f(x)$ within ϵ. L. Wang[2] also shows this by proving that a set of fuzzy systems, each of which being identified as a function of fixed form with different parameter values, is dense in the set of all continuous functions. P. Wang et al[3] provides a constructive method for building a fuzzy system to approximate a function within a prescribed accuracy, while Kosko[4] proposes a fuzzy system with two levels one of which being used to approximate and tune the fuzzy rules.

Some of the above papers suggest constructive methods on how to set up fuzzy systems to represent continuous functions, while others provide only theoretical proofs. When the desired accuracy is in the range of $\epsilon \leq 0.001$, however, they are not practical due to the huge number of rules one must use. Noting that for the same accuracy, the cubic spline interpolation requires much less number of points compared to the Lagrangian type interpolations used in the above papers, we try the cubic spline interpolation to design a fuzzy system for differentiable functions.

In the following, we consider differentiable functions $f(x, y)$ defined on a region containing the the interval $[-1, 1] \times [-1, 1]$. We divide the interval [-1, 1] into 2n subintervals and let $x_j = -1 + jh$, $y_j = -1 + jh$ with $h = \frac{1}{n}$. If $B_i(t)$'s are cubic B-spline functions defined on $[t_{i-2}, t_{i+2}]$ so that $B_i(t_i) = \frac{2}{3}$, $B_i(t_{i-1}) = B_i(t_{i+1}) = \frac{1}{6}$ for i=-1,0,1,...,2n+1, then the spline interpolation function for $f(x, y)$ can be written as

$$S(x, y) = \sum_{i,j=-1}^{2n+1} c_{ij} B_i(x) B_j(y) \qquad (1)$$

with c_{ij}'s obtained from solving a set of linear equations for the interpolation constraints[5].

2. A Fuzzy System to Represent Spline Interpolation

In this section, we describe how to set up a fuzzy system for representing the cubic spline interpolation for a differentiable function $f(x, y)$.

(1) Fuzzy Sets for Input Fuzzification

Let $x_i = -1 + ih$, $y_j = -1 + jh$ with $h = \frac{1}{n}$ and define cubic B-spline functions $B_i(x)$ and $B_j(y)$ as described above. We take these functions $B_i(x)$ and $B_i(y)$ for $i = -1, 0, 1, \ldots, 2n+1$ as fuzzy sets for input variables x and y respectively. If t is an arbitrary point in $[-1, 1]$, then $t \in [x_{k-1}, x_k)$ for some $1 \leq k \leq 2n+2$. When t is fuzzified by $\{B_i(x) | i = -1, 0, 1, \ldots, 2n+1\}$, we obtain $\lambda_i = B_i(t)$ satisfying $\sum_{i=-1}^{2n+1} \lambda_i = 1$ which follows from $\sum_{i=-1}^{2n+1} B_i(x) = 1$ for all x. Note that $\lambda_i = 0$ for $i < k-2$ or for $i \geq k+2$ and hence t has nonzero membership in $B_i(x)$ only when $k-2 \leq i \leq k+1$, so that we have $\sum_{i=k-2}^{k+1} \lambda_i = 1$.

(2) Generation of Fuzzy Rules

First, we sort the $(2n+3)^2$ coefficients c_{ij} in (1) in increasing order and delete the duplicate ones, i.e., delete c_{kl} for example when $|c_{ij} - c_{kl}| \leq 10^{-7}$. Next, we assign ordinal number 1 to the smallest c_{ij} and 2 to the second smallest, and so forth. The largest c_{ij} will have ordinal number N which is less than or equal to $(2n+3)^2$. We then form a rule matrix R so that (i, j)th entry R_{ij} is the ordinal number(fuzzy set number) corresponding to the coefficient c_{ij}. For the fuzzy inferences, we use the Larsen's product rule so that if x belongs to $B_i(x)$ with membership value λ_i and y belongs to $B_j(y)$ with μ_j, then (x, y) belongs to the set R_{ij} with membership value $\lambda_i \mu_j$.

(3) Output Fuzzy Sets

Let $\{t_k \mid k = 1, 2, \ldots, N\}$ be the sorted array of c_{ij}'s. For each k, define an output fuzzy set T_k to be triangular set(spike function) whose support is $[t_{k-1}, t_{k+1}]$ with maximum value of 1 at t_k. For the first fuzzy set, we use an arbitrary interval $[t_0, t_2]$ with $t_0 < t_1$, and similarly for the last set, an interval $[t_{N-1}, t_{n+1}]$ with $t_{N+1} > t_N$ is used. The ordinal numbers in the rule table represent these output fuzzy sets, i.e. the fuzzy set number j in the rule table described above represents the fuzzy set T_j, for $j = 1, 2, \ldots, N$.

(4) Defuzzification

We use the center area defuzzification method for defuzzification of the output. When the output from the fuzzy inferences are fuzzy sets $T_{k(j)}$'s with weights $\nu_{k(j)}$'s for $j = 1, 2, \ldots, 16$, the defuzzification will compute

$$\frac{\sum_{j=1}^{16} \nu_{k(j)} t_{k(j)}}{\sum_{j=1}^{16} \nu_{k(j)}} = \sum_{j=1}^{16} \nu_{k(j)} t_{k(j)} \qquad (2)$$

where $t_{k(j)}$ is the center of support for the fuzzy set $T_{k(j)}$. Note that the denominator $\sum_{j=1}^{16} \nu_{k(j)}$ is of the form $(\sum_{i=-2}^{1} \lambda_{i_0+i})(\sum_{j=-2}^{1} \mu_{j_0+j})$ which is 1 since both factors are 1.

The following theorem proves that the fuzzy system described above is an exact representation of the cubic spline interpolation for arbitrary differentiable functions.

Theorem 1. Let $f(x, y)$ be a differentiable function of x and y and let $S(x, y) = \sum_{i,j=-1}^{2n+1} c_{ij} B_i(x) B_j(y)$ be the spline interpolation of $f(x, y)$ at $\{(x_i, y_j) \mid -1 \leq i, j \leq 2n+1\}$. If F is the fuzzy system designed as above using the coefficients c_{ij}'s, then the output $F(x, y)$ of the fuzzy system at $(x, y) \in [-1, 1] \times [-1, 1]$ is identical to the value of the spline interpolation function (1), i.e. $F(x, y) = S(x, y)$.

Proof. Let (x,y) be an arbitrary point in $[-1,1]\times[-1,1]$, then $x \in [x_{k-1}, x_k)$ and $y \in [y_{l-1}, y_l)$ for some $1 \le k,l \le 2n+2$. If $\lambda_i = B_i(x)$ for $-1 \le i,j \le 2n+1$, then from $\sum_{i=-1}^{2n+1} B_i(t) = 1$ for all $t \in [-1,1]$, we have $\sum_{i=-1}^{2n+1} \lambda_i = 1$. Recall that $\lambda_i = 0$ when $i < k-2$ or $i \ge k+2$, and hence we have $\sum_{i=k-2}^{k+1} \lambda_i = 1$. Therefore, when x is fuzzified, we obtain fuzzy sets numbered $k-2$, $k-1$, k, $k+1$ each with membership values λ_{k-2}, λ_{k-1}, λ_k, λ_{k+1}, whose sum is 1.

Similarly, when y is fuzzified, we have fuzzy sets $l-2$, $l-1$, l, $l+1$, whose membership values being μ_{l-2}, μ_{l-1}, μ_l, μ_{l+1}. When the fuzzy rules are applied, we obtain 16 output fuzzy sets (i,j), for $i = k-2$, $k-1$, k, $k+1$ and $j = l-2$, $l-1$, l, $l+1$ with membership values $\lambda_i\mu_j$.

Finally, the center area defuzzification produces $\frac{\sum_{i,j=-1}^{2n+1} \lambda_i\mu_j c_{ij}}{\sum_{i,j=-1}^{2n+1} \lambda_i\mu_j}$, which becomes $\sum_{i,j=-1}^{2n+1} \lambda_i\mu_j c_{ij}$ since the sum in the denominator is 1. Now, note that this value is the same as $S(x_0,y_0) = \sum_{i,j=-1}^{2n+1} c_{ij}B_i(x_0)B_j(y_0)$ and hence the theorem is proved. Q.E.D.

3. Computation of Support Boundaries

In this section, we describe an explicit method to compute the support boundaries of the output fuzzy sets. As mentioned in the previous section, the output fuzzy sets are defined to be the spike functions with support boundaries consisting of cubic spline interpolation coefficients. We have shown in our earlier work[6] that the interpolation coefficients for $P(x) = x^m$ can be computed directly without solving the matrix equation involved. Let $\lambda_0 = 1$, $\lambda_1 = -1$, and define λ_j's successively by

$$\lambda_{2k} = 1 - \frac{1}{3}\sum_{i=0}^{k-1} {}_{2k}C_{2i}\lambda_{2i}, \qquad (3)$$

$$\lambda_{2k+1} = -1 - \frac{1}{3}\sum_{i=0}^{k-1} {}_{2k+1}C_{2i+1}\lambda_{2i+1}. \qquad (4)$$

with $k = 1,2,\ldots$. Using these λ_j's and the degree of the polynomial m, we define

$$q_{m,j} = \sum_{l=0}^{m} {}_mC_l j^{m-l}\lambda_l \qquad (5)$$

for $j = -n, -n+1, \ldots, n+2$, then it is shown in our earlier work[6] that the spline interpolation function can be written as

$$S(x) = \sum_{i=-1}^{2n+1}\left(\frac{q_{m,-n+i+1}}{6n^m} + \epsilon_i\right)B_i(x). \qquad (6)$$

where $\epsilon_i = 0$ for all i when $m \le 4$.

To compute ϵ_i to be used when $m > 4$, we first let $\alpha_1 = 0.5$, and define α_k successively by $\alpha_{k+1} = 1/(4-\alpha_k)$, for $k = 1,2,\ldots,n$. Let $p_{n+2} = 1/(1-\alpha_n\alpha_{n+1})$, and define p_k successively by $p_k = -\alpha_k p_{k+1} = (-1)^{n-k}\alpha_k\alpha_{k+1}\cdots\alpha_{n+1}\alpha_{n+2}$, for $k = n+1, n, \ldots, 1$. We also define $r_{n+1} = 1/(1-\alpha_{n-1}\alpha_n)$, $r_k = -\alpha_k r_{k+1}, k = n, n-1, \ldots, 2$, and $r_1 = -\frac{1}{4}r_2$. Then the following theorem provides a method to compute the cubic spline interpolation function without solving the matrix equation involved.

Theorem 2[6]. Let m be an even positive integer and let $P(x) = x^m$. If $q_{m,j}$'s and p_j's are as defined above, $Q_{m,j} = \frac{q_{m,j}}{6n^m}$, and $\rho = \frac{m}{3n} - \frac{q_{m,n+2}-q_{m,n}}{6n^m}$, then the $(2n+3)$ spline interpolation coefficients for $P(x)$ are $(Q_{m,-n} + \rho p_{n+2}, Q_{m,-n+1} + \rho p_{n+1}, \cdots, Q_{m,0} + \rho p_2, Q_{m,1} + \rho p_1, Q_{m,2} + \rho p_2, \cdots, Q_{m,n+2} + \rho p_{n+2})$. In case m is an odd integer, the coefficients are $(Q_{m,-n} - \rho r_{n+1}, Q_{m,-n+1} - \rho r_n, \cdots, Q_{m,0} - \rho r_1, Q_{m,1}, Q_{m,2} + \rho r_1, \cdots, Q_{m,n+2} + \rho r_{n+1})$.

Next, we consider $P(x,y) = P_1(x) \times P_2(y)$ where $P_1(x) = x^k$, $P_2(y) = y^l$. The spline interpolation function for $P(x,y)$ becomes

$$S(x,y) = \sum_{i=-1}^{2n+1} c_iB_i(x) \times \sum_{j=-1}^{2n+1} d_jB_j(y)$$

$$= \sum_{i=-1}^{2n+1} \sum_{j=-1}^{2n+1} c_i d_j B_i(x) B_j(y) \qquad (7)$$

where c_i's and d_j's are computed by the algorithm described in Theorem 1. When the above method is used to compute the interpolation coefficients for some of the simple polynomials, we obtain the following results.

Example 1. *The spline interpolation coefficients for $P(x) = x^m$ with $m = 1, 2, 3$ and 4.*

When $q_{m,j}$'s are computed using Theorem 1, we obtain $q_{1,j} = j - 1$, $q_{2,j} = (j-1)^2 - \frac{1}{3}$, $q_{3,j} = j(j-1)(j-2)$, $q_{4,j} = j^2(j-1)^2 - \frac{2}{3}$, $q_{5,j} = j^5 - 5j^4 + \frac{20}{3}j^3 - \frac{10}{3}j + \frac{2}{3}$, $q_{6,j} = j^6 - 6j^5 + 10j^4 - 10j^2 + 4j + \frac{2}{3}$. Hence the spline interpolation coefficients are $c_j^{(1)} = \frac{j-1}{n}$, $c_j^{(2)} = (\frac{j-1}{n})^2 - \frac{1}{3n^2}$, $c_j^{(3)} = \frac{j}{n}\frac{j-1}{n}\frac{j-2}{n}$ and $c_j^{(4)} = (\frac{j}{n})^2(\frac{j-1}{n})^2 - \frac{2}{3n^4}$.

Example 2. *The spline interpolation coefficients for $P(x,y) = x \times y$ and $Q(x,y) = x^2 + y^2$.*

When the results of Example 1 are substituted into relation (7), we obtain $\frac{q_{1,-n+i+1}}{n}\frac{q_{1,-n+j+1}}{n} = (-1 + \frac{i}{n})(-1 + \frac{j}{n})$ for $P(x,y)$. Similarly, the coefficients for $Q(x,y) = x^2 + y^2$ are $(-1 + \frac{i}{n})^2 + (-1 + \frac{j}{n})^2 - \frac{2}{3n^2}$.

4. Some Properties of the Fuzzy Rule Table

In this section, we describe some properties of the fuzzy rule table or equivalently of the rule matrix.

Theorem 3. Let $f(x,y)$ and $g(x,y)$ be differentiable functions and let $\sum_{i,j=-1}^{2n+1} c_{ij} B_i(x) B_j(y)$, $\sum_{ij=-1}^{2n+1} d_{ij} B_i(x) B_j(y)$ be the corresponding spline interpolation functions. If $\{\ (i(n), j(n)) \mid n = 1, 2, \ldots, N\}$ are the indices of c_{ij}'s when they are sorted in an increasing order with duplicate ones deleted and if $\{\ (k(n), l(n)) \mid n = 1, 2, \ldots, N\}$ are the corresponding indices for d_{kl}'s, then the rule table for $P(x,y)$ is identical to that of $Q(x,y)$ if and only if $k(n) = i(n)$, $l(n) = j(n)$, for all $n = 1, 2, \ldots, N$.

Proof. Recall that the (i,j)th entry of the rule matrix R_{ij} is defined to be the ordinal number of c_{ij} or d_{ij} respectively for $f(x,y)$ and $g(x,y)$. Hence, all the corresponding ordinal numbers are the same if and only if the two rule matrices are the same. Q.E.D.

Corollary 1. If $f(x,y) = cg(x,y) + d$ with $c > 0$, then the rule table for $f(x,y)$ is the same as the table for $g(x,y)$. The only difference in the fuzzy system representations of $f(x,y)$ and $g(x,y)$ in this case is in the output fuzzy sets, whose support boundaries are scaled by c and shifted by d.

Theorem 4. If two differentiable functions $f(x,y)$ and $g(x,y)$ have the same fuzzy rule table, then the rule table for the sum $f(x,y) + g(x,y)$ will be the same as that for $f(x,y)$.

Proof. Let c_{ij}, d_{ij}, $-1 \le i, j \le 2n+1$ be the interpolation coefficients for $f(x,y)$ and $g(x,y)$ respectively. Then we have $c_{ij} < c_{kl}$ if and only if $d_{ij} < d_{kl}$ since the rule tables are the same. Therefore, $c_{ij} + d_{ij} < c_{kl} + d_{kl}$ if and only if $c_{ij} < c_{kl}$, i.e. the rule table for $f(x,y) + g(x,y)$ is the same as that for $f(x,y)$. Q.E.D.

Theorem 5. Let $P_0(x,y) = x + y$ and $P(x,y) = x + my$ with a positive integer m, then the (i,j)th entry of the fuzzy rule table R_{ij} is equal to $(i-1) + m_0(j-1) + 1$, where m_0 is the minimum of m and $2n+3$. The center of support for the output fuzzy set corresponding to R_{ij} is at $-1.2(m+1) + (i - 1 + m(j-1))h$.

A routine proof of the above is omitted.

Corollary 2. The fuzzy rule table for polynomials $P_1(x,y) = x + (2n+3)y$ and $P_2(x,y) = x + ky$ are the same, when $k > 2n+3$.

Example 3. When $n = 5$, the two polynomials $P(x,y) = x + 13y$ and $Q(x,y) = x + 14y$ have the same fuzzy rule table and so does $R(x,y) = 2x + 27y$ since $R(x,y) = P(x,y) + Q(x,y)$.

Theorem 6. Let $P(x,y) = x + \alpha y$ and let 2n be the number of subintervals of the interval $[-1,1]$. Then the (i,j)th entry of the fuzzy rule table for $P(x,y)$ is given by $R_{ij} = (i-1)(2n+3) + j$, $i,j = -1,0,\ldots,2n+1$ for all α satisfying $0 < \alpha < \frac{1}{2n+3}$.

Proof. Let c_{ij}, $i,j = -1,0,1,\ldots,2n+1$ be the spline coefficients for $P(x,y) = x + \alpha y$. Note that it is equivalent to prove that the fuzzy set numbers in the rule table increase as the column number increases in a row and as the row number increases from 1 to $2n+3$, i.e. $c_{i,j} < c_{i,j+1}$ and $c_{i,2n+1} < c_{i+1,-1}$ for all i and j. Recall that $q_{1,-n+i+1} = -n + i$ and hence $c_{i,j} = (-1 + \frac{i}{n}) + \alpha(-1 + \frac{j}{n}) = -(1+\alpha) + \frac{1+\alpha j}{n}$. From this, it is trivial to verify $c_{i,j} < c_{i,j+1}$ since $\alpha > 0$. The second inequality follows from $c_{i+1,-1} - c_{i,2n+1} = \frac{1-(2n+2)\alpha}{n}$ which is positive since $\alpha < \frac{1}{2n+3}$ by assumption. Q.E.D.

5. Examples

In this section, we describe two examples of the fuzzy system designed to represent the cubic spline interpolation of a polynomial in x and y. Evaluation results of the fuzzy system are compared with those of a fuzzy system using spike functions for the input fuzzification. We evaluated the fuzzy systems at 10,000 points and computed the average and the maximum of the absolute errors.

Example 4. *Fuzzy system for* $P(x,y) = x^4y^3$

When the procedure described in the previous section is applied to $P(x,y) = x^4 \times y^3$ with n=2, we obtain 7 x 7 fuzzy rules in Table 1, with the centers of support for the 17 output fuzzy sets at -11.8750, -2.96875, -1.56250, -.390625, -.125000, -.062500, -.031250, -.015625, 0., .015625, .031250, .062500, .125000, .390625, 1.56250, 2.96875, 11.8750.

Table 2 shows a summary of the evaluation errors at 10,000 points, including those by a fuzzy system using spike functions for n=2, n=5, n=10. The sizes of the rule tables are 7 × 7, 13 × 13, and 23 × 23 for the spline interpolations, and 5 × 5, 11 × 11, 21 × 21 respectively for the case when spike funcions are used.

As can be seen from Table 2, both the maximum absolute errors and the averages of the absolute errors for the spline cases reduce by $O(\frac{1}{n^4})$, while the reduction rate is only of $O(\frac{1}{n^2})$ for the spike function cases.

Table 1. Fuzzy Rules for $P(x,y) = x^4y^3$

		y						
		1	2	3	4	5	6	7
	1	1	2	9	9	9	16	17
	2	3	4	9	9	9	14	15
	3	13	11	9	9	9	7	5
x	4	6	8	9	9	9	10	12
	5	13	11	9	9	9	7	5
	6	3	4	9	9	9	14	15
	7	1	2	9	9	9	16	17

Table 2. Comparison of Evaluation Errors (Spline vs Spike)

	Maximum Error		Average Error	
n	Spline	Spike	Spline	Spike
2	0.003790	0.219168	0.000521	0.037837
5	0.000095	0.056989	0.000013	0.005399
10	0.000006	0.017504	0.000001	0.001287

Example 5. *Fuzzy system for* $P(x,y) = x^2y + xy^2$

When the procedure described in the previous section is applied to $P(x,y) = x^2y + xy^2$ with n=2, we obtain 7×7 fuzzy rules in Table 3, with the centers of support for the 25 output fuzzy sets at -6.50000, -3.54166, -1.83333, -1.33333, -.833333, -.791666, -.625000, -.291666, -.166666, -.125000, -.083333, -.041666, -.000000, .041666, .083333, .125000, .166666, .291666, .625000, .791666, .833333, 1.33333, 1.83333, 3.54166, 6.50000

Table 4 shows a summary of the evaluation errors at 10,000 points, including those by a fuzzy system using spike functions for n=2, n=5, n=10. Note that both the maximum absolute errors and the averages of the absolute errors for the spline cases are less than 10^{-7}. This is due to the fact that the spline interpolation function is essentially a polynomial of degree 3 and so is the function $P(x,y)$.

Table 3. Fuzzy Rules for $P(x,y) = x^2y + xy^2$

		y 1	2	3	4	5	6	7
	1	1	2	4	16	21	20	13
	2	2	3	7	15	18	13	6
	3	4	7	9	14	13	8	5
x	4	16	15	14	13	12	11	10
	5	21	18	13	12	17	19	22
	6	20	13	8	11	19	23	24
	7	13	6	5	10	22	24	25

Table 4. Comparison of Evaluation Errors (Spline vs Spike)

	Maximum Error		Average Error	
n	Spline	Spike	Spline	Spike
2	0.1×10^{-7}	0.096096	0.1×10^{-8}	0.029799
5	0.1×10^{-7}	0.018000	0.8×10^{-9}	0.004779
10	0.1×10^{-7}	0.004608	0.7×10^{-9}	0.001187

6. Conclusion

The fuzzy system we designed can represent differentiable functions very accurately. As seen in Example 4, at most $(2n+3)^2$ rules are needed when $2n$ subintervals are used for $[-1,1]$ with interpolation error of order $O(\frac{1}{n^4})$. This accuracy comes from the spline interpolation since our system is a representation of the cubic spline interpolation. When spike functions are used, the maximum number of rules is $(2n+1)^2$ for the interpolation error of $O(\frac{1}{n^2})$. Even though our system is only a representation of the spline interpolation function, the fuzzy system representation is better in the sense that it provides more insights on how the coefficients are related to the properties of the functions.

Reference

[1] J. L. Castro and M. Delgado, "Fuzzy Systems with Defuzzification are Universal Approximators", IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics, 26(1) (1996) 149-152.

[2] L. X. Wang, "Fuzzy Systems are Universal Approximators", Proc. IEEE Int. Conf. on Fuzzy Systems, San Diago (1992) 1163-1170.

[3] P. Wang, et al, "Constructive Theory for Fuzzy systems", Fuzzy Sets and Systems, 88 (1997) 195-203.

[4] B. Kosko, "Fuzzy Systems as Universal Approximators", IEEE Trans. on Computers, 43(11) (1994) 1329-1333.

[5] P. M. Prenter, Splines and Variational Methods, John Wiley and Sons, New York (1975) 131-133.

[6] B. S. Moon, "An Explicit Solution of the Cubic Spline Interpolation for Polynomials", J. Korea Soc. for Industrial and Applied Math., 1(1) (1997) 75-81.