Improvements to the Algorithm that Uses Divided Differences to Determine the Coefficients in B-Spline Interpolation

Liliana Mâțiu-Iovan

"Politehnica" University of Timisoara / Faculty of Electronics and Telecommunications,
Bd. V. Pârvan No. 2, 300223, Timişoara, Romania,
Tel :+(40)-256-403369, e-mail: liliana.matiu@etc.upt.ro

Abstract-In numerical measuring processes the interpolation is an important part. Some algorithms of B-spline interpolation were developed along the time. Here are presented two additional methods to reduce the interpolation errors. These methods could be used in any case for uniformly spaced data. These interpolation processes requires supplementary numerical calculations. Always is to choose between a smallest amount of computation and better results of interpolation.

I. Introduction

From ancient times it was studied the interpolation problem: haw to find supplementary information from a given data string. The problem of signal reconstruction and interpolation is still an actual matter. The spline functions were use from a long time, but only in the 60's they were developed and applied on great scale. These functions were utilized for interpolation in the traditional approach. Lately they appear in algorithms that use numerical processing techniques.
In the context of generalized B-spline interpolation is necessary to perform two steps: determine the coefficients (pre-filtering) and after, calculate the interpolated values. Starting from algorithms developed in digital image processing were searched new ways to implement the pre-filtering step. Some algorithms were described in previous works [3]. The B-spline coefficients are calculated using the values of the function and function's derivatives in the knots. There are presented the filters that are applied on the input signal to obtain the coefficients. The second step of the interpolation is performed in the same way in all the cases. The results for some well-known functions were analyzed. A problem can appear if the input data represent a signal sampled at a low sampling frequency and the application requires a smaller value for the interpolation error.

II. Known algorithms

A. Generalized interpolation

The process of general interpolation supposes to perform two steps. First is to determine some coefficients $c(k)$ starting from the input data $y(k)$. The second step uses the coefficients and a basis function to calculate the interpolated values [4]:

$$f(x) = \sum_{k \in Z} c(k)(x-k) \qquad (1)$$

We can tell that the traditional interpolation is a particular case of the generalized interpolation. In this case the coefficients are equal to the input samples: $c(k) = y(k)$.
The spline functions are often used in interpolation like basis functions. An algorithm founded in the literature for image processing use B-spline functions and digital filtering techniques [5]. The main advantages of that spline are the polynomial form, the fact that they are short and the continuity properties. A function of degree n and his derived up to n-1 order are continuous. They can be easy to use and allow realizing simple algorithms ready to implement on numerical systems.

B. Unser's B-spline interpolation algorithm

In many algorithms of interpolation (old, well known techniques or new approaches) are often used the cubic spline functions (n=3). In the traditional manner, the problem is resolved by matrix algebra methods. There are necessary a great amount of operations. A relative recent approach is to use digital filtering techniques. The idea appeared in the 70's and is developed later by Michael Unser and his team [5], [6] and [7]. They elaborate methods that use digital filters for interpolation and image processing. The cubic B-spline function is used. This function denoted by $\beta^3(x)$ is described in (2).

$$\beta^3(x) = \begin{cases} 2/3 - |x|^2 + |x|^3/2, & 0 \le |x| < 1 \\ (2 - |x|)^3/6, & 1 \le |x| < 2 \\ 0, & 2 \le |x| \end{cases} \quad (2)$$

The discrete B-spline function $b_1^3(k)$, the direct B-spline filter (3) and the indirect B-spline filter were defined in [5], [6] and [7].

$$\left[B_1^3(z)\right]^{-1} = \frac{6}{z + 4 + z^{-1}} \quad (3)$$

The algorithm of B-spline interpolation needs to apply the direct B-spline filter to the input signal. The operation is called "direct B-spline transform". We obtain the spline coefficients $c(k)$. The interpolated function $f^n(x/m)$ by a factor m, denoted $f_m^n(x)$, will be obtained by "indirect B-spline transform":

$$f_m^n(x) = \sum_{k \in Z} c(k) b_m^n(x - km) \quad (4)$$

This operation is implemented also by digital filtering [5], [7].
The direct B-spline filter is implemented by 2 filters to calculate the coefficients. The two filters are one a causal filter, and the second, an anti-causal. The recursive algorithm demands some initial conditions. Is performed the signal extension by mirroring and they are taken a finite number of samples. The initial conditions introduce some side errors for the coefficients [2]. Those errors are transmitted in the interpolated signal and could have great importance, especially, if the input signal contains a small number of samples.

III. An algorithm based on numerical differentiation

To perform the spline interpolation in the traditional manner are used the known input samples and some values of the derived function. From this idea, to determine the new initial coefficients there were evaluated also the derivatives for the input function.
Consider $f(x)$ an approximation for the cubic spline function that pass trough all the input values: $f(k)=y(k)$, $k = 0, \ldots, N-1$. In the knots $f(k)$ represents the convolution between the coefficients string and the cubic B-spline function (2). The relation involving the function and the coefficients $c(k)$ can be write:

$$6f(k) = 4\,c(k) + c(k-1) + c(k+1) \quad (5)$$

The cubic B-spline function derivatives of first and second order are analyzed. From these ones are determined the relations between the $f(k)$ derivatives and the coefficients:

$$f'(k) = 0\,c\,(k) - ½\,c\,(k-1) + ½\,c\,(k+1) \quad (6)$$

Compared with the Unser's algorithm, in this new approach is not necessary to perform the signal extension. But it has to establish a way to determine the values for the function derivatives of order one and two. These values must be obtained by numerical methods only from the input samples. The interpolation function is a B-spline (piecewise polynomial), so we can approximate $f(k)$ by a polynomial function on short intervals. From the relation (6) it can be established a general formulation in every knot:

$$c(k+1) - c(k-1) = 2\,f'(k) \quad (7)$$

The algorithm supposes to use the function derivatives and to impose their values. This type of interpolation is called Hermite interpolation.
Dealing with discrete dates, now the problem it is to perform the numerical differentiation. The divided differences can be defined:

$$f'(k) \cong \frac{f(k+h) - f(k-h)}{2h} \quad (8)$$

The same relation can be found by calculating the central derivative for a polynomial function that goes through 3 points.
For any k value, based on (8), the iterative relation for calculating the coefficients became:

$$c(k+1) - c(k-1) = y(k+1) - y(k-1) \quad (9)$$

As it can be observed any differences between 2 coefficients $c(k+h)$ and $c(k)$ depends of the samples values in $k+h$ and k points only. The algorithm is convergent [3].
Any coefficient $c(k)$ can be determined like:

$$c(k) = c(k-2) + y(k) - y(k-2) \quad (10)$$

In this case the initial values are provided in [3]. The coefficients are calculated with this method. The interpolated values will be obtained using (4), where m represents the interpolation factor.

IV. Two methods to improve the algorithm

A. Introduction

The algorithm described in section III present a major disadvantage. Every value of the coefficients is influenced by errors from the others coefficients. Here are searched several ways to improve the results.

From the inverse B-spline filter is determined a relation between the input samples $y(k)$ and the B-spline coefficients. This is written as:

$$6y(k) = 4\,c(k) + c(k\text{-}1) + c(k\text{+}1) \Rightarrow \; c(k) = \frac{6y(k) - c(k-1) - c(k+1)}{4} \qquad (11)$$

where $c(k)$ represents the current coefficient. The relation (11) will be used further, trying to reduce the interpolation errors.

B. The first improved method

One solution is to use different equations to determine the even and the odd coefficients. The even coefficients are calculated with the known method, described in (10). The odd coefficients are obtained with the additional method. The coefficients calculated with this method will be denoted by $c_1(k)$

$$c_1(k) = c_1\,(k\text{-}2) + \; y(k) - y(k\text{-}2) \qquad (12)$$

$$c_1(k-1) = \frac{6y(k-1) - c_1(k-2) - c_1(k)}{4}, \text{ for } k=2, 4, 6, ... \; (13)$$

The interpolation by a factor m is performed like in the other case, this time using the new coefficients.

The new method was utilized to perform the interpolation by m=2 for some well know signals. The input data strings represent samples from the signals $y(k)=\sin(2\pi k/M)$ and $y(k)=\cos(2\pi k/M)$, with k=0,…, N-1 for different values of M and N. Here are analyzed two situations for different sampling frequencies (the values for M are 12 and 120). The input strings have N =37, respectively N =361 samples. First, it was applied the algorithm in section III and there were obtained the interpolated values. Next, it was used the new method. The results were compared. It was considered irrelevant any value smaller than 10^{-8}.

By evaluating the interpolation errors we can make some observation. For the algorithm in section III, the interpolation errors are of 10^{-2} order in case of M =12, and 10^{-4} order for M =120. The interpolation errors are zero for all the points were the coefficients were determined using (13). In the other points, the errors are comparable with the ones for the first algorithm. This is happening for all the studied cases. As it can be seen, the results are not better for all the data.

C. The second improved method

The second solution is to determine all the coefficients $c(k)$ with the algorithm described in section III and to perform a supplementary step. This is to apply for all of them the next formula:

$$c_2(k) = \frac{6y(k) - c(k-1) - c(k+1)}{4} \qquad (14)$$

The new of coefficients $c_2(k)$ are utilized in (4) to determine the interpolated values. In this case, the entire process has three phases:

- to determine the coefficients $c(k)$ from the input samples $y(k)$;
- to calculate the new coefficients $c_2(k)$ from $c(k)$;
- to establish the interpolation factor m and compute the values in (4).

This algorithm was utilized for the same input data $y(k)=\sin(2\pi k/M)$ and $y(k)=\cos(2\pi k/M)$. There were analyzed the same cases: M =12 and M =120. The interpolation errors for some points α on the function characteristic are presented in Table I. The input samples correspond to $y(k)=\cos(2\pi k/M)$. In the third column are presented the interpolation errors obtained in case of the algorithm described in section III. The values in the last column correspond to the new algorithm.

In case of the input signal being $y(k)=\sin(2\pi k/M)$, some values for the interpolation errors are presented in Table II. There are compared the results for the algorithms described in section III and section IV.C.

The values obtained from the last algorithm are approximately two times smaller than the others. The interpolation errors are reduced two times by introducing the supplementary step in the algorithm.

Table I. Interpolation errors for $y(k)=\cos(2\pi k/M)$.

α	M	Section III	Section IV.C
0	12	0.01415608	0
	120	0.00000156	0
π/3	12	0.00817301	0.00366878
	120	0.00022684	0.00011404
π/2	12	0.03867513	0.01525105
	120	0.00045525	0.00022809

Table II. Interpolation errors for $y(k)=\sin(2\pi k/M)$.

α	M	Section III	Section IV.C
0	12	0.03050211	0.01116454
	120	0.00003582	0.00001195
π/3	12	0.00817301	0.00558227
	120	0.00035978	0.00018558
π/2	12	0.02232909	0.00408650
	120	0.00042099	0.00021614

Similar results were acquired in other cases where the input data string represents samples obtained from continuous signals. The situation of noisy signals as input might present interest. It will be studied also the case in witch the samples correspond to a signal with some discontinuities. The results will be presented in a further work.

V. Conclusions

The presented algorithms and methods use known techniques combined in a new manner. The values for the function derivatives are utilized also in the traditional approach to resolve the interpolation problem. Here were used the divided differences because we work with discrete functions. The way to determine these divided differences may be improved.

The additional methods presented in this paper offer some changes. The results are not dramatically improved. The extra amount of computation necessary for the second method is not justified in many practical examples. The relations developed here can be utilized also with others algorithms. A better way to calculate the divided differences combined with these additional methods could offer smaller errors.

Based on several studies and test, we gone choose one of the interpolation algorithms to be implemented on a digital signal processor and integrated in to a measurement system.

References

[1] Gh. Micula, *Functii Spline si Aplicatii*, Editura Tehnica, Bucuresti, 1978.

[2] L. Stoica, "Contributions in Recursive Filtering for B-splines Interpolation in Signal Processing", *Scientific Bulletin of the "Politehnica" University of Timisoara, Romania, Transactions on Electronics and Communications*, Vol. 51(65), No 2, "Etc2006", pp. 44-49, Sept. 2006.

[3] L. Stoica, "A New Algorithm for Determining the Coefficients in B-spline Interpolation", *Scientific Bulletin of the "Politehnica" University of Timisoara*, *Romania, Transactions on Electronics and Communications*, Vol. 51(65), No 2, "Etc2006", pp. 5-8, Sept. 2006.

[4] P. Thevenaz, T. Blu, M. Unser, "Interpolation Revisited", *IEEE Transactions on Medical Imaging*, Vol. 19, No. 7, pp. 739-758, July 2000.

[5] M. Unser, "Splines: a Perfect Fit for Signal and Image Processing", *IEEE Signal Processing Magazine*, Vol. 16, No. 6, pp. 22-38, Nov. 1999.

[6] M. Unser, A. Aldroubi, M. Eden, "B-spline Signal Processing: Part I - Theory", *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, pp. 821-833, Feb. 1993.

[7] M. Unser, A. Aldroubi, M. Eden, "B-spline Signal Processing: Part II – Efficient Design and Applications", *IEEE Transactions on Signal Processing*, Vol. 41, No. 2, pp. 834-848, Feb. 1993.