American Journal of Applied Mathematics
2013; 1(2): 24-27
Published online June 30, 2013 (http://www.sciencepublishinggroup.com/j/ajam)
doi: 10.11648/j.ajam.20130102.11

Piecewise cubic approximation for data

Emine Can[1], Sabri Ali Ümekkan[1], Canan Köroğlu[2]

[1]Kocaeli University, Dept. of Art and Science, Kocaeli, Turkey
[2]Hacettepe University, Dept. of Mathematics, Ankara, Turkey

Email address:
eminecan@kocaeli.edu.tr(Emine Can), sa.umekkan@hotmail.com(S. Ali Ümekkan), ckoroglu@hacettepe.edu.tr(Canan Köroğlu)

To cite this article:
Emine Can, Sabri Ali Ümekkan, Canan Köroğlu. Piecewise Cubic Approximation for Data *American Journal of Applied Mathematics*. Vol. 1, No. 2, 2013, pp. 24-27. doi: 10.11648/j.ajam.20130102.11

Abstract: In this paper, we consider features concerning approximation for data by using piecewise interpolation techniques. Numerical examples are given which compare piecewise cubic interpolation methods.

Keywords: Piecewise, Cubic Bessel Interpolation, Cubic Hermite Interpolation, Cubic Interpolation

1. Introduction

Interpolation on a discrete set of data points is a problem, encountered in many fields of science and engineering [1-3]. For example, data analysis is commonly used for image processing, learning, decision making, application in numerical hydrodynamics, and astrophysical applications. We are interested in numerically fitting a curve through a given finite set of points. If smoothness of the interpolation curve is major importance, cubic splines are often method to choose [10]. Cubic splines interpolate (pass through) the data with piecewise cubic polynomials [5-6]. The use of low-order polynomials is especially attractive for curve fitting, because they reduce the computational requirements, and numerical instabilities that arise with higher degree curves [9]. In this paper, we propose different piecewise interpolation techniques for data [7].

Numerical examples are presented to show graphically the differences using the suggested piecewise cubic approximations.

2. Piecewise Interpolation Techniques

2.1. Construction of Polynomial

Piecewise interpolation divides the function to pieces, and constructs polynomials for each piece. Hence, the piecewise interpolation requires points from nearest intervals of the data we have.

Suppose that we have given monotone data as follows,

$$(x_i, f_i), \text{ and } i = 1,..,n+1$$

$$a = x_0 < x_1 <..< x_{n+1} = b$$

$$f[x_i, x_{i+1}] = (f_{i+1} - f_i)/x_{i+1} - x_i,\ i = 0,..,N$$

Data are called monotonically increasing if

$$f[x_i, x_{i+1}] \geq 0 \qquad [3]$$

$$f(x) = P_i(x), \qquad x_i \leq x \leq x_{i+1}, \qquad i = 1,..,n-1$$

And, each interval f agrees with some polynomial P_i order 4.

$$P_i(x_i) = g(x_i), P_i(x_{i+1}) = g(x_{i+1}), \qquad i = 1,..,n-1$$

$$P_i{}'(x_i) = S_i, \qquad P_i{}'(x_{i+1}) = S_{i+1}$$

In order to compute the coefficients of the ith polynomial piece, P_i we use its Newton form.

$$P_i(x) = P_i(x_i) + (x - x_i)P_i(x_i, x_i) + (x - x_i)^2 P_i(x_i, x_i, x_{i+1}) + (x - x_i)^2(x - x_{i+1})P_i(x_i, x_i, x_{i+1}, x_{i+1})$$

2.1.1. Algorithm

Step 1: Set the data points (x_i, f_i).
Step 2: Construct the cubic polynomial P_i, which use the nearest intervals.
Step 3: Determine the coefficients according to chosen method to use.
Step 4: Combine the ploynomials P_i for each piece.
Step 5: Plot the function.

2.1.2. Piecewise Cubic Interpolation

Here, we focus only on this algorithm which use C^1 piecewise cubic S. It is well known that C^1 piecewise cubic can approximate a four times continuously differentiable function f to an order $O(h^4)$ with maximum spacing of the knots. The C1 cubic interpolant, which preserves monotonicity has the following form. It begins with some assignments of slopes S_i, at the points $x_i, n = 0, \ldots, n$ which is third-order accurate when underlying function f is smooth [8]. After this assignments,polynomial can be written as follows [4],

$$P_i(x) = c_{1,i} + c_{2,i}(x - x_i) + c_{3,i}(x - x_i)^2 + c_{4,i}(x - x_i)^3$$

coefficients can be gathered from divided difference table.

$$c_{1,i} = P_i(x_i) = g(x_i)$$

$$c_{2,i} = P_i'(x_i) = s_i$$

$$c_{3,i} = P_i''(x_i)/2 = [g(x_i, x_{i+1}) - s_i]/(x_{i+1} - x_i) - c_{4,i}(x_{i+1} - x_i)$$

$$c_{4,i} = P_i'''(x_i)/6 = [s_i + s_{i+1} - 2g(x_i, x_{i+1})]/(x_{i+1} - x_i)^2$$

2.1.3. Piecewise Cubic Hermite Interpolation

Define a cubic hermite spline is a third degree spline with each polynomial of the spline in hermite form, and its values and first derivatives at the end points of the corresponding domain interval. Piecewise cubic hermite interpolation form is defined as [4],

$$P_i(x) = a_{1,i} + a_{2,i}(x - x_i) + a_{3,i}(x - x_i)^2 + a_{4,i}(x - c_i)^3$$

coefficients can be found as follows,

$$a_{1,i} = P_i(x_i) = g(x_i)$$

$$a_{2,i} = P_i'(x_i) = s_i$$

$$a_{3,i} = P_i''(x_i) = [g(x_i, x_{i+1}) - s_i]/(x_{i+1} - x_i)$$

$$a_{4,i} = P_i'''(x_i) = [s_{i+1} + s_i - 2g(x_i, x_{i+1})]/(x_{i+1} - x_i)^2$$

2.1.4. Piecewise Cubic Bessel Interpolation

The Same computation will apply with Piecewise Cubic Bessel Interpolation. But, the only difference is this method requires two additional points to construct polynomial,

$$P_i'(x_i) = S_i = \frac{(x_i - x_{i-1})g(x_i, x_{i+1}) + (x_{i+1} - x_i)g(x_{i-1}, x_i)}{(x_i - x_{i-1}) + (x_{i+1} - x_i)}$$

which gives Bessel interpolation [4].

3. Numerical Examples

In this section, we consider the figures generated by interpolation techniques for two of typical data sets in the literature [7].

3.1. Example 1

Here, $\boldsymbol{n = 9}$ and a radiochemical dataset of $\boldsymbol{x_i}$'s and $\boldsymbol{f_i}$'s are given as below,

Table 1. *Radiochemical dataset*

x_i	f_i
7.99	0
8.09	2.76E-05
8.19	0.04375
8.7	0.169183
9.2	0.469428
10	0.94374
12	0.998636
15	0.999916
20	0.999994

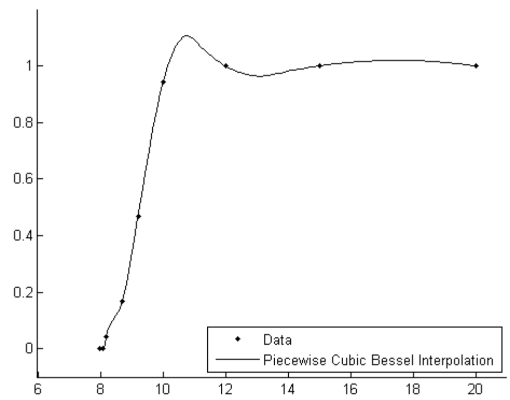

Figure 1.1. *Piecewise Cubic Bessel Interpolation*

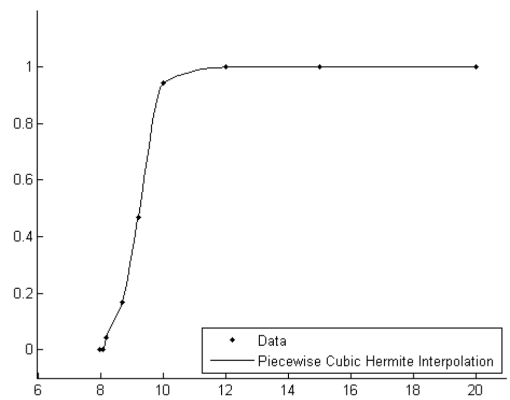

Figure 1.2. *Piecewise Cubic Hermite Interpolation*

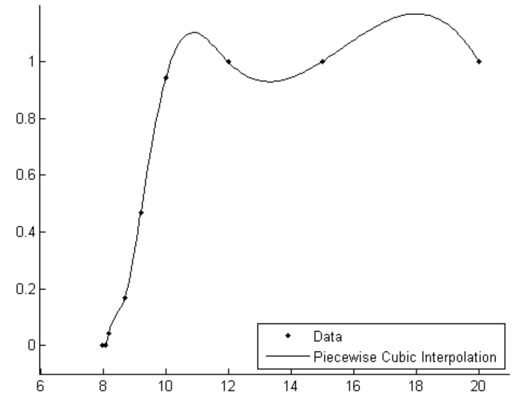

Figure 1.3. *Piecewise Cubic Interpolation*

3.2. Example 2

Here, $\boldsymbol{n = 9}$ and Akima's dataset of $\boldsymbol{x_i}$'s and $\boldsymbol{f_i}$'s are given below,

Table 2. *Akima's dataset*

x_i	f_i
0	10
2	10
3	10
5	10
6	10
8	10
9	10.5
11	15
12	50

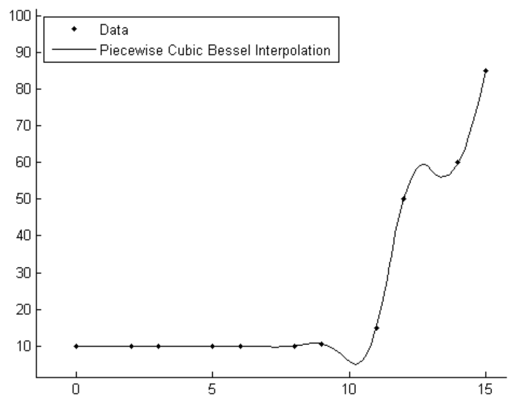

Figure 2.1. *Piecewise Cubic Bessel Interpolation*

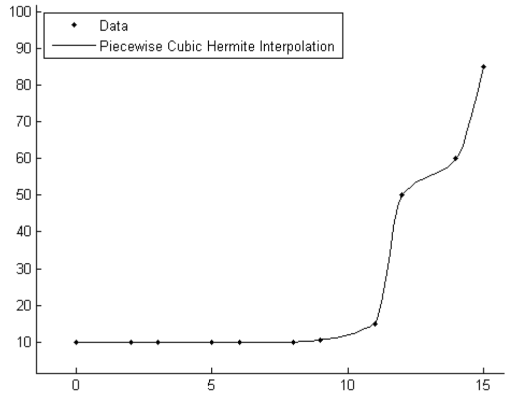

Figure 2.2. *Piecewise Cubic Hermite Interpolation*

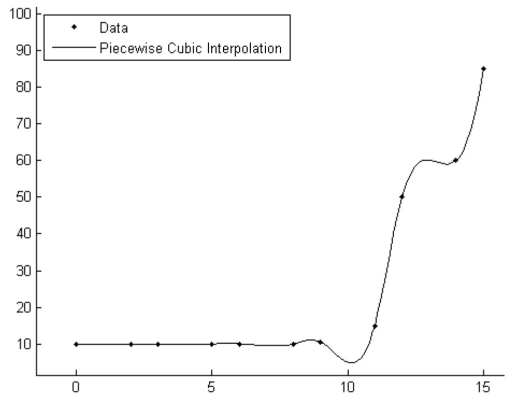

Figure 2.3. *Piecewise Cubic Interpolation*

4. Conclusions

In this paper we proposed different piecewise interpolation techniques for data [7]. Numerical examples are presented to show graphically the differences using the suggested piecewise cubic approximations. For these two examples, we prove that these interpolation techniques produce a pleasing interpolant, when the slopes for the data do not change from a large to a small value. Fig1.2 and Fig.2.2 obtained by piecewise cubic interpolation technique, that changes quickly (due to high order convergence) and are not pleasing as well as others which produced by some of the other interpolation techniques.

Acknowledgements

This study is supported by Kocaeli University, Project No: 2013/44.

References

[1] Tae-Wan Kim, Boris Kvasov, A shape-preserving approximation by weighted cubic splines, Journal of Computational And Applied Mathematics, 236 (2012) 4383–4397.

[2] İ.N. Askerzade Askerbeyli, Şahin Emrah Amrahov, Two approaches for fuzzy interpolation, Journal of Computing, 2010,2151-9617.

[3] F.N. Fritsch and R. E. Carlson, Monotone piecewise cubic interpolation, Siam J.Numer. Anal., 17(1980) 238-246

[4] Carl de Boor, A Practical Guide to Splines, Applied Mathematical Sciences Vol. 27

[5] Abbasbandy S. and Babolian E., Interpolation of fuzzy data by natural splines, Korean J. Comput. Appl. Math. 5 (1998) 457-463.

[6] Abbasbandy S., Interpolation of fuzzy data by complete splines, Korean J. Comput. Appl. Math. 8 (2001) 587-594.

[7] Akima H., A new method of interpolation and smooth curve fitting based on local procedures, J. Assoc. Comput. Mech. 17(1970) 589-602.

[8] Yan Z. By., Piecewise cubic curve fitting algorithm, Math. of Comput. 49(1987) 203-213.

[9] Kershaw D. By., The orders of approximation of the first derivative of cubic splines at the knots, Math. of Comput. 26(1972) 191-198.

[10] Anderson L.E., And Elfving T., Interpolation and approximation by monotone cubic splines, Journal of Approximation Theory, 66(1991) 302-303.