Wind velocity data interpolation using a weighted cubic spline

M.M. Romadanova*

1 Saint Petersburg State University of Architecture and Civil Engineering, 2nd Krasnoarmeyskaya st., 4, 190005, Saint Petersburg, Russia

Abstract. To predict meteorological conditions, it is essential to construct interpolating functions considering the nature of the data. The paper addresses an algorithm for constructing a weighted cubic spline that preserves the monotonicity of the data. The implementation of this algorithm is shown through the example of wind velocity data interpolation when it is required to maintain the positivity of the values. Wind velocity always takes positive values, and its minimum value can be very close to zero. The paper shows that in such a case the construction of an ordinary cubic spline can result in negative values. It presents several numerical examples for the implementation of the proposed algorithm at different sets of initial parameters and makes it possible to draw comparison with the construction of a simple cubic spline and the construction of a spline in MATLAB.

1 Introduction

Spline construction is used in various meteorological applications related to data interpolation. Among such meteorological data, the following can be mentioned: precipitation, extent of snow cover, temperature (maximum, minimum, and average), humidity, as well as wind velocity and direction. However, in practical problem-solving, classical cubic splines – which are twice continuously differentiable functions – may not retain the geometric properties of the initial data, such as positivity, monotonicity, or convexity. Therefore, to ensure the required properties of a function, various modifications of classical cubic splines have been developed.

Wu et al. [1] studied interpolation of scattered data by using compactly supported radial basis functions, subjected to the constraint of preserving positivity. The authors addressed the application of their algorithm to the interpolation of wind velocity data that can only take positive values. Karim and Raju [2] also considered wind velocity data interpolation. In their paper, they constructed a rational cubic spline and derived conditions on parameters, sufficient to preserve the positivity of spline values. Azizan et al. [3] discussed the construction of a cubic spline for rainfall data interpolation. Sarfraz et al. [4] considered the construction of a rational cubic spline with two parameters to preserve the positivity of the data. All mentioned papers addressed spline construction through the example of meteorological data interpolation aimed to ensure the geometric property of positivity.

* Corresponding author: Romadanova@yandex.ru
Kvasov B.I. [5] proposed algorithms for interpolating using weighted cubic splines with the aim of preserving the monotonicity and convexity of the data.

This paper considers a modification of an algorithm for constructing a weighted cubic spline that preserves the monotonicity of the data. It also presents examples of the algorithm implementation for wind velocity data interpolation. The examples given show that in this case preserving the monotonicity of the values makes it possible to retain their positivity.

2 Problem formulation

2.1 Definition of a weighted cubic spline

Let us assume that on the segment \([a, b]\) in grid nodes \(a = x_0 < x_1 < \ldots < x_{N+1} = b\), values of some function are set:

\[
(x_i, f_i), \ i=0, \ldots, N+1.
\] (1)

Let us introduce the following notation:

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

Data (1) shall be called monotone if \(f[f_{x_i}, x_{i+1}] \geq 0, \ i = 0, \ldots N\).

The shape-preserving interpolation problem [5] is to construct a sufficiently smooth function \(S\) such that \(S(x_i) = f_i, \ i=0, \ldots, N+1\), and \(S\) is monotone on the segments where the initial data is monotone.

A weighted cubic spline with the weights \(w_i > 0\), \(i = 0, \ldots, N\) is a function \(S\) such that:

1) between the grid nodes \(x_0, x_1, \ldots, x_{N+1}\), the function \(S\) is a cubic polynomial;
2) \(S \in C^k[a, b], \ k \geq 1\);
3) \(w_i S''(x_i) = w_i S''(x_i^+), \ i = 1, \ldots, N\).

Let us assume that the cubic spline \(S\) satisfies the interpolation conditions:

\[
S(x_i) = f_i, \ i = 0, \ldots, N+1.
\] (2)

For an unambiguous definition of a spline, we also need boundary conditions. The following types of constraints are those used most often:

I. Establishment of boundary values for the first derivative: \(S'(a) = f_0', \ S'(b) = f_{N+1}'\).
II. Establishment of boundary values for the second derivative: \(S''(a) = f_0'', \ S''(b) = f_{N+1}''\).

2.2 Construction of a weighted cubic spline

Let us introduce the following notations:

\[
S'(x_i) = m_i, \ i = 0, \ldots, N+1.
\] (3)

The spline \(S\) can be considered as a Hermite cubic spline that satisfies conditions (2)-(3). Then for \(x \in [x_i, x_{i+1}]\), it can be written as follows:

\[
S(x) = f_i (1-t)^2 (1+2t) + f_{i+1} t^2 (3-2t) + m_i h_i t (1-t)^2 - m_{i+1} h_i t^2 (1-t),
\]

where \(t = (x - x_i) / h_i\).
The cubic spline represented in such a form on each interval \([x_i, x_{i+1}]\) is continuous with its first derivative on the entire \([a, b]\) segment. Using the smoothness condition \(w_i\ S''(x_i) = w_i\ S''(x_{i+1}), i = 1, ..., N\), we obtain the following:

\[
\lambda_i m_i + 2 m_i + \mu_i m_{i+1} = c_i, i = 1, ..., N
\]

(4)

where

\[
c_i = 3 (\mu_i f[x_i, x_{i+1}] + \lambda_i f[x_{i-1}, x_i])
\]

(5)

and

\[
\lambda_i = \frac{w_i h_i}{(w_i h_i + w_i h_{i+1})}, \mu_i = 1 - \lambda_i.
\]

(6)

It is required to add equations arising from the boundary conditions to equations (4). Thus, we obtain a system of \(N+2\) equations to determine \(N+2\) unknowns \(m_i, i = 0, ..., N+1\).

In the case of boundary conditions of type I, we get the following:

\[
2 m_0 = c_0
\]

\[
\lambda_i m_{i+1} + 2 m_i + \mu_i m_{i+1} = c_i, i = 1, ..., N
\]

\[
2 m_{N+1} = c_{N+1}
\]

(7)

where \(c_0 = 2f_0', c_{N+1} = 2f_{N+1}'\) and coefficients \(c_i, \lambda_i, \) and \(\mu_i\) have the form of (5) and (6), respectively.

### 3 Algorithm for constructing a weighted cubic spline that preserves the monotonicity of the data

Kvasov B.I. [5] obtained sufficient conditions for the monotonicity of an interpolating cubic spline. For a monotonically increasing function, the following inequalities shall be fulfilled:

\[
(w_i h_{i+1}) / (w_i h_i) \geq f[x_{i+1}, x_i] / f[x_i, x_{i-1}] - 2,
\]

(8)

\[
(w_{i-1} h_i) / (w_i h_{i-1}) \geq f[x_i, x_{i+1}] / f[x_{i-1}, x_i] - 2,
\]

(9)

Based on these conditions, we can formulate an algorithm for constructing a weighted cubic spline that preserves the monotonicity of the data:

1. First, we determine intervals where the function increases or decreases.
2. Then we determine the first weight parameter \(w_0\) by the following equation:

\[
w_i = (1 + C_i [x_i, x_{i+1}]^2)^{\beta_i}, C_i \geq 1, \beta_i \geq 0, i = 0, ..., N,
\]

choosing some coefficients \(C_0 \geq 1\) and \(\beta_0 \geq 0\).

3. Then we set values for the first derivative: \(f_0'\) and \(f_{N+1}'\), as well as the value for the first derivative at the boundaries of the monotonicity intervals.

4. At each data monotonicity interval, we choose the first parameter \(w_{i-1}\). To ensure spline connection, we take the first parameter \(w_{i-1}\) equal to the last one on the previous segment. If the function monotonically decreases on the interval, we assume that we work with \(-f_i\).

Then we choose \(w_i\) equal to the previous value \(w_{i-1}\). If in this case inequalities (8) and (9) are fulfilled, we assume that \(w_i = w_{i-1}\). If one of the inequalities is not fulfilled, then we calculate \(w_i\) based on the inequality that is not fulfilled at \(w_i = w_{i-1}\), replacing the inequality sign with an equal sign.
If at any step \( w_i < \varepsilon \), then we assume that \( w_i = \varepsilon \), where \( \varepsilon \) is a sufficiently small positive number. This way, we find all weight parameters \( w_i \) on each function increasing or decreasing interval.

5. Finally, we find a solution for the system of equations (7) and construct a weighted cubic spline \( S \) on each monotonicity interval individually.

### 4 Numerical results

Let us show the implementation of this algorithm for the interpolation of hydrometeorological data, namely wind velocity. Table 1 presents wind velocity data [2].

<table>
<thead>
<tr>
<th>Times (min)</th>
<th>0</th>
<th>0.25</th>
<th>0.5</th>
<th>1</th>
<th>1.2</th>
<th>1.8</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity (km/min)</td>
<td>2</td>
<td>0.8</td>
<td>0.5</td>
<td>0.1</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
</tr>
</tbody>
</table>

The velocity of wind is inherently positive; therefore, we need the resulting approximating function to preserve this geometrical property.

Fig. 1 shows data interpolation using a cubic spline, which can be considered as a weighted cubic spline with all weight coefficients equal to one. Here and in all figures below, the points denote the initial data, and the line denotes the constructed interpolating spline. As can be seen in the figure, when an ordinary cubic spline is constructed, there is an interval on which the constructed spline takes negative values.

![Fig. 1. Example of an interpolating curve using a cubic spline](image)

Fig. 2 shows the interpolation of the data from Table 1 using the MATLAB interp1 function. In this case, the pHIP construction method was applied, which uses shape-preserving piecewise cubic interpolation. The pHIP function can preserve the positivity of wind velocity, but the interpolating curve is not visually pleasing since on some intervals, the interpolating curves are very tight as well as very sharp [2].
Let us show now the construction of a weighted cubic spline using the proposed algorithm. For the first example, we determine the function increasing and decreasing intervals: at $t \in [0, 1]$, the function decreases; at $t \in [1, 1.2]$, the function increases; at $t \in [1.2, 1.8]$, the function decreases; at $t \in [1.8, 2]$, the function increases.

Let us set the first weight parameter by Eq. (10) with parameters $C_0 = 10$, $\beta_0 = 0.1$.

Let us then set values for the first derivative: $f_0' = 0$ and $f_{N+1}' = 0$, as well as equal to zero at the ends of the function increasing and decreasing intervals.

At each data monotonicity interval, we determine the weight parameters $w_i$ according to the algorithm.

Then we find a solution for the system of equations (7) and construct a weighted cubic spline on each monotonicity interval individually.

Fig. 3 shows the resulting weighted cubic spline. Table 2 presents values of the first derivative and weight parameters.
### Table 2. Values of the first derivative and weight parameters for Fig. 3

<table>
<thead>
<tr>
<th>Function increasing /decreasing intervals</th>
<th>Times (min)</th>
<th>Velocity (km/min) at the beginning of the interval</th>
<th>Velocity (km/min) at the end of the interval</th>
<th>( f_i' )</th>
<th>( f_{i+1}' )</th>
<th>( w_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>decreases [0, 0.25]</td>
<td>2</td>
<td>0.8</td>
<td>0</td>
<td>-3.45</td>
<td>0.5802</td>
<td></td>
</tr>
<tr>
<td>decreases [0.25, 0.5]</td>
<td>0.8</td>
<td>0.5</td>
<td>-3.45</td>
<td>-0.45</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>decreases [0.5, 1]</td>
<td>0.5</td>
<td>0.5</td>
<td>-0.45</td>
<td>0</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>increases [1, 1.2]</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>decreases [1.2, 1.8]</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>increases [1.8, 2]</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>2.5</td>
<td>1.1604</td>
<td></td>
</tr>
</tbody>
</table>

We can change the shape of the interpolating curve by setting other values of the first derivative on the boundaries of the intervals. For the next example, we determine the values of the first derivative at the first and last points by the following equations:

\[
 f_0' = \frac{(f_1 - f_0)}{(x_1 - x_0)}, \quad f_{N+1}' = \frac{(f_{N+1} - f_N)}{(x_{N+1} - x_N)}.
\]

At intermediate points, similarly to the previous example, we take the values of the first derivative equal to 0.

### Table 3. Values of the first derivative and weight parameters for Fig. 4

<table>
<thead>
<tr>
<th>Function increasing /decreasing intervals</th>
<th>Times (min)</th>
<th>Velocity (km/min) at the beginning of the interval</th>
<th>Velocity (km/min) at the end of the interval</th>
<th>( f_i' )</th>
<th>( f_{i+1}' )</th>
<th>( w_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>decreases [0, 0.25]</td>
<td>2</td>
<td>0.8</td>
<td>-4.8</td>
<td>-2.55</td>
<td>0.5802</td>
<td></td>
</tr>
<tr>
<td>decreases [0.25, 0.5]</td>
<td>0.8</td>
<td>0.5</td>
<td>-2.55</td>
<td>-0.75</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>decreases [0.5, 1]</td>
<td>0.5</td>
<td>0.1</td>
<td>-0.75</td>
<td>0</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>increases [1, 1.2]</td>
<td>0.1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>decreases [1.2, 1.8]</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>1.1604</td>
<td></td>
</tr>
<tr>
<td>increases [1.8, 2]</td>
<td>0.5</td>
<td>1</td>
<td>2.5</td>
<td>1.1604</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparing Fig. 3 and Fig. 4, we can see differences in the shape of the outermost splines.

5 Discussion and conclusion

In the course of the work, an algorithm for interpolation using a weighted cubic spline that preserves the monotonicity of the data was developed. The algorithm is based on the determination of the function increasing and decreasing intervals. In this case, the values of weight parameters are chosen in such a way that \( w_i \cdot S''(x_i -) = w_i \cdot S''(x_i +) \). Then a weighted cubic spline is constructed on each monotonicity interval individually so that it would be possible to preserve the continuity of the first derivative. Based on a numerical example of wind velocity data interpolation, it was shown that this algorithm can be used to construct an interpolating spline retaining positive values.

References