Monotonicity Preserving using Ball Cubic Interpolation

Samsul Ariffin Abdul Karim

Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak Darul Ridzuan, Malaysia

Abstract: This paper studies the use of Ball cubic interpolant for monotonicity preserving interpolation. The shape parameters in the description of the cubic interpolant are subjected to the monotonicity constrained. The necessary and sufficient conditions for the monotonicity of the cubic interpolant have been derived. Two propositions for the monotone preserving have been developed. The main finding indicated that the proposed cubic interpolant gives the very visual pleasing results. By having $G^1$ continuity, the user will have greater flexibility to control the shape of the monotonic interpolating curves without needing to modify the derivative.

Keywords: monotonicity, Ball cubic interpolant, shape parameter, necessary and sufficient conditions.

INTRODUCTION

Shape preserving interpolations are important in Computer Graphics (CG) and Scientific Visualization. Monotonicity preserving exists in various sciences and engineering applications. For examples the devices in the specification of Digital to Analog Converters (DACs), Analog to Digital Converters (ADCs) and sensors are always monotonic and any non-monotonicity existing in the polynomial interpolation are unacceptable (Hussain and Hussain, 2007). Approximation of couples and quasi couples in statistics and the dose-response curves and surfaces in biochemistry and pharmacology are other examples in which the monotonicity exists in the data sets (Beliakov, 2005). Shape preserving for monotone data have been discussed in details by previous researchers in this rapid growing fields. For examples, Fristch and Butland (1984), and Fristch and Carlson (1980) have proposed the monotonicity preserving using cubic spline. In their construction, the existence of the monotonicity interpolant depends to the monotonicity region which is combined by rectangle and ellipses. Passow and Roulier (1977) also consider the monotone and convex shape preserving using polynomial quadratic spline. Schumaker (1983) and Lahtinen (1996) discussed the shape preserving interpolation using quadratic spline polynomial. They preserved the data (monotone and convex data) by inserting extra knots in which shape violation are found. Butt and Brodlie (1993) study the positivity preserving using cubic spline polynomial by inserting extra knots (one or two). Having an extra knot, the computation to generate the cubic interpolating curves will be increased. Furthermore it is not an easy task to teach the user about how to insert the knots.

Besides that, one may use the rational interpolant for shape preserving. For example, Sarfraz (2000, 2003), Sarfraz et al. (2001), Hussain and Hussain (2007), Hussain et al. (2010) and the references cited therein, have studied the use of rational cubic spline with cubic, quadratic and linear denominator respectively for the shape preserving interpolation. Sarfraz et al. (2005) has proposed cubic spline interpolation for positive, monotone and convex data preservation with $G^1$ continuous. Motivated by this paper, the authors in Karim et al. (2012a) have proposed Ball cubic interpolation for positivity preserving by giving two sufficient conditions in which the interpolant will be positive. The results show that by using Proposition 2 in that paper, Ball cubic interpolant gives better results (in terms of the smoothness) of the interpolating curves. This paper is the continuation of our earlier work in Karim et al. (2012a, 2012b). The main feature of our proposed Ball cubic interpolation for monotonicity preservation is the derivation of the necessary and sufficient conditions for the cubic interpolant to be monotone. This condition provides greater flexibility in controlling the monotonicity of the data. Our main result slightly gives better results as compared with the work by Sarfraz et al. (2005).

The remainder of the paper is organized as follows. Section 2 introduces the cubic Ball polynomial, Section 3 discusses the Ball cubic interpolant and Section 4 discusses the proposed interpolant in the context of monotonicity preservation together with the implementation algorithm. All numerical results will be discussed in Section 5 including comparison with Sarfraz et al. (2005), and Fristch and Carlson (1980). The coding was developing in Matlab called pchip. Summary and conclusions are given in Section 6.
Ball Cubic Interpolant:

In this section, a GC\(^1\) Ball cubic interpolant proposed in Karim et al. (2012a, 2012b) will be reviewed. Suppose that \( \{(x_i, f_i), i = 1, \ldots, n\} \) is a given set of data points, where \( x_1 < x_2 < \ldots < x_n \).

Let \( h_i = x_{i+1} - x_i \), \( \Delta_i = \frac{(f_{i+1} - f_i)}{h_i} \) and a local variable, \( \theta = \frac{(x - x_i)}{h_i} \) where \( 0 \leq \theta \leq 1 \).

Now for \( x \in [x_i, x_{i+1}], i = 1, 2, \ldots, n-1 \),

\[
s(x) = s(x_i + h_i \theta) \equiv S_i(\theta),
\]

Where

\[
S_i(\theta) = A_0 (1 - \theta)^2 + 2 A_1 \theta(1 - \theta)^2 + 2 A_2 \theta^2 (1 - \theta) + A_3 \theta^2
\]

To make the cubic function (1) be GC\(^1\), one needs to impose the following interpolatory properties:

\[
s(x_i) = f_i, \quad s(x_{i+1}) = f_{i+1},
\]

\[
s^{(1)}(x_i) = \frac{d}{r_i}, \quad s^{(1)}(x_{i+1}) = \frac{d_{i+1}}{r_i},
\]

where \( s^{(1)}(x) \) denotes derivative with respect to \( x \) and \( d_i \) denotes the derivative value which is given at the knot \( x_i, i = 1, 2, \ldots, n \). The parameters \( r_i \) \((r_i > 0, r_i \neq 1)\) will be constrained in order to generate the positive Ball cubic interpolant on entire given interval \([x_i, x_{i+1}], i = 1, 2, \ldots, n-1\).

Hence, it can be shown that, the unknowns \( A_i, i = 0, 1, 2, 3 \) have the following values:

\[
A_0 = f_1, \quad A_1 = f_{i+1},
\]

\[
A_i = f_i + \frac{h_i d_i}{2r_i}, \quad A_2 = f_{i+1} - \frac{h_i d_{i+1}}{2r_i},
\]

Thus the Ball cubic interpolant \( S \in GC^1[x_1, x_n] \) in (1) can be rewritten as follows:

\[
S(x) = S_i(\theta),
\]

where

\[
S_i(\theta) = f_i (1 - \theta)^2 + 2 \left(f_i + \frac{h_i d_i}{2r_i}\right) \theta (1 - \theta)^2 + 2 \left(f_{i+1} - \frac{h_i d_{i+1}}{2r_i}\right) \theta^2 (1 - \theta) + f_{i+1} \theta^2
\]

When \( r_i = 1 \), the Ball cubic interpolant (3) reduces to \( C^1 \) Ball cubic polynomial in Hermite-like form which is in general not a shape preserving interpolant. The shape parameters, \( r_i \) with \( i = 1, 2, \ldots, n-1 \) can be utilized in order to modify the shape of the interpolating curve either to tighten or loosen.

It should be noted that in most applications, normally, the first derivative \( d_i, i = 1, 2, \ldots, n \), will be not given and its must be estimated by using method initiated by Delbourgo and Gregory (1985). In this paper, the mean arithmetic method will be used to estimate the derivative.

Monotonicity Preserving Interpolation:

The Ball cubic interpolant in (3) does not necessarily preserve monotonicity of the data sets. Figure 1 and Figure 2 shows the results when the default Ball cubic interpolant has been used to interpolate the monotone data. Clearly at certain interval the shape violations are found.

The user may control the shape preserving (in our case it is monotone) by keep changing the shape parameter values, \( r_i \). This is really time consuming and does not give any flexibility to the user in controlling the shape of the monotone data. Thus, it is desirable to achieved shape preserving interpolation for monotone data through an automated and efficient ways. This can be done by doing some mathematical derivations. Let
\((x_i, f_i), i = 1, \ldots, n\) be a given monotone data set, where \(x_1 < x_2 < \ldots < x_n\). For a strictly monotonic increasing (decreasing), the necessary condition should be \(f_1 < f_2 < \ldots < f_n\) (or \(f_1 > f_2 > \ldots > f_n\) for monotonic decreasing), i.e. \(\Delta_i \geq 0\) (or \(\Delta_i \leq 0\) for monotonic decreasing data). In this section, the necessary and sufficient condition for the \(GC^1\) monotonicity of Ball cubic interpolant will be derived by finding the conditions on shape parameter, \(r_i\). For a monotone Ball cubic interpolant \(S(x)\) in (3), the first derivative must satisfy the following inequalities:

\[
d_i \geq 0, i = 1, 2, \ldots, n (\text{for monotonic increasing}) \tag{4}
\]
\[
d_i \leq 0, i = 1, 2, \ldots, n (\text{for monotonic decreasing}) \tag{5}
\]

From basic calculus, \(S'(x)\) is monotonic increasing if and only if

\[
S'(x) \geq 0, \quad x_i \leq x \leq x_{i+1} \tag{6}
\]

Now,

\[
S'(x) = a_i \theta^2 + b_i \theta + c_i \tag{7}
\]

where

\[
a_i = 3\left(\frac{d_i + d_{i+1}}{r_i}\right) - 6\Delta_i,
\]
\[
b_i = 6\Delta_i - \frac{4d_i}{r_i} - \frac{2d_{i+1}}{r_i},
\]
\[
c_i = \frac{d_i}{r_i}.
\]

Now, let \(\theta = \frac{s}{1+s}, s \geq 0\), \(S'(x)\) in (7) can be rewritten in the following form:

\[
U(s) = \beta_i \theta^2 + \gamma_i \theta + \delta_i \tag{8}
\]

where

\[
\beta_i = \frac{d_{i+1}}{r_i},
\]
\[
\gamma_i = 2\left(3\Delta_i - \frac{d_i + d_{i+1}}{r_i}\right),
\]
\[
\delta_i = \frac{d_i}{r_i}.
\]

Applying the results in Proposition 3 by Schmidt and Hess [17], one can show that the following conditions are required for the interpolant in (3) to reproduce monotone (increasing) interpolating curves:

\[
r_i > \frac{d_i + d_{i+1}}{3\Delta_i} \tag{9}
\]

Equation (9) can be summarized as Proposition 1 below.

**Proposition 1** (Monotonicity of Ball Cubic Interpolant)

For a strictly monotone data and \(\Delta_i > 0\), the Ball Cubic Interpolant (defined over the interval \([x_1, x_n]\) is monotone with \(S \in GC^1[x_1, x_n]\) if in each subinterval \([x_i, x_{i+1}]\), \(i = 1, 2, \ldots, n-1\) the following sufficient condition is satisfied:

782
\[ r_i = l_i + \frac{d_i + d_{i+1}}{3\Delta_i}, l_i > 0 \]  

**Remarks 1:** After the values of the shape parameters \( r_i, i = 1, 2, ..., n-1 \), is calculated using (10), it should be noted that its values must not be equal to 1 for \( GC^d \) otherwise the resultant interpolant will be \( C^1 \) at the join point.

**Remarks 2:** If the data are constant on certain interval, i.e. \( \Delta_i = 0 \), then it is necessary to set \( d_i = d_{i+1} = 0 \), hence \( S(x) = f_i = f_{i+1} \) is a constant on the interval \([x_i, x_{i+1}]\), \( i = 1, 2, ..., n-1 \).

**Remarks 3:** The sufficient conditions in (10), is the same as in Safraz et al. (2005). The main difference is that, our use of Ball cubic interpolant while Safraz et al. (2005) use cubic spline polynomial.

From Schmidt and Hess (1988), the following is necessary and sufficient conditions for the monotonicity of Ball cubic interpolant.

**Proposition 2.** For a strictly monotone data and \( \Delta_i > 0 \), the Ball Cubic Interpolant (defined over the interval \([x_i, x_n]\)) is monotone with \( S \in GC^d \) \([x_i, x_n]\) if in each subinterval \([x_i, x_{i+1}]\), \( i = 1, 2, ..., n-1 \) the following necessary and sufficient conditions are satisfied:

\[ r_i = \gamma_i + \frac{d_i + d_{i+1} - \sqrt{d_i d_{i+1}}}{3\Delta_i}, \gamma_i > 0 \]  

**Proof**

From (8) and the results by Schmidt and Hess (1988),

\[ \gamma_i > -2\sqrt{\beta_i \delta_i}, \]  

This provides the following algebraic manipulations:

\[ 2 \left( 3\Delta_i - \frac{d_i + d_{i+1}}{r_i} \right) > -2 \frac{\sqrt{d_i d_{i+1}}}{r_i}, r_i > 0 \]  

\[ 3\Delta_i - \frac{d_i + d_{i+1}}{r_i} > -\frac{\sqrt{d_i d_{i+1}}}{r_i} \]

\[ r_i > \frac{d_i + d_{i+1} - \sqrt{d_i d_{i+1}}}{3\Delta_i} \]

This completes the proof. Condition in (11) provides more visual pleasing results as compare with the conditions in (10). Indeed, the resultant shape preserving interpolation monotone curves are smooth compared to the Sarfraz et al. (2005) method.

**Remarks 5:** The necessary and sufficient condition in (11) is also valid for Sarfraz et al. (2005) method. Figure 6 and Figure 10 shows the results when method in Sarfraz et al. (2005) is applied with condition (11).

**Remarks 6:** To accommodate both monotonic increasing and decreasing, the values of shape parameter in Proposition 1 and Proposition 2 can be modified as follows

\[ r_i > \frac{|d_i| + |d_{i+1}|}{3|\Delta_i|} \]  

and

\[ r_i > \frac{|d_i| + |d_{i+1}| - \sqrt{d_i d_{i+1}}}{3|\Delta_i|} \]

In this paper, the proposed method will be tested on strictly monotone increasing data sets only. Similar idea may be used for strictly monotone decreasing.

An algorithm to generate \( C^1 \) monotonicity-preserving curves using the results in Proposition 1 or Proposition 2 is given as follows.
Algorithm for monotonicity-preserving:

1. Input the number of data points, \( n \), and data points \( \{x_i, f_i\}_{i=1}^{n} \).
2. For \( i = 1, 2, \ldots, n \), estimate \( d_i \) using arithmetic mean method (AMM).
3. For \( i = 1, 2, \ldots, n-1 \)
   - Define \( h_i \) and \( \Delta_i \)
   - Calculate the shape parameter \( r_i \) using (10) or (11) and
     \[ r_n = l_n + \frac{d_{n-1} + d_n}{3\Delta_{n-1}}, \]
   - Calculate the inner control ordinates \( A_1 \) and \( A_2 \) and generate the piecewise interpolating monotone curves using (3).

RESULTS AND DISCUSSION

In order to test the applicability of our proposed Ball cubic interpolation with \( GC^3 \) continuity, two sets of data taken from Akima (1970) and Sarfraz (2000) were used.

**Table 1:** A monotone data from Akima (1970).

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10.5</td>
<td>15</td>
<td>50</td>
<td>60</td>
<td>85</td>
</tr>
</tbody>
</table>

**Table 2:** A monotone data from Sarfraz (2000).

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>6</th>
<th>10</th>
<th>29.5</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>0.01</td>
<td>15</td>
<td>15</td>
<td>25</td>
<td>30</td>
</tr>
</tbody>
</table>

**Fig. 1:** Default Ball cubic interpolation for data in Table 1.
**Fig. 2:** Default Ball cubic interpolation for data in Table 2.

**Fig. 3:** Shape preserving interpolation Ball cubic interpolant using Proposition 1 with $l_i = 0.2, 0.2, 0.2, 0.05, 0.05$ for data in Table 1.

**Fig. 4:** Shape preserving interpolation Ball cubic interpolant using Proposition 2 with $\zeta_i = 0.6$ for data in Table 1.
Fig. 5: Shape preserving interpolation using Sarfraz et al. (2005) for data in Table 1.

Fig. 6: Shape preserving interpolation using Sarfraz et al. (2005) with proposed Proposition 2 ($\zeta_i = 0.4$) for data in Table 1.

Fig. 7: Shape preserving interpolation Ball cubic interpolant using Proposition 1 with $l_i = 0.2, 0.2, 0.02, 0.02$, for data in Table 2.
Fig. 8: Shape preserving interpolation Ball cubic interpolant using Proposition 2 with \( \zeta_i = 0.2, 0.2, 0.02, 0.02 \) for data in Table 2.

Fig. 9: Shape preserving interpolation using Sarfraz et al. (2005) for data in Table 2.

Fig. 10: Shape preserving interpolation using Sarfraz et al. (2005) with proposed Proposition 2 \( (\zeta_i = 0.2, 0.2, 0.02, 0.02) \) for data in Table 1.
Fig. 11: Shape preserving interpolation using \textit{pchip} in Matlab for data in Table 1.

Fig. 12: Shape preserving interpolation using \textit{pchip} in Matlab for data in Table 2.

Figure 1 and Figure 2 show the default $C^1$ Ball cubic interpolation for data in Table 1 and Table 2 respectively. Figure 3 and Figure 4 shows the results of $GC^1$ Ball cubic interpolation for data sets in Table 1 using Proposition 1 and Proposition 2 respectively. Clearly shape preserving monotonic curves using Proposition 2 gives more smooth result compare to monotonic curves using Proposition 1. This fact can be noticed on the interval $[12, 14]$. Figure 5 shows the monotonic curves using Sarfraz et al. (2005) and Figure 6 shows monotonic curves using Sarfraz et al. (2005) our Proposition 2. Figure 7 and Figure 8 show monotonic curves using Ball cubic interpolation with Proposition 1 and Proposition 2 for data in Table 2 respectively. Figure 9 and Figure 10 show shape preserving interpolation using Sarfraz et al. (2005). Finally Figure 11 and Figure 12 show the shape preserving interpolation using piecewise cubic Hermite interpolating polynomial (pchip) as documented in Matlab. From all the figures, it can be noticed that the proposed Proposition 2 gives smoother results as compared with Proposition 1. In fact our Proposition 2 was proven can be applied in Sarfraz et al. (2005) technique (see Figure 6 and Figure 10). From inspection, the proposed $GC^1$ Ball cubic interpolation works well for both data sets and the results are comparable with the original Fristch and Carlson (1980) method (\textit{pchip} function in Matlab). With $GC^1$ continuity, the modification of the derivative in which the necessary and sufficient conditions for monotonicity are violated is not required. To apply pchip, for data in Table 1, the derivative to be modified are $d_6,d_9,$ and $d_{10}$ and for data sets in Table 2, $d_4$ required to be modified. Another
interesting point is that, the proposed $G^1$ Ball cubic interpolation scheme provides good alternative to the existing method for shape preserving using any spline polynomial with degree three.

**Final Remark:** At the end point of the interpolation, the shape parameter $r_n = l_n + \frac{d_{n-1} + d_n}{3\Delta n-1}$, $l_n > 0$.

**Conclusion:**

The main topic being addressed in this paper is the preservation of the monotone data (in our case the monotonic increasing) using $G^1$ Ball cubic interpolation scheme. Two propositions for the monotonicity interpolant have been proposed. The results show that our proposed Ball cubic interpolant works well as compared with the method by Sarfraz et al. (2005). Indeed, using Proposition 2, the obtained results are much smoother especially near the join point compared to the work by Sarfraz et al. (2005). Our proposed method does not constrain the derivatives, thus there is no need to modify the derivative on interval in which the necessary and sufficient conditions for monotonicity are violated as originally proposed by Fritsch and Butland (1984) and Fritsch and Carlson (1980). It is our opinion, to suggest that, in order to preserves the monotone data, the user are encouraged to use our proposed method with Proposition 2. Furthermore, the results also show that the proposed schemes provides greater flexibility to the user in controlling and manipulating the resulting monotonic interpolating curves due to its easiness to control the shape of the curves by using $G^1$ compared to $C^1$ continuity. The curve scheme can also be generalized to the surface cases. The author is keen to discuss it in his forthcoming papers.

**ACKNOWLEDGMENTS**

The author would like to acknowledge Universiti Teknologi PETRONAS (UTP) for the financial support received in the form of a research grant: Short Term Internal Research Funding (STIRF) No. 76/10.11 and 35/2012.

**REFERENCES**


