Abstract. Bézier curve is a parametric polynomial that is applied to produce good piecewise interpolation methods with more advantage over the other piecewise polynomials. It is, therefore, crucial to construct Bézier curves that are smooth and able to increase the accuracy of the solutions. Most of the known strategies for determining internal control points for piecewise Bezier curves achieve only partial smoothness, satisfying the first order of continuity. Some solutions allow you to construct interpolation polynomials with smoothness in width along the approximating curve. However, they are still unable to handle the locations of the inner control points. The partial smoothness and non-controlling locations of inner control points may affect the accuracy of the approximate curve of the dataset. In order to improve the smoothness and accuracy of the previous strategies, a new piecewise cubic Bézier polynomial with second-order of continuity $C^2$ is proposed in this study to estimate missing values. The proposed method employs geometric construction to find the inner control points for each adjacent subinterval of the given dataset. Not only the proposed method preserves stability and smoothness, the error analysis of numerical results also indicates that the resultant interpolating polynomial is more accurate than the ones produced by the existing methods.

Keywords: Interpolation Polynomial, Bézier Curve, Bézier Spline, SSE, MAE, RMSE

1. Introduction. The Missing values of dataset are the common issues in many areas of sciences such as statistics, computer sciences, and geophysics [1-3]. Several interpolation methods employed piecewise polynomials to estimate missing values. One of them is Bezier curve which is a parametric polynomial used extensively in computer-aided design (CAD) [4, 5], numerical analysis [6, 7], hitch avoidance path determination of unicycle robots [8, 9], lane changing [10, 11], and roundabouts [12, 13] due to its flexibility, stability, and simplicity in representation. By taking the advantages of Bézier curve, researchers started to construct a piecewise cubic Bézier curve at every subinterval of data points in order to improve the smoothness of the interpolating polynomial and consequently increase the accuracy.

Ge and Kang [14] proposed two algorithms of piecewise Bezier functions. The first algorithm produces an approximation function for a dataset, while, the resultant function in the second algorithm interpolates through a dataset. However, both algorithms only satisfy the second order geometric continuity ($G^2$). In Pollock [15], piecewise cubic Bézier curves with the second order of continuity ($C^2$) have been achieved by adopting the construction of a natural cubic spline strategy. Three years later, a geometric technique piecewise Bezier interpolating was proposed by Shemanarev [16]. The resultant polynomial seemed smooth at data points, behaving like the first order geometric continuity ($G^1$) although he did not test the order of continuity. Yau and Wang [17] pre-
sented a new method for deriving piecewise cubic Bézier interpolating polynomial with the first order of continuity \( C^1 \). Nonetheless, the calculations required in this strategy are very time-consuming. Subsequently, Saaban, Zainudin, and Bakar [18] combined Bézier and Said-Ball functions to estimate the missing values of solar radiation in Kedah. In order to further improve the estimation of the missing values of solar radiation datasets in Penang, Karim [19] preserved the positivity and monotonicity by deriving sufficient conditions for rational cubic Ball interpolant. However, the cubic Ball interpolation satisfies the first order of continuity \( C^1 \). To control the piecewise Bezier curves, Saaban, Zainudin, and Abu Bakar [20] constructed piecewise parametric polynomials with \( C^1 \) continuity by imposing sufficient positivity conditions for Bézier curve. The same year, Ueda et al. [21] proposed an algorithm using multi objective simulated annealing to determine a piecewise cubic Bézier Polynomial with \( C^1 \) continuity. However, their resultant polynomial is not interpolated through all data points, which means that the polynomial is only an approximate curve. By taking the advantage of the diagonal matrix, Stelia, Potapenko, and Sirenko [22] determined the coefficients of linear system equations to find the inner control points. Unfortunately, the resulted piecewise cubic Bezier polynomial only fulfils the first order of continuity \( C^1 \). Moreover, an improvement of image upscaling resolution has been attempted by Zulkifli et al. [23] utilizing a rational cubic Ball function.

In this article, a geometric structure of piecewise parametric interpolating polynomial employing cubic Bézier curves is proposed for locating the inner control points.

2. Piecewise Cubic Bézier Curve. A piecewise cubic Bézier curve is constructed by a sequence of cubic Bézier curves interpolated at the data points \( W_i = (x_i, y_i) \), \( i = 0, \ldots, n, \) to produce a smooth and continuous curve (refer to Fig. 1). According to Elber [24] and Quarteroni, Sacco, and Saleri [25], Bézier function \( P(t) \) is given by:

\[
P(t) = \sum_{i=0}^{n} p_i B_i^n(t),
\]

where \( p_i = (p_{x_i}, p_{y_i}) \) are a control points, and \( B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \) are a Bernstein polynomials, where:

\[
\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad \text{for all } i.
\]
By using a cubic Bézier curve between every two adjacent data points [26], the following piecewise parametric polynomial \( F(t) \) of cubic Bézier curves \( P_k(t) \) is constructed:

\[
F(t) = \begin{cases} 
P_0(t) & t \in [0,1], \\
P_1(t) & t \in [0,1], \\
\vdots & \vdots \\
P_{n-1}(t) & t \in [0,1], 
\end{cases}
\]

for all \( k = 0, \ldots, n-1 \), where

\[
P_k(t) = (1-t)^3 p_0^k + 3(1-t)^2 t p_1^k + 3(1-t) t^2 p_2^k + t^3 p_3^k
\]

(1)

and

\[
P_k(t) = (x_k(1), y_k(t)),
\]

with

\[
x_k(t) = (1-t)^3 p_{x0}^k + 3(1-t)^2 t p_{x1}^k + 3(1-t) t^2 p_{x2}^k + t^3 p_{x3}^k,
\]

and

\[
y_k(t) = (1-t)^3 p_{y0}^k + 3(1-t)^2 t p_{y1}^k + 3(1-t) t^2 p_{y2}^k + t^3 p_{y3}^k.
\]

Fig. 1. Cubic bézier curve between every two adjacent data points
Each subinterval requires two inner control points and two end points for constructing a cubic Bézier spline. Since there are \( n \) subintervals, \( 2n \) inner control points are needed to construct \( n \) cubic Bézier splines. The control points for each cubic Bézier curve is given by:

\[
p^k_\zeta = (p^{k}_{x\zeta}, p^{k}_{y\zeta}),
\]

where \( \zeta = 0,\ldots,3 \).

Several previous studies have found good strategies for locating the inner control points for piecewise Bézier curves. Most of them, however, achieve partial smoothness by satisfying the first order of continuity \( C^1 \) as in Saaban et al. [18, 20], Stelia et al. [22], and Zulkifli et al. [23]. Although, some researchers achieved to construct interpolating polynomials with wider smoothness along the approximating curve, including Pollock [15], they are still unable to handle the locations of the inner control points. The partial smoothness and/or non-controlling locations of inner control points may affect the accuracy of the approximate curve of the dataset.

3. Proposed Piecewise Cubic Bézier Polynomial. In order to improve the smoothness and accuracy of the previous strategies, a new piecewise interpolating polynomial known as \( C^2 \) Geometric Bézier Polynomial (C2GBP) is proposed in this study. To construct this polynomial, the inner control points will be located geometrically depending on the polygon of the dataset.

3.1. Construction of C2GBP. In the section, the construction of the new piecewise interpolating polynomial is discussed. The procedure details for constructing C2GBP are as follows:

**Step 1.** Find the straight lines in all subintervals.

Let \( W_k W_{k+1} \) be straight lines in subintervals, where \( k = 0,\ldots,n-1 \), as shown in Figure 2.

![Fig. 2. Construction of C2GBP](image)
The slope $\omega_k$ is defined as:

$$\omega_k = \frac{y_{k+1} - y_k}{x_{k+1} - x_k},$$

whose y-intercept is:

$$\delta_k = y_k - \omega_k x_k.$$

**Step 2.** Find the straight lines connecting $W_{j-1}$ with $W_{j+1}$ where $j = 0, \ldots, n - 1$.

Let $W_{j-1}W_{j+1}$ be straight lines as shown in Figure 2 with slopes $a_j$ defined by:

$$a_j = \frac{y_{j+1} - y_{j-1}}{x_{j+1} - x_{j-1}},$$

whose y-intercepts are:

$$b_j = y_{j-1} - a_j x_{j+1}.$$

The equations for straight lines of $W_{j-1}W_{j+1}$ are $y = a_j x + b_j$.

**Step 3.** Find y-intercepts of the straight lines pass through $W_1$ with slope $a_1$.

Let $\Omega_1$ be a straight line passing through $W_1$ with slope $a_1$ (refer to Fig. 3.1). The y-intercepts of the line is defined as:

$$\sigma_1 = y_1 - a_1 x_1.$$  \hspace{0.5cm} (2)

**Step 4.** Find interception point $(p = (p_x, p_y))$ of the straight line $W_0W_2$ with perpendicular of the same straight line passing through $W_1$. 
Let $\nu$ be a straight line perpendicular to $\overline{W_0W_2}$ passing through $W_1$. Therefore, the slope of $\nu$ is $-\frac{1}{a_1}$, while $y$-intercept is given by:

$$b_1^\nu = y_1 + \frac{1}{a_1}x_1,$$

therefore, the interception point is defined as:

$$\rho_x = \frac{b_1^\nu - b_1}{a_1 + \frac{1}{a_1}},$$

and $y$-intercept is:

$$p_y = a_1p_x + b_1.$$

**Step 5.** Find $y$-intercept $(h_0)$ of the straight line with slope $\omega_0$.

Let $\gamma_0$ be a value calculated by the distance $\Im(W_1\rho_0)$ between $W_1$ and $\rho_0$, where:

$$\Im(W_1\rho_0) = \sqrt{(y_1 - \rho_{\gamma_0})^2 + (x_1 - \rho_{\gamma_0})^2}.$$

The value of $\gamma_0$ is defined as:

$$\gamma_0 = \frac{m\Im(W_0W_1)}{q},$$

where $m$ is the number of inner control points, and $q$ is the degree of the polynomial in each sub-interval. In our construction, the values of $q$ and $m$ are 3 and 2, respectively.

Then, $y$-intercepts is defined by:

$$h_0 = \delta_0 \pm \gamma_0,$$

where the positive/negative value of $\gamma_0$ is determined using algorithm below:
Input: \( W_0, \omega_0, \delta_0, b_1, \sigma_1, \Xi(W_0W_1), \Xi(W_1\rho_0) \).

Output: \( y \)-intercepts \( (h_0) \) of a straight line parallel to \( W_0W_1 \) with a distance of \( \gamma_0 \).

Start

\[
\gamma_0 = \left(2\Xi(W_0W_1)\right)/3
\]

if \( b_1 \leq \sigma_1 \) then

\[
h_0 = \delta_0 + \gamma_0
\]

else if \( b_1 > \sigma_1 \) then

\[
h_0 = \delta_0 - \gamma_0
\]

End

Step 6. Find the inner control point \( p_2^0 \) in the first subinterval.

The inner control point of the first subinterval \( p_2^0 \) is defined as interception of the line \( \Phi_0 \) with the line \( \Omega_1 \), where \( \Phi_0 \) is the straight line given as \( y = \omega_0x - h_0 \).

Hence,

\[
p_{x_2}^0 = \frac{h_0 - \sigma_1}{a_1 - \omega_0},
\]

and \( y \)-intercept is:

\[
p_{y_2}^0 = \omega_0 p_{x_2}^0 + h_0.
\]

Step 7. Find the straight lines connecting \( W_0 \) with \( p_2^0 \).

Let \( W_0p_2^0 \) be straight lines, as shown in Figure 2.

The slope \( \eta_0 \) is:

\[
\eta_0 = \frac{p_{y_2}^0 - y_0}{p_{x_2}^0 - x_0}
\]
with y-intercept:

\[ \theta_0 = p_{y_2}^0 - \eta_0 p_{x_2}^0. \]

The straight line of \( W_0 p_2^0 \) is \( y = \eta_0 x + \theta_0 \).

**Step 8.** Find the inner control point \( p_1^1 \) at second subinterval.

Let \( \mathcal{Z}_1 \) is a distance between \( p_2^0 \) and \( W_1 \) defined by:

\[ \mathcal{Z}_1 = \sqrt{(y_1 - p_{y_2}^0)^2 + (x_1 - p_{x_2}^0)^2}, \]

whose slope is:

\[ \sigma_1 = \frac{y_1 - p_{y_2}^0}{x_1 - p_{x_2}^0} \]

and y-intercept:

\[ \xi_1 = y_1 - \sigma_1 x_1. \]

Let \( \mathcal{Z}_1 \) is a distance between the inner control point \( p_1^1 \) and \( W_1 \), given by:

\[ \mathcal{Z}_1 = \sqrt{(p_{y_1}^1 - y_1)^2 + (p_{x_1}^1 - x_1)^2}, \]

Since \( p_1^1 \) satisfies the equation of line \( \Omega_1 \), then \( p_{y_1}^1 = \sigma_1 p_{x_1}^1 + \xi_1 \).

Hence,

\[
\left[ (a_1)^2 + 1 \right] \left( p_{x_1}^1 \right)^2 + \left[ 2a_1 \sigma_1 - 2y_1a_1 - 2x_1 \right] p_{x_1}^1 + \left[ (y_1)^2 - 2y_1 \sigma_1 + \sigma_1^2 + (x_1)^2 - (\mathcal{Z}_1)^2 \right] = 0
\]  \hspace{1cm} (3)

using quadratic formula to find the value of \( p_{x_1}^1 \) in Equation (3). Substituting \( p_{x_1}^1 \) into (2) yields the value of \( p_{y_1}^1 \).
Step 9. Find \( y\)-intercept \( (h_1) \) of the straight line \( \omega_1 \) with slope \( \omega_1 \) as described in the following algorithms:

**Input:** \( \omega_1, \, p_1^1, \, \delta_1, \, b_1, \, \sigma_1 \).

**Output:** \( y\)-intercepts \( (h_1) \) of the straight line with slope \( \omega_1 \).

**Start**

\[
h_1 = p_{y1}^1 - \omega_1 p_{x1}^1
\]

\[
\gamma_1 = |h_1 - \delta_1|
\]

**if** \( b_1 > \sigma_1 \land b_2 > \sigma_2 \) or \( b_1 < \sigma_1 \land b_2 < \sigma_2 \) **then**

\[
h_1 < h_1
\]

**else if** \( b_1 \geq \sigma_1 \land b_2 \leq \sigma_2 \) **then** (inflection)

\[
h_1 = \delta_1 + \gamma_1
\]

**else if** \( b_1 \leq \sigma_1 \land b_2 \geq \sigma_2 \) **then** (inflection)

\[
h_1 = \delta_1 - \gamma_1
\]

**End**

**Step 10.** Find \( y\)-intercepts \( (\theta_1) \) of the straight line connecting \( W_2 \) with \( p_1^1 \).

Let \( W_2p_1^1 \) be straight lines as shown in Figure 2.

The slope \( \eta_1 \) is defined as:

\[
\eta_1 = \frac{y_2 - p_{y1}^1}{x_2 - p_{x1}^1}
\]

and \( y\)-intercept is:

\[
\theta_1 = p_{y1}^1 - \eta_1 p_{x1}^1
\]

The straight line of \( W_2p_1^1 \) is represented by \( y = \eta_1 x + \theta_1 \).
When there is an inflection of the data points polygon, the slope will be calculated by replacing $W_2$ with $\chi^1$, as given below:

Let $r$ be a perpendicular of the straight $y = \omega_1 x + h_1$, hence, the slope of $r$ is given by $-\frac{1}{\omega_1}$, while $y$-intercept is

$$b^h_1 = p^1_y + \frac{1}{\omega_1} p^1_{x_1}.$$

This leads to the interception point $\chi^1 = \left(\chi^1_x, \chi^1_y\right)$ where:

$$\chi^1_x = \frac{b^h_1 - h_1}{\omega_1 + \frac{1}{\omega_1}},$$

with $y$-intercept:

$$\chi^1_y = -\frac{1}{\omega_1} \chi^1_x + b^h_1.$$

Following algorithm demonstrates the $y$-intercept ($\vartheta_1$) construction that covers the inflection areas in a polygon dataset. Refer to Figure 3.1 for line $\vartheta_1$.

**Input:** $W_2, \omega_1, p^1_x, h_1, b_1, \sigma_1, b_2, \sigma_2$.

**Output:** $y$-intercepts ($\vartheta_1$) of the straight line with slope $\omega_1$.

**Start**

$$b^h_1 = p^1_y + \frac{1}{\omega_1} p^1_{x_1}$$

$$\chi^1_x = \frac{b^h_1 - h_1}{\omega_1 + \frac{1}{\omega_1}}$$

$$\chi^1_y = -\frac{1}{\omega_1} \chi^1_x + b^h_1.$$
\( \eta_1 = \frac{y_2 - p^1_{y_1}}{x_2 - p^1_{x_1}} \)

**else if** \( b_1 \geq \sigma_1 \) \& \( b_2 \leq \sigma_2 \) then (inflection)

\[
\eta_1 = \frac{\chi_y - p^1_{y_1}}{\chi_x - p^1_{x_1}}
\]

**end**

End

\( \vartheta_1 = p^1_{y_1} - \eta_1 p^1_{x_1}. \)

**Step 11.** Find the inner control point \( p^1_0 \) and \( p^1_2 \)

Let \( p^0_1 \), \( p^0_2 \) are intercept points of straight lines \( \vartheta_0, \vartheta_1 \) with straight the line \( W_0W_2 \) respectively,

\[
p^0_{x_1} = \frac{h_0 - \varepsilon}{a_1 - \omega_0}; \quad p^0_{y_1} = \omega_0 p^0_{x_1} + h_0. \quad (4)
\]

\[
p^0_{x_2} = \frac{h_1 - \varepsilon}{a_1 - \omega_1}; \quad p^0_{y_2} = \omega_1 p^0_{x_2} + h_1. \quad (5)
\]

In order to define the value of \( \varepsilon \), we will use the following relation:

\[
4\varSigma_1 = \sqrt{(p^1_{y_2} - p^1_{y_1})^2 + (p^1_{x_2} - p^1_{x_1})^2}. \quad (6)
\]

Substituting (4), (5) in (6) gives:

\[
16(\varSigma_1)^2 = \left( a_1 \left( \frac{h_1 - \varepsilon}{a_1 - \omega_1} \right) + h_1 \right)^2 - \\
-2 \left( \omega_1 \left( \frac{h_1 - \varepsilon}{a_1 - \omega_1} \right) + h_1 \right) \left( \omega_0 \left( \frac{h_0 - \varepsilon}{a_1 - \omega_0} \right) + h_0 \right) + \\
\left( \omega_0 \left( \frac{h_0 - \varepsilon}{a_1 - \omega_0} \right) + h_0 \right)^2 + \left( \frac{h_1 - \varepsilon}{a_1 - \omega_1} \right)^2 - \\
-2 \left( \frac{h_1 - \varepsilon}{a_1 - \omega_1} \right) \left( \frac{h_0 - \varepsilon}{a_1 - \omega_0} \right) + \left( \frac{h_0 - \varepsilon}{a_1 - \omega_0} \right)^2.
\]
which leads to:
\[
\left[ \frac{\left( \omega_1^2 + 1 \right) - 2 \left( \omega_1 \omega_0 \right) + \left( \omega_0^2 + 1 \right)}{(a_1 - \omega_1)^2} \right] \varepsilon^2 - \\
- \left[ \frac{2 h_1 (a_1 + 1)}{(a_1 - \omega_1)^2} \right] - \left[ \frac{2 a_0 h_1 a_1 + 2 a_0 h_1 a_1 + 2 (h_1 + h_0)}{(a_1 - \omega_1)(a_1 - \omega_0)} \right] - \left[ \frac{2 h_0 (a_1 \omega_0 + 1)}{(a_1 - \omega_0)^2} \right] \varepsilon + \\
+ \left[ \frac{h^2 (a_1 + 1)}{(a_1 - \omega_1)^2} - \frac{2 h_1 h_0 (a_1^2 + 1)}{(a_1 - \omega_1)(a_1 - \omega_0)} - \frac{h_0^2 (a_1^2 + 1)}{(a_1 - \omega_0)^2} - 16 \Omega^2 \right] = 0. 
\]

(7)

By solving Equation (7) using quadratic formula, we can find the value of \( \varepsilon \) which then substituted into (4), (5) to get the value of \( P_1^0 \) and \( P_2^1 \).

**Step 12.** Repeat Step 8 to find inner control point \( P_1^s \) where \( s = 2, \ldots, n - 1 \). Replacing \( \Omega \) with \( \Omega_s \), \( a \) with \( a_s \), \( x_1 \) with \( x_s \), \( P_2^0 \) with \( P_2^{s-1} \), \( P_1^l \) with \( P_1^s \), \( a_1 \) with \( \tau_s \), \( \Omega_1 \) with \( \Omega_s \), and \( \sigma_1 \) with \( \mu_s \), the inner control point \( P_1^s \) can be found, where \( \tau_s \) define as \( \tau_s = \frac{y_s - P_{y_2}^{s-1}}{x_s - P_{x_2}^{s-1}} \), \( \mu_s = y_s - \tau_s x_s \).

**Step 13.** Repeat Step 9 to find \( y \)-intercepts \( (h_s) \) of the straight line with slope \( \omega_s \).

Replacing \( \omega_1 \) with \( \omega_s \), \( h_1 \) with \( h_s \), \( P_1^l \) with \( P_1^s \), \( a_1 \) with \( \omega_s \), \( \delta_1 \) with \( \delta_s \), \( b_1 \) with \( b_s \), \( \sigma_1 \) with \( \sigma_s \), \( h_1 \) with \( h_s \), and \( \gamma_1 \) with \( \gamma_s \) to find \( h_s \).

**Step 14.** Repeat Step 10 to find \( y \)-intercepts \( (\vartheta_s) \) of the straight line connecting \( W_{s+1} \) with \( P_1^s \).

Replacing \( \eta_1 \) with \( \eta_s \), \( W_2 \) with \( W_{s+1} \), \( P_1^l \) with \( P_1^s \), \( a_1 \) with \( \omega_s \), \( h_1 \) with \( h_s \), \( b_1^h \) with \( h_s^h \), and \( \chi_1 \) with \( \chi_s \) to find \( \vartheta_s \).

**Step 15.** Find the inner control point \( P_2^s \)

The distance between \( P_2^{s-1} \) and \( P_1^s \) given as:
\[
2 \Omega_s = \sqrt{\left( P_{y_1}^s - P_{y_2}^{s-1} \right)^2 + \left( P_{x_1}^s - P_{x_2}^{s-1} \right)^2}
\]
Suppose the distance between the inner control points $P_1^{s-1}$ and $P_2^s$ is $4\mathcal{J}_s$. Then:

$$4\mathcal{J}_s = \sqrt{(p_{y_2}^s - p_{y_1}^{s-1})^2 + (p_{x_2}^s - p_{x_1}^{s-1})^2}.$$ 

Since $P_2^s$ satisfies the equation of the line $\varphi_s$, then $p_{x_2}^s = \eta_s p_{x_2}^s + \vartheta_s$, which yields:

$$\left[\eta_s^2 + 1\right](p_{x_2}^s)^2 + \left[2\eta_s \vartheta_s - 2p_{y_1}^{s-1}\eta_s - 2p_{x_1}^{s-1}\right]p_{x_2}^s + \left[(p_{y_1}^{s-1})^2 - 2p_{y_1}^{s-1}\vartheta_s + \vartheta_s^2 + (p_{x_1}^{s-1})^2 - 16\mathcal{J}_s^2\right] = 0 \quad (8)$$

The quadratic formula will be employed in Equation (8) to find the value of $p_{x_2}^s$. As well, substituting $p_{x_2}^s$ into (2) to produce $p_{y_2}^s$.

The values of inner control points are then substituted in (1) in order to obtain PCBP as below:

$$F(t) = \begin{cases} (x_0(t), y_0(t)) & x_0(t) \in [x_0, x_1], y_0(t) \in [y_0, y_1], t \in [0, 1], \\
(x_1(t), y_1(t)) & x_1(t) \in [x_1, x_2], y_1(t) \in [y_1, y_2], t \in [0, 1], \\
& \vdots \\
(x_{n-1}(t), y_{n-1}(t)) & x_{n-1}(t) \in [x_{n-1}, x_n], y_{n-1}(t) \in [y_{n-1}, y_n], t \in [0, 1]. \end{cases}$$

### 3.2. Limitations

The limitations of the proposed method are as follows:

1. The data points should be arranged ascending as $W_0 \leq W_1 \leq \ldots \leq W_n$.
2. The number of data points should be not less than three data.
3. The domain of the dataset is $\mathbb{R}$.
4. The distance between every adjacent date points should be not more or less than $(1:1.25)$, or increase at every next subinterval.

### 3.3. Proof of Smoothness

To prove that the proposed parametric interpolating polynomial C2GBP fulfills the second order of continuity $C^2$, the following conditions must be satisfied:

(a) $P_{j-1}(1) = P_j(0)$ for each $j = 1, \ldots, n-1$;

(b) $P'_{j-1}(1) = P'_j(0)$ for each $j$;

(c) $P''_{j-1}(1) = P''_j(0)$ for each $j$. 

By definition, the first condition (a) satisfied.
In order to investigate the condition (b), the values (1) and (0) are substituted in the first derivative of Equation (1) which is given by:

\[ P_j'(t) = 3(1-t)^2(p_j^i - p_0^i) + 6t(1-t)(p_j^i - p_j^i) + 3t^2(p_3^i - p_2^i). \]  

(9)

Substituting the values (1) and (0) into Equation (9) yields:

\[ P_j'(1) = P_{j+1}'(0), \]

which leads to:

\[ p_3^j - p_2^j = p_1^{j+1} - p_0^{j+1} \]

and

\[ p_1^{j+1} + p_2^j = p_0^{j+1} + p_3^j. \]

Since \( p_3^j = p_0^{j+1} = W_j \), then

\[ W_j - p_2^j = p_1^{j+1} - W_j, \]

which means the distance between \( W_j \) and \( p_2^j \) is equal to the distance between \( p_1^{j+1} \) and \( W_j \), which is already achieved in Step 7.

In order to investigate the condition \( P_j''(1) = P_{j+1}''(0) \) in (c) we need to substitute into the second derivative of Equation (1).

Since \( P_j''(t) = 6(1-t)(p_j^i - 2p_1^i + p_0^i) + 6t(p_3^i - 2p_2^i + p_1^i) \), then

\[ 2p_2^j - p_1^j + p_2^{j+1} - 2p_1^{j+1} = p_3^j - p_0^{j+1}. \]

Since \( p_3^j = p_0^{j+1} \) then:

\[ 2p_2^j - p_1^j + p_2^{j+1} - 2p_1^{j+1} = 0, \]

which leads to

\[ p_2^{j+1} - p_1^j = 2(p_1^{j+1} - p_2^j). \]
This implies the distances \((S_j)\) from \(p_2^{j+1}\) to \(p_1^j\) are two times those from \(p_1^{j+1}\) to \(p_2^j\). Furthermore, the distances between \(p_1^{j+1}\) and \(p_2^j\) are twice the distances between \(W_j\) and \(p_2^j\) (or \(p_1^{j+1}\) and \(W_j\)). Hence, the distances from \(p_2^{j+1}\) to \(p_1^j\) are \((4S_j)\), as shown in Equation (6).

4. Numerical Results. In this section, the numerical results obtained by the proposed parametric interpolating polynomial in solving test problems will be compared with the previous studies in terms of accuracy.

4.1. Test Problems. Three test problems (functions) were used to verify the accuracy of the proposed parametric polynomial interpolation. The obtained results were then compared with the Natural Cubic Spline (Spline) [27], Cubic Hermit Interpolating Polynomial (Pchip) [28], Modified Akima Piecewise Cubic Hermit Interpolation (mAkima) [29], Rational Cubic Ball Interpolation (Ball) [19], and Cubic Natural Curve (Pollock) [15], in terms of errors.

**Problem 1:**

Function: \(y = \sin x, \ x \in [-3.5, 7].\)

Data points: \(x = [-3.5, -1.5, 0.9, 4.2, 7].\)

Figure 3 shows the behaviour of the test function and its comparison with six other curves obtained by employing six different approximation methods.

![Fig. 3. Comparison between six different methods with the test function for approximating dataset in Problem 1](image-url)
Problem 2:

Function: $y = \frac{x^3 - 2}{x^3 + 1}, \ x \in [-2.52, 4.84]$

Data points: $x = [-2.52, -1.13, -0.12, 1.23, 4.84]$

Analogous to Problem 1, Figure 4 shows the behaviour of seven curves for the test function and its comparison with other approximation parametric interpolating polynomials.

![Figure 4](attachment:problem2.png)

Fig. 4. Comparison between six different methods with the test function for approximating dataset in Problem 2

Problem 3:

Function: $y = \sinh \frac{5x}{2}, \ x \in [-2.72, 6.91]$

Data points: $x = [-2.72, -1.13, 0.62, 3.73, 6.91]$

Similarly, Figure 5 illustrates the comparison of the test function with other approximate parametric polynomials.
4.2. Error Analysis. Error values can be measured by using one or more error estimating formulas obtained by calculating the distance on every test point on the test curve with the approximate curve over the whole subintervals. The errors were using 99 test points on the entire curve between every adjacent data points, i.e. the total number of test points ($\lambda$) along the entire curve is $100n+2$ where $n+1$ is the number of data points. Three types of errors were used; Sum of Squared Estimate, Mean Absolute Error, and Root Mean Square Error (RMSE). Sum of Squared Estimate (SSE) is the sum of the squared differences between each test points on the comparison curves, defined by:

$$SSE = \sum_{Y=0}^{\lambda} (y_{Y}^j - y_{Y}^e)^2,$$

where $\lambda = 100n + 2$, $n$ is the number of data points, $y_{Y}^j$ are the test points on the test curve, and $y_{Y}^e$ are the test points on the approximating curve. Meanwhile, Mean Absolute Error (MAE) calculates the average difference
between the lengths of distance between every test points on the comparison curves. The formula for MAE is:

\[
MAE = \frac{\sum_{\lambda=0}^{\lambda} |y_{\lambda}^T - y_{\lambda}^e|}{\lambda}.
\]

Finally, Root Mean Square Error (RMSE) measures the square radical of the squared differences between the gap lengths of test points on the comparison curves divided by the number of test points as given in the following formula:

\[
RMSE = \sqrt{\frac{\sum_{\lambda=0}^{\lambda} (y_{\lambda}^T - y_{\lambda}^e)^2}{\lambda}}.
\]

4.3. Results and Discussion. The numerical results show the comparison between the six methods in terms of accuracy. In general, the proposed parametric interpolating polynomial performs better than the other existing interpolating polynomial considered in this study. Figure 3 demonstrates C2GBP is capable of preserving the curvature compared with the other five previous methods in Problem 1. An irregular inflexion curve was detected in Problem 2. The numerical results indicate that C2GBP manages to handle this situation better than the other methods by producing the smallest errors as displayed in Figure 4. Figure 5 presents the numerical results obtained in the employed method for solving the increase steep of the curve occurred in Problem 3. The results show that C2GBP also excels in non-oscillating curve output since the inner control points are geometrically constructed. The advantage of C2GBP is that it is able to control the curvature at subintervals which increases accuracy.

The results of the approximate parametric interpolating polynomials for solving Problems 1-3 in terms of errors are also displayed in Tables 1 to 3, respectively.

Table 1. Comparison of the new method with the existing methods in terms accuracy for solving Problem 1

| No. | Error Method | Spline | Pchip | mAkima | Ball   | Pollock | C2GBP   |
|-----|--------------|--------|-------|--------|--------|---------|---------|
| 1   | SSE          | 30.1732| 22.3251| 21.4038| 16.4903| 18.8969 | 0.9798  |
| 2   | MAE          | 0.1438 | 0.0500 | 0.0220 | 0.0753 | 0.0430  | 0.0030  |
| 3   | RMSE         | 4.5673 | 3.2846 | 3.4321 | 2.8704 | 3.3622  | 0.7389  |
for solving Problem 2

| No. | Error Method | Spline | Pchip | mAkima | Ball   | Pollock | C2GBP   |
|-----|--------------|--------|-------|--------|--------|---------|---------|
| 1   | SSE          | 400.5244 | 13.3164 | 9.2046 | 20.3031 | 13.5424 | 2.8842  |
| 2   | MAE          | 0.5216  | 0.1207 | 0.1030 | 0.1377 | 0.1283 | 0.0688  |
| 3   | RMSE         | 10.4459 | 2.4161 | 2.0618 | 2.7570 | 2.5692 | 1.3780  |

Table 3. Comparison of the new method with the existing methods in terms accuracy for solving Problem 3

| No. | Error Method | Spline | Pchip | mAkima | Ball   | Pollock | C2GBP   |
|-----|--------------|--------|-------|--------|--------|---------|---------|
| 1   | SSE          | 1.9909×10^{+15} | 2.3463×10^{+15} | 2.6544×10^{+15} | 2.5843×10^{+15} | 3.2945×10^{+15} | 7.1770×10^{+14} |
| 2   | MAE          | 1.0663×10^{+6}  | 9.8778×10^{+5}  | 1.0557×10^{+6}  | 1.2654×10^{+6}  | 1.4839×10^{+6}  | 4.9519×10^{+5}  |
| 3   | RMSE         | 2.1352×10^{+7}  | 1.9780×10^{+7}  | 2.1139×10^{+7}  | 2.5340×10^{+7}  | 2.9716×10^{+7}  | 9.9161×10^{+6}  |

The errors in terms of SSE, MAE, RMSE in Tables 1 to 3 suggest that C2GBP is the best option to be applied to approximate dataset in all test problems.

Figures 6-8 illustrate the error rates on all $\lambda$ along approximate curves by using the RMSE, in order to provide a more accurate description. Curvature amplitude reveals that the error ratio of the curves increases which means the closer the curve to $x$-coordinate, the less the error is. It is worth to mention that the error at the dataset is zero since the interpolating points are the dataset.
Fig. 6. Comparison between six different methods in terms of error ratio using RMSE for approximating dataset in solving Problem 1

Fig. 7. Comparison between six different methods in terms of error ratio using RMSE for approximating dataset in solving Problem 2
The Bar Graph representation of Tables 1 to 3 are shown in Figures 9-11, respectively.
5. Conclusion. This study has successfully constructed a piecewise cubic Bézier polynomial using a geometric technique to find suitable Bézier
inner control point locations for each sub-interval. The proposed method
gives an approximate cubic Bézier curve representing the dataset with inter-
polating at all data points. The proposed procedure succeeded in achieving
the second-order of parametric continuity between every adjacent sub-
interval of the data points. The newly constructed parametric interpolating
polynomial was then compared with the existing natural cubic spline,
piecewise cubic Hermite interpolating polynomial, modified Akima piece-
wise cubic Hermite interpolation, rational cubic Ball interpolation, and nat-
ural cubic Bézier curve using the same datasets. Three different error testing
methods have been used by taking (100) test points for each sub-interval.
The numerical results show that the proposed method is more accurate than
the other existing methods shown in this study. All the details of the com-
parison have been indicated in tables and graphs for each testing. The re-
sulting curve is very appropriate to find a fit, smooth, and accurate represent-
ation of the data points. The proposed method can also be used in many
applications, as in image processing and geographic information systems.
As well, it is expandable to include many applications in two-dimension.

References
1. Alawadi F. New Pattern Recognition Methods for Identifying Oil Spills from Satellite
Remote Sensing Data. Image and Signal Processing for Remote Sensing XV. 2009.
vol. 7477. pp. 74770X.
2. Amenta N., Bern M. Surface Reconstruction by Voronoi Filtering. Discrete &
Computational Geometry. 1999. vol. 22. pp. 481–504.
3. Kondrashov D., Ghil M. Spatio-Temporal Filling of Missing Points in Geophysical
Data Sets. Nonlinear Processes in Geophysics. 2006. vol. 13(2). pp. 151–159.
4. Chen F., Lou W. Degree Reduction of Interval Bézier Curves. CAD Computer Aided
Design. 2000. vol. 32(10). pp. 571–582.
5. Wu Q.B., Xia F.H. Shape modification of Bézier curves by constrained optimization.
Journal of Zhejiang University: Science. 2005. vol. 6. pp. 124–127.
6. Al-Shemary M.A.F. Interpolation by Using Bézier Curve Numerically with Image
Processing Applications. Journal of Al-Qadisiyah for Computer Science and
Mathematics. 2011. vol. 3(2). pp. 400–409.
7. Sederberg T.W., Farouki R.T. Approximation by Interval Bézier Curves. IEEE
Computer Graphics and Applications. 1992. vol. 12. pp. 87–59.
8. Hwang J.H., Arkin R.C., Kwon D.S. Mobile Robots at your Fingertip: Bezier Curve
on-line Trajectory Generation for Supervisory Control. 2003 EEE/RSJ International
Conference on Intelligent Robots and Systems. 2003. vol. 2. pp. 1444–1449.
9. Škrjanc I., Klančar G. Cooperative Collision Avoidance Between Multiple Robots
Based on Bernstein-Bézier Curves. International Conference on Information
Technology Interfaces. 2007. pp. 34–43.
10. Ho M.L., Chan P.T., Rad A.B. Lane Change Algorithm for Autonomous Vehicles via
Virtual Curvature Method. Journal of Advanced Transportation. 2009. vol. 43(1). pp. 47–70.
11. Korzeniowski D., Ślaski G. Method of Planning a Reference Trajectory of a Single
Lane Change Maneuver with Bézier Curve. IOP Conference Series: Materials
Science and Engineering. 2016. vol. 148(1). 243 p.
12. Lattarulo R. et al. Urban Motion Planning Framework Based on N-Bézier Curves
Considering Comfort and Safety. Journal of Advanced Transportation. 2018. 29 p.
13. Perez J., Godoy J., Villagra J., Onieva E. Trajectory Generator for Autonomous Vehicles in Urban Environments. 2013 IEEE International Conference on Robotics and Automation. 2013. pp. 409–414.

14. Ge Q.J., Kang D. Motion Interpolation With G2 Composite Bezier Motions. 1995. pp. 520–525.

15. Pollock D.S.G. Signal Processing and its Applications. Handbook of time series analysis, signal processing, and dynamics. 1999. 543 p.

16. Shemanarev M. A Very Simple Method of Smoothing. Retrieved from The Anti-Grain Geometry 2002. Available at: http://www.antigrain.com/agg_research/bezier_interpolation.html (accessed: 12.12.2020).

17. Yau H.T., Wang J.B. Fast Bezier Interpolator with Real-Time Lookahead Function for High-Accuracy Machining. International Journal of Machine Tools and Manufacture. 2007. vol. 47(10). pp. 1518–1529.

18. Saaban A., Zainudin L., Bakar M.N.A. On Piecewise Interpolation Techniques for Estimating Solar Radiation Missing Values in Kedah. AIP Conference Proceedings, 1635 (Icoqsia). 2014. pp. 217–221.

19. Karim S.A.A. Shape Preserving by Using Rational Cubic Ball Interpolant. Far East Journal of Mathematical Sciences. 2015. vol. 96(2). pp. 211–230.

20. Saaban A., Zainudin M.L., Abu Bakar M.N. Piecewise Positivity Preserving Cubic Bezier Interpolation for Estimating Solar Radiation Missing Value in Penang, Malaysia. Journal of Mathematics and Statistics. 2016. vol. 12(4). pp. 302–307.

21. Ueda E.K. et al. Piecewise Bézier Curve Fitting by Multiobjective Simulated Annealing. IFAC-PapersOnLine. 2016. vol. 49(31). pp. 49–54.

22. Stelia O., Potapenko L., Sirenko I. Application of Piecewise-Cubic Functions for Constructing a Bezier Type Curve of C1 Smoothness. Eastern-European Journal of Enterprise Technologies. 2018. vol. 4(2). pp. 46–52.

23. Zulkifli N.A.B. et al. Image Interpolation Using a Rational Bi-Cubic Ball. Mathematics. 2019. vol. 7(11). 29 p.

24. Elber G. Interpolation Using Bézier Curves. Graphics Gems III (IBM Version). 1992. vol. 0. pp. 133–136.

25. Quarteroni A., Sacco R., Saleri F. Numerical Mathematics. Applied Mathematics. 2000. vol. 37. 57 p.

26. Hansford D. Bézier Techniques. Handbook of Computer Aided Geometric Design 2002. pp. 75–109.

27. Burden R.L., Faires J.D. Numerical Analysis (Tenth Edit). Brooks/Cole. 2015.

28. Rabbath C.A., Corriveau D. A comparison of piecewise cubic Hermite interpolating polynomials, cubic splines and piecewise linear functions for the approximation of projectile aerodynamics. Defence Technology. 2019. vol. 15(5). pp. 741–757.

29. Moler C. Makima Piecewise Cubic Interpolation. MathWorks. 2019.

Fadhel Mustafa – Senior Lecturer, Department of Applied Mathematics, Al-Muthanna University. Research interests: numerical analysis, bezier interpolation, numerical approximation, numerical interpolation, image upscaling, image interpolation, computer aided geometric design. The number of publications – 8. mustafa@mu.edu.iq; 66001, Samawa, Iraq; office phone: +9647800226700.

Omar Zurni – Ph.D., Dr.Sci., Professor, Universiti Utara Malaysia. Research interests: mathematical modelling, algorithms, numerics, differentiation, numerical methods, parallel computing, calculations, MPI, ordinary equations, parallel algorithm. The number of publications – 200. zurni@uum.edu.my; 06010, Sintok, Malaysia; office phone: +60194443993.

Acknowledgements. I would like to thank Universiti Utara Malaysia for supporting this research.
Фадхель М.А., Омар З.Б. Геометрический кусочно-кубический интерполяционный многочлен Безье с непрерывностью $C^2$.

Аннотация. Кривая Безье – это параметрический полином, который применяется для получения хороших методов кусочной интерполяции с большим преимуществом перед другими кусочными полиномами. Следовательно, критически важно построить кривые Безье, которые были бы гладкими и могли бы повысить точность решений. Большинство известных стратегий определения внутренних контрольных точек для кусочных кривых Безье обеспечивают только частичную гладкость, удовлетворяющую первому порядку непрерывности. Некоторые решения позволяют строить интерполяционные полиномы с гладкостью по ширине вдоль аппроксимируемой кривой. Однако они все еще не могут обрабатывать расположение внутренних контрольных точек. Частичная гладкость и неконтролируемое расположение внутренних контрольных точек могут повлиять на точность приближительной кривой набора данных. Чтобы улучшить гладкость и точность предыдущих стратегий, предлагается новый кусочно-кубический многочлен Безье второго порядка непрерывности $C^2$ для оценки пропущенных значений. Предлагаемый метод использует геометрическое построение для поиска внутренних контрольных точек для каждого смежного подынтервала указанного набора данных. Не только предлагаемый метод сохраняет стабильность и гладкость, анализ ошибок численных результатов также показывает, что результатирующий интерполирующий полином более точен, чем те, которые получены с помощью существующих методов.

Ключевые слова: Полином интерполяции, кривая Безье, сплайн Безье, SSE, MAE, RMSE

Литература
1. Alawadi F. New Pattern Recognition Methods for Identifying Oil Spills from Satellite Remote Sensing Data // Image and Signal Processing for Remote Sensing XV. 2009. vol. 7477. pp. 74770X.
2. Amenta N., Bern M. Surface Reconstruction by Voronoi Filtering // Discrete & Computational Geometry. 1999. vol. 22. pp. 481–504.
3. Kondrashov D., Ghil M. Spatio-Temporal Filling of Missing Points in Geophysical Data Sets // Nonlinear Processes in Geophysics. 2006. vol. 13(2). pp. 151–159.
4. Chen F., Lou W. Degree Reduction of Interval Bézier Curves // CAD Computer Aided Design. 2000. vol. 32(10). pp. 571–582.
5. Wu Q.B., Xia F.H. Shape modification of Bézier curves by constrained optimization // Journal of Zhejiang University: Science. 2005. vol. 6. pp. 124–127.
6. Al-Shemary M.A.F. Interpolation by Using Bézier Curve Numerically with Image Processing Applications // Journal of Al-Qadisiyah for Computer Science and Mathematics. 2011. vol. 3(2). pp. 400–409.
7. Sederberg T.W., Farouki R.T. Approximation by Interval Bézier Curves // IEEE Computer Graphics and Applications. 1992. vol. 12. pp. 87–59.
8. Hwang J.H., Arkin R.C., Kwon D.S. Mobile Robots at your Fingertip: Bezier Curve on-line Trajectory Generation for Supervisory Control // 2003 EEE/RSJ International Conference on Intelligent Robots and Systems. 2003. vol. 2. pp. 1444–1449.

9. Škrjanc I., Klančar G. Cooperative Collision Avoidance Between Multiple Robots Based on Bernstein-Bézier Curves // International Conference on Information Technology Interfaces. 2007. pp. 34–43.

10. Ho M.L., Chan P.T., Rad A.B. Lane Change Algorithm for Autonomous Vehicles via Virtual Curvature Method // Journal of Advanced Transportation. 2009. vol. 43(1). pp. 47–70.

11. Korzeniowski D., Ślaski G. Method of Planning a Reference Trajectory of a Single Lane Change Manoeuver with Bezier Curve // IOP Conference Series: Materials Science and Engineering. 2016. vol. 148(1). 243 p.

12. Lattarulo R. et al. Urban Motion Planning Framework Based on N-Bézier Curves Considering Comfort and Safety // Journal of Advanced Transportation. 2018. 29 p.

13. Perez J., Godoy J., Villagra J., Onieva E. Trajectory Generator for Autonomous Vehicles in Urban Environments // 2013 IEEE International Conference on Robotics and Automation. 2013. pp. 409–414.

14. Ge Q.J., Kang D. Motion Interpolation With G2 Composite Bezier Motions. 1995. pp. 520–525.

15. Pollock D.S.G. Signal Processing and its Applications // Handbook of time series analysis, signal processing, and dynamics. 1999. 543 p.

16. Shemanarev M. A Very Simple Method of Smoothing. Retrieved from The Anti-Grain Geometry 2002. URL: http://www.antigrain.com/agg_research/bezier_interpolation.html (дата обращения: 12.12.2020).

17. Yau H.T., Wang J.B. Fast Bezier Interpolator with Real-Time Lookahead Function for High-Accuracy Machining // International Journal of Machine Tools and Manufacture. 2007. vol. 47(10). pp. 1518–1529.

18. Saaban A., Zainudin L., Bakar M.N.A. On Piecewise Interpolation Techniques for Estimating Solar Radiation Missing Values in Kedah // AIP Conference Proceedings, 1635 (Icoqsia). 2014. pp. 217–221.

19. Karim S.A.A. Shape Preserving by Using Rational Cubic Ball Interpolant // Far East Journal of Mathematical Sciences. 2015. vol. 96(2). pp. 211–230.

20. Saaban A., Zainudin M.L., Abu Bakar M.N. Piecewise Positivity Preserving Cubic Bezier Interpolation for Estimating Solar Radiation Missing Value in Penang, Malaysia // Journal of Mathematics and Statistics. 2016. vol. 12(4). pp. 302–307.

21. Ueda E.K. et al. Piecewise Bézier Curve Fitting by Multiobjective Simulated Annealing // IFAC-PapersOnLine. 2016. vol. 49(31). pp. 49–54.

22. Stelia O., Potapenko L., Sirenko I. Application of Piecewise-Cubic Functions for Constructing a Bezier Type Curve of C1 Smoothness // Eastern-European Journal of Enterprise Technologies. 2018. vol. 4(2). pp. 46–52.

23. Zulkifli N.A.B. et al. Image Interpolation Using a Rational Bi-Cubic Ball // Mathematics. 2019. vol. 7(11). 29 p.

24. Elber G. Interpolation Using Bézier Curves // Graphics Gems III (IBM Version). 1992. vol. 0. pp. 133–136.

25. Quarteroni A., Sacco R., Saleri F. Numerical Mathematics // Applied Mathematics 2000. vol. 37. 57 p.
26. *Hansford D.* Bézier Techniques // Handbook of Computer Aided Geometric Design 2002. pp. 75–109.
27. *Burden R.L., Faires J.D.* Numerical Analysis (Tenth Edit) // Brooks/Cole. 2015.
28. *Rabbath C.A., Corriveau D.* A comparison of piecewise cubic Hermite interpolating polynomials, cubic splines and piecewise linear functions for the approximation of projectile aerodynamics // Defence Technology. 2019. vol. 15(5). pp. 741–757.
29. *Moler C.* Makima Piecewise Cubic Interpolation // MathWorks. 2019.

Фадхель Мустафа Аббас – старший преподаватель, кафедра прикладной информатики, Университет Аль-Мутанна. Область научных интересов: численный анализ, интерполяция Безье, численное приближение, численная интерполяция, масштабирование изображения, интерполяция изображений, компьютерное геометрическое проектирование. Число научных публикаций – 8. mustafa@mu.edu.iq; 66001, Самава, Ирак; р.т.: +9647800226700.

Омар Зурни Б – д-р филос. наук, профессор, Малазийский университет Утара. Область научных интересов: математическое моделирование, алгоритмы, вычисления, дифференцирование, численные методы, параллельные вычисления, обыкновенные уравнения, параллельный алгоритм. Число научных публикаций – 200. zurni@uum.edu.my; 06010, Синток, Малайзия; р.т.: +60194443993.

Поддержка исследований. Проект осуществлен при поддержке Малазийского университета Утара.