

## Research Article

# Modified Bézier Curves with Shape-Preserving Characteristics Using Differential Evolution Optimization Algorithm

**Mohammad Asif Zaman and Shuvro Chowdhury**

*Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka, 1000, Bangladesh*

Correspondence should be addressed to Mohammad Asif Zaman; [asifzaman13@gmail.com](mailto:asifzaman13@gmail.com)

Received 27 October 2012; Accepted 6 February 2013

Academic Editor: William J. Layton

Copyright © 2013 M. A. Zaman and S. Chowdhury. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A parametric equation for a modified Bézier curve is proposed for curve fitting applications. The proposed equation contains shaping parameters to adjust the shape of the fitted curve. This flexibility of shape control is expected to produce a curve which is capable of following any sets of discrete data points. A Differential Evolution (DE) optimization based technique is proposed to find the optimum value of these shaping parameters. The optimality of the fitted curve is defined in terms of some proposed cost parameters. These parameters are defined based on sum of squares errors. Numerical results are presented highlighting the effectiveness of the proposed curves compared with conventional Bézier curves. From the obtained results, it is observed that the proposed method produces a curve that fits the data points more accurately.

## 1. Introduction

Bézier curve is a curve fitting tool for constructing free-form smooth parametric curves. Bézier curves are widely used in computer aided geometry design, data structure modelling, mesh generating techniques, and computer graphics applications [1–3]. These curves are also used in different fields of mechanical and electrical engineering for modelling complex surface geometries [1]. Because of the wide field of applications, efficient techniques for improving and controlling the shape of Bézier curves have become an important field of research [4, 5].

A standard curve fitting problem is defined by a set of raw data points, referred to as *control points*. The polygon obtained by connecting all the control points is termed as the *control polygon*. For most applications, it is required to find a free-form curve that most closely follows the control polygon. Conventional least-squares curve fitting techniques can fit a mathematical equation through the control points. However, this technique is not always applicable as the control points do not necessarily follow any standard mathematical equation models. Spline interpolation results in a continuous curve

that matches the control polygon to a high degree [1], but the curve is expressed in terms of piece-wise defined functions. In many cases a single equation for the whole curve is required for mathematical operations. In such cases spline interpolations are not useful. In case of an interpolating polynomial, the interpolated curve goes through all the control points; however, the interpolated region between two control points often deviates significantly from the control polygon. This problem arises because the interpolating polynomial only tries to match the control points and does not take into account the slope variation of the control polygon. As a result, the shape of the curve between the control points is not always accurate. A Bézier curve, unlike a polynomial interpolated curve, does not necessarily go through all the control points. It goes through the two terminal control points. The number and orientation of the intermediate control points govern the shape of the Bézier curve [6]. Because of the Bernstein basis used by the Bézier curve, it creates a smooth shape and in most cases follow the general trend of the control polygon and its shape [7, 8]. This fact is illustrated in Figure 1. The data points used for the test function,  $f_1$ , are tabulated in Table 1. The figure shows that interpolating polynomial shows highly

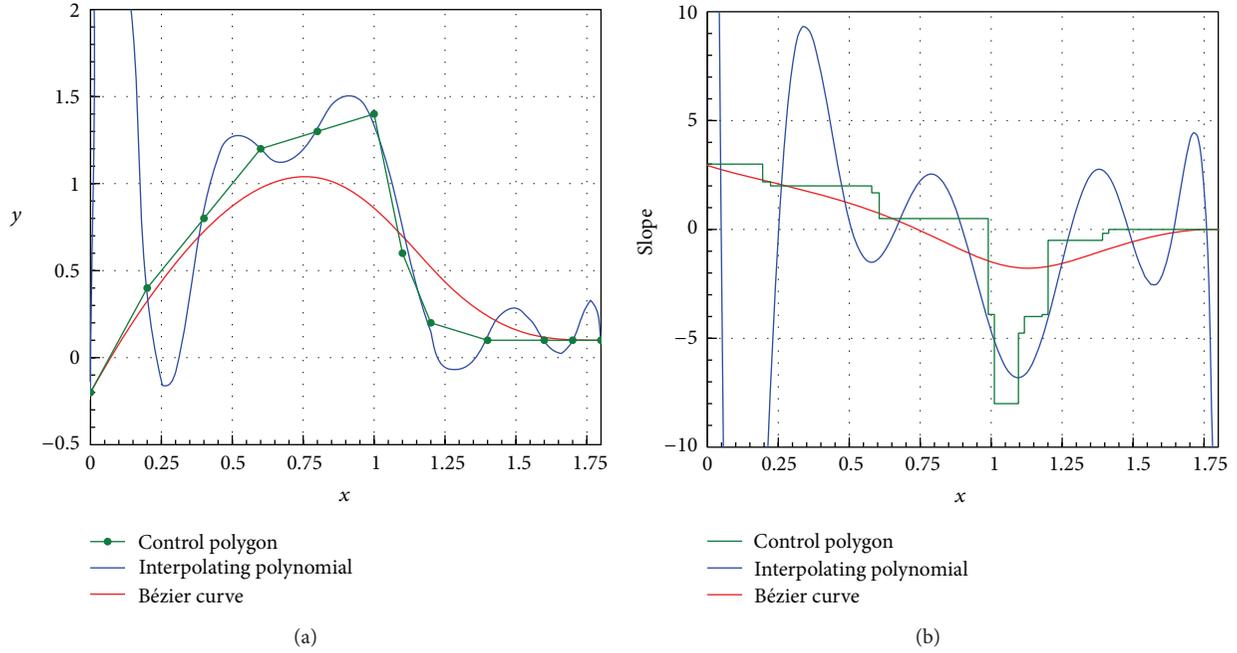


FIGURE 1: Comparison of interpolated curve and Bézier curve for test function  $f_1$ : (a) fitted curve, (b) slope of the curve.

TABLE 1: Table for test function,  $f_1$ .

$i$	$x_i$	$y_i$	$y_{bi}$	$y_{mi}$	$\zeta_i$	Performance
0	0	-0.2	-0.2000	-0.2207	1.1034	
1	0.2	0.4	0.3262	0.4005	0.8550	
2	0.4	0.8	0.7225	0.7892	2.4631	
3	0.6	1.2	0.9747	1.1809	-1.1786	$\varepsilon_{bc} = 1.5609$
4	0.8	1.3	1.0345	1.4154	2.6882	$\varepsilon_{mc} = 0.2710$
5	1.0	1.4	0.8562	1.0731	2.9621	$\varepsilon_{bs} = 117.7463$
6	1.1	0.6	0.6948	0.7126	-2.7567	$\varepsilon_{ms} = 77.0282$
7	1.2	0.2	0.5154	0.3577	-2.8568	$f_{cost,b} = 13.3355$
8	1.4	0.1	0.2373	0.0419	2.6472	$f_{cost,m} = 7.9738$
9	1.6	0.1	0.1174	0.0998	1.9715	
10	1.7	0.1	0.1016	0.1159	0.9570	
11	1.8	0.1	0.1000	0.0997	0.9974	

oscillatory nature and fails to follow the shape. However, the Bézier curve follows the shape of the curve and approximates the slope of the curve reasonably accurately throughout the region of interest.

Although Bézier curves usually follow the general trend of the control polygon, it often underestimates the slope of the control polygon. Due to inherit smoothing characteristics, it fails to follow control polygons with sharply varying sides. Also, Bézier curves lack flexibility of controlling the shape of the curve. It is not possible to generate multiple Bézier curves with different shapes for the same control polygon. This flexibility is often desired to fine-tune and optimize a curve-fitting problem. For this reason, modified versions of Bézier curves with shaping parameters have been developed [9, 10].

The methods proposed in [4, 5] give satisfactory results, but require complex computations. A simpler, yet effective approach is provided in [11], but the method does not describe the process of finding the optimal shaping parameters for a given control polygon. Some work has also concentrated on modifying the basis of the curves [12, 13]. Although effective, these methods are also based on complex computations which are difficult to implement through computer coding. In this paper, parametric equations of a modified Bézier curve is presented. The proposed equations contain shaping parameters that can be used to accurately fit a given set of control points. The optimum value of the shaping parameters is calculated using Differential Evolution (DE) optimization algorithm.

DE algorithm was first developed by Storn and Price in 1997 [14]. Since then, it has been successfully used in many applications [15]. However, only a few work has investigated the use of DE algorithm for Bézier curves [16, 17]. The work of [16] concentrated on creating new set of control points, rather than changing the equation of the Bézier curve. As a result, it inherits the same weakness of conventional Bézier curves when it comes to sharply varying control points. The work of [17] does not directly relate to curve fitting problems. In this paper, the DE algorithm is used to find the shaping parameters of a proposed modified Bézier curve parametric equation. The paper presents a generalized method that can be used for curve fitting applications for any arbitrary set of control points.

The paper is organized as follows. Section 2 contains formulation of *modified Bézier curve* and definition of *optimality*. Section 3 describes the method of optimizing the shaping parameters using DE algorithm. Section 4 presents

the numerical simulation results. Concluding remarks are made in Section 5.

## 2. Formulation of Modified Bézier Curve and Defining Optimality

For a set of  $(n+1)$  control points  $(x_i, y_i)$ , where,  $i = 0, 1, \dots, n$ , the parametric equations of the Bézier curve are given by [1]:

$$\begin{aligned} x(u) &= \sum_{i=0}^n \binom{n}{i} (1-u)^{n-i} u^i x_i, \\ y(u) &= \sum_{i=0}^n \binom{n}{i} (1-u)^{n-i} u^i y_i, \end{aligned} \quad (1)$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}. \quad (2)$$

The parameter  $u \in [0, 1]$  takes on continuous values. As  $u$  sweeps from 0 to 1, the parametric equations defined in (1) traces a continuous curve from  $(x_0, y_0)$  to  $(x_n, y_n)$ . The intermediate control points determine the shape of the curve. For a given set of control points, the shape of the Bézier curve cannot be modified. To remove this limitation, additional *shaping parameters* must be introduced in the parametric equations.

To incorporate additional parameters in the Bézier curve equation, a few constraints must be presatisfied. It is required that the modified Bézier curve spans the same region as the conventional Bézier curve, which is limited by the minimum and maximum value of the  $x$  coordinates of the control points. The shaping parameters should not affect the  $x$  values of the curve. Therefore, the shaping parameters should appear on the parametric equation of  $y$  only. The shaping parameters should provide the flexibility of both increasing or decreasing the  $y$  coordinate of any curve point. Taking these requirements into consideration, the proposed modified Bézier curve equations are selected as:

$$\begin{aligned} x(u) &= \sum_{i=0}^n \binom{n}{i} (1-u)^{n-i} u^i x_i, \\ y(u) &= \sum_{i=0}^n \Lambda_i (1-u)^{n-i} u^i y_i. \end{aligned} \quad (3)$$

Here,  $\Lambda_i$ 's are termed as the *shaping coefficients*, which appear in the place of binomial coefficients. As proposed modified Bézier curve is expected to produce a curve that *modifies* the shape of the conventional curve, the shaping coefficients can be considered as modified versions of the binomial coefficients. The shaping coefficient can therefore be

expressed as

$$\Lambda_i = \zeta_i \binom{n}{i}, \quad i = 0, 1, 2, \dots, n. \quad (4)$$

Here,  $\zeta_i$ 's are defined as the *shaping parameters*. By finding the optimum values of  $\zeta_i$ , a desired curve that follows the control polygon can be generated.

After defining the modified Bézier curve equations, it becomes necessary to mathematically define the *optimum curve* for a given control polygon. A numerical quantity must be assigned to all possible curves obtained from (3) for different sets of shaping parameter values that define how well the curve matches the control polygon. The obvious choice for such a case is the sum of squares error between the control polygon and the modified Bézier curve. To find this error, it is first noted that for a curve fitting problem solved by computer coding, a discrete set of values of the continuous parameter  $u$  must be selected.  $(N+1)$  values of  $u$ , uniformly selected in the range 0 to 1 as  $u_k$ ,  $k = 0, 1, 2, \dots, N$ , will result in  $(N+1)$  values of  $x$  and  $y$  obtained from (3). The following notations are used to express these points on the curve:

$$x_{mk} = x(u_k), \quad y_{mk} = y(u_k), \quad k = 0, 1, 2, \dots, N. \quad (5)$$

The corresponding point, the control polygon,  $(x_{ck}, y_{ck})$ , can be obtained from the linear interpolation of the control points as

$$y_{ck} = y_p + (x_{ck} - x_p) \frac{y_q - y_p}{x_q - x_p}. \quad (6)$$

Here,  $(x_p, y_p)$  and  $(x_q, y_q)$  are two closest control points that bound  $(x_{ck}, y_{ck})$  such that  $x_p < x_{ck} < x_q$ , and no other control points exist within this range. Now, the sum of squares error between the control polygon and the modified Bézier curve,  $\varepsilon_{mc}$ , is given by

$$\varepsilon_{mc} = \sum_{k=0}^N (y_{mk} - y_{ck})^2. \quad (7)$$

It is noted that  $y_{mk}$  is a function of  $\zeta_i$ . The optimal curve can be defined as the curve that generated by a specific set of shaping parameters that minimizes (7). However, this definition of optimality does not ensure a curve that follows the *shape* of the control polygon. The error defined in (7) becomes zero for a polynomial interpolated curve, which, as shown before, does not necessarily guarantee the consistency in shape. To ensure that the optimal curve follows the shape, the *slope* of the control polygon must be incorporated in the definition of optimality.

The slope of the modified curve can be calculated from using the chain rule:

$$\left(\frac{dy}{dx}\right)_m = \left(\frac{dy}{du}\right)_m \times \left(\frac{du}{dx}\right)_m. \quad (8)$$

Completing the differentiations using (3), the slope of the curve at a discrete point corresponding to  $u_k$  is given by

$$\left(\frac{dy}{dx}\right)_{mk} = \frac{\sum_{i=0}^n \Lambda_i (i - nu_k) (1 - u_k)^{n-i-1} u_k^{i-1} y_i}{\sum_{j=0}^n \binom{n}{j} (j - nu_k) (1 - u_k)^{n-j-1} u_k^{j-1} x_j}. \quad (9)$$

The slope of the control polygon at a point  $x_k$  can be approximated by the forward difference formula:

$$\left(\frac{dy}{dx}\right)_{ck} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}. \quad (10)$$

So, the sum of squares error in estimating the slope of the control polygon,  $\varepsilon_{ms}$ , is given by

$$\varepsilon_{ms} = \sum_{k=0}^N \left\{ \left(\frac{dy}{dx}\right)_{mk} - \left(\frac{dy}{dx}\right)_{ck} \right\}^2. \quad (11)$$

The optimal curve should have a low value of  $\varepsilon_{ms}$  as well as a low value of  $\varepsilon_{cs}$ .

To define the optimality of the curve, a *cost function*,  $f_{\text{cost}}$ , is defined in the following way:

$$f_{\text{cost}}(\zeta_0, \zeta_1, \dots, \zeta_n) = w_1 \varepsilon_{mc} + w_2 \varepsilon_{ms}. \quad (12)$$

Here,  $w_1$  and  $w_2$  are weight factors. It is noted that the cost function depends on the values of the shaping parameters. The weighted sum definition of the cost function ensures that both errors are incorporated. The curve created by a set of shaping parameter values that results in the lowest value of the cost function is defined as the optimum curve. The optimum set of values of the shaping parameter can be determined using DE optimization algorithm.

Although this paper concentrates on planar Bézier curves, the proposed method can be extended for general Bézier curves. The planar curves are defined by (3). In case of generalized curves, there will be additional parametric equations corresponding to each dimension. A different set of shaping parameter must be incorporated for each dimension. The error for each dimension can be calculated separately and then added together to give a single cost function. The optimum values of all the shaping parameters can be determined using the DE algorithm by minimizing cost function.

### 3. Optimizing the Shaping Parameters Using DE Algorithm

DE is a population based heuristic evolutionary optimization algorithm [14]. The algorithm represents the potential solutions in terms of  $D$  dimensional vectors defined in a solution space. The number of unknowns is equal to the dimension. The algorithm works with a predefined number of potential solution vectors simultaneously. The number

of these solution vectors is called *population size*. These population of vectors are modified in each iterative step to form a new *generation*. Throughout successive generations, the vectors converge to the optimum solution.

For the curve fitting problem discussed in this paper, the solution vectors represent a set of values of the shaping parameters,  $\zeta_i$ . Since there are  $n$  unknowns, they can be expressed by  $n$  dimensional vectors  $\mathbf{Z}$ . The  $r$ th vector in the  $G$ th generation can be represented as

$$\mathbf{Z}_{r,G} = [\zeta_{G,1} \ \zeta_{G,2} \ \dots \ \zeta_{G,n}]. \quad (13)$$

Here, the subscript  $G$  denotes the generation number. If  $N_p$  is the population size, then the  $G$ th generation is composed of  $N_p$  number of such vectors,  $\{\mathbf{Z}_{1,G}, \mathbf{Z}_{2,G}, \dots, \mathbf{Z}_{N_p,G}\}$ .

The solution space must be limited by defining possible range of values of  $\mathbf{Z}$  vectors in each dimension. This range is selected based on the optimization problem and trial and error. In this paper, the limiting values in each dimension are selected as  $-3$  to  $3$ . This implies that the shaping parameters can have values ranging from  $-3$  to  $3$ . The initial solution vectors are randomly selected from a uniform distribution [14]. The successive generations are formulated using standard *mutation*, *crossover*, and *selection* operation [14, 15]. As the DE algorithm is well covered in the literature, these processes are not described here. The selection operation of DE algorithm is driven by the cost function which relates the DE algorithm with the physical problem. The DE algorithm creates successive generations with new potential solution vectors that have lower cost values. Thus, over successive generations, the DE algorithm converges to the optimum solution. The optimum solution gives the desired shaping parameters that produce the lowest value defined by (12).

### 4. Numerical Simulation and Results

The proposed method is implemented and tested using computer coding. For simulation, the weight factors of (12) are selected as  $w_1 = 1$  and  $w_2 = 0.1$ . The allowed variation range of  $\zeta_i$ , for all  $i$ , is selected as  $-3$  to  $3$ . The values of these parameters are selected from trial and error methods. 40 discrete values of the parameter  $u$  are selected uniformly over the range 0 to 1 for simulation. A classical DE algorithm with population size 20, mutation scale factor = 0.9, and crossover coefficient = 0.9 is selected [15]. Maximum generation number is limited to 160.

For testing purposes, the proposed modified Bézier curve is used to fit four test functions. The performance of the modified curve is measured in terms of sum of squares error between the curve and control polygon,  $\varepsilon_{mc}$ ; sum of squares error between the slope of the curve and the slope of the control polygon,  $\varepsilon_{ms}$ ; and cost function,  $f_{\text{cost},m}$ . Conventional Bézier curves are also used for fitting the same test functions. The performance of the conventional curve is measured in terms of the same quantities which are represented as  $\varepsilon_{bc}$ ,  $\varepsilon_{bs}$ , and  $f_{\text{cost},b}$ . The test functions and corresponding results are

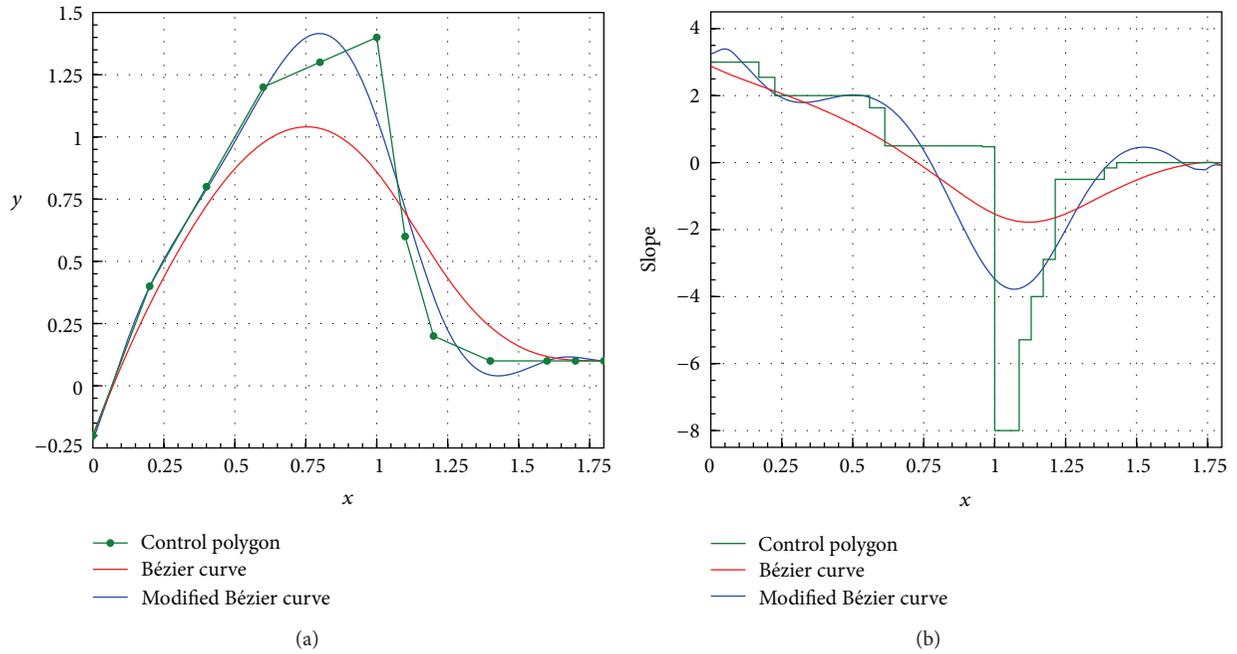


FIGURE 2: Comparison of modified Bézier curve and conventional Bézier curve for test function  $f_1$ : (a) fitted curve, (b) slope of the curve.

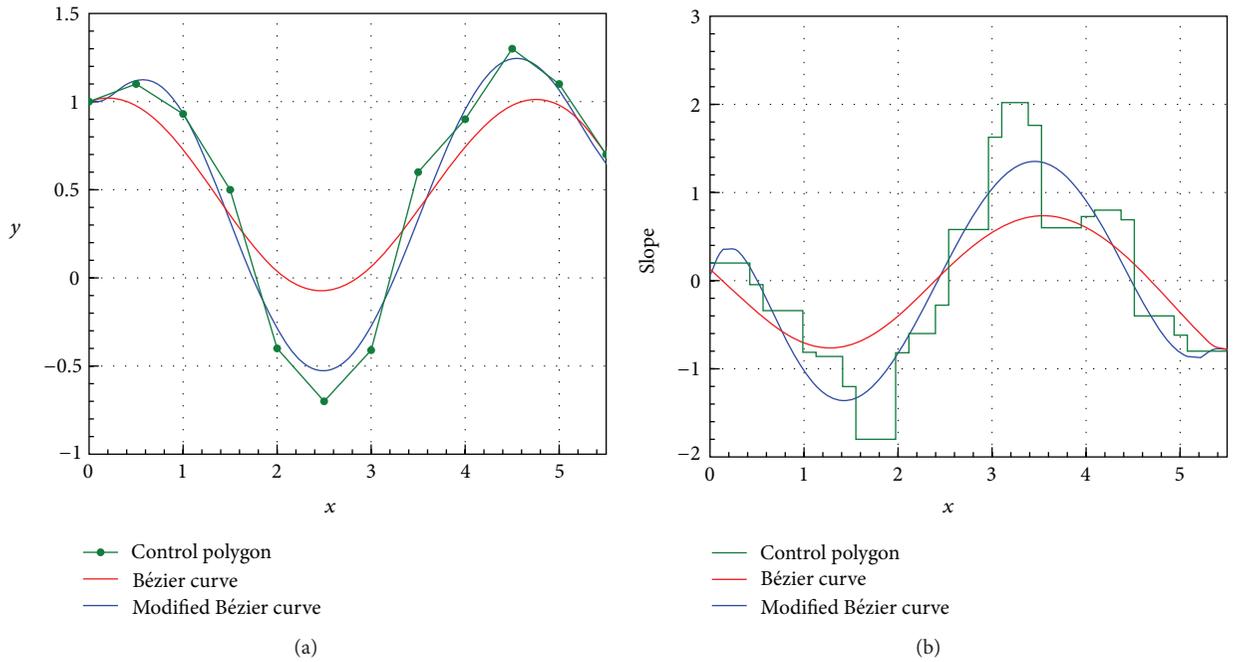


FIGURE 3: Comparison of modified Bézier curve and conventional Bézier curve for test function  $f_2$ : (a) fitted curve, (b) slope of the curve.

shown in Figures 2, 3, 4, and 5. The data points of the results are highlighted in Tables 1, 2, 3, and 4. In the data table,  $x_i$  and  $y_i$  indicate the control points,  $y_{bi}$  refers to corresponding  $y$  values obtained from the conventional Bézier curve,  $y_{mi}$  refers to corresponding  $y$  values obtained from the modified

curve, and  $\zeta_i$  refers to the values of the shaping parameters. The results show that the modified Bézier curve follows the shape of the sharply varying functions much better compared to the conventional Bézier curve. The sum of squares error of the curve and the control polygon is much smaller for the

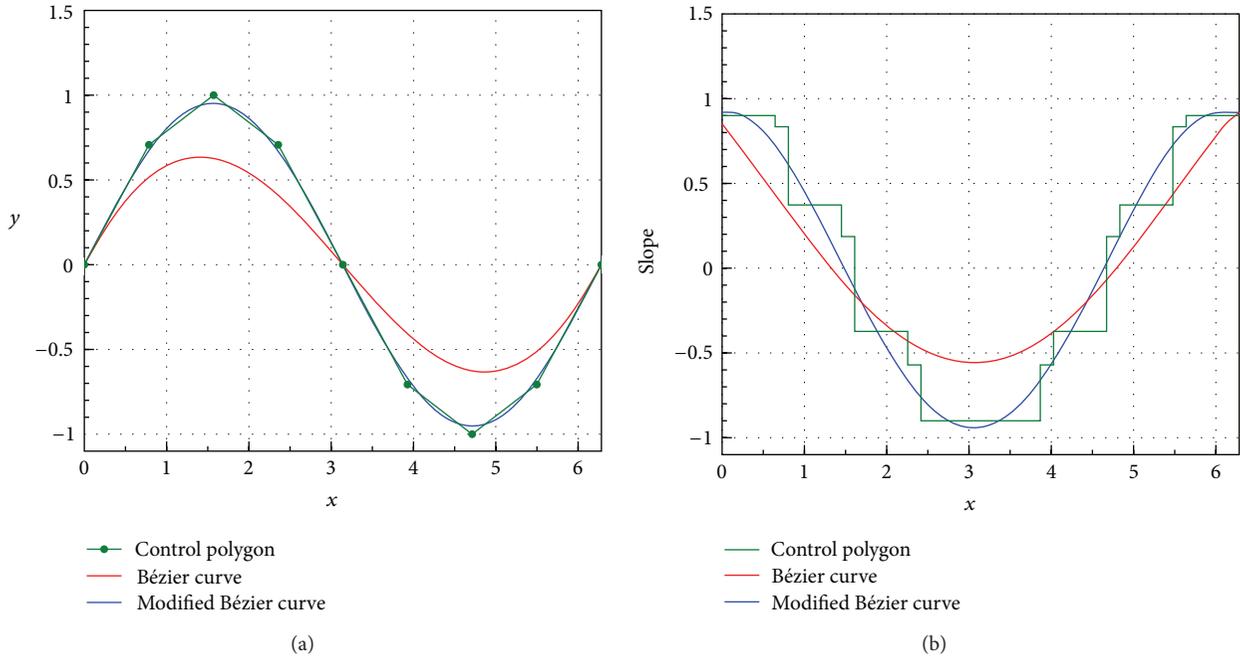


FIGURE 4: Comparison of modified Bézier curve and conventional Bézier curve for test function  $f_3$ : (a) fitted curve, (b) slope of the curve.

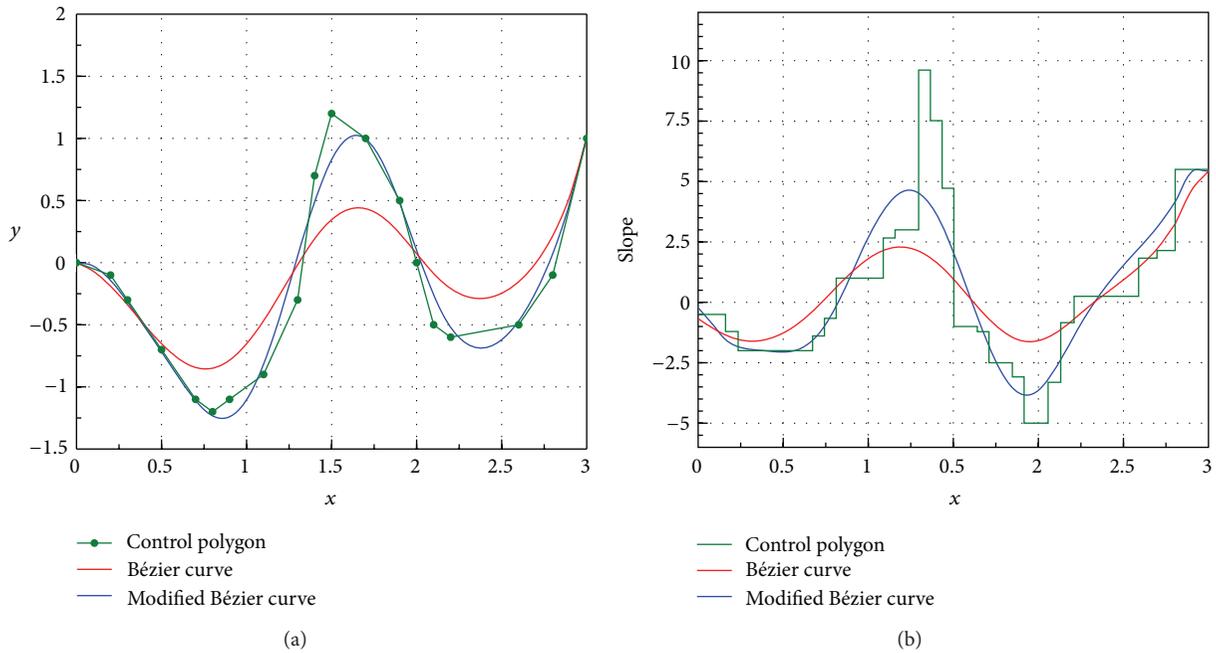


FIGURE 5: Comparison of modified Bézier curve and conventional Bézier curve for test function  $f_4$ : (a) fitted curve, (b) slope of the curve.

case of modified Bézier curve. The sum of squares error of the slope of the curve and the slope of the control polygon is also smaller for the modified Bézier curve.

To ensure that the DE algorithm reached convergence, the cost versus generation number plot was observed for all

the test functions. All the plots show a decreasing nature in the beginning and flats out around iteration 110 to 160. The flat values of average cost and minimum cost indicate that saturation is reached [15]. For illustration purposes, only the plot for test function  $f_1$  is shown in Figure 6.

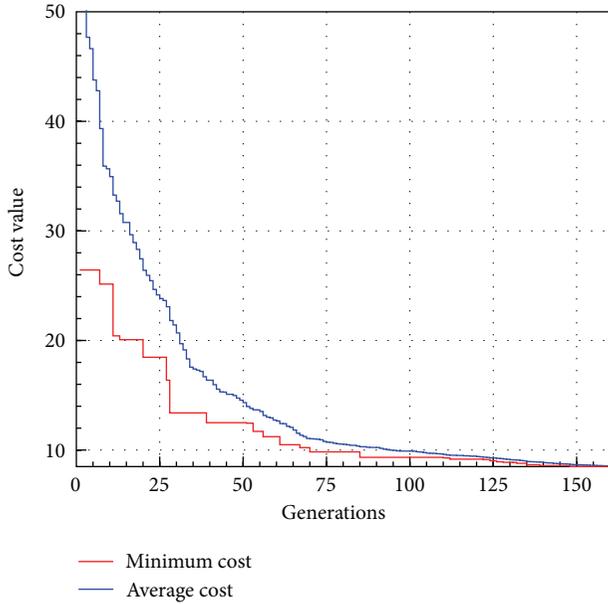


FIGURE 6: Cost values versus iteration number for test function  $f_1$ .

TABLE 2: Table for test function,  $f_2$ .

$i$	$x_i$	$y_i$	$y_{bi}$	$y_{mi}$	$\zeta_i$	Performance
0	0	1.00	1.0000	0.9971	0.9971	
1	0.5	1.10	0.9766	1.1213	0.8039	
2	1.0	0.93	0.7251	0.9270	2.3924	
3	1.50	0.50	0.3534	0.3192	0.5058	$\epsilon_{bc} = 2.9740$
4	2.00	-0.40	0.0379	-0.2884	2.0301	$\epsilon_{mc} = 0.3168$
5	2.50	-0.70	-0.0723	-0.5262	2.8836	$\epsilon_{bs} = 12.9069$
6	3.00	-0.41	0.0630	-0.2787	2.6308	$\epsilon_{ms} = 5.7855$
7	3.50	0.60	0.3825	0.3249	0.7992	$f_{cost,b} = 4.2646$
8	4.00	0.90	0.7396	0.9499	2.1468	$f_{cost,m} = 0.8954$
9	4.50	1.30	0.9769	1.2445	1.3437	
10	5.00	1.10	0.9786	1.0629	0.9038	
11	5.50	0.70	0.7000	0.6471	0.9244	

TABLE 3: Table for test function,  $f_3$ .

$i$	$x_i$	$y_i$	$y_{bi}$	$y_{mi}$	$\zeta_i$	Performance
0	0	0	0	0	-2.2409	
1	0.7854	0.7071	0.5135	0.6729	1.0097	
2	1.5708	1.0000	0.6263	0.9517	1.5724	$\epsilon_{bc} = 1.9505$
3	2.3562	0.7071	0.4047	0.6647	2.1276	$\epsilon_{mc} = 0.0105$
4	3.1416	0.0000	0.0040	0.0066	-2.6606	$\epsilon_{bs} = 2.7549$
5	3.9270	-0.7071	-0.4047	-0.6648	2.1291	$\epsilon_{ms} = 0.5728$
6	4.7124	-1.0000	-0.6263	-0.9514	1.5718	$f_{cost,b} = 2.2260$
7	5.4978	-0.7071	-0.5135	-0.6724	1.0081	$f_{cost,m} = 0.0678$
8	6.2832	-0.0000	-0.0000	-0.0000	1.5754	

### 5. Conclusion

A modified parametric equation is developed by modifying the equation of Bézier curve. The resulting modified curve allows control over the shape of the fitted curve by

TABLE 4: Table for test function,  $f_4$ .

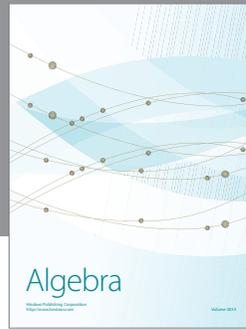
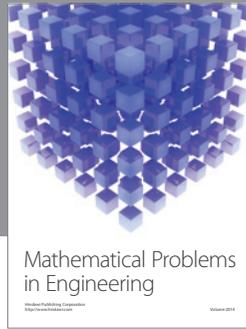
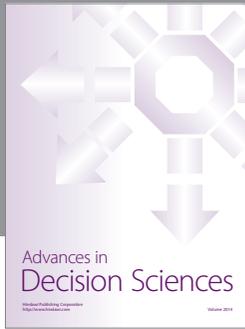
$i$	$x_i$	$y_i$	$y_{bi}$	$y_{mi}$	$\zeta_i$	Performance
0	0	0	0	0	0.5461	
1	0.20	-0.10	-0.1851	-0.1394	0.2394	
2	0.30	-0.30	-0.3436	-0.3256	-0.3020	
3	0.50	-0.70	-0.6508	-0.7163	2.8020	
4	0.70	-1.10	-0.8440	-1.1115	-0.6660	
5	0.80	-1.20	-0.8495	-1.2315	1.9287	
6	0.90	-1.10	-0.7862	-1.2413	2.0117	$\epsilon_{bc} = 4.7260$
7	1.10	-0.90	-0.4626	-0.8119	2.7428	$\epsilon_{mc} = 0.6024$
8	1.30	-0.30	-0.0211	0.0393	2.9284	$\epsilon_{bs} = 155.1711$
9	1.40	0.70	0.1929	0.4972	1.9595	$\epsilon_{ms} = 78.8906$
10	1.50	1.20	0.3446	0.8282	2.9170	$f_{cost,b} = 20.2431$
11	1.70	1.00	0.4338	0.9920	2.5532	$f_{cost,m} = 8.4915$
12	1.90	0.50	0.2366	0.4872	0.9447	
13	2.00	0	0.0771	0.1093	-1.6003	
14	2.10	-0.50	-0.0844	-0.2551	2.8826	
15	2.20	-0.60	-0.2089	-0.5216	1.9124	
16	2.60	-0.50	-0.1548	-0.4667	1.9470	
17	2.80	-0.10	0.2163	0.0712	1.8031	
18	3.00	1.00	1.0000	1.0213	1.0213	

introducing shaping parameters. Using DE algorithm, the optimum value of the shaping parameters can be found. The optimum curve matches the shape of the control polygon and the slope of the control polygon with a much higher degree of accuracy compared to the conventional Bézier curve. Multiple test functions are used to test the proposed method. A number of performance parameters are defined, which shows that the proposed method outperforms conventional Bézier curve. It is found that for sharply varying data points, the proposed method produces a curve that follows the control polygon more closely compared to conventional Bézier curve. As the proposed method is general, it can be used on any discrete sets of data of points to produce a highly accurate shape preserving curve.

### References

- [1] F. Curtis Gerald and O. Patrick Wheatley, *Applied Numerical Analysis*, Pearson, 7th edition.
- [2] K. Anastasiou and C. T. Chan, "Automatic triangular mesh generation scheme for curved surfaces," *Communications in Numerical Methods in Engineering*, vol. 12, no. 3, pp. 197–208, 1996.
- [3] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes, "Isogeometric finite element data structures based on Bézier extraction of T-splines," *International Journal for Numerical Methods in Engineering*, vol. 88, no. 2, pp. 126–156, 2011.
- [4] W. Wen-tao and W. Guo-zhao, "Bézier curves with shape parameter," *Journal of Zhejiang University Science A*, vol. 6, no. 6, pp. 497–501, 2005.
- [5] Q.-B. Wu and F.-H. Xia, "Shape modification of Bézier curves by constrained optimization," *Journal of Zhejiang University Science*, vol. 6, no. 1, pp. 124–127, 2005.
- [6] F. A. Sohel, G. C. Karmakar, and L. S. Dooley, "An improved shape descriptor using bezier curves," in *Pattern Recognition*

- and Machine Intelligence*, vol. 3776 of *Lecture Notes in Computer Science*, pp. 401–406, Springer, Berlin, Germany, 2005.
- [7] M. I. Grigor'cprimeev, V. N. Malozemov, and A. N. Sergeev, "Bernstein polynomials and composite Bézier curves," *Journal of Computational Mathematics and Mathematical Physics*, vol. 46, no. 11, pp. 1962–1971, 2006.
  - [8] G. T. Tachev, "Approximation of a continuous curve by its Bernstein-Bézier operator," *Mediterranean Journal of Mathematics*, vol. 8, no. 3, pp. 369–381, 2011.
  - [9] L. Yang and X.-M. Zeng, "Bézier curves and surfaces with shape parameters," *International Journal of Computer Mathematics*, vol. 86, no. 7, pp. 1253–1263, 2009.
  - [10] Q. Chen and G. Wang, "A class of Bézier-like curves," *Computer Aided Geometric Design*, vol. 20, no. 1, pp. 29–39, 2003.
  - [11] N. F. Fijasri, S. H. Yahaya, and J. M. Ali, "Bézier-like quartic curve with shape control," in *2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications*, pp. 1–6, June 2006.
  - [12] W.-Q. Shen and G.-Z. Wang, "A class of quasi Bézier curves based on hyperbolic polynomials," *Journal of Zhejiang University Science*, vol. 6, no. 1, pp. 116–123, 2005.
  - [13] Ç. Dişibüyük and H. Oruç, "A generalization of rational Bernstein-Bézier curves," *BIT. Numerical Mathematics*, vol. 47, no. 2, pp. 313–323, 2007.
  - [14] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
  - [15] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*, Springer, 2005.
  - [16] P. Pandunata and S. M. H. Shamsuddin, "Differential evolution optimization for Bézier curve fitting," in *Proceedings of the 7th International Conference on Computer Graphics, Imaging and Visualization (CGIV '10)*, pp. 68–72, August 2010.
  - [17] T. Rogalsky, "Bézier parameterization for optimal control by differential evolution," in *International Conference on Genetic and Evolutionary Methods*, pp. 28–34, July 2011.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

