



# Reconstruction of the initial curve from a two-dimensional shape for the B-spline curve fitting

Almas Jahanshahloo<sup>1,a</sup>, ALireza Ebrahimi<sup>2,b</sup>

<sup>1</sup> Department of Mathematics, East Tehran Branch, Islamic Azad University, Tehran, Iran

<sup>2</sup> Computer Geometry and Dynamical Systems Laboratory, Faculty of Mathematical Sciences, Yazd University, Yazd, Iran

Received: 21 February 2022 / Accepted: 15 March 2022

© The Author(s), under exclusive licence to Società Italiana di Fisica and Springer-Verlag GmbH Germany, part of Springer Nature 2022

**Abstract** Curve reconstruction is a significant challenge in computer-aided geometric design, computational atomic and molecular physics, engineering design, virtual reality and data visualization. This study discusses a method of producing B-spline parametric curve from the large number of data points. The introduced scheme includes three major parts for curve reconstruction: (1) formulation of the B-spline curve fitting as a nonlinear least squares optimization problem, (2) construction of the precise initial B-spline curve using properly determined control points, and (3) usage of the diagonal approximation BFGS method to identify the location parameters and the control points simultaneously. The modeling examples demonstrate that the suggested techniques are successful and can therefore significantly reduce fitting error by adjusting the number and location of control points.

## 1 Introduction

Constructing curve and surface from discrete data points plays an essential role and is used as a basic step in the scope of computer-aided geometric design (CAGD) and manufacturing [1]. These data points are captured using scanning devices or extracting the edge of the generic shapes. Processing and storing a high number of points necessitates a lot of memory and is both difficult and time-consuming [2]. The purpose of reconstructing the shapes is to create an appropriate mathematical model from a given set of points [3].

Among all type of curves, B-spline function is one of the most favorite bases for curve reconstruction and shape modeling [4]. B-spline curve is a piecewise polynomial with useful properties such as smoothness, local effects, affine invariance, strong convex hull, great flexibility, numerical stability of calculation and coefficient changes, which is very useful for complex geometry [5].

The use of B-spline curve in calculations of atomic and molecular structure and dynamics has been one of the most significant developments in computational atomic and molecular physics in recent years. The B-splines basis has the feature of being “complete enough” with a small number of basis functions and can therefore contend with finite-difference approaches. This explains the recent trend toward the use of B-splines in atomic calculations, while B-splines also offer significant improvements over older basis sets for molecular issues [6].

The main procedure of this study for curve fitting with B-spline curve is to formulate it as a nonlinear least squares problem where the order and knots are fixed. We fit the B-spline curve  $C(\mathcal{P}, \mathcal{T})$ , as accurately as desired, to a set of data information about a known 2D shape  $Q$ . Here,  $\mathcal{P}$  and  $\mathcal{T}$  are the set of shape parameters and location parameters, respectively. The following are the basic steps in this procedure:

**Step 1.** Convert the fitting of B-spline curves to a nonlinear least squares optimization problem.

**Step 2.** Construct a precise initial B-spline curve using properly selecting the location parameters ( $\mathcal{T}$ ) and the shape parameters ( $\mathcal{P}$ ).

**Step 3.** Select appropriate optimization technique for resolving the nonlinear least squares equation obtained in step 1 and use initial B-spline curve archived in step 2 as the starting point of the optimization method.

These steps can be applied to solve the problem of B-spline curve fitting that occurs in practical applications. Each of these steps can be studied as an independent study. Our main contribution in this work is to construct a suitable initial curve using accurate selection of shape parameters. The accuracy and speed of convergence of the optimization approach are improved when the starting B-spline curve is appropriate. In fact, the main goal is to bring the primary B-spline curve closer to the desired shape. By constructing a proportionate starting B-spline curve, the quantity of iterations of the optimization approach to reach the optimal solution and find the final curve is reduced. After constructing the primary B-spline curve, we use a minimization procedure based on generalized BFGS technique and, in special cases, the diagonal approximation BFGS technique to find the final B-spline curve

<sup>a</sup> e-mail: [Almasj63@yahoo.com](mailto:Almasj63@yahoo.com)

<sup>b</sup> e-mails: [a.ebrahimi@stu.yazd.ac.ir](mailto:a.ebrahimi@stu.yazd.ac.ir); [a.ebrahimi65@gmail.com](mailto:a.ebrahimi65@gmail.com) (corresponding author)

and solve nonlinear least squares optimization problem. The diagonal approximation BFGS method optimizes the shape parameters and location parameters concurrently. This optimization method does not require to execute foot point projection and eliminate the cost of solving a very large linear system of equations at every time step. The complexity per step of the diagonal approximation BFGS method is  $O(n)$ , and it requires only  $O(n)$  memory allocations [7].

The approach suggested in this study for creating the initial curve of two-dimensional shapes is entirely novel, and earlier research has not addressed this problem.

The structure of the current paper is arranged as follows. In Sect. 2, we summarily overview relevant previous literature in curve fitting and its main application. Section 3 gives some notations about B-spline curve, and the B-spline curve fitting is formulated as a nonlinear least squares optimization problem. In Sect. 4, we construct a primary B-spline curve using choosing the shape parameters. The diagonal approximation BFGS method as a particular example of the generalized BFGS approach is employed in Sect. 5 to optimize the nonlinear least squares problem. In Sect. 6, we illustrate the efficiency and accuracy of the introduced approach through numerical results and comparisons with existing methods. Section 7 discusses the experimental results and potential applications of the proposed method. The main conclusions with our project for future study are presented in Sect. 8.

## 2 Related work

In this section, we overview some significant techniques for B-spline curve fitting (and its special cases Bézier curve fitting) to provide a history for our work. The development of novel computational approaches for curve fitting has gotten a lot of interest in recent years. The subject of curve fitting has different applications in many theoretical areas of the literature, such as approximation theory [8,9], numerical optimization [10,11], statistics [12] and geometric modeling [2,3,13,14]. Moreover, the curve fitting has fundamental importance in several applied sciences, including computer-aided design [1,5,15–18], font designing [19–21], data compression [4,22–24], image processing [25–29], cultural heritage preservation [30], video compression [31–34], computer graphics [35], computer-aided manufacturing (CAM) [36,37], computer animation [38], virtual reality [39] and many others.

Wang et al. [11] presented the squared distance minimization (SDM) to approximate a set of point clouds using B-spline curve. Yang et al. [40] commenced with a preliminary B-spline curve and then combined SDM with process for changing the quantity and positions of the control points of the operational parametric B-spline curve. Park and Lee [41] introduced a method with four phases: parameterization, significant point choice, placement of the knot, and fitting a B-spline curve to a set of sorted data points using least-squares optimization.

The Gauss–Newton approach for non-uniform rational B-splines (NURBS) curve and surface fitting is proposed in [42]. In [43], the variable projection approach is used to separate the NURBS curves fitting into a linear least squares problem. For the NURBS curve fitting, Bergstrom et al. [10] have discussed a subproblem in the parameter estimation using the Gauss Newton technique.

Speer et al. [44] employed the Gauss–Newton approach to optimize the location parameters and control points at the same time, which every iteration necessitated to find a solution of the linear system equations and store the Jacobian matrices. Using the L-BFGS method, Zheng et al. [45] were able to optimize control points as well as foot points at the same time, without having to find a solution of the linear system equations in each iteration. They manually create the primary shape and place additional control points at the section that has a large fit-error margin. A method based on using a diagonal approximation BFGS method is suggested in [46] to optimize the related nonlinear least squares problem. After finding the solution of the linear system equations, they were able to determine the starting control points. Later in 2018, Ebrahimi and Loghmani [47] suggested a procedure for determining the number and position of control points in order to generate an appropriate starting B-spline curve.

In recent years, nature-inspired metaheuristic strategy has been used to answer complex optimization problems that cannot be solved using conventional optimization algorithms. Some of the more recent approaches in this field include the use of the particle swarm optimization (PSO) method [48,49], multiobjective evolutionary algorithm [50], firefly algorithm [51], simulated annealing [52], memetic bat algorithm [53] and estimation of distribution algorithms [54].

Bachau et al. [6] prepared a survey that focused on studies in the applications of B-splines in atomic and molecular physics sets up to 2000. Weber et al. [55] showed the versatility of the B-spline model as a tool in the analysis of hydrogenated amorphous carbon (a-C:H) thin films. The dielectric function of the a-C:H film is parameterized using B-splines in their work. The least square fitting approach using cubic B-spline basis functions is developed in [56] to reduce the impact of statistical fluctuations on gamma ray spectra. In [57], a modeling methodology for an electric heating flow controlling valve using a B-spline function and the recursive least squares method is suggested.

Many researchers have introduced various approaches about nonlinear physics [58–61]. For picosecond-pulse attenuation in a multiphase inhomogeneous optical fiber with various frequencies, a symbolic calculation on a three-coupled variable-factor nonlinear Schrödinger system is provided in [62]. Gao et al. [63] carried out symbolic calculations on the Boussinesq–Burgers equation system to determine the height of shallow water waves in a lake or along an ocean beach. The symbolic computing for the water waves on a  $(2 + 1)$ -dimensional generalized variable-coefficient Boiti–Leon–Pempinelli system is presented in [64]. The open sea is framed in terms of a generalized  $(2 + 1)$ -dimensional dispersive long-wave system for shallow water in [65].

### 3 B-spline notations and problem description

Assume  $m_1$  and  $k$  are two integers such that  $1 \leq k \leq m_1 + 1$ . Suppose  $U = \{u_0, u_1, \dots, u_{m_1+k}\}$  be a finite monotone increasing of real numbers, which is termed a knot vector. The associated B-spline basis function  $N_i^k(t)$  of order  $k$  (or equivalently, degree  $k - 1$ ) may be recursively given using the Cox–de Boor relations, as shown as follows:

$$N_i^1(t) = \begin{cases} 1, & u_i \leq t \leq u_{i+1} \\ 0, & \text{otherwise,} \end{cases}$$

and for  $k > 1$ , we have

$$N_i^k(t) = \frac{t - u_i}{u_{i+k-1} - u_i} N_{i,k-1}(t) + \frac{u_{i+k} - t}{u_{i+k} - u_{i+1}} N_{i+1,k-1}(t), \tag{1}$$

If required, (1) use the convention  $\frac{0}{0} = 0$ . The interval  $[u_i, u_{i+k+1})$  is known as the support domain of  $N_i^k(t)$  since  $N_i^k(t) = 0$  for  $t \notin [u_i, u_{i+k+1})$ . Let  $\mathcal{P} := \{P_i\}_{i=0}^{m_1} \in \mathbb{R}^2$  be  $m_1 + 1$  control points, the set of shape parameters, and  $N_i^k(t)$  be a B-spline basis function of order  $k$  expressed on knot vector  $U$ , then a B-spline parametric curve  $C(t)$  of order  $k$  (or equivalently, degree  $k - 1$ ) is formulated as follows:

$$C(t) = \sum_{i=0}^{m_1} P_i N_i^k(t), \tag{2}$$

The external knots  $u_0, \dots, u_{k-1}$  and  $u_{m_1+1}, \dots, u_{m_1+k}$  are often adjusted such that  $u_0, \dots, u_{k-1} = 0$  and  $u_{m_1+1}, \dots, u_{m_1+k} = 1$ . This guarantees that  $C(0) = P_0$  and  $C(1) = P_{m_1}$ , i.e., that the ends of the curve  $C$  correspond with the ends of the polygonal chain  $\{P_i\}_{i=0}^{m_1}$ .

The main objective of this study is to fit B-spline curve (2), with order  $k$  and fixed knot vectors  $U$ , to a collection of two-dimensional data points. We assign the location parameters  $\mathcal{T} := \{t_j\}_{j=0}^{n_1}$  for each of the data points  $\{Q_j\}_{j=0}^{n_1}$ . The problem of fitting B-spline curve  $C(t)$  to data point  $\{Q_j\}_{j=0}^{n_1}$  is often expressed as a nonlinear least squares optimization problem and can be stated as

$$\min_{\mathcal{P}, \mathcal{T}} \frac{1}{2} \sum_{j=0}^{n_1} \|C(t_j) - Q_j\|^2 = \min_{\mathcal{P}, \mathcal{T}} \frac{1}{2} \sum_{j=0}^{n_1} \left\| \sum_{i=0}^{m_1} P_i N_i^k(t_j) - Q_j \right\|^2 \tag{3}$$

The goal of solving the nonlinear least squares problem (3) is to find  $n_1 + 1$  location parameters and  $m_1 + 1$  shape parameters (each with two coordinates) to design B-spline curve  $C(t)$  which closely approximates the point cloud  $\{Q_j\}_{j=0}^{n_1}$ . As a result, the size of the optimization space is  $2(m_1 + 1) + n_1 + 1$ , where  $m_1 \ll n_1$ .

Using the diagonal approximation BFGS method, we will simultaneously optimize the shape and location parameters. We need an initial approximation of the shape and location parameters before we can use the optimization method. The proposed method is presented in the next section for selecting the initial control points (shape parameters), which produces a initial B-spline curve that is quite similar to the intended shape.

### 4 The initial shape

Producing a suitable initial B-spline curve is critical in curve fitting and increasing the converging speed of the optimization method. The shape and location parameters must be approximated correctly in order to create a suitable starting curve. The initial location parameters are easily determined using uniform or centripetal parameterization. However, new strategies must be proposed to determine the initial shape parameters, including the number and position of certain control points. The procedure for constructing the initial curve and approximating the number and position of control points is detailed as follows.

Let  $\{Q_j\}_{j=0}^{n_1}$  be  $n_1$  points on a shape and  $L$  be the length parameter that depends on the amount of point clouds ( $n_1$ ) and shape complexity. We calculate for  $s = 0, 1, \dots, L$  the perpendicular distance  $D_{r,s}$  from point  $Q_r$  to the straight line joining the point  $Q_{r-L+s}$  and  $Q_{r+s}$ . Then, the chord-to-point distance accumulation (CPDA) at the point  $Q_r$  is computed as

$$H_j = \sum_{s=0}^L D_{j,s}, \quad j = 0, 1, \dots, n_1.$$

Therefore, we use the chord-to-point distance accumulation to assign the values  $\{H_j\}_{j=0}^{n_1}$  to data points  $\{Q_j\}_{j=0}^{n_1}$ . The chord-to-point distance accumulation  $\{H_j\}_{j=0}^{n_1}$  is smoothed using the moving average filter to reduce noise and improve method accuracy. The local maximums on the values  $\{H_j\}_{j=0}^{n_1}$  are now extracted, and the total quantity of these local maxima specifies the total quantity of intermediate control points. Let  $J = \{j_1, \dots, j_{m_1-1}\}$  be the index set of the data point set for which  $\{H_j\}_{j=0}^{n_1}$  has the local

maximum. The primitive location of the intermediate control points is obtained as follows:

$$P_r = Q_{j_r} - \left( \max_{0 \leq j \leq n} H_j \right) \times H_{j_r} \times N_{j_r}, \quad r = 1, 2, \dots, m_1 - 1.$$

where  $N_{j_r}$  is the principal unit normal vector at  $Q_{j_r}$  which may be calculated via numerical differentiation. The B-spline curve interpolates the starting and ending control points using choosing the appropriate knot vector. Therefore, the first and last control points are selected as follows:

$$P_0 = Q_0, \quad P_{m_1} = Q_{n_1}.$$

Figure 1 demonstrates an example of the suggested method for constructing the primary B-spline curve with a length parameter of 8. The point cloud set is shown in (a), and the change of the chord-to-point distance accumulation  $\{H_j\}_{j=0}^{n_1}$  together with the finding of local maximums is shown in (b). Arrows in (c) indicate the tangent and normal vectors at the locations of local maximums. The initial B-spline curves produced using the suggested technique are shown in (d). In real-world applications, we come across shape with varying degrees of complexity, number of points, resolution and object scaling. The length parameter  $L$  is used to generate the primary B-spline curve according to the kind of shape and the required accuracy. Figure 2 illustrates a comparison of the suggested approach for various length parameters  $L$  and the time required for computation as given in Table 2. As shown in Fig. 2, decreasing the length parameter  $L$  more precisely calculates the curvature of the shape and increases the number of local maximums. As a consequence, lowering the length parameter  $L$  raises the number of control points and produces a more precise initial B-spline curve. Furthermore, as shown in Table 1, when the length parameter  $L$  is decreased, the number of calculations required to calculate the chord-to-point distance accumulation reduces, and the time required to identify the initial location of the control points lowers. Our suggested method, which has a parameter based on the complexity of the shape and the number of data point set, generates a primary B-spline curve with appropriate speed and the required approximation. Although the produced initial curve accurately approximates the shape structure, we utilize an appropriate optimization technique in the next section for a highly good approximation of the desired shape and use the generated initial curve as the starting point of the optimization process.

### 5 The diagonal approximation BFGS method

Thus far, we have converted the B-spline curve fitting problem into an unconstrained optimization problem and generated a primary B-spline curve which can be applied to start the optimization process. In this part, we will discuss the diagonal approximation BFGS technique for solving the nonlinear optimization problems (3), which is based on the results presented in [66].

Assume we need to solve an unconstrained optimization problem

$$\min f(\mathbf{x}) \tag{4}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is a nonlinear function to optimize and  $n$  is large. We assume that  $f(x)$  is twice continuously differentiable and bounded from below. A quasi-Newton technique produces the following sequence given an initial point  $x_0 \in \mathbb{R}^n$  and an initial approximation  $\mathfrak{B}_0 \in \mathbb{R}^{n \times n}$  to the Hessian matrix of function  $f$  in  $x_0$ :

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \dots, \tag{5}$$

where  $d_k \in \mathbb{R}^n$  denotes the search direction as determined by the linear system solution:

$$\mathfrak{B}_k d_k = -g_k,$$

where  $g_k = g(x_k) = \nabla f(x_k)$  and matrix  $\mathfrak{B}_k$  represents a quasi-Newton approximation to the Hessian  $\nabla^2 f(x_k)$  that solves the secant equation

$$\mathfrak{B}_{k+1} s_k = y_k,$$

where  $s_k = x_{k+1} - x_k$  and  $y_k = g_{k+1} - g_k$ . In practice, the step size  $\alpha_k$  in (5) is decided by the Wolfe line search criteria as follows:

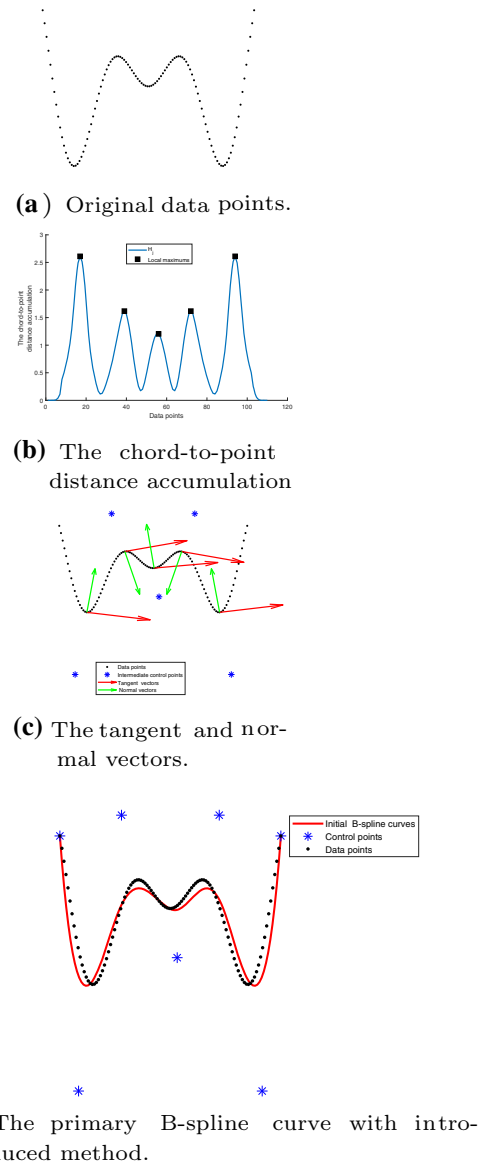
$$\begin{cases} f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k g(x_k)^T d_k, \\ g(x_k + \alpha_k d_k)^T d_k \geq c_2 g(x_k)^T d_k, \end{cases} \tag{6}$$

where the positive constants  $c_1$  and  $c_2$  are chosen in such a way that they satisfy  $0 < c_1 < c_2 < 1$ .

The Broyden–Fletcher–Goldfarb–Shanno (BFGS) technique is one of the most well-known quasi-Newton approaches, in which the updates are described by

$$\mathfrak{B}_{k+1} = \Upsilon(\mathfrak{B}_k, s_k, y_k), \tag{7}$$

**Fig. 1** Construction of the primary parametric B-spline curve with a length parameter of 8



where

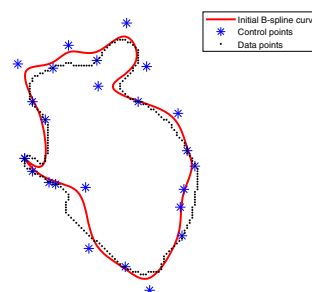
$$\Upsilon(\mathfrak{B}, s, y) = \mathfrak{B} + \frac{1}{y^T s} y y^T - \frac{1}{s^T \mathfrak{B} s} \mathfrak{B} s s^T \mathfrak{B}. \tag{8}$$

The BFGS technique provides global and local superlinear convergence speed under suitable assumptions about the objective function  $f(x)$  and starting point  $x_0$ .

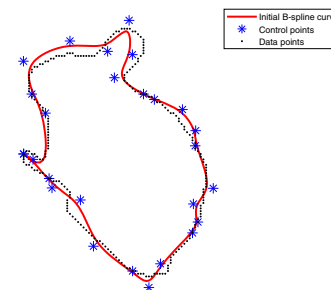
We are confronted with large-scale optimization problem in this research. Standard solution methods for (4), such as one-step secant (OSS-OSSV) method, BFGS method and limited memory-BFGS (L-BFGS) method, are not appropriate for large-scale optimization problems [67,68]. Although the OSS-OSSV approach has a reasonable computational cost of  $O(n)$ , it has an extremely lower convergence rate. The BFGS approach requires  $O(n^2)$  memory to store the approximate Hessian of  $f(x)$  which is a high computational cost. The L-BFGS technique, which has a computational complexity of  $O(mn)$ , where  $m$  is the number of steps stored in memory by parameter declaration, diverges for some large-scale problems. The primary goal is to save the second-order information of the matrix  $\mathfrak{B}_k$ , which is generated by the BFGS technique to approximate a complete Hessian of  $f(x)$ , in a manner that enables to minimize the high ( $O(n^2)$ ) computing cost per step of BFGS. To decrease the computational cost per step and memory space complexity of the BFGS technique, the following relation can be used to substitute  $\mathfrak{B}_k$  with a simpler matrix  $\tilde{\mathfrak{B}}_k$ :

$$\mathfrak{B}_{k+1} = \Upsilon(\tilde{\mathfrak{B}}_k, s_k, y_k), \tag{9}$$

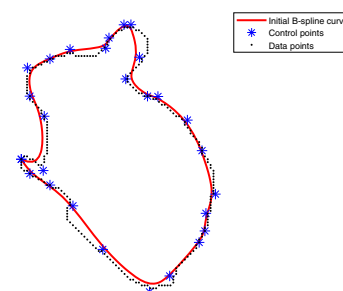
**Fig. 2** A variation on the initial shape using a different length parameter value  $L$



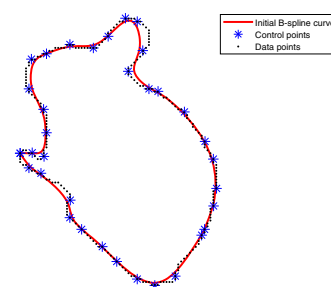
**(a)** The initial parametric B-spline curve with  $L = 5$ .



**(b)** The initial parametric B-spline curve with  $L = 4$ .



**(c)** The initial parametric B-spline curve with  $L = 3$ .

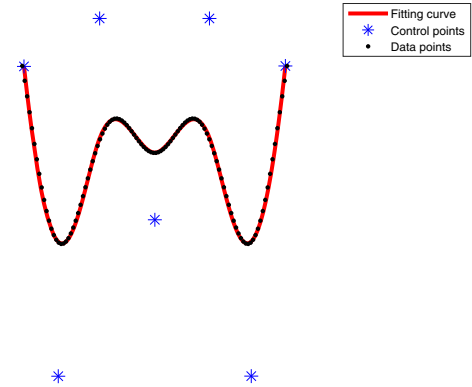


**(d)** The initial parametric B-spline curve with  $L = 2$ .

**Table 1** Performance of the proposed methodology for various length parameter  $L$  values in Fig. 2

| Length parameter | Number of control points | Computational time (in s) |
|------------------|--------------------------|---------------------------|
| $L = 5$          | 25                       | 0.2604                    |
| $L = 4$          | 27                       | 0.2586                    |
| $L = 3$          | 28                       | 0.2570                    |
| $L = 2$          | 35                       | 0.2535                    |

**Fig. 3** Fitting curve with optimization algorithm 1



This iterative process produces the algorithm known as the generalized BFGS (G-BFGS). The specific requirements on the trace and the determinant of  $\tilde{\mathfrak{B}}_k$  for the convergence of the G-BFGS method are given in [7,66]. It is possible to consider the traditional methods OSS-OSSV and L-BFGS to be special versions of the G-BFGS algorithm by adjusting the  $\tilde{\mathfrak{B}}_k = I$  and

$$\tilde{\mathfrak{B}}_k = \Upsilon(\Upsilon(\dots \Upsilon(\mathfrak{B}_k^0, s_{k-m+1}, y_{k-m+1}) \dots, s_{k-2}, y_{k-2}), s_{k-1}, y_{k-1}),$$

respectively, where  $I$  is the identity matrix and  $\mathfrak{B}_k^0 = (y_k^T y_k / s_k^T y_k) I$ .

Assume  $D_B$  represents the diagonal matrix generated from the diagonal components of the main matrix  $B$ . The diagonal approximation BFGS (D-BFGS) technique is introduced by choosing  $\tilde{\mathfrak{B}}_k = D_{\mathfrak{B}_k}$  as follows:

$$\mathfrak{B}_{k+1} = \Upsilon(D_{\mathfrak{B}_k}, s_k, y_k). \tag{10}$$

The D-BFGS approach maintains an appropriate compromise between the OSS-OSSV and BFGS techniques. Both the space and time complexity of the D-BFGS method are  $O(n)$  which is particularly well suited for solving large-scale problems. Algorithm 1 illustrates the inverse form of the D-BFGS technique using the Sherman–Morrison–Woodbury matrix inversion lemma.

**Algorithm 1** The inverse form of the D-BFGS method.

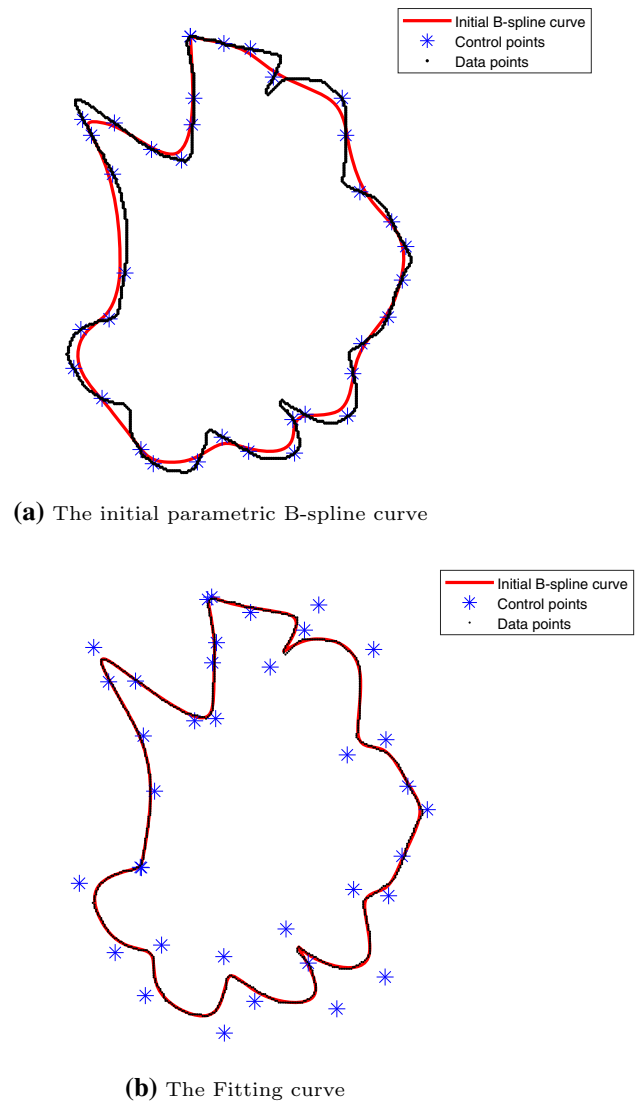
Given a starting point  $\mathbf{x}_0 \in \mathbb{R}^n$  and  $\mathcal{H}_0 = I$  (identity matrix);  
**for**  $k = 0, 1, \dots$  until convergence **do**  
 Compute  $d_k = \begin{cases} -\mathcal{H}_k g_k, & \text{Secant algorithm (S),} \\ -D\mathcal{H}_k g_k, & \text{Non -secant algorithm (NS),} \end{cases}$   
 Compute the step size  $\alpha_k$  by the Wolfe line search ;  
 Compute the next iterate  $x_{k+1} = x_k + \alpha_k d_k$ ;  
 Set  $s_k = x_{k+1} - x_k, \quad y_k = g_{k+1} - g_k$ ;  
 Update  $\mathcal{H}_{k+1} = \Omega(D\mathcal{H}_k, s_k, y_k)$ , where

$$\Omega(\mathcal{H}, s, y) = \left( I - \frac{y s^T}{y^T s} \right)^T \mathcal{H} \left( I - \frac{y s^T}{y^T s} \right) + \frac{s s^T}{y^T s}. \tag{11}$$

**end for**

We use Algorithm 1 to optimize the nonlinear least squares problem (3) that the starting point of the optimization algorithm is chosen in accordance with the approach provided in Sect. 4. Figure 3 shows the final fitted curves for the data points in Fig. 1 after the use of the optimization Algorithm 1.

**Fig. 4** Constructed B-spline curve for dataset in Example 1



### 6 Demonstration

In this section, we implement the proposed approach for challenging shapes with a high data points number to evaluate its performance. Additionally, we compare the suggested approach to the results reported in [47] in terms of the number of control points, starting and final errors, and execution time. The execution was carried out in MATLAB R2020a on a laptop equipped with a 2.60 GHz Intel Core i5 CPU and 6 GB RAM. While the technique we present is applicable to B-spline curve fitting of any order, we chose B-spline curve of order 4 for 2D shape reconstruction in the performed examples because of its relative simplicity and widespread implementation in the fields of shape modeling and CAGD. In our method, the knot vectors  $U$  remain constant during the optimization process. For the purpose of fitting B-spline curves, we consider the following non-periodic knot vectors:

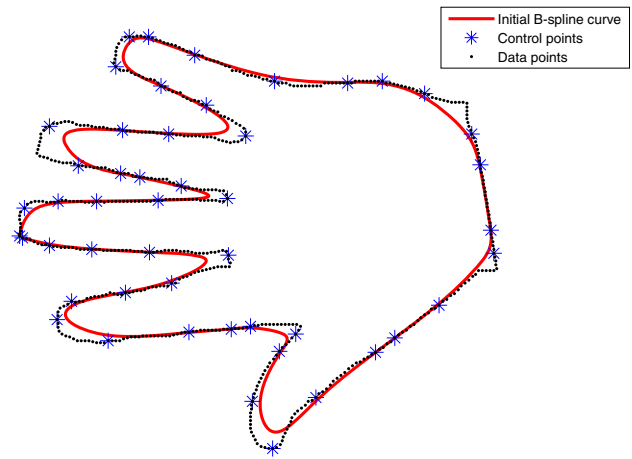
$$\begin{cases} u_0 = u_1 = \dots = u_{k-1} = 0, & u_{m_1+1} = u_{m_1+2} = \dots = u_{m_1+k} = 1, \\ u_{k+j} = \frac{j+1}{m_1-k+2}, & j = 0, 1, \dots, m_1-k. \end{cases}$$

The starting point parameterization is required to give a parameter  $\{t_j\}_{j=1}^{n_1}$  to each of the data points  $\{Q_j\}_{j=1}^{n_1}$  in the nonlinear least squares optimization problem (3). As follows, the starting parameters  $\{t_j\}_{j=1}^{n_1}$  for  $\{Q_j\}_{j=1}^{n_1}$  are assigned using the uniform parameterization method:

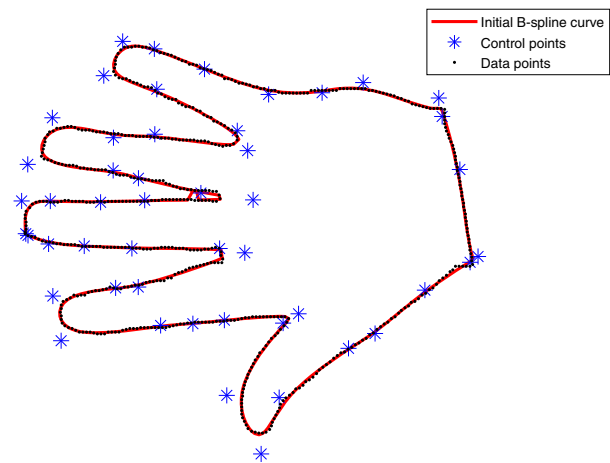
$$\begin{cases} t_0 = 0, & t_{n_1} = 1, \\ t_j = \frac{j}{n_1}, & j = 1, \dots, n_1 - 1. \end{cases}$$



**Fig. 5** Constructed B-spline curve for dataset in Example 2



(a) The initial parametric B-spline curve



(b) The Fitting curve

The fitting error of the  $i$ th iteration is computed in this study using the following equation:

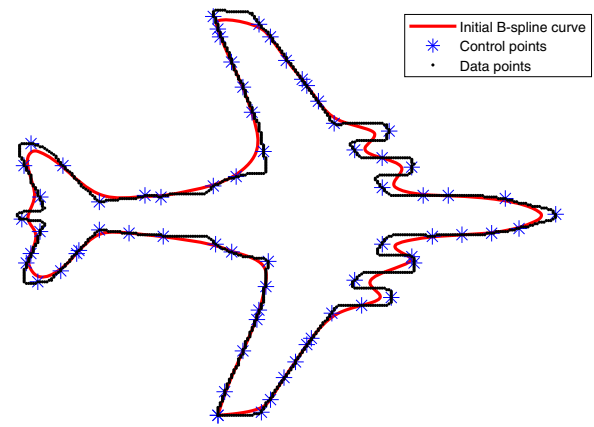
$$E_i = \left( \sum_{j=0}^{n_1} \frac{1}{n_1} \|C(t_j) - Q_j\|^2 \right)^{\frac{1}{2}}.$$

Following that, we evaluate the validity and effectiveness of the suggested method on both simulation and real-world data points. We explore with three different 2D shapes. The following examples feature highly complicated characteristics such as complex shapes, self intersections, a significant amount of data points and a high genus. The fitting curves are demonstrated in Figs. 4, 5 and 6, in which the data point and the primary B-spline curves are in (a) and the final fitting curves are shown in (b).

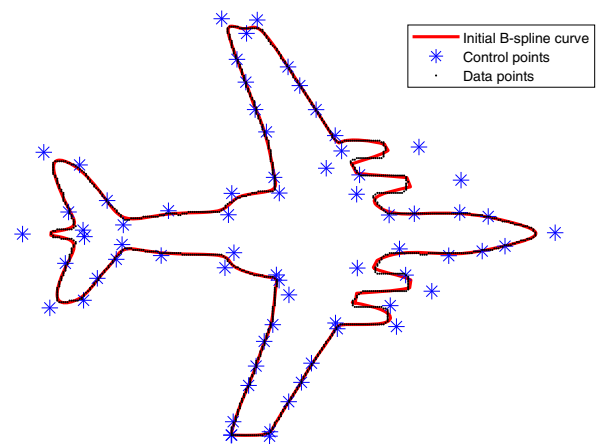
- **Example 1** (Shows in Fig. 4). 840 points are drawn from the “Leaf” dataset. In this example,  $L = 2$  is utilized as the length parameter.
- **Example 2** (Shows in Fig. 5). 620 points are drawn from the “Palm” dataset. In this example,  $L = 3$  is utilized as the length parameter.
- **Example 3** (Shows in Fig. 6). 1100 points are drawn from the “Airplane” dataset. In this example,  $L = 4$  is utilized as the length parameter.

The comparison of the quantity of the control points, starting and final errors, and time elapsed to construct the initial curve (in seconds) for Examples 1–3 created by our algorithm with those obtained by procedure of [47] is given in Table 2.

**Fig. 6** Constructed B-spline curve for dataset in Example 3



**(a)** The initial parametric B-spline curve



**(b)** The Fitting curve

## 7 Analysis of the results

The outcomes of the proposed approach and the factors that affected them are discussed in this section. Additionally, we consider how this study could be used in the actual world.

Despite the complexity and high number of points in the shapes, as shown in Figs. 4, 5 and 6 (a), the suggested approach generates an appropriate initial curve that is extremely near to the data points. Furthermore, the final B-spline curve produced by the initial curve and optimization method 1, as shown in Figs. 4, 5 and 6 (b), closely matches the shape. This illustrates that the optimization algorithm with the suggested starting point correctly solves the nonlinear least squares problem 3.

The numerical results in Table 2 demonstrate that the suggested approach outperforms the method in [47] in all aspects, including the quantity of control points, the fitting error and the amount of time it takes to build the initial curve.

For all test problems, the suggested method decreases the number of control points on primary B-spline curve. This parametric curve with fewer control points performs faster and requires less memory to store control points. However, even though our approach uses fewer control points, the initial and final fitting errors are lower than those of method [47], indicating that the location and number of control points are correctly set. In addition, our approach produces the initial curve in a shorter amount of time, which is very essential for practical applications.

The following are the major features of our approach based on observations and numerical results:

- The subject of fitting B-spline curves is expressed as a standard nonlinear least squares problem.
- Our approach provides an initial estimate for the location and quantity of the control points.
- The suggested technique produces a primary curve that is really similar to the shape of the desired.
- Because the suggested approach does not utilize any minimization methods or complicated computations to create the initial form, it consumes a relatively tiny portion of the computing time.

**Table 2** Experimental results of B-spline curve fitting for Examples 1–3

| Examples  | Methods     | Number control point | Start error | Final error | Time |
|-----------|-------------|----------------------|-------------|-------------|------|
| Example 1 | Our method  | 36                   | 0.7462      | 0.0170      | 0.21 |
|           | Method [47] | 42                   | 0.7506      | 0.0173      | 0.29 |
| Example 2 | Our method  | 49                   | 1.2755      | 0.0451      | 0.16 |
|           | Method [47] | 53                   | 1.2862      | 0.0510      | 0.26 |
| Example 3 | Our method  | 70                   | 0.4506      | 0.0216      | 0.36 |
|           | Method [47] | 77                   | 0.4525      | 0.0221      | 0.45 |

- In order to increase the accuracy of the initial B-spline curve, the length parameter ( $L$ ) might be used.
- The diagonal approximation BFGS technique, which is utilized in this study, uses the initial B-spline curve as a starting point and simultaneously optimizes location parameters and control points.
- The diagonal approximation BFGS technique has a space and time complexity of  $O(n)$ , which makes it suitable for large-scale optimization problem.
- A comparison study with method in [47] demonstrates that the suggested method, which uses fewer control points, generates a more precise B-spline curve in a shorter amount of time.

## 8 Conclusion and future work

The subject of fitting B-spline curve to a dataset of two-dimensional points is investigated in this research. The curve fitting problem is rewritten as an unconstructed optimization problem in the first phase. The number and location of control points are computed in the second phase, which is the main phase of the research, as a practical matter of which the primary B-spline curve is as near to the intended shape as possible. The last phase utilizes the diagonal approximation BFGS optimization approach to find the precise positions of the location parameters and control points simultaneously. The main advantage of diagonal approximation BFGS is that the complexity of each step is  $O(n)$  and it only needs  $O(n)$  memory allocations. Our method performs well for a variety of scenarios, including open, semi-closed and closed curves. The numerical examples demonstrate that the suggested approach produces excellent results even when challenging factors such as complex shapes, singularities, a nonzero genus and concavities are present. All of the examples show that we have a better approach than the previous method in terms of the number of control points, the fitting error and the execution time. Indeed, the suggested approach with fewer control points results in a more precise B-spline curve for data point fitting. This technology will also be used to fit data from physical experiments in the future. Our subsequent research will focus on expanding the proposed method for creating the initial B-spline surface.

## References

1. G.E. Farin, G. Farin, *Curves and Surfaces for CAGD: A Practical Guide* (Morgan Kaufmann, Burlington, MA, 2002)
2. J. Gallier, J.H. Gallier, *Curves and Surfaces in Geometric Modeling: Theory and Algorithms* (Morgan Kaufmann, Burlington, MA, 2000)
3. M. Sarfraz, *Interactive Curve Modeling* (Springer, Berlin, 2008)
4. A. Ebrahimi, G. Barid Loghmani, M. Sarfraz, Capturing outlines of planar generic images by simultaneous curve fitting and sub-division. *J. AI Data Min.* **8**(1), 105–118 (2020)
5. L. Piegl, W. Tiller, *The NURBS book*. [SI]: Springer Science & Business Media. Citado **6**, 34–39 (2012)
6. H. Bachau, E. Cormier, P. Declava, J. Hansen, F. Martín, Applications of B-splines in atomic and molecular physics. *Rep. Prog. Phys.* **64**(12), 1815 (2001)
7. S. Cipolla, C. Di Fiore, F. Tudisco, P. Zellini, Adaptive matrix algebras in unconstrained minimization. *Linear Algebra Appl.* **471**, 544–568 (2015)
8. M.G. Cox, *Algorithms for spline curves and surfaces*. NPL Report (1990)
9. R. Franke, L.L. Schumaker, A bibliography of multivariate approximation, in *Topics in Multivariate Approximation*. ed. by C.K. Chui, L.L. Schumaker, F.I. Utreras (Elsevier, Amsterdam, 1987), pp. 275–335
10. P. Bergström, O. Edlund, I. Söderkvist, Efficient computation of the Gauss–Newton direction when fitting NURBS using ODR. *BIT Numer. Math.* **52**(3), 571–588 (2012)
11. W. Wang, H. Pottmann, Y. Liu, Fitting B-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph. (ToG)* **25**(2), 214–238 (2006)
12. N.R. Draper, H. Smith, *Applied Regression Analysis*, vol. 326 (Wiley, Hoboken, 1998)
13. M. Sarfraz, S. Samreen, M.Z. Hussain, A quadratic trigonometric weighted spline with local support basis functions. *Alex. Eng. J.* **57**(2), 1041–1049 (2018)
14. E. Zieniuk, M. Kapturczak, Modeling the shape of boundary using NURBS curves directly in modified boundary integral equations for Laplace’s equation. *Comput. Appl. Math.* **37**(4), 4835–4855 (2018)
15. C. Kee, S. Lee, B-spline scale-space of spline curves and surfaces. *Comput.-Aided Des.* **44**(4), 275–288 (2012)
16. X. Li, T.W. Sederberg, S-splines: a simple surface solution for IGA and CAD. *Comput. Methods Appl. Mech. Eng.* **350**, 664–678 (2019)

17. A. Ebrahimi, G.B. Loghmani, A composite iterative procedure with fast convergence rate for the progressive-iteration approximation of curves. *J. Comput. Appl. Math.* **359**, 1–15 (2019)
18. S. Kirmani, M. Bilal Riaz et al., Shape preserving fractional order KNR C1 cubic spline. *Eur. Phys. J. Plus* **134**(7), 1–8 (2019)
19. M. Sarfraz, S.A. Raza, Capturing outline of fonts using genetic algorithm and splines. In: *Proceedings Fifth International Conference on Information Visualisation. IEEE*; (2001), p. 738–743
20. M. Sarfraz, F. Razzak, A Web based system to capture outlines of Arabic fonts. *Inf. Sci.* **150**(3–4), 177–193 (2003)
21. M. Sarfraz, M. Razzak, An algorithm for automatic capturing of the font outlines. *Comput. Graph.* **26**(5), 795–804 (2002)
22. A. Ebrahimi, G. Barid Loghmani, M. Sarfraz, Capturing outlines of generic shapes with cubic Bézier curves using the Nelder–Mead simplex method. *Iran. J. Numer. Anal. Optim.* **9**(2), 103–121 (2019)
23. A. Masood, S. Ejaz, Optimal curve fitting approach to represent outlines of 2D shapes. *Imaging Sci. J.* **58**(6), 308–320 (2010)
24. A. Masood, M. Sarfraz, Capturing outlines of 2D objects with Bézier cubic approximation. *Image Vis. Comput.* **27**(6), 704–712 (2009)
25. S. Abbas, M.Z. Hussain, M. Irshad, Image interpolation by rational ball cubic B-spline representation and genetic algorithm. *Alex. Eng. J.* **57**(2), 931–937 (2018)
26. S. Biswas, One-dimensional B–B polynomial and Hilbert scan for graylevel image coding. *Pattern Recognit.* **37**(4), 789–800 (2004)
27. S. Biswas, B.C. Lovell, *Bézier and Splines in Image Processing and Machine Vision* (Springer, Berlin, 2007)
28. M.Z. Hussain, S. Abbas, M. Irshad, Quadratic trigonometric B-spline for image interpolation using GA. *PLoS ONE* **12**(6), e0179721 (2017)
29. A. Majeed, Z.R. Yahya, J.Y. Abdullah, M. Rafique et al., Construction of occipital bone fracture using B-spline curves. *Comput. Appl. Math.* **37**(3), 2877–2896 (2018)
30. M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, et al. The digital Michelangelo project: 3D scanning of large statues. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*; (2000), p. 131–144
31. M.A. Khan, A new method for video data compression by quadratic Bézier curve fitting. *Signal Image Video Process.* **6**(1), 19–24 (2012)
32. M.A. Khan, An efficient algorithm for compression of motion capture signal using multidimensional quadratic Bézier curve break-and-fit method. *Multidimens. Syst. Signal Process.* **27**(1), 121–143 (2016)
33. M.A. Khan, Y. Ohno, Compression of video data using parametric line and natural cubic spline block level approximation. *IEICE Trans. Inf. Syst.* **90**(5), 844–850 (2007)
34. M. Ebadi, A. Ebrahimi, Video data compression by progressive iterative approximation. *Int. J. Interact. Multimedia Artif. Intell.* **6**(6), 189–196 (2021)
35. D. Salomon, *Curves and Surfaces for Computer Graphics* (Springer, Berlin, 2007)
36. N.M. Patrikalakis, T. Maekawa, *Shape Interrogation for Computer Aided Design and Manufacturing*, vol. 15 (Springer, Berlin, 2002)
37. F. Pérez-Arribas, I. Castañeda-Sabadell, Automatic modelling of airfoil data points. *Aerosp. Sci. Technol.* **55**, 449–457 (2016)
38. J. Rossignac, P. Borrel, Multi-resolution 3D approximations for rendering complex scenes, in *Modeling in Computer Graphics*, ed. by B. Falcidieno, T.L. Kunii (Springer, Berlin, 1993), pp. 455–465
39. M. Leu, X. Peng, W. Zhang, Surface reconstruction for interactive modeling of freeform solids by virtual sculpting. *CIRP Ann.* **54**(1), 131–134 (2005)
40. H. Yang, W. Wang, J. Sun, Control point adjustment for B-spline curve approximation. *Comput.-Aided Des.* **36**(7), 639–652 (2004)
41. H. Park, J.H. Lee, B-spline curve fitting based on adaptive curve refinement using dominant points. *Comput.-Aided Des.* **39**(6), 439–451 (2007)
42. N. Carlson, M. Gulliksson, S. Kartalopoulos, A. Buikis, N. Mastorakis, L. Vladareanu, Surface fitting with NURBS—a Gauss Newton with trust region approach. In: *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering* (No. 13). WSEAS; (2008)
43. P. Bergström, I. Söderkvist, Fitting NURBS using separable least squares techniques. *Int. J. Math. Model. Numer. Optim.* **3**(4), 319–334 (2012)
44. T. Speer, M. Kuppe, J. Hoschek, Global reparametrization for curve approximation. *Comput. Aided Geom. Des.* **15**(9), 869–877 (1998)
45. W. Zheng, P. Bo, Y. Liu, W. Wang, Fast B-spline curve fitting by L-BFGS. *Comput. Aided Geom. Des.* **29**(7), 448–462 (2012)
46. A. Ebrahimi, G.B. Loghmani, B-spline curve fitting by diagonal approximation BFGS methods. *Iran. J. Sci. Technol. Trans. A: Sci.* **43**(3), 947–958 (2019)
47. A. Ebrahimi, G.B. Loghmani, Shape modeling based on specifying the initial B-spline curve and scaled BFGS optimization method. *Multimedia Tools Appl.* **77**(23), 30331–30351 (2018)
48. A. Gálvez, A. Iglesias, Efficient particle swarm optimization approach for data fitting with free knot B-splines. *Comput.-Aided Des.* **43**(12), 1683–1692 (2011)
49. A. Gálvez, A. Iglesias, Particle-based meta-model for continuous breakpoint optimization in smooth local-support curve fitting. *Appl. Math. Comput.* **275**, 195–212 (2016)
50. A.Y. Hasegawa, C. Tormena, R.S. Parpinelli, Bézier curve parametrization using a multiobjective evolutionary algorithm. *Int. J. Comput. Sci. Appl.* **11**(2), 1–18 (2014)
51. A. Gálvez, A. Iglesias, Firefly algorithm for explicit B-spline curve fitting to data points. *Math. Probl. Eng.* **2013**, 1–12 (2013)
52. F. Javidrad, An accelerated simulated annealing method for B-spline curve fitting to strip-shaped scattered points. *Int. J. CAD/CAM* **12**(1), 9–19 (2012)
53. A. Iglesias, A. Gálvez, M. Collantes, Four adaptive memetic bat algorithm schemes for Bézier curve parameterization. In: *Transactions on Computational Science XXVIII*. Springer; p. 127–145 (2016)
54. X. Zhao, C. Zhang, B. Yang, P. Li, Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation. *Comput.-Aided Des.* **43**(6), 598–604 (2011)
55. J. Weber, T. Hansen, M. Van de Sanden, R. Engeln, B-spline parametrization of the dielectric function applied to spectroscopic ellipsometry on amorphous carbon. *J. Appl. Phys.* **106**(12), 123503 (2009)
56. Z. Meng-Hua, L. Liang-Gang, Q. Dong-Xu, Y. Zhong, X. Ao-Ao, Least square fitting of low resolution gamma ray spectra with cubic B-spline basis functions. *Chin. Phys. C* **33**(1), 24 (2009)
57. X. Zhang, C. Yang, B-spline function modeling of electric heating flow regulating valve. In: *Journal of Physics: Conference Series*, vol. 1748. IOP Publishing; p. 052036 (2021)
58. M. Wang, B. Tian, C.C. Hu, S.H. Liu, Generalized Darboux transformation, solitonic interactions and bound states for a coupled fourth-order nonlinear Schrödinger system in a birefringent optical fiber. *Appl. Math. Lett.* **119**, 106936 (2021)
59. Y. Shen, B. Tian, Bilinear auto-Bäcklund transformations and soliton solutions of a (3+1)-dimensional generalized nonlinear evolution equation for the shallow water waves. *Appl. Math. Lett.* **122**, 107301 (2021)
60. X.T. Gao, B. Tian, Y. Shen, C.H. Feng, Comment on “Shallow water in an open sea or a wide channel: Auto- and non-auto-Bäcklund transformations with solitons for a generalized (2+1)-dimensional dispersive long-wave system.”. *Chaos Solitons Fractals* **151**, 111222 (2021)
61. D.Y. Yang, B. Tian, Q.X. Qu, C.R. Zhang, S.S. Chen, C.C. Wei, Lax pair, conservation laws, Darboux transformation and localized waves of a variable-coefficient coupled Hirota system in an inhomogeneous optical fiber. *Chaos Solitons Fractals* **150**, 110487 (2021)
62. X.Y. Gao, Y.J. Guo, W.R. Shan, Optical waves/modes in a multicomponent inhomogeneous optical fiber via a three-coupled variable-coefficient nonlinear Schrödinger system. *Appl. Math. Lett.* **120**, 107161 (2021)

63. X.Y. Gao, Y.J. Guo, W.R. Shan, Bilinear forms through the binary Bell polynomials, N solitons and Bäcklund transformations of the Boussinesq–Burgers system for the shallow water waves in a lake or near an ocean beach. *Commun. Theor. Phys.* **72**(9), 095002 (2020)
64. X.Y. Gao, Y.J. Guo, W.R. Shan, Symbolic computation on a (2+1)-dimensional generalized variable-coefficient Boiti–Leon–Pempinelli system for the water waves. *Chaos Solitons Fractals* **150**, 111066 (2021)
65. X.Y. Gao, Y.J. Guo, W.R. Shan, Looking at an open sea via a generalized (2+1)-dimensional dispersive long-wave system for the shallow water: scaling transformations, hetero-Bäcklund transformations, bilinear forms and N solitons. *Eur. Phys. J. Plus* **136**(8), 1–9 (2021)
66. C. Di Fiore, S. Fanelli, F. Lepore, P. Zellini, Matrix algebras in quasi-Newton methods for unconstrained minimization. *Numer. Math.* **94**(3), 479–500 (2003)
67. J. Nocedal, S. Wright, *Numerical Optimization* (Springer, Berlin, 2006)
68. W. Sun, Y.X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, vol. 1 (Springer, Berlin, 2006)