P-ISSN 2586-9000
E-ISSN 2586-9027
Homepage : *https://tci-thaijo.org/index.php/SciTechAsia* Science & Technology Asia
Vol.28 No.1 January - March 2023 Page: [90-107]

Original research article

Progressive Iterative Approximation Method with Memory and Sequences of Weights for Least Square Curve Fitting

Saknarin Channark[1,2], Poom Kumam[1,2,3]* , Parin Chaipunya[1,2], Wachirapong Jirakitpuwapat[4]

[1]*Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand*
[2]*Center of Excellence in Theoretical and Computational Science, Science Laboratory Building, Faculty of Science, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand*
[3]*Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan*
[4]*National Security and Dual-Use Technology Center, National Science and Technology Development Agency, Pathum Thani 12120, Thailand*

Received 1 November 2021; Received in revised form -
Accepted 3 August 2022; Available online 20 March 2023

ABSTRACT

The progressive iterative approximation method with memory and sequences of weights for least square curve fitting (SSLSPIA) is presented in this paper. This method improves the MLSPIA method by varying the weights of the moving average between iterations, using three sequences of weights derived from the singular values of a collocation matrix. It is proved that a sequence of fitting curves with an appropriate alternative of weights converge to the solution of least square fitting and that the convergence rate of the new method is faster than that of the MLSPIA method. Some examples and applications in this paper prove the SSLSPIA method is superior.

Keywords: Least square curve fitting; Progressive iterative approximation; Sequences of weights

1. Introduction

Least-squares fitting (LSF) to data points is defined using a parametric curve which is a commonly used method in scientific and engineering research, involving geometric modeling and Computer Aided Design (CAD) [1–3]. It is widely used inside numerous applications. Due to its ra-

***Corresponding author:** poom.kum@kmutt.ac.th **doi:** xx.xxxxx/scitechasia.xxxx.xx

pidity, it's especially useful when dealing with enormous data sets. LSF can be divided into two types which are derived from several metric descriptions. Control points and parameter knots are variables in one, and the problems are addressed using nonlinear least squares. Due to the nonlinear nature of them, a sufficient initial fitting is necessary. The second is data collection that has been organized by giving each data point a parameter. The control points can then be obtained by solving the linear LSF. This may be a simpler construction method because it is linear. Details and examples are provided in [1, 2, 4–21] and references therein.

Smoothness and evenness are both enhanced with the fitting curves and surfaces, according to user-defined parameters (occasionally regarded, [2,5,6,12,22]). Although there is no ideal alternative to these user-defined settings, the smoothness and precision of the fits may be impacted. Researchers devised a method for defining a relatively small amount of control points [9] in order to improve fitting efficiency. When examining linear LSF within a certain amount of control points, a value of parameters comes from each point of the data set, we will assume no fairness terms in this article. Standard fits to each type of data collection can obtain the control points by directly resolving consistent systems of linear equations. It is possible to collect and solve a new type of linear equation.

One of the LSF methods that solves a linear system is the MLSPIA method, which is a least square form of the PIA method [23] and LSPIA method [8] created directly in [24]. It constructs a sequence of converging curves to arrive at a least square fitting solution by repeatedly approximating control points and constructing a sequence of converging curves. It is a straightforward approach to obtain a suitable solution that maintains the properties by maintaining the shape. When an extra control point is introduced in [8, 13, 24], just the iterative formula has to be changed rather than developing a new linear system.

In this work, Algorithm 1 presents a progressive iterative approximation method with memory and sequences of weights for least square curve fitting (SSLSPIA) to improve convergence rate. The method frequently creates a sequence of curves by iteratively updating the control points and the sequence's weighted sums. The justification for naming this new technique "SSLSPIA" is detailed in section 3. The following are our contributions to the SSLSPIA method:

- Whether or not the collocation matrix has a deficient column rank, the SSLSPIA method is feasible.
- The SSLSPIA method has a higher convergence rate than the MLSPIA method when the collocation matrix has full column rank [24].
- The SSLSPIA method conserves the MLSPIA benefits, including the ability to choose a large amount of set points, and allowing the number of starting control points, the shape conserving property, and parallel execution.
- The SSLSPIA method uses semi-constant knots for fitting curve which differs from knots in the MLSPIA method.
- The MLSPIA method is a variant of the SSLSPIA method that uses constant weight sequences.

The following is how the article is organized. In Section 2, there is a brief overview of related works. Section 3 proposes an iterative format and a convergence analysis for the SSLSPIA method. Section 4 performs the implementation, which includes three examples, and Section 5 describes the SSLSPIA applications. Finally, Section 6 expresses the conclusion.

2. Related work

Progressive Iterative Approximation (PIA) methods feature iterative forms similar to geometric interpolation (GI) methods, as illustrated in [25, 26], and other examples. The PIA methods rely on parametric distance identically to the GI methods. The initial discoveries were made in the 1970s by de Boor [27, 28], and Yamaguji [29] . Each version of the PIA method creates a sequence of curves by adjusting parameters and directly solving a linear system.

Assume that $\{Q_j\}_{j=1}^m$ is an ordered point set for fitted and $\{0 = t_1 < t_2 < t_3 < \cdots < t_m = 1\}$ is an increasing sequence parameters of $\{Q_j\}_{j=1}^m$. At the beginning of the iteration, we select $\{P_i^0\}_{i=1}^n$ from $\{Q_j\}_{j=1}^m$ to be the control point set and generate a segment of blending curve $C^0(t)$,i.e.,

$$C^0(t) = \sum_{i=1}^{n} B_i(t)P_i^0, \qquad t \in [t_1, t_m], \tag{2.1}$$

where $\{B_i(t)\}_{i=1}^n$ is a normalized totally positive (NTP) that is used as the blending base for real functions.

We know that if $B_i(t) \geq 0, i \in \{1, 2, 3, \dots, n\}$ and $\sum_{i=1}^n B_i(t) = 1$, the basis is NTP. For each t in $[0, 1]$, hold. Any non-decreasing series of collocation matrix is a totally positive matrix, meaning that every one of the minors in the matrix is non-negative [30, 31]. On $0 = t_1 < t_2 < t_3 < \cdots < t_m = 1$, the collocation matrix of the NTP blending basis is

$$B_{m\times n} = \begin{pmatrix} B_1(t_1) & B_2(t_1) & \dots & B_n(t_1) \\ B_1(t_2) & B_2(t_2) & \dots & B_n(t_2) \\ \ddots & \ddots & \dots & \ddots \\ B_1(t_m) & B_2(t_m) & \dots & B_n(t_m) \end{pmatrix}, \tag{2.2}$$

With a point set $\{Q_j\}_{j=1}^m$ and starting control point set $\{P_i^0\}_{i=1}^n$, the PIA method assigned in [23], the LSPIA method in [8] and the MLSPIA method in [24] approximate the curve with the NTP basis by the following curves repetitively:

$$C^{k+1}(t) = \sum_{i=1}^{n} B_i(t)P_i^{k+1}, \qquad t \in [0, 1],\ k \geq 0, \tag{2.3}$$

the control points are adjusted

$$P_i^{k+1} = P_i^k + \Delta_i^k, \qquad i \in \{1, 2, 3, \dots, n\},\ k \geq 0, \tag{2.4}$$

where adjusting the vector

$$\Delta_i^k = \begin{cases} Q_i - C^k(t_i), & \text{for PIA method,} \\ \mu \sum_{j=1}^m B_i(t_j)(Q_j - C^k(t_j)), & \text{for LSPIA method,} \end{cases}$$

μ is a constant, $i \in \{1, 2, 3, \dots, n\}$, $k \geq 0$, and adjusting the vector for MLSPIA method

$$\begin{cases} \Delta_i^0 = \nu \sum_{j=1}^m B_i(t_j)\Phi_j^0, & \\ \Delta_i^k = (1-\omega)\Delta_i^{k-1} + \gamma\delta_i^k + (\omega-\gamma)\delta_i^{k-1}, & k \geq 1, \\ \delta_i^k = \nu \sum_{j=1}^m B_i(t_j)(Q_j - C^k(t_j)), & k \geq 0, \end{cases}$$

by Φ_j^0 is a vector where $\{Q_j\}_{j=1}^m$ is located for every $j \in \{1, 2, 3, \dots, m\}$, and ω, γ, ν are real weights.

The PIA method in the format of Eqs. (2.3)-(2.4), described and originally labeled in [32], depends on the notion of benefit and loss improvement as proposed in [27, 28]., and is based on the non-uniform B-spline foundation. Of [33], the PIA approach in [23] evolved into a weighted PIA method, which accelerates convergence.

In the conventional PIA format, however, the amount of control points is equal to the amount of input points. This is not feasible when the amount of input points is really large. A new extended PIA (EPIA) format was introduced in [34], in which the amount of control points is less than the amount of input points. Because of its local property and parallel processing capabilities, the extended PIA is an ideal approach for fitting an extensive amount of data.

In addition, [35] has been expanded to approximate NURBS curves. According to the local property of the PIA method [36], if a subset of the control points is repeatedly updated, only the limit curve interpolates the subset of input points, leaving the others unchanged. The PIA methods converge with appropriate weights and have the properties of convexity conserving, as well as the evident exposition of curves, locality, and adaptivity. By altering only the control points which affect this segment to lower-cost [23, 32], the locality and adaptivity ensure improved probability that the resultant curve segment approximation is accurate. More details are shown in [13].

The LSPIA method assigned in [8] is primarily designed for large sets of data and inherits the benefits of PIA methods in the form of Eqs. (2.3)-(2.4). The LSPIA method has two distinct advantages. For starters, the LSPIA method can fit much larger sets quickly and reliably. Second, when using the LSPIA method for incremental data fitting, a new set of repetitively may be begun from the initial result of the previous output, saving a significant processing. T-splines in [7] are used to construct it. Then, in [10], replace the B-spline basis with the generalized B-spline basis to get the weighted LSF curve. In the nonsingular situation, the LSPIA method [8] converges, whereas [11] proves convergence in the singular case.

The PIA method with memory for least square fitting (MLSPIA) is one (LSF) method which is ineffectual for solving linear systems, a least square version of PIA method [23], and the LSPIA method [8] created directly in [24] to enhance convergence rate. It repeatedly estimates control points and creates a sequence of converging curves to arrive at a least square fitting solution. It is a way to obtain a suitable solution that maintains the properties by maintaining the form.The construction of the MLSPIA method in the format of Eqs. (2.3)-(2.4), assigned in [8], is analogous to the LSPIA method in the format of Eqs. (2.3)-(2.4), assigned in [8], except for the difference of Δ_i^k for each possible i and k, which is similar to the LSPIA method. For appropriate weights ω, ν, γ, it also converged to the LSF curve of $\{Q_j\}_{j=1}^m$.

In this article, we will compare the MLSPIA method from the difference of three real weights to the difference of three real sequences of weights. It will converge to the LSF curve as well. Furthermore, for the cubic B-spline fitting curve, we use semi-constant knots, which is different from knots used in the LSPIA and MLSPIA methods.

3. The SSLSPIA method

We will present the SSLSPIA method, which is a progressive iterative approximation method for the LSF curve with memory and weight sequences, and analyze its convergence rate.

With the provided data point set $\{Q_j\}_{j=1}^m$ and starting control point set $\{P_i^0\}_{i=1}^n$, and $m > n$, the NTP basis $B_1(t), B_2(t), B_3(t), \ldots, B_n(t)$ and the sequence $0 = t_1 < t_2 < t_3 < \cdots < t_m = 1$, the PIA method [23], LSPIA method [8], and MLSPIA method [24] iteratively approximate the curve by Eqs. (2.3)-(2.4) in the vector space where $\{Q_j\}_{j=1}^m$ is located.

The SSLSPIA method iteratively approximates the curve using Eqs. (2.3)-(2.4) by resetting the control points with the adjusting vector again from the previous curve.

$$\begin{cases} \Delta_i^0 = \nu_k \sum_{j=1}^m B_i(t_j)\Phi_j^0, \\ \Delta_i^k = (1-\omega_k)\Delta_i^{k-1} + \gamma_k \delta_i^k + (\omega_k - \gamma_k)\delta_i^{k-1}, \ \ k \geq 1, \\ \delta_i^k = \nu_k \sum_{j=1}^m B_i(t_j)(Q_j - C^k(t_j)), \qquad k \geq 0, \end{cases} \tag{3.1}$$

where Φ_j^0 is a point in the vector space which $\{Q_j\}_{j=1}^m$ is located for every $j \in \{1,2,3,\ldots,m\}$, and $\omega_k, \gamma_k, \nu_k$ are real sequences of weights.

It is called the SSLSPIA method as its construction is expected to develop the difference of three real weight ω, γ, ν to the difference of three real sequences of weights $\omega_k, \gamma_k, \nu_k$ for each possible k, which is similar to the MLSPIA method defined in [24]. It will also converge to the LSF curve of $\{Q_j\}_{j=1}^m$ for suitable sequences of weights ω_k, ν_k, γ_k which we can discuss later. It is a memory method because when we use Eq. (3.1) to compute Δ_i^{k-1}, we have to gather and utilize $\Delta_i^{k-1}$and δ_i^{k-1} from the preceding curve, as well as compute δ_i^k for every $k \geq 1$ and $i \in \{1,2,3,\ldots,n\}$. Furthermore, as shown in section 4, we use semi-constant knots for the cubic B-spline fitting curve, which differ from knots used in the LSPIA and MLSPIA methods. In [24], the SSLSPIA method has a higher convergence rate than MLSPIA.

The equivalent formulation of the SSLSPIA methods developed by Eqs. (2.3)-(3.1) is first given. The proof following line by line of the proof of [24].

Lemma 3.1. *Let $\{P_i^0\}_{i=1}^n$ be an starting control point set and let $\{\Phi_j^0\}_{j=1}^m$ be a point set. The P_i^k, $i \in \{1,2,3,\ldots,n\}$, $k \geq 0$ constructed by Eqs. (2.4)-(3.1). They can be rewritten as*

$$P_i^{k+1} = P_i^k + \nu_k \sum_{j=1}^m B_i(t_j)\Phi_j^k, \ i \in \{1,2,3,\ldots,n\}, \ k \geq 0, \tag{3.2}$$

where

$$\Phi_j^{k+1} = (1-\omega_k)\Phi_j^k + \omega_k Q_j - \sum_{i=1}^n B_i(t_j)\left[\omega_k P_i^k + \gamma_k \nu_k \sum_{j1=1}^m B_i(t_{j1})\Phi_{j1}^k\right], k \geq 0. \tag{3.3}$$

Proof. It is simple to affirm that Eq. (3.2) contains for $k = 0$ via Eqs. (2.4)-(3.1). Assume that for some $k \geq 0$, Eq. (3.2) holds true. As a consequence, for each $i \in \{1,2,3,\ldots,n\}$,

$$\begin{aligned} \omega_k \delta_i^k + \gamma_k(\delta_i^{k+1} - \delta_i^k) &= \nu_k \omega_k \sum_{j=1}^m B_i(t_j)\left(Q_j - C^k(t_j)\right) \\ &\quad - \nu_k\gamma_k \sum_{j=1}^m B_i(t_j)\left(C^{k+1}(t_j) - C^k(t_j)\right) \\ &= \nu_k \omega_k \sum_{j=1}^m B_i(t_j)\left(Q_j - \sum_{i=1}^n B_i(t_j)P_i^k\right) \\ &\quad - \nu_k\gamma_k \sum_{j=1}^m B_i(t_j)\sum_{i=1}^n B_i(t_j)\left(P_i^{k+1} - P_i^k\right), \end{aligned}$$

holds by Eq. (3.1), we get by Eqs. (2.4)-(3.1) for every $i \in \{1,2,3,\ldots,n\}$ that

$$\begin{aligned} \Delta_i^{k+1} &= (1-\omega_k)\Delta_i^k + \omega_k\delta_i^k + \gamma_k(\delta_i^{k+1} - \delta_i^k) \\ &= (1-\omega_k)\left(P_i^{k+1} - P_i^k\right) \\ &\quad + \nu_k\omega_k \sum_{j=1}^m B_i(t_j)\left(Q_j - \sum_{i=1}^n B_i(t_j)P_i^k\right) \\ &\quad - \nu_k\gamma_k \sum_{j=1}^m B_i(t_j)\sum_{i=1}^n B_i(t_j)\left(P_i^{k+1} - P_i^k\right) \\ &= (1-\omega_k)\nu_k \sum_{j=1}^m B_i(t_j)\Phi_j^k \\ &\quad + \nu_k\omega_k \sum_{j=1}^m B_i(t_j)\left(Q_j - \sum_{i=1}^n B_i(t_j)P_i^k\right) \\ &\quad - \nu_k^2\gamma_k \sum_{j=1}^m B_i(t_j)\sum_{i=1}^n B_i(t_j)\sum_{j1=1}^n B_i(t_{j1})\Phi_{j1}^k \\ &= \nu_k \sum_{j=1}^m B_i(t_j)\Phi_j^{k+1}, \end{aligned}$$

if we give

$$\Phi_j^{k+1} = (1-\omega_k)\Phi_j^k + \omega_k Q_j - \sum_{i=1}^n B_i(t_i)\left[\omega_k P_i^k + \gamma_k\nu_k \sum_{j1=1}^m B_i(t_{j1})\Phi_{j1}^k\right].$$

It follows that, for $i \in \{1, 2, 3, \dots, n\}$,

$$P_i^{k+2} = P_i^{k+1} + \Delta_i^{k+1} = P_i^{k+1} + \nu_k \sum_{j=1}^{m} B_i(t_j)\Phi_j^{k+1},$$

which means that Eq. (3.2) holds for $k+1$. From mathematical induction, Eq. (3.2) holds for every $k \geq 0$. □

Denote

$$\begin{cases} \Phi_i^k = \left[\Phi_1^k, \Phi_2^k, \Phi_3^k, \dots, \Phi_m^k\right]^T, & k \geq 0, \\ P_i^k = \left[P_1^k, P_2^k, P_3^k, \dots, P_n^k\right]^T \end{cases} \tag{3.4}$$

and

$$Q_i = [Q_1, Q_2, Q_3, \dots, Q_m]^T, B = \left(B_i(t_j)\right)_{m\times n}, \tag{3.5}$$

where $P_i^k, i \in \{1, 2, 3, \dots, n\}$, and $\Phi_j^k, j \in \{1, 2, 3, \dots, m\}$, are assigned by Eqs. (3.2)-(3.3), respectively, for every $k \geq 0$. Therefore Eqs. (3.2)-(3.3) can be written as

$$\begin{cases} \Phi^{k+1} = \left[(1-\omega_k)I_m - \gamma_k\nu_k BB^T\right]^T \Phi^k + \omega_k(Q - BP^k), \\ P^{k+1} = \nu_k B^T \Phi^k + P^k, \quad k \geq 0, \end{cases} \tag{3.6}$$

or equivalent to,

$$\begin{pmatrix} \Phi^{k+1} \\ P^{k+1} \end{pmatrix} = H_{\omega_k,\gamma_k,\nu_k} \begin{pmatrix} \Phi^k \\ P^k \end{pmatrix} + C_{\omega_k,\gamma_k,\nu_k}, \quad k \geq 0, \tag{3.7}$$

where

$$H_{\omega_k,\gamma_k,\nu_k} = \begin{pmatrix} (1-\omega_k)I_m - \gamma_k\nu_k BB^T & -\omega_k B \\ \nu_k B^T & I_n \end{pmatrix}, \tag{3.8}$$

and

$$C_{\omega_k,\gamma_k,\nu_k} = \begin{pmatrix} \omega_k Q \\ 0 \end{pmatrix}.$$

The matrix iterative format can be thought of as Eq. (3.7). As a result, we analyze the convergence of $\begin{pmatrix} \Phi^k \\ P^k \end{pmatrix}$ provided by Eq. (3.7) from $\begin{pmatrix} \Phi^0 \\ P^0 \end{pmatrix}$ while considering the curves generated by the SSLSPIA method.

Assumption 1. *Assume that Eq.* (3.5) *assigns $m > n$ and B. The singular values of the matrix B are $rank(B) = r$ and $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_r > 0$. Suppose that*

$$0 < \omega_k < 2,\ \omega_k - \frac{\omega_k}{\sigma_1^2\nu_k} < \gamma_k < \frac{\omega_k}{2} - \frac{\omega_k - 2}{\sigma_1^2\nu_k},\ \nu_k > 0. \tag{3.9}$$

Lemma 3.2. *Under the Assumption 1, suppose that $\sum_r = diag(\sigma_1, \sigma_2, \sigma_2, \dots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_r > 0$. λ is an eigenvalue of*

$$\hat{H}_{\omega_k,\gamma_k,\nu_k} = \begin{pmatrix} (1-\omega_k)I_r - \gamma_k\nu_k \sum_r^2 & 0 & -\omega_k \sum_r \\ 0 & (1-\omega_k)I_{m-r} & 0 \\ \nu_k \sum_r & 0 & I_r \end{pmatrix}, \tag{3.10}$$

equivalent to $\lambda = 1 - \omega_k$ or λ is a solution of the one of the equations as follows

$$\lambda^2 + \left[\gamma_k\nu_k\sigma_i^2 - (2-\omega_k)\right]\lambda + \sigma_i^2\nu_k(\omega_k - \gamma_k) + 1 - \omega_k = 0, \tag{3.11}$$

where $i \in \{1, 2, 3, \dots, r\}$.

Proof. Because the matrices $H_{\omega_k,\gamma_k,\nu_k}$ and $\hat{H}_{\omega_k,\gamma_k,\nu_k}$ are orthogonally identical, their eigenvalues are the same. We consider that

$$\begin{aligned} &|\hat{H}_{\omega_k,\gamma_k,\nu_k} - \lambda I_{m+r}| \\ &= \begin{vmatrix} (1-\omega_k-\lambda)I_r - \gamma_k\nu_k \sum_r^2 & 0 & -\omega_k \sum_r \\ 0 & (1-\omega_k-\lambda)I_{m-r} & 0 \\ \nu_k \sum_r & 0 & (1-\lambda)I_r \end{vmatrix} \\ &= (1-\omega_k-\lambda)^{m-r} \\ &\quad \times \prod_{i=1}^{r} \left[(1-\lambda)(1-\omega_k-\lambda-\gamma_k\nu_k\sigma_i^2) + \omega_k\nu_k\sigma_i^2\right]. \end{aligned}$$

So λ is an eigenvalue of $\hat{H}_{\omega_k,\gamma_k,\nu_k}$ equivalent to $\lambda = 1 - \omega_k$ or

$$\prod_{i=1}^{r} \left[(1-\lambda)(1-\omega_k-\lambda-\gamma_k\nu_k\sigma_i^2) + \omega_k\nu_k\sigma_i^2\right] = 0.$$

Therefore, $\lambda = 1 - \omega_k$ or λ is a solution of the equations Eq. (3.11). □

Lemma 3.3. *[37] The solutions of the real quadratic equation $\lambda^2 - b\lambda + c = 0$ are smaller than unity in modulus, equivalent to $|c| < 1$ and $|b| < 1 + c$.*

Lemma 3.4. *Under the Assumption 1, $\rho(\hat{H}_{\omega_k,\gamma_k,\nu_k}) < 1$, where $\rho(\hat{H}_{\omega_k,\gamma_k,\nu_k})$ denotes the spectral radius of $H_{\omega_k,\gamma_k,\nu_k}$ assigned by Eq.* (3.10)*, equivalent to weights $\omega_k, \gamma_k, \nu_k$ satisfy Eq.* (3.9)*.*

Proof. Defined weights $\omega_k, \gamma_k, \nu_k$, by Lemma 3.3 unity in modulus are larger than all the solutions of equations in Eq. (3.11) if and only if

$$\begin{cases} |\sigma_i^2 \nu_k(\omega_k - \gamma_k) + 1 - \omega_k| < 1, \ i \in \{1, 2, 3, \dots, r\}, \\ |\gamma_k \nu_k \sigma_i^2 - (2 - \omega_k)| < 2 - \omega_k + \sigma_i^2 \nu_k(\omega_k - \gamma_k). \end{cases} \tag{3.12}$$

Then, by Lemma 3.2 $\rho(\hat{H}_{\omega_k,\gamma_k,\nu_k}) < 1$ if and only if $|\omega_k - 1| < 1$ and Eq. (3.11) holds. Or equivalently, equations in Eq. (3.11) are smaller than unity in modulus if and only if

$$\begin{cases} 0 < \omega_k < 2, \\ \sigma_i^2 \nu_k \omega_k > 0, \\ \omega_k - 2 < \sigma_i^2 \nu_k(\omega_k - \gamma_k) < \omega_k, \\ \sigma_i^2 \nu_k(\omega_k - 2\gamma_k) > 2(\omega_k - 2), \end{cases} \quad i \in \{1, 2, 3, \dots, r\}. \tag{3.13}$$

Eq. (3.9) is equivalent to Eq. (3.13) because $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \geq \dots \geq \sigma_r^2 > 0$. □

Theorem 3.5. *Assume that the Assumption 1. Then, for every starting control point set $\{P_i^0\}_{i=1}^n$ and every point set $\{\Phi_j^0\}_{j=1}^m$, the curve sequences which are created by the SSLSPIA method, assigned by Eq.* (2.3)*, Eq.* (2.4) *and Eq.* (3.1)*, with weights $\omega_k, \gamma_k, \nu_k$, converge to the LSF curve of the given $\{Q_j^0\}_{j=1}^m$.*

Proof. The rewritten format of the iteration matrix $H_{\omega_k,\gamma_k,\nu_k}$ assigned by Eq. (3.8). Assume that the singular value decomposition (SVD) of B is

$$B = U \begin{pmatrix} \sum_r & 0 \\ 0 & 0 \end{pmatrix} V^T, \tag{3.14}$$

for $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are the orthogonal matrices and $\sum_r = diag(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_r > 0$. Suppose

$$W = \begin{pmatrix} U & 0 \\ 0 & V \end{pmatrix}, \tag{3.15}$$

then $W \in \mathbb{R}^{(m+n)\times(m+n)}$ is the orthogonal matrix. Next,

$$U^T B V = \begin{pmatrix} \sum_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m\times n},$$

$$V^T B^T U = \begin{pmatrix} \sum_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{n\times m},$$

and

$$U^T B B^T U = \begin{pmatrix} \sum_r^2 & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m\times m},$$

it holds that

$$W^T H_{\omega_k,\gamma_k,\nu_k} W = \begin{pmatrix} \hat{H}_{\omega_k,\gamma_k,\nu_k} & 0 \\ 0 & I_{n-r} \end{pmatrix}, \tag{3.16}$$

where

$$\hat{H}_{\omega_k,\gamma_k,\nu_k} = \begin{pmatrix} (1-\omega_k)I_r - \gamma_k \nu_k \sum_r^2 & 0 & -\omega_k \sum_r \\ 0 & (1-\omega_k)I_{m-r} & 0 \\ \nu_k \sum_r & 0 & I_r \end{pmatrix}. \tag{3.17}$$

We focus on the convergence of the sequence $\{P^k\}_{k=0}^\infty$ because proving that the curve sequence $\{C^k(t)\}_{k=0}^\infty$ generated by the SSLSPIA method converges to a LSF curve is equivalent to proving that the repetitively sequence $\{P^k\}_{k=0}^\infty$ created by Eq. (3.6) converges to a solution of $B^T BX = B^T Q$, where B and Q are assigned by Eq. (3.5). For any weights ω_k, γ_k, ν_k satisfying Eq. (3.9), let $\alpha = Q - B\beta$, where β is the solution of $B^T BX = B^T Q$, then it can be simply affirmed that $(\alpha^T \beta^T)^T$ is the result of the equation

$$(I_{m+n} - H_{\omega_k,\gamma_k,\nu_k})(x^T y^T)^T = C_{\omega_k,\gamma_k,\nu_k}, \tag{3.18}$$

where $H_{\omega_k,\gamma_k,\nu_k}$ and $C_{\omega_k,\gamma_k,\nu_k}$, are assigned in Eq. (3.8). That means, the equation Eq. (3.18) is compatible. Assume $\begin{pmatrix} \Phi \\ P \end{pmatrix}$ be the result of the equation Eq. (3.18), i.e.,

$$(I_{m+n} - H_{\omega_k,\gamma_k,\nu_k}) \begin{pmatrix} \Phi \\ P \end{pmatrix} = C_{\omega_k,\gamma_k,\nu_k}. \tag{3.19}$$

Then, for all $k \geq 1$

$$\begin{pmatrix}\Phi^k - \Phi\\ P^k - P\end{pmatrix} = H_{\omega_k,\gamma_k,\nu_k}\begin{pmatrix}\Phi^{k-1} - \Phi\\ P^{k-1} - P\end{pmatrix}$$
$$= \ldots = H_{\omega_k,\gamma_k,\nu_k} \circ H_{\omega_{k-1},\gamma_{k-1},\nu_{k-1}}$$
$$\circ \ldots \circ H_{\omega_1,\gamma_1,\nu_1}\begin{pmatrix}\Phi^0 - \Phi\\ P^0 - P\end{pmatrix}. \quad (3.20)$$

The convergence rate of any method equivalent to Eq. (3.7) is usually defined using the spectral radius of the iteration matrix. When the spectral radius is small, the convergence rate is fast. From Lemma 3.4, Eq. (3.9) assures that $\rho(\hat{H}_{\omega_k,\gamma_k,\nu_k}) < 1$ for the weights chosen $\omega_k, \gamma_k, \nu_k$ and $\hat{H}_{\omega_k,\gamma_k,\nu_k}$ defined by Eq. (3.17), so $\lim_{k\to\infty}(H_{\omega_k,\gamma_k,\nu_k} \circ H_{\omega_{k-1},\gamma_{k-1},\nu_{k-1}} \circ \ldots \circ H_{\omega_1,\gamma_1,\nu_1}) = 0$. By Eq. (3.16), it holds that

$$\lim_{k\to\infty}(W^T H_{\omega_k,\gamma_k,\nu_k} \circ \ldots \circ H_{\omega_1,\gamma_1,\nu_1} W)$$
$$= \lim_{k\to\infty}\begin{pmatrix}\hat{H}_{\omega_k,\gamma_k,\nu_k} \circ \ldots \circ \hat{H}_{\omega_1,\gamma_1,\nu_1} & 0\\ 0 & I_{n-r}\end{pmatrix}$$
$$= \begin{pmatrix}0 & 0\\ 0 & I_{n-r}\end{pmatrix},$$

where W is assigned by Eq. (3.15). Consequently,

$$\lim_{k\to\infty}(W^T H_{\omega_k,\gamma_k,\nu_k} \circ \ldots \circ H_{\omega_1,\gamma_1,\nu_1} W)$$
$$= W\begin{pmatrix}0 & 0\\ 0 & I_{n-r}\end{pmatrix}W^T. \quad (3.21)$$

By Eq. (3.20), $\lim_{k\to\infty}\begin{pmatrix}\Phi^k - \Phi\\ P^k - P\end{pmatrix}$ exists. Therefore, $\lim_{k\to\infty}\begin{pmatrix}\Phi^k\\ P^k\end{pmatrix}$ exists, too. Assume $\begin{pmatrix}\Phi^\infty\\ P^\infty\end{pmatrix}$ be the limit of $\begin{pmatrix}\Phi^k\\ P^k\end{pmatrix}$ as $k \to \infty$, i.e., $\lim_{k\to\infty}\begin{pmatrix}\Phi^k\\ P^k\end{pmatrix} = \begin{pmatrix}\Phi^\infty\\ P^\infty\end{pmatrix}$, thus by Eq. (3.20) and Eq. (3.21),

$$\begin{pmatrix}\Phi^\infty\\ P^\infty\end{pmatrix} = \begin{pmatrix}\Phi\\ P\end{pmatrix} + \begin{pmatrix}\Phi^\infty - \Phi\\ P^\infty - P\end{pmatrix}$$
$$= \begin{pmatrix}\Phi\\ P\end{pmatrix} + W\begin{pmatrix}0 & 0\\ 0 & I_{n-r}\end{pmatrix}W^T\begin{pmatrix}\Phi^0 - \Phi\\ P^0 - P\end{pmatrix}, \quad (3.22)$$

holds as $k \to \infty$ in Eq. (3.20). As

$$(I_{m+n} - H_{\omega_k,\gamma_k,\nu_k})W\begin{pmatrix}0 & 0\\ 0 & I_{n-r}\end{pmatrix}$$
$$= W\begin{pmatrix}I_{m+n} - \hat{H}_{\omega_k,\gamma_k,\nu_k} & 0\\ 0 & 0\end{pmatrix}\begin{pmatrix}0 & 0\\ 0 & I_{n-r}\end{pmatrix} = 0, \quad (3.23)$$

from Eq. (3.16), we get by Eq. (3.19) that

$$(I_{m+n} - H_{\omega_k,\gamma_k,\nu_k})\begin{pmatrix}\Phi^\infty\\ P^\infty\end{pmatrix}$$
$$= (I_{m+n} - H_{\omega_k,\gamma_k,\nu_k})\begin{pmatrix}\Phi\\ P\end{pmatrix}$$
$$+ \left[(I_{m+n} - H_{\omega_k,\gamma_k,\nu_k})W\begin{pmatrix}0 & 0\\ 0 & I_{n-r}\end{pmatrix}\right]W^T\begin{pmatrix}\Phi^0 - \Phi\\ P^0 - P\end{pmatrix}$$
$$= C_{\omega_k,\gamma_k,\nu_k}. \quad (3.24)$$

Thus, $\begin{pmatrix}\Phi^\infty\\ P^\infty\end{pmatrix}$ is a solution of Eq. (3.18) too, and $\begin{pmatrix}\Phi^k\\ P^k\end{pmatrix}$ converges to a solution of Eq. (3.18). As $\omega\nu \neq 0$, from Eq. (3.24) we have

$$\begin{pmatrix}\omega_k\Phi^\infty + \gamma_k\nu_k BB^T\Phi^\infty + \omega_k BP^\infty\\ -\nu_k B^T\Phi^\infty\end{pmatrix}$$
$$= \begin{pmatrix}\omega_k I_m + \gamma_k\nu_k BB^T & \omega_k B\\ -\nu_k B^T & 0\end{pmatrix}\begin{pmatrix}\Phi^\infty\\ P^\infty\end{pmatrix}$$
$$= (I_{m+n} - H_{\omega_k,\gamma_k,\nu_k})\begin{pmatrix}\Phi^\infty\\ P^\infty\end{pmatrix}$$
$$= \begin{pmatrix}\omega_k Q_1\\ \vdots\\ \omega_k Q_m\\ 0\\ \vdots\\ 0\end{pmatrix}.$$

Thus,

$$\omega_k\Phi^\infty + \gamma_k\nu_k BB^T\Phi^\infty + \omega_k BP^\infty = \omega_k Q,$$

and

$$B^T\Phi^\infty = 0,$$

which implies

$$\omega_k(\Phi^\infty + BP^\infty - Q) = -\gamma_k\nu_k BB^T\Phi^\infty,$$

$$\Phi^\infty + BP^\infty - Q = 0,$$

it means that

$$\Phi^\infty + BP^\infty = Q.$$

By eliminating Φ^∞, we obtain

$$B^T BP^\infty = B^T Q,$$

so the sequence$\{P^k\}_{k=0}^\infty$ converges to the solution of a LSF. We have a convergence theorem for blending curve fitting based on the above discussion. □

Corollary 3.6. *Assume that Eq. (3.5) assigns $m > n$ and B. The singular values of the matrix B are $rank(B) = r$ and $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq \sigma_r > 0$.*

$$If \quad 0 < \omega_k < 2, \;\; 0 < \gamma_k < \frac{\omega_k}{2}, \;\; 0 < \nu_k < \frac{1}{\sigma_1^2}, \tag{3.25}$$

so, for every starting control point set $\{P_i^0\}_{i=1}^n$ and every point set $\{\Phi_j^0\}_{j=1}^m$, the curve sequences generated by the SSLSPIA method, assigned by Eqs. (2.3)-(2.4) and (3.1), with weights $\omega_k, \gamma_k, \nu_k$, converge to the LSF curve of the given $\{Q_j^0\}_{j=1}^m$.

Proof. Clearly Eq. (3.25) satisfies condition Eq. (3.9). □

We next summarize our algorithm in the matrix form, ready to be implemented. This matrix form is taken into account when we test our algorithm with examples in the next section. Since we focus on plane curves, each Q_j and P_i have a 2D representation $Q_j = (Q_j(1), Q_j(2))$ and $P_i^k = (P_i^k(1), P_i^k(2))$, respectively. We therefore adopt the following matrices

$$\mathrm{Q} := \begin{pmatrix} Q_1(1) & Q_1(2) \\ Q_2(1) & Q_2(2) \\ \vdots & \vdots \\ Q_m(1) & Q_m(2) \end{pmatrix} \text{ and } \mathrm{P}^k := \begin{pmatrix} P_1^k(1) & P_1^k(2) \\ P_2^k(1) & P_2^k(2) \\ \vdots & \vdots \\ P_n^k(1) & P_n^k(2) \end{pmatrix}.$$

4. Implementation and examples

4.1 Implementation

From MLSPIA method [24], given an ordered point set $\{Q_j\}_{j=1}^m$,we assign the parameters $\{t_j\}_{j=1}^m$ for $\{Q_j\}_{j=1}^m$ with the normalized accumulated chord parameterization method [8, 38], that is, $t_1 = 0$, $t_m = 1$,

$$t_j = t_{j-1} + \frac{\|Q_j - Q_{j-1}\|}{D}, \qquad j \in \{2, 3, 4 \ldots, m-1\}, \tag{4.1}$$

where $D = \sum_{j=2}^m \|Q_j - Q_{j-1}\|$ is the sum of chord length. In addition, the knots for the cubic B-spline fitting curve $C(t) = \sum_{i=1}^n B_{i,3}(t)P_i$, are defined as $\{0, 0, 0, 0, \bar{t}_4, \bar{t}_5, \ldots, \bar{t}_n, 1, 1, 1, 1\}$, where

$$\bar{t}_{i+3} = (1 - \alpha)t_{j-1} + \alpha t_j, \tag{4.2}$$

where $d = \frac{m}{n-2}$, $\alpha = id - i$, $j = \lfloor id \rfloor$ and $i \in \{1, 2, 3 \ldots, n-3\}$.

For SSLSPIA method, the semi-constant knots for the cubic B-spline fitting curve $C(t) = \sum_{i=1}^n B_{i,3}(t)P_i$ which is different from knots in MLSPIA method, are defined as

Algorithm 1: SSLSPIA

```
Input: Data matrix Q, initial control point
       matrix $P^0$ with $m > n$, initial guess
       matrix $\Phi^0$ of size $m \times 2$, acceptable
       error tol $> 0$, and maximum
       number of iterations MaxIter $> 0$.
Parameter: Weight sequences $\omega_k, \gamma_k, \nu_k$
Define the increasing
  sequence$0 = t_1 < t_2 < \cdots < t_m = 1$
Compute NTP basis matrix B as in Eq. (2.2)
Set an initial error $E^{-1}$ = Inf.
Step 0:
  $\delta^0 \leftarrow \nu_0 B^\top (Q - C^0)$;
  $\Delta^0 \leftarrow \nu_0 B^\top \Phi^0$;
  $P^1 \leftarrow P^0 + \Delta^0$;
  $C^1 \leftarrow BP^1$;
  $E^0 \leftarrow \|Q - C^1\|_F^2$;
while { $k = 1$ : MaxIter } do
    $\delta^k \leftarrow \nu_k B^\top (Q - C^k)$;
    $\Delta^k \leftarrow (1 - \omega_k)\Delta^{k-1} + \gamma_k \delta^k$
            $+(\omega_k - \gamma_k)\delta^{k-1}$;
    $P^{k+1} \leftarrow P^k + \Delta^k$;
    $C^{k+1} \leftarrow BP^{k+1}$;
    $E^k \leftarrow \|Q - C^{k+1}\|_F^2$;
    if $|E^k - E^{k-1}|$ < tol then
        STOP ;
    $k \leftarrow k + 1$;
Output: Control points $P^k$
```

$\{0, 0, 0, 0, \bar{t}_4, \bar{t}_5, \ldots, \bar{t}_n, 1, 1, 1, 1\}$, where,

$$\begin{aligned} t_i &= \bar{t}_i, \\ \bar{t}_{i+3} &= \bar{t}_{i+2} + R, \quad i \in \{1, 2, 3, \ldots, n-3\}, \\ R &= \frac{1}{n-2}. \end{aligned} \tag{4.3}$$

In the SSLSPIA method, we split t_i in the close interval related to the number of data points such that $\|t_{i-1} - t_i\| = \|t_i - t_{i+1}\|$. On the other hand, the MLSPIA method splits t_j in the close interval according to the distance between the data points.

In our execution, we choose the starting control points $\{P_i\}_{i=1}^n$ as $P_1 = Q_1,\ P_n = Q_m$,

$$P_i = Q_{f(i)}, \qquad i \in \{2, 3, 4 \ldots, n-1\} \tag{4.4}$$

where $f(i) = \lfloor \frac{(mi)}{n} \rfloor$.

4.2 Examples and weights

The following are three instances that demonstrate the effectiveness of the SSLSPIA method :

- Example 1: A subdivision curve generated using the incenter subdivision technique yielded 198 points (music notation).
- Example 2: $r = sin\frac{\theta}{4}$ ($\theta \in [0, 8\pi]$) is a sample of 402 points taken evenly from the polar coordinate equation.
- Example 3: 315 points with G-shape font characteristics measured and smoothed.

The spectral radius of the iteration matrix is commonly used to quantify the convergence rate of any method similar to Eq. (3.7). The convergence rate is quick when the spectral radius is small.

From the convergence rate of MLSPIA method for approximating curves at the weights $\omega = \omega^*$, $\gamma = \gamma^*$, $\nu = \nu^*$ is

$$\rho(\hat{H}_{\omega^*,\gamma^*,\nu^*}) = \frac{\sigma_1 - \sigma_r}{\sigma_1 + \sigma_r} \le \frac{\sigma_1^2 - \sigma_r^2}{\sigma_1^2 + \sigma_r^2} < 1.$$

ω^*, γ^*, and ν^* are assigned here by

$$\omega^* = \gamma^* = \frac{4\sigma_1\sigma_r}{(\sigma_1 + \sigma_r)^2}, \nu^* = \frac{1}{\sigma_1\sigma_r},$$

$\rho(\hat{H}_{\omega^*,\gamma^*,\nu^*})$ is the spectral radius of the matrix $\hat{H}_{\omega^*,\gamma^*,\nu^*}$, and σ_r and σ_1 are the smallest as well as the largest singular values of the matrix B, respectively.

With the same assumption of Theorem 3.5, we get the convergence rate of SSLSPIA method Eq. (3.6) for approximating curves at the weights $\omega_k = \omega_k^*$, $\gamma_k = \gamma_k^*$, $\nu = \nu_k^*$ is

$$\rho(\hat{H}_{\omega_k^*,\gamma_k^* t,\nu_k^*}) = \frac{\sigma_1 - \sigma_r}{\sigma_1 + \sigma_r} \le \frac{\sigma_1^2 - \sigma_r^2}{\sigma_1^2 + \sigma_r^2} < 1. \tag{4.5}$$

ω_k^*, γ_k^*, and ν_k^* are defined here by

$$\begin{aligned} &\omega_k^* = \frac{4\sigma_1\sigma_r}{(\sigma_1 + \sigma_r)^2}, \quad \nu_k^* = \frac{1}{\sigma_1\sigma_r}, \\ &\gamma_k^* = \left(\omega_k^* - \frac{\omega_k^*}{\sigma_1^2\nu_k^*}\right) \\ &+ \left[1 - \frac{7}{(2k)}\right]\left[\left(\frac{\omega_k^*}{2} - \frac{(\omega_k^* - 2)}{\sigma_1^2\nu_k^*}\right) - \left(\omega_k^* - \frac{\omega_k^*}{\sigma_1^2\nu_k^*}\right)\right]. \end{aligned} \tag{4.6}$$

$\rho(\hat{H}_{\omega_k^*,\gamma_k^*,\nu_k^*})$ is the spectral radius of the matrix $\hat{H}_{\omega_k^*,\gamma_k^*,\nu_k^*}$, and σ_r and σ_1 are the smallest and the largest singular values of the matrix B, respectively, as in Theorem 3.5.

Moreover from Corollary 3.6, we defined ω_k^*, γ_k^*, ν_k^* by

$$\begin{aligned} \omega_k^* &= 1.95,\ \nu_k^* = \frac{1}{\sigma_1^2} \times \left(1 - \frac{1}{10k}\right), \\ \gamma_k^* &= \frac{\omega_k^*}{2} \times \left(1 - \frac{1}{100k}\right). \end{aligned} \tag{4.7}$$

4.3 Numerical results

The iteration process is stopped for the comparison of the MLSPIA and SSLSPIA methods if $\|E_{k+1} - E_k\| < 10^{-7}$, where the fitting inaccuracy of the $k-$th iteration is computed by

$$E_k = \sum_{j=1}^{m} \|Q_j - \sum_{i=1}^{n} B_i(t_j)P_i^k\|^2, \ k \geq 0, \qquad (4.8)$$

and we compare E_k of each iteration to show the efficiency and validity of the SSLSPIA.

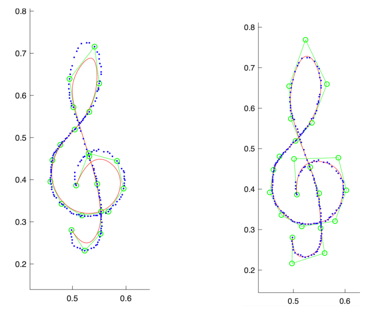
All the examples are operated on a laptop with a 2.4 GHz Quad-Core Intel Core i5 processor and 8 GB memory via MATLAB R2019a.

To consider the amount of starting control points for the SSLSPIA and MLSPIA methods, we first present the iteration numbers of three cases in Table 1. We can see that the iteration numbers do not rely on the amount of starting control points. It implies we may select any amount of control points that are appropriate for each data set. After that,then all of the starting control points in this experiment are chosen from $\lfloor \frac{m}{7} \rfloor$.

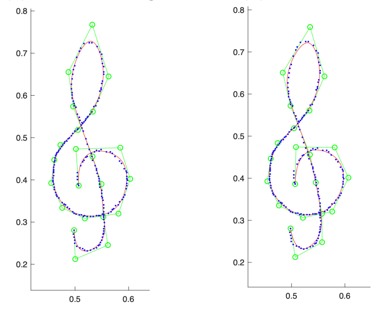
Table 1. The iteration numbers for a various amount of control points.

		IT				
Example	Method	$\lfloor \frac{m}{7} \rfloor$	$\lfloor \frac{m}{6} \rfloor$	$\lfloor \frac{m}{5} \rfloor$	$\lfloor \frac{m}{4} \rfloor$	$\lfloor \frac{m}{3} \rfloor$
1	MLSPIA	30	30	33	34	33
	SSLSPIA	24	25	27	28	29
2	MLSPIA	40	39	39	30	77
	SSLSPIA	38	33	37	28	30
3	MLSPIA	40	44	37	37	86
	SSLSPIA	37	32	35	35	40

In Example 1, 2, and 3, the cubic B-spline curves created by the MLSPIA method, SSLSPIA Eq. (4.6) method, and SSLSPIA Eq. (4.7) method at the initial fitting and fitting result are shown in Figs. 1,2, and 3, respectively. The blue points in each of the three instances represent known points that form a dotted limit curve to be fitted, green points represent control points gained at the appropriate step, and the red line shows the curve of each iteration.

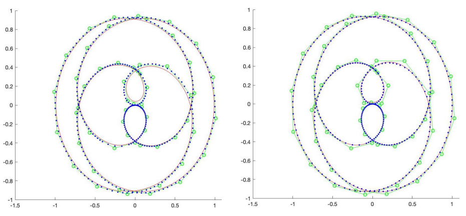


(a) Initial fitting **(b)** MLSPIA

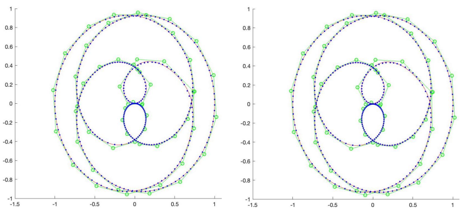


(c) SSLSPIA Eq. (4.6) **(d)** SSLSPIA Eq. (4.7)

Fig. 1. 198 points is fitted with 28 control points.

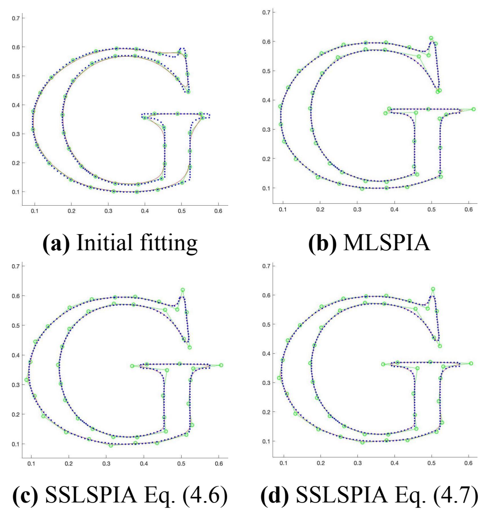


(a) Initial fitting **(b)** MLSPIA



(c) SSLSPIA Eq. (4.6) **(d)** SSLSPIA Eq. (4.7)

Fig. 2. 402 points is fitted with 57 control points.


(a) Initial fitting **(b)** MLSPIA

(c) SSLSPIA Eq. (4.6) **(d)** SSLSPIA Eq. (4.7)

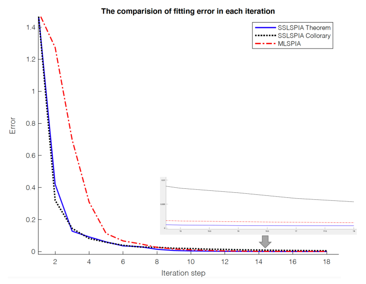
Fig. 3. 315 points is fitted with 45 control points.

In Figs. 4a-4c, we can see that the SSLSPIA Eq. (4.6) method has the fitting inaccuracy of the $k-$th iteration less than the MLSPIA method and the SSLSPIA Eq. (4.7) method, respectively.

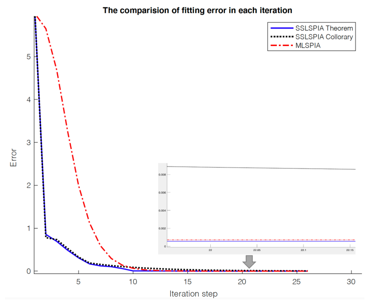
In Table 2 columns “IT” (iteration numbers), “CPU time(s)” (ten times the average amount of CPU times(s)) and “$\|E_{k+1} - E_k\|$” (the fitting error difference of the $k-$th and $(k+1)-$th iteration) for the MLSPIA and SSLSPIA methods are listed. We can observe that the SSLSPIA method Eq. (4.6) requires fewer iteration numbers, uses less CPU time, and has a minor fitting error difference between the $k-$th and $(k+1)-$th iteration. We can now conclude from the experiments that, at a significance level of 0.05, the SSLSPIA methodEq. (4.6) is faster than the MLSPIA method and the SSLSPIA method Eq. (4.7) in terms of convergence rate.

5. Applications

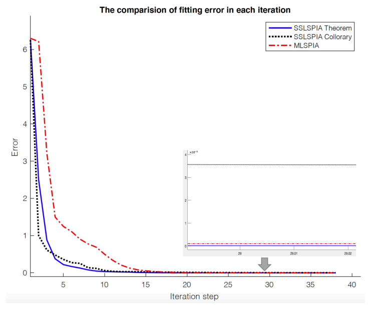
Traditional Thai Patterns, or Lai Thai, arose as a result of Buddhism’s influence. Originally, Thai master artists drew



(a) E_k of Example 1.



(b) E_k of Example 2.



(c) E_k of Example 3.

Fig. 4. The comparision of E_k of Example 1-3.

inspiration for their designs from candle flames, lotus blossoms, and jasmine garlands.Ancient masters were able to produce patterns that may be used to identify Thainess. The origins of Thai patterns may be traced back to the ancient Chiang Saen period. They then went through the Sukho Thai, Ayutthaya, and Rattanakosin eras in succession. Current Thai patterns are in sta-

Table 2. Comparision of IT, CPU time(s) and $\|E_{k+1} - E_k\|$.

Examples	Methods	IT	CPUtime(s)	$\|E_{k+1} - E_k\|$
1	MLSPIA	30	0.0065976	7.21051E-08
	SSLSPIA Eq. (4.6)	24	0.0038542	2.05646E-08
	SSLSPIA Eq. (4.7)	88	0.0068420	8.50730E-08
2	MLSPIA	40	0.0095162	9.15621E-08
	SSLSPIA Eq. (4.6)	38	0.0059756	6.45981E-08
	SSLSPIA Eq. (4.7)	191	0.0128047	9.88253E-08
3	MLSPIA	42	0.0065685	8.25362E-08
	SSLSPIA Eq. (4.6)	37	0.0046745	8.16037E-08
	SSLSPIA Eq. (4.7)	74	0.0072864	9.68625E-08

ble and flawless shapes from generations of refining and evolution of designs.

Ancient traditional Thai patterns in various locations in Thailand have become antiquated and fade away. Thus, these patterns deserve to be preserved and maintained in an image file format that can be expanded and relayed to younger generations for learning. However, the technicians nowadays copy the patterns by using stencil paper. This might cause damage to the prototype patterns and it takes a long time to copy. For this reason, our developed SSLSPIA Method can be effectively applied in this work as the following examples.

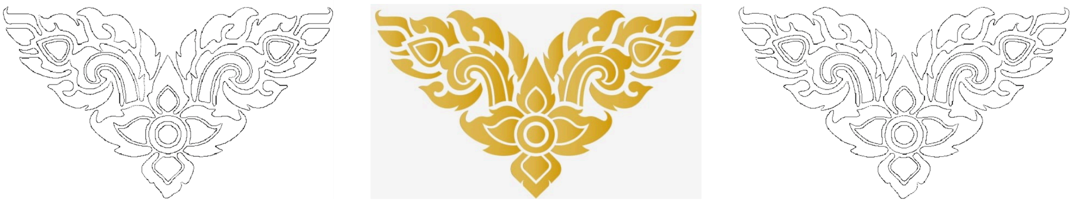
The procedure starts with inputting the image of Thai patterns and converting the image to black and white. After that, collect the coordinates at the edges of the image to be the data point sets $\{Q_j\}_{j=1}^{m}$ for the fitting curve. As we know, each Thai patterns image has many data point sets or lines. This means that we must fit the data point set $\{Q_j\}_{j=1}^{m}$ by using the MLSPIA method and the SSLPIA method line by line for all lines.

The amount of data points obtained in each line from the picture varies in this experiment. The data is large, and the image's coordinates round to an integer at the edges. It is for this reason that the total of starting control points is set to $max\left\{10, \lfloor \frac{mQ}{10} \rfloor\right\}$ with mQ being the amount of data points set to $\{Qj\}j = 1^m$ and the iteration process being stopped if $\|E_{k+1} - E_k\| < 10^{-7}$. We present three examples to illustrate the SSLSPIA Eq. (4.6) method's applications, as indicated in Section 3.

Traditional Thai Patterns (I), (II) and (III), generated by using the MLSPIA method and the SSLSPIA method are presented in Figs. 5-7 respectively.

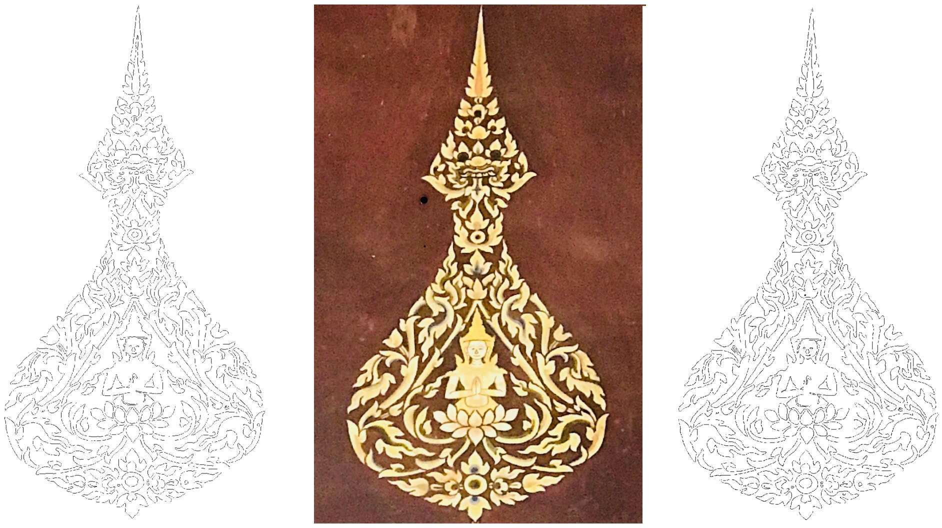
Then we compare E_k of Traditional Thai Patterns (I), (II) and (III) in Figs. 8a, 8b, and 8c respectively. It shows that the SSLSPIA method has the fitting inaccuracy of the $k-$th iteration less than the MLSPIA method, which means that the convergence speed of the SSLSPIA method is faster than the MLSPIA method.

In Table 3 columns "CPU time(s)" (ten times the average amount of CPU times(s)) "$\|E_{k+1} - E_k\|$" (the fitting error difference of the $k-$th and $(k + 1)$-th iteration) "PSNR" (Peak Signal-to-Noise Ratio) "SNR" (Signal-to-Noise Ratio) "SSIM" (Structural Similarity Index) and"RE" (Relative Entropy Index) [39] for the MLSPIA and SSLSPIA methods are listed. We can see that the SSLSPIA method uses less computer processing time than the MLSPIA method. The inaccuracy in fitting difference of the $k-$th and $(k + 1)-$th iterations



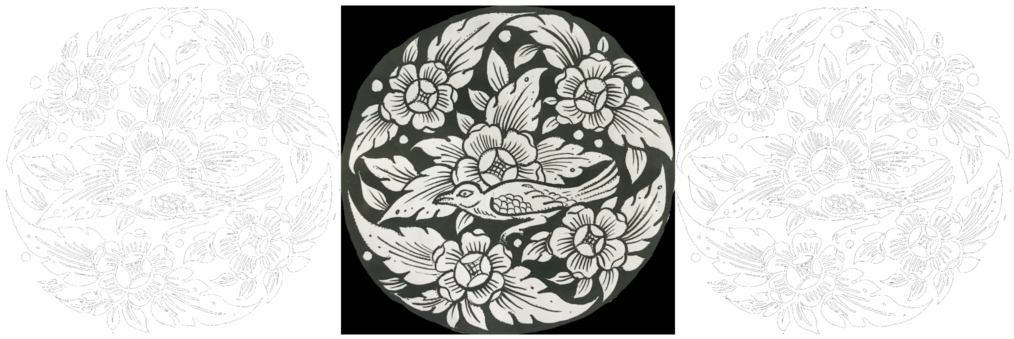
(a) Image via MLSPIA. **(b)** Original Image. **(c)** Image via SSLSPIA.

Fig. 5. Traditional Thai pattern (I) generated by using MLSPIA method and SSLSPIA method.



(a) Image via MLSPIA. **(b)** Original Image. **(c)** Image via SSLSPIA.

Fig. 6. Traditional Thai pattern (II) generated by using MLSPIA method and SSLPIA method.



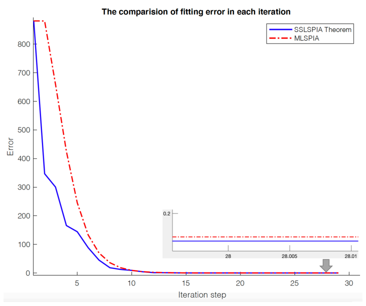
(a) Image via MLSPIA. **(b)** Original Image. **(c)** Image via SSLSPIA.

Fig. 7. Traditional Thai pattern (III) generated by using MLSPIA method and SSLPIA method.

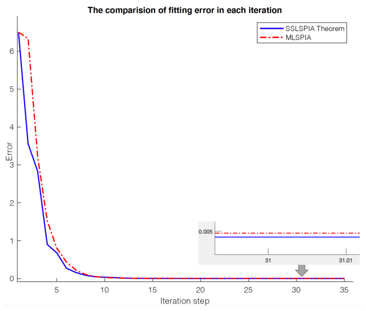
are too small. The PSNR, SNR, and SSIM values of the SSLSPIA method are higher than the MLSPIA method, and the RE value of the SSLSPIA method is less than that of the MLSPIA method. From the results, we can summarize that the convergence rate

Table 3. Comparision of IT, CPU time(s) , $\|E_{k+1} - E_k\|$, PSNR, SNR, SSIM and RE

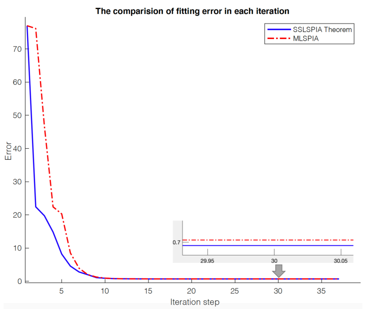
Patterns	Methods	CPU time(s)	$\|E_{k+1} - E_k\|$	PSNR	SNR	SSIM	RE
I (Fig 5)	MLSPIA	3.150	3.07029E-08	17.92859883	17.8638222	0.85980192	0.00722463
	SSLSPIA	3.031	2.61640E-08	17.97269449	17.8643835	0.86080893	0.00722380
II (Fig 6)	MLSPIA	12.268	8.07340E-08	15.73578343	15.5508787	0.77210345	0.01192810
	SSLSPIA	11.856	9.12400E-08	15.74444306	15.6542628	0.77251080	0.01190534
III(Fig 7)	MLSPIA	81.666	5.79711E-08	17.67461186	17.6133920	0.76816473	0.00765417
	SSLSPIA	72.977	9.20428E-08	17.68775476	17.6265349	0.83274903	0.00763077



(a) E_k of Thai Patterns (I).



(b) E_k of Thai Patterns (II).



(c) E_k of Thai Patterns (III).

Fig. 8. The comparision of E_k of Thai Patterns (I), (II) and (III).

of the SSLSPIA method is faster than the MLSPIA method at a significance level of 0.05. Moreover,the SSLSPIA method has 0.82 percent higher image quality than the MLSPIA method.

6. Conclusion

A discussion of the progressive iterative approximation method with memory and sequences of weights for least square curve fitting (SSLSPIA) is presented in this paper. The method improves the MLSPIA method by varying the weights of the moving average between iterations, using the three real sequences of weights derived from the singular values of the collocation matrix and uses semi-constant knots which differ from knots in the MLSPIA method. At a significance level of 0.05, it is shown that a sequence of fitting curves with appropriate weight alternatives would converge to the LSF solution, and that the SSLSPIA method's convergence rate is faster than the MLSPIA method described in [24]. Furthermore, the SSLSPIA method has a higher image quality than the MLSPIA method by 0.82 percent when applied to Traditional Thai pattern preservation in an image file.

Acknowledgements

The authors acknowledge the financial support provided by the Center of Excellence in Theoretical and Computational Science (TaCS-CoE), KMUTT. This research project is supported by Thailand Science Research and Innovation (TSRI) Basic Research Fund: Fiscal year 2021 under project number 64A306000005. Moreover, the first author was supported by the Petchra Pra Jom Klao Ph.D. Research Scholarship from King Mongkut's University of Technology Thonburi (Grant No. 54/2561). The guidance of Professor Sompong Dhompongsa is gratefully acknowledged, and thanks to Professor Zheng-Da Huang and Dr. Hui-Di Wang for the suggestion to the writing program.

References

[1] Piegl L, Tiller W. The NURBS Book (2nd Ed.). Berlin, Heidelberg: Springer-Verlag; 1997.

[2] Farin G, Hoschek J, Kim MS. Handbook of computer aided geometric design. North-Holland, Amsterdam; 2002.

[3] de Boor C, Rice J. Least Squares Cubic Spline Approximation I - Fixed Knots. 2001 12.

[4] Borges CF, Pastva T. Total least squares fitting of Bézier and B-spline curves to ordered data. Comput Aided Geom Design. 2002;19(4):275–289.

[5] Wang W, Pottmann H, Liu Y. Fitting B-Spline Curves to Point Clouds by Curvature-Based Squared Distance Minimization. ACM Trans Graph. 2006 Apr;25(2):214–238.

[6] Kineri Y, Wang M, Lin H, Maekawa T. B-spline surface fitting by iterative geometric interpolation/approximation algorithms. Computer-Aided Design. 2012 jul;44(7):697–708.

[7] Lin H, Zhang Z. An efficient method for fitting large data sets using T-splines. SIAM J Sci Comput. 2013;35(6):A3052–A3068.

[8] Deng C, Lin H. Progressive and iterative approximation for least squares B-spline curve and surface fitting. Comput-Aided Des. 2014;47:32–44.

[9] Gálvez A, Iglesias A, Avila A, Otero C, Arias R, Manchado C. Elitist Clonal Selection Algorithm for Optimal Choice of Free Knots in B-Spline Data Fitting. Appl Soft Comput. 2015 Jan;26(C):90–106.

[10] Zhang L, Ge X, Tan J. Least square geometric iterative fitting method for generalized B-spline curves with two different kinds of weights. The Visual Computer. 2015 11;32.

[11] Lin H, Cao Q, Zhang X. The Convergence of Least-Squares Progressive Iterative Approximation for Singular Least-Squares Fitting System. Journal of Systems Science and Complexity. 2018 dec;31(6):1618–1632.

[12] Ebrahimi A, Loghmani GB. Shape modeling based on specifying the initial B-spline curve and scaled BFGS optimization method. Multimedia Tools and Applications. 2018 may;77(23):30331–30351.

[13] Lin H, Maekawa T, Deng C. Survey on geometric iterative methods and their applications. Computer-Aided Design. 2018 feb;95:40–51.

[14] Liu M, Li B, Guo Q, Zhu C, Hu P, Shao Y. Progressive iterative approximation for regularized least square bivariate B-spline surface fitting. J Comput Appl Math. 2018;327:175–187.

[15] Chen J, Wang GJ. Progressive iterative approximation for triangular Bézier surfaces. Computer-Aided Design. 2011 aug;43(8):889–895.

[16] Hu Q. An iterative algorithm for polynomial approximation of rational triangular Bézier surfaces. Applied Mathematics and Computation. 2013 may;219(17):9308–9316.

[17] Pereyra V, Scherer G. Large scale least squares scattered data fitting. Applied Numerical Mathematics. 2003;44(1):225–239.

[18] Delgado J, Peña JM. Progressive iterative approximation and bases with the fastest convergence rates. Computer Aided Geometric Design. 2007;24(1):10–18.

[19] Shaohui D, Guozhao W. Numerical Analysis of the Progressive Iterative Approximation Method. Journal of Computer-Aided Design & Computer Graphics. 2012;7.

[20] Lin H. Adaptive data fitting by the progressive-iterative approximation. Computer aided geometric design. 2012;29(7):463–473.

[21] Chen J, Wang G, Jin C. Two kinds of generalized progressive iterative approximations. Acta Automatica Sinica. 2012;38(1):135–1393.

[22] Vaitkus M, Várady T. Parameterizing and extending trimmed regions for tensor-product surface fitting. Comput-Aided Des. 2018;104:125–140.

[23] Lin HW, Bao HJ, Wang GJ. Totally positive bases and progressive iteration approximation. Comput Math Appl. 2005;50(3-4):575–586.

[24] Huang ZD, Wang HD. On a progressive and iterative approximation method with memory for least square fitting. Computer Aided Geometric Design. 2020;82:101931.

[25] Lin H. The convergence of the geometric interpolation algorithm. Computer-Aided Design. 2010 jun;42(6):505–508.

[26] Maekawa T, Matsumoto Y, Namiki K. Interpolation by geometric algorithm. Computer-Aided Design. 2007 apr;39(4):313–323.

[27] de Boor C. How does Agee's smoothing method work? 79-73; 1979. p. 299–302.

[28] Qi D ea Tian Z. The method of numeric polish in curve fitting. Acta Mathematica Sinica. 1975;18:173–184.

[29] Yamaguchi F. A Design Method of Free Form Surfaces by a Computer Display (1st Report). Journal of the Japan Society of Precision Engineering. 1977;43(506):168–173.

[30] Carnicer JM, García-Esnaola M, Peña JM. Convexity of rational curves and total positivity. J Comput Appl Math. 1996;71(2):365–382.

[31] Delgado J, Peña JM. A shape preserving representation with an evaluation algorithm of linear complexity. Comput Aided Geom Design. 2003;20(1):1–10.

[32] Lin H, Wang G, Dong C. Constructing iterative non-uniform B-spline curve and surface to fit data points. Sci China Ser F. 2004;47(3):315–331.

[33] Lu L. Weighted progressive iteration approximation and convergence analysis. Comput Aided Geom Design. 2010;27(2):129–137.

[34] Lin H, Zhang Z. An extended iterative format for the progressive-iteration approximation. vol. 35. Elsevier BV; 2011. p. 967–975.

[35] Shi L-M WRH. An iterative algorithm of NURBS interpolation and approximation. J Math Res Exposition. 2006;26(5):735–743.

[36] Lin H. Local progressive-iterative approximation format for blending curves and patches. Comput Aided Geom Design. 2010;27(4):322–339.

[37] Young DM. Iterative Solution of Large Linear Systems. Elsevier; 1971.

[38] Schoenberg IJ, Whitney A. On Polya Frequency Function. III. The Positivity of Translation Determinants With an

Application to the Interpolation Problem by Spline Curves. Transactions of the American Mathematical Society. 1953;74(2):246–259.

[39] Wachirapong Jirakitpuwaput ea Poom Kumam. Image Quality Assessment: From Difference to Relative Entropy. Thai Journal of Mathematics (Accepted). 2021.