Communications in Information and Systems
Volume 17, Number 1, 25–43, 2017

# The Chebyshev accelerating method for progressive iterative approximation

Chengzhi Liu, Xuli Han, and Juncheng Li*

This paper proposes a new accelerating method for the progressive iterative approximation by using normalized totally positive bases. We prove that the method achieves an order-of-magnitude improvement compared to the weighted progressive iterative approximation. Moreover, we have shown that the well-known weighted progressive iterative approximation is a special application of the proposed method. The convergence is also analysed for all normalized totally positive bases. At the end, some numerical examples are given to illustrate the efficiency of the proposed method.

## 1. Introduction

The progressive iterative approximation(PIA) is used to find a curve or surface to approximate to a given set of data points. It has been extensively applied in various fields of science, such as Computer Aid Geometry Design(CAGD), inverse engineering, huge data fitting etcs. The PIA was firstly proposed by Qi et al.[1] in 1975, and de Boor[2] proved the convergence of PIA for the uniform cubic B-spline basis in 1979. In 2005, Lin et al.[3] showed that the PIA property holds for any normalized totally positive (NTP) basis. While in practise, large number of experiments have shown that the spectral radius of PIA is close to unity, which means that the convergence rate is usually very slow. Marco et al.[4] indicated that the collocation matrix is exponentially ill-conditioned as the polynomial degree increases. Deng and Wang[5] also pointed out that the main reason of slow convergence was the ill-conditioned matrix.

In recent years, researchers have been working on finding fast and accurate algorithms to accelerate the convergence rate of PIA. In 2010, Lu[6] proposed the weighted progressive iterative approximation, which speeded

*Research supported by the National Natural Science Foundation of China (Grants No. 11771453) and Hunan Provincial Natural Science Foundation of China (Grants No. 2017JJ3124).

up the convergence rate by inserting a parameter. Lin[7] pointed out the local property of the PIA and proposed the local progressive-iterative approximation format. The generalized minimal residual(GMRES) method was used as an alternative iterative method by Carnicer et al.[8]. By introducing the idea of precondition, Carnicer et al.[9] proposed several variants of the PIA iterations . In 2011, Lin and Zhang[10] developed the extended PIA (EPIA) format, in which the number of the control points were less than that of the given data points. In 2012, Deng and Wang[5] presented a method to accelerate the convergence rate by QR decomposition. In 2014, Deng and Lin[11] obtained the progressive iterative approximation for least square fitting (LSPIA). In [11], only a part of the data points were selected as the control points for iteration. In 2015, Liu and Deng[12] proposed the progressive iterative approximation for interpolating a set of points by non-uniform cubic B-spline curves based on the Jacobi iterative method.

Carnicer et al.[8] showed that the interpretation of the PIA iterations can be regarded as the classic Richardson iteration for linear system. Deng et al.[5] also deduced the equivalence of the progressive iterative approximation method and the algebraic interpolation method. It is well known that the Chebyshev semi-iterative method is useful when we know even more about spectrum of the iterative matrix. The main purpose of this paper is to use the Chebyshev semi-iterative method in constructing an effective approach to accelerate the convergence rate of PIA.

This paper is organized as follows: In Section 1, the progressive iterative approximation and the weighted progressive iterative approximation is introduced. In Section 2, the Chebyshev semi-iterative method is used to speed up the convergence rate of PIA. In Section 3, the convergence of the algorithm is discussed. At the end, some numerical examples are given to illustrate the efficiency of our methods.

## 2. Preliminary

### 2.1. Brief introduction of PIA

Given a sequence of control points $\{\mathbf{p}_i\}$ in $\mathbb{R}^2$ or $\mathbb{R}^3$, whose $i$th point is assigned to a parameter value $t_i, i = 0, 1, \ldots, n$. Let $\{u_i(t)\}_{i=0}^{n}$ be a blending basis, which is known as the normalized total positive (NTP) basis if they are nonnegative and satisfy $\sum\limits_{i=0}^{n} u_i(t) = 1$. Then we can construct the initial curve

$$\mathbf{r}^{(0)}(t) = \sum_{j=0}^{n} \mathbf{p}_j^{(0)} u_j(t),$$

where $\mathbf{p}_i^{(0)} = \mathbf{p}_i$ for all $i = 0, 1, \ldots, n$. Then the $(k+1)$th curve can be generated by

$$\mathbf{r}^{(k+1)}(t) = \sum_{j=0}^{n} \mathbf{p}_j^{(k+1)} u_j(t), \quad (k = 0, 1, 2, \ldots). \tag{1}$$

where $\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \Delta_i^{(k)}$, $\Delta_i^{(k)} = \mathbf{p}_i - \mathbf{r}^{(k)}(t_i)$.

Therefore,we get a sequence of curves $\mathbf{r}^{(k)}(t)$, and the initial curve has the progressive iterative approximation property if $\lim_{k\to\infty} \mathbf{r}^{(k)}(t_i) = \mathbf{p}_i$.

The iterative progress can be written in the matrix form as follows

$$\Delta^{(k+1)} = (I - B)\Delta^{(k)} = (I - B)^{k+1}\Delta^{(0)}. \tag{2}$$

where $I$ is the identity matrix and $B$ is the collocation matrix of a system $(u_0(t), ..., u_n(t))$ at the parameters $t_i, i = 0, 1, \ldots, n$. From (1) and (2), we can also define the sequence of control polygons as follows.

$$\begin{aligned} \mathbf{p}_i^{(k+1)} &= \mathbf{p}_i^{(k)} + \mathbf{p}_i - \mathbf{r}^{(k+1)}(t_i) \\ &= \mathbf{p}_i^{(k)} + \mathbf{p}_i - \sum_{j=0}^{n} \mathbf{p}_j^{(k)} u_j(t_i), \quad i = 0, 1, \ldots, n. \end{aligned} \tag{3}$$

Similarly, (3) can be written as the matrix form

$$\mathbf{P}^{(k+1)} = \mathbf{P} + (I - B)\mathbf{P}^{(k)}, \tag{4}$$

where $\mathbf{P}^{(k)} = [\mathbf{p}_0^{(k)}, \mathbf{p}_1^{(k)}, \ldots, \mathbf{p}_n^{(k)}]$,$\mathbf{P} = [\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n]$.

The PIA property means that the sequence of control polygons $\mathbf{P}^{(k)} = [\mathbf{p}_0^{(k)}, \mathbf{p}_1^{(k)}, \ldots, \mathbf{p}_n^{(k)}]$ converges to the control polygon of the interpolating curve $\mathbf{Q} = [\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_n]$ when $k \to \infty$, hence we have

$$\sum_{j=0}^{n} \mathbf{Q}_j u_j(t_i) = \mathbf{p}_i, \quad i = 0, 1, \ldots, n.$$

### 2.2. Alternative comprehension of PIA

Here we give an alternative comprehension of PIA. The main idea of PIA is to find an optimal control polygon $\mathbf{Q} = [\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_n]$ to minimize the distances between the curve $\mathbf{r}(t)$ and the points $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n$, that is

$$\min f(\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_n) = \min \sum_{i=0}^{n} \|\mathbf{p}_i - \mathbf{r}(t_i)\|^2 = \min \sum_{i=0}^{n} \left\| \mathbf{p}_i - \sum_{j=0}^{n} \mathbf{Q}_j u_j(t_i) \right\|^2.$$

To minimize the $f(\mathbf{Q}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_n)$, set the gradient of $f$ equals to zero, i.e.,

$$\frac{\partial f}{\partial \mathbf{Q}_j} = -2 \sum_{j=0}^{n} u_j(t_i) \left\| \mathbf{p}_i - \sum_{j=0}^{n} \mathbf{Q}_j u_j(t_i) \right\| = 0, \quad i = 0, 1, \ldots, n.$$

Hence,we have $\mathbf{p}_i - \sum_{j=0}^{n} \mathbf{Q}_j u_j(t_i) = 0, i = 0, 1, \ldots, n$, which can be written as a linear system

$$B\mathbf{Q} = \mathbf{P}. \tag{5}$$

Therefore the control polygon $\mathbf{Q}$ can be obtained by solving the linear system (5). It is well known that the classical Richardson iterative method (See[15].) can be used to solve the linear system (5), hence we have the iterative process $\mathbf{P}^{(k+1)} = (I - B)\mathbf{P}^{(k)} + \mathbf{P}$.

**Remark 1.** The PIA can also be seem as the classical iterative method for solving a linear system $B\mathbf{Q} = \mathbf{P}$, and $I - B$ is the iterative matrix. As is known to all, the iterative progress converges if and only if the spectral radius is less than unity, i.e., $\rho(I - B) < 1$. Generally speaking, the smaller the spectral radius is, the faster the rate of convergence is. This means that we can use the other methods to construct a sequence of control polygons which converges to the control polygon of the interpolating curve.

Here we introduce a lemma which will be used in the following section.

**Lemma 2.** *(See* [6]*.) Let $B = \big(u_j(t_i)\big)_{i=0,1,\ldots,n}^{j=0,1,\ldots,n}$ be a collocation matrix of an NTP basis, and let $\lambda_i (i = 0, 1, \ldots, n)$ be its eigenvalues sorted in nonincreasing order. Then,*

**(a):** $\lambda_0 = \rho(B) = 1$ *and* $0 < \lambda_i \leq 1$ *for all* $i = 0, 1, \ldots, n$;

**(b):** $0 \leq \rho(I - B) = 1 - \lambda_n < 1$, *the eigenvalues of* $I - B$ *satisfy* $0 \leq \lambda_i(I - B) \leq 1 - \lambda_n, (i = 0, 1, \ldots, n)$.

### 2.3. The weighted progressive iterative approximation

In order to accelerate the convergence of the PIA, Lu[6] proposed the weighted progressive iterative approximation via multiplying by a weight $\omega$, i.e.,

$$\mathbf{p}_i^{(k+1)} = \mathbf{p}_i^{(k)} + \omega \Delta_i^{(k+1)}, \tag{6}$$

where $\omega$ is a positive real number between 0 and 2, and it is taken to guarantee the convergence rate of the iterative process. Obviously, it degenerates into the progressive iterative approximation when $\omega = 1$.

From (2) and (6), we can obtain that the matrix form of the weighted progressive iterative approximation is

$$\mathbf{P}^{(k+1)} = (I - \omega B)\mathbf{P}^{(k)} + \omega \mathbf{P}. \tag{7}$$

Lu[6] has obtained the optimal value of $\omega$ by the following lemma:

**Lemma 3.** *The weighted progressive iterative approximation has the fastest convergence rate when* $\omega = \frac{2}{1+\lambda_n}$, *and in such case* $\rho(I - \omega B) = \frac{1-\lambda_n}{1+\lambda_n}$.

## 3. The Chebyshev accelerating technique for PIA

In this section, we give another way to accelerate the convergence rate of the iterative process. Suppose $\mathbf{P}^{(0)}, \mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(m)}$ have been generated by the iterative process (4), we consider the linear combinations of the vectors $\mathbf{P}^{(j)} (j = 0, 1, \ldots, m)$, that is

$$\mathbf{q}^{(m)} = \sum_{j=0}^{m} a_{m,j} \mathbf{P}^{(j)}, \tag{8}$$

where $\sum_{j=0}^{m} a_{m,j} = 1$. Both sides of (8) are subtracted by $\mathbf{Q}$ simultaneously and let $\delta_m = \mathbf{q}^{(m)} - \mathbf{Q}$, then

$$\delta_m = \sum_{j=0}^{m} a_{m,j}(\mathbf{P}^{(j)} - \mathbf{Q}) = \left[\sum_{j=0}^{m} a_{m,j}(I - B)^j\right](\mathbf{P}^{(0)} - \mathbf{Q}). \tag{9}$$

By using the 2-norm, we have

$$\begin{aligned}\|\delta_m\|_2 &= \left\|\left[\sum_{j=0}^{m} a_{m,j}(I - B)^j\right](\mathbf{P}^{(0)} - \mathbf{Q})\right\|_2 \\ &\leq \left\|\left[\sum_{j=0}^{m} a_{m,j}(I - B)^j\right]\right\|_2 \left\|(\mathbf{P}^{(0)} - \mathbf{Q})\right\|_2.\end{aligned}$$

Let $p_m(t) = \sum_{j=0}^{m} a_{m,j} t^j$, then we can define the matrix polynomial $p_m(I - B) = \sum_{j=0}^{m} a_{m,j}(I - B)^j$, which follows from (9) that

$$\|\delta_m\|_2 \leq \left\|p_m(I - B)\right\|_2 \left\|(\mathbf{P}^{(0)} - \mathbf{Q})\right\|_2. \tag{10}$$

If the norm $\left\|p_m(I - B)\right\|_2$ is small enough, then $\|\delta_m\|_2$ is so close to zero as $m$ increases. This means that we could find an optimal polynomial $p_m(t)$ to minimize the error norm $\|\delta_m\|_2$. As is known to all, the Chebyshev polynomials are suitable for these whose eigenvalues of the iterative matrix distributed at the interval $(-1, 1)$. Fortunately, the lower and upper bounds of the eigenvalues of the iterative matrix $(I - B)$ can be calculated accurately. Hence, we can use the Chebyshev polynomials to speed up the rate of convergence.

Consider the Chebyshev polynomials $C_m(x)$ generated by the recursion

$$C_{m+1}(x) = 2xC_m(x) - C_{m-1}(x), m = 1, 2, \ldots, \tag{11}$$

where $C_0(x) = 1$ and $C_1(x) = x$. For the minimal eigenvalue $\lambda_n$ given in Lemma1, then the polynomial in (10) can be chosen as follows

$$p_m(x) = \frac{C_m(\frac{2x+\lambda_n-1}{1-\lambda_n})}{C_m(\mu)}. \tag{12}$$

where $\mu = \frac{1+\lambda_n}{1-\lambda_n}$. It is easy to verify that $p_m(1) = 1$.

**Remark 4.** As is mentioned in (8), given an iterative sequence, one can construct another sequence which will converge faster than the given method. This method is known as the semi-iterative method. Especially, it is called Chebyshev semi-iterative method when the polynomial is constructed by using Chebyshev polynomials. Golub et al. have pointed the polynomial (12) is the optimal polynomial which minimizes $\|\delta_m\|_2$, readers can refer to [16] for more details.

For the sake of understanding, we discuss the accelerated iterations with the linear and quadratic polynomials separately. Since $C_0(x) = 1, C_1(x) = x, C_2(x) = 2x^2 - 1$, from (12) we have

$$p_0(x) = 1, \quad p_1(x) = \frac{2x - 1 + \lambda_n}{1 + \lambda_n},$$
$$p_2(x) = \frac{8x^2 - 8(1 - \lambda_n)x + (1 - \lambda_n)^2}{\lambda_n^2 + 6\lambda_n + 1}.$$

(1) If $m = 1$, then $p_1(I - B) = \frac{2(I-B)-(1-\lambda_n)I}{1+\lambda_n} = \frac{(1+\lambda_n)I-2B}{1+\lambda_n}$. By (9), we have

$$\mathbf{q}^{(1)} - \mathbf{Q} = \frac{(1 + \lambda_n)I - 2B}{1 + \lambda_n}(\mathbf{P}^{(0)} - \mathbf{Q}).$$

Note that $B\mathbf{Q} = \mathbf{P}$ and $\mathbf{q}^{(0)} = \mathbf{P}^{(0)} = \mathbf{P}$, then

$$\mathbf{q}^{(1)} = \frac{(1 + \lambda_n)I - 2B}{1 + \lambda_n}\mathbf{q}^{(0)} + \frac{2}{1 + \lambda_n}\mathbf{P}. \tag{13}$$

Suppose $\mathbf{q}^{(k)}$ is an approximation to $\mathbf{Q}$, a natural way to generate a new approximation $\mathbf{q}^{(k+1)}$ is to compute

$$\mathbf{q}^{(k+1)} = \frac{(1 + \lambda_n)I - 2B}{1 + \lambda_n}\mathbf{q}^{(k)} + \frac{2}{1 + \lambda_n}\mathbf{P}, \tag{14}$$

which is just the weighted progressive iterative approximation with $\omega = \frac{2}{1+\lambda_n}$. Hence we have the following theorem.

**Theorem 5.** *The iterative process (14) can be regarded as a particular application of the weighted progressive iterative approximation with the optimal weight* $\omega = \frac{2}{1+\lambda_n}$.

*Proof.* For the weighted progressive iterative approximation with $\omega = \frac{2}{1+\lambda_n}$, we have

$$\begin{aligned}\mathbf{P}^{(k+1)} &= (I-\omega B)\mathbf{P}^{(k)} + \omega\mathbf{P}\\ &= (I - \frac{2}{1+\lambda_n}B)\mathbf{P}^{(k)} + \frac{2}{1+\lambda_n}\mathbf{P}\\ &= \frac{(1+\lambda_n)I - 2B}{1+\lambda_n}\mathbf{P}^{(k)} + \frac{2}{1+\lambda_n}\mathbf{P},\end{aligned}$$

which is the same as the process (14), then the conclusion holds. □

(2) If $m = 2$, then $p_2(I-B) = \frac{8(I-B)^2 - 8(1-\lambda_n)(I-B) + (1-\lambda_n)^2 I}{\lambda_n^2 + 6\lambda_n + 1}$. Similarly, we have the quadratic Chebyshev polynomial accelerating iteration

$$\mathbf{q}^{(k+1)} = p_2(I-B)\mathbf{q}^{(k)} - \frac{8B - 8(1+\lambda_n)I}{\lambda_n^2 + 6\lambda_n + 1}\mathbf{P}. \tag{15}$$

In order to obtain a more practical accelerating procedure, we take the more generally situation into account for calculating $\mathbf{q}^{(k+1)}$. By substituting $p_{m-1}(I-B), p_m(I-B)$,$p_{m+1}(I-B)$into (9) and using the recurrence relation (11), we have the following relationship(For more details about the deduction, refer to the semi-iterative methods in [17, 18]).

$$\begin{aligned}\mathbf{q}^{(m+1)} = &\frac{\rho_{m+1}}{1+\lambda_n}\{[2(I-B) - (1-\lambda_n)I]\mathbf{q}^{(m)} + 2\mathbf{P}\}\\ &+ (1-\rho_{m+1})\mathbf{q}^{(m-1)}], \quad m \geq 1.\end{aligned} \tag{16}$$

where $\rho_1 = 1, \rho_2 = \frac{2\mu^2}{2\mu^2-1}$ and $\rho_{m+1} = (1 - \frac{\rho_m}{4\mu^2})^{-1}, m \geq 2$. From (13), we have $\mathbf{q}^{(0)} = \mathbf{P}^{(0)}$ and $\mathbf{q}^{(1)} = \frac{2(I-B)-(1-\lambda_n)I}{1+\lambda_n}\mathbf{q}^{(0)} + \frac{2}{1+\lambda_n}\mathbf{P}$.

Thus we obtain the Chebyshev accelerating technique for the PIA(CA-$m$ for short), here $m$ represents the degree of the polynomial $p_m(t)$. Particularly, the iterative process (14) is the CA-1(also named weighted progress iteration approximation by Theorem 5) and the iterative process (15) is the CA-2 method. The convergence analysis will be discussed in the next section.

**Remark 6.** Note that the iterative matrix of CA-2 is

$$\frac{8(I-B)^2 - 8(1-\lambda_n)(I-B) + (1-\lambda_n)^2 I}{\lambda_n^2 + 6\lambda_n + 1},$$

we have to calculate $(I-B)^2$. While in the iterative process (16), CA-$m$ not needs to compute the multiplication of matrices. As is known to all, the multiplication of matrices brings large computation costs, therefore, we are more willing to use the iterative process (16) in practise because it has almost the same computationally demanding as the Weighted PIA.

## 4. Convergence analysis

Firstly, we introduce several definitions to measure the rate of convergence.

**Definition 7.** (See*[17]*.) The virtual average spectral radius of iterative process (16) is $[\rho\big(p_m(I-B)\big)]^{\frac{1}{m}}$.

**Definition 8.** (See*[17]*.) The average rate of convergence of iterative process (16) is defined by

$$R_m = -\frac{1}{m}\log\rho\big(p_m(I-B)\big),$$

and the asymptotic average rate of converge is defined by

$$R = \lim_{m\to\infty} R_m = -\lim_{m\to\infty}[\rho\big(p_m(I-B)\big)]^{\frac{1}{m}}.$$

**Theorem 9.** *For any normalized totally positive (NTP) basis, the iterative sequence* $\{\boldsymbol{q}^{(k)}\}_{k=0}^{\infty}$ *generated by CA-m method converges if* $m \geq 1$*. Moreover, the virtual average spectral radius decreases as m increases, in other words, the rate of convergence will be faster when the degree of the Chebyshev polynomial is larger.*

*Proof.* In particular case, if $m=1$, then the spectral radius

$$\begin{aligned}\rho(p_1(I-B)) &= \frac{\max\limits_{0\leq i\leq n}\left|C_1\big(\frac{2\lambda_i(I-B)+\lambda_n-1}{1-\lambda_n}\big)\right|}{C_1(\mu)} \\ &= \frac{\max\limits_{0\leq i\leq n}\left|\frac{2\lambda_i(I-B)+\lambda_n-1}{1-\lambda_n}\right|}{\frac{1+\lambda_n}{1-\lambda_n}} = \frac{1-\lambda_n}{1+\lambda_n}.\end{aligned}$$

Here $\lambda_i(I-B)$ denotes the eigenvalues of the matrix $I-B$. It follows from Lemma 2 that the spectral radius of CA-1 equals to the weighted PIA and hence the iterative sequence $\{\mathbf{q}^{(k)}\}_{k=0}^{\infty}$ generated by CA-1 converges.

If $m > 1$, it follows from (10) that

$$\begin{aligned}\rho(p_m(I-B)) &= \max_{0\le i\le n} |p_m(\lambda_i(I-B))| \\ &= \frac{\max_{0\le i\le n}\left|C_m\left(\frac{2\lambda_i(I-B)+\lambda_n-1}{1-\lambda_n}\right)\right|}{C_m(\mu)}, \quad m\ge 1.\end{aligned}$$

The Chebyshev polynomials can also be expressed as

$$C_m(x) = \begin{cases}\cos(m \arccos x), & x\le 1,\ m\ge 0,\\ \cosh(m \operatorname{arg}\cosh x), & x\ge 1,\ m\ge 0.\end{cases} \tag{17}$$

From Lemma 1, it is easy to verify that $\max_{0\le i\le n}\left|C_m\left(\frac{2\lambda_i(I-B)+\lambda_n-1}{1-\lambda_n}\right)\right| \le 1$, and hence we have

$$\rho(p_m(I-B)) = \frac{1}{C_m(\mu)}, \quad m\ge 1.$$

Recall that $\mu = \frac{1+\lambda_n}{1-\lambda_n} > 1$, from (17), we have

$$\begin{aligned}C_m(\mu) &= \cosh(m \operatorname{arg}\cosh \mu)\\ &= \cosh\left(m\ln\left(\mu^2+\sqrt{\mu^2-1}\right)\right)\\ &= \frac{1}{2}\left\{\left(\mu^2+\sqrt{\mu^2-1}\right)^m + \left(\mu^2+\sqrt{\mu^2-1}\right)^{-m}\right\}\end{aligned}$$

For convenience, let $x = \left(\mu^2+\sqrt{\mu^2-1}\right)^m$, it is easy to verify $x > 1$, and hence

$$\rho(p_m(I-B)) = \frac{1}{C_m(\mu)} = \frac{2x}{1+x^2} < 1. \tag{18}$$

Thus, we conclude that the CA-$m$($m \ge 1$) converges.

Moreover, since

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{2x}{1+x^2}\right) = \frac{2(1-x^2)}{(1+x^2)^2} < 0, \quad x > 1,$$

and $x$ is a monotonically increasing function of $m$, then $\rho(p_m(I-B))$ is strictly decreasing as $m$ increasing, which can be obtained from the monotonicity of composite functions. Thus, the rate of convergence will be faster as $m$ increases. □

Since the virtual average spectral radius decreases as $m$ increases, we have already verify that the weighted progressive iterative approximation is a special situation of the Chebyshev acceleration. Thus, the spectral radius of the CA-$m(m > 1)$ is less than the weighted progressive iterative approximation, and therefore we have the following corollary.

**Corollary 10.** If the degree of the Chebyshev polynomial $m > 1$, then the virtual average spectral radius of the CA-$m$ method is less than the weighted progressive iterative approximation, that is, $[\rho\big(p_m(I-B)\big)]^{\frac{1}{m}} < \rho(I-\omega B)$.

**Remark 11.** For one thing, the iterative process (16) converges faster then the weighted PIA when $\mu = \frac{1+\lambda_n}{1-\lambda_n}$ in Theorem 9. While in practise, it needs a large amount of computation to calculate the smallest eigenvalues $\lambda_n$, and it is not worth doing so at the expense of computation. For another, Young [17] have pointed out that a small overestimation of $\mu$ results in a small decrease in the rate of convergence, but a comparable underestimation of $\mu$ results in a much larger decrease. Since $\mu = \frac{1+\lambda_n}{1-\lambda_n}$, hence the value of $\lambda_n$ may fluctuate instead of the true value in practise.

We remark that the spectral radius of $I - B$ is small, in other words, $\lambda_n$ is so close to 0 and $\mu$ is so close to unity. Based on this hypothesis, we have the following theorem.

**Theorem 12.** *The CA-$m(m > 1)$ method results in an order-of-magnitude improvement in the rate of convergence compared to the weighted progressive iterative approximation.*

*Proof.* Firstly, for the CA-$m(m > 1)$ method, according to (18) and Definition 2, the average rate of convergence is given by

$$R_m = -\frac{1}{m}\log\rho\big(p_m(I-B)\big) = -\frac{1}{m}\log\frac{2\tau^m}{1+\tau^{2m}},$$

where $\tau = \mu^2 + \sqrt{\mu^2-1}$.

Hence the asymptotic average rate of converge is defined by

$$R = \lim_{m\to\infty} R_m = \lim_{m\to\infty} -\frac{1}{m}\log\frac{2\tau^m}{1+\tau^{2m}} = -\log\tau.$$

Secondly, for the weighted progressive iterative approximation method,

$$R_1 = -\log\rho\big(p_1(I-B)\big) = -\log\frac{2\tau}{1+\tau^2}.$$

Moreover, for $m$ sufficiently large and $\tau$ sufficiently close to unity, we have

$$\lim_{\tau\to 1}\frac{R}{(R_1)^{\frac{1}{2}}}=\lim_{\tau\to 1}\frac{-\log\tau}{\left(-\log\frac{2\tau}{1+\tau^2}\right)^{\frac{1}{2}}}=\sqrt{2}.$$

Hence the CA-$m(m>1)$ method results in an order-of-magnitude improvement in the rate of convergence compared to the weighted PIA. □

## 5. Numerical examples

In this section, some numerical examples are given to show the efficiency of our method. Firstly we introduce the error norms of the $k$-th curve as follows,

$$\varepsilon_k=\max_{0\le i\le n}\|\Delta_i^{(k)}\|=\max_{0\le i\le n}\|\mathbf{p}_i-\mathbf{r}^{(k)}(t_i)\|,$$

where the norm is the Euclidean norm.

**Example 13.** Consider the Lemniscate of Gerono given by the parametric equations

$$\begin{cases} x(t)=\cos t, \\ y(t)=\sin t\cos t, \end{cases} \qquad t\in[0,2\pi].$$

A sequence of 11 points $\{\mathbf{p}_i\}_{i=0}^{10}$ is selected from the Lemniscate of Gerono in the following way

$$\mathbf{p}_i=\left(x\left(-\frac{\pi}{2}+i\frac{\pi}{5}\right),\ y\left(-\frac{\pi}{2}+i\frac{\pi}{5}\right)\right),\quad i=0,1,\ldots,10.$$

**Example 14.** Consider the helix of radius 5 given by

$$(x(t),y(t),z(t))=(5\cos t,5\sin t,t),\quad t\in[0,6\pi].$$

19 points $\{\mathbf{p}_i\}_{i=0}^{18}$ are selected from the helix in the following way

$$\mathbf{p}_i=\left(x\left(i\frac{\pi}{3}\right),\ y\left(i\frac{\pi}{3}\right),z\left(i\frac{\pi}{3}\right)\right),\quad i=0,1,\ldots,18.$$

In order to make a comparison with the other method, the Bézier curves [6] and the Said-Ball curves[6] are used to approximate these points in the Example 1 and 2 separately. The parameter values corresponding to these points are the same as the method in Lu[6], i.e.,$t_i=i/n, i=0,1,...,n$. The virtual spectral radius of the Chebyshev acceleration as well as the others are listed in Table 1 and 2, and the error norms are also listed in Table 3 and 4 (The CA-2 is the iterative process (14), which is generated by the linear polynomial $p_1(I-B)$ defined in (12); the CA-4 is generated by the quartic polynomial $p_4(I-B)$; the CA-$m$ is the iterative process (16), and the others in this paper are similar). Without loss of generality, several Chebyshev semi-iterative schemes and the weighted PIA(WPIA for short in the tables) are used in these examples for comparison. As it could be expected, the rate of convergence suffers from the ill-conditioned matrix whose the spectral radius is close to unity. It is evident from Table 3 and 4 that the accelerating techniques described in this paper is faster than the weighted PIA, i.e., with the same iterations, the accuracy of our method is improved by roughly an order-of-magnitude. Especially, the effect will be more obvious when the number of polynomials increases.

In addition, as we can see from Table 3 and 4, for the errors generated by CA-4 after 10 iterations, it will be reached by CA-2 after 20 iterations, however, it will be reached by weighted PIA after about 40 iterations. It shows that the CA-2 is about two times faster than the weighted PIA and the CA-4 is about four times faster than the weighted PIA, which confirms to Corollary 1. In the view of accelerating the rate of convergence, we are more involved with increasing the degree of polynomials $p_m$, but that's not something we are inclined to do in practice, this is because it will increase calculates quantity, e.g., in the iterative process (15), $(I-B)^2$ will cost a great deal of calculation. Therefore, the iterative process (16) is the best selection for accelerating the rate of convergence.

Figure 1 shows the Bézier curves when fitting the Lemniscate of Gerono; Figure 2 shows the Said-Ball curves when fitting the Lemniscate of Gerono. The Bézier curves and the Said-Ball curves fitting to the helix are shown in Figure 3 and 4 respectively. Among all the figures, the sub-figures (a)-(d) are generated by the weighted PIA, CA-2, CA-4 and CA-$m$ respectively, and all the curves are generated after 20 iterations. As we can see from these figures that the curves generated by the Chebyshev accelerating method are closer to the points needed to be interpolated (the points with a circle in the figures). All the figures indicate that our method make the iterative process converge faster than the weighted PIA.

Table 1: The (virtual) spectral radius of the Chebyshev acceleration compared with the other methods when fitting the Lemniscate of Gerono.

| Bézier curve | | | | Said-Ball curve | | | |
|---|---|---|---|---|---|---|---|
| PIA | WPIA | CA-2 | CA-4 | PIA | WPIA | CA-2 | CA-4 |
| 0.999637 | 0.999275 | 0.998551 | 0.997112 | 0.999820 | 0.999641 | 0.998565 | 0.994281 |

Table 2: The (virtual) spectral radius of the Chebyshev acceleration compared with the other methods when fitting the helix.

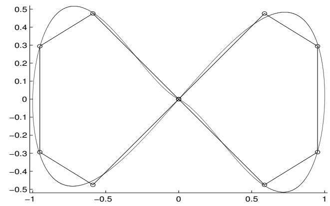
| Bézier curve | | | | Said-Ball curve | | | |
|---|---|---|---|---|---|---|---|
| PIA | WPIA | CA-2 | CA-4 | PIA | WPIA | CA-2 | CA-4 |
| 0.99999984 | 0.99999967 | 0.99999870 | 0.99999479 | 0.99999996 | 0.99999992 | 0.99999968 | 0.99999873 |

Table 3: The error norms of the Chebyshev acceleration compared with WPIA after $k$ iterations when fitting the Lemniscate of Gerono.

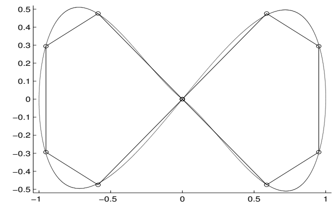
| $k$ | Bézier curve | | | | Said-Ball curve | | | |
|---|---|---|---|---|---|---|---|---|
| | WPIA | CA-2 | CA-4 | CA-$m$ | WPIA | CA-2 | CA-4 | CA-$m$ |
| 0 | 4.6201e-01 | 4.6201e-01 | 4.6201e-01 | 4.6201e-01 | 5.4704e-01 | 5.4704e-01 | 5.4704e-01 | 5.4704e-01 |
| 10 | 5.4465e-02 | 2.3400e-02 | 5.2019e-03 | 2.1528e-02 | 8.9794e-02 | 6.3642e-02 | 1.0186e-02 | 4.2300e-02 |
| 20 | 2.3697e-02 | 9.3462e-03 | 1.5658e-03 | 7.4514e-03 | 5.2518e-02 | 1.5067e-02 | 3.8575e-03 | 9.5411e-03 |
| 30 | 1.4613e-02 | 7.1746e-03 | 4.7112e-04 | 5.5941e-03 | 3.3296e-02 | 1.0199e-02 | 2.0563e-03 | 7.2201e-03 |
| 40 | 1.2718e-02 | 5.3573e-03 | 1.4176e-04 | 2.7534e-03 | 2.2521e-02 | 9.0690e-03 | 1.3022e-03 | 5.3525e-03 |

Table 4: The error norms of the Chebyshev acceleration compared with WPIA after $k$ iterations when fitting the helix.

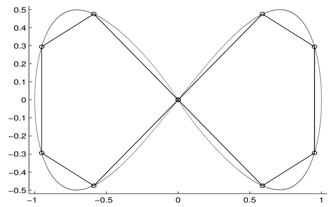
| $k$ | Bézier curve | | | | Said-Ball curve | | | |
|---|---|---|---|---|---|---|---|---|
| | WPIA | CA-2 | CA-4 | CA-$m$ | WPIA | CA-2 | CA-4 | CA-$m$ |
| 0 | 4.6246e-00 | 4.6246e-00 | 4.6246e-00 | 4.6246e-00 | 5.6981e-00 | 5.6981e-00 | 5.6981e-00 | 5.6981e-00 |
| 10 | 9.8209e-01 | 2.7880e-01 | 2.3579e-01 | 2.6314e-01 | 1.8124e-00 | 7.0115e-01 | 1.2181e-01 | 4.7184e-01 |
| 20 | 4.6530e-01 | 1.1148e-01 | 3.7426e-02 | 1.0271e-01 | 8.3426e-01 | 1.7397e-01 | 5.1675e-02 | 1.2633e-01 |
| 30 | 2.7866e-01 | 7.8482e-02 | 2.0300e-02 | 5.2454e-02 | 4.9451e-01 | 1.1476e-01 | 3.0864e-02 | 7.3371e-02 |
| 40 | 2.0137e-01 | 5.9935e-02 | 1.5128e-02 | 3.4744e-02 | 3.3185e-01 | 8.2361e-02 | 2.2266e-02 | 3.8842e-02 |



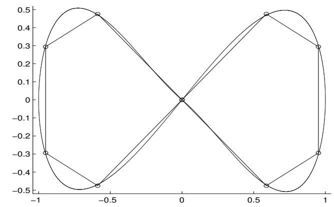
(a) Weighted PIA



(b) CA-2

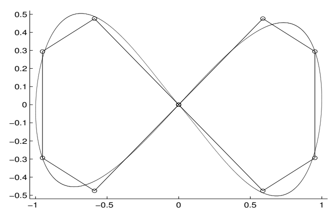


(c) CA-4

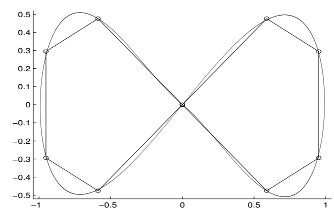


(d) CA-$m$

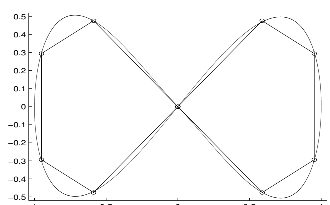
Figure 1: Bézier curves when fitting the Lemniscate of Gerono after 20 iterations.



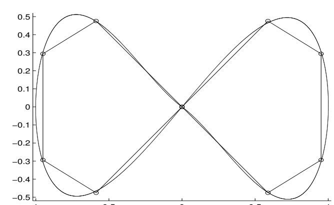
(a) Weighted PIA



(b) CA-2

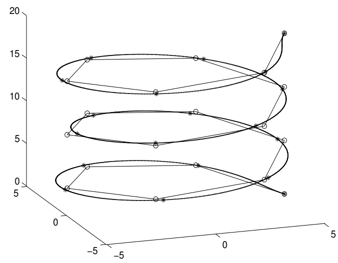


(c) CA-4

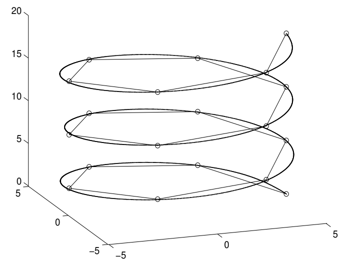


(d) CA-$m$

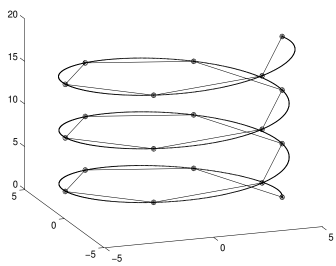
Figure 2: Said-Ball curves when fitting the Lemniscate of Gerono after 20 iterations.



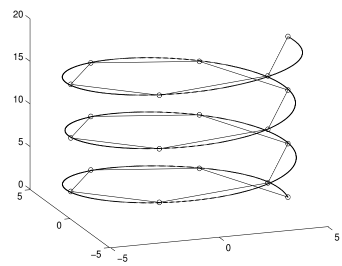
(a) Weighted PIA



(b) CA-2



(c) CA-4



(d) CA-$m$

Figure 3: Bézier curves when fitting the helix after 20 iterations.

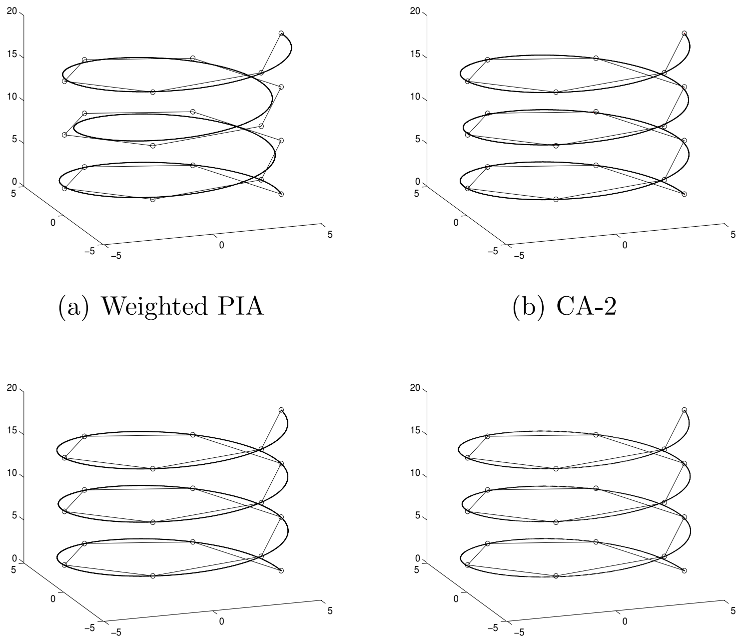
## 6. Conclusions

This paper presents an accelerating method for the progressive iterative approximation. We deduce that the weighted progressive iterative approximation method is a special situation of our method. For any normalized totally positive (NTP) basis, we have shown that the proposed method is convergent. What's more, the Chebyshev accelerating method has an order-of-magnitude improvement compared with the weighted PIA. Numerical examples show that our method make the algorithm faster than the weighted PIA.

## References


[1] D. X. Qi, Z. X. Tian, Y. X. Zhang, et al., *The method of numeric polish in curve fitting*, Acta Mathematica Sinica **18** (1975), no. 3, 173–184.

[2] C. de Boor, *How does Agee's smoothing method work*, Proceedings of the 1979 army numerical analysis and computers conference, ARO Report. (1979), 299–302.



(a) Weighted PIA (b) CA-2

(c) CA-4 (d) CA-$m$

Figure 4: Said-Ball curves when fitting the helix after 20 iterations.

[3] H. W. Lin, H. J. Bao, and G. J. Wang, *Totally positive bases and progressive iterative approximation*, Computers and Mathematics with Applications **50** (2005), no. 3, 575–586.

[4] A. Marco and J. J. Marti Nez, *A fast and accurate algorithm for solving Bernstein-Vandermonde linear systems*, Linear Algebra and Its Applications **422** (2006), no. 2, 616–628.

[5] S. H. Deng and G. Z. Wang, *Numerical analysis of the progressive iterative approximation method*, Journal of Computer-Aided Design and Graphics **24** (2012), no. 7, 879–884.

[6] L. Lu, *Weighted progressive iterative approximation and convergence analysis*, Computer Aided Geometric Design **27** (2010), no. 2, 129–137.

[7] H. W. Lin, *Local progressive-iterative approximation format for blending curves and patches*, Computer Aided Geometric Design **27** (2010), no. 4, 322–339.

[8] J. M. Carnicer and J. Delgado, *On the progressive iterative approximation property and alternative iterations*, Elsevier Science Publishers B. V. 2011.

[9] J. M. Carnicer and J. Delgado, and J. M. Peña, *Progressive iterative approximation and the geometric algorithm*, Computer-Aided Design **44** (2012), no. 2, 143–145.

[10] H. W. Lin and Z. Zhang, *An extended iterative format for the progressive-iteration approximation*, Computers and Graphics **35** (2011), no. 5, 967–975.

[11] C. Deng and H. Lin, *Progressive and iterative approximation for least squares B-spline curve and surface fitting*, Computer-Aided Design **47** (2014), no. 1, 32–44.

[12] X. Liu and C. Deng, *Jacobi-PIA Algorithm for non-uniform cubic B-spline curve interpolation*, Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal of Computer-Aided Design and Computer Graphics **27** (2015), no. 3, 485–491.

[13] J. Delgadoa and J. M. Peña, *Progressive iterative approximation and bases with the fastest convergence rates*, Computer Aided Geometric Design **24** (2007), no. 1, 10–18.

[14] J. M. Carnicer and J. Delgado, and J. M. Peña, *Richardson method and totally nonnegative linear systems*, Linear Algebra and Its Applications **433** (2010), no. 11, 2010–2017.

[15] Y. Saad, *Iterative methods for sparse linear systems*, IEEE Computational Science and Engineering **3** (2009), no. 4, 88–88.

[16] G. H. Golub and R. S. Varga, *Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods*, Numerische Mathematik **3** 1961, no. 1, 157–168.

[17] D. M. Young, *Iterative solution of large linear systems*, Academic Press, 1971.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computation*, third edition, The Johns Hopkins University Press, 1996.

School of Mathematics and Statistics
Central South University
Changsha, 410083, China
and College of Mathematics and Finance
Hunan University of Humanities, Science and Technology
Loudi 417000, China
*E-mail address*: 162101002@csu.edu.cn

School of Mathematics and Statistics
Central South University
Changsha, 410083, China
*E-mail address*: xlhan@csu.edu.cn

College of Mathematics and Finance
Hunan University of Humanities, Science and Technology
Loudi 417000, China
*E-mail address*: 14311993@qq.com