Appl. Comput. Harmon. Anal. ••• (••••) •••–•••

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Letter to the Editor

An enhancement algorithm for cyclic adaptive Fourier decomposition

Tao Qian[a], Jianzhong Wang[b], Weixiong Mai[c,*]

[a] *Macau University of Science and Technology, Macao*
[b] *Department of Mathematics and Statistics, Sam Houston State University, Huntsville, TX 77341, United States of America*
[c] *School of Mathematics (Zhuhai), Sun Yat-Sen University (Zhuhai), China*

A R T I C L E I N F O

Article history:
Received 30 August 2018
Received in revised form 8 January 2019
Accepted 10 January 2019
Available online xxxx
Communicated by Charles K. Chui

MSC:
41A20
30H10
30J10
65T50

Keywords:
Best rational approximation
Blaschke products
Hardy space
Gradient descent method

A B S T R A C T

One important problem in the theory of Hardy space is to find the best rational approximation of a given order to a function in the Hardy space H^2 on the unit disk. It is equivalent to finding the best Blaschke form with free poles. The cyclic adaptive Fourier decomposition method is based on the grid search technique. Its approximative precision is limited by the grid spacing. This paper proposes two enhanced methods of the cyclic adaptive Fourier decomposition. The proposed algorithms utilize the gradient descent optimization to tune the best pole-tuple on the mesh grids, reaching higher precision. Their performances are confirmed by several examples.

1. Introduction

Throughout the paper, we denote by $\mathbb{D}$ the open unit disc, and $H^2 = H^2(\mathbb{D})$ the Hardy H^2-space on $\mathbb{D}$:

$$H^2 = \{f(z) = \sum_{k=0}^{\infty} c_k z^k \ : \ \sum_{k=0}^{\infty} |c_k|^2 < \infty\},$$

which is equipped with the inner product

* Corresponding author.
E-mail addresses: tqian@must.edu.mo (T. Qian), jzwang@shsu.edu (J. Wang), maiweixiong@gmail.com (W. Mai).

https://doi.org/10.1016/j.acha.2019.01.003
1063-5203/© 2019 Published by Elsevier Inc.

$$\langle f,g\rangle = \frac{1}{2\pi i}\oint_{\partial\mathbb{D}} f(z)\overline{zg(z)}\,dz = \frac{1}{2\pi i}\oint_{\partial\mathbb{D}} \frac{f(z)\overline{g(z)}}{z}\,dz, \quad f,g\in H^2,$$

and $||\cdot|| = \langle\cdot,\cdot\rangle^{\frac{1}{2}}$. Now we recall the classical definition of *best n-rational approximation*. Let p and q be polynomials, and all zeros of q be outside the closed unit disc. In this paper, we always assume that p and q are coprime so that all rational functions of the form of p/q are *non-degenerate*. The *order of a rational function* p/q is defined by $\mathrm{ord}(p/q) = \max\{\deg(p), \deg(q)\}$. A *best n-rational approximation* to $f \in H^2$ is an n-order rational function p_1/q_1 that satisfies

$$\|f - p_1/q_1\| \le \|f - p/q\|, \quad \mathrm{ord}(p/q) \le n.$$

An important type of rational approximation is Blaschke form approximation, which is briefly introduced in the following: For a given n-vector $\mathbf{a} = [a_1, \cdots, a_n]^T \in \mathbb{D}^n$, the n-Takenaka–Malmquist orthonormal rational function system $\{B_k\}_{k=1}^n$ is defined by

$$B_k(z) = \frac{\sqrt{1-|a_k|^2}}{1-\bar{a}_k z}\prod_{j=1}^{k-1}\frac{z-a_j}{1-\bar{a}_j z}.$$

It is known that the linear subspace of H^2 spanned by $\{B_k\}_{k=1}^n$ is invariant under the permutations of $\mathbf{a}$. We denote it by $L(\mathcal{A})$, where $\mathcal{A}$ is the n-tuple $\{a_1, a_2, \cdots, a_n\}$, and call a function in $L(\mathcal{A})$ *n-Blaschke form*. Then, the n-Blaschke form

$$f_n = \sum_{k=1}^{n}\langle f, B_k\rangle B_k \tag{1}$$

is the orthogonal projection of $f \in H^2$ to $L(\mathcal{A})$. We define the squared H^2-error of the projection (1) by

$$A(f;\mathcal{A}) = \|f - f_n\|^2 = \|f\|^2 - \sum_{k=1}^{n}|\langle f, B_k\rangle|^2,$$

where n is *the approximation degree* and $E(f,\mathcal{A}) = \sum_{k=1}^n |\langle f, B_k\rangle|^2$ *the energy of f* (at $\mathcal{A}$). We say that $\mathcal{B} = \mathcal{B}(n) = \{b_1, b_2, \cdots, b_n\}$ is a *best n-tuple* if it induces an *n-best Blaschke form* approximation to f in the sense that $A(f,\mathcal{B}) = \min_{\mathcal{A}\subset\mathbb{D}} A(f,\mathcal{A})$, which is equivalent to

$$\mathcal{B} = \arg\max_{\mathcal{A}\subset\mathbb{D}} E(f,\mathcal{A}).$$

The existence of the minimum of $A(f,\mathcal{A})$ and, correspondingly, the maximum of $E(f,\mathcal{A})$, has long been proved ([10]). The relation between n-Blaschke form and n-order rational approximation was specified in [12]. Since the methods and algorithms for them are similar, in this paper we focus on only n-best Blaschke forms, whose properties are discussed in [7,9–12].

Rational approximation is of great significance in both pure and applied mathematics. As an example, in system identification, one wishes to approximate system functions by rational functions.

Although the study of the n-best rational approximation has a long history [1,6,7,10], practical algorithms for finding the approximation are still under research. In literature, Baratchart's group in [2,4] proposed the method based on the second derivative test, treating the coefficients of the polynomial q as parameters. Qian's group proposed the adaptive Fourier decomposition algorithm (AFD) [11] and its improvement *cyclic AFD* (CAFD) [12], which used the poles of the approximative rational function as parameters. In practice, a search scheme is created to find the best tuple over the rectangular grids ([11,12]).

In a pole-tuple search algorithm, a main issue is how to ensure accuracy of the best tuple within an acceptable search time. Assume that the values $f(e^{it_j})$, $1 \le j \le J$, are known, and the search is taken over an n-dimensional tensor product net $\mathcal{M}^n$, where the one-dimensional rectangular ϵ-net $\mathcal{M} \subset \mathbb{D}$, has $N \times M$ nodes. Then an exhaustive search on $\mathcal{M}^n$ needs $O((JNM)^n)$ times of operations, which is unpractical when the approximation degree n is large. To reduce the computational cost, Qian introduced *coordinate maximum* [12, Definition 1] and proved that it is identical with the best n-tuple if the target function satisfies a certain condition. Based on this fact, AFD suggests n rounds of coordinate-by-coordinate search on $\mathcal{M}$. In each round, it finds the maximum for only a variable over $\mathcal{M}$: $\hat{b}_j = \arg\max_{a_j \subset \mathcal{M}} E(f, \mathcal{A})$, $1 \le j \le n$. Therefore, AFD needs only $O(JNM)$ operations, dramatically reducing the time used in an exhaustive search. The nature of AFD is finding §n§-best parameters not simultaneously, but sequentially. CAFD improves AFD by running several cycles of AFD till the coordinate maximum on $\mathcal{M}^n$ is obtained, which provides a practical algorithm for finding the best n-tuple. In CAFD the accuracy of the best n-tuple is significantly effected by the grid gap ϵ and the number J. In a certain sense, to improve the accuracy of CAFD, one must choose a smaller ϵ and a larger J, both of which would increase the time cost.

In this paper we apply complex gradient descent (CGD) method in the optimization of $E(f, \mathcal{A})$. Theoretically, CGD method can exactly locate the best n tuple in $\mathbb{D}^n$, breaking through the limitation of the grid gap. We initialize the CGD using either a randomly selected n-tuple or an n-tuple found by CAFD. Furthermore, we utilize the polar-type mesh grids in CAFD, which enables us to employ the fast Fourier transformation to accelerate the search. To clarify the notations, we denote by CAFDr and CAFDp the CAFD with rectangular mesh grids and polar-type mesh grids, respectively, and denoted by CAFD-CGD the CGD method with an initial n-tuple found by CAFDp.

The paper is organized as follows: In Section 2, we develop the *complex gradient descent algorithm* (CGD). In Section 3, we study the uniqueness of the best n-tuple, followed by the introduction of polar-type mesh grids and the algorithm of CAFD-CGD. In the last section, we give several illustrative examples to show the performance of the proposed methods and the comparison of CAFDr, CAFDp, CGD, and CAFD-CGD.

2. Complex gradient descent algorithm (CGD)

We first introduce some notions and notations. For $\mathbf{z} = [z_1, \cdots, z_n]^T \in \mathbb{C}^n$, its conjugate is denoted by $\bar{\mathbf{z}} = [\bar{z}_1, \cdots, \bar{z}_n]^T \in \mathbb{C}^n$. Write $\mathbf{z} = \mathbf{x} + i\mathbf{y}$, $\mathbf{r} = [\mathbf{x}, \mathbf{y}]$, and $\mathbf{c} = [\mathbf{z}, \bar{\mathbf{z}}]$. Hence, a complex function $f(\mathbf{z}) : \mathbb{C}^n \to \mathbb{C}$, with a little abuse of notation, has the following different forms:

$$f(\mathbf{z}) = f(\mathbf{z}, \bar{\mathbf{z}}) = f(\mathbf{c}) = f(\mathbf{x}, \mathbf{y}) = f(\mathbf{r}).$$

As usual, we define the cogradient operator by $\frac{\partial}{\partial \mathbf{z}} = \left[\frac{\partial}{\partial z_1}, \frac{\partial}{\partial z_2}, \cdots, \frac{\partial}{\partial z_n}\right]$, the conjugate cogradient operator by $\frac{\partial}{\partial \bar{\mathbf{z}}} = \left[\frac{\partial}{\partial \bar{z}_1}, \frac{\partial}{\partial \bar{z}_2}, \cdots, \frac{\partial}{\partial \bar{z}_n}\right]$, and the gradient of a differentiable function $f(\mathbf{z}, \bar{\mathbf{z}})$ by $\nabla_{\mathbf{z}} f = \left(\frac{\partial f}{\partial \mathbf{z}}\right)^H$, where $(\cdot)^H$ denotes the Hermitian transpose. We also denote by $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^H \mathbf{b}$ the inner product of two complex n-vectors $\mathbf{a}, \mathbf{b} \in \mathbb{C}^n$.

We now assume that $\mathbf{a} \in \mathbb{C}^n$ is a local minimal-value point of a real-valued function $g(\mathbf{z})$. Let $\mathbf{a}_0$ be the initial guess of $\mathbf{a}$, which resides in a neighborhood of $\mathbf{a}$. In a gradient descent method, $\mathbf{a}$ is found as the limit of the sequence of $(\mathbf{a}_k)$:

$$\mathbf{a}_{k+1} = \mathbf{a}_k - t_k \nabla g(\mathbf{a}_k), \quad k = 0, 1, 2, \cdots, \tag{2}$$

where ∇g in (2) is Lipschitz continuous with constant $L > 0$, i.e.,

$$\|\nabla g(\mathbf{a}) - \nabla g(\mathbf{b})\| \le L \|\mathbf{a} - \mathbf{b}\|.$$

In this paper, we adopt *backtracking line search*, in which a fixed β, $0 < \beta < 1$, is employed for formulating t_k by $t_k = \beta t_{k-1}$, $t_1 = 1$. Then the iteration is terminated when

$$g(\mathbf{a}_k - t_k \nabla g(\mathbf{a}_k)) > g(\mathbf{a}_k) - \frac{t_k}{2}\|\nabla g(\mathbf{a}_k)\|^2. \tag{3}$$

We call the method above *Complex Gradient Descent Method (CGD)*. The convergence theorems of the real gradient descent methods in [3,5,8] can also be applied for CGD.

In our algorithm, we set $g = -E$ in (2), where $E(\mathbf{a}) = E(f, \mathcal{A})$ is the energy function of f at $\mathcal{A}$. Besides, to guarantee that after each iterative step the new n-tuple $\mathbf{a}_k$ is still in $\mathbb{D}^n$, t_k in (3) must satisfy

$$\mathbf{a}_k + t_k \nabla E(\mathbf{a}_k) \in \mathcal{N}_{\mathbf{a}_k} \cap \mathbb{D}^n,$$

where $\mathcal{N}_{\mathbf{a}_k} = \{\mathbf{z}; \|(a_k)_j - z_j\| < r,\ 1 \le j \le n\}$.

An effective formulation of the gradient $-\nabla E (= \nabla A)$ is essential for CGD. We establish it as follows: Write

$$e_a(z) = \frac{\sqrt{1-|a|^2}}{1-\bar{a}z}, \quad a \in \mathbb{D},$$

and let P_ℓ, $l = 1, \cdots, n$, be the permutation of the index set $\{1, 2, \cdots, n\}$ such that $P_\ell(n) = \ell$. For a given analytic function $f \in H^2$, we inductively define n functions $f_{P_\ell(j)}$, $1 \le j \le n$, by

$$\begin{aligned} f_{P_\ell(1)}(z) &= f(z), \\ f_{P_\ell(j)}(z) &= \frac{1 - z\bar{a}_{P_\ell(j-1)}}{z - a_{P_\ell(j-1)}} \left(f_{P_\ell(j-1)}(z) - \langle f_{P_\ell(j-1)}, e_{a_{P_\ell(j-1)}} \rangle e_{a_{P_\ell(j-1)}}(z) \right). \end{aligned} \tag{4}$$

It was proved that all $f_{P_\ell(j)}(z)$ are analytic in $\mathbb{D}$ [9]. Since the energy function $E(\mathbf{a})$ is invariant under the permutation, it has n different representations:

$$E(\mathbf{a}) = \sum_{j=1}^{n} \left(1 - |a_{P_\ell(j)}|^2\right) \left| f_{P_\ell(j)}(a_{P_\ell(j)}) \right|^2, \quad \ell = 1, \cdots, n. \tag{5}$$

Note that the variable $a_\ell = a_{P_\ell(n)}$ occurs in only the last term of the sum in (5). Since $f_{P_\ell(n)}$ is analytic, we have $\frac{\overline{f_{P_\ell(n)}}}{\partial z_\ell} = 0$, so that

$$\frac{\partial(-E(\mathbf{a}))}{\partial z_\ell} = \overline{f_{P_\ell(n)}(a_\ell)} \left(\overline{a_\ell} f_{P_\ell(n)}(a_\ell) - (1 - |a_\ell|^2) f'_{P_\ell(n)}(a_\ell) \right), \quad \ell = 1, \cdots, n, \tag{6}$$

where $f'_{P_\ell(n)}(z)$ can be computed by

$$f'_{P_\ell(n)}(z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{f_{P_\ell(n)}(e^{i\theta}) e^{-i\theta}}{(1 - ze^{-i\theta})^2} d\theta.$$

Remark 1. The permutation P_ℓ is not unique. In practice, we choose $P_\ell = P^\ell$, where P is the 1-*shift* permutation: $P(1, 2, \cdots, n) = (2, \cdots, n, 1)$.

We present the pseudo code of CGD in Algorithm 1. Its inputs consist of the function $f \in H^2$, an initial n-tuple $\mathbf{a}$, a tolerance ε, and the parameter β in the backtracking line search. Here, the initial n-tuple $\mathbf{a}$ can be chosen randomly or found by Algorithm 3 (CAFDp) in the next section. When the tolerance condition $\|\nabla E(\mathbf{a})\|^2 < \varepsilon$ holds, the algorithm terminates.

Remark 2. In fact, in practice (see Section 4) we set $\beta = \frac{1}{||f||^2}$ if $||f||^2 > 1$; or $\beta = \frac{1}{2^K||f||^2}$ if $||f||^2 \le 1$, where K is an integer such that $2^K||f||^2 > 1$. Such a setting of β in this paper is just to ensure that $\mathbf{a}_k \in \mathbb{D}^n$ at each step, but we note that this is not an optimal choice of β. We note that the parameters t_k, which are essentially determined by β, greatly affect the accuracy and the effectiveness of Algorithm 1 (CGD).

Algorithm 1 CGD: Complex gradient descent algorithm for finding the best tuple.

```
Require: $\mathbf{a}, f, \beta$, neighbor size $r$, and the tolerance $\varepsilon$.
 1: Compute $\nabla E(\mathbf{a})$ using $f$ and $\mathbf{a}$.
 2: while $\|\nabla E(\mathbf{a})\|^2 > \varepsilon$ do
 3:     Find $s_1 > 0, s_2 > 0$ such that $\|\mathbf{a} + s_1\nabla E(\mathbf{a})\|_\infty = 1$ and $\|s_2\nabla E(\mathbf{a})\|_\infty = r$. Set $s = min(s_1, s_2)$.
 4:     Compute $\mathbf{c} = \mathbf{a} + s\nabla E(\mathbf{a})$.
 5:     while $E(\mathbf{c}) < E(\mathbf{a}) + \frac{s}{2}\|\nabla E(\mathbf{a}))\|^2$ do
 6:         Update $s$: $s = \beta s$.
 7:         Update $\mathbf{c}$: $\mathbf{c} = \mathbf{a} + s\nabla E(\mathbf{a})$.
 8:     end while
 9:     Update $\mathbf{a}$: $\mathbf{a} = \mathbf{c}$.
10:     Re-compute $\nabla E(\mathbf{a})$.
11: end while
12: Set the output: $\mathbf{b} = \mathbf{a}$.
Ensure: $\mathbf{b}$
```

3. CAFD on a polar mesh grid

A main difficulty for best n-tuple search algorithms is that the energy function $E(\mathbf{a})$ lacks the uniqueness of critical points. Therefore, a coordinate maximum of $E(\mathbf{a})$ is not necessarily a global one. Besides, lack of the uniqueness of the critical points discourages the approach of globally convex optimization.

Example 3.1. Let $f(z) = z^k$, $k \in \mathbb{N}$. We consider its Blaschke-form approximation of degree 1. By (5), the error function $A(\mathbf{a})$ has the form of

$$||f||^2 - (1 - |a|^2)|f(a)|^2 = 1 - (1 - |a|^2)|a|^{2k},$$

which reaches the global minimum when $|a| = \sqrt{\frac{k}{k+1}}$.

The example shows that, when the approximation degree is one, the global minimum may occur on a manifold.

When the approximation degree $n > 1$, due to the invariance of the energy function $E(\mathbf{z})$ under the permutations of $\mathbf{z}$, $E(\mathbf{z})$ reaches its global maximum at least at $n!$ distinct points in $\mathbb{D}^n$, of which each is a permutation of another one. This fact totally denies the uniqueness of the n-best tuple. When $n > 1$, we have the following result on the set of best n-tuples.

Theorem 1. *When $n > 1$, the set of best n-tuples of the Blaschke-form approximation of degree n for a function $f \in H^2$ cannot contain a continuous curve in $\mathbb{D}^n$.*

Proof. Denote by S the set of the best n-tuples of the approximation. Then $\nabla E(\mathbf{a}) = 0$ for any $\mathbf{a} \in S$. By (6), $\frac{\partial E(\mathbf{a})}{\partial z_k} = 0$ if and only if either $f_{P_k(n)}(a_k) = 0$, or

$$\overline{a_k} f_{P_k(n)}(a_k) - (1 - |a_k|^2) f'_{P_k(n)}(a_k) = 0. \tag{7}$$

By (5), $E(\mathbf{a})$ does not have the maximum value when $f_{P_k(n)}(a_k) = 0$. Therefore, if $\mathbf{a} \in S$, (7) must hold for each k.

Assume there is a continuous curve $\Gamma \subset S$. Denote by Γ_k the projection of Γ on the k-th coordinate (complex) plane. Then at least one Γ_k is a continuous curve in $\mathbb{D}$. Assume $k = n$ and $P_n = I$. Then by (7),

$$\overline{a_n} f_n(a_n) - (1 - |a_n|^2) f_n'(a_n) = 0, \quad \forall a_n \in \Gamma_n.$$

It follows that

$$\frac{z f_n'(z)}{f_n(z)} = \frac{|z|^2}{1-|z|^2}, \quad \forall z \in \Gamma_n.$$

Hence, $g(z) = \frac{z f_n'(z)}{f_n(z)}$ is real-valued and has no poles on Γ_n. Therefore, $g(z)$ is analytic and $g(z) = c$ on $\mathcal{O}$, where $\mathcal{O}$ is a connected domain with $\Gamma_n \subset \mathcal{O}$. It follows that $f_n(z) = rz^c$ on $\mathbb{D}$. However, by the formula (4), $f_n(z)$ has at least n distinct zeros in $\mathbb{D}$, which leads to a contradiction. The proof is complete. □

According to Theorem 1, we make the following conjecture:

Conjecture 1. *When $n > 1$, the energy function $E(\mathbf{z})$ in (5) has exactly $n!$ best n-tuples, i.e., all of its best n-tuples are permutations of a single point in $\mathbb{D}^n$.*

Assume the conjecture holds. By the similar argument given in the proof of [12, Corollary 4], we can confirm that a coordinate maximum point for $E(\mathbf{z})$ is also a global maximum point. In this case, on given grids, we can employ CAFD to find an n-tuple, which is nearest to a best n-tuple.

Considering the geometric structure of the unit disk, we suggest making the CAFD search over a *polar grid set.*

3.1. Fast evaluation algorithm (FEVAL)

Definition 1. Let $M > 1$ and $N > 1$ be two positive integers, and $\epsilon = \frac{1}{M}$, $\delta = \frac{1}{N}$. A set of polar ϵ–δ grids on $\mathbb{D}$ is the node set

$$\mathcal{G} = \{z;\ z = m\epsilon e^{2n\delta\pi i},\ 1 \le m < M, 1 \le n \le N\}.$$

To find an initial n-tuple for CAFD-CGD algorithm, we employ CAFDp, which is carried out over polar mesh grids.

Evaluating $f_\ell(z)$ in (5), (which needs the evaluations of $\langle f_\ell, e_z\rangle$ in (4)) for all z in a mesh grid set costs most time in CAFD. For instance, over an $N \times M$ rectangular mesh grid set $\mathcal{M}$, we need $O(JNM)$ operations for the evaluations assuming the sample values $f(e^{it_j})$, $1 \le j \le J$, are given. Adopting Fast Fourier Transform (FFT), the following FEVAL algorithm estimates the evaluations over a polar mesh grids set $\mathcal{G}$ using only $O(NM \log N)$ operations for $J = N$. Recall that the Fourier coefficients are defined as

$$\hat{f}(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(e^{it}) e^{-int}\, dt, \quad f \in H^2.$$

For $f \in H^2$ and $z = re^{it}$, $0 < r < 1$, by the definition of H^2 we have

$$\langle f, e_z\rangle = \sqrt{1-r^2} f(z) = \sqrt{1-r^2} \sum_{k=0}^{\infty} r^k \hat{f}(k) e^{ikt}. \tag{8}$$

For an infinite sequence $\mathbf{c} = (c_0, c_1, \cdots, c_k, \cdots) \in \ell^2$, we define the scaling operator $\mathfrak{R}_r : \ell^2 \to \ell^2 : \mathbf{d} = \mathfrak{R}_r(\mathbf{c})$, where $d_k = r^k c_k$, $k = 0, 1, 2, \cdots$. We also denote by $\mathfrak{F} : H^2 \to \ell^2$: $\mathfrak{F}(f(e^{it})) = \left(\hat{f}(k)\right)_{k=0}^{\infty}$ the discrete Fourier Transform. Then its inverse $\mathfrak{F}^{-1}$ has the form $\mathfrak{F}^{-1}(\hat{f}) = \sum_{k=0}^{\infty} \hat{f}(k)e^{ikt} = f(e^{it})$. Thus, we can write $f(z)$ as $f(re^{it}) = \mathfrak{F}^{-1} \circ \mathfrak{R}_r \circ \mathfrak{F}(f(e^{it}))$.

Based on (8), we present the pseudo code of FEVAL in Algorithm 2, where $\mathbf{f} = [f_1, \cdots, f_N]^T$ is the vector of sample values of $f \in H^2$ at $T = \{t_1, \cdots, t_N\}, t_j = e^{\frac{2j\pi i}{N}}$; $F(m, n) \approx \langle f, e_z \rangle, z = (m\epsilon)e^{\frac{2n\pi i}{N}} \in \mathcal{G}$; and $\mathbf{F} = [F(m, n)]_{m,n=1}^{M,N}$.

Algorithm 2 FEVAL: Evaluating $\langle f, e_z \rangle$ over the mesh grid set $\mathcal{G}$.

```
Require: $M, N, \epsilon$ and $\mathbf{f}$.
1: Initialization: Compute $\hat{f}^{(0)} = \text{FFT}(\mathbf{f})$ and scale it to $\hat{f}^{(1)} = \mathfrak{R}_\epsilon(\hat{f}^{(0)})$.
2: Output the first row of $\mathbf{F}_e$: $F(1,:) = \sqrt{1-\epsilon^2}\text{IFFT}(\hat{f}^{(1)})$.
3: for  m = 2 : M do
4:     Compute the Fourier coefficient sequence on the $m$-th circle: $\hat{f}^{(m)} = \mathfrak{R}_{m\epsilon}(\hat{f}^{(1)})$.
5:     Output the $m$-th row of $\mathbf{F}_e$: $F(m,:) = \sqrt{1-m^2\epsilon^2}\text{IFFT}(\hat{f}^{(m)})$.
6: end for
Ensure: F
```

3.2. CAFDp

In CAFDp, the search starts from a randomly chosen n-vector $\mathbf{a} \in \mathcal{G}^n$. Fixing $a_1, \cdots, a_{n-1}$, the algorithm first finds the maximal-value point $\tilde{a}_n$ for $|\langle f_n, e_z \rangle|$ over the grid set $\mathcal{G}$. Then, after a_n is replaced by $\tilde{a}_n$ and $\mathbf{a}$ is permuted by the 1-shift permutation, the search process above will be repeated till no replacement can be made. We present CAFDp in the following:

Algorithm 3 CAFDp: CAFDp for finding the best tuple.

```
Require: $\mathbf{a}, f, \epsilon, \delta$, and the tolerance $\eta$.
 1: Create the polar-type $(\epsilon, \delta)$-grid set $\mathcal{G}$ on $\mathbb{D}$.
 2: Randomly select $\mathbf{a} = [a_1, \cdots, a_n] \in \mathbb{D}^n$ as the starting $n$-tuple for the algorithm.
 3: Create the function $f_n$ and compute the partial energy $V = |\langle f_n, e_{a_n} \rangle|$.
 4: Initialize parameter for WHILE loop and set the WHILE-LOOP light $s = 1$.
 5: while $s \neq 0$ do
 6:     Reset $s = 0$.
 7:     for $j = 1 \to n$ do
 8:         Compute $V_t = \max_{z \in \mathcal{G}} |\langle f_n, e_z \rangle|, a_t = \arg\max_{z \in \mathcal{G}} |\langle f_n, e_z \rangle|$.
 9:         if $V_t > V + \eta$ then
10:             Update: $a_n = a_t, V = V_t, s = s + 1$.
11:         end if
12:         Permute $\mathbf{a}$ using $\mathbf{a} = P\mathbf{a} = [a_n, a_1, \cdots, a_{n-1}]$.
13:         Update $f_n$ based on the new $\mathbf{a}$.
14:         Update $V = |\langle f_n, e_{a_n} \rangle|$.
15:     end for
16: end while
Ensure: a
```

In Algorithm 3, the inputs consist of a function $f \in H^2$, a randomly selected n-tuple $\mathbf{a} \in \mathbb{D}^n$, two parameters $\epsilon > 0$ and $\delta > 0$ for polar grid set $\mathcal{G}$, and a tolerance $\eta > 0$. It outputs an n-tuple.

4. Illustrative examples

In this section, we illustrate the accuracy and effectiveness of CGD and CAFD-CGD algorithms in approximating the functions in H^2 and in recovering the tuple of a Blaschke form. In the experiments we compare the four methods: CAFDr, CAFDp, CGD, and CAFD-CGD.

For making fair comparisons, we sample the functions at J equidistant points on the unit circle for all experiments, where J would be respectively equal to 2^6, 2^7, 2^8 and 2^9. Moreover, we use the same randomly

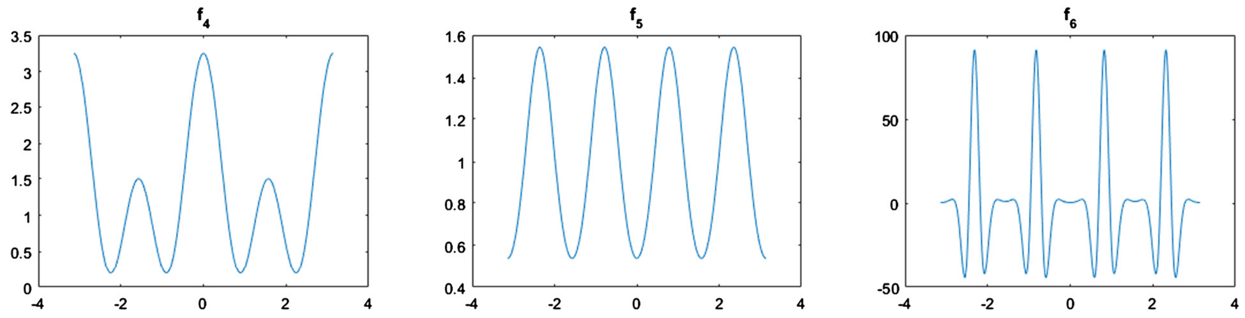

Fig. 1. Real parts of $1+\tau^2+\tau^4+1/(3+\tau^2), \cos\tau^2, \frac{\cos 6\tau^2}{2+\tau^2}$.

Table 1
Comparison of CAFDr, CAFDp, CGD and CAFD-CGD in approximation (Example 4.1).

Sample	2^6 (64)		2^7 (128)		2^8 (256)		2^9 (512)	
	Err	Ord	Err	Ord	Err	Ord	Err	Ord
$1+\tau^2+\tau^4+\frac{1}{3+\tau^2}$								
CAFDr	8.0×10^{-4}	5	6.8×10^{-4}	5	6.9×10^{-4}	5	7.0×10^{-4}	5
CAFDp	0.0119	5	0.0073	5	0.0034	5	0.0039	5
CGD	6.9×10^{-7}	5	7.0×10^{-7}	5	7.0×10^{-7}	5	7.1×10^{-7}	5
CAFD-CGD	1.6×10^{-6}	5	8.8×10^{-7}	5	1.3×10^{-6}	5	1.5×10^{-6}	5
$\cos\tau^2$								
CAFDr	9.5×10^{-4}	5	0.0011	5	0.0011	5	0.0011	5
CAFDp	0.0016	5	0.0015	5	8.8×10^{-4}	5	0.0013	5
CGD	3.5×10^{-6}	5	3.5×10^{-6}	5	3.4×10^{-6}	5	3.4×10^{-6}	5
CAFD-CGD	3.5×10^{-6}	5	3.5×10^{-6}	5	3.5×10^{-6}	5	3.4×10^{-6}	5
$\frac{\cos 6\tau^2}{(2+\tau^2)}$								
CAFDr	0.0121	14	0.0024	14	0.0019	14	0.0019	14
CAFDp	0.0065	14	0.0056	14	0.0034	14	0.0050	14
CGD	0.0063	14	3.8×10^{-5}	14	3.6×10^{-5}	14	3.6×10^{-5}	14
CAFD-CGD	0.0072	14	3.5×10^{-5}	14	3.5×10^{-5}	14	3.3×10^{-5}	14

selected initial n-tuple in each experiment for all the methods. In CGD, β is set as in Remark 2. In CAFDp, we set $\epsilon = 0.033$ and $\delta = \frac{1}{J}$, which produce $\frac{J}{\epsilon}$ nodes on the polar mesh grid set. In CAFDr, we set the grid gap to 0.02, which produces 7089 nodes on the rectangular mesh grid set.

To our experience, usually the accuracies of CAFDr and CAFDp could not be improved significantly even smaller grip gaps are used. Note that we apply only CAFDp to find an initial tuple in our CAFD-CGD algorithm. We also note that the step size t_k in (2) is crucial for accuracy and effectiveness of CGD and CAFD-CGD.

For convenience, in all examples we write $\tau = e^{it}$, where t is equidistantly sampled on $[-\pi, \pi)$.

4.1. Approximating functions in H^2

In this subsection, we illustrate an example in approximating functions in H^2. The graphs show only the real parts of the complex-valued functions.

Example 4.1. In this example, we consider three functions: $f_1(\tau) = 1+\tau^2+\tau^4+1/(3+\tau^2)$, $f_2(\tau) = \cos\tau^2$ and $f_3(\tau) = \frac{\cos 6\tau^2}{2+\tau^2}$, where we use 5-order, 5-order and 14-order to approximate them, respectively. The results are shown in Fig. 1, and the comparison is given in Table 1. In all cases, the performances of CGD and CAFD-CGD are better than those of CAFDr and CAFDp. As the number of samples increases, the accuracies of all methods are not significantly improves.

Table 2
Comparison of CAFDr, CGD and CAFD-CGD in approximation and tuple-distance (f_4). "Err" is the L^2-relative error, and "TDis" is the tuple distance.

Samples	2^6 (64)		2^7 (128)		2^8 (256)		2^9 (512)	
	Err	TDis	Err	TDis	Err	TDis	Err	TDis
CAFDr	0.1292	1.3426	0.0522	1.0725	0.0022	0.2128	0.0022	0.2128
CGD	0.2463	1.5678	0.1017	1.3207	1.1×10^{-8}	0.0010	7.0×10^{-11}	0.0001
CAFD-CGD	0.0433	0.8920	0.0592	0.8036	1.1×10^{-8}	0.0010	1.6×10^{-8}	0.0001

4.2. Recovering tuple of Blaschke form

In this subsection, we recover n-tuple $\mathbf{b} = [b_1, \cdots, b_n]$ of the n-Blaschke form

$$f(\tau) = \sum_{k=1}^{n} c_k B_{b_1,\cdots,b_k}(\tau). \tag{9}$$

When $\mathbf{b}$ is recovered, the coefficient vector $\mathbf{c} = [c_1, \cdots, c_n]$ in (9) can be computed by (1). To recover an n-tuple, the approximation degree has to be n too. Because we only input a finite number of values of $f(\tau)$ on the unit circle in a numerical algorithm, its output tuple will be deviated with recovering errors. We measure such an error by the following tuple distance, which is defined as follows: Let $\mathcal{P}$ be the set of all permutations on a vector $\mathbf{u} = [u_1, \cdots, u_n]$. The distance between two n-tuples $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{D}^n$ is

$$d(\mathbf{u}, \mathbf{v}) = \min_{P \in \mathcal{P}} \|P\mathbf{u} - \mathbf{v}\|.$$

Because we employ CAFDp to initialize the tuple of CAFD-CGD only, we exclude it in the algorithm comparisons in this subsection.

Example 4.2. In the following, we give the experimental results of a Blaschke form functions f_4. The results of f_4 are given as follows. Note that f_4 is the 7-Blaschke form as given in (9), where

$$\begin{aligned}\mathbf{b} = [&-0.3424 + 0.7947i, 0.6862 + 0.0548i, -0.0462 - 0.7202i, 0.8778 + 0.0707i, 0.7157 + 0.4333i,\\ &0.6468 - 0.1387i, -0.3625 + 0.7803i].\\ \mathbf{c} = [&0.5405 - 0.5808i, -1.4449 + 0.8751i, -0.9677 + 1.3954i, 0.2021 + 0.3210i, -0.3479 + 1.6234i,\\ &1.2901 + 1.0624i, 1.3412 + 0.2141i].\end{aligned}$$

The initial guess 7-tuple for this example is chosen at random:

$$\begin{aligned}\mathbf{a}_0 = [&-0.0341 + 0.3272i, 0.1736 + 0.8756i, 0.2296 - 0.3632i, -0.3373 - 0.4516i, -0.0290 + 0.0595i,\\ &-0.6847 + 0.0060i, -0.3766 - 0.0745i].\end{aligned}$$

The comparisons of L^2-relative error and the tuple distance are given in Table 2, and the 7-tuples recovered by CAFDr, CGD, and CAFD-CGD are shown in Fig. 2. From Table 2 and Fig. 2 we see that the performances of CGD and CAFD-CGD beat CAFDr, and the recovering errors of CGD and CAFD-CGD significantly decrease as the number of samples increases.

4.3. Conclusion

Our experiments confirm the improvement of the proposed methods (CGD and CAFD-CGD) both in L^2 approximation and best n-tuple recovering, comparing with CAFDr and CAFDp. In most cases, the

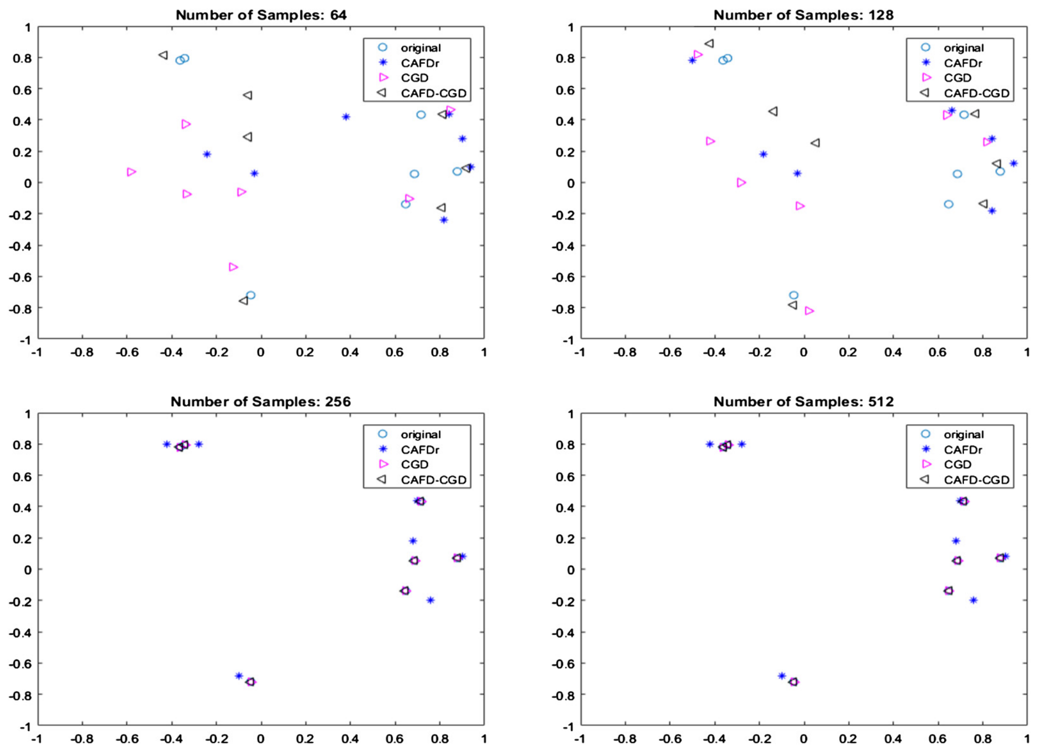

Fig. 2. Example 4.2.

performances of CGD and CAFD-CGD are similar. The experiments indicate the importance of the cardinality of given samples in the proposed algorithms. It seems that 2^8 is a suitable cardinality for CGD and CAFD-CGD. Besides, the parameter t_k in (2) is crucial for CGD because it greatly influences the accuracy and effectiveness of CGD and CAFD-CGD. How to choose a suitable t_k at each iterative step of CGD and CAFD-CGD needs a further study.

References

[1] L. Baratchart, A remark on uniqueness of best rational approximation of degree 1 in L^2 of the circle, Electron. Trans. Numer. Anal. 25 (2006) 54–66.
[2] L. Baratchart, M. Cardelli, M. Olivi, Identification and rational L^2 approximation, a gradient algorithm, Automatica 27 (1991) 413–418.
[3] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[4] P. Fulcheri, M. Olivi, Matrix rational H^2 approximation: a gradient algorithm based on Schur analysis, SIAM J. Control Optim. 36 (6) (1998) 2103–2127.
[5] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009.
[6] J.L. Walsh, Interpolation and Approximation by Rational Functions in the Complex Domain, 5th edition, Colloquium Publications, American Mathematical Society, July 1935.
[7] W. Mi, T. Qian, F. Wan, A fast adaptive model reduction method based on Takenaka–Malmquist systems, Systems Control Lett. 61 (1) (2012) 223–230.
[8] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Kluwer Academic Publishers, 2004.
[9] T. Qian, Yanbo Wang, Adaptive Fourier series – a variation of greedy algorithm, Adv. Comput. Math. 34 (3) (2011) 279–293.
[10] T. Qian, E. Wegert, Optimal approximation by Blaschke forms, Complex Var. Elliptic Equ. 58 (1) (2013) 123–133.
[11] T. Qian, Liming Zhang, Zhi-Xiong Li, Algorithm of adaptive Fourier decomposition, IEEE Trans. Signal Process. 59 (12) (2011) 5899–5902.
[12] T. Qian, Cyclic AFD algorithm for best approximation by rational functions of given order, Math. Methods Appl. Sci. 37 (6) (2014) 846–859.