MATHEMATICS OF COMPUTATION
Volume 80, Number 276, October 2011, Pages 2169–2184
S 0025-5718(2011)02484-1
Article electronically published on March 31, 2011

A POLYNOMIAL INTERPOLATION PROCESS AT QUASI-CHEBYSHEV NODES WITH THE FFT

HIROSHI SUGIURA AND TAKEMITSU HASEGAWA

ABSTRACT. Interpolation polynomial p_n at the Chebyshev nodes $\cos \pi j/n$ $(0 \le j \le n)$ for smooth functions is known to converge fast as $n \to \infty$. The sequence $\{p_n\}$ is constructed recursively and efficiently in $O(n \log_2 n)$ flops for each p_n by using the FFT, where n is increased geometrically, $n = 2^i$ $(i = 2, 3, \dots)$, until an estimated error is within a given tolerance of ε. This sequence $\{2^j\}$, however, grows too fast to get p_n of proper n, often a much higher accuracy than ε being achieved. To cope with this problem we present quasi-Chebyshev nodes (QCN) at which $\{p_n\}$ can be constructed efficiently in the same order of flops as in the Chebyshev nodes by using the FFT, but with n increasing at a slower rate. We search for the optimum set in the QCN that minimizes the maximum error of $\{p_n\}$. Numerical examples illustrate the error behavior of $\{p_n\}$ with the optimum nodes set obtained.

1. Introduction

Polynomial interpolation [8] of a given real-valued function $f(x)$ in terms of the Chebyshev polynomial $T_k(x) = \cos k\theta$, where we set $x = \cos\theta$,

$$p_n(x) = \sum_{k=0}^{n} a_k^n T_k(x), \quad -1 \le x \le 1, \tag{1.1}$$

is often used in many problems of scientific computing; see Battles and Trefethen [1] Boyd [3], Boyd and Gally [4], Mason and Handscomb [12], and Reddy and Weideman [16]. The Chebyshev interpolation $p_n(x)$ (1.1) that interpolates $f(x)$ at the Chebyshev nodes $x_j^n = \cos \pi j/n$ $(0 \le j \le n)$ [1, 15, 21], or the Chebyshev points of the second kind (Berrut and Trefethen [2]) is particularly useful in many applications, say, in integration (the Clenshaw-Curtis rule [6, 21, 22, 23]) and differentiation [16].

The Chebyshev interpolation has the following advantages with some problems. The Chebyshev coefficients a_k^n are efficiently evaluated by using the FFT (fast Fourier transform) [7], or Fourier cosine transform [5, 9, 15]. For analytic functions the polynomial p_n converges fast; indeed $|a_k^n| = O(\rho^{-k})$ $(\rho > 1)$ [21]. The interpolation error can be estimated by using the last several, say m, a_k^n $(n - m < k \le n)$ [1, 6, 17]. Since the Chebyshev nodes $\{x_j^{2n}\}_{j=0}^{2n} = \{x_j^n\}_{j=0}^{n} \cup \{\cos \pi(2j-1)/2n\}_{j=1}^{n}$ for p_{2n} as shown in section 2, the sequence of $\{p_n\}$ can be recursively constructed by doubling n, $n = 2^i$ $(i = 2, 3, \dots)$ until the required accuracy is achieved [14].

Received by the editor March 23, 2009 and, in revised form September 21, 2010.
2010 *Mathematics Subject Classification.* Primary 65D05, 41A10; Secondary 42A15.
Key words and phrases. Chebyshev interpolation, Chebyshev nodes, quasi-Chebyshev nodes, Chinese remainder theorem, FFT, error estimate, computational complexity, fast algorithm.

However, this sequence $n = 2^i$ grows so fast that one might often obtain an interpolation polynomial with much higher accuracy than the required one, particularly when the number of function evaluations $n+1$ required is high.

To cope with the above-mentioned problem, Hasegawa, Torii and Sugiura (HTS) [11] present a modified set of the Chebyshev nodes (a van der Corput sequence) at which a sequence of p_n (1.1) is constructed recursively with n increasing like $n = 3 \times 2^i, 4 \times 2^i, 5 \times 2^i$ $(i = 1, 2, \dots)$, namely at an average rate $\sqrt[3]{2}$, as well as with the FFT, but interpolation errors being a little worse. Sugiura and Torii [19] generalize the above HTS sequence [11] to make a complex polynomial interpolation at quasi-equidistributed nodes on the unit disk on the complex plane.

The purpose of this paper is to extend the HTS sequence [11] to more general ones by presenting new sets of nodes, or quasi-Chebyshev nodes (QCN) on $[-1, 1]$, a real version of the above quasi-equidistributed nodes [19]. We search for the optimum set among those QCN that give the sequence of interpolation polynomials $\{p_n\}$ with the maximum error minimized. HTS's sequence [11] proves to be near but not optimum among those of the average rate $\sqrt[3]{2}$. In recursively updating p_n to generate the sequence $\{p_n\}$ we also make use of the Chinese remainder theorem (CRT) [13], which is nicely used in the polynomial approach to fast Fourier cosine (FCT) and sine transforms by Steidl and Tasche [18]. We discuss the computational complexity in the fast algorithm to construct p_n and the error analysis.

This paper is organized as follows. In section 2 we define the quasi-Chebyshev interpolation at the QCN. In section 3 we review the modulo computation with the Chebyshev polynomials. In section 4 we give a fast algorithm for quasi-Chebyshev interpolation with the FFT in details before discussing the computational complexity. In section 5 we discuss the error of interpolation polynomials. In section 6 we search for optimum sequences of the QCN. In section 7 we outline an algorithm for automatic interpolation with the given tolerance. Section 8 gives numerical examples to illustrate the behavior of interpolation errors.

2. Quasi-Chebyshev interpolation

We begin by reviewing the Chebyshev interpolation. For a positive integer n let $V_{n+1}(x)$ be a polynomial defined by

$$V_{n+1}(x) = \{T_{n+1}(x) - T_{n-1}(x)\}/2. \tag{2.1}$$

Then the Chebyshev nodes $x_j^n = \cos(\pi j/n)$ are zeros of $V_{n+1}(x)$. The coefficients a_k^n of the Chebyshev interpolation p_n for f are determined so that $p_n \equiv f \pmod{V_{n+1}}$. The CRT [13, p. 27], [18] is useful to construct the sequence of interpolation polynomials recursively. For example, assume that $n = 2^j$ and p_n is given. Let q_{n-1} be a polynomial satisfying

$$q_{n-1} := \sum_{k=0}^{n-1} b_k T_k(x) \equiv f \pmod{T_n}. \tag{2.2}$$

Then since $V_{2n+1}(x) = 2V_{n+1}(x)T_n(x)$ and $V_{n+1}(x)$ and $T_n(x)$ are relatively prime, the coefficients a_k^{2n} of the polynomial $p_{2n} \equiv f \pmod{V_{2n+1}}$ (1.1) are obtained from a_k^n of p_n and b_k of q_{n-1} so that $p_{2n} \equiv p_n \pmod{V_{n+1}}$ and $p_{2n} \equiv q_{n-1} \pmod{T_n}$. Note that b_k in (2.2) are efficiently evaluated by the FFT, particularly the FCT [18].

Now we define the quasi-Chebyshev nodes as follows. Let λ be a positive integer. Then we see that

$$T_\lambda(x) = 2^{\lambda-1} \prod_{\alpha\in A(\lambda)} (x-\alpha), \quad A(\lambda) := \{\cos\pi(2l-1)/(2\lambda) : 1 \le l \le \lambda\}. \tag{2.3}$$

Since $T_\lambda(T_n(x)) = T_{\lambda n}(x)$ we have from (2.3)

$$V_{2\lambda n+1}(x) = 2V_{\lambda n+1}(x)T_{\lambda n}(x) = 2^\lambda V_{\lambda n+1}(x) \prod_{\alpha\in A(\lambda)} \{T_n(x)-\alpha\}. \tag{2.4}$$

Definition 2.1 (quasi-Chebyshev nodes)**.** Let $B = \{\alpha_1, \dots, \alpha_\mu\}$ be a subset of $A(\lambda)$ (2.3) consisting of μ elements in $A(\lambda)$, where $0 \le \mu \le \lambda$. Let n be a positive integer. Let $W_n(B;x)$ be a polynomial defined by

$$W_n(B;x) = 2^\mu V_{\lambda n+1}(x) \prod_{l=1}^{\mu} \{T_n(x) - \alpha_l\}. \tag{2.5}$$

Then we define the quasi-Chebyshev nodes by the zeros of $W_n(B;x)$.

Note that if $B = \phi$ or $B = A(\lambda)$, then we have the Chebyshev nodes since $W_n(\phi;x) = V_{\lambda n+1}(x)$ or $W_n(A(\lambda);x) = V_{2\lambda n+1}(x)$, respectively.

Definition 2.2 (quasi-Chebyshev interpolation)**.** We define the quasi-Chebyshev interpolation $\Phi_n(B)f$ by

$$\Phi_n(B)f = \sum_{k=0}^{(\lambda+\mu)n} a_k^{(\lambda+\mu)n} T_k \equiv f \pmod{W_n(B)}. \tag{2.6}$$

Let $\{B_0, B_1, \dots, B_\iota\}$ $(0 < \iota \le \lambda)$ be a sequence such that

$$\phi = B_0 \subset B_1 \subset \cdots \subset B_\iota = A(\lambda). \tag{2.7}$$

Let $n = 2^j$ $(j \ge 0)$. Then by using the sequence of polynomials defined by

$$V_{\lambda n+1} = W_n(B_0), W_n(B_1), \dots, W_n(B_\iota) = W_{2n}(B_0) = V_{2\lambda n+1},$$

we have the sequence of interpolation polynomials $\{\Phi_n(B_0)f, \dots, \Phi_n(B_\iota)f\}$ such that

$$\Phi_n(B_i)f \equiv f \pmod{W_n(B_i)}, \quad i = 0, 1, \dots, \iota. \tag{2.8}$$

In the sequence $\{\Phi_n(B_i)f\}$ $(0 \le i < \iota,\ n = 2^j,\ j = 0, 1, \dots)$ the degree of the polynomial increases geometrically at an average rate $\sqrt[\iota]{2}$.

Example 2.3. The HTS sequence [11] is obtained by choosing $\lambda = 4$, $\iota = 3$ and

$$\begin{aligned}
&W_n(B_0) = V_{4n+1},\\
&W_n(B_1) = 2V_{4n+1}(T_n - \cos 3\pi/8),\\
&W_n(B_2) = 2W_n(B_1)(T_n - \cos 5\pi/8) = 2V_{4n+1}(T_{2n} - \cos 3\pi/4),\\
&W_n(B_3) = 2V_{4n+1}T_{4n} = V_{8n+1} = W_{2n}(B_0).
\end{aligned}$$

3. Modulo computation with the Chebyshev polynomials

The modulo computation on the Chebyshev polynomials plays an important role in constructing the sequence of interpolation polynomials with the CRT. We collect some relations on the Chebyshev polynomials; see [12, 18] for details.

The Chebyshev polynomial of the second kind $U_k(x)$ is defined by

$$U_k(x) = \sin(k+1)\theta/\sin\theta, \quad x = \cos\theta, \quad k \geq 0.$$

We define $U_{-1}(x) = 0$ and $U_{-k}(x) = -U_{k-2}(x)$ $(k \geq 2)$. Further, we define $T_{-k}(x) = T_k(x)$ $(k \geq 1)$. For V_{n+1} given by (2.1) the following relations are easily verified:

$$T_{n+m} = 2T_nT_m - T_{|m-n|} = 2U_{m-1}V_{n+1} + T_{|m-n|}. \tag{3.1}$$

Lemma 3.1. *For integers m and k and a positive integer n such that $0 \leq k \leq n$ we have*

$$T_{2mn+k} \equiv T_k \pmod{V_{n+1}}. \tag{3.2}$$

Proof. Since from (3.1) we have

$$T_{2n+k} = 2U_{n+k-1}V_{n+1} + T_k \equiv T_k \pmod{V_{n+1}},$$

we can verify (3.2) for any m recursively. □

Lemma 3.2. *Let m be an integer and n be a positive integer. Then for k and β such that $0 \leq k \leq n/2$ and $|\beta| \leq 1$, respectively, we have*

$$T_{mn+k} \equiv U_m(\beta)T_k - U_{m-1}(\beta)T_{n-k} \pmod{T_n - \beta}. \tag{3.3}$$

Proof. The proof is by induction on m. The case $m = 0$ is trivial since $U_{-1}(\beta) = 0$ and $U_0(\beta) = 1$. The case $m = 1$ holds since from (3.1)

$$T_{n+k} = 2T_nT_k - T_{n-k} \equiv 2\beta T_k - T_{n-k}.$$

Let $l \geq 1$ and assume that (3.3) holds for $0 \leq m \leq l$. Then from (3.1) and the recurrence relation [12, p. 31] $U_{l+1}(\beta) = 2\beta\, U_l(\beta) - U_{l-1}(\beta)$ $(l \geq 0)$ we have

$$\begin{aligned} T_{(l+1)n+k} &= 2T_nT_{ln+k} - T_{(l-1)n+k} \\ &\equiv 2\beta\{U_l(\beta)T_k - U_{l-1}(\beta)T_{n-k}\} - \{U_{l-1}(\beta)T_k - U_{l-2}(\beta)T_{n-k}\} \\ &= \{2\beta U_l(\beta) - U_{l-1}(\beta)\}T_k - \{2\beta U_{l-1}(\beta) - U_{l-2}(\beta)\}T_{n-k} \\ &= U_{l+1}(\beta)T_k - U_l(\beta)T_{n-k}. \end{aligned}$$

The case $m < 0$ is verified similarly but we omit the proof. □

4. Fast algorithm for quasi-Chebyshev interpolation

4.1. Constructing the quasi-Chebyshev interpolation.

We present an algorithm to construct the sequence of interpolation polynomials

$$\Phi_n(B_0)f, \Phi_n(B_1)f, \ldots, \Phi_n(B_\iota)f,$$

satisfying (2.8). Particularly, assuming that $\Phi_n(B_0)f = \sum_{k=0}^{\lambda n} a_k^{\lambda n}\, T_k$ and

$$\Psi_n^{[l]}f = \sum_{k=0}^{n-1} b_k^{[l]}\, T_k \equiv f \pmod{T_n - \alpha_l} \quad (1 \leq l \leq \mu), \tag{4.1}$$

are given, we compute the coefficients $a_k^{(\lambda+\mu)n}$ of $\Phi_n(B)f$ satisfying (2.6). The computation of $b_k^{[l]}$ in (4.1) is shown in the following subsection. For simplicity we rewrite $a_k^{\lambda n}$ as $b_k^{[0]}$ to express $\Phi_n(B_0)f$ as follows:

$$\Phi_n(B_0)f = \sum_{k=0}^{\lambda n} b_k^{[0]}\, T_k \equiv f \pmod{W_n(B_0) = V_{\lambda n+1}}. \tag{4.2}$$

We use the CRT (4.3) below to determine $a_k^{(\lambda+\mu)n}$ in (2.6) so that $\Phi_n(B)f \equiv \Phi_n(B_0)f \pmod{W_n(B_0)}$ and $\Phi_n(B)f \equiv \Psi_n^{[l]}f \pmod{T_n - \alpha_l}$, namely

$$\Phi_n(B_0)\{\Phi_n(B)f\} = \Phi_n(B_0)f, \quad \Psi_n^{[l]}\{\Phi_n(B)f\} = \Psi_n^{[l]}f\ (1 \le l \le \mu). \tag{4.3}$$

Now we give the computational procedure of (4.3) to obtain $a_k^{(\lambda+\mu)n}$ in some detail. We begin by expressing (4.2) and (4.1) as follows:

$$\begin{aligned}\Phi_n(B_0)f = &\sum_{m=0}^{\lambda} b_{mn}^{[0]}\, T_{mn} + \sum_{m=0}^{\lambda-1} b_{mn+n/2}^{[0]}\, T_{mn+n/2} \\ &+ \sum_{k=1}^{n/2-1}\sum_{m=-\lambda}^{\lambda-1} b_{|mn+k|}^{[0]}\, T_{mn+k},\end{aligned} \tag{4.4}$$

$$\Psi_n^{[l]}f = b_0^{[l]}\, T_0 + b_{n/2}^{[l]}\, T_{n/2} + \sum_{k=1}^{n/2-1} \{b_k^{[l]}\, T_k + b_{n-k}^{[l]}\, T_{n-k}\}. \tag{4.5}$$

If n is odd, then the second terms on the right-hand sides of (4.4) and (4.5) must be omitted and $n/2-1$ means $\lfloor n/2 \rfloor - 1$. Recall that $T_{-k} = T_k$ $(k > 0)$. By defining

$$\begin{cases} \boldsymbol{b}_0^{[0]} = \boldsymbol{b}_0^{[0]}(f) = (b_0^{[0]}, b_n^{[0]}, \ldots, b_{\lambda n}^{[0]})^T, \\ \boldsymbol{b}_k^{[0]} = \boldsymbol{b}_k^{[0]}(f) = (b_{|-\lambda n+k|}^{[0]}, b_{|-\lambda n+n+k|}^{[0]}, \ldots, b_{|\lambda n-n+k|}^{[0]})^T \quad (0 < k < n/2), \\ \boldsymbol{b}_{n/2}^{[0]} = \boldsymbol{b}_{n/2}^{[0]}(f) = (b_{n/2}^{[0]}, b_{n+n/2}^{[0]}, \ldots, b_{\lambda n-n/2}^{[0]})^T, \end{cases} \tag{4.6}$$

$$\boldsymbol{b}_k^{[l]} = \boldsymbol{b}_k^{[l]}(f) = (b_k^{[l]}, b_{n-k}^{[l]})^T \quad (0 < k < n/2, \quad 1 \le l \le \mu), \tag{4.7}$$

we define $\boldsymbol{b}_k$ $(0 \le k \le n/2)$ by

$$\begin{cases} \boldsymbol{b}_0 = \boldsymbol{b}_0(f) = ({\boldsymbol{b}_0^{[0]}}^T, b_0^{[1]}, \ldots, b_0^{[\mu]})^T \in \mathbb{R}^{\lambda+\mu+1}, \\ \boldsymbol{b}_k = \boldsymbol{b}_k(f) = ({\boldsymbol{b}_k^{[0]}}^T, {\boldsymbol{b}_k^{[1]}}^T, \ldots, {\boldsymbol{b}_k^{[\mu]}}^T)^T \in \mathbb{R}^{2\lambda+2\mu}\ (0 < k < n/2), \\ \boldsymbol{b}_{n/2} = \boldsymbol{b}_{n/2}(f) = ({\boldsymbol{b}_{n/2}^{[0]}}^T, b_{n/2}^{[1]}, \ldots, b_{n/2}^{[\mu]})^T \in \mathbb{R}^{\lambda+\mu}. \end{cases} \tag{4.8}$$

Further, we define

$$\boldsymbol{b} = \boldsymbol{b}(f) = (\boldsymbol{b}_0^T, \boldsymbol{b}_1^T, \ldots, \boldsymbol{b}_{n/2}^T)^T \in \mathbb{R}^{\lambda n+\mu n+1}. \tag{4.9}$$

Similarly, for simplicity we omit the superscript of $a_k^{(\lambda+\mu)n}$ in (2.6) as a_k to write

$$\begin{aligned}\Phi_n(B)f = &\sum_{m=0}^{\lambda+\mu} a_{mn} T_{mn} + \sum_{m=0}^{\lambda+\mu-1} a_{mn+n/2} T_{mn+n/2} \\ &+ \sum_{k=1}^{n/2-1}\ \sum_{m=-\lambda-\mu}^{\lambda+\mu-1} a_{|mn+k|} T_{mn+k}.\end{aligned} \tag{4.10}$$

By defining the coefficient vectors $\boldsymbol{a}_k$ $(0 \le k \le n/2)$ by

$$\begin{aligned}
\boldsymbol{a}_0 &= \boldsymbol{a}_0(f) = (a_0, a_n, \ldots, a_{(\lambda+\mu)n})^T,\\
\boldsymbol{a}_k &= \boldsymbol{a}_k(f) = (a_{|-(\lambda+\mu)n+k|}, a_{|-(\lambda+\mu-1)n+k|}, \ldots, a_{|(\lambda+\mu-1)n+k|})^T\\
&\qquad\qquad (0 < k < n/2),\\
\boldsymbol{a}_{n/2} &= \boldsymbol{a}_{n/2}(f) = (a_{n/2}, a_{n+n/2}, \ldots, a_{\lambda n+\mu n-n/2})^T,
\end{aligned}$$

we define $\boldsymbol{a}$ by

$$\boldsymbol{a} = \boldsymbol{a}(f) = (\boldsymbol{a}_0^T, \boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_{n/2}^T)^T \in \mathbb{R}^{\lambda n+\mu n+1}. \tag{4.11}$$

Then, we can get $\boldsymbol{a}(f)$ by solving the linear system of equations derived from (4.3), which gives $\boldsymbol{b}(\Phi_n(B)f) = \boldsymbol{b}(f)$, or

$$\boldsymbol{b}_k(\Phi_n(B)f) = \boldsymbol{b}_k(f) \quad (0 \le k \le n/2). \tag{4.12}$$

Here we actually derive the system of equations for $\boldsymbol{a}(f)$ from (4.12). To this end we compute $\boldsymbol{b}(T_{mn+k})$ $(-\infty < m < \infty,\ 0 \le k \le n/2)$. Since from (3.2) we see that $T_{2\lambda mn+k} \equiv T_k \pmod{V_{\lambda n+1}}$ and recalling that $T_{-k} = T_k$ we can obtain the following results:

$$\begin{aligned}
&\Phi_n(B_0)T_{mn} = T_{\widehat{m}n}, \quad \widehat{m} := \lambda - |\operatorname{mod}(m, 2\lambda) - \lambda| \in [0, \lambda],\\
&\Phi_n(B_0)T_{mn+k} = T_{\overline{m}n+k} \quad (0 < k < n/2),\\
&\qquad \overline{m} := \operatorname{mod}(m+\lambda, 2\lambda) - \lambda \in [-\lambda, \lambda),\\
&\Phi_n(B_0)T_{mn+n/2} = T_{\widetilde{m}n+n/2},\\
&\qquad \widetilde{m} := \lambda - 1/2 - |\operatorname{mod}(m, 2\lambda) - \lambda + 1/2| \in [0, \lambda).
\end{aligned} \tag{4.13}$$

From (4.6) and (4.13) we have

$$\begin{aligned}
&\boldsymbol{b}_0^{[0]}(T_{mn+k}) = \delta_{0,k}\boldsymbol{p}_m^{[0]}, \quad \boldsymbol{b}_j^{[0]}(T_{mn+k}) = \delta_{j,k}\boldsymbol{q}_m^{[0]} \quad (0 < j < n/2),\\
&\boldsymbol{b}_{n/2}^{[0]}(T_{mn+k}) = \delta_{n/2,k}\boldsymbol{r}_m^{[0]},
\end{aligned} \tag{4.14}$$

for $0 \le k \le n/2$, where $\delta_{j,k} = 1$ if $j = k$; otherwise $\delta_{j,k} = 0$, and

$$\begin{aligned}
\boldsymbol{p}_m^{[0]} &:= (\delta_{0,\widehat{m}}, \delta_{1,\widehat{m}}, \ldots, \delta_{\lambda,\widehat{m}}) = (\overset{0}{0}, \ldots, 0, \overset{\widehat{m}}{1}, 0, \ldots, 0)^T \in \mathbb{R}^{\lambda+1},\\
\boldsymbol{q}_m^{[0]} &:= (\delta_{-\lambda,\overline{m}}, \delta_{-\lambda+1,\overline{m}}, \ldots, \delta_{\lambda-1,\overline{m}}) = (\overset{-\lambda}{0}, \ldots, 0, \overset{\overline{m}}{1}, 0, \ldots, 0)^T \in \mathbb{R}^{2\lambda},\\
\boldsymbol{r}_m^{[0]} &:= (\delta_{0,\widetilde{m}}, \delta_{1,\widetilde{m}}, \ldots, \delta_{\lambda-1,\widetilde{m}}) = (\overset{0}{0}, \ldots, 0, \overset{\widetilde{m}}{1}, 0, \ldots, 0)^T \in \mathbb{R}^{\lambda}.
\end{aligned} \tag{4.15}$$

On the other hand, from (3.3) and (4.1) we have for $1 \le l \le \mu$,

$$\begin{aligned}
&\Psi_n^{[l]}T_{mn} = p_m^{[l]} := U_m(\alpha_l) - \alpha_l U_{m-1}(\alpha_l),\\
&\Psi_n^{[l]}T_{mn+k} = U_m(\alpha_l)T_k - U_{m-1}(\alpha_l)T_{n-k} \quad (0 < k < n/2),\\
&\Psi_n^{[l]}T_{mn+n/2} = r_m^{[l]}T_{n/2}, \quad r_m^{[l]} := U_m(\alpha_l) - U_{m-1}(\alpha_l).
\end{aligned} \tag{4.16}$$

From (4.7) and (4.16) we have

$$\begin{aligned}
&b_0^{[l]}(T_{mn+k}) = \delta_{0,k}\, p_m^{[l]}, \quad \boldsymbol{b}_j^{[l]}(T_{mn+k}) = \delta_{j,k}\, \boldsymbol{q}_m^{[l]} \quad (0 < j < n/2),\\
&b_{n/2}^{[l]}(T_{mn+k}) = \delta_{n/2,k}\, r_m^{[l]},
\end{aligned} \tag{4.17}$$

for $0 \le k \le n/2$, where

$$\boldsymbol{q}_m^{[l]} := (U_m(\alpha_l), -U_{m-1}(\alpha_l))^T \in \mathbb{R}^2.$$

$$P = \left[\begin{array}{cccc|} I_{\lambda-\mu} & & & \\ & I_\mu & & J_\mu \\ & & 1 & \\ \times & \times & \times & \times \end{array}\right] \overset{\lambda+1}{\hookleftarrow}, \quad Q = \begin{bmatrix} & I_\mu & & & I_\mu \\ & & I_{2\lambda-2\mu} & & \\ I_\mu & & & I_\mu & \\ \times & \times & \times & \times & \times \end{bmatrix}$$

$$R = \begin{bmatrix} I_{\lambda-\mu} & & \\ & I_\mu & J_\mu \\ \times & \times & \times \end{bmatrix}, \ I_\mu := \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} \in \mathbb{R}^{\mu\times\mu}, \ J_\mu := \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix} \in \mathbb{R}^{\mu\times\mu}$$

FIGURE 1. The matrices P, Q and R

It follows from (4.8), (4.14) and (4.17) that for $0 \le k \le n/2$ and for $0 < j < n/2$,

$$\begin{aligned} &\boldsymbol{b}_0(T_{mn+k}) = \delta_{0,k}\, \boldsymbol{p}_m, \quad \boldsymbol{p}_m := (\boldsymbol{p}_m^{[0]T}, p_m^{[1]}, \dots, p_m^{[\mu]})^T \in \mathbb{R}^{\lambda+\mu+1}, \\ &\boldsymbol{b}_j(T_{mn+k}) = \delta_{j,k}\, \boldsymbol{q}_m, \quad \boldsymbol{q}_m := (\boldsymbol{q}_m^{[0]T}, \boldsymbol{q}_m^{[1]T}, \dots, \boldsymbol{q}_m^{[\mu]T})^T \in \mathbb{R}^{2\lambda+2\mu}, \\ &\boldsymbol{b}_{n/2}(T_{mn+k}) = \delta_{n/2,k}\, \boldsymbol{r}_m, \quad \boldsymbol{r}_m := (\boldsymbol{r}_m^{[0]T}, r_m^{[1]}, \dots, r_m^{[\mu]})^T \in \mathbb{R}^{\lambda+\mu}. \end{aligned} \tag{4.18}$$

Finally, from (4.9) and (4.18) we have

$$\begin{aligned} \boldsymbol{b}(T_{mn+k}) &= (\boldsymbol{b}_0(T_{mn+k}), \boldsymbol{b}_1(T_{mn+k}), \dots, \boldsymbol{b}_{n/2-1}(T_{mn+k}), \boldsymbol{b}_{n/2}(T_{mn+k})) \\ &= (\delta_{0,k}\boldsymbol{p}_m^T,\ \delta_{1,k}\boldsymbol{q}_m^T,\ \dots\ ,\ \delta_{n/2-1,k}\boldsymbol{q}_m^T,\ \delta_{n/2,k}\boldsymbol{r}_m^T)^T. \end{aligned} \tag{4.19}$$

Define the matrices P, Q and R by

$$\begin{aligned} P &= (\boldsymbol{p}_0, \boldsymbol{p}_1, \dots, \boldsymbol{p}_{\lambda+\mu}) \in \mathbb{R}^{(\lambda+\mu+1)\times(\lambda+\mu+1)}, \\ Q &= (\boldsymbol{q}_{-\lambda-\mu}, \boldsymbol{q}_{-\lambda-\mu+1}, \dots, \boldsymbol{q}_{\lambda+\mu-1}) \in \mathbb{R}^{(2\lambda+2\mu)\times(2\lambda+2\mu)}, \\ R &= (\boldsymbol{r}_0, \boldsymbol{r}_1, \dots, \boldsymbol{r}_{\lambda+\mu-1}) \in \mathbb{R}^{(\lambda+\mu)\times(\lambda+\mu)}, \end{aligned}$$

respectively, see Figure 1. Then since from (4.10) and (4.18) we have for $0 < j < n/2$,

$$\begin{aligned} \boldsymbol{b}_0(\Phi_n(B)f) &= \sum_{m=0}^{\lambda+\mu} a_{mn}\boldsymbol{b}_0(T_{mn}) = \sum_{m=0}^{\lambda+\mu} a_{mn}\boldsymbol{p}_m = P\,\boldsymbol{a}_0, \\ \boldsymbol{b}_j(\Phi_n(B)f) &= \sum_{m=-\lambda-\mu}^{\lambda+\mu-1} a_{|mn+j|}\boldsymbol{b}_j(T_{mn+j}) = \sum_{m=-\lambda-\mu}^{\lambda+\mu-1} a_{|mn+j|}\boldsymbol{q}_m = Q\,\boldsymbol{a}_j, \\ \boldsymbol{b}_{n/2}(\Phi_n(B)f) &= \sum_{m=0}^{\lambda+\mu-1} a_{mn+n/2}\boldsymbol{b}_{n/2}(T_{mn+n/2}) = \sum_{m=0}^{\lambda+\mu-1} a_{mn+n/2}\boldsymbol{r}_m = R\,\boldsymbol{a}_{n/2}, \end{aligned}$$

it follows that

$$P\,\boldsymbol{a}_0(f) = \boldsymbol{b}_0(f),\ Q\,\boldsymbol{a}_j(f) = \boldsymbol{b}_j(f)\ (0 < j < n/2),\ R\,\boldsymbol{a}_{n/2}(f) = \boldsymbol{b}_{n/2}(f). \tag{4.20}$$

Now we examine the number of multiplications required to obtain the values of a_k ($0 \le k \le (\lambda+\mu)n$) in the equations (4.20) above. Assume that P^{-1}, Q^{-1} and R^{-1} are given. Then $(\lambda+\mu+1)^2$ multiplications are required to obtain $\boldsymbol{a}_0 = P^{-1}\boldsymbol{b}_0$. Similarly, $(\lambda+\mu)^2$ and $4(\lambda+\mu)^2$ multiplications are required for $\boldsymbol{a}_{n/2}$ and $\boldsymbol{a}_j$, respectively. The total number of multiplications $M_n^C(B)$ is

$$\begin{aligned} M_n^C(B) &= (\lambda+\mu+1)^2 + (\lambda+\mu)^2 + \lfloor n/2 \rfloor 4(\lambda+\mu)^2 \\ &= 2n(\lambda+\mu)^2 + O(1) \le K^C(B)(\lambda+\mu)n, \end{aligned} \tag{4.21}$$

where $K^C(B)$ is a constant independent of n.

4.2. **Fast algorithm for $\Psi_n^{[l]} f$.** We present a fast algorithm based on the real FFT [20] for evaluating the coefficients $b_k^{[l]}$ of $\Psi_n^{[l]} f$ (4.1).

Theorem 4.1. *Let* $\alpha_l = \cos 2\pi\tau$ *and* $\xi_j = 2\pi(j+\tau)/n$ $(0 \le j < n)$. Further, for $0 \le k < n$, let

$$c_k = \frac{1}{n}\sum_{j=0}^{n-1} f(\cos\xi_j) e^{-ik\xi_j} = \frac{e^{-i2\pi k\tau/n}}{n}\sum_{j=0}^{n-1} f(\cos\xi_j) e^{-i2\pi kj/n}. \tag{4.22}$$

Then we have the coefficients $b_k^{[l]}$ of $\Psi_n^{[l]} f$ (4.1) as follows:

$$b_0^{[l]} = c_0, \quad b_k^{[l]} = 2(\Re c_k - \alpha_l \Re c_{n-k})/(1-\alpha_l^2) \quad (0 < k < n). \tag{4.23}$$

Proof. From the relations

$$\begin{aligned} b_k^{[l]} &= 2\Im(\overline{c}_{n-k})/\sin 2\pi\tau \ (0<k<n), \qquad & b_0^{[l]} &= \Im(\overline{c}_n)/\sin 2\pi\tau, \\ c_{n-k} &= \overline{c}_k \exp(-2\pi i\tau), \qquad & c_n &= c_0 \exp(-2\pi i\tau), \end{aligned}$$

given in [10, (3.12),(3.13)] it is easy to verify (4.23). □

Remark 4.2. If $n = 2^j$ $(j = 1, 2, \dots)$, then the summations on the rightmost-hand side of (4.22) can be efficiently computed by the real FFT [20] with $n \log_2 n$ real multiplications.

It follows that the number of multiplications $M_n^{(1)}$ required to obtain c_k from (4.22) is given by

$$M_n^{(1)} \le n\log_2 n + K^{(1)} n, \tag{4.24}$$

where $K^{(1)}$ is a constant independent of n.

4.3. **Computational costs for quasi-Chebyshev interpolation.** We examine the computational costs, particularly the number of multiplications, required to construct the quasi-Chebyshev interpolation. Here we assume that $n = 2^j$ $(j = 0, 1, \dots)$.

Lemma 4.3. *For* $n = 2^j$ $(j = 0, 1, \dots)$ let $M_n(B_0)$ denote the number of multiplications required to obtain the coefficients $b_k^{[0]}$ of $\Phi_n(B_0) f$ (4.2). Then we have

$$M_n(B_0) \le \lambda n \log_2 n + K^{(0)} \lambda n, \tag{4.25}$$

where $K^{(0)} = \max\{M_1(B_0)/\lambda, K^{(1)} + 2K^C(A(\lambda))\}$.

Proof. The proof is by induction. The case $n = 1$ is trivial. Assume that (4.25) holds for some $n = 2^j \geq 1$. We can verify the case $2n = 2^{j+1}$ as follows. Since $\Phi_{2n}(B_0)f = \Phi_n(A(\lambda))f$ we have from (4.21) and (4.24)

$$\begin{aligned} M_{2n}(B_0) &\leq M_n(B_0) + \lambda M_n^{(1)} + M_n^C(A(\lambda)) \\ &\leq \lambda n \log_2 n + K^{(0)}\lambda n + \lambda(n \log_2 n + K^{(1)} n) + K^C(A(\lambda))2\lambda n \\ &= 2\lambda n \log_2 n + \{K^{(0)}/2 + K^{(1)}/2 + K^C(A(\lambda))\}2\lambda n \\ &\leq 2\lambda n \log_2 n + (K^{(0)}/2 + K^{(0)}/2)2\lambda n \\ &= 2\lambda n \log_2 n + K^{(0)} 2\lambda n. \end{aligned}$$ □

Similarly, we have the following theorem.

Theorem 4.4. *For $n = 2^j$ $(j = 0, 1, \dots)$ let $M_n(B)$ denote the number of multiplications required to obtain the coefficients $a_k^{(\lambda+\mu)n}$ of $\Phi_n(B)f$ (2.6). Let $N = (\lambda + \mu)n$. Then $M_n(B) = N \log_2 N + O(N)$.*

5. Interpolation error

In this section we study the error analysis of the quasi-Chebyshev interpolation (2.6). Here we assume that n is a positive integer. Assume that the Chebyshev series expansion of a given function $f(x)$ defined on $[-1, 1]$ given by

$$f(x) = \sum_{k=0}^{\infty} \widehat{a}_k T_k(x), \tag{5.1}$$

is of absolute convergence for any $x \in [-1, 1]$, namely $\sum_{k=0}^{\infty} |\widehat{a}_k| < \infty$, where $\widehat{a}_0 = (1/\pi)\int_0^{\pi} f(\cos t)\, dt$, $\widehat{a}_k = (2/\pi)\int_0^{\pi} f(\cos t) \cos kt\, dt$ $(k > 0)$.

We begin by writing the quasi-Chebyshev interpolation $\Phi_n(B)T_k$ $(k \geq 0)$ of the basis T_k with the coefficients vector $\boldsymbol{t}_k^{(n)}$ as follows:

$$\Phi_n(B)T_k = \sum_{j=0}^{(\lambda+\mu)n} t_{jk}^{(n)} T_j, \quad \boldsymbol{t}_k^{(n)} := (t_{0k}^{(n)}, t_{1k}^{(n)}, \dots, t_{(\lambda+\mu)n,k}^{(n)})^T. \tag{5.2}$$

We define the error coefficients $\omega^{(n)}(B)$ for $\Phi_n(B)$ and $\omega(B)$ for B by

$$\omega^{(n)}(B) = \max_{k > (\lambda+\mu)n} \|\boldsymbol{t}_k^{(n)}\|_1, \quad \omega(B) = \max_{n \geq 1} \omega^{(n)}(B), \tag{5.3}$$

respectively, where $\|\cdot\|_1$ denotes the 1-norm of a vector.

Theorem 5.1. *The error of the quasi-Chebyshev interpolation $\Phi_n(B)f$ (2.6) and the sum of errors of the coefficients $a_k (= a_k^{(\lambda+\mu)n})$ are bounded, respectively, as follows:*

$$\|\Phi_n(B)f - f\|_\infty \leq \{\omega(B) + 1\} \sum_{k > (\lambda+\mu)n} |\widehat{a}_k|, \tag{5.4}$$

$$\sum_{k=0}^{(\lambda+\mu)n} |a_k - \widehat{a}_k| \leq \omega(B) \sum_{k > (\lambda+\mu)n} |\widehat{a}_k|, \tag{5.5}$$

where $\widehat{a}_k$ are the coefficients of the series (5.1).

Proof. We begin by proving (5.5). Since $\Phi_n(B)T_k = T_k$ $(0 \le k \le (\lambda+\mu)n)$ from (5.1) and (5.2) we have

$$\Phi_n(B)f = \sum_{k=0}^{\infty} \widehat{a}_k \Phi_n(B)T_k = \sum_{j=0}^{(\lambda+\mu)n} \widehat{a}_j T_j + \sum_{k>(\lambda+\mu)n} \widehat{a}_k \sum_{j=0}^{(\lambda+\mu)n} t_{jk}^{(n)} T_j$$
$$= \sum_{j=0}^{(\lambda+\mu)n} (\widehat{a}_j + \sum_{k>(\lambda+\mu)n} \widehat{a}_k t_{jk}^{(n)}) T_j.$$

From the above relation and (2.6) it follows that $a_j - \widehat{a}_j = \sum_{k>(\lambda+\mu)n} \widehat{a}_k t_{jk}^{(n)}$ $(0 \le j \le (\lambda+\mu)n)$, which gives

$$\sum_{j=0}^{(\lambda+\mu)n} |a_j - \widehat{a}_j| = \sum_{j=0}^{(\lambda+\mu)n} \Big| \sum_{k>(\lambda+\mu)n} \widehat{a}_k t_{jk}^{(n)} \Big| \le \sum_{k>(\lambda+\mu)n} |\widehat{a}_k| \sum_{j=0}^{(\lambda+\mu)n} |t_{jk}^{(n)}|$$
$$= \sum_{k>(\lambda+\mu)n} \|\boldsymbol{t}_k^{(n)}\|_1 |\widehat{a}_k| \le \omega^{(n)}(B) \sum_{k>(\lambda+\mu)n} |\widehat{a}_k| \le \omega(B) \sum_{k>(\lambda+\mu)n} |\widehat{a}_k|.$$

Similarly, we can verify (5.4) as follows:

$$|f - \Phi_n(B)f| \le \sum_{k>(\lambda+\mu)n} |\widehat{a}_k| \cdot |T_k - \Phi_n(B)T_k|$$
$$\le \sum_{k>(\lambda+\mu)n} |\widehat{a}_k| \left(|T_k| + |\Phi_n(B)T_k|\right) \le \{1 + \omega^{(n)}(B)\} \sum_{k>(\lambda+\mu)n} |\widehat{a}_k|,$$

since $|T_k(x)| \le 1$ for $|x| \le 1$ and we have

$$|\Phi_n(B)T_k| \le \|\boldsymbol{t}_k^{(n)}\|_1 \le \omega^{(n)}(B) \quad (k > (\lambda+\mu)n). \qquad \square$$

The following theorem is helpful to evaluate $\omega(B)$.

Theorem 5.2. *For $\omega^{(n)}(B)$ and $\omega(B)$ defined by* (5.3) *we have*

$$\omega^{(n)}(B) = \max_{0 \le k \le 2\lambda n} \|\boldsymbol{t}_k^{(n)}\|_1, \quad \omega(B) = \omega^{(4)}(B) = \max_{0 \le k \le 8\lambda} \|\boldsymbol{t}_k^{(4)}\|_1. \tag{5.6}$$

Proof. To verify the first equation of (5.6) it suffices to show that

$$\boldsymbol{t}_{4m\lambda n+k}^{(n)} = \boldsymbol{t}_k^{(n)}, \quad \boldsymbol{t}_{(4m-2)\lambda n+k}^{(n)} = \boldsymbol{t}_{2\lambda n-k}^{(n)}, \tag{5.7}$$

for $0 \le k \le 2\lambda n$ and $m \ge 1$. From (3.2) we see that $T_{4m\lambda n+k} \equiv T_k \pmod{V_{2\lambda n+1}}$. On the other hand, since from (2.3) and (2.4) we see that $W_n(B)$ divides $V_{2\lambda n+1}$ we have $T_{4m\lambda n+k} \equiv T_k \pmod{W_n(B)}$, namely $\Phi_n(B)T_{4m\lambda n+k} = \Phi_n(B)T_k$, which gives the first relation of (5.7). Similarly, we can derive the second relation of (5.7) since $T_{(4m-2)\lambda n+k} \equiv T_{k-2\lambda n} = T_{2\lambda n-k} \pmod{V_{2\lambda n+1}}$.

We proceed to prove the second equation of (5.6). Let $\boldsymbol{u}_m = P^{-1}\boldsymbol{p}_m$, $\boldsymbol{v}_m = Q^{-1}\boldsymbol{q}_m$ and $\boldsymbol{w}_m = R^{-1}\boldsymbol{r}_m$. Then since from (4.19) and (4.20) we have that $\boldsymbol{a}(T_{mn+k}) = (\delta_{0,k}\boldsymbol{u}_m^T, \delta_{1,k}\boldsymbol{v}_m^T, \ldots, \delta_{n/2-1,k}\boldsymbol{v}_m^T, \delta_{n/2,k}\boldsymbol{w}_m^T)^T$, where $\boldsymbol{a}$ is defined by (4.11), it follows that

$$\|\boldsymbol{t}_{mn+k}^{(n)}\|_1 = \|\boldsymbol{a}(T_{mn+k})\|_1 = \begin{cases} \|\boldsymbol{u}_m\|_1 & (k=0), \\ \|\boldsymbol{v}_m\|_1 & (0<k<n/2), \\ \|\boldsymbol{w}_m\|_1 & (k=n/2). \end{cases} \tag{5.8}$$

From the first equation of (5.6) we have

$$\omega^{(n)}(B) = \max_{\substack{0\le k\le n\\ 0\le m<2\lambda}} \|\boldsymbol{t}^{(n)}_{mn+k}\|_1 = \max\left\{ \max_{\substack{0\le k\le n/2\\ 0\le m<2\lambda}} \|\boldsymbol{t}^{(n)}_{mn+k}\|_1, \max_{\substack{0\le k\le n/2\\ 0\le m<2\lambda}} \|\boldsymbol{t}^{(n)}_{mn+n-k}\|_1 \right\}$$

$$= \max\left\{ \max_{\substack{0\le k\le n/2\\ 0\le m<2\lambda}} \|\boldsymbol{t}^{(n)}_{mn+k}\|_1, \max_{\substack{0\le k\le n/2\\ 0\le m<2\lambda}} \|\boldsymbol{t}^{(n)}_{(-m-1)n+k}\|_1 \right\} = \max_{\substack{0\le k\le n/2\\ -2\lambda\le m<2\lambda}} \|\boldsymbol{t}^{(n)}_{mn+k}\|_1,$$

which gives with (5.8)

$$\omega^{(n)}(B) = \max_{-2\lambda\le m<2\lambda} \max\{\|\boldsymbol{u}_m\|_1, \|\boldsymbol{v}_m\|_1, \|\boldsymbol{w}_m\|_1\}, \tag{5.9}$$

if $n \ge 4$ and is even. Otherwise, if $n \ge 3$ and odd, then $\|\boldsymbol{w}_m\|_1$ must be neglected in the right-hand side of (5.9). If $n \le 2$, then $\|\boldsymbol{v}_m\|_1$ must be neglected. The above discussion reveals that $\omega^{(n)}(B) \le \omega^{(4)}(B)$ for any positive integer n. □

Remark 5.3. For $\omega(B)$ (5.6) we have $\omega(B) \ge 1$ because $\|\boldsymbol{t}^{(4)}_0\|_1 = 1$, particularly, $\omega(B_0) = 1$. If $\lambda = 1$, then $A(1) = \{0\}$ in (2.3), namely we have $W_n(B_0) = V_{n+1}$ and $W_n(B_1) = V_{2n+1} = W_{2n}(B_0)$. This means that the sequence of the Chebyshev interpolation $\{\Phi_n(B_0)f\}$ ($n = 2^j$, $j = 0, 1, 2, \dots$) at the Chebyshev nodes is the best in that the error coefficients are always smallest.

In the next section we search for the sequence of interpolation polynomials with error coefficients as small as possible but the sequence of degrees of polynomials increases slower than that of the above sequence.

6. Search for the optimum sequence

The numerical and non-numerical computations in this section were carried out by using the Mathematica 5.2 on the Apple iBook with PowerPC G4.1, 1.2GHz.

Assume that n is a power of two. We search for a sequence of polynomials $\Phi_n(B_j)f$ of $\{B_j\}$ ($0 \le j \le \iota$) satisfying (2.7) such that the degree of polynomial $n(\lambda + \mu_j)$, where $\mu_j = |B_j|$, increases monotonously, namely

$$n\lambda < n(\lambda+\mu_1) < n(\lambda+\mu_2) < \cdots < n(\lambda+\mu_{\iota-1}) < 2n\lambda < \cdots,$$

where

$$1 \le \mu_j - \mu_{j-1} \le \mu_{j+1} - \mu_j \ (0 < j < \iota), \quad \mu_\iota - \mu_{\iota-1} \le 2\mu_1.$$

Recall that $\mu_0 = 0$ and $\mu_\iota = \lambda$. Our particular interest is in the sequence $\{B_j\}$ of the smallest ω, where ω is defined by

$$\omega = \max_{0\le j<\iota} \omega(B_j),$$

for $\lambda \ge 2$ and $1 \le \iota \le \lambda$. From the practical point of view we search for the cases $\iota = 2, 3$ and 4 only in the ranges $2 \le \lambda \le 17$, $3 \le \lambda \le 19$ and $4 \le \lambda \le 20$, respectively. The results obtained are shown in Table 1. Recall that $W_n(B_0) = V_{\lambda n+1}$.

Remark 6.1. The sequence of HTS [11] shown in the Example 2.3 is the case where $\iota = 3$ and $\lambda = 4$ and $\omega = 5.82\cdots$, slightly larger than the optimum value $5.47\cdots$ shown in Table 1.

TABLE 1. Optimum node set

ι	2	3	4
ω	**3**	**5.47**$\cdots$	**6.75**$\cdots$
λ	3	5	9
$(\mu_1, \mu_2, \mu_3, \mu_4)$	$(1, \lambda)$	$(1, 3, \lambda)$	$(2, 4, 6, \lambda)$
degree of p_N	$\{3,4\}\times 2^i$	$\{5,6,8\}\times 2^i$	$\{9,11,13,15\}\times 2^i$
B_1	$\{\cos \pi/2\}$		$\{\cos(\pi/2 \pm \pi/9)\}$
$B_2 \backslash B_1$	$\{\cos(1 \pm (1 - 1/\lambda))\pi/2\}$		
$B_3 \backslash B_2$		$\{\cos(\pi/2 \pm \pi/5)\}$	$\{\cos(\pi/2 \pm 2\pi/9)\}$
$B_4 \backslash B_3$			$\{\cos \pi/2, \cos(\pi/2 \pm 3\pi/9)\}$
$W_n(B_1)/(2W_n(B_0))$	T_n		$T_{2n} + \cos 2\pi/9$
$W_n(B_2)/(2W_n(B_1))$	$T_{2n} - \cos \pi/\lambda$		
$W_n(B_3)/(2W_n(B_2))$		$T_{2n} + \cos 2\pi/5$	$T_{2n} + \cos 4\pi/9$
$W_n(B_4)/(2W_n(B_3))$			T_{3n}

7. OUTLINE OF THE ALGORITHM FOR AUTOMATIC INTERPOLATION

We outline the recursive construction of a sequence of interpolation polynomials with an estimated error being within the tolerance ε, namely, the interpolation process based on the sequence (2.8) where $B_i = \{\alpha_j : 1 \le j \le \mu_i\}$ $(1 \le i \le \iota)$ $(1 \le \iota \le \lambda)$ and $\mu_i = |B_i|$ and $\mu_0 = 0$. The details of the implementation will appear elsewhere. We assume that the estimate $E(\Phi_n(B)f)$ for the error $\|\Phi_n(B)f - f\| \approx E(\Phi_n(B)f)$ is given. Schemes for estimating the error are presented in Sloan and Smith [17] and HTS [11], where the last several Chebyshev coefficients $= |a^{(\lambda+\mu)n}_{(\lambda+\mu)n-i}|$ $(i \ge 0)$ in (2.6) are used.

Outline of the algorithm
Output is an approximate interpolating polynomial with an estimated error.

1. Initialization.
   $n = 1$; $i = 1$;
   Compute $\Phi_1(B_0)f \equiv f \pmod{V_{\lambda+1}}$.
2. Computation of $\Psi^{[l]}_n f$.
   While $i \le \iota$ {
   - compute $\Psi^{[l]}_n f \equiv f \pmod{T_n - \alpha_l}$ $\quad (\mu_{i-1} < l \le \mu_i)$.
   - if $E(\Phi_n(B_i)f) \le \varepsilon$, do Step 3.
   - $i = i + 1$;

   }
   Combine $\Phi_n(B_0)f$ with $\Psi^{[l]}_n f$ $(1 \le l \le \mu_\iota)$ to obtain $\Phi_{2n}(B_0)f$.
   $n = 2n$; $i = 1$; repeat Step 2.
3. Treatment on the convergence.
   Combine $\Phi_n(B_0)f$ with $\Psi^{[l]}_n f$ $(1 \le l \le \mu_i)$ to obtain $\Phi_n(B_i)f$.
   exit.

TABLE 2. Errors $E_N^{(C)}$ at the Chebyshev nodes and $E_N^{(Q)}$ at the quasi-Chebyshev nodes. The values in bold face of the ratio $(= E_N^{(Q)}/E_N^{(C)})$ mean maxima.

	Chebyshev	quasi-Chebyshev					
		$\iota = 2$		$\iota = 3$		$\iota = 4$	
N	$E_N^{(C)}$	$E_N^{(Q)}$	ratio	$E_N^{(Q)}$	ratio	$E_N^{(Q)}$	ratio
5	5.96			5.96	1		
6	5.40	5.40	1	5.80	1.07		
8	4.40	4.93	1.12	5.45	1.24		
9	3.96					3.96	1
10	3.56			3.56	1		
11	3.19					3.82	1.20
12	2.86	2.86	1	3.35	1.17		
13	2.56					6.17	2.41
15	2.04					3.53	1.73
16	1.81	2.34	1.29	3.62	2.00		
18	1.44					1.44	1
20	1.13			1.13	1		
22	8.92×10^{-1}					1.32	1.48
24	7.01×10^{-1}	7.01×10^{-1}	1	9.82×10^{-1}	1.40		
26	5.50×10^{-1}					2.01	**3.66**
30	3.73×10^{-1}					1.08	2.88
32	3.09×10^{-1}	4.30×10^{-1}	1.39	9.24×10^{-1}	2.99		
36	2.09×10^{-1}					2.09×10^{-1}	1
40	1.39×10^{-1}			1.39×10^{-1}	1		
44	9.19×10^{-2}					1.27×10^{-1}	1.38
48	6.01×10^{-2}	6.01×10^{-2}	1	6.72×10^{-2}	1.12		
52	3.91×10^{-2}					1.27×10^{-1}	3.24
60	1.66×10^{-2}					5.39×10^{-2}	3.24
64	1.10×10^{-2}	1.71×10^{-2}	1.55	3.35×10^{-2}	**3.04**		
72	4.80×10^{-3}					4.80×10^{-3}	1
80	2.06×10^{-3}			2.06×10^{-3}	1		
88	8.79×10^{-4}					1.54×10^{-3}	1.75
96	3.82×10^{-4}	3.82×10^{-4}	1	6.20×10^{-4}	1.62		
104	1.65×10^{-4}					4.88×10^{-4}	2.95
120	3.04×10^{-5}					8.31×10^{-5}	2.73
128	1.32×10^{-5}	2.31×10^{-5}	**1.76**	3.43×10^{-5}	2.60		
$\vdots$							
480	1.03×10^{-21}					3.08×10^{-21}	2.99
512	3.54×10^{-23}	6.22×10^{-23}	**1.76**	9.29×10^{-23}	2.63		

8. Numerical examples

The computations in this section are carried out in the multiple precision arithmetic with the Mathematica. We show the errors E_N of interpolation polynomials p_N at the optimum node set in the QCN for each $\iota = 2, 3, 4$ given in section 6 for the function f given by

$$f(x) = \frac{1 - ux}{1 - 2ux + u^2} = \sum_{k=0}^{\infty} u^n\, T_k(x), \quad u = 0.9.$$

Here we define the error E_N by

$$E_N = \max_{0 \le i \le \nu} |p_N(x_i) - f(x_i)|,$$

where $x_i = \cos \pi i/\nu$ for some large ν, say $\nu = 2^{14}$. Although we computed the errors E_N for many values of $N \le 2048$, indeed $E_{2048} = O(10^{-93})$, and we show the results of small values of N in Table 2 and Figures 2, 3 and 4. Table 2 lists E_N ($5 \le N \le 128$, $N = 480, 512$) at the Chebyshev nodes and those at the optimum node set in the QCN, where $N = 3 \times 2^i, 4 \times 2^i$, $N = 5 \times 2^i, 6 \times 2^i, 8 \times 2^i$, and $N = 9 \times 2^i, 11 \times 2^i, 13 \times 2^i, 15 \times 2^i$, $(i = 1, 2, \dots)$ in the cases $\iota = 2, 3, 4$, respectively. Figures 2, 3 and 4 illustrate the comparison of the behaviors of the errors E_N at the optimum node sets (solid lines) with those at the Chebyshev nodes (broken lines). Comparing Figures 2, 3, and 4 reveals that although the behavior of the errors with small N for $\iota = 4$ is a little worse than those for $\iota = 2, 3$, as expected by the searched result of optimum sequences, $\omega = 3, 5.47, 6.75$ for $\iota = 2, 3, 4$, respectively, the required accuracy is attained with the smallest value of N when large N is required. It is found that the interpolation scheme of $\iota = 4$ is advantageous when the number of function evaluations required is high.

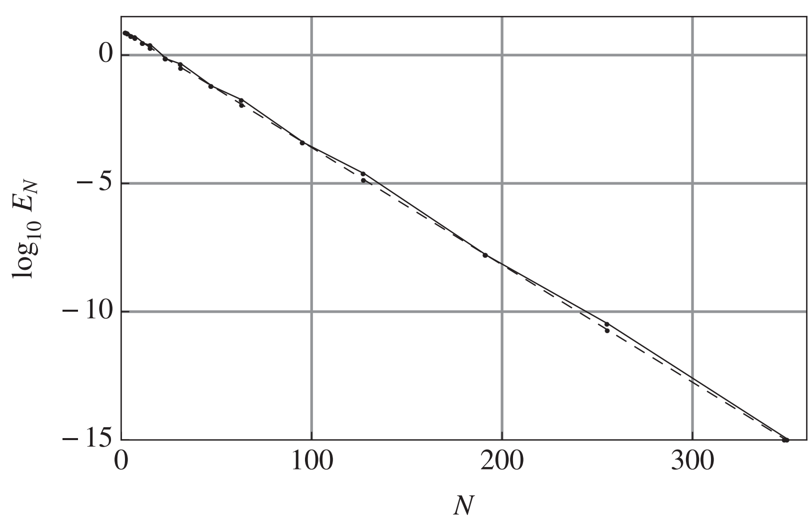

FIGURE 2. Errors E_N of polynomials p_N interpolating $(1 - ux)/(1-2ux+u^2)$, $u = 0.9$ in the case $\iota = 2$, where $N+1 = 2^j(3+\mu)+1$ ($\mu = 0, 1$) denotes the number of function evaluations. The solid line joins the errors E_N based on the present node set while the broken line joins those on the Chebyshev nodes.

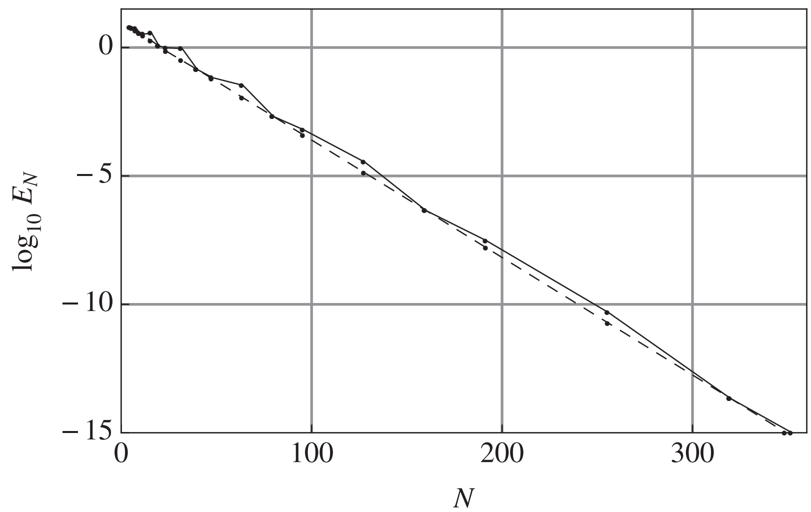

FIGURE 3. Errors E_N in the case $\iota = 3$, where $N = 2^j(5+\mu)$ $(\mu = 0, 1, 3)$

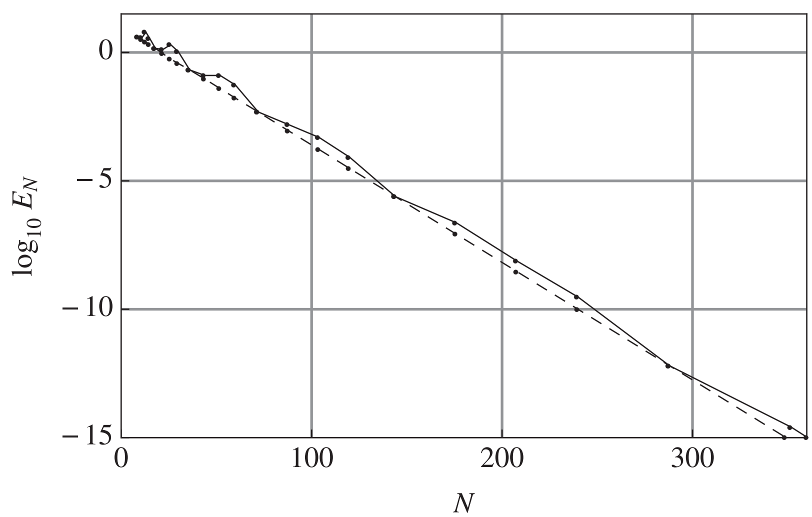

FIGURE 4. Errors E_N in the case $\iota = 4$, where $N = 2^j(9+\mu)$ $(\mu = 0, 2, 4, 6)$

REFERENCES

1. Z. Battles, L. N. Trefethen, *An extension of MATLAB to continuous functions and operators*, SIAM J. Sci. Comput., **25** (2004), 1743–1770. MR2087334 (2005e:41001)
2. J. P. Berrut, L. N. Trefethen, *Barycentric Lagrange interpolation*, SIAM Rev., **46** (2004), 501–517. MR2115059 (2005k:65018)
3. J. P. Boyd, *Computing zeros of a real interval through Chebyshev expansion and polynomial rootfinding*, SIAM J. Numer. Anal., **40** (2003), 1666-1682. MR1950617 (2003m:65071)

4. J. P. Boyd, D. H. Gally, *Numerical experiments on the accuracy of the Chbeyshev-Frobenius computation matrix method for finding the zeros of a truncated series of Chebyshev polynomials*, J. Comput. Appl. Math., **205** (2007), 281–295. MR2324840 (2008b:65065)
5. M. Branders, R. Piessens, *An extension of Clenshaw-Curtis quadrature*, J. Comp. Appl. Math., **1** (1975), 55-65. MR0371022 (51:7245)
6. C. W. Clenshaw, A. R. Curtis, *A method for numerical integration on an automatic computer*, Numer. Math., **2** (1960), 197–205. MR0117885 (22:8659)
7. J. W. Cooley, J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp., **19** (1965), 297–301. MR0178586 (31:2843)
8. P. J. Davis, *Interpolation and Approximation*, Dover, New York, NY, 1963. MR0157156 (28:393)
9. W. M. Gentleman, *Implementing Clenshaw-Curtis quadrature, II. Computing the cosine transformation*, Comm. ACM, **15** (1972), 337–342. MR0327001 (48:5343)
10. T. Hasegawa, T. Torii, I. Ninomiya, *Generalized Chebyshev interpolation and its application to automatic quadrature*, Math. Comp., **41** (1983), 537–553. MR717701 (84m:65037)
11. T. Hasegawa, T. Torii, H. Sugiura, *An algorithm based on the FFT for a generalized Chebyshev interpolation*, Math. Comp., **54** (1990), 195–210. MR990599 (91c:65009)
12. J. C. Mason, D. C. Handscomb, *Chebyshev Polynomials*, Chapman & Hall, 2003. MR1937591 (2004h:33001)
13. H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*, Springer, Berlin, 1981. MR606376 (83e:65219)
14. R. B. Platte, L. N. Trefethen, *Chebfun: A new kind of numerical computing*, Report No. 08/13, Oxford University Computing Laboratory, October (2008).
15. D. Potts, G. Steidl, M. Tasche, *Fast Algorithms for discrete polynomial transforms*, Math. Comp., **224** (1998), 1577–1590. MR1474655 (99b:65183)
16. S. C. Reddy, J. A. C. Weideman, *The accuracy of the Chebyshev differencing method for analytic functions*, SIAM J. Numer. Anal., **42** (2005), 2176–2187. MR2139243 (2006b:65026)
17. I. H. Sloan, W. E. Smith, *Product integration with the Clenshaw-Curtis points: Implementation and error estimates*, Numer. Math., **34** (1980), 387–401. MR577405 (81g:65030)
18. G. Steidl, M. Tasche, *A polynomial approach to fast algorithms for discrete Fourier-cosine and Fourier-sine transforms*, Math. Comp., **56** (1991), 281–296. MR1052103 (91h:65225)
19. H. Sugiura, T. Torii, *Polynomial interpolation on quasi-equidistributed nodes on the unit disk*, SIAM J. Numer. Anal., **29** (1992), 1154–1165. MR1173191
20. P. N. Swarztrauber, *Symmetric FFTs*, Math. Comp., **175** (1986), 323–346. MR842139 (88a:65157)
21. L. N. Trefethen, *Is Gauss quadrature better than Clenshaw-Curtis*, SIAM Rev., **50** (2008), 67–87. MR2403058 (2009c:65061)
22. J. Waldvogel, *Fast construction of the Fejér and Clenshaw-Curtis quadrature rules*, BIT, **46** (2006), 195–202. MR2214855 (2007k:65046)
23. J. A. C. Weideman, L. N. Trefethen, *The kink phenomenon in Fejér and Clenshaw-Curtis quadrature*, Numer. Math. **107** (2007), 707–727. MR2342649 (2008i:65048)

Department of Information Systems and Mathematical Sciences, Nanzan University, Seto, Aichi, 489-0863, Japan
E-mail address: sugiurah@ms.nanzan-u.ac.jp

Department of Information Science, University of Fukui, Fukui, 910-8507, Japan
E-mail address: hasegawa@fuis.fuis.u-fukui.ac.jp