Stability of Parallel Algorithms to Evaluate Chebyshev Series

R. Barrio
GME, Department of Applied Mathematics
University of Zaragoza, E-50009 Zaragoza, Spain

(Received February 2000; revised and accepted July 2000)

Abstract—In this paper, we present rounding error bounds of recent parallel versions of Forsythe's and Clenshaw's algorithms for the evaluation of finite series of Chebyshev polynomials of the first and second kind. The backward errors are studied by using the matrix formulation of the algorithm, whereas the forward error is also studied by means of a more direct approach that permits us to obtain sharper bounds. The bounds show an almost stable behavior as in the sequential algorithms. This fact is illustrated with several numerical tests. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords—Roundoff error, Parallel algorithms, Polynomial evaluation.

1. INTRODUCTION

Polynomial evaluation is one of the most common problems in scientific computing. Therefore, this problem has attracted the attention of many researchers and several algorithms suitable for parallel evaluation of power series have been proposed [1–5]. For the particular case of finite series of Chebyshev polynomials, a parallel algorithm was presented recently [6,7] that permits their efficient evaluation.

The error analysis for the evaluation of polynomials has received a great deal of attention in the literature. Backward and forward error analysis for Horner’s rule was first given by Wilkinson [8]. The behavior of the evaluation of Chebyshev representations of a polynomial is studied in [9–15], where one can find an error analysis of the evaluation of a Chebyshev series using Clenshaw's algorithm and variations of it. However, for the parallel algorithms, there is no theoretical analysis. It is known that, in general, the parallel algorithms can be much more unstable than the sequential algorithms [16], but in some particular problems, the parallel algorithms are as stable as the sequential algorithms [17,18]. Therefore, it is interesting to analyze the behavior of the recently proposed parallel algorithms to evaluate a Chebyshev series. In the analysis, we have used a matrix formulation that is equivalent to the algorithms given in [6,7]. This formulation permits us to use the classical rounding error techniques for linear systems. Also, we present an alternative forward error bound analysis based on a direct method [12,13,19]. From these bounds, it is established that the parallel algorithms are almost as stable as the sequential ones.
The parallel bounds are similar to those of the sequential algorithms, although new points where the rounding error can grow appear in the parallel algorithms.

The paper is organized as follows. In Section 2, we review the sequential and parallel Forsythe and Clenshaw algorithms for the evaluation of a Chebyshev series. In Section 3, we introduce the stability analysis of the parallel algorithms and, in Section 4, we show some numerical tests to compare the theoretical bounds and the simulated rounding errors.

2. PRELIMINARIES

In this section, we summarize some classical results on error analysis (see, for example, the excellent monography of Higham [20]). In the paper, we assume that the computations are carried out in a floating-point arithmetic that obeys the models

\[ \text{fl}(x \text{ op } y) = (x \text{ op } y)(1 + \rho), \quad \text{fl} (x \text{ op } y) = \frac{(x \text{ op } y)}{1 + \alpha}, \quad |\rho|, |\alpha| \leq u, \quad (1) \]

where \( \text{op} \in \{+, -, \times, \div\} \) and \( u \) is the unit roundoff of the computer. Also, we denote \( \gamma_n := nu/(1 - nu) = nu + \mathcal{O}(u^2) \) and we assume the notation \( \hat{a} \) for the computed value of \( a \).

First we present two algorithms for the serial evaluation of finite series of Chebyshev polynomials. The algorithms are Clenshaw's [21] and Forsythe's [22] algorithms. The Chebyshev polynomials are orthogonal polynomials in the real interval \([-1, 1]\). A general family of orthogonal polynomials \( \{\phi_n(x)\} \) satisfies a triple recurrence relation

\[ \phi_0(x) = 1, \quad \phi_1(x) = \alpha_1(x), \]
\[ \phi_k(x) - \alpha_k(x) \phi_{k-1}(x) - \beta_k \phi_{k-2}(x) = 0, \quad k \geq 2, \quad (2) \]

with \( \alpha_k(x) \) a linear polynomial of \( x \). We remark that for the Chebyshev polynomials of the first kind, the coefficients are \( \alpha_1(x) = x, \alpha_i(x) = 2x \) (for all \( i > 1 \)), and \( \beta_i = -1 \) and for the Chebyshev polynomials of the second kind \( \alpha_1(x) = 2x \) and \( \beta_i = -1 \).

Let \( p_n(x) = \sum_{i=0}^{n} c_i \phi_i(x) \) be a finite series of Chebyshev polynomials. Clenshaw's algorithm to evaluate \( p_n(x) \) can be expressed as

\[ q_{n+1} = q_{n+2} = 0 \]
\[ \text{for} \quad k = n \quad \text{to} \quad 1 \quad \text{by} \quad -1 \]
\[ q_k = c_k + 2x q_{k+1} - q_{k+2} \]
\[ \text{end} \]
\[ p_n(x) = c_0 + \alpha_1(x) q_1 - q_2 \]

with \( \alpha_1(x) = x \) for the Chebyshev polynomials of the first kind and \( \alpha_1(x) = 2x \) for the Chebyshev polynomials of the second kind.

This algorithm can also be formulated in a matrix way. Let \( A \in \mathbb{R}^{(n+1) \times (n+1)} \) be the matrix

\[ A = \begin{pmatrix}
1 & -\alpha_1 & 1 \\
1 & -2x & \ddots \\
& \ddots & \ddots & 1 \\
& & 1 & -2x \\
& & & 1 \\
\end{pmatrix}; \quad (3) \]

then Clenshaw's algorithm is equivalent to solving the tridiagonal upper triangular linear system

\[ Aq = c, \quad (4) \]

where \( q, c \in \mathbb{R}^{n+1} \) are the vectors \( q = (q_0, q_1, \ldots, q_n)^\top \) and \( c = (c_0, c_1, \ldots, c_n)^\top \).
The Forsythe algorithm, which uses in a direct way the recurrence (2), can be expressed, for the particular case of Chebyshev polynomials, as

\[ t_0 = 1, \quad f_0 = c_0, \]
\[ t_1 = a_1(x), \quad f_1 = f_0 + c_1 t_1, \]
for \( k = 2 \) to \( n \)
\[ t_k = 2x t_{k-1} - t_{k-2}, \]
\[ f_k = f_{k-1} + c_k t_k, \]
end
\[ p_n(x) = f_n \]

This algorithm can also be formulated in a matrix form. First, evaluating the orthogonal polynomials is equivalent to solving the linear system

\[ F \phi = e_{n+1}, \]
where \( F \in \mathbb{R}^{(n+1) \times (n+1)} \) is given by

\[
F = \begin{pmatrix}
1 & -2x & 1 \\
1 & \ddots & \ddots \\
& \ddots & -2x & 1 \\
& & 1 & -a_1 \\
& & 1 & 1
\end{pmatrix},
\]
and \( \phi, e_{n+1} \in \mathbb{R}^{n+1} \) are the vectors \( \phi = (\phi_n(x), \ldots, \phi_1(x), \phi_0(x))^T \), and \( e_{n+1} = (0, \ldots, 0, 1)^T \), and then to perform the inner product

\[ p_n(x) = c^T \phi = \sum_{i=0}^{n} c_i \phi_i(x). \]

In the formulation, \( \phi_i(x) \) stands for the Chebyshev polynomial \( T_i(x) \) or \( U_i(x) \).

Now we present the parallel versions of the above algorithms [6,7]. In the parallel algorithms, we suppose that the degree \( n \) of the polynomial satisfies \( n = kp - 1 \), \( p \) being the number of processors (in the general case, the analysis is similar).

The parallel Clenshaw's algorithm (hereafter, the ChPC algorithm) to evaluate \( p_n(x) = \sum_{r=0}^{n} c_r T_r(x) \) can be written as the following algorithm.

**Step ChPC-1:** First, calculate an initialization: \( T_k(x), \quad k = 2, \ldots, p. \)

**Step ChPC-2:** Next, compute \( P^{0,p}(x), \ldots, P^{p-1,p}(x) \) in parallel using \( p \) processors, with

\[ P^{m,p}(x) = B_{0}^{m,p} T_m(x) - B_{1}^{m,p} T_{p-m}(x), \]

where \( B_0^{m,p} \) and \( B_1^{m,p} \) are evaluated with the recurrence

\[ B_{k+1}^{m,p} = B_{k}^{m,p} - 0, \]
\[ B_{r+1}^{m,p} = c_{m+r+1} T_{r+1}(x) B_{r+2}^{m,p} - B_{r+1}^{m,p}, \quad r = k - 1, \ldots, 0. \]

**Step ChPC-3:** Finally, compute the value of the polynomial

\[ p_n(x) = \sum_{m=0}^{p-1} P^{m,p}(x). \]

The parallel Forsythe's algorithm (hereafter, the ChPF algorithm) to evaluate \( p_n(x) = \sum_{r=0}^{n} c_r T_r(x) \) can be written as the following algorithm.
Step ChPF-1: First, calculate an initialization: $T_i(x)$, $i = 2, \ldots, 2p$.

Step ChPF-2: Next, compute $F^0P(x), \ldots, F^{p-1}P(x)$ in parallel using $p$ processors, with the recurrence for the processor $m$,

$$
\begin{align*}
F^{m,p} &= c_m T_m(x) + c_{m+p} T_{m+p}(x) \\
T_{r+p+m}(x) &= 2T_p(x)T_{(r-1)p+m}(x) - T_{(r-2)p+m}(x) \\
F^{m,p} &= F^{m,p} + c_{r+p} T_{r+p+m}(x)
\end{align*}
$$

(11)

Step ChPF-3: Finally, compute the value of the polynomial

$$
p_n(x) = \sum_{m=0}^{p-1} F^{m,p}(x).
$$

(12)

The parallel algorithms to evaluate finite series of Chebyshev polynomials of the second kind ($\sum_{r=0}^{n} c_r U_r(x)$) are slightly different from the algorithms for the Chebyshev polynomials of the first kind (see [7] for more details), and the error analysis of both algorithms is quite similar, with very small differences, so we only focus our attention on the first case.

3. STABILITY ANALYSIS

In this section, we analyze the algorithms by using a matrix formulation involving the solution of tridiagonal upper triangular linear systems. Besides, due to the parallel nature of the algorithms, the matrices have a block structure. The parallel algorithm reduces the evaluation of the sequential recurrence to the evaluation of $p$ subrecurrences.

First, we analyze the ChPC algorithm [7]. This algorithm can be reformulated by using a block matrix notation. Let $S \in \mathbb{R}^{(n+1)\times(n+1)}$,

$$
S = \text{diag}\{S_p, S_p, \ldots, S_p\},
$$

(13)

with $S_p \in \mathbb{R}^{k \times k}$,

$$
S_p = \begin{pmatrix}
1 & -2T_p(x) & 1 \\
1 & -2T_p(x) & 1 \\
& \ddots & \ddots & \ddots \\
1 & -2T_p(x) & 1 \\
1 & -2T_p(x) & 1
\end{pmatrix}.
$$

(14)

Also, we define the vectors $e_{p+1}, \phi_{0:p} \in \mathbb{R}^{p+1}$, and $q^*, c^* \in \mathbb{R}^{n+1}$ given by $e_{p+1} = (0, \ldots, 0, 1)^T$, $\phi_{0:p} = (\phi_0(x), \ldots, \phi_p(x))^T$ (the values of the Chebyshev polynomials), and $c^* = (c_0^*, c_1^*, \ldots, c_{p-1}^*)$ with $c_{k}^* = (c_1, c_{i+p}, c_{i+2p}, \ldots, c_{i+(k-1)p})$ (the vector of the coefficients of the polynomial). Note that as each processor needs a different set of coefficients, we have joined in blocks the terms used for each processor, so we have permuted the vectors $q$ and $c$ in a suitable way, obtaining $q^*$ and $c^*$. Besides, for the initialization process, we need the matrix $S_1 \in \mathbb{R}^{(p+1)\times(p+1)}$,

$$
S_1 = \begin{pmatrix}
1 & -2x & 1 \\
1 & -2x & 1 \\
& \ddots & \ddots & \ddots \\
1 & -2x & 1 \\
1 & -2x & 1
\end{pmatrix},
$$

(15)

where we remark that $\phi_1(x) = x$ in the case of $\{T_i(x)\}$ and $\phi_1(x) = 2x$ in the case of $\{U_i(x)\}$.
Thus, the ChPC algorithm can be rephrased as follows.

**Step ChPC-1**: Solve the triangular upper system: \( S_1 \phi_{0,p} = e_{p+1} \) (initialization process).

**Step ChPC-2**: Solve in parallel the triangular upper system: \( S q^* = c^* \).

**Step ChPC-3**: Finally, evaluate the polynomial: \( p_n(x) = \sum_{i=0}^{p-1} P_i^p(x) \) with \( P_i^p(x) = q^*_{i+k+1} - q^*_{i+k+2} \).

In the following, in order to simplify the notation, we will write \( q,c \) instead of \( q^*,c^* \).

Now we can study the backward stability of the ChPC algorithm in the evaluation of a Chebyshev series in \( p \) processors. In fact, the only part that we have to comment on is Step ChPC-2 in the parallel algorithm; that is, the solution of the triangular upper tridiagonal systems of the algorithm. The solution of \( S q = c \) satisfies the general bounds for triangular systems [20] applied to triangular and tridiagonal systems; that is

\[
(S + \Delta S)q = c, \quad \text{with } |\Delta S| \leq \gamma_2 |S|.
\]

(16)

We note that the errors in the initialization process do not affect the body of the parallel algorithm (Step ChPC-2*). By using (16) and taking into account the structure of the matrix \( S \), we can easily give forward error bounds of the algorithm.

**Theorem 1.** The relative normwise forward error in the solution \( q \) of system (4) obtained by the parallel Clenshaw algorithm, in the absence of errors in the evaluation of \( T_p(x) \), is bounded as follows:

\[
\frac{\|\delta q\|_{\infty}}{\|q\|_{\infty}} \leq \gamma_2 \text{cond} (S_p),
\]

where \( \text{cond} (S_p) = \| (S_p^{-1}) S_p \|_{\infty} \) is the Bauer-Skeel's [23] componentwise condition number.

Now we analyze the ChPF algorithm [6]. Let \( R \in \mathbb{R}^{(n+1) \times (n+1)} \),

\[
R = \text{diag} \{ R_p, R_p, \ldots, R_p \}, \quad p \text{ times}
\]

with \( R_p \in \mathbb{R}^{k \times k} \),

\[
R_p = \begin{pmatrix}
1 & -2T_p(x) & 1 \\
1 & -2T_p(x) & 1 \\
\vdots & \vdots & \vdots \\
1 & -2T_p(x) & 1 \\
1 & 1 & 1
\end{pmatrix},
\]

and let \( I \in \mathbb{R}^{(n+1) \times (n+1)} \),

\[
I = \begin{pmatrix}
\mathbb{I}_{k-2} & M_2 & \cdots & M_2^T \\
\mathbb{I}_{k-2} & M_2 & \cdots & M_2^T \\
\mathbb{I}_{k-2} & \cdots & \cdots & \cdots \\
\mathbb{I}_{k-2} & \cdots & \cdots & M_2^T \\
\mathbb{I}_{k-2} & \cdots & \cdots & M_2^T
\end{pmatrix},
\]

(18)

where \( \mathbb{I}_{k-2} \) stands for the identity matrix \( \in \mathbb{R}^{(k-2) \times (k-2)} \) and \( M_2 = (1 - 2x) \). The matrix \( M_2^* = (1 - x) \) gives the difference between the algorithm for Chebyshev polynomials of the first
and second kind (with $M_2$ instead of $M_2^2$). Also, we define the vectors $e_{n+1} = (0, \ldots, 0, 1)^T$, $c^* = (c_{k-1}^p, \ldots, c_k^p, c_0^p)^T$ with $c_k = (c_{k+(k-1)p}, c_{k+(k-2)p}, \ldots, c_p, c_0)$ (the permuted vector of the coefficients $c$ of the polynomial) and $\phi^* = (\phi_{k-1}^p, \ldots, \phi_k^p, \phi_0^p)^T$ with $\phi_k = (\phi_{k+(k-1)p}(x), \phi_{k+(k-2)p}(x), \ldots, \phi_{k+p}(x), \phi_0(x))$ (the permuted vector of the values of the Chebyshev polynomials). The vectors $e_{n+1}, \phi^*$ and $c^* \in \mathbb{R}^{n+1}$.

Thus, the ChPF algorithm can be rephrased as follows.

**Step ChPF-1**: Solve the triangular upper system: $Iy = e_{n+1}$.

**Step ChPF-2**: Solve in parallel the triangular upper system: $R\phi^* = y$.

**Step ChPF-3**: Finally, evaluate the polynomial by means of the inner product:

$$p_n(x) = \sum_{m=0}^{p-1} \left( \sum_{j=0}^{k-1} c_{jp+m} \phi_{jp+m}(x) \right) = \sum_{m=0}^{p-1} c_{k}^m (\phi_k^m)^T. \quad (19)$$

As above and in the following, we use the notation $\phi$ and $c$ instead of $\phi^*$ and $c^*$.

Also, we note that if we permute the files of the matrix $I$ (18), we obtain the notation $\phi$ and $c$ instead of $\phi^*$ and $c^*$.

Therefore, looking at the rounding errors, this matrix behaves as the submatrix $I_{2p}$.

Now, we can study the backward stability of the ChPF algorithm in the evaluation of a Chebyshev series in $p$ processors.

**Theorem 2.** The computed value $\tilde{p}_n(x)$ by means of the parallel Forsythe algorithm satisfies

$$\tilde{p}_n(x) = (c + \Delta c)^T \tilde{\phi}, \quad (21)$$

with $\Delta c = (\Delta c_{ip+m})$ such that $|\Delta c_{ip+m}| \leq \gamma_{(k+p)-(i+m)} \cdot |c_{ip+m}| + O(u^2)$ and where $\tilde{\phi} = (\tilde{\phi}_i(x))$ is the computed solution of (5), which satisfies

$$(F + \Delta F) \tilde{\phi} = e_{n+1}, \quad |\Delta F| \leq \gamma_0 \|I R\|. \quad (22)$$

**Proof.** Equation (21) is obtained by applying the classical results for inner products (see [20, Section 3.1]) and taking into account that in the parallel algorithm, we perform first the inner product in each subseries, and finally, we perform the addition of the partial sums.

Next, we study the error in the solution of the upper tridiagonal systems of the algorithm. The solution of $Iy = e_{n+1}$ and $R\phi = y$ (Steps ChPF-1* and ChPF-2* in the algorithm) satisfy the general bounds for triangular systems [20] applied to triangular and tridiagonal systems; that is,

$$\begin{align*}
(I + \Delta I)\hat{y} &= e_{n+1}, \quad \text{with } |\Delta I| \leq \gamma_2 \|I\|,
(R + \Delta R) \hat{\phi} &= \hat{y}, \quad \text{with } |\Delta R| \leq \gamma_2 \|R\|.
\end{align*} \quad (23)$$

So, since $F = IR$, then $F\phi = IR \phi = e_{n+1}$, and it follows that

$$e_{n+1} = (F + \Delta F) \hat{\phi} = (I + \Delta I)(R + \Delta R) \hat{\phi} = (F + \Delta I R + \Delta R + I \Delta I \Delta R) \hat{\phi}.$$
Thus, we have

\[ |\Delta F| \leq \gamma_1 |I| |R| + \gamma_2 |I| |R| + \gamma_2 \gamma_1 |I| |R| \leq \gamma_6 |I| |R|, \]

where we have applied that \( \gamma_i + \gamma_j \leq \gamma_{i+j} \) and \( \gamma_i \gamma_j \leq \gamma_{\min(i,j)} \). Now the result follows because \( |I| |R| = |I \cdot R| \) due to the special structure of the matrices \( I \) and \( R \).

As a direct consequence of Theorem 2, we obtain the forward stability of the algorithm.

**Theorem 3.** The relative normwise forward error in the solution \( \phi \) of system (5) obtained by the parallel Forsythe algorithm, in the absence of errors in the evaluation of \( T_p(x) \), is bounded as follows:

\[
\frac{\|\delta \phi\|_\infty}{\|\phi\|_\infty} \leq \gamma_6 \text{cond} (IR).
\]

In the ChPF algorithm, we also have to take into account the rounding error in the parallel evaluation of the inner product (19). Following the classical analysis of the rounding errors in the evaluation of inner products and sums, we easily obtain

\[
|p_n(x) - \Phi(p_n(x))| = 2u \sum_{m=0}^{p-1} \left\{ \sum_{j=0}^{k-1} |c_{jp+m}| |\phi_{jp+m}(x)| \right\} + u \sum_{m=0}^{p-1} \left\{ \sum_{j=0}^{k-1} |c_{jp+m}| |\phi_{jp+m}(x)| \right\} + O(u^2)
\]

\[
= u \sum_{m=0}^{p-1} \left\{ \sum_{j=0}^{k-1} (2k + p - 2 - 2j - m) |c_{jp+m}| |\phi_{jp+m}(x)| \right\} + O(u^2).
\]

In the previous analysis, we have assumed that the value \( T_p(x) \) is given without any rounding error. Normally, this term has to be calculated in an initialization process by means of the triple recurrence or by means of Clenshaw's algorithm. In that case, we introduce rounding errors in the calculation of \( T_p(x) \). Following the analysis of the Clenshaw algorithm, but now with the matrix \( S_1 \) of equation (15), we obtain

\[
\frac{\|\delta T_p(x)\|_\infty}{T_p(x)} \leq \gamma_2 \text{cond} (S_1).
\]

The condition number verifies \( \text{cond} (S_1) \leq \|S_1^{-1}\|_\infty \|S_1\|_\infty \leq 4 \|S_1^{-1}\|_\infty \). Since

\[
S_1^{-1} = (s_{ij}^{-1}) = \begin{cases} 0, & j < i, \\ U_{j-i}(x), & j \geq i, \end{cases}
\]

and taking into account the bounds [24]

\[
\|U_i(x)\|_\infty \leq i + 1, \\
|U_i(x)| = \frac{\left| \sin \left( (i+1) \arccos x \right) \right|}{\left| \sin(\arccos x) \right|} \leq \frac{1}{\sqrt{1 - x^2}}, \quad x \neq \pm 1,
\]

we obtain

\[
\text{cond} (S_1) \leq 4 \sum_{i=0}^{p} |U_i(x)| \leq \begin{cases} 2(p+1)(p+2), & x \in [-1, 1], \\ \frac{4(p+1)}{\sqrt{1 - x^2}}, & x \neq \pm 1, \end{cases}
\]

and so

\[
\text{cond} (S_1) \leq \min \left\{ 2(p+1)(p+2), \frac{4(p+1)}{\sqrt{1 - x^2}} \right\}.
\]
Once we have the bound for the rounding errors in the evaluation of \( T_p(x) \), we have to introduce it into the backward error bounds of the parallel algorithms. Thus, in the ChPC algorithm, we introduce \( \hat{S} \) instead of \( S \) in equation (16) because in the matrix \( S \), we need the term \( T_p(x) \), which is not calculated exactly, then \( \hat{S} = S + \Delta \hat{S} \) with \( |\Delta \hat{S}| \leq E_T, |S| \) being \( E_T \), such that \( \left( |T_p(x) - T_p(x)| / |T_p(x)| \right) \leq E_T \), and so \( E_T \leq \gamma_2 \text{cond} (S_1) \). Therefore, up to second order in \( u \), we obtain the following theorem.

**Theorem 4.** The relative normwise forward error in the solution \( q \) of system (4) obtained by the parallel Clenshaw algorithm is bounded, up to second order in \( u \), as follows:

\[
\frac{\| \delta q \|_\infty}{\| q \|_\infty} \leq u \cdot \min \left\{ 2 + 4(p + 1)(p + 2), 2 + \frac{8(p + 1)}{\sqrt{1 - x^2}} \right\} \text{cond} (S_p) + O (u^2). \tag{28}
\]

In the ChPF algorithm, we introduce \( \hat{R} \) instead of \( R \) in equation (23) and so \( \hat{R} = R + \Delta \hat{R} \) with \( |\Delta \hat{R}| \leq E_T, |R| \).

It is interesting to remark that equation (28) tells us that we have to expect a higher rounding error when we evaluate the Chebyshev polynomials at the end of the interval \( x = \pm 1 \).

Since the forward error bounds appear in the condition number of the matrix \( S_p \) of equation (14), it is interesting to compare it with the sequential case, where a forward error bound is given by [20]

\[
\frac{\| \delta q \|_\infty}{\| q \|_\infty} \leq \gamma_2 \text{cond} (A), \tag{29}
\]

with \( A \) given by (3). Following an analogous analysis like that used in the above for matrix \( S_1 \), as \( T_p(x) \in [-1, 1] \), we obtain that

\[
\text{cond} (A) \leq 4 \sum_{j=0}^{n} |U_j(x)| \leq \min \left\{ 2(n + 1)(n + 2), \frac{4(n + 1)}{\sqrt{1 - x^2}} \right\},
\]

\[
\text{cond} (S_p) \leq 4 \sum_{i=0}^{k-1} |U_i (T_p(x))| \leq \min \left\{ 2k(k + 1), \frac{4k}{\sqrt{1 - T_p(x)^2}} \right\}. \tag{30}
\]

To give an idea of the behavior of the ratio between the parallel and the sequential algorithms, we give an estimation. We remark that the following formula is not a bound, it is just an estimation of the ratio. Taking into account Theorem 4 and equations (29) and (30), we obtain

\[
\frac{\| \delta q \|_{\text{parallel}}}{\| \delta q \|_{\text{sequential}}} \approx \frac{(2 + E_T_p) \text{cond} (S_p)}{2 \text{cond} (A)} \sim \frac{1 + 4 \sum_{l=0}^{p} |U_l(x)|}{\sum_{i=0}^{n} |U_i(x)|} \tag{31}
\]

that can be estimate (30) by

\[
\frac{\| \delta q \|_{\text{parallel}}}{\| \delta q \|_{\text{sequential}}} \sim \begin{cases} 
\text{est}_1 = \frac{2(p + 1)(p + 2)k(k + 1)}{(n + 1)(n + 2)}, & x = 1 - \varepsilon, -1 + \varepsilon, \\
\text{est}_2 = \frac{2(p + 1)k(k + 1)}{n + 1}, & x = \cos \frac{m \pi}{p} \pm \varepsilon, (m = 1, \ldots, p - 1), \\
\text{est}_3 = \frac{4(p + 1)k}{n + 1} \frac{1}{\sqrt{1 - T_p(x)^2}}, & x \neq \cos \frac{m \pi}{p}, (m = 0, \ldots, p).
\end{cases}
\]

Note that

\[
\text{est}_1 \searrow 2, \quad \text{est}_2 \simeq 2k = \frac{2n}{p}, \quad \text{est}_3 \simeq \frac{4}{\sqrt{1 - T_p(x)^2}}.
\]
From the estimations, we expect to have an increment of the rounding errors close to the relative extrema of $T_p(x)$; that is, at the points $x_i = \cos(m \pi/p)$, $m = 1, \ldots, p-1$. The reason for this is that at these points, the parallel recurrence evaluates $p$ subrecurrences at the point $T_p(x_i) = \pm 1$.

In Figure 1, we illustrate formula (31), that is, an estimation of the ratio of the theoretical rounding error bounds in the parallel and sequential Clenshaw’s algorithms, in the evaluation of a Chebyshev series of degree $n = 3199$ for several numbers of processors $p$. In the figure, we observe the phenomena predicted by the above estimations: when the number of processors grows, new points appear where the error ratio increases significantly. We also remark that the increment in the error, presented at these points, decreases in size as $p$ increases.

![Figure 1](image_url)

Figure 1. Estimation of the ratio between the theoretical rounding error bounds (31) in the parallel and sequential Clenshaw algorithms in the evaluation of a Chebyshev series of degree $n = 3199$ for several numbers of processors $p$.

### 3.1. Direct Analysis

In this section, we present an analysis of the forward error of the parallel algorithms in a more direct way, following the analysis of [12,13,19]. This analysis gives sharper rounding error bounds than the previous one in the cases when the condition numbers of the matrices are high [19] and when the coefficients of the polynomial are small, because this new bound depends on the particular polynomial that we have. We present only the results for the ChPC algorithm. For the ChPF algorithm, the analysis is quite similar.

First, we present one theorem from [19] that gives a forward error bound for the sequential Clenshaw algorithm applied to the evaluation of a finite series of a general family of orthogonal polynomials.
**Theorem 5.** The error in the evaluation of an orthogonal series \( p_n(x) = \sum_{k=0}^{n} c_k \phi_k(x) \) by means of the Clenshaw algorithm verifies

\[
|\hat{p}_n(x) - p_n(x)| \leq u \sum_{s=0}^{n} \rho_s(x)|c_s| + \mathcal{O}(u^2),
\]

where

\[
\begin{align*}
\rho_0(x) &= 4 |\phi_0(x)|, \\
\rho_s(x) &= 4 |\phi_s(x)| + \sum_{k=1}^{s-1} \Delta_{k,s} |\phi_k(x)|, \quad \text{for } s = 1, \ldots, n,
\end{align*}
\]

and

\[
\Delta_{k,s} = 2 |a_{k+1,s+1}^{-1}| + 3 |\alpha_{k+1}| |a_{k+2,s+1}^{-1}|, \quad \text{for } k = 1, \ldots, s - 1,
\]

where \( a_{k,s} \) are the elements of \( A^{-1} \).

Note that the matrix \( A \) is adapted to our particular family of polynomials, Chebyshev polynomials, but in the case of a general family of orthogonal polynomials, where it can be applied to Theorem 5, there will appear the coefficients \( \alpha_i(x) \) and \( \beta_i \) of the general recurrence (2).

Our study of the ChPC algorithm follows the proof of Theorem 5 [19].

**Theorem 6.** The error in the parallel evaluation on \( p \) processors of a Chebyshev series \( p_n(x) = \sum_{i=0}^{n} c_i \phi_i(x) \), with \( \phi_i(x) = T_i(x) \) or \( U_i(x) \), by means of the parallel Clenshaw algorithm verifies

\[
|\hat{p}_n(x) - p_n(x)| \leq u \sum_{m=0}^{p-1} \sum_{s=0}^{k-1} \rho_s^m(x)|c_{s+p+m}| + \mathcal{O}(u^2),
\]

where

\[
\begin{align*}
\rho_0^m(x) &= (p + 3 - m) |\phi_m(x)|, \\
\rho_s^m(x) &= (p + 3 - m) |\phi_{s+p+m}(x)| + \sum_{j=1}^{s-1} \Delta_{j,s}^m |\phi_{j+p+m}(x)|, \quad s = 1, \ldots, k - 1,
\end{align*}
\]

and

\[
\Delta_{j,s}^m = 2 |U_{s-j}(x)| + (3 + E_{T_p}) |2T_p(x)| |U_{s-j-1}(x)|, \quad j = 1, \ldots, s - 1,
\]

being \( E_{T_p} \) the relative rounding error in the evaluation of \( T_p(x) \) (equations (26) and (27)).

**Proof.** The proof follows from Theorem 5. The only differences are that now the coefficient \( \alpha_{k+1} = 2 T_p(x) \) and that we have to consider the rounding error \( E_{T_p} \) in the evaluation of \( T_p(x) \), which is given by (27). The term \( E_{T_p} \) is introduced in the terms \( \Delta_{j,s}^m \), the place where we consider the coefficients \( \alpha_i \). Also, we have to take into account Step ChPC-3, which gives the terms \( \sum_{m=0}^{p-1} \sum_{s=0}^{k-1} (p - 1 - m)|\phi_{s+p+m}(x)||c_{s+p+m}| \). Besides, in this case, the term \( a_{j,s}^{-1} \) is the element \((j,s)\) of \( S_p^{-1} \) (14) and it is equal to \( U_{s-j} \).

Note that in the rounding errors due to Step ChPC-3 of the parallel algorithm, we have supposed that the evaluation of the addition is performed from \( m = 0 \) to \( p - 1 \), but the order can be changed, and in some cases, we cannot control the order in the addition, as in global reduction operations in MPI. Therefore, if the order of evaluation changes, it also will change the factor \((p - 1 - m)\) that multiplies each subset of coefficients. Also, if we can evaluate the factor \( 2T_p(x) \) with high precision, then \( E_{T_p} \) can be considered to be depreciable, obtaining lower rounding errors in the parallel process.
Figure 2. Absolute rounding errors of the parallel Clenshaw algorithm for the evaluation of series of degree $n = 3199$ of Chebyshev polynomials of the first kind with random coefficients (Problem 2).

Figure 3. Relative theoretical rounding error bounds (C and D) and numerical simulations (S) of the parallel Clenshaw algorithm for the evaluation of a series of degree $n = 3199$ of Chebyshev polynomials of the first kind with monotonically decreasing coefficients (Problem 81).

4. NUMERICAL TESTS

We have tested the ChPC and ChPF algorithms in order to analyze the effects of rounding errors, but, since the behavior of both algorithms is very similar, we only present the results for the ChPC algorithm. In the simulations, we have studied the algorithms with finite series of
degree \( n = 3199 \) of Chebyshev polynomials of the first kind. We have used two sets of coefficients: set S1 of monotonically decreasing coefficients \( c_i = 1/(i+1)^2 \) and set S2 of random coefficients normally distributed with mean 0 and variance 1. For each series, each set of coefficients and each point, we have performed 500 simulations in double precision with unit roundoff \( \alpha \approx 2.2 \times 10^{-16} \). All the tests were done on a workstation SUN ULTRASPARC 1 and the programs were written in FORTRAN 77. On each test, we consider as the result the maximum absolute rounding error of the 500 simulations of polynomial evaluation.

The first question is to see if the increments in the rounding errors detected in the theoretical analysis also appear in the numerical tests. In Figure 2, we present the absolute rounding errors in the evaluation of a Chebyshev series with random coefficients (set S2). In this case, the increments at the relative extrema of the Chebyshev polynomial \( T_p(x) \), that is \( x_i = \cos\left(\frac{m \pi}{p}\right) \), \( m = 1, \ldots, p-1 \), are clear (in the figure, we have plotted these points with discontinuous vertical lines).

In Figure 3, we present the numerical simulations (S) of the relative rounding error in the evaluation of a polynomial with the set S1 of coefficients. In addition, we give the two theoretical bounds, Theorems 4 (bound C) and 6 (bound D). The bound D, as it is an absolute rounding error bound, is divided by the absolute value of the series. From the figures, we observe that bound D approaches the numerical simulations much better than bound C. The numerical simulations do not present all of the increments derived from the theoretical analysis, only in the figures of \( p = 4 \) and \( p = 8 \) some increments are perceptible. The reason is that this finite series is very stable, and then the rounding errors are very small (see the figure with \( p = 1 \)). On the contrary, in Figure 4, we present the relative rounding error in the evaluation of a polynomial with the set S2 of coefficients and the two theoretical bounds. In this case, the increments appear in all
the points predicted by the theory. Besides, both theoretical bounds are close to the numerical simulations, giving sharp error bound estimations. The size of the increments, detected in the numerical and theoretical analysis, decreases when $p$ grows, but the number of points where the instability appears increases (it is equal to $p - 1$).

4.1. Conclusions

From the numerical tests and the theoretical rounding error bounds, we conclude that the parallel algorithms are almost as stable as the sequential ones, but at some points, the rounding errors increase slightly. Note that these points are not random inside the interval, their position is well known, which is the relative extrema of $T_p(x)$ when we use $p$ processors.

REFERENCES