MATHEMATICS OF COMPUTATION, VOLUME 33, NUMBER 147
JULY 1979, PAGES 1019–1031

Stable Evaluation of Polynomials in Time $\log n$

By Roland Kusterer and Manfred Reimer

Abstract. An algorithm is investigated which evaluates real polynomials of degree n in time $\log n$ at asymptotically minimum costs. The algorithm is considerably stable with respect to round-off.

1. Introduction. The usual algorithms which evaluate a real polynomial of degree n on the interval $[-1, 1]$ *in time* $\log n$ are based on the monomial representation; see Borodin and Munro [1]. Therefore, due to big coefficients, they do not avoid large intermediate results in absolute value so that their *error norm*, which is a measure for the instability of the algorithm caused by round-off (for definition see Reimer [4]), is increasing rapidly at an exponential rate (Reimer [3]). On the other hand, Clenshaw's algorithm based on the Chebyshev polynomials of the second kind is an example of an algorithm with an $O(n^2 \log n)$ error norm (see Reimer [3]) which is highly favorable if we think of round-off only. Unfortunately, it does not allow parallel processing, i.e. it cannot be performed at a time rate increasing slower than n.

For this reason we are going to investigate a different algorithm which is combining

(i) asymptotically minimum costs in total ($\sim n$ additions/multiplications),

(ii) possibility of parallel processing (in time $\sim 2 \log n$, dual logarithm),

(iii) favorable error norm growing not faster than at an $O(n^{5/2})$-rate.

We should emphasize that, though there are algorithms slightly superior to ours with respect to each single issue above, we not know of any where the conditions (i), (ii) and (iii) are valid simultaneously.

2. The Algorithm. Let T_ν denote the Chebyshev polynomial of the first kind and with degree ν. Assume P to be any real polynomial of degree $n = 2^k - 1 \in \mathbf{N}$. Then, starting with $S_0^{(k)} := P$, we get the decomposition

$$S_0^{(k)} = \tau_{k-1} S_{01}^{(k-1)} + S_{00}^{(k-1)},$$

where $\tau_{k-1} := 2T_{2^{k-1}}$, and where $S_{01}^{(k-1)}$ and $S_{00}^{(k-1)}$ are polynomials of degree $2^{k-1} - 1$. The decomposition is repeated with the polynomials $S_{0j}^{(k-1)}$ instead of $S_0^{(k)}$ and with $k-1$ instead of k, and so on. So we obtain polynomials $S_{\cdots j}^{(k-i)}$, where $i \in \{0, 1, \ldots, k\}$ and $j \in \{0, 1\}$, which satisfy the recurrence relations

$$S_{\cdots j}^{(k-i)} = \tau_{k-i-1} S_{\cdots j,1}^{(k-i-1)} + S_{\cdots j,0}^{(k-i-1)} \tag{2.1}$$

Received July 14, 1977; revised September 4, 1978.
AMS (MOS) subject classifications (1970). Primary 65D20, 65G05.
Key words and phrases. Polynomial evaluation, parallel processing, growth of the error norm.

for $i = 0, 1, \ldots, k-1$ and $j = 0, 1$, the degree of $S_{\cdots}^{(k-i)}$ not exceeding $2^{k-i} - 1$. Finally, we come to an end with some constants

$$S^{(0)}_{m_k,\ldots,m_0},\ m_0, \ldots, m_{k-1} \in \{0, 1\}, \quad m_k = 0, \tag{2.2}$$

defined by P only and from which, in turn, the $S_{\cdots}^{(k-i)}$ $(i = k-1, \ldots, 0)$ and P itself can be computed by means of (2.1).

The effectiveness of this process depends on the method used for generating the τ_i. We use the recurrence relation

$$\tau_i = \tau_{i-1}^2 - 2 \qquad (i = 1, 2, \ldots, k-1), \tag{2.3}$$

where

$$\tau_0(x) = 2x. \tag{2.4}$$

3. Costs in Total. The τ_i $(i = 0, 1, \ldots, k-1)$ are computed with

$$k - 1 \text{ additions}, \quad k \text{ multiplications}$$

in total. As to (2.1), there are 2^i expressions $S_{\cdots}^{(k-i)}$ to be computed at step i with each requiring one addition and one multiplication. Therefore, all steps together require

$$2^k - 1 \text{ additions/multiplications;}$$

and the algorithm computes $P(x)$ with

$$\begin{aligned} n + \log(n+1) - 1 \quad &\text{additions,} \\ n + \log(n+1) \quad &\text{multiplications} \end{aligned} \tag{3.1}$$

in total, $\log n$ denoting the dual logarithm, here and in what follows. Asymptotically, we have n additions/multiplications which is minimum; see Borodin and Munro [1].

4. Parallel Processing. If $(n+1)/2$ processors are available, acting on the same storage, then all $S_{\cdots}^{(k-i)}$ can be computed simultaneously for fixed $i \in \{1, \ldots, k\}$ provided τ_{k-i-1} has been computed by one additional processor in advance. Hence, the algorithm computes $P(x)$ with $(n+3)/2$ processors in $1 + 2\log(n+1)$ time units, one time unit counted for any arithmetical operation.

5. Magnitude of the Intermediate Results. Recall that $n = 2^k - 1 = 2\mu - 1$, $\mu = 2^{k-1}$. Hence,

$$P = \sum_{\nu=0}^{2\mu-1} A_\nu T_\nu,$$

where $T_{\mu+\nu} = 2T_\mu T_\nu - T_{\mu-\nu}$ for $\nu \leqslant \mu$, $\nu, \mu \in \mathbf{N}_0$. From this it follows that

$$P = (2T_\mu)\left\{\frac{A_\mu}{2} + \sum_{\nu=1}^{\mu-1} A_{\mu+\nu} T_\nu\right\} + \left\{A_0 + \sum_{\nu=1}^{\mu-1} (A_\nu - A_{2\mu-\nu}) T_\nu\right\}$$

or, by (2.1),

$$S_{00}^{(k+1)} = A_0 + \sum_{\nu=1}^{\mu-1} (A_\nu - A_{2\mu-\nu}) T_\nu, \quad S_{01}^{(k-1)} = \frac{A_\mu}{2} + \sum_{\nu=1}^{\mu-1} A_{\mu+\nu} T_\nu.$$

On the right-hand sides, each of the A_ν occurs only once. Hence,

$$\|S_{0j}^{(k-1)}\| \leqslant \sum_{\nu=0}^{n} |A_\nu| \qquad (j \in \{0, 1\})$$

if $\|\cdot\|$ denotes the maximum norm with respect to the interval $[-1, 1]$. This estimate seems to be poor, since, for instance, $S_{00}^{(k-1)}$ interpolates P at the zeros of T_μ and could be estimated by the corresponding Lebesgue constant, which is $O(\log n)$; see Ehlich and Zeller [2]. But if we repeat our arguments to $S_{0j}^{(k-1)}$ instead of $S_0^{(k)}$, and so on, we see that

$$\|S_{\cdots}^{(k-i)}\| \leqslant \sum_{\nu=0}^{n} |A_\nu|$$

is generally valid, i.e. for $i = 0, 1, \ldots, k$. Now, define

$$A^{(n)} := \sup\left\{ \sum_{\nu=0}^{n} |A_\nu| \,\middle|\, \left\| \sum_{\nu=0}^{n} A_\nu T_\nu \right\| \leqslant 1 \right\}. \tag{5.2}$$

Then

$$\|S^{(k-i)}\| \leqslant A^{(n)} \|P\|, \tag{5.3}$$

and by Cauchy-Schwarz's inequality we obtain

$$A^{(n)} \leqslant \sqrt{2(n+1)}. \tag{5.4}$$

This estimate cannot be improved essentially, as can be seen as follows. According to Shapiro, there exists a polynomial

$$F(z) = \sum_{\nu=0}^{n} A_\nu z^\nu, \qquad A_\nu \in \{+1, -1\},$$

for any fixed $n \in \mathbf{N}$ such that

$$|F(z)| \leqslant 5\sqrt{n} \quad \text{for } |z| = 1;$$

see Rivlin [5]. Now let

$$P := \sum_{\nu=0}^{n} A_\nu T_\nu.$$

Then, if we perform the cosine-transformation, we see that $\|P\| \leqslant 5\sqrt{n}$. Hence, we have

$$A^{(n)} \geqslant \frac{\sum_{\nu=0}^{n} |A_\nu|}{\|P\|} \geqslant \frac{1}{5}\sqrt{n+1};$$

and the order of the bound in (5.4) cannot be improved. This, however, does not mean that (5.3) is necessarily strict in the same sense.

6. Error Norm. Let $\|P\| \leqslant 1$. We assume that any arithmetical operation is performed with a relative error (due to round-off) not exceeding $\epsilon > 0$ in absolute value (compare Wilkinson [7]), and that the calculation is started with some $\hat{S}_{\cdots}^{(0)}$ approximating the constants $S_{\cdots}^{(0)}$ within the terms of

$$|\hat{S}_{\cdots}^{(0)} - S_{\cdots}^{(0)}| \leqslant |S_{\cdots}^{(0)}|\,\epsilon \leqslant A^{(n)}\epsilon, \tag{6.1}$$

$x \in [-1, 1]$ being assumed to be exact.

The calculation generates some approximations $\hat{\tau}_i$ and $\hat{S}_{\cdots}^{(k-i)}$ for τ_i and $S_{\cdots}^{(k-i)}$, respectively. Note that all these quantities are polynomials with respect to all single round-off errors so that, for $\epsilon \longrightarrow 0$,

$$\hat{\tau}_i = \tau_i + O(\epsilon), \qquad \hat{S}_{\cdots}^{(k-i)} = S_{\cdots}^{(k-i)} + O(\epsilon)$$

for $i = 0, \ldots, k$, where the O-constants can be chosen to be equal for all the finite quantities in question.

First, we are going to estimate $|\hat{\tau}_i - \tau_i|$. Since there are numbers ϵ', ϵ'' such that

$$\hat{\tau}_i = [\hat{\tau}_{i-1}^2(1 + \epsilon') - 2]\,(1 + \epsilon''), \qquad |\epsilon'|, |\epsilon''| \leqslant \epsilon,$$

we obtain

$$\hat{\tau}_i - \tau_i = \hat{\tau}_{i-1}^2 - \tau_{i-1}^2 + \epsilon'\hat{\tau}_{i-1}^2 + \epsilon''(\hat{\tau}_{i-1}^2 - 2) + O(\epsilon^2)$$

from which it follows that

$$|\hat{\tau}_i - \tau_i| \leqslant a|\hat{\tau}_{i-1} - \tau_{i-1}| + b \qquad (i = 1, \ldots, k-1), \tag{6.2}$$

where

$$a = a(\epsilon) = 4 + O(\epsilon), \qquad b = b(\epsilon) = 6\epsilon + O(\epsilon^2). \tag{6.3}$$

Note that $\|\tau_i\| = 2$.

From (6.2) we obtain

$$|\hat{\tau}_i - \tau_i| \leqslant a^i|\hat{\tau}_0 - \tau_0| + b\,\frac{a^i - 1}{a - 1} \qquad (i = 1, \ldots, k-1),$$

where $|\hat{\tau}_0 - \tau_0| \leqslant 2\epsilon$. Together this yields

$$|\hat{\tau}_i - \tau_i| \leqslant 4^{i+1}\epsilon + O(\epsilon^2) \tag{6.4}$$

for $i = 0, 1, \ldots, k-1$.

Next we estimate $|\hat{S}_{\cdots}^{(i)} - S_{\cdots}^{(i)}|$ for $1 \leqslant i \leqslant k$. Since there are numbers ϵ', ϵ'' such that

$$\hat{S}_{\cdots}^{(i)} = [\hat{\tau}_{i-1}\hat{S}_{\cdots 1}^{(i-1)}(1 + \epsilon') + \hat{S}_{\cdots 0}^{(i-1)}](1 + \epsilon''),$$

where $|\epsilon'|, |\epsilon''| \leqslant \epsilon$, we find that

$$\hat{S}_{\cdots}^{(i)} = \tau_{i-1}\hat{S}_{\cdots 1}^{(i-1)} + \hat{S}_{\cdots 0}^{(i-1)} + D_{\cdots}^{(i)}$$

with

$$D_{\cdots}^{(i)} = (\hat{\tau}_{i-1} - \tau_{i-1})\hat{S}_{\cdots 1}^{(i-1)} + (\epsilon' + \epsilon'')\hat{\tau}_{i-1}\hat{S}_{\cdots 1}^{(i-1)} + \epsilon''\hat{S}_{\cdots 0}^{(i-1)} + O(\epsilon^2).$$

Now, using (6.4) and (5.3), we obtain

$$|\hat{S}_{\cdots}^{(i)} - S_{\cdots}^{(i)}| \leqslant 2|\hat{S}_{\cdots 1}^{(i-1)} - S_{\cdots 1}^{(i-1)}| + |\hat{S}_{\cdots 0}^{(i-1)} - S_{\cdots 0}^{(i-1)}| + \epsilon M_{i-1}$$

with

$$M_{i-1} = (4^i + 5)A^{(n)} + O(\epsilon) \qquad (1 \leqslant i \leqslant k).$$

So, if we define $\sigma_i := \max\{|\hat{S}_{\cdots}^{(i)} - S_{\cdots}^{(i)}|: \ldots\}$, then we obtain $\sigma_i \leqslant \epsilon M_{i-1} + 3\sigma_{i-1}$ $(1 \leqslant i \leqslant k)$, where, by (6.1), $\sigma_0 \leqslant A^{(n)}\epsilon$. From this we get the estimate

$$\sigma_i \leqslant 4^{i+1}A^{(n)}\epsilon + O(\epsilon^2)$$

$(i = 1, \ldots, k)$, and with $i = k$, $\hat{P} := \hat{S}_0^{(k)}$, we finally obtain the inequality

$$|\hat{P} - P| \leqslant \sigma_k \leqslant 4^{k+1}A^{(n)}\epsilon + O(\epsilon^2).$$

Hence, if N_n denotes the error norm of the algorithm (see Reimer [4]), we have

$$N_n \leqslant 4(n+1)^2 A^{(n)} \leqslant 4\sqrt{2}\,(n+1)^{5/2}, \tag{6.5}$$

compare (5.4).

THEOREM 1. *The algorithm defined in Section* 2 *is in possession of an* $O(n^{5/2})$ *error-norm.*

We should mention, that the algorithm is normal in the sense of Reimer [4], which implies, that the error-norm has a growth of the order n^2, at least.

7. Calculation of the Constants From the Chebyshev-Fourier Coefficients. The algorithm presented works if the constants (2.2) are known. In order to characterize these constants, define

$$\widetilde{T}_i := \frac{1}{2}\prod_{\nu=0}^{k-1} [2T_{2^\nu}]^{i_\nu} \tag{7.1}$$

for $i = \Sigma_{\nu=0}^{k-1} i_\nu 2^\nu$, $i_\nu\{0, 1\}$ $(\nu = 0, 1, \ldots, k-1)$. Obviously, $\widetilde{T}_i$ is a polynomial of degree exactly i, where $\widetilde{T}_0 = ½\, T_0 = ½$.

Now, let

$$P = \sum_{\nu=0}^{n} A_\nu T_\nu = \sum_{\mu=0}^{n} \widetilde{A}_\mu \widetilde{T}_\mu. \tag{7.2}$$

Then, by (7.1) we have

$$S_{00}^{(k-1)} = \sum_{\nu=0}^{2^{k-1}-1} \widetilde{A}_\nu \widetilde{T}_\nu, \quad S_{01}^{(k-1)} = \sum_{\nu=0}^{2^{k-1}-1} \widetilde{A}_{2^{k-1}+\nu} \widetilde{T}_\nu. \tag{7.3}$$

This means that the coefficients of $P = S_0^{(k)}$ with respect to the $\widetilde{T}_\mu$ occur as the coefficients of $S_{00}^{(k-1)}$ and $S_{01}^{(k-1)}$, respectively, and so on, where

$$S_{m_k,\ldots,m_0}^{(0)} = \frac{\widetilde{A}_m}{2} \quad \text{if } m = \sum_{i=0}^{k} m_i 2^i. \tag{7.4}$$

Recall that $m_k = 0$. By (7.4) we are able to calculate the constants $S_{\cdots}^{(0)}$ from the coefficients A_ν if we can perform the basis transformation from the T_ν to the $\widetilde{T}_\mu$. This problem is dealt with in what follows. If $n \in \mathbf{N}$ is arbitrary, then n has a unique

representation

$$n = \sum_{i=1}^{\kappa(n)} N_i, \qquad N_i \in \{2^j | j \in \mathbf{N}_0\}, \tag{7.5}$$

where $1 \leqslant N_1 < N_2 < \cdots < N_{\kappa(n)}$, $\kappa(n) \leqslant \log n$. Note that

$$\sum_{i=1}^{j-1} N_i < N_j \quad \text{for } j = 1, 2, \ldots, \kappa(n). \tag{7.6}$$

Now, define

$$\mathcal{N}(n) := \left\{ m \,|\, m = \sum_{i=1}^{\kappa(n)} \epsilon_i N_i \in \mathbf{N}, \epsilon_i \in \{+1, -1\} \right\}.$$

Obviously,

$$\mathcal{N}(n) \subset \{1, 2, \ldots, n\}. \tag{7.7}$$

THEOREM 2. *For $n \in \mathbf{N}$ we have*

$$\widetilde{T}_n = \sum_{m \in \mathcal{N}(n)} T_m. \tag{7.8}$$

Proof. The statement is true for all $n < 2^k$ if $k = 1$. The proof is to be performed by induction with respect to k. The statement be true for all $\bar{n} \in \mathbf{N}$, $\bar{n} < 2^{k-1}$, where $k \in \mathbf{N}$, $k > 1$. Let $n < 2^k$ be arbitrary. Without loss of generality we may assume, however, that $2^{k-1} \leqslant n < 2^k$. Then, we have $N_\kappa = 2^{k-1}$ for $\kappa = \kappa(n)$ and $n = N_\kappa + \bar{n}$, $0 \leqslant \bar{n} < 2^{k-1}$. Now, if $\bar{n} = 0$, then $n = 2^{k-1}$ and $\mathcal{N}(n) = \{n\}$, $\widetilde{T}_n = T_n$, and the statement is true.

In what follows, let $1 \leqslant \bar{n} < 2^{k-1}$. Then, by assumption, the statement is true for $\bar{n}$. But

$$\widetilde{T}_n = (2T_{N_\kappa})\widetilde{T}_{\bar{n}} = (2T_{N_\kappa}) \sum_{m \in \mathcal{N}(\bar{n})} T_m.$$

Hence, for $z \in \mathbf{C}$, $|z| = 1$, we obtain

$$\begin{aligned}\widetilde{T}_n\left(\frac{z + z^{-1}}{2}\right) &= (z^{N_\kappa} + z^{-N_\kappa}) \sum_{n \in \mathcal{N}(\bar{n})} \frac{z^m + z^{-m}}{2} \\ &= \sum_{m \in \mathcal{N}(\bar{n})} \left\{ \frac{z^{N_\kappa + m} + z^{-N_\kappa - m}}{2} + \frac{z^{N_\kappa - m} + z^{N_\kappa + m}}{2} \right\}.\end{aligned}$$

Due to (7.5) and (7.6), the exponents $N_\kappa + m$ and $N_\kappa - m$ in this sum, together, exhaust $\mathcal{N}(n)$ exactly once while m is running through $\mathcal{N}(\bar{n})$. This yields

$$\widetilde{T}_n\left(\frac{z + z^{-1}}{2}\right) = \sum_{m \in \mathcal{N}(n)} T_m\left(\frac{z + z^{-1}}{2}\right),$$

and Theorem 2 is proved.

Note always that $\widetilde{T}_0 = \frac{1}{2} T_0$. Now, if we restrict n to the numbers $0 \leqslant n \leqslant 2^k - 1$, then (7.8) defines a basis transformation in the space of all real polynomials of degree not exceeding $2^k - 1$ which can also be characterized by means of the "incidence matrix"

$$E_k := (e_{n,m})_{n,m=1,\dots,2^k-1}$$

defined by

$$e_{n,m} = \begin{cases} 1 & \text{if } m \in N(n) \\ 0 & \text{if } m \notin N(n) \end{cases} \quad (n, m = 1, \dots, 2^k - 1).$$

For, if $T := (T_1, \dots, T_{2^k-1})'$, $\widetilde{T} := (\widetilde{T}_1, \dots, \widetilde{T}_{2^k-1})'$, then

$$\widetilde{T} = E_k T. \tag{7.9}$$

The matrices E_k are in possess of an important recurrence structure which enables us to describe even E_k^{-1} quite easily.

To see that, define for any matrix $A = (a_{ij})_{i,j=0,\dots,q}$, the following associated matrices:

$$|A := (a_{i,q-j})_{i,j=0,\dots,q}, \qquad \bar{A} := (a_{q-i,j})_{i,j=0,\dots,q}.$$

Note, that in case of the unit matrix I we have

$$|I = \bar{I}.$$

THEOREM 3. *We have*

$$E_k = \begin{pmatrix} E_{k-1} & 0 & 0 \\ 0 & 1 & 0 \\ |E_{k-1} & 0 & E_{k-1} \end{pmatrix} \tag{7.10}$$

for $k = 2, 3, \dots$, where E_k is a lower triangular matrix.

Proof. Obviously $m \notin N(n)$ if $m > n$, i.e. E_k is a lower triangular matrix. To prove the symmetry in the lower half of the matrix, let

$$2^{k-1} \leqslant n < 2^k, \quad m \in N(n), \quad m \geqslant 2^{k-1}.$$

Then, because of $N_\kappa = 2^{k-1}$ ($\kappa = \kappa(n)$), we have

$$m = N_\kappa + x, \qquad x = \sum_{i=1}^{\kappa-1} \epsilon_i N_i, \qquad \epsilon_i \in \{+1, -1\},$$

and because of (7.6), $x < N_\kappa$ is valid. Hence, $\bar{m} = N_\kappa - x \in N(n)$; and the symmetry holds as stated. Note that if $m = N_\kappa$, then $x = 0$, and in view of (7.6), we have $\kappa = 1$, $n = N_\kappa = 2^{k-1}$, and Theorem 3 is proved.

We note that $n \in N(n)$ so that the diagonal of E_k is all ones. In particular, we have $E_1 = (1)$ and by Theorem 3 we obtain

$$E_2 = \begin{pmatrix} 1 & & \\ & 1 & \\ 1 & & 1 \end{pmatrix}, \quad E_3 = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ 1 & & 1 & & & & \\ & & & 1 & & & \\ & & 1 & & 1 & & \\ & 1 & & & & 1 & \\ 1 & & 1 & & 1 & & 1 \end{pmatrix}, \dots,$$

where zero-elements are omitted. We are now able to prove

THEOREM 4. *We have*

$$E_k^{-1} = \begin{pmatrix} E_{k-1}^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ -\overline{E}_{k-1}^{-1} & 0 & E_{k-1}^{-1} \end{pmatrix} \tag{7.11}$$

for $k = 2, 3, \ldots,$ *where* E_k^{-1} *is a lower triangular matrix.*

Proof. It is clear that E_k^{-1} is a lower triangular matrix, and because of (7.10) it suffices to prove that

$$-\overline{E_{k-1}^{-1}} \cdot E_{k-1} + E_{k-1}^{-1} \cdot |E_{k-1} = 0.$$

However, it can easily be seen that, for arbitrary $q \times q$ matrices A and B, we have

$$\overline{A} \cdot B = \overline{A \cdot B}, \quad A \cdot |B = |(A \cdot B).$$

Hence, for $A \cdot B = I$ we have

$$-\overline{A} \cdot B + A \cdot |B = -\overline{I} + |I = 0$$

and this completes the proof.

We note that $E_1^{-1} = (1)$ so that we obtain by (7.10)

$$E_2^{-1} = \begin{pmatrix} 1 & & \\ & 1 & \\ -1 & & 1 \end{pmatrix}, \quad E_3^{-1} = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ -1 & & 1 & & & & \\ & & & 1 & & & \\ 1 & & -1 & & 1 & & \\ & -1 & & & & 1 & \\ -1 & & & & -1 & & 1 \end{pmatrix}, \ldots.$$

Obviously, nonzero elements occur in E_k^{-1} and in $|\overline{E}_k'$ exactly at the same places, in E_k^{-1} with alternating signs in each column. Hence, the inverse transformation of (7.9), i.e.

$$T = E_k^{-1}\, \widetilde{T}, \tag{7.12}$$

can be performed quite easily. The same is true, of course, for the corresponding transformation of the coefficients A_ν to the coefficients $\widetilde{A}_\mu$, see (7.2).

Finally, we note that the number of ones occurring in a row of E_k attains its maximum $G_k = 2^{k-1}$ only in the last row, whereas, for $k \geqslant 3$, the number of ones occurring in a column attains its maximum F_k exactly in the i_kth and the j_kth column, where i_k and j_k satisfy the recurrence relation

$$i_k = 2^{k-1} - i_{k-1}, \quad j_k = 2^{k-1} + i_{k-1}$$

$(k = 3, 4, \ldots)$ with $i_1 = i_2 = 1$. From this we obtain

$$i_k = \frac{1}{3}(2^k - (-1)^k) \quad (k = 1, 2, \ldots). \tag{7.13}$$

F_k itself proves to be the kth *Fibonacci-number*, ($F_0 = F_1 = 1, F_k = F_{k-1} + F_{k-2}$ for $k = 2, 3, \ldots$). The first values of i_k, j_k and F_k are listed below:

k	0	1	2	3	4	5	6
i_k		1	1	3	5	11	21
j_k				5	11	21	43
F_k	1	1	2	3	5	8	13

We should remark that, because of the symmetry between E_k and E_k^{-1} with respect to nonzero elements, the maximum number of nonzero elements in a row or in a column of E_k^{-1} is G_k or F_k, respectively. Recall that

$$G_k = 2^{k-1}, \qquad F_k = \frac{1}{\sqrt{5}}\left\{\left(\frac{1+\sqrt{5}}{2}\right)^{k+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{k+1}\right\};$$

hence, both E_k and E_k^{-1} are occupied relatively dense.

8. **Example.** We are going to compare our algorithm with Horner's and with Clenshaw's method. In order to obtain reliable results, we investigate primarily the magnitude of the intermediate results occurring, and this because they are responsible for the actual relative error $e(x)$ in the final numerical result

$$\hat{P}(x) = P(x)\,(1 + e(x)),$$

which, on its part, is random.

Similarly as in [3], we choose the polynomial

$$P(x) = \frac{1}{2} + \sum_{i=1}^{11} \frac{(-1)^{i+1}}{4i^2 - 1} T_{2i}(x)$$

of degree 22 to be the test polynomial. Since this polynomial is chosen because of its virtue that it is approximating the function $\pi|x|/4$ on $[-1, 1]$, we can expect it not to be biased too much with respect to any of our algorithms. Now, due to round-off, we obtain only

$$P(x) \approx \sum_{i=0}^{11} A_{2i} T_{2i} \approx \sum_{i=0}^{11} \tilde{A}_{2i}\, \tilde{T}_{2i} \approx \sum_{i=0}^{11} B_{2i}\, x^{2i},$$

the coefficients being given in Table 1. The three polynomials are evaluated at $x = 0.99580764$ with eight-digit floating-point arithmetic, each by the corresponding algorithm. The intermediate results s_i are listed in Table 3 and Table 4.

TABLE 1

$2i$	A_{2i}	$\frac{1}{2}\cdot\widetilde{A}_{2i}$	B_{2i}
0	0.5	0.5	0.021739131
2	0.33333333	0.15538015	5.7391303
4	−0.066666667	−0.031089963	−82.898547
6	0.028571429	0.010270406	835.61734
8	−0.015873016	−0.0079365080	−5116.0243
10	0.010101010	0.0029991935	19807.225
12	−0.0069930070	−0.0022433707	−50090.998
14	0.0051282051	0.0010161150	83837.848
16	−0.0039215686	−0.0019607843	−92035.326
18	0.0030959752	0.00051279060	63692.266
20	−0.0025062657	−0.0012531329	−25194.614
22	0.0020703934	0.0010351970	4341.9288

For definition of the s_i compare [3]. The final relative error is given in the following table:

TABLE 2

	Horner	Clenshaw	New Algorithm
$\lvert e(x)\rvert$	$2\cdot 10^{-4}$	$4,5\cdot 10^{-8}$	$1,7\cdot 10^{-8}$

We see again that Horner's algorithm generates very big intermediate results which lead to a big relative error $e(x)$, while, in contrary, both of the other algorithms yield only small intermediate results and a very favorable final result. Obviously, our new algorithm seems slightly superior even to Clenshaw's algorithm, as could be expected from the rate of growth of their error-norms.

9. Final Remarks. The order of the error-norm of Clenshaw's algorithm based on the Chebyshev polynomials of the first (second) kind is between n^3 and $n^3\log n$ (n^2 and $n^2\log n$); see Reimer [3]. The corresponding order of our algorithm is between n^2 and $n^{5/2}$ and the algorithm is, therefore, numerically more stable than the first, possibly as stable as the second algorithm of Clenshaw. This fact is confirmed by the example above. By the possibility of parallel processing, our algorithm is preferable if high speed is required. All our reasoning is concerned with the case where the degree is $n = 2^k - 1$. If the degree is between 2^{k-1} and $2^k - 1$, the polynomial can be treated to be a polynomial of degree $2^k - 1$, but then costs and evaluation time seem to be higher, with respect to n, than in (i) and (ii). This, however, is not really true, since certain $S_{\cdots}^{(k-i)}$ vanish and need not be computed, as is shown in our example.

TABLE 3

	Horner	Clenshaw
i	s_i	s_i
22	4341.9288	0.0020703934
21	4323.7259	0.0041234272
20	-20889.015	0.0036356216
19	-20801.441	0.0031173324
18	42978.032	0.0056688805
17	42797.853	0.0081728967
16	-49416.897	0.0066868170
15	-49209.724	0.0051446704
14	34834.429	0.0086875924
13	34688.391	0.012157672
12	-15548.033	0.0085328062
11	-15482.85o	0.0048363954
10	4389.2847	0.011200443
9	4370.8832	0.017470578
8	- 763.46542	0.0077212114
7	- 760.26470	-0.0020928952
6	78.539943	0.016681976
5	78.210675	0.035316974
4	- 5.0157593	- 0.013010817
3	- 4.9947314	- 0.061229516
2	0.76533861	0.22439851
1	0.76213004	0.50814502
0	0.78067405	1.2876309 *

* $P = s_o - xs_1 = 0.78161621$; $x = 0.99580764$

TABLE 4

i	$S^{(o)}_{...j} = \frac{1}{2}\tilde{A}_i$	$S^{(1)}_{...j} = 2 \cdot T_1(x) \cdot S^{(o)}_{...1} + S^{(o)}_{...o}$	$S^{(2)}_{...j} = 2 \cdot T_2(x) \cdot S^{(1)}_{...1} + S^{(1)}_{...o}$	$S^{(3)}_{...j} = 2 \cdot T_4(x) \cdot S^{(2)}_{...1} + S^{(2)}_{...o}$	$S^{(4)}_{...j} = 2 \cdot T_8(x) \cdot S^{(3)}_{...1} + S^{(3)}_{...o}$	$S^{(5)}_{o} = p(x) = 2 \cdot T_{16}(x) \cdot S^{(4)}_{o1} + S^{(4)}_{oo}$
0	$S^{(o)}_{...o} = 0{,}5$	$S^{(1)}_{...o} = 0.5$	$S^{(2)}_{...o} = 0.80555996$			
1	$...1 = 0$					
2	$...o = 0.15538015$	$S^{(1)}_{...1} = 0.15538015$		$S^{(3)}_{ooo} = 0.78522026$		
3	$...1 = 0$					
4	$...o = -0.031089963$	$...o = -0.031089963$	$S^{(2)}_{...1} = -0.010892886$			
5	$...1 = 0$					
6	$...o = 0.010270406$	$...1 = 0.010270406$			$S^{(4)}_{oo} = 0.78150931$	
7	$...1 = 0$					
8	$...o = -0.0079365080$	$...o = -0.0079365080$	$...o = -0.0020384995$			
9	$...1 = 0$					
10	$...o = 0.0029991935$	$...1 = 0.0029991935$		$S^{(3)}_{oo1} = -0.0024962522$		
11	$...1 = 0$					
12	$...o = -0.0022433707$	$...o = -0.0022433707$	$...1 = -0.00024514855$			
13	$...1 = 0$					
14	$...o = 0.0010161150$	$...1 = 0.0010161150$				$S^{(5)}_{o} = 0.78161620$
15	$...1 = 0$					
16	$...o = -0.0019607843$	$...o = -0.0019607843$	$...o = -0.00095236543$			
17	$...1 = 0$					
18	$...o = 0.00051279060$	$...1 = 0.00051279060$		$S^{(3)}_{o1o} = 0.00050896865$		
19	$...1 = 0$					
20	$...o = -0.0012531329$	$...o = -0.0012531329$	$...1 = 0.00078261461$		$S^{(4)}_{o1} = 0.00050896865$	
21	$...1 = 0$					
22	$...o = 0.0010351970$	$...1 = 0.0010351970$		$S^{(3)}_{o11} = 0$		
23	$...1 = 0$					

$x = 0.99580764$

Mathematisches Institut
Universität Dortmund
Postfach 50 05 00
D-4600 Dortmund 50, West Germany

1. A. BORODIN & I. MUNRO, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier, New York, 1975.

2. H. EHLICH & K. ZELLER, "Auswertung der Normen von Interpolationsoperatoren," *Math. Ann.*, v. 164, 1966, pp. 105–112.

3. M. REIMER, "Auswertungsverfahren für Polynome in mehreren Variablen," *Numer. Math.*, v. 23, 1975, pp. 321–336.

4. M. REIMER, "Auswertungsalgorithmen fast-optimaler numerischer Stabilität für Polynome," *Computing*, v. 17, 1977, pp. 289–296.

5. TH. J. RIVLIN, *The Chebyshev Polynomials*, Wiley, New York, 1974.

6. H. S. SHAPIRO, *Extremal Problems for Polynomials and Power Series*, Master's thesis, M.I.T., Cambridge, Mass., 1951.

7. J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1963.