On the Number of Nonscalar Multiplications Necessary to Evaluate Polynomials

Article *in* SIAM Journal on Computing · March 1973

DOI: 10.1137/0202007 · Source: DBLP

Mike Paterson
The University of Warwick

SIAM J. COMPUT.
Vol. 2, No. 1, March 1973

ON THE NUMBER OF NONSCALAR MULTIPLICATIONS NECESSARY TO EVALUATE POLYNOMIALS*

MICHAEL S. PATERSON† AND LARRY J. STOCKMEYER‡

Abstract. We present algorithms which use only $O(\sqrt{n})$ nonscalar multiplications (i.e. multiplications involving "x" on both sides) to evaluate polynomials of degree n, and proofs that at least $\sqrt{n}$ are required. These results have practical application in the evaluation of matrix polynomials with scalar coefficients, since the "matrix × matrix" multiplications are relatively expensive, and also in determining how many multiplications are needed for polynomials with rational coefficients, since multiplications by integers can in principle be replaced by several additions.

Key words. Polynomial evaluation, nonscalar multiplications, rational coefficients, matrix polynomial.

1. Introduction. A well-known result given by Motzkin [2] and Winograd [6] is that, even with preliminary adaptation of the coefficients, at least $n/2$ multiplications are required to evaluate a polynomial of degree n if the coefficients of the polynomial are algebraically independent. However we frequently wish to evaluate polynomials with rational or integer coefficients for which this result does not apply, and so we are led to investigate the number of multiplications required to evaluate rational polynomials. Our main theorem is that $\sqrt{n}$ are necessary, and we present algorithms to demonstrate that $O(\sqrt{n})$ are sufficient.

Apart from providing a satisfactory answer to a theoretical problem our results have some practical applications. Because multiplication by an integer can be replaced by repeated additions, the only multiplications which are counted in the above results are those where the indeterminate of the polynomial appears in both multiplicands, that is the *nonscalar* multiplications. However in some other applications, such as the evaluation of matrix polynomials with scalar coefficients, we are again concerned to minimize the number of nonscalar multiplications because these may be much more expensive than additions and subtractions, or multiplication by a scalar. For practical purposes therefore our most important contributions are the algorithms in § 3, which use only $O(\sqrt{n})$ nonscalar multiplications.

We define an *algorithm* α *over a scalar field* S as $\alpha = \alpha(1), \alpha(2), \cdots, \alpha(k)$, where

(i) $\alpha(1) \in \mathrm{S} \cup \{x\}$ and

(ii) either $\alpha(r) \in \mathrm{S} \cup \{x\}$ or else $\alpha(r) = (\circledcirc, i, j)$, where $1 \leqq i, j < r$, and $\circledcirc \in \{+, -, \times, \div\}$ for $1 < r \leqq k$.

We say that $\alpha(r)$ defines a *nonscalar multiplication/division* if

$\alpha(r) = (\times, i, j)$ and neither $\alpha(i) \in \mathrm{S}$ nor $\alpha(j) \in \mathrm{S}$, or

$\alpha(r) = (\div, i, j)$ and $\alpha(j) \notin \mathrm{S}$.

* Received by the editors December 3, 1971, and in revised form January 23, 1973. This work was done, in part, at Project MAC, an M.I.T. research project sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract N000-70-A-0362-001.

† School of Computer Science, University of Warwick, Coventry, Warwickshire, England.

‡ Project MAC, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

We define the associated elements $\lambda_1, \cdots, \lambda_k \in \mathrm{S}(x)$ as

$$\lambda_r = \begin{cases} \alpha(r) & \text{if } \alpha(r) \in \mathrm{S} \cup \{x\} \\ \lambda_i \circledcirc \lambda_j & \text{if } \alpha(r) = (\circledcirc, i, j) \end{cases} \quad \text{for } 1 \leqq r \leqq k.$$

We say that α computes the polynomial $p(x) \in \mathrm{S}[x]$ if $\lambda_r = p(x)$ for some r, $1 \leqq r \leqq k$.

2. Lower bounds. If a polynomial $p(x)$ can be computed with k nonscalar multiplications/divisions, then the algorithm can be expressed by the following scheme $\mathfrak{A}_k$, where the m_{ij}, $\tilde{m}_{ij}$ are in S and $\circledcirc_r$ is $\times$ or $\div$.

$\underline{\mathfrak{A}_k}$.

$$\mu_{-1} = 1, \qquad \mu_0 = x.$$

For $r = 1, \cdots, k$,

$$\mu_r = \left(\sum_{i=-1}^{r-1} m_{r,i}\mu_i\right) \circledcirc_r \left(\sum_{i=-1}^{r-1} \tilde{m}_{r,i}\mu_i\right).$$

Finally,

$$p(x) = \sum_{i=-1}^{k} m_{0,i}\mu_i.$$

The m_{ij}, $\tilde{m}_{ij}$ will be called the *parameters* of the algorithm. It is useful to think of $\mathfrak{A}_k$ as a parameterization mapping:

$$\mathfrak{A}_k : \mathrm{S}^{M(k)} \to \{p(x) \in \mathrm{S}(x) | p(x) \text{ can be computed in } \leqq k$$
$$\text{nonscalar multiplication/divisions}\},$$

where $M(k)$ = the number of parameters in $\mathfrak{A}_k$.

For the time being we shall consider algorithms without divisions.

THEOREM 1. *For any $n > 2$, there are rational polynomials of degree n which require $\sqrt{n}$ nonscalar multiplications for their evaluation by any algorithm over* **R** *without divisions.*

Proof. Since we are considering algorithms without divisions and since $(m + \Sigma) \times (\tilde{m} + \tilde{\Sigma}) = \Sigma \times \tilde{\Sigma} + m \cdot \tilde{\Sigma} + \tilde{m} \cdot \Sigma + m \cdot \tilde{m}$ we can take $m_{r,-1} = \tilde{m}_{r,-1} = 0$ for $r = 1, \cdots, k$. The constant terms are thus removed from the multiplicands and replaced subsequently by scalar multiplications and additions. A further reduction of $\mathfrak{A}_k$ is easily seen. By suitable scalar multiplication we can arrange that in any multiplicand the first nonzero parameter is 1, and of course we can assume without loss of generality that there is some nonzero parameter.

After these reductions,

$$\mu_1 = x \times x$$

and

$$\mu_2 = (ax^2 + bx) \times (cx^2 + dx).$$

The reader may readily verify that since $\mu_1 = x^2$ has already been computed, μ_2 may as well be in the form $\mu_2 = (m_{21}\mu_1 + m_{20}\mu_0) \times \mu_1$ where $m_{21} = 1$, or $m_{21} = 0$

and $m_{20} = 1$. Counting up the number of parameters in such a reduced form of $\mathfrak{A}_k$ we get, for $k \geqq 2$,

$$\begin{aligned} M(k) &\leqq 1 + (4 + 6 + \cdots + 2(k-1)) + k + 2 \\ &= k^2 + 1. \end{aligned}$$

Suppose $p(x) = q_n x^n + \cdots + q_1 x + q_0$, $q_i \in \mathbf{Q}$, can be evaluated in k nonscalar multiplications by some algorithm $\mathfrak{A}_k^*$, being a reduced form of $\mathfrak{A}_k$. Carry out the operations specified by the algorithm $\mathfrak{A}_k^*$ formally, and view the result as a polynomial in x whose coefficients are integer polynomials in the parameters, that is,

$$p(x) = a_n(\vec{m})x^n + \cdots + a_1(\vec{m})x + a_0(\vec{m}),$$

where $\vec{m}$ denotes the vector of parameters of length $M(k) = k^2 + 1$ and $a_i \in \mathbf{Z}[\vec{m}]$, $i = 0, \cdots, n$. Regarding the $\vec{m}$ as indeterminates, the a_i are in $\mathbf{Q}(\vec{m})$ which has degree of transcendence $M(k)$ over $\mathbf{Q}$. Now suppose $M(k) < n + 1$; then the set $\{a_i | i = 0, \cdots, n\}$ must be algebraically dependent and so satisfy a nontrivial polynomial relation. For further discussion of this see, for example, [5, § 64]. That is, for some nontrivial integer polynomial P^*,

$$P^*(a_n(\vec{m}), \cdots, a_0(\vec{m})) \equiv 0.$$

There are $\frac{1}{3}(k+1)!(k-1)!$ reduced forms $\mathfrak{A}_k^*$ of $\mathfrak{A}_k$. Let $P(a_n, \cdots, a_0) \equiv \prod_* P^*(a_n, \cdots, a_0)$. If all rational polynomials of degree n were computable by the algorithm scheme $\mathfrak{A}_k$, we should have

$$P(\mathbf{Q}^{n+1}) \equiv 0.$$

However, since P is continuous and the rationals are dense this would imply that $P \equiv 0$, contrary to assumption. Therefore,

$$k^2 + 1 \geqq n + 1 \quad \text{or} \quad k \geqq \sqrt{n}.$$

This completes the proof.

Note that we have actually shown that $\{(q_n, \cdots, q_0) \in \mathbf{Q}^{n+1} | q_n x^n + \cdots + q_1 x + q_0$ requires $\sqrt{n}$ nonscalar multiplications$\}$ is a dense subset of $\mathbf{R}^{n+1}$.

The problem of evaluating integer polynomials is related to the problem of evaluating rational polynomials because if $p(x) \in \mathbf{Q}[x]$ can be evaluated in $\leqq k$ nonscalar multiplications by an algorithm over $\mathbf{Q}$ then, for some $z_0 \in \mathbf{Z}$, $z_0 \cdot p(x) \in \mathbf{Z}[x]$ can be evaluated in $\leqq k$ nonscalar multiplications by an algorithm over $\mathbf{Z}$. Hence Theorem 2 (with $\sqrt{n} - 1$ in place of $\sqrt{n} - \frac{1}{2}$) may be deduced as a corollary of Theorem 1, though we know of no useful implication in the opposite direction. We give here a different, combinatorial, proof of Theorem 2.

THEOREM 2. *For any $n > 1$, there are integer polynomials of degree n which require at least $\sqrt{n} - \frac{1}{2}$ nonscalar multiplications for their evaluation by any algorithm over $\mathbf{Z}$ without divisions.*

Proof. Consider the finite ring $\mathbf{F} = \{0, 1\}$ and the ring homomorphism $H : \mathbf{Z} \to \mathbf{F}$ given by

$$H(z) = \begin{cases} 1 & \text{if } z \text{ odd}, \\ 0 & \text{if } z \text{ even}. \end{cases}$$

If $z_n x^n + \cdots + z_1 x + z_0 \in \mathbf{Z}[x]$ can be evaluated by an algorithm over $\mathbf{Z}$ using k nonscalar multiplications, then certainly $w_n x^n + \cdots + w_1 x + w_0 \in \mathbf{F}[x]$, where $w_i = H(z_i)$ for $i = 0, \cdots, n$, can be evaluated by an algorithm over $\mathbf{F}$ using k nonscalar multiplications.

As in the proof of Theorem 1, we can assume that algorithms over $\mathbf{F}$ without divisions can be expressed as a certain reduced form of $\mathfrak{A}_k$. Again, we can assume that no constant terms appear in the multiplicands so that the rth multiplication has the form

$$\mu_r = \Sigma \times \tilde{\Sigma},$$

where $\Sigma = \sum_{i=0}^{r-1} m_{r,i}\mu_i$ and $\tilde{\Sigma} = \sum_{i=0}^{r-1} \tilde{m}_{r,i}\mu_i$. Since $\Sigma \times \tilde{\Sigma} = \tilde{\Sigma} \times \Sigma$, and since neither Σ nor $\tilde{\Sigma}$ should be identically zero, there are at most

$$\tfrac{1}{2}(2^r - 1)2^r < 2^{2r-1}$$

effectively distinct and useful choices for the rth multiplication for $r \geqq 2$. Therefore the number of different polynomials in $\mathbf{F}[x]$ computable by algorithms over $\mathbf{F}$ with k nonscalar multiplications and no divisions is less than

$$\left(\prod_{r=2}^{k} 2^{2r-1}\right) 2^{k+2} = 2^{k^2+k+1}.$$

(More careful counting would yield 2^{k^2+k-2} if desired.) Since there are 2^{n+1} polynomials in $\mathbf{F}[x]$ of degree n or less, if $k^2 + k \leqq n$ then some of these cannot be computed using only k multiplications. The result follows.

For algorithms which use divisions as well we obtain a result similar to Theorem 1.

THEOREM 3. *For any n, there are rational polynomials of degree n which require $\sqrt{n} - 2$ nonscalar multiplications/divisions for their evaluation by an algorithm over* $\mathbf{R}$.

Proof. Consider each of the 2^k algorithmic forms we obtain from $\mathfrak{A}_k$ by choosing a multiplication or division at each nonscalar step. Each introduces at most $k^2 + 4k + 2$ parameters. If we suppose that

$$k^2 + 4k + 2 < n + 1$$

then there is a nontrivial polynomial $P_i \in \mathbf{Q}[x_{n+1}]$ for each form, such that if $q_n x^n + \cdots + q_0 \in \mathbf{Q}[x]$ is computed by the ith algorithmic form, then $P_i(q_n, \cdots, q_0) = 0$. If

$$P(q_n, \cdots, q_0) = \prod_{i=1}^{2^k} P_i(q_n, \cdots, q_0)$$

and all nth degree rational polynomials can be evaluated using k nonscalar multiplications/divisions, we have $P(\mathbf{Q}^{n+1}) \equiv 0$. But $P \not\equiv 0$ and P is continuous which gives a contradiction, proving the result.

3. Algorithms.

3.1. A fast algorithm which uses rational preprocessing. Before giving algorithms which use $O(\sqrt{n})$ nonscalar multiplications, we present an algorithm which will be used later but which is also interesting in its own right. The algorithm is designed to work for real polynomials with algebraically independent coefficients and will therefore use at least $n/2$ multiplications.

Motzkin [2], Eve [1] and Pan [3] have exhibited algorithms which use $n/2 + O(1)$ multiplications. However, the preprocessing used by these algorithms involves finding the roots of polynomials of large degree and this may be computationally difficult in itself. Therefore, we are led to investigate how close we can get to $n/2$ multiplications by using algorithms with rational preprocessing. The best known result is given by the following.

THEOREM 4. *Any polynomial of degree n can be evaluated using $n/2 + O(\log n)$ multiplications. Moreover, the scalars used by the algorithm are rational functions of the coefficients of the polynomial.*

Proof.

ALGORITHM A. Assume $n = 2^m - 1$. First compute $x^2, x^4, x^8, \cdots, x^{2^{m-1}}$. This requires $\log_2 n$ multiplications. Let $N(d)$ = number of multiplications used by the algorithm to evaluate a monic polynomial of degree d given that we have $x^2, x^4, \cdots, x^{2^{m-1}}$. $N(1) = 0$ because $x + a_0$ requires no multiplications.

Now $x^{2p-1} + a_{2p-2}x^{2p-2} + \cdots + a_1x + a_0 = (x^p + c)(x^{p-1} + a_{2p-2}x^{p-2} + \cdots + a_{p+1}x + a_p) + x^{p-1} + b_{p-2}x^{p-2} + \cdots + b_1x + b_0$, where $c = a_{p-1} - 1$ and $b_j = a_j - ca_{p+j}$ for $j = 0, \cdots, p - 2$. Therefore $N(2^i - 1) = 2N(2^{i-1} - 1) + 1$ which yields

$$N(2^i - 1) = 2^{i-1} - 1 \quad \text{for } i = 1, \cdots, m.$$

So,

$$N(n) = N(2^{m-1}) = 2^{m-1} - 1 = (n + 1)/2 - 1.$$

Allowing one more multiplication for the monic division, we have

$$N(n) = (n + 1)/2.$$

Total multiplications $= (n + 1)/2 + \log_2 n$ if $n = 2^m - 1$. For general n, we can break the polynomial into pieces of length $2^i - 1$, evaluate them separately and put them back together using the powers $x^2, x^4, \cdots, x^{2[\log_2 n]}$. The putting-back-together can require at most another $\log_2 n$ multiplications for a total of $n/2 + 2\log_2 n$. This completes the proof.

Essentially the same algorithm was discovered independently by Rabin and Winograd and is given in [4].

3.2. Algorithms which use $O(\sqrt{n})$ nonscalar multiplications. We now present two algorithms which use $O(\sqrt{n})$ nonscalar multiplications. These algorithms can in theory be used to evaluate any rational polynomial $p(x)$ by evaluating $z_0 \cdot p(x)$ by an algorithm over $\mathbf{Z}$ for some appropriate $z_0 \in \mathbf{Z}$. As mentioned before, we can perform an integer scalar multiplication by successive additions.

First we describe a technique for producing $O(\sqrt{n})$ nonscalar multiplications algorithms from $O(n)$ algorithms in which we count all multiplications. The idea is to evaluate $x^2, x^3, x^4, \cdots, x^k$ for some k and then view a polynomial of degree $n = km$ in x as a polynomial of degree m in x^k whose coefficients are themselves polynomials of degree $k - 1$. These polynomials of degree $k - 1$ act like scalars because once we have $x^2, x^3, \cdots, x^{k-1}$ we can compute them free of charge. The only complication which arises is that these polynomial "scalars" become polynomials of larger degree when multiplied whereas numerical scalars remain single numbers when multiplied.

As an example of this method applied to Horner's rule, we obtain the following $O(\sqrt{n})$ algorithm.

ALGORITHM B.

$$\begin{aligned} a_{km-1}x^{km-1} &+ \cdots + a_1x + a_0 \\ &= (\cdots((a_{km-1}x^{k-1} + \cdots + a_{k(m-1)})x^k \\ &\quad + a_{k(m-1)-1}x^{k-1} + \cdots + a_{k(m-2)})x^k \\ &\quad\quad \cdot \\ &\quad\quad\quad \cdot \\ &\quad + a_{2k-1}x^{k-1} + \cdots + a_{k+1}x + a_k)x^k \\ &\quad + a_{k-1}x^{k-1} + \cdots + a_1x + a_0. \end{aligned}$$

This requires about $k + m = k + n/k$ nonscalar multiplications. Minimizing $k + n/k$ gives $k = \sqrt{n}$, or $2\sqrt{n}$ nonscalar multiplications. Note that this algorithm uses about n additions and $n - \sqrt{n}$ scalar multiplications.

The best known coefficient of $\sqrt{n}$ is obtained by applying the technique to the $n/2 + O(\log n)$ algorithm of Theorem 4.

THEOREM 5. *Any polynomial of degree n can be evaluated using* $\sqrt{2n} + O(\log n)$ *nonscalar multiplications. Moreover, the scalars used by the algorithm are rational functions of the coefficients of the polynomial.*

Proof.

ALGORITHM C. Assume $n = k(2^{m-1})$.

(i) Compute $x^2, x^3, x^4, \cdots, x^k$ (k multiplications).

(ii) Compute $x^{2k}, x^{4k}, x^{8k}, \cdots, x^{2^{m-1}k}$ ($\log_2(n/k) = m$ multiplications).

Let $N(d)$ = number of nonscalar multiplications used by the algorithm to evaluate a monic polynomial of degree d given that we have the powers computed in (i) and (ii) above. $N(k) = 0$ because scalar multiplication is free.

We split a monic polynomial up in an analogous way to that of Algorithm A, but the derivation is a little more complicated.

Let $p(x)$ be a monic polynomial of degree $k(2p - 1)$ which we express in the form

$$p(x) = q(x) \cdot x^{kp} + r(x),$$

where q is monic,

$$\deg(q) = k(p - 1), \qquad \deg(r) \leqq kp - 1.$$

Formally dividing the polynomial $r(x) - x^{k(p-1)}$ by $q(x)$ we obtain rational polynomials $c(x)$ and $s(x)$ satisfying:

$$r(x) - x^{k(p-1)} = c(x) \cdot q(x) + s(x),$$

$$\deg(c) \leqq k - 1, \qquad \deg(s) \leqq k(p - 1) - 1.$$

Then

$$p(x) \equiv (x^{kp} + c(x)) \cdot q(x) + x^{k(p-1)} + s(x).$$

Hence, taking $p = 2^{i-1}$, where $i \leqq m$, we get

$$N(k(2^i - 1)) = 2N(k(2^{i-1} - 1)) + 1$$

which yields

$$N(k(2^m - 1)) = 2^{m-1} - 1 \approx n/2k.$$

The total number of nonscalar multiplications is

$$n/2k + k + \log_2 (n/k).$$

Minimizing with respect to k gives $k \approx \sqrt{n/2}$, or $\sqrt{2n} + \log_2 \sqrt{2n}$ nonscalar multiplications.

As before, for general n, this algorithm may require an extra $\log \sqrt{2n}$ multiplications giving a final total of

$$\sqrt{2n} + \log_2 n + O(1) \text{ nonscalar multiplications.}$$

This completes the proof.

Note that this algorithm uses about $n + \sqrt{n/2}$ additions and $n - \sqrt{n/2}$ scalar multiplications.

Acknowledgment. We gratefully acknowledge the many helpful discussions with Albert Meyer concerning this work.

REFERENCES

[1] J. EVE, *The evaluation of polynomials*, Numer. Math., 6 (1964), pp. 17–21.

[2] T. S. MOTZKIN, *Evaluation of polynomials and evaluation of rational functions*, Bull. Amer. Math. Soc., 61 (1955), p. 163.

[3] V. YA. PAN, *Methods of computing values of polynomials*, Uspekhi Mat. Nauk, 21 (1966), no. 6; English transl., Russian Math. Surveys., 21 (1966), pp. 105–136.

[4] M. O. RABIN AND S. WINOGRAD, *Fast evaluation of polynomials by rational preparation*, IBM Res. Rep. RC 3645, Yorktown Heights, N.Y., 1971.

[5] B. L. VAN DER WAERDEN, *Modern Algebra*, Frederick Ungar, New York, 1949.

[6] S. WINOGRAD, *On the number of multiplications necessary to compute certain functions*, Comm. Pure Appl. Math., 23 (1970), pp. 165–179.