THE COMPLEXITY OF BASIC COMPLEX OPERATIONS

Helmut Alt
Fachbereich 10/Angewandte Mathematik u. Informatik
Universität des Saarlandes
D-66 Saarbrücken
BRD

and

Jan van Leeuwen
Department of Computer Science
University of Utrecht
P.O. Box 80.012, 3508 TA Utrecht
the Netherlands

Technical Report RUU-CS-79-4

June 1979

Department of Computer Science
University of Utrecht
P.O. Box 80.012, 3508 TA Utrecht
the Netherlands

all correspondence to:

Dr. Jan van Leeuwen
Department of Computer Science
University of Utrecht
P.O. Box 80.012
3508 TA Utrecht
the Netherlands

THE COMPLEXITY OF BASIC COMPLEX OPERATIONS*

Helmut Alt** and Jan van Leeuwen***

Abstract. It is wellknown that the product of two complex numbers X+iY and U+iV requires exactly 3 real m/d. We study the algebraic complexity of several other operations from the repertoire of complex arithmetic. We show, for instance, that complex inversion requires exactly 4 m/d and that general complex division requires at least 5 m/d. For the latter problem an upper-bound of 6 m/d is known, leaving some speculation as to its precise complexity. The proofs illustrate several criteria which may be of more general use in assessing the complexity of concrete sets of rational functions.

1. Introduction

Given a set of formulae $\{\phi_1,\ldots,\phi_n\}$, algebraic complexity theory concerns itself with the question to assess good (or even perfect) lower- and upper-bounds on the number of operations needed to evaluate the joint formulae. In this paper we shall investigate the number of real multiplications and divisions (m/d for short) required to compute the following basic operations from the repertoire of complex arithmetic, assuming that each complex number involved is initially given by means of its real and imaginary part:

a) $(X+iY)(U+iV)$

b) $\dfrac{1}{U+iV}$

c) $\dfrac{X}{U+iV}$

d) $\dfrac{X+iY}{U+iV}$

---

* A preliminary version of this paper will appear as an extended abstract [3] in the conference record of the 2nd Int. Conference on Fundamentals of Computation Theory, Berlin/Wendisch-Rietz (DDR), Sept. 17-21, 1979.

** Author's address: Fachbereich 10/Angewandte Mathematik u. Informatik, Universität des Saarlandes, D-66 Saarbrücken, BRD.

***Author's address: Department of Computer Science, University of Utrecht, P.O. Box 80.012, 3508 TA Utrecht, the Netherlands.

It is wellknown that a) requires exactly 3 m/d. It was first proved by Munro [7] and Winograd [13] and has become a common example in several textbooks (see e.g. Aho, Hopcroft and Ullman [2]). Strictly speaking only the need for 3 multiplications is usually proved, the case in which divisions are allowed as well requires a more careful argument.

It is usually left unobserved that one can save operations over the straightforward algorithm even when evaluating b), c) and d). An algorithm of Smith [8] dating back to 1962 already demonstrates that complex divisions can be performed using only 6 m/d. The following expressions clearly indicate how b), c) and d) can be evaluated using only 4, 4 and 6 m/d respectively:

b) $$\frac{1}{U+iV} = \frac{U}{U^2+V^2} - i\,\frac{V}{U^2+V^2} = \frac{1}{U+V.V/U} - i\,\frac{V/U}{U+V.V/U}$$

c) $$\frac{X}{U+iV} = \frac{XU}{U^2+V^2} - i\,\frac{XV}{U^2+V^2} = \frac{X}{U+V.V/U} - i\,\frac{X}{U+V.V/U}\,.\,V/U$$

d) $$\frac{X+iY}{U+iV} = \frac{XU+YV}{U^2+V^2} - i\,\frac{XV-YU}{U^2+V^2} = \frac{X+Y.V/U}{U+V.V/U} - i\,\frac{X.V/U-Y}{U+V.V/U}$$

Having formulated a) through d) as the evaluation of particular pairs of rational functions in X, Y, U and V, it is natural to ask if we can do any better than the number of m/d each of the given expressions suggests. Whereas there are scores of useful criteria known for multiplicative complexity, there are only a few which work when divisions are allowed and which give sufficiently accurate estimates in this case. We shall prove the following:

a) requires exactly 3 m/d (by a simple proof),
b) requires exactly 4 m/d,
c) requires exactly 4 m/d too,
d) requires at least 5 m/d.

With an upperbound of 6, our result for d) leaves room for some speculation about the true algebraic complexity of complex division.

While the results are of interest by themselves and further at least our knowledge of the "complexity" of complex arithmetic, we should emphasize that our study aims primarily at understanding some of the mathematical complications involved when proving tight lowerbounds for concrete rational formulae. Hence we shall pay some attention to the particular techniques we employ in the proofs, as their usefulness may extend beyond the scope of the present paper.

2. Straight-line computations

It is important that a precise concept of "computation" is agreed upon, in order that we can meaningfully define the notion of algebraic complexity. We shall briefly review the framework commonly adopted for this purpose (see e.g. Aho, Hopcroft and Ullman [1] or Borodin and Munro [4]).

Let F be a field of char 0 and $\{x_1,\ldots,x_k\}$ a set of independent (and commuting) indeterminates. A straight-line program π is any finite sequence of instructions

$$s_i \leftarrow l_{i1}\ op_i\ l_{i2} \quad (i = 1,\ldots,r)$$

in which for each $1 \le i \le r$

$l_{i1}(l_{i2})$ is either a scalar (from F), an indeterminate, or an s_j for some $1 \le j < i$

and

op_i is any one from a (usually bounded) set of permissible operations.

We shall always choose operations from $\{+,-,*,/\}$. Each s_i can be identified with the rational function in $\{x_1,\ldots,x_k\}$ it "computes". A straight-line program π is said to compute a set of formulae $\{\phi_1,\ldots,\phi_n\}$ if each ϕ_j figures as the associated function of at least one step of π.

It is common practice to contract additions, subtractions, scalar multiplications and divisions by scalars, such that straight-line programs can be redefined as finite sequences

$$s_i \leftarrow l_{i1}\ op_i\ l_{i2} \quad (i = 1,\ldots,r)$$

in which for some $r' \le r$ and all $1 \le i \le r'$

$l_{i1}(l_{i2})$ is some F-linear combination of x_1 to x_k and s_1 to s_{i-1},

and

op_i is in $\{*,/\}$, with both operands non-scalar when $op_i = *$ and at least the denumerator non-scalar when $op_i = /$,

and the last r-r' steps are merely additive or scalar. We shall only "count" the first r' steps and ignore the remaining.

Definition. The algebraic complexity of a set of formulae $\{\phi_1,\ldots,\phi_n\}$ is the smallest number of counted steps a straight-line program computing the joint formulae can have.

Few techniques are known to determine lowerbounds on the (algebraic) complexity of given sets of rational functions. The only general criterion we know is due to Strassen [9], other useful observations are made in e.g. Kung [6] and Strassen [10]. We shall try to lift some of the considerations known from polynomial complexity to the present domain of rational function complexity. We shall first give a careful extension of the known technique of elimination of indeterminates.

Let us call x_j an "essential" constituent of $\{\phi_1,\ldots,\phi_n\}$ if there is a $1 \le i \le n$ such that no $\alpha \in F$ can make $\phi_i - \alpha x_j$ become fully independent of x_j. It means that x_j is essential just when in any straight-line program for $\{\phi_1,\ldots,\phi_n\}$, x_j must occur in some nontrivial (i.e. nonscalar) multiplication or division.

Lemma 2.1. Suppose it takes r m/d to compute $\{\phi_1,\ldots,\phi_n\}$. Let x_j be an essential constituent. Then there exists a rational function R in $\{x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_k\}$ such that $\{\phi_1',\ldots,\phi_n'\}$ can be computed in $\le$ r-1 m/d, where for $1 \le i \le n$ ϕ_i' is obtained from ϕ_i by substituting $x_j := R$.

Proof

Consider an arbitrary, contracted straight-line program π computing $\{\phi_1,\ldots,\phi_n\}$ using precisely r m/d. Look for the first counted step

$$s_i \leftarrow l_{i1}\ op_i\ l_{i2}$$

in which x_j occurs. There are two cases to consider.

Case (i): $op_i = *$.

Clearly one of the operands (at least) must be of the form $\alpha x_j + R'$, for some non-zero $\alpha \in F$ and rational function R' only depending on $\{x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_k\}$. Substituting $x_j := \frac{\varepsilon - R'}{\alpha}$ will trivialize the step into a scalar multiplication by ε, which can subsequently be absorbed to obtain a new contracted straight-line program π' using one multiplication less than π ... provided we were careful enough to choose ε such that no later divisions did become singular. Since F is infinite, there is an ample supply of ε's that will do.

Case (ii): $op_i = /$.

This time we must examine both l_{i1} and l_{i2}, and conclude that s_i must be of the form

$$s_i \leftarrow (\alpha x_j + R')/(\beta x_j + R'').$$

where at least one of $\alpha,\beta \in F$ is nonzero and R',R" again are rational functions in indeterminates $\neq x_j$. Substituting

$$x_j := \frac{1}{\varepsilon\beta-\alpha} R' - \frac{\varepsilon}{\varepsilon\beta-\alpha} R''$$

will trivialize s_i into a scalar and again eliminate a counted step from π, provided we choose ε carefully to keep later divisions from becoming singular. Such ε can again be found.

□

A second and extremely useful technique can be to show that all (non-trivial) linear combinations of ϕ_1 to ϕ_n must remain "hard". The following lemma is intimately related to the various "rank theorems" known in the litterature (see e.g. Winograd [12], van Leeuwen and van Emde Boas [11], Fiduccia and Zalcstein [5], also Borodin and Munro [4]), but has apparently never been explicitly stated as we do here. Its proof was already shown in Alt and van Leeuwen [2], but is included again for completeness.

Lemma 2.2. Suppose it takes r m/d to compute $\{\phi_1,\ldots,\phi_n\}$, for some $r \geq n-1$. Then there exists a nontrivial linear combination $\sum_1^n \alpha_i\phi_i$ which can be computed in $\leq r-n+1$ m/d.

Proof

Let $s_1,\ldots,s_r$ be the "counted" steps of some contracted straight-line program π computing $\{\phi_1,\ldots,\phi_n\}$ in r m/d. Considering the definition of computation, there must exist a scalar matrix M and a vector $v = [v_1 \ldots v_n]^T$ of linear (hence zero-cost) correction terms such that

$$\begin{bmatrix} \phi_1 \\ \vdots \\ \phi_n \end{bmatrix} = \begin{pmatrix} \cdots & & \cdots \\ & M & \\ \cdots & & \cdots \end{pmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_r \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

The matrix M' consisting of the last n-1 columns of M must be singular (because it has more rows than columns), hence there is some nontrivial linear combination of the rows of M such that the last n-1 entries of the resulting row are all zero. It means that for some nontrivial set of α_i's

$$\sum_1^n \alpha_i\phi_i = \overbrace{[\ldots\ldots}^{r-n+1} \vdots 00\ldots00\ldots0] \begin{bmatrix} s_1 \\ \vdots \\ s_r \end{bmatrix} + \sum_1^n \alpha_i v_i$$

Hence $\sum_1^n \alpha_i \phi_i$ can be computed by means of a (new) straight-line program that only needs to perform the counted steps of π up to s_{r-n+1}!

□

As we can sometimes reason more easily about single forms than we can about sets, lemma 2.2 can be extremely versatile. Alt and van Leeuwen [2] have indicated how the lemma leads to a completely elementary proof of the fact that general products in n-dimensional algebras A always need $\geq 2n-1$ multiplications, provided that A contains no zero-divisors (a result originally proved by more sophisticated means in Fiduccia and Zalcstein [5], van Leeuwen and van Emde Boas [11] and Winograd [14]). We shall use it to prove the one result about complex arithmetic known until now (Winograd [13]), this time in an entirely elementary manner. Assume F is a real field.

Theorem 2.3. The product of X+iY and U+iV requires 3 m/d.

Proof

Recall that the task consists of evaluating the following formulae

$$XU - YV$$
$$XV + YU$$

It is wellknown that 3 m/d suffice.

Suppose that 2 m/d would do. By lemma 2.2 there must exist α_1 and α_2 (not both zero) such that

$$\alpha_1(XU-YV) + \alpha_2(XV+YU)$$

could be evaluated in $\leq 2-2+1=1$ m/d. The one counted step can impossibly be a division, hence must be a multiplication. Rewriting the form as

$$[\alpha_1 X + \alpha_2 Y \quad \alpha_2 X - \alpha_1 Y] \begin{bmatrix} U \\ V \end{bmatrix}$$

it is clear that Winograd's column-rank criterion ([12], see also [1], [4]) applies, forcing us to conclude that

$$1 \geq \#\text{multiplic} \geq \text{column-rank}[\alpha_1 X + \alpha_2 Y \quad \alpha_2 X - \alpha_1 Y] = 2.$$

Contradiction!

□

3. Complexity of division and other basic operations

We prolong our assumption that F be a real field throughout this section.

The proof that complex product requires 3 m/d made use of the paradigm suggested before: an overly optimistic upperbound assumption allowed us to contrive a single form whose complexity would not exceed 1 m/d, but had to.

If we try a similar argument for other sets of formulae, then it is unlikely that we can reduce it down to a case similar in simplicity. We shall attempt to go one level higher and characterize the formulae computable using at most 2 m/d.

Let $l_1, l_2, \ldots$ be generic names for scalars and linear functions in $\{x_1, \ldots, x_k\}$.

Lemma 3.1. Each rational function computed by a straight-line program using ≤ 2 m/d can be written as $P/l_1(l_2 l_3 + l_4)$, for some polynomial P.

Proof

The result is trivially true for straight-line computations using ≤ 1 m/d. Consider an arbitrary, contracted straight-line program π using precisely 2 m/d. Let its counted steps be s_1 and s_2 (in order). Only a limited number of different cases can occur.

Case (i): $op_1 = *$, $op_2 = *$.

If this happens, only polynomial expressions can be extracted from π.

Case (ii): $op_1 = /$, $op_2 = *$.

We observe that s_1 and s_2 must have the following form. Note that s_2 may make use of the quotient computed in s_1.

$$s_1 = \frac{l_1}{l_2},$$

$$s_2 = (\alpha\frac{l_1}{l_2} + l_3)(\beta\frac{l_1}{l_2} + l_4) = \frac{P'}{l_2 \cdot l_2},$$

for some appropriate polynomial P'. The forms we can possibly derive from these steps without using further m/d, must always be of type

$$\gamma\frac{l_1}{l_2} + \delta\frac{P'}{l_2 l_2} + l_5$$

and (hence) can be written as $P/l_2 \cdot l_2$ for some polynomial P. This certainly satisfies our lemma.

Case (iii): $op_1 = *$, $op_2 = /$.

Now s_1 and s_2 must be of the following form. Note again that s_2 can use the result of s_1 or add in new, linear "stuff".

$$s_1 = l_1 l_2,$$

$$s_2 = \frac{\alpha l_1 l_2 + l_3}{\beta l_1 l_2 + l_4}$$

Any scalar combination of s_1, s_2 and the indeterminates must be of the form $P/\beta l_1 l_2 + l_4$. Again the statement of the lemma is satisfied.

Case (iv): $op_1 = /$, $op_2 = /$.

This time computed forms can get a bit more complicated. We can still represent s_1 as

$$s_1 = \frac{l_1}{l_2}$$

The result of s_2 can be anything of the form

$$s_2 = \frac{\alpha\frac{l_1}{l_2} + l_3}{\beta\frac{l_1}{l_2} + l_4} = \frac{\alpha l_1 + l_2 l_3}{\beta l_1 + l_2 l_4}$$

The expressions we can now obtain as a result must be of the form

$$\gamma\frac{l_1}{l_2} + \delta\frac{\alpha l_1 + l_2 l_3}{\beta l_1 + l_2 l_4} + l_5$$

and (hence) can be written as $P/l_2(\beta l_1 + l_2 l_4)$ for some polynomial P. All expressions of this sort fit the most general case allowed for in the statement of our lemma.

□

The proof of lemma 3.1 shows not every expression of type $l_1(l_2 l_3 + l_4)$ will occur as a denumerator, but we do not need a more precise assertion. The case-analysis could be extended to straight-line programs which use more m/d, although the "type" of expressions becomes increasingly unmanageable. (Hence one should switch to a degree-argument. Compare Kung [6].)

The given characterization can be rephrased as follows. In the formulation we rely on familiarity with the concept of polynomial divisibility.

Lemma 3.2. Let P, Q_1 and Q_2 be polynomials such that $P/Q_1 Q_2$ can be computed using ≤ 2 m/d. If Q_1 has no divisor of the form $l_1(l_2 l_3 + l_4)$, then $Q_1 | P$.

Proof

If P/Q_1Q_2 can be computed using ≤ 2 m/d, then lemma 3.1 learns that there exists a polynomial P_1 such that

$$\frac{P}{Q_1Q_2} = \frac{P_1}{l_1(l_2l_3 + l_4)}$$

. Rewrite this to

$$P = \frac{Q_1 . P_1 Q_2}{l_1(l_2l_3 + l_4)}$$

and read it as saying that the right-hand side of the equality must be purely polynomial. As Q_1 cannot (by assumption) absorb the denumerator or even any of its nonscalar factors (!), we conclude that

$$R_1 = \frac{P_1 Q_2}{l_1(l_2l_3 + l_4)}$$

must be polynomial. Since $P = Q_1 . R_1$, it means that Q_1 must divide P.

□

The lemma gives an exact account of one's intuition that rational functions computable in ≤ 2 m/d cannot have too complicated factors in their denumerator, unless they can be divided out of the expression.

We now have all tools ready to prove sharp lowerbounds on the complexity of the remaining operations from the basic repertoire of complex arithmetic. We shall treat complex inversion first, as it is simple and gives an immediate example of the use of lemma 3.2 as stated.

Theorem 3.3. The computation of $1/U+iV$ requires exactly 4 m/d.

Proof

We recall that the task consists of evaluating the following two formulae

$$\frac{U}{U^2+V^2}$$

$$\frac{V}{U^2+V^2}$$

We saw that 4 m/d suffice.

Suppose that 3 m/d would do. By lemma 2.2 there must be α_1 and α_2 (not both zero) such that

$$\alpha_1 \frac{U}{U^2+V^2} + \alpha_2 \frac{V}{U^2+V^2}$$

i.e.,

$$\frac{\alpha_1 U + \alpha_2 V}{U^2 + V^2}$$

can be computed using $\leq 3-2+1 = 2$ m/d. Since U^2+V^2 has no divisors of the form $l_1(l_2l_3+l_4)$, lemma 3.2 shows this can only be when

$$U^2 + V^2 \mid \alpha_1 U + \alpha_2 V$$

, an impossibility. Contradiction.

□

Division of a real by a complex number is also computable in 4 m/d and can impossibly be easier than inversion. Hence

Theorem 3.4. The computation of $X/U+iV$ requires exactly 4 m/d.

The hardest proof concerns our lowerbound on the complexity of complex division. It requires an intricate combination of all three of the techniques we have developed.

Theorem 3.5. The computation of $X+iY/U+iV$ requires at least 5 m/d.

Proof

The task consists of evaluating the following two forms

$$\frac{XU + YV}{U^2 + V^2}$$

$$\frac{XV - YU}{U^2 + V^2}$$

Observe that X, Y, U and V are all essential constituents.

Suppose there was a straight-line program π evaluating the forms in ≤ 4 m/d. Look for the first, counted step containing X or Y. Without loss of generality we may assume the step contains an X. By the same argument as in lemma 2.1 there must exist a rational function R in {U,V} and scalar α, such that the substitution

$$X := \alpha Y + R$$

eliminates the step and results in a set of formulae computed using 1 m/d less. The choice of α and R can be made such that no later division becomes singular. We conclude that the forms

$$\frac{\{\alpha Y+R\}U + YV}{U^2 + V^2}$$

$$\frac{\{\alpha Y+R\}V - YU}{U^2+V^2}$$

must be computable in ≤ 3 m/d.

By lemma 2.2 there must exist β_1 and β_2 (not both zero) such that

$$\beta_1 \frac{\{\alpha Y+R\}U + YV}{U^2+V^2} + \beta_2 \frac{\{\alpha Y+R\}V - YU}{U^2+V^2}$$

can be computed in $\leq 3-2+1 = 2$ m/d. Let $R = {}^P/_Q$, with P and Q relatively prime. The degenerated case R=0 will be accomodated for by choosing P=0 and Q=1. Rearrange the formula into

$$\frac{P\{\beta_1 U+\beta_2 V\} + Y.Q.\{(\alpha\beta_1-\beta_2)U + (\alpha\beta_2+\beta_1)V\}}{(U^2+V^2).Q}$$

By lemma 3.2 (and noting the algebraic nature of U^2+V^2 again) this expression can be evaluated in ≤ 2 m/d only if

$$U^2+V^2 \mid P\{\beta_1 U+\beta_2 V\} + Y.Q\{(\alpha\beta_1-\beta_2)U + (\alpha\beta_2+\beta_1)V\}$$

Observing that P, Q are both polynomials in U and V, this can only be when

$$U^2+V^2 \mid P.\{\beta_1 U+\beta_2 V\} \qquad (*)$$

$$U^2+V^2 \mid Q.\{(\alpha\beta_1-\beta_2)U + (\alpha\beta_2+\beta_1)V\} \qquad (**)$$

If P=0, then Q=1 and (**) can be satisfied only when

$$(\alpha\beta_1-\beta_2)U + (\alpha\beta_2+\beta_1)V = 0$$

. It would mean that

$$\begin{bmatrix}\alpha\beta_1-\beta_2\\ \alpha\beta_2+\beta_1\end{bmatrix} = \begin{pmatrix}\beta_1 & -\beta_2\\ \beta_2 & \beta_1\end{pmatrix}\begin{bmatrix}\alpha\\ 1\end{bmatrix} = 0$$

while the matrix involved is nonsingular (F is real!). Contradiction.

If $P\neq 0$, then (*) can be satisfied only when

$$U^2+V^2 \mid P$$

Likewise (**) can be satisfied only when

$$U^2+V^2 \mid Q$$

, as we have just seen that the linear factor in the expression on the right can never be zero (which would have been the only possibility to preclude that $U^2+V^2 \mid Q$). It follows that P and Q must both contain U^2+V^2 as a factor,

a clear violation of their relative primeness. Contradiction.

This shows that there exists no straight-line program computing complex division in fewer than 5 m/d.

□

4. Conclusion

We have investigated the algebraic complexity of several sets of rational functions, as they present themselves in a number of basic operations from complex arithmetic. The results are summarized in the following chart

operation	lowerbound	upperbound
complex product	3 m/d	3 m/d
complex inversion	4 m/d	4 m/d
complex division	5 m/d	6 m/d

We see our intuition confirmed that complex division is "harder" than complex product. It is intriguing that even the inversion of a single complex number is more complex, algebraically speaking, than the product of two. No definite answer has yet been obtained to the question of what the precise complexity of complex division is. We conjecture that the lowerbound of 5 can be improved and (hence) that 6 m/d are optimal.

5. References

[1] Aho, A.V., J. Hopcroft and J.D. Ullman, The design and analysis of computer algorithms, Addison-Wesley Publ. Co., Reading, Mass., 1974.

[2] Alt, H. and J. van Leeuwen, A classroom note on computing products in finite-dimensional algebras, EATCS Bulletin, number 8, June 1979, pp. 14-17.

[3] Alt, H. and J. van Leeuwen, The complexity of complex division (extended abstract), in: Conference record of the 2nd Int. Conference on Fundamentals of Computation Theory, Berlin/Wendisch-Rietz (DDR), Sept. 17-21, 1979 (to appear).

[4] Borodin, A. and I. Munro, The computational complexity of algebraic and numeric problems, Theory of Comput. Series, Vol. 1, American Elsevier Publ. Co., New York, NY., 1975.

[5] Fiduccia, C.M. and Y. Zalcstein, Algebras having linear multiplicative complexities, J.ACM 24 (1977) 311-331.

[6] Kung, H.T., The computational complexity of algebraic numbers, Proc. 5th Annual ACM Symp. on Theory of Computing, 1973, pp. 334-342.

[7] Munro, I., Some results concerning efficient and optimal algorithms, Proc. 3rd Annual ACM Symp. on Theory of Computing, 1971, pp. 40-44.

[8] Smith, R.L., Algorithm 116: complex division, CACM 5 (1962) 435.

[9] Strassen, V., Evaluation of rational functions, in: R.E. Miller and J.W. Thatcher (eds.), Complexity of computer computations, Plenum Press, New York, NY., 1972, pp. 1-10.

[10] Strassen, V., Vermeidung von Divisionen, J. für die Reine u. Angew. Mathematik 264 (1973) 184-202.

[11] van Leeuwen, J. and P. van Emde Boas, Some elementary proofs of lower bounds in complexity theory, Linear Alg. and its Appl. 19 (1978) 63-80.

[12] Winograd, S., On the number of multiplications necessary to compute certain functions, Comm. Pure and Appl. Math. 23 (1970) 165-179.

[13] Winograd, S., On the multiplication of 2x2 matrices, Linear Alg. and its Appl. 4 (1971) 381-388.

[14] Winograd, S., The effect of the field of constants on the number of multiplications, Conf. Record 16th Annual IEEE Symp. on Foundations of Computer Science, Berkeley, 1975, pp. 1-2.