Strassen's Algorithm is not Optimal
Trilinear Technique of Aggregating, Uniting and Canceling for Constructing Fast Algorithms for Matrix Operations

V. Ya. Pan

Mathematical Sciences Department
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

Abstract: A new technique of trilinear operations of aggregating, uniting and canceling is introduced and applied to constructing fast linear non-commutative algorithms for matrix multiplication. The result is an asymptotic improvement of Strassen's famous algorithms for matrix operations.

Key words: fast algorithms, complexity of computation, arithmetic complexity, linear algebraic problems, matrix multiplication, bilinear forms, trilinear form.

1. Introduction.

Probably the most exciting result in algebraic complexity theory was obtained by V. Strassen in 1968 (see [21]). He discovered that matrix multiplication (MM), matrix inversion (MI), evaluation of determinant (ED) and solving linear system of equations (SLS) can be done by $O(N^\alpha)$ arithmetic operations (where N is the size of the problem that is the order of the square matrices involved, and $\alpha = \log_2 7 \approx 2.807$), rather than by $O(N^3)$ operations required in classical methods. Strassen's algorithms reduce these 4 problems to a problem of constructing a fast linear algorithm for multiplying two 2×2 matrices. It seemed surprising that fast algorithms for all these (and for some other important problems like the transitive closure problem in graph theory, see e.g. [1], of any large order could be immediately constructed if a fast linear algorithm of a certain type (we will use the notation LA for such algorithms) for multiplying two matrices of a specific order was given. Even a small improvement of such a linear algorithm for matrix multiplications (e.g., reducing the complexity of LA in comparison with Strassen's algorithm by even 1 for any size $N = 2^k$) would automatically result in asymptotic improvement of the algorithms for the above-mentioned problems MM, MI, ED, SLS, etc. The attempts to find such an improvement of linear algorithms for matrix multiplication were numerous before and, particularly after the publication of Strassen's paper. Despite the several bright ideas suggested and despite the progress in understanding the problem (see [3-12,15,18,20,22-28], for surveys, see [2-8] and also Remarks 1-3 and Section 10 in the present paper), no algorithms were constructed, such that they would give an asymptotic improvement of Strassen's method. In this paper a new technique for transformations LA from a trivial one with complexity n^3 to fast ones is presented. The transformations are the chains of elementary ones which will be called trilinear operations. Each LA can be written as a chain of these elementary operations of 2-4 kinds. Such a representation makes the ways of constructing fast LA more comprehensive. Trilinear operations of uniting terms reduce the complexity of LA. Thus the main objective will be in increasing the number of unitings in the chains. Though the exploration of this technique has just been started (not counting a short period in 1972, see [18]), its power has already been demonstrated in this paper. LA which are asymptotically faster than Strassen's are described in Sections 7-8. By using these algorithms and the mentioned reduction of the other problems to constructing LA, all the problems MM, MI, ED, SLS, etc. of the size N can be solved by $O(N^{2.795})$, rather than by $O(N^{2.807})$, arithmetic operations.

It is well known (see [18,22]) that any LA can be written in bilinear, as in the trilinear version and both are equivalent. However, in this paper, all the new fast LA are presented in the trilinear version which seems more appropriate for them. The bilinear version and some auxiliary techniques are exposed in the next section. The trilinear version and an example of a fast LA from [18] are described in Section 3. In Section 4 the

CH1397-9/78/0000-0166$00.75 © 1978 IEEE

technique of trilinear aggregating and uniting is introduced and applied to constructing fast LA. LA from [18] is analyzed and generalized as a model. Fast LA which for several *n* give non-asymptotic improvement of all previously published LA are presented in Section 4 (also see Table 9 in Section 10). It is not clear, however, if a similar procedure of trilinear aggregating and uniting can also produce an asymptotic improvement of Strassen's algorithm. Yet this is done by combining such a procedure with trilinear canceling in Sections 5-8. An example of the trilinear canceling is described in Section 6. In Section 9, the results of this paper are listed and illustrated by tables. Sections 10 and 11 contain open problems and acknowledgments. The reader interested only in an asymptotically fast LA can find it in Section 8. No preliminary knowledge is required to obtain the *Main Theorem* from Section 8, except a well-known theorem from [21] (see Theorem 1 in the next section) and the fact about equivalency of the bilinear and trilinear representations of LA. The latter fact is also well-known and can be easily obtained by comparing formulas (1) in Section 2 and (2) in Section 3.

2. Some Notation, Definitions and Auxiliary Techniques

Let integers *n* and M be given. Let A,B,C denote n×n matrices, a_{ij}, b_{ij}, c_{ij} denote their entrees, such that

$$A = \| a_{ij} \|,\ B = \| b_{ij} \|,\ C = \| c_{ij} \| .$$

Let L(A),L(B),L(C) denote linear forms of the entrees of the matrices A,B,C. Let M triplets of linear forms $L^1_q(A), L^2_q(B), L^3_q(C)$ be given, such that

$$L^1_q(A) = \sum_{i,j=0}^{n-1} \alpha^q_{ij} a_{ij},$$

$$L^2_q(B) = \sum_{i,j=0}^{n-1} \beta^q_{ij} b_{ij},$$

$$L^3_q(C) = \sum_{i,j=0}^{n-1} \gamma^q_{ij} c_{ij},\ q = 1,.,M.$$

Then let for any pair of matrices A,B and for any pair of integers (k,l), such that $0 \le k,l \le n-1$ the following system of identities hold:

$$\left.\begin{aligned} &p_q = L^1_q(A)\ L^2_q(B),\ q=1,2,...,M. \\ &\sum_{s=0}^{n-1} a_{ks} b_{sl} = \sum_{q=1}^{M} \gamma^q_{kl} p_q = \sum_{q=1}^{M} \gamma^q_{kl}\ L^1_q(A)\ L^2_q(B) \\ &k,l = 0,1,...,n-1. \end{aligned}\right\} (1)$$

Then if the entrees of the matrices A and B are given, (1) describes an algorithm for computing C = AB which is called a linear non-commutative algorithm for multiplying two n×n matrices A and B. *n* is called a size of the problem, M is called a complexity of the algorithm. We will use a notation LA for the latter.

Remark 1. The definitions and the technique presented in this paper can be easily generalized for the problem of multiplying n×p by p×m matrices with any n,p,m and in many cases for the problem of the evaluation of a set of bilinear forms or (which is the same) of a trilinear form (see [5,18,22]). Note that the size of the problem MM is always determined by a triplet (m,n,p) of integers. It is easy to observe (see [18] or [12]) that the complexity of the optimal LA for a problem MM is invariant to all 6 possible permutations in a triplet (m,n,p), e.g., substituting the problem of multiplying m×n by n×p matrices for the problem of multiplying p×m by m×n matrices. Thus the described LA = LA(n) can be turned into LA(m,n,p) for non-square matrix multiplications.

Strassen's fast algorithms for MM, MI, ED, SLS are based on the two following theorems:

Theorem 1. (V. Strassen [21]).

Let a positive integer *n* and a linear, non-commutative algorithm LA for multiplying two n×n matrices be given such that its complexity is equal to M. Then algorithms for solving the problems MM, MI, ED, SLS by only $O(N^{\log_n M})$ arithmetic operations can be constructed for any N. Here N is the order of square matrices involved in the problems MM, MI, ED, SLS.

Remark 2. The inverse theorems expressing the lower bounds on the complexity of the problems MM and MI through the lower bounds on the complexity of LA also hold (see the theorems for MM without divisions in [26] and in the general case in [18,22]).

Theorem 2. (V. Strassen [21]).
There exists a linear non-commutative algorithm LA for multiplying two 2×2 matrices whose complexity is equal to 7.

Remark 3. The bound 7 on M = M(2) for the size n = 2 is sharp since M(2) ≤ 7 always, see [21], and M(2) ≥ 7 always, see [10,11]. Moreover, LA for the size n = 2 whose complexity M(2) is equal to 7 is unique to within a linear transformation (see [18] or [12]). A further asymptotic speed-up could be achieved by constructing fast LA for n = 3 such that M = M(3) ≤ 21. Yet this problem turned out to be very difficult (if solvable). Thus the most promising way (as it seemed, at least to the present author) consisted in constructing fast LA for greater size of n×n matrices.

3. Linear Algorithms for Matrix Multiplication as a Representation of a Given Trilinear Form

The evaluation of a set of bilinear forms and of a trilinear form are 2 equivalent problems [5,18,22]. In particular, the evaluation of the product of n×p by p×m matrices and of the track of the product of 3 matrices (n×p one by p×m one by m×n one) are 2 equivalent problems [18]. Here is the equivalency in the case n = p = m.

$$\sum_{i,j,k=0}^{n-1} a_{ij}\, b_{jk}\, c_{ki} = \sum_{q=1}^{M(n)} L_q^1(A)\ L_q^2(B)\ L_q^3(C). \qquad (2)$$

It is easy to verify that (1) and (2) are equivalent. However, some LA can be better expressed in (2) than in (1). Consider the following example from [18].

Notation. For the sake of simplicity in sequel, n is always even and positive, n = 2s,s ≥ 1, and all sub-indices of a, b and c are always considered modulo n, that is, $f_{l+n,m+n} = f_{l,m}$ where f stands for a, b or c.

Algorithm 1.

$$\sum_{i+j+k \text{ is even}} (a_{ij} + a_{k+1,i+1})(b_{jk} + b_{i+1,j+1})(c_{ki} + c_{j+1,k+1}) -$$

$$- \sum_{i,k=0}^{n-1} a_{k+1,i+1} \sum_{j:i+j+k \text{ is even}} (b_{jk} + b_{i+1,j+1})c_{ki} -$$

$$- \sum_{i,j=0}^{n-1} a_{ij}\, b_{i+1,j+1} \sum_{k:i+j+k \text{ is even}} (c_{ki} + c_{j+1,k+1}) -$$

$$- \sum_{k,j=0}^{n-1} \sum_{i:i+j+k \text{ is even}} (a_{ij} + a_{k+1,i+1})b_{jk}\ c_{j+1,k+1} \equiv$$

$$\underset{(a_{ij}, b_{gh}, c_{qi})}{\equiv} \sum_{i,j,k} a_{ij}\, b_{jk}\, c_{ki}.$$

It is easy to verify (see the next section) that for any n algorithm 1 is LA whose complexity is equal to $\frac{n^3}{2} + 3n^2$

Definition Any product of three linear forms is a term. If a trilinear form T is written as a sum of M terms, then this gives a representation R = R(T) of a given form T whose complexity (norm) ‖R‖ is equal to M = M(R). In this case R is said to consist of M terms, include exactly M terms and have a complexity (a norm) M = ‖R‖. Each LA is a representation of a given trilinear form $\sum_{i,j,k} a_{ij}\, b_{jk}\, c_{ki}$ as a sum of M terms, where M is a complexity of LA, M = ‖LA‖.

Remark 4. It is obvious that (unlike the rank of the tensor of a trilinear form see [22]) a norm ‖R(T)‖ is not determined just by a given trilinear form T. Strassen's LA and algorithm 1 give non-trivial examples.

Definition: LA determined by the trivial representation that is by the identity

$$\sum_{i,j,k} a_{ij}\, b_{jk}\, c_{ki} = \sum_{i,j,k} a_{ij}\, b_{jk}\, c_{ki}$$

is called trivial and denoted LA(0). ‖LA(0)‖ = n³.

4. The Aggregating and Uniting of Terms

In this section we will generalize the construction of algorithm 1 based on the following simple identities:

$$\left.\begin{aligned}
&a_{ij} b_{jk} c_{ki} + a_{k_1 i_1} b_{i_1 j_1} c_{j_1 k_1} = \\
&= (a_{ij} + a_{k_1 i_1})(b_{jk} + b_{i_1 j_1})(c_{ki} + c_{j_1 k_1}) - \\
&- a_{k_1 i_1} (b_{i_1 j_1} + b_{jk}) c_{ki} - \\
&- a_{ij} b_{i_1 j_1} (c_{j_1 k_1} + c_{ki}) - \\
&- (a_{k_1 i_1} + a_{ij}) b_{jk} c_{j_1 k_1}.
\end{aligned}\right\} \quad (3)$$

$$\left.\begin{aligned}
&\sum_{i,j,k} a_{k_1 i_1} (b_{i_1 j_1} + b_{jk}) c_{ki} = \\
&= \sum_{k,i} a_{k_1 i_1} c_{ki} \sum_{j} (b_{i_1 j_1} + b_{jk}). \\
&\sum_{i,j,k} a_{ij} b_{i_1 j_1} (c_{j_1 k_1} + c_{ki}) = \\
&= \sum_{i,j} a_{ij} b_{i_1 j_1} \sum_{k} (c_{j_1 k_1} + c_{ki}). \\
&\sum_{i,j,k} (a_{k_1 i_1} + a_{ij}) b_{jk} c_{j_1 k_1} = \\
&= \sum_{j,k} b_{jk} c_{j_1 k_1} \sum_{i} (a_{k_1 i_1} + a_{ij}).
\end{aligned}\right\} \quad (4)$$

Here $i_1 = i_1(i), j_1 = j_1(j), k_1 = k_1(k)$ are considered given functions of i, of j, or of k, e.g., $i_1 = i + 1 j_1 = j + 1, k_1 = k + 1$ for algorithm 1.

Let T(ijk) denote the term a_{ij} b_{jk} c_{ki} of LA(0), T^0(ijk), $-T^m$(ijk), m = 1,2,3 denote 4 terms in the right part of (3).

In a sense, (3) describes aggregating a pair of terms (see formal definitions of trilinear aggregating and uniting in [19]) of LA(0), that is, substituting $T^0(ijk) - \sum_{m=1}^{3} T^m(ijk)$ for the sum $T(ijk) + T(i_1j_1k_1)$. Let $\frac{n^3}{2}$ pairs of terms of LA(0) be aggregated by applying (3) and all terms in the left parts of these (3) be all different. Then we obtain LA by summing all left parts and all right parts of these identities such that $\| LA \| = 2n^3$.

The terms $T^1(i_0jk_0)$ for given k_0,i_0 and for j=0,1,...,n-1 are said to be kin terms and to form a family of *n* kin terms. Their sum can be represented as a term by applying (4). Similarly for the terms $T^2(i_0j_0k)(i_0,j_0$ given, k=0,1,...,n-1) and $T^3(ij_0k_0)(j_0,k_0$ given, i=0,1,...,n-1). We will also call all the terms T^m(ijk), m=1,2,3; i,j,k=0,1,...,n-1, acceptable since applying (4) we can unite their sum in at most $3n^2$ terms.

As a result we obtain a new LA, such that

$$\| LA \| \le \frac{n^3}{2} + 3n^2.$$

This procedure is formally determined in [19] where also formal definitions of aggregating trilinear terms and, in particular, of uniting kin trilinear terms are given and a lower bound $\frac{n^3}{2} + \frac{9}{4} n^2$ on $\| LA \|$ for LA resulting in this procedure is established. Here is an optimal version of this procedure such that $\| LA \| = \frac{n^3}{2} + \frac{9}{4} n^2$ for the resulting LA (in sequel we will show how to change the procedure to gain further improvements).

Notation. $|S|$ is the cardinality of a given set *S*; S(1), S(2),S(3) are 3 sets: of all integers *i*, of all pairs of integers (ij) and of all triplets of integers (ijk), such that in all 3 cases each integer is modulo *n*. (S_1,S_2) and (S_1,S_2,S_3) are the cartesian products of sets S_1,S_2 and S_1,S_2,S_3. $E \subset S(1)$ and $O \subset S(1)$ are 2 subsets of S(1) consisting of all even and of all odd integers modulo *n*. P_2^1 is $S(2) \setminus (O,O) = (E,E) \cup (E,O) \cup (O,E)$, $\tilde{P}_1(fg)$ is S(1), if $(fg) \in (E,E)$, and it is E, if $(fg) \in S(2) \setminus (E,E) = (O,O) \cup (E,O) \cup (O,E)$. $S^1 = S \setminus (O,O,O) = (E,E,E) \cup (E,E,O) \cup (E,O,E) \cup (O,E,E)$, $(k_1,i_1,j_1) = (k+1,i+1,j+1)$ (here S = S(3)).

Now let for each triplet $(ijk)\in S^1$ the terms T(ijk) and $T(k_1 i_1 j_1) = T(k+1,i+1,j+1)$ be aggregated by applying (3), all the left parts and separately all the right parts of these identities (3) for $(ijk)\in S^1$ be summed and all acceptable terms in the right part of the resulting identity be united by applying (4). It is easy to verify that this gives LA whose complexity is $\frac{n^3}{2}+\frac{9}{4}n^2$ (optimal within the class of all LA obtained from LA(0) by this procedure). Here is a formal presentation of this LA.

Algorithm 2

$$T^0 = \sum_{(i,j,k)\in S^1} (a_{ij} + a_{k+1,i+1}) \times$$

$$\times (b_{jk} + b_{i+1,j+1})(c_{ki} + c_{j+1,k+1}),$$

$$T^1 = \sum_{(k,i)\in P_2^1} a_{k+1,i+1} \sum_{j\in \tilde{P}_1(ki)} (b_{jk} + b_{i+1,j+1})c_{ki},$$

$$T^2 = \sum_{(i,j)\in P_2^1} a_{ij}\, b_{i+1,j+1} \sum_{k\in \tilde{P}_1(ij)} (c_{ki} + c_{j+1,k+1}),$$

$$T^3 = \sum_{(j,k)\in P_2^1} \sum_{i\in \tilde{P}_1(jk)} (a_{ij} + a_{k+1,i+1})\, b_{jk}\, c_{j+1,k+1},$$

$$T^0 - T^1 - T^2 - T^3 = \sum_{i,j,k=0}^{n-1} a_{ij}\, b_{jk}\, c_{ki}\,.$$

Exercise 1. Let $(k_1,i_1,j_1) = m(i,j,k) = (k+s,i+s,j+s)$, where n = 2s. Let S^1 be equal to the set of all triplets of integers modulo n such that at least 2 integers in each triplet are less than s. Repeat the described procedure to construct another LA of the complexity $\frac{n^3}{2}+\frac{9}{4}n^2$. What are $P_2^1, \tilde{P}_1(fg)$?

We should change or modify the procedure of constructing LA from LA(0) to reduce ‖LA‖ further.

Here is an example of such a modification resulting in a slight improvement of algorithm 2.

Let all the terms of LA(0), but the terms T(i,i+1,i+2), T(i+1,i+2,i), T(i+2,i,i+1),i=0,1,...,n−1, be aggregated by applying (3) and then be summed. In other words, let $R^0 = R^0(T^0)$ be a representation of a trilinear form T^0 as the sum $\sum_{(ijk)\in\tilde{S}} T^0(ijk)$ where $\tilde{S} = S^1\setminus\tilde{S}^0, \tilde{S}^0$ consists of all the triplets (i,i+1,i+2), (i+1,i+2,i), (i+2,i,i+1) where $i \in E$. Then $\|R^0\| = n^3 - \frac{3n}{2}$, rather than $\frac{n^3}{2}$, but 3n terms T(ijk) are missed in the sum of the left parts of (3). Yet this is fixed by a special uniting procedure (different from (4)). Here is this modified version 2a of Algorithm 2 whose complexity is equal to $n^3 - \frac{3n}{2} + \frac{9}{4}n^2$ for any even n.

Algorithm 2a

$$T^0 = \sum_{(ijk)\in\tilde{\tilde{S}}} (a_{ij} + a_{k+1,i+1})\,(b_{jk} + b_{i+1,j+1})\,(c_{ki} + c_{j+1,k+1}).$$

$$T^1 = \sum_{(ki)\in P_2^1} a_{k+1,i+1} \left[\sum_{j\in\tilde{P}_1(k,i)} (b_{jk} + b_{i+1,j+1}) - \right.$$

$$\left. - \delta_{i,k+1}\, b_{i+1,k} \right] c_{ki}$$

$$T^2 = \sum_{(ij)\in P_2^1} a_{ij}\, b_{i+1,j+1} \left[\sum_{k\in\tilde{P}_1(i,j)} (c_{ki} + c_{j+1,k+1}) - \right.$$

$$\left. - \delta_{j,i+1}\, c_{j+1,i} \right]$$

$$T^3 = \sum_{(jk)\in P_2^1} \left[\sum_{i\in\tilde{P}_1(j,k)} (a_{ij} + a_{k+1,i+1}) - \right.$$

$$\left. - \delta_{k,j+1}\, a_{k+1,j} \right] b_{jk}\, c_{j+1,k+1}$$

$$T^0 - T^1 - T^2 - T^3 = \sum_{i,j,k=0}^{n-1} a_{ij}\, b_{jk}\, c_{ki}$$

$$\delta_{lm} = \begin{cases} 0 & if\, l \neq m \\ 1 & if\, l = m \end{cases}$$

Exercise 2. Reduce in $\frac{3n}{2}$ the complexity of LA constructed in Exercise 1 by excluding $\frac{3n}{2}$ elements from S^1.

5. A procedure of aggregating the triplets of terms of the trivial LA with subsequent uniting the kin terms.

Here is a generalization of (3), (4). To the end of this section the reader may assume t=1 and even drop all superindices in the formulas (5)-(7).

$$\left.\begin{aligned}
& a^{t0}_{ij} b^{t0}_{jk} c^{t0}_{ki} + a^{t1}_{jk} b^{t1}_{ki} c^{t1}_{ij} + a^{t2}_{ki} b^{t2}_{ij} c^{t2}_{jk} = \\
&= \left(a^{t0}_{ij} + a^{t1}_{jk} + a^{t2}_{ki}\right)\left(b^{t0}_{jk} + b^{t1}_{ki} + b^{t2}_{ij}\right)\left(c^{t0}_{ki} + c^{t1}_{ij} + c^{t2}_{jk}\right) \\
&- a^{t0}_{ij} b^{t2}_{ij}\left(c^{t0}_{ki} + c^{t1}_{ij} + c^{t2}_{jk}\right) - a^{t1}_{jk} b^{t0}_{jk}\left(c^{t1}_{ij} + c^{t2}_{jk} + c^{t0}_{ki}\right) \\
&- a^{t2}_{ki} b^{t1}_{ki}\left(c^{t2}_{jk} + c^{t0}_{ki} + c^{t1}_{ij}\right) - a^{t0}_{ij}\left(b^{t0}_{jk} + b^{t1}_{ki}\right) c^{t1}_{ij} - \\
&- a^{t1}_{jk}\left(b^{t1}_{ki} + b^{t2}_{ij}\right) c^{t2}_{jk} - a^{t2}_{ki}\left(b^{t2}_{ij} + b^{t0}_{jk}\right) c^{t0}_{ki} - \\
&- \left(a^{t2}_{ki} + a^{t0}_{ij}\right) b^{t0}_{jk} c^{t2}_{jk} - \left(a^{t0}_{ij} + a^{t1}_{jk}\right) b^{t1}_{ki} c^{t0}_{ki} - \\
&- \left(a^{t1}_{jk} + a^{t2}_{ki}\right) b^{t2}_{ij} c^{t1}_{ij} - a^{t0}_{ij} b^{t1}_{ki} c^{t2}_{jk} - a^{t1}_{jk} b^{t2}_{ij} c^{t0}_{ki} - \\
&\qquad - a^{t2}_{ki} b^{t0}_{jk} c^{t1}_{ij}.
\end{aligned}\right\} \quad (5)$$

Each 4-tuple (t, i, j, k) such that $1 \le t \le 8$, $(ijk) \in S(3)$ (in this section t=1) determines an identity (5). (5) aggregates 3 terms of its left part which will be called desirable ones. The first term in the right part of (5) will be called aggregated one, 9 next terms will be called acceptable ones, and 3 remaining terms will be called inacceptable ones.

$$\left.\begin{aligned}
& \sum_{i,j,k} a^{t0}_{ij} b^{t2}_{ij}\left(c^{t0}_{ki} + c^{t1}_{ij} + c^{t2}_{jk}\right) = \\
&\qquad = \sum_{i,j} a^{t0}_{ij} b^{t2}_{ij} \sum_{k}\left(c^{t0}_{ki} + c^{t1}_{ij} + c^{t2}_{jk}\right). \\
& \sum_{i,j,k} a^{t1}_{jk} b^{t0}_{jk}\left(c^{t1}_{ij} + c^{t2}_{jk} + c^{t0}_{ki}\right) = \\
&\qquad = \sum_{j,k} a^{t1}_{jk} b^{t0}_{jk} \sum_{i}\left(c^{t1}_{ij} + c^{t2}_{jk} + c^{t0}_{ki}\right). \\
& \sum_{i,j,k} a^{t2}_{ki} b^{t1}_{ki}\left(c^{t2}_{jk} + c^{t0}_{ki} + c^{t1}_{ij}\right) = \\
&\qquad = \sum_{k,i} a^{t2}_{ki} b^{t1}_{ki} \sum_{j}\left(c^{t2}_{jk} + c^{t0}_{ki} + c^{t1}_{ij}\right)
\end{aligned}\right\} \quad (6)$$

$$\left.\begin{aligned}
& \sum_{i,j,k} a^{t0}_{ij}\left(b^{t0}_{jk} + b^{t1}_{ki}\right) c^{t1}_{ij} = \sum_{i,j} a^{t0}_{ij} c^{t1}_{ij} \sum_{k}\left(b^{t0}_{jk} + b^{t1}_{ki}\right) \\
& \sum_{i,j,k} a^{t1}_{jk}\left(b^{t1}_{ki} + b^{t2}_{ij}\right) c^{t2}_{jk} = \sum_{j,k} a^{t1}_{jk} c^{t2}_{jk} \sum_{i}\left(b^{t1}_{ki} + b^{t2}_{ij}\right) \\
& \sum_{i,j,k} a^{t2}_{ki}\left(b^{t2}_{ij} + b^{t0}_{jk}\right) c^{t0}_{ki} = \sum_{k,i} a^{t2}_{ki} c^{t0}_{ki} \sum_{j}\left(b^{t2}_{ij} + b^{t0}_{jk}\right) \\
& \sum_{i,j,k}\left(a^{t2}_{ki} + a^{t0}_{ij}\right) b^{t0}_{jk} c^{t2}_{jk} = \sum_{j,k} b^{t0}_{jk} c^{t2}_{jk} \sum_{i}\left(a^{t2}_{ki} + a^{t0}_{ij}\right) \\
& \sum_{i,j,k}\left(a^{t0}_{ij} + a^{t1}_{jk}\right) b^{t1}_{ki} c^{t0}_{ki} = \sum_{k,i} b^{t1}_{ki} c^{t0}_{ki} \sum_{j}\left(a^{t0}_{ij} + a^{t1}_{jk}\right) \\
& \sum_{i,j,k}\left(a^{t1}_{jk} + a^{t2}_{ki}\right) b^{t2}_{ij} c^{t1}_{ij} = \sum_{i,j} b^{t2}_{ij} c^{t1}_{ij} \sum_{k}\left(a^{t1}_{jk} + a^{t2}_{ki}\right)
\end{aligned}\right\} \quad (7)$$

Identities (6) and (7) can be applied to unite the kin terms in the sum of right parts of identities (5). Here (5) are determined by a given t and by serveral triplets (ijk) so that there are many kin terms among the acceptable terms in the sum of the right parts of these (5).

Let S^0 denote a subset $\{(i,i,i)\}$ of S(3), p^* denote the permutation $p^*(i,j,k) = (j,k,i)$. Similarly the previous section, a procedure of constructing LA from LA(0) could be applied. It is assumed that $a^{tm}_{pq} = a_{pq}$, $b^{tm}_{pq} = b_{pq}$, $c^{tm}_{pq} \underset{\sim}{=} c_{pq}$ for (p, q) $\in S(2)$, m=0, 1, 2, in this section. Let a subset $S = S(3) \backslash S^0$ of S(3) be partitioned into 3 subsets S^1, S^2, S^3, such that

$$p^*(S^1) = S^2,\ p^*(S^2) = S^3;\ p^*(S^3) = S^1; |S^1| =$$

$$= |S^2| = |S^3| = \frac{|\tilde{S}|}{3} = \frac{n^3-n}{3}. \qquad (8)$$

Now let us sum all the left and all the right parts of those identities (5) which are determined by t=1 and by $(ijk) \in S^1$. Then the left part of the resulting identity equals

$$\sum_{(ijk)\in S(3)\backslash S^0} a_{ij} b_{jk} c_{ki} = \sum_{(ijk)\in S(3)} a_{ij} b_{jk} c_{ki} - \\ - \sum_{(ijk)\in S^0} a_{ij} b_{jk} c_{ki}$$

and the right part of the resulting identity is a sum of $\frac{13}{3} \times (n^3\text{-}n)$ terms.

Notation. Let (5) be determined by t, $1 \le t \le 8$ and by $(ijk) \in S(3)$. Then $\nu+1$-th term of the right part of (5) is denoted by $\tilde{T}^{0t}_{ijk}$, if $\nu=0$, and by-$\tilde{T}^{\nu t}_{ijk}$, if ν = 1, 2, ..., 12. $R^{0t} = R^{0t}(T^{0t})$ is the representation of $T^{0t} = \sum_{(ijk)\in S^1} \tilde{T}^{0t}_{ijk}$ as the sum of $\tilde{T}^{0t}_{ijk}$; $||R^{0t}|| = \frac{n^3-n}{3}$; $R^{\mu t} = R^{\mu t}(T^{\mu t})$ are the representations of

$T^{\mu t} = \sum_{r=0}^{2} \sum_{(i,j,k)\in S^1} \tilde{T}^{3\mu-r,t}_{ijk}$ as the sums of $\tilde{T}^{\mu t}_{ijk}$; $||R^{\mu t}|| = n^3\text{-}n$, μ = 1, 2, 3, 4. Applying identities (6)-(7) for uniting the kin terms of T^{1t}, T^{2t}, T^{3t}, we can easily obtain new representations $\tilde{R}^{\mu t} = \tilde{R}^{\mu t}(T^{\mu t})$, such that $||\tilde{R}^{\mu t}|| = n^2$, μ = 1, 2, 3. Thus only $||R^{4t}||$ is to be reduced for constructing fast LA, since the complexity of $\sum_{i} a_{ii} b_{ii} c_{ii}$ is just n. Moreover, we can add the sum $\sum_i a_{ii} b_{ii} c_{ii}$ to both parts of the resulting identity, and then each term $a_{ii} b_{ii} c_{ii}$ can be united with one of the terms $\tilde{T}^{1t}(i, i, \tilde{i}(i))$, $\tilde{T}^{2t}(\tilde{i}(i), i, i)$ or $\tilde{T}^{3t}(i, \tilde{i}(i), i)$, where $\tilde{i}(i) \neq i$, by applying one of the following identities:

$$\left.\begin{array}{ll} a_{ii} b_{ii} c_{ii} - a_{ii} b_{ii} (c_{\tilde{i} i} + c_{ii} + c_{i\tilde{i}}) = & - a_{ii} b_{ii} (c_{\tilde{i} i} + c_{i\tilde{i}}). \\ a_{ii} b_{ii} c_{ii} - a_{ii} (b_{\tilde{i} i} + b_{ii} + b_{i\tilde{i}}) c_{ii} = & - a_{ii} (b_{\tilde{i} i} + b_{i\tilde{i}}) c_{ii} \\ a_{ii} b_{ii} c_{ii} - (a_{\tilde{i} i} + a_{ii} + a_{i\tilde{i}}) b_{ii} c_{ii} = & - (a_{\tilde{i} i} + a_{i\tilde{i}}) b_{ii} c_{ii}. \end{array}\right\} (9)$$

This gives a LA whose complexity is equal to $\frac{n^3-n}{3}$ + $3n^2$+ + $||R_4||$. Unfortunately, it is not clear, how under the assumptions of this section the terms of T_4 can be united. However, they will be cancelled in the next section.

6. Trilinear cancelling by assigning coefficients and indices

The terms of R^{4t} will be cancelled by using a new assignment of values for a^{tl}_{mr}, b^{tl}_{pq} c^{tl}_{gh}, where each (m, r))p, q), (g, h) belongs to S(2) and stands for either (ij), or (jk), or (ki) depending on ℓ, ℓ=0, 1, 2.

Notation. $\bar{i} = i+s$, $\bar{j} = j+s$, $\bar{k} = k+s$. S(1, s) = {i, 0≤i≤s-1} ⊂S(1), S(2, s) = {(i, j), 0≤i, j≤s-1} ⊂S(2), S(3, s) = {i, j, k), 0≤i, j, k≤s-1} ⊂S(3). $S^0(s)$ = {(i, i, i), 0≤i≤s-1), $\tilde{S}(s) = S(3, s) \backslash S^0(s)$.

We will assume that $a^{tv}_{mr} = \alpha^{tv}_m a_{m^*r^*}$, $b^{tv}_{pq} = \beta^{tv}_p b_{p^*q^*}$, $c^{tv}_{gh} = \gamma^{tv}_g c_{g^*h^*}$, where $m^* - m$=0 (mod s), $r-r^*$=0 (mod s), $p-p^*$=0 (mod s), $q-q^*$=0 (mod s), $g-g^*$=0 (mod s), $h-h^*$=0 (mod s), each of α^{tv}_m, β^{tv}_q, β^{tv}_q, γ^{tv}_g is ±1. Now m^*, r^*, p^*, q^*, g^*, h^*, α^{tv}_m, β^{tv}_p, γ^{tv}_g and thus a^{tv}_{mr} b^{tv}_{pq}, c^{ty}_{gh} for (m, r) ∊ S(2), (p, q) ∊ S(2), (g, h) ∊ S(2), t=1, 2, 3, ..., 8, v=0, 1, 2, will be determined by the following 8 tables.

Table 1

v	0	1	2
a	ij	jk	ki
b	jk	ki	ij
c	ki	ij	jk

Table 2

v	0	1	2
a	ij	$\bar{j}k$	$k\bar{i}$
b	$j\bar{k}$	ki	$\bar{i}j$
c	$\bar{k}i$	$i\bar{j}$	jk

Table 3

v	0	1	2
a	$\bar{i}\bar{j}$	$j\bar{k}$	ki
b	jk	$\bar{k}i$	$i\bar{j}$
c	$k\bar{i}$	ij	$\bar{j}k$

Table 4

v	0	1	2
a	$\bar{i}j$	jk	$\bar{k}i$
b	$\bar{j}k$	$k\bar{i}$	ij
c	ki	$\bar{i}j$	$j\bar{k}$

Table 8

v	0	1	2
a	$\bar{i}\bar{j}$	$\bar{j}\bar{k}$	$\bar{k}\bar{i}$
b	$\bar{j}\bar{k}$	$\bar{k}\bar{i}$	$\bar{i}\bar{j}$
c	$\bar{k}\bar{i}$	$\bar{i}\bar{j}$	$\bar{j}\bar{k}$

Table 7

v	0	1	2
a	$\bar{i}\bar{j}$	$j\bar{k}$	$\bar{k}i$
b	$\bar{j}k$	$\bar{k}\bar{i}$	$i\bar{j}$
c	$k\bar{i}$	$\bar{i}j$	$\bar{j}\bar{k}$

Table 6

v	0	1	2
a	$i\bar{j}$	$\bar{j}k$	$\bar{k}\bar{i}$
b	$\bar{j}\bar{k}$	$k\bar{i}$	$\bar{i}j$
c	$\bar{k}i$	$\bar{i}\bar{j}$	$j\bar{k}$

Table 5

v	0	1	2
a	$\bar{i}j$	$\bar{j}\bar{k}$	$k\bar{i}$
b	$j\bar{k}$	$\bar{k}i$	$\bar{i}\bar{j}$
c	$\bar{k}\bar{i}$	$i\bar{j}$	$\bar{j}k$

m^*, r^* are determined from the square (a, v) in the intersection of the row a and the column v of table t, and $a^{vt}_{mr} = a_{m^*r^*}$, if there is a circle in the square (a, v); $a^{vt}_{mr} = -a_{m^*r^*}$ otherwise. Similarly, p^*, q^*, b^{vt}_{pq} are determined by the square (b, v) and g^*, h^*, c^{vt}_{gh} are determined by the square (c, v) of table t, 1≤t≤8. The system of circles in tables 1-8 is determined as the following one. All the squares in tables 1, 8 are circled. The other circled square are (1, 2), (2, 3), (3, 1) in tables 2 and 7, (11), (22), 33) in tables 3 and 6, (13), (21), (32) in tables 4 and 5. All remaining squares in tables 2-7 are non-circled. Notice that each pair of tables t and 9-t, t = 1, 2, 3, 4. have the same system of circles.

It is easy to verify that the following lemma holds.

Lemma. Let (ijk) ∊ $\tilde{S}(s)$. Then the sum of the left parts of 8 identities (5) determined by this triplet (ijk) and by t=1, 2, 3, ..., 8, is a sum of 24 different products a_{vm} b_{mr} c_{rv} where either (vmr) = (ijk) (mod s), or (vmr) = (jki) (mod s), or (vmr) = (kij) (mod s). The sum of all inacceptable terms of these 8 identities is zero.

7. Asymptotically fast LA

Now its is enough to combine the operations of aggregating, uniting and cancelling for transforming the trivial LA(0) into a fast LA.

Let $\tilde{S}(s)$ be partitioned into subsets $S^1(s)$, $S^2(s)$, $S^3(s)$, such that

$$\left.\begin{array}{l} |S^1(3)| = |S^2(s)| = |S^3(s)| = \frac{|\tilde{S}(s)|}{3} = \frac{s^3-s}{3}. \\ p^*(S^1(s)) = S^2(s),\ p^*(S^2(s)) = S^3(s),\ p^*(S^3(s)) = S^1(s) \end{array}\right\} (10)$$

Note that (10) is similar to (8).

Let all the left and all the right parts of the identities (5) determined by t = 1, 2, 3, ..., 8 and by all triplets (ijk) ∊ $S^1(s)$ be summed. Then let all kin acceptable terms in the right part of the resulting identity be united into $24s^2 = 6n^2$ terms by applying the identities (6)-(7). The sum of all inacceptable terms is equal to zero, since it follows from lemma. Thus the complexity of this representation of the right part of the identity is equal to $8 (\frac{s^3-s}{3} + 3s^2) = \frac{n^3-4n}{3} + 6n^2$. The left part of the identity is the sum of $8(s^3-s) = n^3-4n$ different terms $T(i_r j_r k_r) = a_{i_r j_r}$ $b_{j_r k_r}$ $c_{k_r i_r}$, r = 1, 2, ..., 4n. Let remaining 4n terms T(ijk) be added to the left and to the right parts of the identity. Then a fast LA, whose complexity M = ||LA|| is equal to $\frac{n^3-4n}{3}$ + 6 n^2 + 4n, has been obtained. However, this LA can be slightly improved further. It is easy to verify, that each $T(i_r j_r k_r)$, 1≤r≤4n, can be united with another term of LA, namely, with one, formed by uniting acceptable terms through the application of (6)-(7) (compare with transforming algorithm 2 into algorithm 2a in section 4). Thus for any even n LA is constructed whose complexity is $\frac{n^3-4n}{3} + 6n^2$. The set $\tilde{S}(s)$ can be partitioned into subsets $S^1(s)$, $S^2(3)$, $S^3(s)$ satisfying (10) in many different ways. For instance, we can set:

$$\left.\begin{array}{l} S^1(s) = S^1_1(s) \cup S^1_2(s),\ S^1_1(s) = \{(i,j,k),\ 0 \le i \le j < k \le s-1\}, \\ S^1_2(s) = \{(i,j,k),\ 0 \le k \le j < i \le s-1\},\ S^2(s) = p^*(S^1(s)), \\ S^3(s) = p^*(S^2(s)),\ p^*(ijk) = (jki) \end{array}\right\} (11)$$

Exercise 3. Construct a fast LA, such that $||LA|| = \frac{n^3-4n}{3} + 6n^2$ by using a similar procedure and by substituting $i^* = i+1$, $j^* = j+1$, $k^* = k+1$ for $\bar{i}$, $\bar{j}$, $\bar{k}$, and S^* = {(i,j,k), i, j, k are even} for $\tilde{S}(s)$.

8. Formal description of an asymptotically fast LA.

Algorithm 3.

$$T^0 = \sum_{(i,j,k)\in S^1(s)} [(a_{ij} + a_{jk} + a_{ki})(b_{jk} +$$

$$+ b_{ki} + b_{ij})(c_{ki} + c_{ij} + c_{jk}) -$$

$$-(a_{ij} - a_{\bar{jk}} + a_{\bar{ki}})(b_{\bar{jk}} + b_{ki} - b_{\bar{ij}}) \times$$

$$\times(-c_{\bar{ki}} + c_{\bar{ij}} + c_{jk}) -$$

$$-(-a_{\bar{ij}} + a_{\bar{jk}} + a_{ki})(b_{jk} - b_{\bar{ki}} + b_{\bar{ij}}) \times$$

$$\times(c_{\bar{ki}} + c_{ij} - c_{\bar{jk}}) -$$

$$-(a_{\bar{ij}} + a_{jk} - a_{\bar{ki}})(-b_{\bar{jk}} + b_{\bar{ki}} + b_{ij}) \times$$

$$\times(c_{ki} - c_{\bar{ij}} + c_{\bar{jk}}) -$$

$$-(a_{\bar{ij}} + a_{\bar{\bar{jk}}} - a_{\bar{ki}})(-b_{\bar{jk}} + b_{\bar{ki}} + b_{\bar{\bar{ij}}}) \times$$

$$\times(c_{\bar{\bar{ki}}} - c_{\bar{ij}} + c_{\bar{jk}}) -$$

$$-(-a_{\bar{ij}} + a_{\bar{jk}} + a_{\bar{\bar{ki}}})(b_{\bar{\bar{jk}}} - b_{\bar{ki}} + b_{\bar{ij}}) \times$$

$$\times(c_{\bar{ki}} + c_{\bar{\bar{ij}}} - c_{\bar{jk}}) -$$

$$-(a_{\bar{\bar{ij}}} - a_{\bar{jk}} + a_{\bar{ki}})(b_{\bar{jk}} + b_{\bar{\bar{ki}}} - b_{\bar{ij}}) \times$$

$$\times(-c_{\bar{ki}} + c_{\bar{ij}} + c_{\bar{\bar{jk}}}) +$$

$$+(a_{\bar{\bar{ij}}} + a_{\bar{\bar{jk}}} + a_{\bar{\bar{ki}}})(b_{\bar{\bar{jk}}} + b_{\bar{\bar{ki}}} + b_{\bar{\bar{ij}}}) \times$$

$$\times(c_{\bar{ki}} + c_{\bar{\bar{ij}}} + c_{\bar{\bar{jk}}})]$$

$$T^1 = \sum_{i,j=0}^{s-1} \{ a_{ij} b_{ij} [(s\text{-}2\delta_{ij}) c_{ij} + \sum_{k=0}^{s-1}{}^{*} (c_{ki} + c_{jk})] +$$

$$+ a_{ij} b_{\bar{ij}} [(s\text{-}\delta_{ij}) c_{\bar{ij}} + \sum_{k=0}^{s-1}{}^{*} (-c_{\bar{ki}} + c_{jk})] +$$

$$+ a_{\bar{ij}} b_{\bar{ij}} [(s\text{-}\delta_{ij}) c_{ij} - \delta_{ji} c_{\bar{\bar{ji}}} +$$

$$+ \sum_{k=0}^{\delta-1}{}^{*} (c_{\bar{ki}} - c_{\bar{jk}})] + a_{\bar{ij}} b_{ij} [(s\text{-}\delta_{ij}) c_{\bar{ij}} -$$

$$- \sum_{k=0}^{s-1}{}^{*} (c_{ki} + c_{\bar{jk}})] +$$

$$+ a_{\bar{ij}} b_{\bar{\bar{ij}}} [(s\text{-}\delta_{ij}) c_{\bar{ij}} - \sum_{k=0}^{s-1}{}^{*} (c_{\bar{\bar{ki}}} + c_{\bar{jk}})] +$$

$$+ a_{\bar{ij}} b_{\bar{ij}} [(s\text{-}\delta_{ij}) c_{\bar{\bar{ij}}} - \delta_{\bar{\bar{ji}}} c_{ji} + \sum_{k=0}^{s-1}{}^{*} (c_{\bar{ki}} - c_{\bar{jk}})] +$$

$$+ a_{\bar{\bar{ij}}} b_{\bar{ij}} [(s\text{-}\delta_{\bar{ij}} c_{\bar{ij}} + \sum_{k=0}^{s-1}{}^{*} (-c_{\bar{ki}} + c_{\bar{\bar{jk}}})] +$$

$$+ a_{\bar{\bar{ij}}} b_{\bar{\bar{ij}}} [(s\text{-}2\delta_{ij}) c_{\bar{\bar{ij}}} + \sum_{k=0}^{s-1}{}^{*} (c_{\bar{\bar{ki}}} + c_{\bar{\bar{jk}}})] \} .$$

$$T^2 = \sum_{i,j=0}^{s-1} \{ a_{ij} \sum_{k=0}^{s-1}{}^{*} (b_{ki} + b_{jk}) c_{ij} -$$

$$- a_{ij} \sum_{k=0}^{s-1}{}^{*} (b_{ki} + b_{\bar{jk}}) c_{\bar{ij}} +$$

$$+ a_{\bar{ij}} \sum_{k=0}^{s-1}{}^{*} (b_{jk} - b_{\bar{ki}}) c_{ij} + a_{\bar{ij}} \times$$

$$\times [\sum_{k=0}^{s-1}{}^{*} (b_{\bar{ki}} - b_{\bar{jk}}) - \delta_{ji} b_{\bar{\bar{ji}}}] c_{\bar{ij}} +$$

$$+ a_{\bar{ij}} [\sum_{k=0}^{s-1}{}^{*} (b_{\bar{ki}} - b_{\bar{jk}}) - \delta_{\bar{\bar{ji}}} b_{ji}] c_{\bar{ij}} +$$

$$+ a_{\bar{ij}} \sum_{k=0}^{s-1}{}^{*} (b_{\bar{\bar{jk}}} - b_{\bar{ki}}) c_{\bar{\bar{ij}}} -$$

$$- a_{\bar{\bar{ij}}} \sum_{k=0}^{s-1}{}^{*} (b_{\bar{\bar{ki}}} + b_{\bar{jk}}) c_{\bar{ij}}$$

$$+ a_{\bar{\bar{ij}}} \sum_{k=0}^{s-1}{}^{*} \left(b_{\bar{ki}} + b_{\bar{\bar{jk}}} \right) c_{\bar{\bar{ij}}} \} .$$

$$T^3 = \sum_{i,j=0}^{s-1} \{ \sum_{k=0}^{s-1}{}^{*} (a_{ki} + a_{jk}) b_{ij} c_{ij} +$$

$$+ [\sum_{k=0}^{s-1}{}^{*} (a_{\bar{ki}} - a_{\bar{jk}}) - \delta_{ji} a_{\bar{\bar{ji}}}] b_{\bar{ij}} c_{\bar{ij}} -$$

$$- \sum_{k=0}^{s-1}{}^{*} \left(a_{ki} + a_{\bar{jk}} \right) b_{\bar{ij}} c_{ij} + \sum_{k=0}^{s-1}{}^{*} \left(a_{jk} - a_{\bar{ki}} \right) b_{ij} c_{\bar{ij}} +$$

$$+ \sum_{k=0}^{s-1}{}^{*} (a_{\bar{j}\bar{k}} - a_{k\bar{i}})\, b_{\bar{i}\bar{j}}\, c_{i\bar{j}} \; - \sum_{k=0}^{s-1}{}^{*} (a_{\bar{k}\bar{i}} + a_{j\bar{k}})\; b_{i\bar{j}}\, c_{\bar{i}\bar{j}} +$$

$$+ \left[\sum_{k=0}^{s-1}{}^{*} (a_{k\bar{i}} - a_{j\bar{k}}) - \delta_{\bar{j}\bar{i}}\, a_{ji} \right] b_{i\bar{j}}\, c_{\bar{i}j} +$$

$$+ \sum_{k=0}^{s-1}{}^{*} (a_{\bar{k}\bar{i}} + a_{\bar{j}\bar{k}})\; b_{\bar{i}\bar{j}}\, c_{\bar{i}\bar{j}} \Big\} .$$

$$\sum_{i,j,k=0}^{n} a_{ij}\, b_{jk}\, c_{jk} = T^0 - T^1 - T^2 - T^3.$$

Here $n=2s$, $\bar{i} = i+s$, $\bar{j} = j+s$, $\bar{k} = k+s$,

S^1 (s) is determined by (10),

δ_{pq} is Croneker's symbol: $\delta_{pq} = \begin{cases} 1 & \text{if } p=q \\ 0 & \text{if } p\neq q, \end{cases}$

the symbol $\sum_{k=0}^{s-1}{}^{*}$ is equivalent with the symbol $\sum_{k=0}^{s-1}$ for $i\neq j$ and with the symbol $\sum_{k\neq i}^{s-1}$ for $i=j$.

Theorem 3. There exists LA having the complexity $\frac{n^3-4n}{3} + 6n^2$.

Proof. See Algorithm 3. □

9. The Main Theorem and Illustrating Tables

Main Theorem. There exist algorithms for MM, MI, ED, SLS involving only $O(N^{\beta})$ arithmetic operations, where $N\times N$ is a size of quadratic matrices involved in a given problem MM, Mi, ED, SLS where $\beta = 1 + \frac{\log(70^2+18\times 70-4)-\lg 3}{\log 70} = 1 + \frac{\log 2052}{\log 70} \approx 2.795$.

The main Theorem follows from theorems 1 and 3 (the latter is applied here for n=70)

Remark 5. Since the proofs of theorems 1 and 3 are constructive, the same is true for the Main Theorem.

Here are two tables illustrating the results of this paper.

Table 9. M(n) for algorithms 2, 2a and for the combinations of the best previously published LA including.[20,23]

n	Algorithm 2	Algorithm 2a	The best previously published
10	725	710	735
12	1188	1170	1127
14	1813	1792	1909
16	2624	2600	2401
18	3645	3618	3703
20	4900	4870	5047
22	6413	6380	6972

For $n\geq 24$ the complexity of algorithm 2a is greater than the complexity of algorithm 3.

Table 10. M(n) and $\log_n$ M(n) for algorithm 3.

n	M(n)	$\log_n$ M(n)
50	56600	2.7974
52	63024	2.7969
54	69912	2.7964
56	77280	2.7961
58	85144	2.7958
60	93520	2.7955
62	102424	2.7954
64	111872	2.79525
66	121880	2.79517
68	132464	2.79513
70	143640	2.79512
72	155424	2.79515
74	167832	2.7952
76	180880	2.7953
78	194584	2.7954
80	208960	2.7955
82	224024	2.7956
84	239792	2.7958
86	256280	2.7959
88	273504	2.7961
90	291480	2.7963

10. Open Problems (Brief Discussion)

In the previous sections new upper bounds on the complexity M = M(n) of LA have been established. They resulted in asymptotically fast algorithms for some important problems. Now two following questions arise. How far can this or similar technique be extended? What are the lower bounds on M(n) and, more generally, on the complexity M(m,n,p) of LA(m,n,p) for matrix multiplications (see Remark 1 in Section 2)? An application of active operation - basic substitution technique *) immediately gives lower bounds $M(m,n,p) \geq (m+n-1)p$ and, in particular, for $m=n=p$, $M(n) \geq 2n^2 - n$.

An application of this technique to a version of LA, which can be called a linear one (see e.g., [6-8]) to distinguish it from a bilinear one described in Section 2 of this paper, and a trilinear one described in Section 3, yields the lower bounds $M(m,n,p) \geq (m-1)(n+1) + np$, $M(n) = M(n,n,n) \geq 2n^2 - 1$ (see [5,28]). These lower bounds can be slightly improved for $(m,n,p) = (2,2,p)$, $(m,n,p) = (2,3,3)$ (see [10,11]), and for $(m,n,p) = (2,3,4)$ and $m=n=p=3$ (see [18]). Even so, the gap between the best known lower and upper bounds is enormous. The present author conjectures that the upper bounds can be essentially reduced by extending his technique. In fact, even a procedure of uniting kin acceptable terms can be done similarly to Sections 5-7 but more accurately. Then an asymptotic improvement of the complexity at least up to $M(n) = \frac{n^3+2n}{3} + \frac{9}{2}n^2$ (see [19]) has been obtained. The present author hopes that the technique introduced in his papers will also be applied for constructing fast algorithms for the evaluation of other kinds of sets of bilinear forms, since the decompositions of LA presented in this paper can easily be extended [19] to a procedure of decomposing any given linear algorithm for evaluation of a given set of bilinear forms or (which is the same) of a given trilinear form.

11. Acknowledgments

My interest in the algebraic complexity theory and particularly in the problem of matrix multiplication was encouraged in the fifties and sixties by Professor V. D. Erokhin (who is not alive now) and by Professor A. G. Vitushkin (at Moscow State University). My coming back to this field in 1977 became possible thanks to Dr. S. Winograd at the IBM T. J. Watson Research Center in Yorktown Heights. I am very grateful to all three of them for this and to the latter also for helpful discussions about the results of this paper and the form of their presentation.

I wish to thank Dr. N. Pippenger at the IBM Research Center for his helpful criticism of my terminology in the initial version of this paper and and also Ms. M. Salvatore and Mrs. T. Asai who typed the paper.

References

1. A. V. Aho, J. E. Hopcroft and J. D. Ullman 1974. **The Design and Analysis of Computer Algorithms,** Addision-Wesley, Reading, Mass.

2. A. Borodin and I. Munro 1975, **The Computational Complexity of Algebraic and Numeric Problems,** American Elsevier.

3. R. W. Brockett and D. Dobkin 1973, "On the optimal evaluation of a set of bilinear forms," **Proc. Fifth Ann. ACM Symp. on Theory of Computing**, pp. 88-95.

4. R. W. Brockett and D. Dobkin 1976, "On the Number of Multiplications Required for Matrix Multiplication," **SIAM J. Comput.**, Vol. 5, No. 4.

5. R. W. Brocket and D. Dobkin 1978, "On the optimal evaluation of a set of bilinear forms," **Linear Algebra and Its Applications**, Vol. 19, pp. 207-235.

6. C.M. Fiduccia 1971, "Fast matrix multiplication," **Proc. Third Ann. ACM Symp. on Theory of Computing**, pp. 45-49.

7. C. M. Fiduccia 1972, "On obtaining upper bounds on the complexity of matrix multiplication," in **Complexity of Computer Computations,** (R. E. Miller and J. W. Thatcher, Eds.) Plenum Press, New York.

8. C. M. Fiduccia 1973, "On algebraic complexity of matrix multiplication," Ph.D. Thesis, Brown University, Providence, R.I.

9. N. Gastinel 1971, "Sur le calcul des products de matrices," **Numer. Math.**, Vol. 17, pp. 222-229.

10. J. E. Hopcroft and L. R. Kerr 1969, "Some Techniques for Proving Certain Simple Programs Optimal," **Proc. of the 1969 Tenth Ann. Symp. on Switching and Automata Theory**, pp. 36-45.

11. J. E. Hopcroft and L. R. Kerr 1971, "On Minimizing the Number of Multiplications Necessary for Matrix Multiplication," **SIAM J. Appl. Math.,** Vol. 20, pp. 30-36.

12. J. E. Hopcroft and J. Musinski 1973, "Duality Applied to the Complexity of Matrix Multiplications and Other Bilinear Forms," **SIAM J. of Computing,** Vol. 2, No. 3, pp. 159-173.

13. Z. M. Kedem and D. G. Kirkpatrick 1977, "Addition Requirements for Rational Functions," **SIAM J. Computing**, Vol. 6, No. 1, pp. 188-199.

*) This technique was introduced for establishing lower bounds on the number of arithmetic operations ± and $\dot{x}$ for polynomial evaluation in [16] (on the number of multiplications/divisions) and in [17] (on the number of additions/subtractions). It was rediscovered in the case of ± in [14] and extended in several directions in [13,22,26].

14. D. G. Kirkpatrick 1972, "On the additions necessary to compute certain functions," **Proc. 4th Ann. ACM Symp. on Theory of Computing,** pp. 94-101.

15. J. D. Laderman 1976, "A noncommutative algorithm for multiplying 3×3 matrices using 23 multiplications," **Bull. of the AMS,** Vol. 82, No. 1.

16. V. Ya. Pan 1962, "On some methods of computing polynomial values," **Problemy Kibernetiki,** Vol. 7, pp. 21-30. (Transl. Problems of Cybernetics, edited by A. A. Lyapunov, USSR, Vol. 7, pp. 20-30, (1962), U. S. Department of Commerce.)

17. V. Ya Pan 1964, "Methods for computing polynomials," Ph.D.Thesis, (in Russian), Department of Mechanics and Mathematics, Moscow State University.

18. V. Ya Pan 1972, "On schemes for the computation of products and inverses of matrices," (in Russian) **Russian Math. Surveys,** Vol. 27, No. 5, pp. 249-250.

19. V. Ya. Pan, to appear.

20. R. L. Probert 1977, "On the composition of matrix multiplication algorithms," **Proc. of Sixth Manitoba Conf. on Num. Math. and Computing,** Congressus Numerantium 18, pp. 357-366.

21. V. Strassen 1969, "Gaussian Elimination is not Optimal," **Num. Math.,** Vol.13, pp. 354-356.

22. V. Strassen 1972, "Evaluation of Rational Functions," in **Complexity of Computer Computations,** (R. E. Miller and J. W. Thatcher, Eds.) Plenum Press, New York, pp.1-10.

23. O. Sykora, "A Fast Non-commutative Algorithm for Matrix Multiplication," to appear.

24. V. Strassen 1973, "Vermeidung von Divisionen," **J. Reine Angew. Math.,** Vol. 264, pp. 184-202.

25. S. Winograd 1968, "A new algorithm for inner product," **IEEE Trans. on Computers,** Vol. C-17, pp. 693-694.

26. S. Winograd (1970), "On the Number of Multiplications Necessary to Compute Certain Functions," **Comm. on Pure and Applied Math.**, Vol. 23, pp. 165-179.

27. S. Winograd (1971), "On Multiplication of 2×2 Matrices," **Linear Algebra and Its Applications**, Vol. 4, pp. 381-388.

28. S. Winograd (to appear).