SIAM J. COMPUT.
Vol. 15, No. 2, May 1986

021

NONCOMMUTATIVE BILINEAR ALGORITHMS FOR 3×3 MATRIX MULTIPLICATION*

RODNEY W. JOHNSON† AND AILEEN M. McLOUGHLIN‡

Abstract. New noncommutative bilinear algorithms for 3×3 matrix multiplication are presented. These have the same complexity, 23 essential multiplications, as the one discovered by Laderman, but are inequivalent to it. Equivalence here refers to a certain group of transformations all of which map noncommutative bilinear matrix-multiplication algorithms into other such algorithms; "inequivalent" means not related by a transformation in the group. This group has been studied by de Groote, who has shown for the case of 2×2 matrix multiplication with 7 essential multiplications that all such algorithms are equivalent to Strassen's. The new algorithms, by contrast, include infinitely many pairwise inequivalent algorithms. The computer search that led to the new algorithms is described.

Key words. matrix multiplication, computational complexity, Strassen's algorithm

1. Introduction. We here present new noncommutative bilinear algorithms for 3×3 matrix multiplication over the field of rational numbers—that is, we specify rational coefficients A^r_{ij}, B^r_{kl}, C^r_{mn} such that, with $N=3$, the equation

$$\sum_{p=1}^{N} X_{np}Y_{pm}=\sum_{r=1}^{M}\left(\sum_{i,j=1}^{N} A^r_{ij}X_{ij}\right)\left(\sum_{k,l=1}^{N} B^r_{kl}Y_{kl}\right)C^r_{mn} \tag{1}$$

is an identity for $N\times N$ matrices X and Y.

The problem of finding such algorithms originally received attention because each such algorithm yields an upper bound $O(n^a)$ on the number of arithmetic operations needed for multiplication of $n\times n$ matrices, where $a=\log_N M$. Strassen's algorithm [1] ($N=2$, $M=7$) had been shown to be optimal for $N=2$ [2], [3], and it seemed that one avenue to reducing the exponent a was to find algorithms for other small values of N and sufficiently small M, such as $N=3$, $M=21$. Recent reductions in the exponent by Pan [4], [5], [6], Bini et al. [7], [8], Schönhage [9], and Coppersmith and Winograd [10] have been achieved by rather different means with the help of several new ideas; for $N\geq 3$, it remains an interesting unsolved problem to determine the minimum complexity of noncommutative bilinear algorithms for $N\times N$ matrix multiplication.

Laderman, in [11], gave an algorithm with $M=23$, the best value known for $N=3$ with rational coefficients. (Unpublished work of Schönhage's [12] indicates that $M=22$ is achievable with *complex* coefficients.) The algorithms we present here have the same complexity, $M=23$ essential multiplications, as Laderman's but are inequivalent to it in the following sense.

Several sorts of transformations on families of coefficients A^r_{ij}, B^r_{kl}, C^r_{mn} have the property of mapping any (noncommutative, bilinear) algorithm for $N\times N$ matrix multiplication again into such an algorithm. We describe these using the notation A^r for the $N\times N$ matrix with elements A^r_{ij}, with similar definitions for B^r and C^r. The transformations include replacement of these coefficient matrices A^r, B^r, C^r by:

$$A^{\pi(r)},\quad B^{\pi(r)},\quad C^{\pi(r)}, \tag{2}$$

* Received by the editors October 3, 1984.

† Naval Research Laboratory, Washington, DC 20375.

‡ Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180.

for some permutation π of the indices $1, \cdots, M$;

$$C^r, \quad A^r, \quad B^r \tag{3}$$

(cyclic permutation);

$$\tilde{C}^r, \quad \tilde{B}^r, \quad \tilde{A}^r, \tag{4}$$

where the tilde denotes transposition;

$$a_r A^r, \quad b_r B^r, \quad c_r C^r, \tag{5}$$

where a_r, b_r, c_r are rational numbers such that $a_r b_r c_r = 1$ for $r = 1, \cdots, M$; and

$$PA^r Q^{-1}, \quad QB^r R^{-1}, \quad RC^r P^{-1} \tag{6}$$

where P, Q, R are invertible $N \times N$ matrices. Two algorithms will be called *equivalent* if the coefficients of one are mapped to those of the other by a sequence of transformations of the forms (2)-(6); otherwise they are *inequivalent.*

De Groote [13], [14], [15] has studied the group generated by such transformations and has shown [14] that Strassen's algorithm is essentially unique: any algorithm with $N = 2$ and $M = 7$ is equivalent to Strassen's. (De Groote worked over an arbitrary field K, not assumed to be the rational numbers. Pan had stated the result without proof in [16] for the case of the rational numbers. Hopcroft and Musinski [17] had treated the case of the integers (mod 2).)

The algorithms presented here show that, in contrast with Strassen's algorithm, Laderman's is not essentially unique; in fact they form a 3-parameter family and a 1-parameter family that contain infinitely many pairwise inequivalent algorithms with $N = 3$ and $M = 23$. The coefficients of the first family of algorithms are shown in Table 1; the 3 parameters are x, y, and z. The coefficients for the second family are in Table 2; the parameter is x. Only for 5 values of r in each table do the entries actually depend on the parameters. We discovered the new algorithms with the help of a computer search, which we describe in the next section. In the third section we discuss the question of the algorithms' inequivalence.

2. Search procedure. A necessary and sufficient condition for (1) to hold identically in X and Y is that the coefficients satisfy

$$\sum_{r=1}^{M} A_{ij}^r B_{kl}^r C_{mn}^r = \delta_{ni} \delta_{jk} \delta_{lm}. \tag{7}$$

Solutions of (7) correspond to zeros of

$$\sum_{i,j,k,l,m,n=1}^{N} \left(\sum_{r=1}^{M} A_{ij}^r B_{kl}^r C_{mn}^r - \delta_{ni} \delta_{jk} \delta_{lm} \right)^2, \tag{8}$$

which is a nonnegative function of the coefficients. We sought solutions of (7) by trying to minimize (8). Although (8) is a sixth-degree polynomial, it is only quadratic in the A_{ij}^r when the other coefficients are held fixed; likewise it is quadratic as a function of the B_{kl}^r alone or of the C_{mn}^r alone. Since the minimum of a quadratic polynomial can be obtained from the solution of a set of linear equations, it is straightforward to minimize (8) with respect to any one of the three sets of coefficients separately. We wrote a program that assigns random starting values to the matrices A^r and B^r and, holding these fixed, determines values for the C^r that minimize (8). Next it determines new values for the B^r that minimize (8) with the A^r and C^r held fixed. It then minimizes with respect to the A^r with the new B^r and the C^r held fixed, minimizes with respect

TABLE 1
Coefficients for three-parameter algorithm family.

r	A^r			B^r			C^r		
	1	0	−1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
2	1	1	−1	1	1	0	1	1	0
	0	0	0	0	0	0	0	0	0
	1	−1	−1	0	1	−1	0	0	0
3	−1	1	1	0	0	0	1	0	0
	1/3	1	1	0	0	0	0	0	0
	0	$1/(1+xy)$	0	0	0	0	1	z	$-(1-y)(1-z)$
4	0	0	0	1	0	x	−1	$-z$	$(1-y)(1-z)$
	0	0	0	0	0	0	y	yz	$-2(1-y)(1-z)$
	0	−1	0	0	−1	1	0	0	1
5	0	1	0	−1	−1	0	1	0	−1
	1/3	1	0	0	0	0	0	0	2
	0	0	1	0	1	−1	0	1	0
6	0	0	−1	0	0	0	0	−1	0
	0	0	−1	0	1	1	3/2	3/2	0
	0	0	0	0	0	0	0	0	2
7	0	0	0	0	1	1/2	0	0	−2
	0	−1	1	0	0	0	0	0	2
	1	0	0	0	0	−1	0	0	0
8	−1	0	0	0	0	0	−1	0	0
	0	0	0	0	0	0	−1	0	0
	0	0	1	1	0	0	1	1	0
9	0	0	0	0	0	0	−1	−1	0
	0	0	0	1	0	0	1	1	0
	0	$1/(1+xy)$	0	0	0	0	x	xz	$-(1+x)(1-z)$
10	0	0	0	y	0	−1	$-x$	$-xz$	$(1+x)(1-z)$
	0	0	0	0	0	0	−1	$-z$	$-2(1+x)(1-z)$
	0	0	0	−1	−1/3	1/3	0	3/2	−1
11	−1	0	1	−2/3	−2/3	0	−3/2	−3/2	0
	0	0	1	−1	−1	0	0	0	0
	0	0	0	0	0	0	0	0	−2
12	0	0	0	1	1	0	−1	0	1
	−1/3	−1	1	0	0	0	0	0	−2
	0	0	−2/3	1	1	−1	0	3/2	0
13	−1/3	0	1	0	0	0	−3/2	−3/2	0
	0	0	1	1	1	0	3	3	0
	0	0	0	0	0	0	0	−1	1
14	1	0	−1	1	1	0	0	0	0
	0	0	0	1	1	0	0	0	0
	0	0	−1/2	1	1/2	−1/2	0	0	0
15	0	0	3/2	0	0	0	0	0	0
	0	0	3/2	1	1/2	−1/2	−2	−2	0
	0	0	0	−1	−1/3	1/3	0	0	−1
16	1	0	−1	1/3	1/3	0	0	0	0
	1	0	−1	0	0	0	0	0	0

TABLE 1—continued

r	A^r			B^r			C^r		
	0	$-z$	0	0	0	0	0	-1	-1
17	0	1	0	-1	0	$1/2$	0	1	1
	0	0	1	0	0	$-1/2$	0	0	-2
	0	0	0	-1	-1	0	0	0	0
18	1	0	0	0	0	0	-1	-1	0
	0	0	0	-1	-1	0	-1	-1	0
	0	$-z$	0	0	0	0	0	$-1/2$	$1/2$
19	0	1	0	0	0	-1	0	$1/2$	$-1/2$
	0	0	0	0	0	-1	0	-1	1
	0	0	0	0	0	0	0	1	-1
20	0	0	0	0	-1	$-1/2$	0	-1	1
	0	0	-1	0	-1	$-1/2$	0	0	0
	0	0	0	0	-1	$-1/2$	0	0	0
21	0	0	0	0	0	0	0	0	$-2/3$
	1	0	0	0	0	0	0	0	$-2/3$
	0	z	-1	0	0	0	0	-1	0
22	0	-1	1	0	0	0	0	1	0
	0	0	0	0	0	-1	0	-1	0
	0	1	0	0	-1	1	0	0	1
23	0	-1	0	0	-1	-1	0	0	-1
	0	-1	0	0	0	0	0	0	2

to the C^r with the new A^r and B^r held fixed, and continues thus cyclically. This results in a decreasing sequence of values for (8); the search is considered to be successful if these appear to be converging to 0 while the coefficients converge to finite limits.

One difficulty that proved troublesome in practice was "zeros at infinity." For some searches, some of the coefficients seemed to be tending to infinity in such a way that (8) was tending to zero. This phenomenon is undoubtedly due to the existence of arbitrary-precision approximating (APA) algorithms in the sense of Bini et al. [7], which implies that there are sequences of values for the coefficients that behave as described. APA algorithms for 3×3 matrix multiplication are known to exist; indeed Schönhage [9] has constructed such an algorithm with $M=21$.

The difficulty was countered with a modification of the expression the programs were attempting to minimize; a term

$$\varepsilon \sum_{rij} ((A^r_{ij})^2+(B^r_{ij})^2+(C^r_{ij})^2) \tag{9}$$

was added to (8). The coefficient ε was adjusted by trial and error, interactively, so that, if possible, the magnitudes of the coefficients would stay bounded or decrease at the same time that the value of (8) was decreasing. If a suitable value for ε could not be found, new random starting values were chosen for the coefficients and the search was begun again.

The procedure just described typically yields tables of rather arbitrary-looking floating-point numbers as values for the coefficients and a small but nonzero positive number, such as 10^{-6}, as the value of (8) computed to machine precision. One would like coefficient values that can be expressed exactly and that yield 0 as the exact value of (8). Simple rational coefficient values are therefore desirable—that is, ratios of small

TABLE 2
Coefficients for one-parameter algorithm family.

r	A^r			B^r			C^r		
	1	0	0	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
2	0	1	0	−1	−1	0	1	−1	0
	0	0	0	0	0	0	0	0	0
	0	−1	1	0	0	0	0	−1	−1
3	0	−1	0	1	0	0	0	1	1
	0	0	0	0	−1	1	1	0	1
	0	0	0	0	0	0	0	−1	0
4	−1	0	−1	0	0	0	0	0	0
	0	0	0	1	1	0	0	0	0
	$-x$	$x-1$	0	0	$1-2/x$	−1	0	0	0
5	0	$x-1$	0	0	1	−1	1/2	0	1/2
	0	1	0	0	1	−1	1/2	0	−1/2
	0	0	0	1	0	0	0	1	0
6	1	0	0	0	0	0	0	−1	−1
	−1/2	0	1/2	−1	−1	0	0	0	1
	−1	1	0	0	$1-x$	$x-1$	0	0	0
7	0	1	0	0	$-x$	x	0	0	0
	0	0	0	0	$-x$	x	1	0	0
	0	0	0	−1	−1	0	0	0	0
8	−1	0	0	0	0	0	0	1	1
	0	0	0	1	1	0	0	0	−1
	0	0	0	0	1	1	0	0	0
9	0	0	0	0	0	0	0	0	0
	1	−1	0	0	−1	1	0	0	1
	0	0	0	0	0	0	0	1	1
10	0	0	1	0	0	0	0	−1	−1
	0	0	0	0	0	−1	0	−1	−1
	−1	$1-1/x$	0	0	1	0	0	0	0
11	−1	$1-1/x$	0	0	0	0	0	0	1
	1	$1/x-1$	0	0	0	0	0	0	−1
	0	0	0	−1	0	0	0	1/2	1/2
12	0	0	0	0	0	0	0	−1/2	−1/2
	−1	0	1	1	0	1	0	0	1/2
	$-x$	$x-1$	0	0	1	−1	0	0	0
13	0	$x-1$	0	0	1	−1	−1/2	0	−1/2
	0	0	0	0	1	−1	1/2	0	1/2
	0	1	−1	0	0	0	1	−1	0
14	0	0	0	1	0	0	−1	1	0
	0	0	0	0	0	0	0	0	0
	0	−1	1	0	0	0	0	0	1
15	0	−1	1	0	0	0	0	0	−1
	0	1	−1	0	1	0	0	0	−1
	0	0	0	0	0	0	0	0	0
16	0	1	0	0	0	−1	0	0	0
	0	0	0	0	0	0	1	−1	0

TABLE 2—*continued*

r	A^r			B^r			C^r		
	0	1	0	0	$1-1/x$	0	0	0	0
17	0	1	0	0	1	0	1	0	0
	0	−1	0	0	1	0	1	0	0
	0	0	0	0	0	1	0	0	0
18	−1	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	−1	0
	0	0	1	0	0	0	1	0	0
19	0	0	0	1	0	0	0	0	0
	0	0	0	1	0	0	0	0	0
	0	−1	1	0	0	0	0	0	1
20	0	−1	0	1	0	0	−1	0	−1
	0	1	0	0	−1	0	−1	0	−1
	0	0	0	1	0	0	0	0	−1/2
21	0	0	0	0	0	0	0	0	0
	−1	0	−1	1	1	0	0	0	0
	0	1	−1	0	0	0	0	−1	−1
22	0	1	−1	0	0	0	0	1	1
	−1/2	0	1/2	0	−1	1	0	0	1
	0	0	0	0	−1/2	−1/2	0	0	0
23	0	0	0	−1	−1/2	−1/2	−1	0	−1
	0	1	0	0	1/2	−1/2	−1	0	−1

integers. (Ideally, one would like the coefficient values to be confined to 0 and ± 1, as indeed they are for Strassen's and Laderman's algorithms.)

The minimization procedure is unlikely to lead to a simple rational solution of (7), even if one exists; with any such solution, transformations (5) and (6) associate an infinite family of equivalent solutions, most of which do not consist of simple rational numbers. We therefore wrote procedures for performing transformations of the forms (5) and (6). When the minimization procedure appeared to be converging to a zero of (8), we used these procedures in an attempt to transform the solution to a simple rational form—if possible, consisting of 1's, 0's, and -1's. We would arbitrarily choose one of the coefficient matrices—for example B^1; we would then find values for the matrices Q and R in (6) and the scalars a_1, b_1, and c_1 in (5) that would transform B^1 to diagonal form with 1's and 0's on the diagonal, as in Table 1. Enough freedom remained in the choice of P, Q, R, and the scalars to permit further simplifications, such as the transformation of A^1 and C^1 and the coefficients for $r=2$ to the form shown in the table. When that freedom had been exhausted, the coefficients for several values of r had attained the values shown in the table to within $\pm 10^{-3}$. We altered these coefficients, replacing the approximate rational values by exact rational values (to machine precision). For other values of r, the coefficients were not close to any obvious simple rational values. In Table 1 these included $r=4$, 10, 17, 19, 22: the 5 rows of the table that show dependence on the parameters x, y, and z.

We wrote a version of the search procedure that would minimize (8) with respect to the coefficients for selected values of r only, holding all coefficients fixed for the remaining r. We held fixed the coefficients that we had set to rational values and found that the search still seemed to be converging to a zero of (7). Moreover, by arbitrarily perturbing some of the nonfixed coefficients, we found it possible to guide the search so that additional coefficients tended toward rational values. We could thus simplify additional rows of the table, replace approximate rational values by exact, and add to the list of values of r for which the coefficients were held fixed. The 5 parameter-dependent rows were the last to be simplified. We found that we could vary certain of the coefficients in these rows practically at will and still obtain good numerical solutions to (7) with the coefficients in the other rows held fixed. This observation led to the parametrized family explicitly presented in Table 1. The same procedure, with different random starting values for the search, led to the family shown in Table 2.

Similar, earlier computer searches for minima of (8) were undertaken by Brent [18], Ungar [19], and Lafon [20], all of whom report successes with $M=7$, $N=2$: computer runs for which (8) became small and appeared to be converging to 0. Except for the ε term (9), the minimization procedure we have described is basically identical to Brent's [18]. Brent reported apparent success with $N=3$, $M=25$, but did not try to simplify the solution by equivalence transformations such as (6), (7). Schönhage [12], by admitting complex coefficients, has obtained numerical solutions (unsimplified) of (7) with $N=3$, $M=22$.

3. Inequivalence. For comparison with Tables 1 and 2, the coefficients of Laderman's algorithm are shown in Table 3.

We have claimed that none of the new algorithms is equivalent to Laderman's algorithm. To prove this, we point out that, except for permutations, the transformations (2)–(6) leave the ranks of the matrices A^r, B^r, and C^r unchanged. Six matrices in Table 3 (A^1 and B^3, for instance) have rank 3. But regardless of the parameters, all the matrices in Table 1 have rank 1 or 2, and just one matrix (B^{23}) in Table 2 has rank 3. Therefore, no combination of transformations (2)–(6) can change the coefficients in Table 3 into those given in Table 1 or Table 2. That is, the algorithm presented in Table 3 is not equivalent to any algorithm of the families presented in Tables 1 and 2.

Similarly, the algorithms of the family in Table 1 are inequivalent to those of Table 2.

Next we consider whether distinct algorithms within a family are equivalent. Let (x', y', z') and (x'', y'', z'') be distinct triples of values for the parameters in Table 1. If

$$x' = -1/y'', \quad y' = -1/x'', \quad z' = z'', \tag{10}$$

then the corresponding algorithms are equivalent by a permutation (2) that interchanges $r=4$ and $r=10$ together with an obvious scaling (5). Otherwise the algorithms are inequivalent. In Table 2, distinct values x' and x'' of the parameter always correspond to inequivalent algorithms.

The proof of these statements is entirely elementary, and too tedious to give in full. We merely sketch the ideas involved. For definiteness, consider the family of Table 2, and write $A(x)$, $B(x)$, $C(x)$ to show the dependence of the coefficients on the parameter.

Let x' and x'' be values of the parameter, and suppose the algorithm with coefficients $A(x')$, $B(x')$, $C(x')$ is equivalent to that with coefficients $A(x'')$, $B(x'')$, $C(x'')$; we need to show that $x' = x''$. For some permutation π, numbers a_r, b_r, c_r, and invertible matrices P, Q, R, one of six sets of equations holds:

$$\begin{aligned} A^r(x') &= a_r P A^{\pi(r)}(x'') Q^{-1}, \\ B^r(x') &= b_r Q B^{\pi(r)}(x'') R^{-1}, \\ C^r(x') &= c_r R C^{\pi(r)}(x'') P^{-1}, \end{aligned} \tag{11}$$

TABLE 3
Coefficients for Laderman's algorithm.

r	A^r	B^r	C^r	r	A^r	B^r	C^r
1	$\begin{matrix}1&1&1\\-1&-1&0\\0&-1&-1\end{matrix}$	$\begin{matrix}0&0&0\\0&1&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\1&0&0\\0&0&0\end{matrix}$	13	$\begin{matrix}0&0&1\\0&0&0\\0&0&-1\end{matrix}$	$\begin{matrix}0&0&0\\0&1&0\\0&-1&0\end{matrix}$	$\begin{matrix}0&0&1\\0&0&1\\0&0&0\end{matrix}$
2	$\begin{matrix}1&0&0\\-1&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&-1&0\\0&1&0\\0&0&0\end{matrix}$	$\begin{matrix}0&1&0\\0&1&0\\0&0&0\end{matrix}$	14	$\begin{matrix}0&0&1\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\1&0&0\end{matrix}$	$\begin{matrix}1&1&1\\1&0&1\\1&1&0\end{matrix}$
3	$\begin{matrix}0&0&0\\0&1&0\\0&0&0\end{matrix}$	$\begin{matrix}-1&1&0\\1&-1&-1\\-1&0&1\end{matrix}$	$\begin{matrix}0&1&0\\0&0&0\\0&0&0\end{matrix}$	15	$\begin{matrix}0&0&0\\0&0&0\\0&1&1\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\-1&1&0\end{matrix}$	$\begin{matrix}0&0&0\\1&0&1\\0&0&0\end{matrix}$
4	$\begin{matrix}-1&0&0\\1&1&0\\0&0&0\end{matrix}$	$\begin{matrix}1&-1&0\\0&1&0\\0&0&0\end{matrix}$	$\begin{matrix}0&1&0\\1&1&0\\0&0&0\end{matrix}$	16	$\begin{matrix}0&0&-1\\0&1&1\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&1\\1&0&-1\end{matrix}$	$\begin{matrix}0&1&0\\0&0&0\\1&1&0\end{matrix}$
5	$\begin{matrix}0&0&0\\1&1&0\\0&0&0\end{matrix}$	$\begin{matrix}-1&1&0\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\1&1&0\\0&0&0\end{matrix}$	17	$\begin{matrix}0&0&1\\0&0&-1\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&1\\0&0&-1\end{matrix}$	$\begin{matrix}0&1&0\\0&0&0\\0&1&0\end{matrix}$
6	$\begin{matrix}1&0&0\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}1&0&0\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}1&1&1\\1&1&0\\1&0&1\end{matrix}$	18	$\begin{matrix}0&0&0\\0&1&1\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\-1&0&1\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\1&1&0\end{matrix}$
7	$\begin{matrix}-1&0&0\\0&0&0\\1&1&0\end{matrix}$	$\begin{matrix}1&0&-1\\0&0&1\\0&0&0\end{matrix}$	$\begin{matrix}0&0&1\\0&0&0\\1&0&1\end{matrix}$	19	$\begin{matrix}0&1&0\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\1&0&0\\0&0&0\end{matrix}$	$\begin{matrix}1&0&0\\0&0&0\\0&0&0\end{matrix}$
8	$\begin{matrix}-1&0&0\\0&0&0\\1&0&0\end{matrix}$	$\begin{matrix}0&0&1\\0&0&-1\\0&0&0\end{matrix}$	$\begin{matrix}0&0&1\\0&0&0\\0&0&1\end{matrix}$	20	$\begin{matrix}0&0&0\\0&0&1\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\0&1&0\end{matrix}$	$\begin{matrix}0&0&0\\0&1&0\\0&0&0\end{matrix}$
9	$\begin{matrix}0&0&0\\0&0&0\\1&1&0\end{matrix}$	$\begin{matrix}-1&0&1\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\1&0&1\end{matrix}$	21	$\begin{matrix}0&0&0\\1&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&1\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\0&1&0\end{matrix}$
10	$\begin{matrix}1&1&1\\0&-1&-1\\-1&-1&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&1\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\1&0&0\end{matrix}$	22	$\begin{matrix}0&0&0\\0&0&0\\1&0&0\end{matrix}$	$\begin{matrix}0&1&0\\0&0&0\\0&0&0\end{matrix}$	$\begin{matrix}0&0&0\\0&0&1\\0&0&0\end{matrix}$
11	$\begin{matrix}0&0&0\\0&0&0\\0&1&0\end{matrix}$	$\begin{matrix}-1&0&1\\1&-1&-1\\-1&1&0\end{matrix}$	$\begin{matrix}0&0&1\\0&0&0\\0&0&0\end{matrix}$	23	$\begin{matrix}0&0&0\\0&0&0\\0&0&1\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\0&0&1\end{matrix}$	$\begin{matrix}0&0&0\\0&0&0\\0&0&1\end{matrix}$
12	$\begin{matrix}0&0&-1\\0&0&0\\0&1&1\end{matrix}$	$\begin{matrix}0&0&0\\0&1&0\\1&-1&0\end{matrix}$	$\begin{matrix}0&0&1\\1&0&1\\0&0&0\end{matrix}$				

or one of five similar sets obtained from these by permutations of $A(x'')$, $B(x'')$, $C(x'')$, on the right, possibly with transposition (cf. (3), (4)). The steps of the proof are to show that (11) holds; that P, Q, and R are scalar multiples of the identity matrix; that $\pi(r)=r$ for all r; and finally that $x'=x''$.

Inspection of a tabulation of the ranks rk A^r, rk B^r, rk C^r of the matrices in Table 2 suffices for a proof that (11) holds.

Consideration of the tabulated ranks also permits a proof, for a few particular values of r, that $\pi(r)=r$. We obtain further information by considering inclusion relations between row spaces or between column spaces of matrices. For instance we have col $C^9(x') \subseteq \text{col } C^{22}(x')$, which by (11) implies col $C^{\pi(9)}(x'') \subseteq \text{col } C^{\pi(22)}(x'')$. From a listing of such inclusion relations, we can show for a number of additional values of r that $\pi(r) = r$. Now when r is such that $\pi(r) = r$ and $C^r(x') = C^r(x'')$, it follows from (11) that the column space col $C^r(x')$ is an invariant subspace for R. It is possible to identify in this way enough invariant subspaces for R to find 4 eigenvectors of R, no 3 of which are linearly dependent. From this it follows that R is a scalar multiple of the identity. A similar analysis shows that P and Q are likewise scalar multiples of the identity.

It now follows from (11) that $A^{\pi(r)}(x'')$, $B^{\pi(r)}(x'')$, and $C^{\pi(r)}(x'')$ are scalar multiples of $A^r(x')$, $B^r(x')$, and $C^r(x')$. For all r this implies $\pi(r) = r$. Then $A^5(x'')$ is a scalar multiple of $A^5(x')$; it follows that $x' = x''$.

REFERENCES

[1] V. STRASSEN, *Gaussian elimination is not optimal*, Numer. Math., 13 (1969), pp. 354–356.

[2] J. E. HOPCROFT AND L. R. KERR, *On minimizing the number of multiplications necessary for matrix multiplication*, SIAM J. Appl. Math., 20 (1971), pp. 30–36.

[3] S. WINOGRAD, *On multiplication of* 2×2 *matrices*, Linear Algebra and Appl., 4 (1971), pp. 381–388.

[4] V. YA. PAN, *Strassen's algorithm is not optimal—trilinear technique of aggregating, uniting and canceling for constructing fast algorithms for matrix operations*, Proc. 19th Annual Symposium on Foundations of Computer Science, Oct. 1978, pp. 166–176.

[5] ——, *Field extension and trilinear aggregating, uniting and canceling for the acceleration of matrix multiplications*, Proc. 20th Annual Symposium on Foundations of Computer Science, 1979.

[6] ———, *New fast algorithms for matrix operations*, this Journal, 9 (1980), pp. 321–342.

[7] D. BINI, M. CAPOVANI, F. ROMANI AND G. LOTTI, $O(n^{2.7799})$ *complexity for* $n\times n$ *approximate matrix multiplication*, Inform. Proc. Lett., 8 (1979), pp. 234–235.

[8] D. BINI, G. LOTTI AND F. ROMANI, *Approximate solutions for the bilinear form computational problem*, this Journal, 9 (1980), pp. 692–697.

[9] A. SCHÖNHAGE, *Partial and total matrix multiplication*, this Journal, 10 (1981), pp. 434–455.

[10] D. COPPERSMITH AND S. WINOGRAD, *On the asymptotic complexity of matrix multiplication*, this Journal, 11 (1982), pp. 472–492.

[11] J. LADERMAN, *A noncommutative algorithm for multiplying* 3×3 *matrices using* 23 *multiplications*, Bull. Amer. Math. Soc., 82 (1976), pp. 126–128.

[12] A. SCHÖNHAGE, private communication.

[13] H. F. DE GROOTE, *On varieties of optimal algorithms for the computation of bilinear mappings.* I: *The isotropy group of a bilinear mapping*, Theoret. Comp. Sci., 7 (1978), pp. 1–24.

[14] ———, *On varieties of optimal algorithms for the computation of bilinear mappings.* II: *Optimal algorithms for* 2×2 *matrix multiplication*, Theoret. Comp. Sci., 7 (1978), pp. 127–148.

[15] ———, *On varieties of optimal algorithms for the computation of bilinear mappings.* III: *Optimal algorithms for the computation of xy and yx where* $x, y \in M_2(K)$, Theoret. Comp. Sci., 7 (1978), pp. 239–249.

[16] V. YA. PAN, *On schemes for computation of the product of matrices and the inverse of a matrix*, Uspekhi Mat. Nauk, 27, 5 (1972), pp. 249–250. (In Russian.)

[17] J. HOPCROFT AND J. MUSINSKI, *Duality applied to the complexity of matrix multiplication and other bilinear forms*, this Journal, 2 (1973), pp. 159–173.

[18] R. P. BRENT, *Algorithms for matrix multiplication*, Tech. Rep. CS 157, Computer Science Dept., Stanford Univ., Stanford, CA, March 1970.

[19] P. UNGAR, private communication.

[20] J. C. LAFON, Ph.D. dissertation, Univ. of Grenoble.