Matrix Multiplication via Arithmetic Progressions

Don Coppersmith and Shmuel Winograd
Department of Mathematical Sciences
IBM Thomas J Watson Research Center
P O Box 218
Yorktown Heights, New York 10598

Abstract.

We present a new method for accelerating matrix multiplication asymptotically. This work builds on recent ideas of Volker Strassen, by using a basic trilinear form which is not a matrix product. We make novel use of the Salem-Spencer Theorem, which gives a fairly dense set of integers with no three-term arithmetic progression. Our resulting matrix exponent is 2.376.

1. Introduction.

A matrix multiplication algorithm is usually built as follows. First an algorithm for a small matrix problem is developed. Then a tensor product construction produces from it an algorithm for multiplying large matrices. Several times over the last seventeen years, the ground rules for constructing the basic algorithm have been relaxed, and with more care in the tensor product construction, it has been shown how to use these more lenient basic constructions to still give efficient algorithms for multiplying large matrices.

Recently Strassen [Str] found a new relaxation of the ground rules. His basic trilinear algorithm computes a trilinear form which is not a matrix product at all. In this trilinear form, the variables are collected into blocks. The block structure (the arrangement of the blocks) is that of a matrix product, and the fine structure (the arrangement of variables within individual blocks) is also that of a matrix product, but the overall structure is not, because the fine structures of different blocks are incompatible. After taking a tensor power of this trilinear form, Strassen operates on the block structure (that of a large matrix product) to reduce it to several block scalar multiplications. Each block scalar multiplication is itself a matrix product (the fine structure), so that he has several disjoint matrix products (sharing no variables). He can then apply Schönhage's τ-theorem to obtain an estimate of the matrix exponent ω:

$$\omega < 2.479.$$

Here we follow Strassen's lead. We use a basic trilinear algorithm closely related to Strassen's. The block structure of our trilinear form is not a matrix product, although the fine structure still is. We use a combinatorial theorem of Salem and Spencer [SS], which gives a fairly dense set of integers containing no three-term arithmetic progression. We hash the indices of the blocks of variables to integers, and set to zero any block of variables whose indices do not map into the Salem-Spencer set. We do this in such a way that if the product $x_I y_J z_K$ is contained in our trilinear form, then the hash values $b_x(I)$, $b_y(J)$, $b_z(K)$ form an arithmetic progression. So for any product of nonzero variables $x_I y_J z_K$ in our trilinear form, we will get $b_x(I) = b_y(J) = b_z(K)$. We choose parameters so that on average each nonzero block of variables X_I is contained in at most one nonzero block product $X_I Y_J Z_K$, and set to zero some blocks of variables to ensure that this condition holds absolutely, not just on average. Then, as Strassen, we have several disjoint matrix products, and can apply Schönhage's τ-theorem to obtain our exponent

$$\omega < 2.376.$$

The rest of the paper is organized as follows. In Section 2 we review Schönhage's τ-theorem. In Section 3 we present Strassen's construction. Section 4 contains the results of the Salem-Spencer theorem. In Section 5 we present an easy version of our construction, which gives an exponent of 2.404. The version presented in Section 6 uses exactly the same ideas, but is complicated by more terms and more indices; this gives an intermediate exponent of 2.388. The full paper contains the general theory, and a far more complicated starting case, yielding 2.376. We make concluding remarks in Section 7.

Readers unfamiliar with previous work in matrix multiplication are referred to Victor Pan's excellent survey [Pan].

Acknowledgments.

We are grateful to James Shearer for the reference to Behrend's construction, and to Victor Pan for that of Salem and Spencer. Arnold Schönhage gave a more symmetric presentation of our starting algorithm in section 5. James Davenport offered helpful comments on an early draft of the paper.

2. Schönhage's Theorem

The basic results from "classical" matrix multiplication can be summarized by Schönhage's τ-theorem:

Theorem (Schönhage): *Assume given a field* F, *and coefficients* $\alpha_{ijh\ell}$, $\beta_{jkh\ell}$, $\gamma_{kih\ell}$ *in* $F(\lambda)$ *(the field of rational functions in a single indeterminate* λ*), such that*

$$\sum_{\ell=1}^{L}\left(\sum_{ijh}\alpha_{ijh\ell}x_{ij}^{(h)}\right)\left(\sum_{jkh}\beta_{jkh\ell}y_{jk}^{(h)}\right)\left(\sum_{kih}\gamma_{kih\ell}z_{ki}^{(h)}\right)$$
$$=\sum_{h}\left(\sum_{i=1}^{m_h}\sum_{j=1}^{n_h}\sum_{k=1}^{p_h}x_{ij}^{(h)}y_{jk}^{(h)}z_{ki}^{(h)}\right)+\sum_{g>0}\lambda^{g}f_g(x_{ij}^{(h)},y_{jk}^{(h)},z_{ki}^{(h)})$$

is an identity in $x_{ij}^{(h)}, y_{jk}^{(h)}, z_{ki}^{(h)}, \lambda$, *where* f_g *are arbitrary trilinear forms. Then given* $\varepsilon > 0$, *one can construct an algorithm to multiply* $N \times N$ *square matrices in* $O(N^{3\tau+\varepsilon})$ *operations, where* τ *satisfies*

$$L=\sum_{h}(m_hn_hp_h)^{\tau}. \tag{1}$$

We will also write the error term as $O(\lambda)$, so that the hypothesis becomes

$$\sum_{\ell=1}^{L}\left(\sum_{ijh}\alpha_{ijh\ell}x_{ij}^{(h)}\right)\left(\sum_{jkh}\beta_{jkh\ell}y_{jk}^{(h)}\right)\left(\sum_{kih}\gamma_{kih\ell}z_{ki}^{(h)}\right)$$
$$=\sum_{h}\left(\sum_{ijk}x_{ij}^{(h)}y_{jk}^{(h)}z_{ki}^{(h)}\right)+O(\lambda).$$

Less formally, the hypothesis is a trilinear algorithm, using L bilinear multiplications to (approximately) compute simultaneously several independent matrix products, of dimension $m_h \times n_h$ times $n_h \times p_h$ (written $< m_h, n_h, p_j >$).

Each of the L bilinear multiplications is a linear combination of x variables, times a linear combination of y variables:

$$M_\ell=\left(\sum_{ijh}\alpha_{ijh\ell}x_{ij}^{(h)}\right)\left(\sum_{jkh}\beta_{jkh\ell}y_{jk}^{(h)}\right)$$

Linear combinations of these products M_ℓ are identically equal (up to errors of order λ) to the desired elements w of the answer matrix:

$$w_{ik}^{(h)} \overset{\text{def}}{=} \sum_{j=1}^{n_h}x_{ij}^{(h)}y_{jk}^{(h)}=\sum_{\ell=1}^{L}\gamma_{kih\ell}M_\ell+O(\lambda).$$

Multiplying both sides by $z_{ki}^{(h)}$, which we view as a dual to $w_{ik}^{(h)}$, and summing, we obtain the single trilinear identity, which contains all the information of the several bilinear identities. The trilinear version is more useful in the present situation in that it reflects the underlying symmetries.

In such a situation, we define the **matrix exponent** obtained from the construction as $\omega = 3\tau$.

3. Strassen's construction

Strassen has found a new relaxation of the ground rules for the construction of the basic algorithm, that is, he has relaxed the hypotheses of the theorem. A key element in his construction is the observation that, using the ability to multiply a pair of $N \times N$ matrices, one can "approximately" (in the λ sense) multiply $(3/4)N^2$ pairs of independent scalars, that is, compute

$$\sum_{\ell=1}^{(3/4)N^2}x_\ell y_\ell z_\ell+O(\lambda) \tag{2}$$

where all the x_ℓ, y_ℓ, z_ℓ are independent. Namely, setting

$$g=\left[\frac{3}{2}(N+1)\right],$$

one obtains

$$\sum_{1\le i,j,k\le N}\left(x_{ij}\lambda^{i^2+2ij}\right)\left(y_{jk}\lambda^{j^2+2j(k-g)}\right)\left(z_{ki}\lambda^{(k-g)^2+2(k-g)i}\right)$$
$$=\sum_{\substack{i+j+k=g\\1\le i,j,k\le N}}x_{ij}y_{jk}z_{ki}+O(\lambda),$$

since the exponent of λ,

$$i^2+2ij+j^2+2j(k-g)+(k-g)^2+2(k-g)i=(i+j+k-g)^2,$$

is zero when $i+j+k=g$ and is positive otherwise. Since any two indices i,j uniquely determine the third $k=g-i-j$, each variable x_{ij} is involved in at most one product. There are about $\left[(3/4)N^2\right]$ triples (i,j,k), $1\le i,j,k\le N$, $i+j+k=g$. Call this construction (*).

Strassen uses the following basic trilinear algorithm, which uses $q+1$ multiplications:

$$\sum_{i=1}^{q}(x_0+\lambda x_i)(y_0+\lambda y_i)(z_i/\lambda)+(x_0)(y_0)\left(-\sum z_i/\lambda\right)$$
$$=\sum_{i=1}^{q}(x_iy_0z_i+x_0y_iz_i)+O(\lambda). \tag{3}$$

This is a viewed as a block inner product:

$$\sum_{i=1}^{q}\left(x_i^1y_0^1z_i+x_0^2y_i^2z_i\right)+O(\lambda).$$

The superscripts denote indices in the block inner product:

$$X^1Y^1Z+X^2Y^2Z,$$

or, dually,

$$W=X^1Y^1+X^2Y^2.$$

One sees the block structure of an inner product, or matrix product of size $< 1,2,1 >$, where the 1×2 block matrix (row vector) X is multiplied by a 2×1 block matrix (column vector) Y to yield a 1×1 block matrix (block scalar) W. We can label x_i and x_0 with different superscripts (put them into different blocks) because they are different variables; similarly y_i and y_0. But the z-variables are involved in both blocks. They are shared. This is the new complication in the basic algorithm. This algorithm does not in itself represent a matrix product.

We examine now the fine structure. The first block, $\sum x_i^1 y_0^1 z_i$, represents a matrix product of size $< q, 1, 1 >$. A $q \times 1$ matrix (column vector) x is multiplied by a 1×1 matrix (scalar) y_0 to yield a $q \times 1$ matrix (column vector) w, which is dual to z. In the second block, $\sum x_0^2 y_i^2 z_i$ represents a matrix product of size $< 1, 1, q >$. A 1×1 matrix (scalar) x_0 is multiplied by a $1 \times q$ matrix (row vector) y to yield a $1 \times q$ matrix (row vector) w. The difficulty comes when we try to add the two blocks. The indices i of w (or z) are "schizophrenic": they don't know whether to behave as row indices or as column indices. Strassen's construction gives a way out of this difficulty.

Take the construction (3) and the two constructions gotten by cyclic permutations of the variables x,y,z, and tensor them together, to get an algorithm requiring $(q+1)^3$ multiplications to compute

$$\sum_{i,j,k=1}^{q} (x_{ij0}^{11} y_{0jk}^{11} z_{i0k}^{11} + x_{ijk}^{21} y_{0jk}^{11} z_{i00}^{12} + x_{ij0}^{11} y_{00k}^{12} z_{ijk}^{21} + x_{ijk}^{21} y_{00k}^{12} z_{ij0}^{22} +$$
$$+ x_{0j0}^{12} y_{ijk}^{21} z_{i0k}^{11} + x_{0jk}^{22} y_{ijk}^{21} z_{i00}^{12} + x_{0j0}^{12} y_{i0k}^{22} z_{ijk}^{21} + x_{0jk}^{22} y_{i0k}^{22} z_{ij0}^{22}) + O(\lambda).$$

This is a block 2×2 matrix product (indicated by the superscripts). Within each block is a smaller matrix product; for example the block $I = 1, J = 1, K = 2$ is the matrix product

$$\sum_{i,j,k=1}^{q} (x_{ij0}^{11} y_{00k}^{12} z_{ijk}^{21})$$

which can be interpreted as a matrix product of size $< q^2, 1, q >$:

$$\sum_{i,j,k=1}^{q} x_{ij,0}^{11} y_{00,k}^{12} z_{k,ij}^{21}.$$

Taking the N^{th} tensor power of this construction, one gets a construction, requiring $(q+1)^{3N}$ multiplications, and producing a block $2^N \times 2^N$ matrix product, each block of which is a matrix product of some size $< m,n,p >$ where $mnp = q^{3N}$. Applying construction (*) to the block structure, one then obtains $(3/4)(2^N)^2$ independent matrix products, each of some size $< m,n,p >$ where $mnp = q^{3N}$. Applying the τ –theorem, one gets

$$\omega \le 3\tau_N, \qquad (q+1)^{3N} = \frac{3}{4} 2^{2N} (q^{3N})^{\tau_N}.$$

Taking N^{th} roots and letting N grow, the (3/4) becomes insignificant, and we have

$$\omega \le 3\tau, \quad (q+1)^3 = 2^2 q^{3\tau}.$$

Letting $q = 5$, Strassen obtains

$$\omega \le \log(6^3/2^2) / \log 5 = \log_5 54 < 2.479.$$

4. The Salem-Spencer Theorem

We will use the following theorem of Salem and Spencer [SS]; see also [Beh].

Theorem (Salem and Spencer): *Given $\varepsilon > 0$, there exists M_ε such that for any $M > M_\varepsilon$, there is a set B of $M' > M^{1-\varepsilon}$ distinct integers*

$$0 < b_1 < b_2 < \cdots < b_{M'} < \frac{M}{2}$$

with no three terms in an arithmetic progression:

$$\text{for } b_i, b_j, b_k \in B, \qquad b_i + b_j = 2b_k \quad \text{iff} \quad b_i = b_j = b_k.$$

We will be considering the ring Z_M of integers modulo M, where M is odd. Because the numbers in the Salem-Spencer set satisfy $0 < b_i < M/2$, no three can form an arithmetic progression mod M:

$$\text{for } b_i, b_j, b_k \in B, \qquad b_i + b_j \equiv 2b_k \bmod M \quad \text{iff} \quad b_i = b_j = b_k. \quad (4)$$

5. New Construction: Easy Case

Start with a modification of the basic algorithm (3). We use $q+2$ multiplications:

$$\sum_{i=1}^{q} \lambda^{-2}(x_0 + \lambda x_i)(y_0 + \lambda y_i)(z_0 + \lambda z_i)$$
$$- \lambda^{-3}\left(x_0 + \lambda^2 \sum x_i\right)\left(y_0 + \lambda^2 \sum y_i\right)\left(z_0 + \lambda^2 \sum z_i\right)$$
$$+ \left(\lambda^{-3} - q\lambda^{-2}\right)(x_0)(y_0)(z_0) \qquad (5)$$
$$= \sum_{i=1}^{q} (x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0) + O(\lambda).$$

We have brought the factors λ^{-3}, $\left(\lambda^{-3} - q\lambda^{-2}\right)$ outside in order to reflect the symmetry.

The x-variables break into two blocks: $\{x_0\}$ and $\{x_1, \ldots, x_q\}$. The latter will be called x_i. Similarly the y- and z- variables break into blocks. We will refer to a *block* of x-variables as X_0 or X_i, (similarly Y and Z), and when we *zero* a block X (resp. Y, Z) we will set to zero all x- (resp. y-, z-) variables with the given index pattern.

Fix $\varepsilon > 0$. Select N large enough so that the M defined below will exceed M_ε.

Take the $3N^{th}$ tensor power of the construction (5). Set to zero all variables except those with exactly N indices of 0 and exactly 2N indices in $\{1, 2, \ldots, q\}$.

Set $M = 2\binom{2N}{N} + 1$. Select random integers $0 \le w_j < M$, $j = 0,1,\ldots,3N$. For each variable (x, y or z) or block (X, Y or Z) compute a *hash* as follows. For each of the 3N index positions j, define

$$\delta_j(I) = 0 \text{ if } j^{th} \text{ position of I is 0}$$
$$\delta_j(I) = 1 \text{ if } j^{th} \text{ position of I is } i \in \{1, 2, \ldots, q\}.$$

Define

$$b_x(I) \equiv \sum_{j=1}^{3N} \delta_j(I) w_j \pmod M$$
$$b_y(J) \equiv w_0 + \sum_{j=1}^{3N} \delta_j(J) w_j \pmod M$$
$$b_z(K) \equiv (w_0 + \sum_{j=1}^{3N} (2 - \delta_j(K)) w_j)/2 \pmod M.$$

Since M is odd, division by 2 is well defined. Since $b_x(I)$ depends only on the block indices of x, we may define b_X, b_Y, b_Z in the obvious way.

Notice that for any variables x_I, y_J, z_K whose product $x_I y_J z_K$ appears in the computed trilinear form, we have

$$b_x(I) + b_y(J) - 2b_z(K) \equiv 0 \mod M. \tag{6}$$

This follows by considering the contribution of each w_j, noticing that in the basic construction

$$\delta_j(I) + \delta_j(J) + \delta_j(K) = 2$$

for each of the three terms $x_0 y_i z_j$, $x_i y_0 z_j$, $x_i y_j z_0$.

Set to zero all variables x_I for which $b_x(I)$ is not in B; similarly y_J and z_K. Then for any nonzero term $x_I y_J z_K$ remaining in our construction, we have

$$b_x(I) + b_y(J) \equiv 2b_z(K) \mod M, \qquad b_x(I), b_y(J), b_z(K) \in B,$$

so that

$$b_x(I) = b_y(J) = b_z(K).$$

For each element $b \in B$ in the Salem-Spencer set, make a list of triples (X_I, Y_J, Z_K) of compatible nonzero blocks, with $b_X(I) = b_Y(J) = b_Z(K) = b$. (A *block* X_I is the set of q^{2N} variables x with nonzero indices in 2N specified places. A *nonzero block* is one which has not yet been set to zero. Blocks X_I, Y_J, Z_K are *compatible* if the locations of their zero indices are pairwise disjoint.) For each triple (X_I, Y_J, Z_K) on the list, if it shares a block (say Z_K) with another triple $(X_{I'}, Y_{J'}, Z_K)$ occurring earlier in the list, we set to zero one of the other blocks (say Y_J), and thus eliminate this triple. (If each of X_I, Y_J, Z_K is shared with previous triples, we will end up eliminating at least two triples by zeroing one block of variables.)

For a fixed element $b \in B$, the expected number of triples in the list, before pruning, is

$$\binom{3N}{N,N,N} M^{-2}.$$

Here $\binom{3N}{N,N,N}$ represents the number of compatible triples (X_I, Y_J, Z_K) and the M^{-2} represents the probability of the (independent) events $b_X(I) = b$ and $b_Y(J) = b$. (If both hold, then $b_Z(K) = b$ follows.) That is, for fixed blocks X_I, Y_J, and fixed integer b mod M, if we randomize the values $w_0, w_1, \ldots, w_{3N}$, then

$$\begin{aligned} &\text{Prob}\{b_X(I) = b_Y(J) = b\} \\ &= \text{Prob}\{b_X(I) = b\}\text{Prob}\{b_Y(J) = b\} \\ &= M^{-1}M^{-1} = M^{-2}, \end{aligned}$$

since the sums $b_X(I)$ and $b_Y(J)$ involve different random variables. The expected number of compatible triples (X_I, Y_J, Z_K) with $b_X(I) = b_Y(J) = b_Z(K) = b$ is the sum of these probabilities (M^{-2}) over the $\binom{3N}{N,N,N}$ possible triples. We do not need independence among triples, since the expected value of a sum of random variables is the sum of their expected values, regardless of independence.

The expected number of unordered pairs (X_I, Y_J, Z_K), $(X_{I'}, Y_{J'}, Z_K)$ sharing a Z-block is

$$\frac{1}{2}\binom{3N}{N,N,N}\left(\binom{2N}{N,N} - 1\right)M^{-3}$$

Again $\binom{3N}{N,N,N}$ counts the compatible triples (X_I, Y_J, Z_K). The binomial coefficient $\binom{2N}{N,N} - 1$ counts the blocks $Y_{J'}$ compatible with Z_K (other than Y_J itself). The factor 1/2 eliminates duplicate entries $((X_I, Y_J, Z_K), (X_{I'}, Y_{J'}, Z_K))$ and $((X_{I'}, Y_{J'}, Z_K), (X_I, Y_J, Z_K))$. The factor M^{-3} is the probability of the independent events $b_Z(K) = b$, $b_Y(J') = b$, $b_X(I) = b$. They are independent even if the indices J' and I are equal, because of the presence of the random variable w_0.

The expected number of pairs of triples sharing Y_J, or sharing X_I, is the same.

Suppose we eliminate a block (Y_J) because of a pair of triples $((X_I, Y_J, Z_K), (X_{I'}, Y_{J'}, Z_K))$ sharing a Z-block. If L triples (not yet eliminated) share this Y-block, then setting Y_J to zero eliminates these L triples, while eliminating at least $\binom{L}{2} + 1$ pairs, namely all those sharing Y_J, and at least the pair sharing Z_K. Since $\binom{L}{2} + 1 \geq L$, we eliminate at least as many pairs as triples. Thus:

Lemma. *The expected number of triples remaining on each list, after pruning, is at least*

$$\begin{aligned} &\binom{3N}{N,N,N}M^{-2} - \frac{3}{2}\binom{3N}{N,N,N}\left[\binom{2N}{N,N} - 1\right]M^{-3} \\ &\qquad \geq \frac{1}{4}\binom{3N}{N,N,N}M^{-2}. \end{aligned} \tag{7}$$

The expected number of triples remaining on all lists, after pruning, is at least

$$H \stackrel{\text{def}}{=} \frac{1}{4}M'\binom{3N}{N,N,N}M^{-2}. \tag{8}$$

This expectation H is an average over the choices of w_j. There is a choice of w_j which achieves at least H; fix such a choice.

Our algorithm computes at least H block scalar multiplications $X_I Y_J Z_K$. The fine structure of each block scalar multiplication is in fact a matrix product of size

$$< q^N, q^N, q^N >,$$

and all the variables are disjoint (by the Salem-Spencer property). From the τ-theorem we obtain

$$\omega \leq 3\tau_N, \quad (q+2)^{3N} \geq \frac{1}{4}M'\binom{3N}{N,N,N}M^{-2}q^{3N\tau_N}.$$

Use Stirling's approximation to obtain

$$(q+2)^{3N} \geq cN^{-1/2+\varepsilon}3^{3N}2^{-2N(1+\varepsilon)}q^{3N\tau_N},$$

where c is a constant. Letting ε go to zero and N to infinity, and taking N^{th} roots, we obtain

$$(q+2)^3 \geq \frac{27}{4} q^{3\tau}$$

$$\omega \leq 3\tau \leq \log_q\left(\frac{4(q+2)^3}{27}\right).$$

Setting $q = 8$ we obtain

$$\omega \leq \log_8\left(\frac{4000}{27}\right) < 2.40364. \tag{9}$$

6. New Construction: Complicated Version.

In fact we can do better.

The following construction is somewhat more cumbersome because of the extra terms involved, but it uses exactly the same ideas to yield an exponent of 2.388.

Begin with the basic algorithm:

$$\begin{aligned}
&\sum_{i=1}^{q} \lambda^{-2}(x_0 + \lambda x_i)(y_0 + \lambda y_i)(z_0 + \lambda z_i) \\
&- \lambda^{-3}\left(x_0 + \lambda^2 \sum x_i\right)\left(y_0 + \lambda^2 \sum y_i\right)\left(z_0 + \lambda^2 \sum z_i\right) \\
&+ \left[\lambda^{-3} - q\lambda^{-2}\right]\left(x_0 + \lambda^3 x_{q+1}\right)\left(y_0 + \lambda^3 y_{q+1}\right)\left(z_0 + \lambda^3 z_{q+1}\right) \qquad (10) \\
&= \sum_{i=1}^{q} (x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0) + x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + x_{q+1} y_0 z_0 \\
&\qquad + O(\lambda).
\end{aligned}$$

The indices now form three classes: $\{0\}$, $\{q+1\}$, and $\{1, 2, \ldots, q\}$, which will again be denoted i.

Take the $3N^{th}$ power of this construction. Set $L = [\beta N]$ where β will be determined later. Set to zero all variables unless they have exactly L indices of $q+1$, $N+L$ indices of 0, and $2N-2L$ other indices.

Set

$$M = 2\binom{N+L}{L,L,N-L}\binom{2N-2L}{N-L,N-L} + 1.$$

Define

$\delta_j(I) = 0$ if j^{th} position of I is 0

$\delta_j(I) = 1$ if j^{th} position of I is $i \in \{1, 2, \ldots, q\}$

$\delta_j(I) = 2$ if j^{th} position of I is $q+1$.

As before, define $b_x(I)$, $b_y(J)$, $b_z(K)$, and set to zero any variable with $b_x(I)$ (resp. $b_y(J)$, $b_z(K)$) not in the Salem-Spencer set. For each b in the Salem-Spencer set, make a list of triples (X_I, Y_J, Z_K) of blocks, with $b_X(I) = b_Y(J) = b_Z(K) = b$, and eliminate entries with duplicated blocks.

For a given block Z_K, the number of pairs of blocks (X_I, Y_J) compatible with Z_K is

$$\binom{N+L}{L,L,N-L}\binom{2N-2L}{N-L,N-L},$$

since the $N+L$ indices of 0 in K correspond to L instances of (0 in I, $q+1$ in J), L instances of ($q+1$ in I, 0 in J), and N-L instances of (i in I, i in J), while the $2N-2L$ indices of i in K correspond to N-L instances of (i in I, 0 in J) and N-L instances of (0 in I, i in J). Since M is twice this size, the elimination of duplicates proceeds as before and leaves a constant fraction of the triples intact.

We have M' lists, each with

$$\frac{1}{4}\binom{3N}{L,L,L,N-L,N-L,N-L} M^{-2}$$

entries, all having independent variables. (The multinomial coefficient indicates that there are L instances of $(q+1,0,0)$ as x- y- z-indices, L of $(0,q+1,0)$, L of $(0,0,q+1)$, N-L of $(i,i,0)$, etc.). Each entry corresponds to a matrix product of size

$$< q^{N-L}, q^{N-L}, q^{N-L} > .$$

Thus our equation is

$$\begin{aligned}
(q+2)^{3N} &\geq \frac{1}{4} M' \binom{3N}{L,L,L,N-L,N-L,N-L} M^{-2} q^{3(N-L)\tau_N} \\
&\simeq cN^{(-1+3\varepsilon/2)N} \left[\frac{27}{\beta^\beta (1+\beta)^{1+\beta} (2-2\beta)^{2-2\beta}}\right]^N q^{3N(1-\beta)\tau_{NC}\varepsilon N}.
\end{aligned}$$

Letting ε tend to zero and N to infinity, and taking N^{th} roots, we get

$$(q+2)^3 \geq \frac{27}{\beta^\beta (1+\beta)^{1+\beta} (2-2\beta)^{2-2\beta}} q^{3(1-\beta)\tau}.$$

For $q = 6$, $\beta = 0.048$, we find

$$\omega \leq 3\tau < 2.38719.$$

A somewhat more complicated construction yields $\omega < 2.376$. Details will be in the full paper.

7. Remarks

The most exciting aspect of Strassen's new approach is that it eliminates a major barrier to proving $\omega = 2$. Namely, if one uses a *fixed, finite* basic algorithm in the hypothesis of Schönhage's τ-theorem, then L, the number of multiplications, must strictly exceed either #x, the number of x-variables, or #y or #z [CW], because the basic algorithm is a matrix multiplication algorithm. In this case, the estimate of ω obtained must be strictly larger than 2. But with Strassen's approach, the basic algorithm is not a matrix multiplication algorithm and does not suffer from this restriction; in fact, the algorithm used to obtain $\omega < 2.388$ has $\#x = \#y = \#z = L$. By the time one gets to the τ-theorem, having taken the tensor power and applied the Salem-Spencer trick, one is not dealing with a fixed algorithm (and fixed values of L and #x) but with a family of algorithms (and a sequence of values L and #x), and the restriction $\#x < L$ is unimportant, since the ratio $(\log \#x)/(\log L)$ can approach 1. Thus we can search for more starting algorithms of the Strassen variety, with $\#x = \#y = \#z = L$, with the hope that one of them might yield the elusive $\omega = 2$.

An open problem is to find a characteristic two analogue of the Salem-Spencer theorem: subsets A, B, C of the vector space $(\mathbf{Z}/2)^n$ such that each $a \in A$ (resp. $b \in B, c \in C$) is contained in exactly one triple

$$(a \in A, b \in B, c \in C), \quad a + b + c = 0,$$

and with $|A| > (2^n)^{1-\varepsilon}$. This would allow the extension of the present techniques to a larger class of basic algorithms.

References.

[Beh] F. A. Behrend, "On sets of integers which contain no three terms in arithmetical progression," *Proc. Nat. Acad. Sci. USA* **32** (1946) 331-332; *MR* **8**, 317.

[CW] D. Coppersmith and S. Winograd, "On the Asymptotic Complexity of Matrix Multiplication," *SIAM Journal on Computing*, Vol. 11, No. 3, August 1982, pp. 472-492.

[Pan] V. Pan, "How to Multiply Matrices Faster," Springer Lecture Notes in Computer Science, vol 179.

[Sch] A. Schönhage, "Partial and Total Matrix Multiplication," *SIAM J. on Computing*, 10, 3, 434-456.

[SS] R. Salem and D. C. Spencer, "On sets of integers which contain no three terms in arithmetical progression," *Proc. Nat. Acad. Sci. USA* **28** (1942) 561-563.

[Str] V. Strassen, "The Asymptotic Spectrum of Tensors and the Exponent of Matrix Multiplication," 1986 FOCS, pp. 49-54; also "Relative bilinear complexity and matrix multiplication," preprint.