On the Complexity of Bilinear Forms*

Noam Nisan †
Institute of Computer Science,
Hebrew University of Jerusalem, Israel.

Avi Wigderson ‡
Institute of Computer Science,
Hebrew University of Jerusalem, Israel.

Abstract

This paper provides some new lower and upper bounds on computing bilinear forms by arithmetic circuits. The complexity measures considered are circuit size, formula size and time-space trade-offs.

1 Introduction

This paper is concerned with the arithmetic complexity of bilinear forms. Such a function $f(x, y)$ over vectors of variables x, y is naturally defined by a matrix M with $f(x, y) = yMx$. Note that the same M also defines a set of linear forms in the x variables, namely Mx. This paper contains several new bounds on the complexity of computing such functions. The complexity measures of interest are circuit and formula size, as well as time-space trade-offs.

In the second section of the paper we are interested in general computation over arbitrary fields, and focus on the smallest product TS of time and space required. Our main result here is proving that for every field and every n there are $n \times n$ matrices M for which computing the linear forms Mx requires nearly quadratic TS, while the bilinear form yMx can be computed in nearly linear TS. The proof uses small, depth 2 superconcentrators to construct matrices M in which all minors have high rank.

The motivation for this result comes for a question of Borodin [Bo94], who asked if there is an space-efficient implementation of the Baur-Strassen theorem [BS82]. This remarkable theorem asserts that the time (or size) it takes to compute all n partial derivatives (one for each variable) of a polynomial f is only larger by a constant factor then the time to compute f itself. Thus time lower bounds on computing n functions can (and do) yield lower bounds for computing a single function.

One frustration of (both arithmetic and Boolean) circuit complexity is that we have quadratic lower bounds on TS for computing n functions (even n linear functions) on n variables, but no similar bounds are known for computing a single one. Observe that the set of linear functions Mx is the set of derivatives according to the y variables of the single bilinear form yMx. Thus an analog of the Baur-Srassen construction, which is efficient in the TS measure, would resolve that particular frustration. However, the result stated above shows that no such analog can exist. In fact, in contrast to time complexity, the gap in time space product between computing a function and computing all its derivatives is almost n, as bad as possible.

*This work was done while the authors visited BRICS at the University of Aarhus, Denmark

†This work was supported by USA-Israel BSF grant 92-00043 and by a Wolfeson research award administered by the Israeli Academy of Sciences.

‡This work was supported by USA-Israel BSF grant 92-00106 and by a Wolfeson research award administered by the Israeli Academy of Sciences.

0

Despite the above, we can give some weak lower bounds on time-space trade-offs, for the same bilinear forms used to exhibit the gap above. We show that they require $2^{T/n}TS = \Omega(n^2)$. This follows a simple adaptation of the Alon-Maass technique ([AM88]), originally devised for oblivious branching programs.

In section 3 we switch gears and move to discuss circuits over the real or complex fields, which can use only bounded constants in the computation. This may be a severe restriction. However, considering this model has several motivations. First and foremost, we have no superlinear lower bounds even on formula size of bilinear forms in the general model, so we might as well start with this restricted one. Second, it was suggested as a natural model (for computing linear functions) by Morgenstern [Mo73] and Chazelle [Ch94]. Morgenstern observes that natural algorithms like FFT use only small constants, and actually proves that under this restriction the $n \log n$ time bound it is optimal. Chazelle argues that for algorithms in comutational geometry, the finite representation of numbers is essenntially equivalent to bounded coefficients, and then proves similar $n \log n$ lower bounds in this model for geometric range-query problems.

The above mentioned results apply only to computing many (linear) functions. We provide lower bounds on the circuit and formula size of single explicit bilinear form, e.g. the one associated with any Hadamard or Discrete Fourier Transform matrices.

We first obtain an $\Omega(n \log n)$ circuit lower bound. It follows a simple observation that the afore mentioned Baur-Strassen construction can be carried over in the restricted model, and thus the above results of, e.g., Morgenstern [Mo73] imply it directly.

We then obtain an $n^{5/4}$ lower bound on the formula size for these bilinear forms, as well as $n^{1+\epsilon}$ lower bounds for other explicit matrices, arising from projective geometries and other block designs. The proof uses the Hoffmann-Wielandt Inequality, which bounds the effect of perturbations on the eigenvalues of a matrix.

2 Definitions and Basic Facts

2.1 Circuits

Let F be a field, and $X = \{x_1, x_2, \cdots, x_n\}$ be indeterminates. A circuit over F with variables X is a directed acyclic graph whose input nodes are labeled with the elements of X. Every internal node of the circuit computes either a $+$ gate or a $\times$ gate. A $+$ gate, with arbitrary coefficients from F on its input wires, computes that linear combination of its inputs. A $\times$ gate computes the product of its inputs. (Note that we explicitely allow unbounded fan-in.) Thus every node computes a polynomial in $F[X]$. A polynomial $f \in F[X]$ is said to be computed by a circuit A if it is computed by one of the nodes of A.

The most basic complexity measure of a function is its circuit size. The **size** of a circuit is the number of edges it has. For any polynomial f we denote by $C(f)$ the size of the smallest circuit computing f. This notion is extended to computing several polynomials: $C(f_1, f_2, \cdots, f_m)$ is the size of the smallest circuit computing them all.

Note: Technically, for the measure C and other measures we will define, we will distinguish the field F over which the circuit is computing by the notation C_F. However, as F will almost always be clear from the context, we shall ususally omit this subscript.

Another fundamental measure is the **depth** of a circuit A, denoted $D(A)$, which is the length of the longest directed path in A. The depth complexity of a polynomial f, $D(f)$, is the smallest depth of a circuit computing f (and similarly for computing many polynomials).

Two restricted notions of circuits give rise to two other complexity measures: formula size and time-space tradeoffs.

A formula is a circuit in which every node has at most one outgoing edge (fan-out 1), thus the graph is a tree. The formula size of f, $L(f)$ is the size of the smallest formula computing f.

Any circuit can be "implemented" by a straight-line program over the input variables and a set of "registers" variables. Such a program is a sequence of instructions $X \leftarrow Y \circ Z$, with $\circ \in$

$\{+, \times\}$ [1], X a register, and each of Y, Z being either a register of input variable. Such a program A computes all polynomials that occupy a register at some point in the program. Its **space** $S(A)$ is the total number of different register used, and its **time** is the total number of instructions.

For any polynomial f denote by $TS(f)$ the smallest value of the product of $T(A) \times S(A)$ over all programs A computing f. For several functions $TS(f_1, \cdots, f_m)$ is similarly defined.

Two simple relations between these measure are:

Proposition 1

- *For every f, $C(f) \leq TS(f) \leq C(f)^2$.*
- *If f is computed by a formula A, then $TS(f) \leq C(A)D(A)$.*

2.2 Derivatives

Let $f \in F[X]$ be a polynomial. Denote by Δf the set of n partial derivatives of f according to each of the variables in X. This is a set of polynomials in $F[X]$ and it is natural to relate its circuit complexity to the of f. It is not hard to see that each derivative can be computed with size $C(f)$, so for every f, $C(\Delta f) \leq nC(f)$. A remarkable theorem of Baur and Strassen [BS82] (see a simple proof by Morgenstern [Mo85]) says that there is no need to lose the factor of n.

Theorem 1 *[BS82] For every f, $C(\Delta f) \leq 3C(f)$.*

This theorem was used in [BS82] to convert lower bounds for computing a set of polynomials into a lower bound for a single function. For example, Strassen's lower bound [St73] $C(x_1^n, x_2^n, \cdots, x_n^n) = \Omega(n \log n)$ immediately implies that $C(x_1^n + x_2^n + \cdots + x_n^n) = \Omega(n \log n)$.

2.3 Linear functions

Let M be an $n \times n$ matrix with entries in F. We associate with M the vector of n linear functions l_M on variables X defined by the matrix vector product $l_M(\bar{x}) = M\bar{x}$, where $\bar{x} = (x_1, x_2, \cdots, x_n)$.

[1] Again, + has arbitrary coefficients

Clearly, every M can be computed by **linear** circuits, i.e. circuits that do not use multiplication. Let $C^l(l_M)$ denote the size of the smallest linear circuit computing l_M. While it is open whether using multiplications can be significantly useful over finite fiels, it is easy to see that over infinite fields it cannot.

Proposition 2 *For every infinite field F, and every matrix M, $C^l(l_M) = C(l_M)$.*

While simple counting shows that for most matrices M, $C(l_M)$ is nearly quadratic in n, it is a major open problem to exhibit an explicit infinite family of matrices for which $C(l_M)$ is superlinear. However, for the TS measure such bounds can be obtained, via the elegant sufficient condition of Valiant [Va76], (see also Tompa [To80]).

Theorem 2 *[Va76]*

- *Let M be an $n \times n$ matrix in which every square submatrix is nonsingular. Then $TS(l_M) = \Omega(n^2)$.*
- *Let M be an $m \times n$ matrix with $m \leq n$. Assume that for every $s \leq m$, every $s \times (n - m/2)$ submatrix of M has rank at least $s/2$. Then $TS(l_M) = \Omega(m^2)$.*

We conclude by remarking that there are many explicit families of matrices with all minors nonsingular, and thus quadratic lower bounds on $TS(l_M)$ are available. Examples include all $n \times n$ matrices M of the form:

- $M_{ij} = \omega^{ij}$, with ω a pth root of unity, and $p > n$ a prime.
- $M_{ij} = 1/(i + j)$.

2.4 Bilinear functions

Again let M be a matrix over F. We associate with it the bilinear form b_M over the $2n$, variables $\{x_1, \cdots, x_n, y_1, \cdots, y_n\}$ by defining $b_M(\bar{x}, \bar{y}) = \bar{y}M\bar{x}$.

A **bilinear** circuit on variables $\{x_1, \cdots, x_n, y_1, \cdots, y_n\}$ is one in which every multiplication gate has exactly two inputs, one of which is a linear function

of the x variables and the other a linear function of the y's. Again it is clear that every bilinear function is computed by such a circuit, and we denote by $C^b(b_M)$ the size of the smallest bilinear circuit for b_M.

Again, it is not known if significant savings are possible in general using arbitrary circuits for computing bilinear forms, but for infinite fields they don't.

Proposition 3 *For every infinite field F and every matrix M, $C^b(b_M) = O(C(b_M))$.*

Now observe that for any matrix M, the set of linear functions l_M is a subset of the derivatives Δb_M of the bilinear function b_M. Thus, an immediate corollary to Theorem 1 is

Corollary 1 *For every matrix M over any field, $C(l_M) \leq 3C(b_M)$.*

In a similar way to bilinear fcircuits we define bilinear formulae, and $L^b(b_M)$. It is interesting to note that no analog of proposition 3 is known for this measure. Note that bilinear formulae have a very simple structure - without loss of generality they have depth 3: a sum of (t, say) products of two linear forms. Formally, any such formula for b_M gives the decomposition:

Equation 1 $M = \sum_{i=1}^{t} M_i$, *with* $M_i = \bar{u}^i \otimes \bar{v}^i$.

Here $\otimes$ denotes exterior product and thus each M_i has rank 1. The size of the formula with this decomposition is simply the total number of nonzero entries in all vectors $\bar{u}^i, \bar{v}^i$. Equivalently, setting U to be the $n \times t$ matrix whose columns are $\bar{u}^i$, and V the $t \times n$ matrix whose rows are $\bar{v}^i$, we have

Equation 2 $M = UV$

Again, the size of the formula giving this decomposition is the total number of nonzero entries in U and V.

2.5 Superconcentrators

For a dircted acyclic graph $G(V, E)$ denote by I (resp. O) the subset of vertices V with indegree (resp. outdegree) 0, and call them inputs (resp. outputs). The size of G, $C(G)$, is its number of edges, and the depth of G, $D(G)$ is the length of the longest directed path in G (necessarily between from an input to an output).

G is called an **n-superconcentrator** if $|I| = |O| = n$, and for every k and every two subsets $I' \subseteq I$ and $O' \subseteq O$ with $|I'| = |O'| = k$ there are k vertex disjoint paths from I' to O' in G.

Let $c_d(n)$ denote the size of the of the smallest n-superconcentrator of depth d. Determining the functions c_d for various values of d, as well as finding explicit small and shallow superconcentrators has been a major object of study [Pi78, Pi82, DDPW]. We shall need the following two upper bounds on the size of superconcentrators of depth 2, the first being nonconstructive and the second explicit.

Theorem 3 *[Pi82]* $c_2(n) = O(n(\log n)^2)$.

Theorem 4 *[WZ93] There is a polynomial time algorithm which for every n outputs an n-superconcentrator of depth 2 and size $n^{1+o(1)}$. (The best bound on the $o(1)$ term is actually $(\log n)^{-1/2}$ [SZ94].)*

In any depth 2 graph G, the vertices decompose to three sets, $V = I \cup R \cup O$, which are resp. the inputs, middle vertices, and outputs (for any direct edge between I and O we can insert a new vertex to split it in two, without effecting the size by more than a factor of 2). Let $|I| = |O| = n$ and $|R| = t$. Any assignment of weights w (from a field F) to the edges of G (i.e. $w : E \to F$), naturally defines a matrix over F as follows. First, replace every non-edge of G by an edge of weight 0. Then, for every $i \leq t$ define two n-vectors $\bar{u}^i$ (resp. $\bar{v}^i$) by $\bar{u}^i_k$ (resp. $\bar{v}^i_k$) is the weight on the edge (I_k, R_i) (resp. (R_i, O_j). Define $M_i = \bar{u}^i \otimes \bar{v}^i$, and $M = M(G, w) = \sum_{i=1}^{t} M_i$. This gives a decomposition as in Equation 1, and thus the fomula size upper bound:

Lemma 1 *For every depth 2 graph G and weights w, $L^b(b_{M(G,w)}) \leq |E|$.*

2.6 Schwartz's Lemma

An extremely useful lemma regarding the zeros of multivariate polynomials was proved by Schwartz [Sc80].

Theorem 5 *[Sc80] Let $f \in F[X]$ be a polynomial of total degree d, which is not identically zero. Let H be any finite subset of F. Let $w : X \to H$ be a random function, assigning to each variable an independent random, uniformly distributed element of F. Then $Pr[f(w) = 0] \leq d/|H|$.*

2.7 Algebra

For this section our field is the reals R. For $\bar{v} \in R^n$, let

- $|\bar{v}| = \sum_{i=1}^{n} |v_i|$ be the l_1 norm of $\bar{v}$,
- $||\bar{v}|| = (\sum_{i=1}^{n} v_i^2)^{\frac{1}{2}}$ be the l_2 norm of $\bar{v}$,
- $|\bar{v}|_\infty = \max_i |v_i|$ be the l_∞ norm of $\bar{v}$.

Let A be an $n \times m$ matrix. $|A|$, $|A|_\infty$ and $||A||$ will denote the norms of A viewed as an $n \cdot m$ vector. Assume A is symmetric. Then A is diagonalizable, i.e. there is an unitary matrix P such that $P^t AP = \bar{\lambda}(A)I$, where $\bar{\lambda}(A) = (\lambda_1, \lambda_2, \cdots, \lambda_n)$ is the vector of eigenvalues of A, and we assume that $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$.

An important relation between the effect of perturbing a matrix on its eigenvalues is the Hoffman-Wielandt inequality (see e.g. [Wi65]).

Theorem 6 *(Hoffman-Weilandt Inequality) Let A, B be matrices with $A - B$ symmetric. Then*

$$||\lambda(A) - \lambda(B)|| \leq ||A - B||$$

Some $n \times n$ matrices we shall see later are:

- H — the Sylvester matrix. Here $n = 2^k$, and considering the indices as subsets of $[k]$, $H_{ij} = (-1)^{i \cap j}$. For all $i \in [n]$, $|\lambda_i(H)| = \sqrt{n}$
- DFT — the Discrete Fourier Transform. Here ω is an nth primitive root of unity, and $DFT_{ij} = \omega^{ij}$. For all $i \in [n]$, $|\lambda_i(DFT)| = \sqrt{n}$
- PP — the Projective Plane (see [Ha86]). Here $n = p^2 + p + 1$ for some prime p. PP is a Boolean matrix, with exactly $p + 1$ 1's in every row and column. For all $i \in [n]$, $|\lambda_i(PP)| = p \simeq n^{1/4}$

3 Times-Space Tradeoffs

In this section we show that there is no efficient analog to Theorem 1 of Baur and Strassen. In fact, the gap between $TS(f)$ and $TS(\Delta f)$ can be nearly linear - the worst possible. We exhinbit families of matrices M for which $TS(l_M)$ is almost quadratic, while $TS(b_M)$ is nearly linear. The best gap we can achieve will depend on how large entries in M are allowed to be (and hence on the size of the base field F), as well as how explicit the matrices are.

Lemma 2 *Let $G(V, E)$ be any n-superconcentrator. Let F be any field, and H any finite subset of it. Let $w : E \to H$ be chosen uniformly at random. Then*

1. *The probability that all square minors in $M(G, w)$ are nonsingular is at least $1 - 2n2^{2n}/|H|$.*
2. *The probability that all square minors in $M(G, w)$ have at least half the maximum rank is at least $1 - n^3/|H|$.*

Proof: Let $V = I \cup R \cup O$ as before, the inpus, middle and output vertices of G, resp. Fix s and an $s \times s$ minor N of $M(G, w)$, say on rows $I' \subseteq I$ and columns $O' \subseteq O$. Since G is a superconcentrator, there are s vertex disjoint paths from I' to O'. By renaming, we can assume wlog that I' (resp. O') are the first s vertices of I (resp. O), and moreover that the disjoint paths are formed by the edges (I_k, R_k) and (R_k, O_k) for all $1 \leq k \leq s$.

Fix the weights assigned by w to all other edges in E in an arbitrary way, and let a_k (resp. b_k) be variables (indeterminates) whose value will be the weight assigned to (I_k, R_k) (resp. (R_k, O_k)). Call this set of $2s$ variables Z. Thus our new probability space now assignes independent random weights from H to Z.

Let $\hat{N}$ be our minor N before assigning values to Z so the entries $\hat{N}_{k,m}$ are polynomials over these variables. ¿From the definition of M in subsection 2.5 the following should be clear.

Claim 1
- *For every $1 \leq k \neq m \leq s$, $\hat{N}_{k,m}$ is a linear function.*

- *For every k, $\hat{N}_{k,k}$ is a quadratic polynomial with exactly one monomial of degree 2 having a nonzero coefficient: the monomial $a_k b_k$.*

Now we are ready to prove the two parts of the theorem, starting with part (1). Consider $det(\hat{N})$, again as a polynomial in $F[Z]$. Clearly it is not identically zero, as the monomial $\Pi_{k=1}^{s} a_k b_k$ has a nonzero coefficient, and is of degree $2n$. By Theorem 5 we know that $Pr[det(N) = 0] = Pr[det(\hat{N}(w)) = 0] \leq 2n/|H|$.

Since the bound above holds conditioned on every fixed choice of values w assignes outside Z, it also holds over the whole probability space for this fixed minor N. Thus the final bound follows from the fact that there are 2^{2n} minors.

Proving part (2) is slightly more complicated. Fix N, and values outside Z again. We shall assign the values to a_k, b_k for a single k at a time, going from 1 to s. If N_k denotes the $k \times k$ principal minor of N (with $N_0 = 1$), we will prove:

Claim 2 *$Pr[rk(N_k) = rk(N_{k-1})] \leq 2/|H|$, where the probability is taken for every fixed choice of $a_1, b_1, \cdots, a_{k-1}, b_{k-1}$, and a random choice of a_k, b_k.*

Let us see why this implies part (2). Note that the probability that the event in the claim happens at least $s/2$ times (which will cause $rk(N) \leq s/2$) is at most $2^s(2/|H|)^{s/2}$. There are at most n^{2s} minors of size s, and it is easy to calculate the validity of the bound in (2) by summing this estimate over all $s \leq n$.

To prove the claim, assume $rk(N_{k-1}) = t$, and let P be any $t \times t$ nonsingular minor of N_{k-1} (taking $Q = N_0$ if $t = 0$. Add to P the (appropriate entries of) kth row and column of N (resp. $\hat{N}$), to create the matrix Q (resp. matrix of variables $\hat{Q}$). It is clear that $Pr[rk(N_k) = t] = Pr[rk(Q) = t] = Pr[det(\hat{Q}) = 0]$. But observe that $det(\hat{Q})$ is a quadratic polynomial in the two variables a_k, b_k. Moreover the only degree 2 monomial ($a_k b_k$) has coefficient $det(P)$ whcih is nonzero by the choice of P, so the claim follows again form Schwartz's lemma.

□

¿From Lemma 2, together with Theorem 2, Theorem 3, Lemma 1 and Proposition 1 we deduce the main result of this section, namely a nearly linear gap between $TS(f)$ and $TS(\Delta f)$.

Theorem 7 *For every n, if $|F| > n^4$, there is an $n \times n$ matrix M with entries in F satisfying:*

- *$L^b(b_M) = O(n(\log n)^2)$, and hence $TS(b_M) = O(n(\log n)^2)$.*
- *$TS(l_M) = \Omega(n^2)$*

The last theorem in this section achieves a similar gap can be obtained for small finite fields. We demonstarte it only for $F = GF(2)$.

Theorem 8 *For every n, there is an $n \times n$ binary matrix M' so that the following bounds hold for computations both over $F = GF(2)$ and $F = Q$.*

- *$L^b(b_{M'}) = O(n(\log n)^2)$, and hence $TS(b_{M'}) = O(n(\log n)^2)$.*
- *$TS(l_{M'}) = \Omega(n^2/(\log n)^2)$*

Proof: For any m set $t = 4 \log m$, $n = mt$, and $G(V, E)$ a depth 2 m-superconcentrator. By Theorem 2 there is an assignement $w : E \to GF(2^t)$, such that for every $s \leq m$, every $s \times s$ minor of $M = M(G, w)$ has rank $\geq s/2$ over the field $GF(2^t)$. Now define the $m \times n$ matrix M' over $GF(2)$, simply by expanding every entry in M into its standard t-bit representation. The bounds in the theorem hold for this M'.

To obtain the lower bound on TS view the columns of M' as partitioned into m blocks (each corresponding to an original column in M), of t bits each. Now consider any $s \times (n - m/2)$ minor of M'. It must contain at least $m/2$ complete blocks. Since the sum of any subset of these rows will yield 0^t in any complete block iff that sum yields 0 over $GF(2^t)$ in the corresponding column of M, we immediately see that the rank of this minor is at least $s/2$ over $GF(2)$ (and thus also over Q). Using the second part of Theorem 2, and $n = 4m \log m$ the lower bound on $TS(b_{M'})$ follow.

To obtain the upper bound on the formula size, consider now the columns of M' as partitioned to t blocks (corresponding to each bit position in the t-bit representation of elements in

$GF(2^t)$), of m columns each. Call the t matrices derived from this decomposition M'_i $(1 \leq i \leq t)$. Let $w_i(e)$ be the ith bit of the weight assigned by w to the edge e of G. It is clear that for every i, $M'_i = M(G, w_i)$. But note that $L^b(b_{M'}) \leq \sum_{i=1}^{t} L^b(b_{M'_i}) \leq tC(G)$. By taking G from Theorem 3, and using $n = O(m \log m)$ again we get the upper bound. □

Note: In describing the matrices $M(G, w)$ there were two nonconstructive parts: the superconcentrator G (constructed probabilistically in Theorem 3) and w that was chosen at random. As we have explicit superconcentrators that are not much larger (Theorem 4), it would be extremely interesting to find an explicit weight assignment w to the edges of any depth 2 superconcentrator G, that will make all minors of $M(G, w)$ nonsingular. A way around this problem was suggested to us by Ben-Or: simply use new variables for the weights. This creates a polynomial of degree 4, but does establish the separation for explicit functions. Formally, combining this idea with Theorems 4, 8 gives:

Corollary 2 *For every field there is a polynomial time algorithm that for every n outputs a degree 4 polynomial f_n on n variables, satisfying $TS(f_n) = O(n^{1+o(1)})$, and $TS(\Delta f_n) = \Omega(n^{2-o(1)})$.*

We conclude this section with a weak lower bound on time-space trade-offs. While weak, note that it implies that linear time algorithms require linear space.

Theorem 9 *Let $M = M(G, w)$ be an $n \times n$ matrix with all minors nonsingular (over some field F). Assume A is a bilinear straight line program (over the same field F) computing b_M, with $T(A) = T$ and $S(A) = S$. Then $2^{T/n}TS = \Omega(n^2)$.*

Proof: Let $T = nk/4$. The straight line program A defines a sequence σ of variables from $\{x_1, \cdots, x_n, y_1, \cdots, y_n\}$ of length at most $nk/4$, in the order they appear in A. By the main lemma in [AM88], there are subsets X_0, Y_0 of the x and y variables, respectively, such that $|X_0| = |Y_0| = n/(2^k)$ satisfying the following property: if we remove all other variables from σ, the resulting subsequence has at most k alternations between x and y variables.

Setting the remaining variables to 0, we get a reduced program A_0 computing the bilinear form on the variables X_0, Y_0, defined by the submatrix M_0 of M on rows X_0 and columns Y_0. Note that $rk(M_0) = n/(2^k)$. Assume wlog that A uses a different set of registers for linear forms of the x's (say R_X) and of the y's (say R_Y). Then A_0, by the property above, can be partitioned into k "stages" such that in any x-stage multiplications can involve only registers from R_Y (but not y variables), and conversely for y-stages.

The lower bound now follows from the following easy facts.

1. Every multiplication gate creates one bilinear form of rank 1.
2. Since $|R_Y| \leq S$, the dimension of the span of rank 1 bilinear forms generated in one x-stage is at most S (and analogously for y-stages).
3. The rank of the output of A_0 is at most the dimension of the span of all rank 1 bilinear forms it computes.
4. Rank is subadditive, so the total dimension is at most Sk, yielding $Sk \geq n/(2^k)$.

□

4 Circuits and formulae with small constants

For this section we restrict attention to the real and complex number fields. We impose the following restriction on our circuit: any plus gate computes a linear combination of its inputs, with coefficients of absolute value at most 1. For any polynomial f, (and similarly for a set of polynomials) denote by $C_1(f)$ and $L_1(f)$ the size of the smallest circuit and formula computing f. We note that the analogs of Propositions 2, 3 hold for C_1 as well.

As we look at asymptotic complexity, the choice of bound 1 is clearly arbitrary, and can be replaced by any other constant. Also, it is clearly

interesting to study C_1 and L_1 only for polynomials with small coefficients. There are no examples of such polynomials for which the restricted model is significantly weaker than the general one. This question becomes especially interesting, since nontrivial lower bounds can be proved in the restricted model.

The first to prove such lower bounds was Morgenstern [Mo73]. He observed that the standard FFT algorithm for the Discrete Fourier Transform works in the restricted model. If DFT denotes the $n \times n$ DFT matrix, then $C_1(l_{DFT}) \leq \frac{1}{2} n \log n$. Morgenstern showed that this is best possible.

Theorem 10 *[Mo73]* $C_1(l_{DFT}) \geq \frac{1}{2} n \log n$

The proof uses a very elegant volume argument. In fact, his proof yield, for any matrix M, the lower bound

Theorem 11 $C_1(l_M) \geq \log |det(M)|$

Note that $det(M) = \Pi_i \lambda_i$, where the λ_i's are the eigenvalues of M. For the DFT and Hadamard mtrices all eigenvalues are $\sqrt{n}$ in absolute value.

Very recently Chazelle [Ch94] used more refined entropy argumnets to derive similar bounds in a slightly stronger model, for problems arising from 2-dimentional range-query computations.

An immediate question, is whether such lower bounds can be proved for one function (rather then a set of functions). The immediate answer follows from observing that the construction of Baur and Strassen (Theorem 1) does not introduce any new constants (except 1). This implies the following circuit lower bounds on explicit bilinear functions.

Corollary 3 *For every* M $C_1(b_M) = \Omega(\log |det(M)|)$. *In particular,* $C_1(b_{DFT})$, $C_1(b_H)$, $C_1(b_{PP}) = \Omega(n \log n)$.

Next we turn attention to formula size of bilinear functions. We again stress that in the unrestricted model, the only nontrivial lower bound on explicit bilinear forms is the the $\Omega(n \log n)$ bound in the end of the previous section. For the restricted model, such a lower bound trivially follows the circuit lower bound above. However, we can do much better, in the main theorem of this section.

Theorem 12 $L_1^b(b_H) = \Omega(n^{\frac{5}{4}})$, $L_1^b(b_{PP}) = \Omega(n^{\frac{9}{8}})$

Comments While we cannot obtain a general clean expression for the formula lower bound in the above Theorem 12, the proof will reveal that the clean form for the examples we have is

$$L_1^b(b_A) = \Omega(\sum_i \sqrt{|\lambda_i(A)|})$$

There are a few things to note with respect to this form.

- It is very interesting to compare it to the circuit size lower bounds in Theorem 11 and Corollary 3, in their dependence on the spectrum of the matrix, namely

$$C_1(b_A) = \Omega(\sum_i \log |\lambda_i(A)|)$$

- It will be clear from the proof below that a similar formula lower bound can be proved for other "regular" matrices A. These include, for example, all incidence matrices of points vs. subspaces in projective geometries, and more generally block designs [Ha86].

- This form of a formula lower bound is tight in general. For any *diagonal* matrix A it can be easily seen that $L_1^b(b_A) = O(\sum_i \sqrt{|\lambda_i(A)|})$.

- Recently [Lo95] generalized our proof and was also able to obtain size-depth tradeoffs for the the liner function l_A.

Now we return to the proof of Theorem 12. Before proving it, we need the following simple lemma.

Lemma 3 *For every* U, V *matrices such that* UV *diagonalizable,*

$$||U||^2 + ||V||^2 \geq 2|\lambda(UV)|$$

Proof: First we observe that without loss of generality, $UV = \lambda(UV)I$. (Otherwise, let P be the unitary matrix for which $PUVP = \lambda(UV)I$,

and set $U' \leftarrow PU$, $V' \leftarrow VP$ and note that $||U'|| = ||U||$ and $||V'|| = ||V||$.) The rest follows from the basic fact that $a^2 + b^2 \geq 2ab$ for any two reals a, b.

□

Proof: (of Theorem 12) Assume that $L_1^b(b_A) \leq \frac{1}{10}nd$, for some $n \times n$ diagonalizable matrix A and some parameter d. Then by the decomposition of Equation 2, and the fact that the formula uses only bounded constants, $\exists U, V$ s.t. $UV = A$ and $|U| + |V| \leq \frac{1}{10}nd$. Let S_U (resp. S_V) be the set of rows of U (resp. columns of V) having entries larger than d in absolute value. Clearly, $|S_U|, |S_V| \leq \frac{n}{10}$, so $S = S_U \cup S_V$ satisfies $|S| \leq \frac{n}{5}$.

Let U' (resp. V') be the matrix U (resp. V) in which we replace the rows (resp. columns) of S by zeros. Thus $|U'|_\infty, |V'|_\infty \leq d$ and with the bound above we have $||U'||^2 + ||V'||^2 \leq nd^2$.

Let $A' = U'V'$. Note that A and A' differ only in the rows and columns indexed by S. Thus we have:

- For $A = H, ||A - A'||^2 \leq \frac{2}{5}n^2$
- For $A = PP, ||A - A'||^2 \leq \frac{2}{5}n^{3/2}$

By the Hoffman-Wielandt inequality, and the properties of these matrices listed at the end of section 2.7, at most $\frac{4}{5}$ of the λ_is in H' (resp. PP') differ from the corresponding ones in H (resp. PP) by more than $\sqrt{n/2}$ (resp. $(n/2)^{\frac{1}{4}}$; note that this bound uses the fact that every row and column in PP has only $O(\sqrt{n})$ 1's). Thus $|\lambda(H')| = \Omega(n^{3/2})$, and $|\lambda(PP')|\Omega(n^{5/4})$. By Lemma 3, we are done. □

Acknowledgements

We are greatful to Allan Borodin for motivating us to look at time-space trade-off question, as well as for many illuminating converstions. We also thank Michael Ben-Or for helpful comments on an earlier version of this paper.

References

[AM88] N. Alon and W. Maass, "Meanders and their applications in lower bounds arguments", *JCSS*, 37, pp. 118–129, 1988.

[Bo94] A. Borodin, *Private communication*, 1994.

[BS82] W. Baur, V. Strassen, "The complexity of partial derivatives", *Theoretical Computer science*, 22, pp. 317–330, 1982.

[Ch94] B. Chazelle, "A Spectral Approach to Lower Bounds", *Proc. of the 35th FOCS*, 1994, to appear.

[DDPW] D. Dolev, C. Dwork, N. Pippenger, and A. Wigderson, "Superconcentrators, Generalizers, and Generalized Connectors with Limited Depth," *Proc. of the 15th STOC*, pp. 42-51, 1983.

[Ha86] M. Hall, *Combinatorial Theory*, John Wiley, 1986.

[Lo95] S. V. Lokam, "Size-depth tradeoffs for linear transformations", manuscript, 1995.

[Mo73] J. Morgenstern, "Note on a lower bound of the linear complexity of the fast Fourier transform", *JACM* 20 (2), pp. 305–306, 1973.

[Mo85] J. Morgenstern, "How To Compute Fast A Function And All Its Derivatives", *SIGACT News*, 16 (4), pp. 60–62, Spring 1985.

[Pi78] N. Pippenger, "Superconcentrators", *SIAM J. Comput.* 6, pp. 298–304, 1978.

[Pi82] N. Pippenger, "Superconcentrators of Depth 2," *J. Comp. and Sys. Sci.* 24, pp. 82-90, 1982.

[Sc80] J. T. Schwartz, "Fast Probabilistic Algorithms for Verification of Polynomial Identities", *JACM* 27 (4), pp. 701–717, 1980.

[St73] V. Strassen, "Die Berechnungskomplexitat vo elementarsymmetrischen Funktionen und von Interpolationskoefizienten", *Numer. Math.* 20, pp. 238–251, 1973.

[SZ94] A. Srinivasan and D. Zuckerman, "Computing with Very Weak Random Sources," *Proc. of the 35th FOCS*, 1994, to appear.

[To80] M. Tompa, "Time Space Tradeoffs for Computing Functions, Using Connectivity Properties of Their Circuits," *J. Comp. and Sys. Sci*, 20, pp. 118-132, 1980.

[Va76] L.G. Valiant, "Graph Theoretic Properties in Computational Complexity," *J. Comp. and Sys. Sci.* 13, pp. 278-285, 1976.

[Wi65] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, pp. 104–109, 1965.

[WZ93] A. Wigderson, D. Zuckerman, "Expanders that Beat the Eigenvalue Bound, Explicit Construction and Applications", *Proc. of the 25th STOC*, pp. 245–251, 1993.