An $n \log n$ Lower Bound for Fourier Transform Computation in the Well Conditioned Model

Nir Ailon
Department of Computer Science
Technion Israel Institute of Technology
Haifa, Israel
nailon@cs.technion.ac.il

April 1, 2014

Abstract

Obtaining a non-trivial (super-linear) lower bound for computation of the Fourier transform in the linear circuit model has been a long standing open problem for over 40 years.

An early result by Morgenstern from 1973, provides an $\Omega(n \log n)$ lower bound for the unnormalized Fourier transform when the constants used in the computation are bounded. The proof uses a potential function related to a determinant. The result does not explain why the normalized Fourier transform (of unit determinant) should be difficult to compute in the same model. Hence, the result is not scale insensitive.

More recently, Ailon (2013) showed that if only unitary 2-by-2 gates are used, and additionally no extra memory is allowed, then the normalized Fourier transform requires $\Omega(n \log n)$ steps. This rather limited result is also sensitive to scaling, but highlights the complexity inherent in the Fourier transform arising from introducing entropy, unlike, say, the identity matrix (which is as complex as the Fourier transform using Morgenstern’s arguments, under proper scaling).

In this work we extend the arguments of Ailon (2013). In the first extension, which is also the main contribution, we provide a lower bound for computing any scaling of the Fourier transform. Our restriction is that, the composition of all gates up to any point must be a well conditioned linear transformation. The lower bound is $\Omega(R^{-1} n \log n)$, where $R$ is the uniform conditon number. Second, we assume extra space is allowed, as long as it contains information of bounded norm at the end of the computation.

The main technical contribution is an extension of matrix entropy used in Ailon (2013) for unitary matrices to a potential function computable for any matrix, using Shannon entropy on “quasi-probabilities”.

1 Introduction

The (discrete) normalized Fourier transform is a complex linear mapping sending an input $x \in \mathbb{C}^n$ to $y = Fx \in \mathbb{C}^n$, where $F$ is an $n \times n$ unitary matrix defined by

$$F(k, \ell) = n^{-1/2} e^{-i 2\pi k \ell / n}.$$

The unnormalized Fourier transform matrix is defined as $n^{1/2} F$. (The unnormalized Fourier transform is often simply referred to, in literature, as the “Fourier transform”.) The Fast Fourier
Transform (FFT) of Cooley and Tukey \cite{2} is a method for computing the Fourier transform (normalized or not - the adjustment is easy) of a vector \( x \in \mathbb{C}^n \) in time \( O(n \log n) \) using a so called linear algorithm. A linear algorithm, as defined in \cite{3}, is a sequence \( \mathcal{F}_0, \mathcal{F}_1, \ldots \), where each \( \mathcal{F}_i \) is a set of affine functions, for each \( i \geq 0 \) \( \mathcal{F}_{i+1} = \mathcal{F}_i \cup \{ \lambda_i f + \mu_i g \} \) for some \( \lambda_i, \mu_i \in \mathbb{C} \) and \( f, g \in \mathcal{F}_i \), and \( \mathcal{F}_0 \) contains (projections onto) the input variables as well as constants.

It is trivial that computing the Fourier Transform requires a linear number of steps, but non-trivial lower bound is known without making very strong assumptions about the computational model. Papadimitriou, for example, computes in \cite{4} an \( \Omega(n \log n) \) lower bounds for Fourier transforms in finite fields using a notion of an information flow network. It is not clear how to extend that result to the Complex field. There have also been attempts \cite{5} to reduce the constants hiding in the upper bound of \( O(n \log n) \), while also separately counting the number of additions versus the number of multiplications (by constants). In 1973, Morgenstern proved that if the modulus of the \( \lambda_i \)’s and \( \mu_i \)’s is bounded by 1 then the number of steps required for computing the unnormalized Fourier transform in the linear algorithm model is at least \( \frac{1}{2} n \log_2 n \). It should be noted that Cooley and Tukey’s unnormalized FFT indeed can be expressed as a linear algorithm with coefficients of the form \( e^{iz} \) for some real \( z \), namely, complex numbers of unit modulus.

The main idea of Morgenstern is to define a potential function for each \( \mathcal{F}_i \) in the linear algorithm sequence, equaling the maximal determinant of a square submatrix in a certain matrix corresponding to \( \mathcal{F}_i \). The technical step is to notice that the potential function can at most double in each step. The determinant of the unnormalized Fourier transform is \( n^{n/2} \), hence the lower bound of \( \frac{1}{2} n \log_2 n \).

The determinant of the normalized Fourier transform, however, is 1. Morgenstern’s method can therefore not be used to derive any useful lower bound for computing the normalized Fourier transform in the linear algorithm model with constants of at most unit modulus. Using constants of modulus \( 1/\sqrt{2} \) in the normalized version of FFT, on the other hand, does compute the normalized Fourier transform in \( O(n \log n) \) steps.

The normalized and unnormalized Fourier transforms are proportional to each other, and hence we don’t believe there should be a difference between their computational complexities in any reasonable computational model. It is important to note that, due to the model’s weakness, Morgenstern’s result teaches us, upon inspection of the proof, that both matrices \( \sqrt{n} \mathcal{F} \) (the unnormalized Fourier transform) and \( \sqrt{n} \text{Id} \) are in the same complexity class. More generally, it tells us that all unitary matrices scaled up by the same constant (\( \sqrt{n} \) in this case) are in the same complexity class. Ailon \cite{11} hence studied the complexity of the Fourier transform within the unitary group. In his result he showed that, if the algorithm can only apply 2 by 2 unitary transformations at each step, then at least \( \Omega(n \log n) \) steps are required for computing the normalized Fourier transform. The proof is done by defining a potential function on the matrices \( M_i \) defined by composing the first \( i \) gates. The potential function is simply the Shannon entropies of the probability distributions defined by the squared modulus of elements in the matrix rows. (Due to unitarity, each row, in fact, thus defines a probability distribution).

This work takes the idea in \cite{11} a significant step forward, and obtains a \( \Omega(n \log n) \) lower bound for any scaling of the Fourier transform in a stronger model of computation which we call the uniformly well conditioned. At each step, the algorithm can either multiply a variable by a nonzero constant, or perform a unitary transformation involving 2 variables. The matrix \( M_i \) defining the composition of the first \( i \) steps must be well conditioned with constant \( R \). This means that \( \| M_i \| \cdot \| M_i^{-1} \| \leq R \), where \( \| \cdot \| \) is spectral norm. Taking this number into account, the actual lower bound we obtain is \( \Omega(R^{-1} n \log n) \). It should be noted that well conditionedness is related to numerical stability:
The less well conditioned a transformation is, the larger the set of inputs on which numerical errors would be introduced in any computational model with limited precision. This result is presented in Section 3.

Another limitation of [1] is that no additional memory (extra variables) were allowed in the computation. (This limitation is not present in [3].) In Section 5 this limitation is removed, assuming a bound on the amount of information held in the extra space at the end of the computation.

1.1 Main Technique

Ailon [1] defined the entropy of a unitary matrix \( M \in \mathbb{C}^{n \times n} \) to be

\[
\Phi(M) = \sum_{i=1}^{n} \sum_{j=1}^{n} f(M(i,j)) \tag{1.1}
\]

where for any nonnegative \( x \),

\[
f(x) = \begin{cases} 
0 & x = 0 \\
-|x|^2 \log |x|^2 & x > 0
\end{cases} \tag{1.2}
\]

Since \( M \) is unitary, for any row \( i \) the numbers \((|M(i,1)|^2, \ldots, |M(i,n)|^2)\) form a probability distribution vector, from which we can view \( \Phi(M) \) as the sum of the Shannon entropy of \( n \) distributions. Note that \( \Phi(M) \) is always in the range \([0, n \log n]\). (Throughout, we will take all logarithms to be in base 2, as common in information theory). Ailon [1] claimed, using a simple norm preservation argument, that for any (complex) Givens matrix \( S \),

\[
|\Phi(M) - \Phi(SM)| \leq 2, \tag{1.3}
\]

where we remind the reader that a Givens matrix is any unitary transformation acting on two coordinates. Since \( \Phi(\text{Id}) = 0 \) and \( \Phi(F) = n \log n \), the conclusion was that at least \( \frac{1}{2} n \log n \) Givens operations are required to compute the (normalized) Fourier transformation \( F \).

The starting point of this work is extending the definition of \( \Phi \) in 1.1 to any (nonsingular) matrix. Indeed, there is no reason to believe that an optimal Fourier transform algorithm must be confined to the unitary group. Using (1.1) verbatim does not help proving a lower bound, as one can easily see that \( \Phi(M) \) can change by \( \Omega(\log n) \) if we multiply a row of \( M \) by a nonzero constant \( C \) such that \( |C| \neq 1 \).

The main contribution of this work is to define a generalized matrix entropy function

\[
\hat{\Phi}(M) := \sum_{i=1}^{n} \sum_{j=1}^{n} \hat{f}(M(i,j), M^{-1}(j,i)) \tag{1.4}
\]

where for all \( x, y \in \mathbb{C} \),

\[
\hat{f}(x,y) = \begin{cases} 
0 & x \cdot y = 0 \\
-x \cdot y \cdot \log |x \cdot y| & x \cdot y \neq 0
\end{cases} \tag{1.5}
\]

Note that if \( M \) is unitary then \( M(i,j) = \overline{M^{-1}(j,i)} \), where \( \overline{\cdot} \) is complex conjugate. This implies that \( \hat{M} \) defined in (1.4) is an extension of (1.1) from the unitary to the nonsingular group. Also note that for all \( i \), the numbers \( M(i,1)M^{-1}(1,i), \ldots, M(i,n)M^{-1}(n,i) \) sum up to one (by definition...
of matrix inversion) but they do not form a probability distribution vector because they may be negative or > 1 in general, hence we think of them as quasi-probabilities. Lemma 3.1 below shows that a Givens rotation applied to \( M \) can change \( \Phi(M) \) by at most \( O(R) \), where \( R \) is the condition number of \( M \). Multiplying rows of \( M \) by nonzero constants clearly does not change \( \Phi(M) \) (although it may change its condition number).

1.2 Contribution and Limitations

We believe that our main contribution is in showing a lower bound on computation of the Fourier transform that does not depend on scaling. Indeed, if \( M_i \) is the matrix defined by the composition of the first \( i \) steps (see exact model definition below), then \( \Phi(M_i) \) is completely insensitive to rescaling by any arbitrary large or small nonzero number, because such an operation has no effect on neither \( \Phi \) or on condition numbers. We argue that the generalized matrix entropy \( \Phi \) defined in (1.4), which is interesting in its own right, is an important key to understanding the complexity of one of the most important linear transformations used in science and engineering, for which an algorithm that is believed to be optimal has been around for half a decade. We point out the following shortcoming of this work, which also gives rise to interesting open problems: Although constant well conditionedness enhances numerical stability, for many applications it is reasonable to work with condition numbers that grow with \( n \), even polynomially. Our result implies nontrivial bounds for condition number up to \( o(\log n) \). It would be interesting to prove interesting lower bounds for less well conditioned computations.

2 The Well Conditioned Model of Computation

Before defining our model, we will simplify the analysis by working over \( \mathbb{R} \) instead of over \( \mathbb{C} \). This can be done by representing the input (and output) using \( 2n \) variables, half dedicated to the real part and half to the imaginary part of the complex input. Accordingly, each matrix element \( F(k, \ell) = n^{-1/2} e^{-i 2 \pi k \ell / n} \) of the complex Fourier transform becomes a \( 2 \times 2 \) rotation matrix with angle \( -2 \pi k \ell / n \), multiplied by \( n^{-1/2} \).

We will henceforth simply assume that \( n \) is even and use \( F \) to denote a real orthogonal \( n \times n \) matrix computing the complex Fourier transform of order \( n/2 \) on an input \( \hat{x} \in \mathbb{C}^{n/2} \), where the real part of \( \hat{x} \) is stored in \( n/2 \) coordinates and the imaginary part in the remaining \( n/2 \).

For a matrix \( M \), we let \( M(i) \) denote the \( i \)th column of \( M \). Our model of computation consists of layers \( L_0, \ldots, L_m \), each containing exactly \( n \) nodes and representing a vector in \( \mathbb{R}^n \). The first layer, \( L_0 \in \mathbb{R}^n \), is the input. The last layer \( L_m \in \mathbb{R}^n \) is the output.

For \( i = 1, \ldots, m \), the \( i \)th gate connects layer \( i - 1 \) with later \( i \). There are two types of gates: rotations and constants.

If gate \( i \) is a rotation, then there are two indices \( k_i, \ell_i \in [n] \), \( k_i < \ell_i \), and an orthogonal matrix

\[
A_i = \begin{pmatrix}
a_{i}(1,1) & a_{i}(1,2) \\
a_{i}(2,1) & a_{i}(2,2)
\end{pmatrix} = \begin{pmatrix}
\cos \theta_i & \sin \theta_i \\
-\sin \theta_i & \cos \theta_i
\end{pmatrix}.
\]

For each \( j \notin \{k_i, \ell_i\} \), \( L_i(j) = L_{i-1}(j) \). The values of \( L_i(k_i) \) and \( L_i(\ell_i) \) are given as

\[
\begin{pmatrix}
L_i(k_i) \\
L_i(\ell_i)
\end{pmatrix} = A_i \begin{pmatrix}
L_{i-1}(k_i) \\
L_{i-1}(\ell_i)
\end{pmatrix}.
\]
Note that the transformation taking \( L_{i-1} \) to \( L \) is a Givens rotation.

If gate \( i \) is of type constant, then it is defined by an index \( k_i \in [n] \) and a nonzero \( c_i \). For each \( j \neq k_i, \) \( L_i(j) = L_{i-1}(j) \). Additionally, \( L_i(k_i) = c_i L_{i-1}(k_i) \).
We will encode the circuit using the sequence
\[
(k_i, \ell_i, \theta_i, c_i)_{i=1}^m,
\]
where we formally define \( c_i \) to be 0 for rotation gates, and \( \ell_i = 0 \) for constant gates.

Let \( M_i \) be the matrix transforming \( L_0 \) (as a column vector) to \( L_i \). We say that \( M_i \) is the \( i \)’th defining matrix of the circuit. If gate \( i \) is a rotation, then \( M_i \) is obtained from \( M_{i-1} \) by replacing rows \( k_i \) and \( \ell_i \) in \( M_{i-1} \) by the application of \( A_i \) to these rows. If gate \( i \) is diagonal, then \( M_i \) is obtained from \( M_{i-1} \) by multiplying row \( k_i \) of \( M_{i-1} \) by \( c_i \). Also, \( M_0 = \text{Id} \).

**Definition 2.1.** A layered circuit of depth \( m \) is \( R \)-uniformly well conditioned if
\[
\max_{i \in [m]} \{ \| M_i \| \cdot \| M_i^{-1} \| \} \leq R.
\]

### 3 The Main Result

**Theorem 3.1.** If an \( R \)-uniformly well conditioned layered circuit \( C = (k_i, \ell_i, \theta_i, c_i)_{i=1}^m \) computes a transformation that is proportional to the Fourier transform \( F \), then the number of rotations is \( \Omega(R^{-1} n \log n) \).

**Proof.** We begin with an observation, which can be proven with a simple induction: For any \( i \in [m] \), \( (M_i^{-1})^T \) is the \( i \)’th defining matrix of a circuit \( C' \) defined by \( (k_i, \ell_i, \theta_i, c'_i)_{i=1}^m \), where \( c'_i = 1/c_i \) if the \( i \)’th gate of \( C \) is of type constant, and 0 otherwise. A clear consequence of this observation is that if the \( i \)’th gate of \( C \) is of type constant, then
\[
\Phi(M_{i-1}) = \Phi(M_i) .
\]

Indeed, just notice that for \( p = k_i \) and any \( q \in [n] \), \( M_i(p, q) = c_i M_{i-1}(p, q) \) and \( (M_i^{-1})^T(p, q) = c_i^{-1} (M_{i-1}^{-1})^T(p, q) \).

We analyze the effect of rotation gates on \( \Phi \). To this end, we need the following lemma.

**Lemma 3.2.** Define \( \hat{f} \) as in (1.5). For 4 real numbers \( w, x, y, z \), define
\[
\Psi(w, x, y, z) = \hat{f}(w, x) + \hat{f}(y, z) .
\]

Now define
\[
\alpha(w, x, y, z) = \sup_{\theta \in [0, 2\pi]} \Psi(w \cos \theta + y \sin \theta, x \cos \theta + z \sin \theta, -w \sin \theta + y \cos \theta, -x \sin \theta + z \cos \theta),
\]
\[
\beta(w, x, y, z) = \inf_{\theta \in [0, 2\pi]} \Psi(w \cos \theta + y \sin \theta, x \cos \theta + z \sin \theta, -w \sin \theta + y \cos \theta, -x \sin \theta + z \cos \theta) .
\]

Then
\[
\sup_{w, x, y, z} \frac{\alpha(w, x, y, z) - \beta(w, x, y, z)}{\sqrt{(w^2 + y^2)(x^2 + z^2)}} = O(1) .
\]
(we formally define the last fraction as 0 if either \( w^2 + y^2 = 0 \) or \( x^2 + z^2 = 0 \). Note that in this degenerate case both \( \alpha(w, x, y, z) = 0 \) and \( \beta(w, x, y, z) = 0 \).)
The proof of the lemma is deferred to Section 4. Now let \( i \) be such that the \( i \)th gate is a rotation. Then
\[
\Phi(M_i) - \Phi(M_{i-1}) = \sum_{q=1}^{n} \left[ \Psi(M_i(k_i, q), M_i^{-1}(q, k_i), M_i(\ell_i, q), M_i^{-1}(q, \ell_i)) 
- \Psi(M_{i-1}(k_i, q), M_{i-1}^{-1}(q, k_i), M_{i-1}(\ell_i, q), M_{i-1}^{-1}(q, \ell_i)) \right]
\]
By Lemma 3.1, hence for some global \( C > 0 \)
\[
|\Phi(M_i) - \Phi(M_{i-1})| \leq C \sum_{q=1}^{n} \sqrt{(M_i(k_i, q)^2 + M_i(\ell_i, q)^2)(M_i^{-1}(q, k_i)^2 + M_i^{-1}(q, \ell_i)^2)}
\]
\[
\leq C \sqrt{\left( \sum_{q=1}^{n} M_i(k_i, q)^2 + M_i(\ell_i, q)^2 \right) \left( \sum_{q=1}^{n} M_i^{-1}(q, k_i)^2 + M_i^{-1}(q, \ell_i)^2 \right)}
\]
\[
\leq C \left\| M_i \right\| \cdot \left\| M_i^{-1} \right\| \leq CR ,
\]
where the second inequality is Cauchy-Schwarz, and the third is from the definition of condition number. Hence,
\[
|\Phi(M_i) - \Phi(M_{i-1})| \leq O(R) .
\]

Now notice that \( \Phi(M_0) = \Phi(\text{Id}) = 0 \) and \( \Phi(M_m) = \Phi(F) = n \log n \). Hence \( m = \Omega(R^{-2} n \log n) \), as required.

\[\Box\]

4 Proof of Lemma 3.1

If either \((w, y) = (0, 0)\) or \((x, z) = (0, 0)\) then the LHS of (3.1) is clearly 0. Assume first that the vectors \((w, y)\) and \((x, z)\) are not proportional to each other. Without loss of generality, we can assume that the vector direction \((1, 0) \in \mathbb{R}^2\) is an angle bisector of the two segments connecting the origin with \((w, y)\) and \((x, z)\). In words, there exist numbers \(r, s > 0\) and an angle \(\phi\) such that
\[
(w, y) = \left( r \cos \frac{\phi}{2}, r \sin \frac{\phi}{2} \right)
\]
\[
(x, z) = \left( s \cos \frac{\phi}{2}, -s \sin \frac{\phi}{2} \right).
\]
By symmetry, we can assume that \(\phi \in [0, \pi]\).

Assume first that \(\phi \leq \pi/2\). With this notation, we have for all \(\theta \in [0, 2\pi]\)
\[
w \cos \theta + y \sin \theta = r \cos \left( \frac{\phi}{2} + \theta \right) \quad x \cos \theta + z \sin \theta = s \cos \left( -\frac{\phi}{2} + \theta \right) \quad (4.1)
\]
\[
-w \sin \theta + y \cos \theta = r \sin \left( \frac{\phi}{2} + \theta \right) \quad -x \sin \theta + z \cos \theta = s \sin \left( -\frac{\phi}{2} + \theta \right) \quad (4.2)
\]
Therefore,
\[
\Psi(w \cos \theta + y \sin \theta, x \cos \theta + z \sin \theta, -w \sin \theta + y \cos \theta, -x \sin \theta + z \cos \theta) = 
\]
\[
-\, rs \cos \left(\frac{\phi}{2} + \theta\right) \cos \left(-\frac{\phi}{2} + \theta\right) \log \left| rs \cos \left(\frac{\phi}{2} + \theta\right) \cos \left(-\frac{\phi}{2} + \theta\right) \right| \quad \text{(4.3)}
\]
\[
-\, rs \sin \left(\frac{\phi}{2} + \theta\right) \sin \left(-\frac{\phi}{2} + \theta\right) \log \left| rs \sin \left(\frac{\phi}{2} + \theta\right) \sin \left(-\frac{\phi}{2} + \theta\right) \right|. \quad \text{(4.4)}
\]

We view the last expression as a function of \(\theta\), and write \(\Psi(\theta)\) for shorthand. The function \(\Psi\) is differentiable everywhere except \(\phi \in Q = \{\pm \frac{\pi}{2} + j\frac{\pi}{2}\} \) for \(j = 0, 1, 2, \ldots\). For \(\phi \in Q\), it is not hard to see that \(\Psi\) is not a local optimum. It hence suffices to find local optima of \(\Psi\) for \(\phi \not\in Q\).

Consider first the range \(\theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)\). In this range, the argument inside the absolute value in (4.3) is positive, while the one inside (4.4) is negative. Differentiating with respect to \(\theta\), we get
\[
\frac{d}{d\theta} \Psi(\theta) = rs \left[ \sin \left(\frac{\phi}{2} + \theta\right) \cos \left(-\frac{\phi}{2} + \theta\right) + \cos \left(\frac{\phi}{2} + \theta\right) \sin \left(-\frac{\phi}{2} + \theta\right) \right] 
\]
\[
\times \left[ 1 + \log \left( rs \cos \left(\frac{\phi}{2} + \theta\right) \cos \left(-\frac{\phi}{2} + \theta\right) \right) \right] 
\]
\[
+ \, rs \left[ \cos \left(\frac{\phi}{2} + \theta\right) \sin \left(-\frac{\phi}{2} + \theta\right) + \sin \left(\frac{\phi}{2} + \theta\right) \cos \left(-\frac{\phi}{2} + \theta\right) \right] 
\]
\[
\times \left[ 1 - \log \left( rs \sin \left(\frac{\phi}{2} + \theta\right) \sin \left(-\frac{\phi}{2} + \theta\right) \right) \right] 
\]
\[
= rs (\sin 2\theta) \left[ 2 + \log \left( -\frac{\cos \left(\frac{\phi}{2} + \theta\right) \cos \left(-\frac{\phi}{2} + \theta\right)}{\sin \left(\frac{\phi}{2} + \theta\right) \sin \left(-\frac{\phi}{2} + \theta\right)} \right) \right]. 
\]

One checks using standard trigonometry that the last log is nonnegative. Hence, \(d\Psi/d\theta\) vanishes only when \(\theta = 0\). For this value,
\[
\Psi(0) = -rs \cos^2 \left(\frac{\phi}{2}\right) \log \left( rs \cos^2 \left(\frac{\phi}{2}\right) \right) + rs \sin^2 \left(\frac{\phi}{2}\right) \log \left( rs \sin^2 \left(\frac{\phi}{2}\right) \right). \quad \text{(4.5)}
\]

We now study the case \(\phi \in (\phi/2, \pi/2 - \phi/2)\). In this range, the argument inside the absolute value in both (4.3) and (4.4) is positive. For this case, using a similar derivation as above, the derivative \(d\Psi/d\theta\) equals
\[
\frac{d}{d\theta} \Psi(\theta) = rs (\sin 2\theta) \log \left( \frac{\cos \left(\frac{\phi}{2} + \theta\right) \cos \left(-\frac{\phi}{2} + \theta\right)}{\sin \left(\frac{\phi}{2} + \theta\right) \sin \left(-\frac{\phi}{2} + \theta\right)} \right). 
\]

By our assumption on \(\phi\), the last derivation vanishes exactly when \(\theta = \pi/4\). For this value,
\[
\Psi(\pi/4) = -rs \cos \left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos \left(-\frac{\phi}{2} + \frac{\pi}{4}\right) \log \left( rs \cos \left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos \left(-\frac{\phi}{2} + \frac{\pi}{4}\right) \right) 
\]
\[
-\, rs \sin \left(\frac{\phi}{2} + \frac{\pi}{4}\right) \sin \left(-\frac{\phi}{2} + \frac{\pi}{4}\right) \log \left( rs \sin \left(\frac{\phi}{2} + \frac{\pi}{4}\right) \sin \left(-\frac{\phi}{2} + \frac{\pi}{4}\right) \right)
\]
\]

7
By basic trigonometry, one verifies that
\[
\cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right) = \frac{1}{2} \cos^2\left(\frac{\phi}{2}\right) - \frac{1}{2} \sin^2\left(\frac{\phi}{2}\right)
\]
\[
\sin\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \sin\left(-\frac{\phi}{2} + \frac{\pi}{4}\right) = \frac{1}{2} \cos^2\left(\frac{\phi}{2}\right) - \frac{1}{2} \sin^2\left(\frac{\phi}{2}\right).
\]
Plugging in our derivation of \(\Psi(\pi/4)\), we get
\[
\Psi(\pi/4) = -rs \left(\cos^2\left(\frac{\phi}{2}\right) - \sin^2\left(\frac{\phi}{2}\right)\right) \log \left(r \cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)\right).
\]
(4.6)

It is not hard to verify that \(\Psi(0)\) and \(\Psi(\pi/4)\) are the only extremal values of \(\Psi\). Now notice that in the expression \(|\Psi(\pi/4) - \Psi(0)|\), the term \(\log(rs)\) is cancelled out, and we are left with \(|\Psi(\pi/4) - \Psi(0)| = rs g(\phi)\), where
\[
g(\phi) = \cos^2\left(\frac{\phi}{2}\right) \log \left(\frac{\cos^2\left(\frac{\phi}{2}\right)}{\cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)}\right) + \sin^2\left(\frac{\phi}{2}\right) \log \left(\frac{\cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)}{\sin^2\left(\frac{\phi}{2}\right)}\right).
\]
The function \(g(\phi)\) is bounded in the range \(\phi \in (0, \pi/4]\), by which we conclude that for some global constant \(C\),
\[
\left|\sup_{\theta} \Psi(\theta) - \inf_{\theta} \Psi(\theta)\right| \leq Crs.
\]
This concludes the proof for the case \(\phi \leq \pi/2\). Other cases are argued similarly, we omit the details. This concludes the proof of the lemma.

5 Using Additional Space

We assume in the section that aside from the \(n\) input variables, the algorithm has access to an additional memory of total size \(N\). We would like to explore to what extent this additional memory could help in Fourier computation, using the framework developed in the previous sections. We will assume throughout that
\[
N \leq n \log n,
\]
(5.1)
because, assuming all extra memory is accessed in the linear circuit, \(N/2\) is a lower bound on the depth of the circuit. Additionally, we will assume that this additional memory is initialized as 0.

For convenience, we will work with linear circuits as defined in Section 2 over \(\mathbb{R}^{n+N}\), and assume the the \(N\) extra input variables are initialized to 0. Note that this assumption is not restrictive, because the linear contribution of the extra \(N\) input variables to the \(n\) output variables must be null.

As a warmup, we will also assume that the \(N\) extra output variables are identically 0. In other words, that there is no “garbage information” in the \(N\) output variables. This means that, if the circuit has depth \(m\) then then \([M_m]_{[n],[n]} = F\) where for a matrix \(A\) and integer sets \(I, J\) \([A]_{I,J}\) denotes the submatrix of \(A\) corresponding to rows \(I\) and columns \(J\). We will later relax this assumption.
Theorem 5.1. If an R-uniformly well conditioned layered circuit \( C \) computes a transformation \( M \) such that \( M_{[n],[n]} = F \) and \( M_{[n+N]\,[n],[n]} = 0 \), then the number of rotations in the circuit is \( \Omega(R^{-1}n \log n) \).

Proof. We proceed as in the proof of Theorem 3.1 except we now work with a partial entropy function defined as follows:

\[ \Phi_n(M) := -\sum_{i=1}^{n+N} \sum_{j=1}^{n} \hat{f}(M(i,j), M^{-1}(j,i)) , \]  

(5.2)

It is easy to see that, as before, for any \( i \) such that the \( i \)'th gate is a rotation, \( |\Phi_n(M_i) - \Phi_n(M_{i+1})| = O(R) \). We also notice that, by the assumptions, we must have \( [M^{-1}_n]_{[n],[n]} = F^{-1} \) and \( [M^{-1}_n]_{[n],[N+n],[n]} = 0 \). This implies, as before, that \( \Phi_m(M_0) = 0 \) and \( \Phi_m(M_m) = n \log n \), leading to the claimed result.

It is arguably quite restrictive to assume that the extra space must be cleaned from any information “garbage” at the end of the computation. In particular, by inspection of the last proof, the “garbage” could have a negative contribution to \( \Phi_n \), possibly reducing the computational lower bound. This assumption is relaxed in the following, by limiting the norm of the “garbage”.

Theorem 5.2. Let \( C \) be an R-uniformly well conditioned layered circuit of depth \( m \). Assume that \( [M_m]_{[n],[n]} = F \) and that additionally each column of \( [M_m]_{[N+n],[n],[n]} \) has Euclidean norm at most \( 1/(4R) \). Then \( m = \Omega(R^{-1}n \log n) \).

Note that in both Theorems 5.1 and 5.2, the normalization chosen in the theorems is immaterial. For example, we could have replaced \( F \) and \( 1/(4R) \) in Theorem 5.2 with \( cF \) and \( c/(4R) \), respectively, for any nonzero \( c \). Hence these lower bounds are insensitive to scaling. We chose the specific normalization to eliminate extra constants in the analysis.

Proof. We will work with the potential function \( \Phi_n \) defined in (5.2), except that we cannot know the exact value of \( \Phi_n(M_m) \) as in the proof of Theorem 5.1. Denote the columns of \( [M_m]_{[N+n],[n],[n]} \) by \( u_1, \ldots, u_n \in \mathbb{R}^N \), and the columns of \( [(M_m^{-1})^T]_{[N+n],[n],[n]} \) by \( v_1, \ldots, v_n \in \mathbb{R}^N \). By the assumption of the lemma,

\[ \max\{\|u_1\|, \ldots, \|u_n\|\} \leq 1/(4R) . \]  

(5.3)

Hence the norm of any of the first \( n \) columns of \( M_m \) is in the range \( [1, 1 + 1/(4R)] \). This implies that the spectral norm \( \|M_m\| \) is at least 1. By the well conditionedness of \( M_m \), we conclude that \( \|M_m^{-1}\| \) is at most \( R \), by which we conclude that

\[ \max\{\|v_1\|, \ldots, \|v_n\|\} \leq R . \]  

(5.4)

Using standard optimization tools (e.g. Lagrange multipliers) over the constraints (5.3) and (5.4), we have that for any \( j \in [n] \),

\[ \sum_{i=1}^{N} \hat{f}(u_j(i), v_j(i)) \geq -\frac{1}{4} \log 4 - \frac{1}{4} \log N = -\frac{1}{2} - \frac{1}{2} \log N \geq -\frac{1}{2} - \frac{1}{2} \log n , \]  

(5.5)

where in the rightmost inequality we used the assumption that \( N \leq n \log n \leq n^2 \).
Now let \( h_1, \ldots, h_n \in \mathbb{R}^n \) denote the columns of \( [M_m]^{[n]}_{[n]} = F \), and \( z_1, \ldots, z_n \in \mathbb{R}^n \) denote columns \( 1 \ldots n \) of \( \left[(M_m^{-1})^T\right]^{[n]}_{[n]} \). We have that for all \( j \in [n] \),

\[
\max\{\|h_j\|_\infty\} \leq \sqrt{2/n} \quad \max\{\|z_j\|_2\} \leq R ,
\]

where the left inequality is by properties of the normalized Fourier transform \( F \) and the right one is by the assumption of well conditionedness of \( M_m \). By definition of matrix inverse, we have that for all \( j \in n \),

\[
z_j^T h_j + v_j^T u_j = 1 .
\]

But by Cauchy Schwarz and (5.3) and (5.4), \( |v_j^T u_j| \leq 1/4 \). Hence,

\[
z_j^T h_j \geq 3/4 .
\]

Combining (5.6) and (5.7) we conclude by using standard optimization techniques (e.g. using Lagrange multipliers) that

\[
\sum_{i=1}^{n} \hat{f}(u_j(i), v_j(i)) \geq \frac{3}{4} \log n - \frac{3}{4}(1 + \log R) .
\]

Combining (5.5) and (5.8) we conclude that

\[
\Phi_n(M_m) \geq \frac{1}{2} n \log n - n \left( \frac{5}{4} + \frac{3}{4} \log R \right) .
\]

For \( n = \Omega(R) \) the last expression if at least \( \frac{1}{n} \log n \). Since as usual \( \Phi_n(M_0) = 0 \) we conclude that \( m = \Omega(R^{-1} n \log n) \) as required. \( \square \)

References

[1] Nir Ailon. A lower bound for fourier transform computation in a linear model over 2x2 unitary gates using matrix entropy. *Chicago J. of Theo. Comp. Sci.*, 2013.

[2] J. W Cooley and J. W Tukey. An algorithm for the machine computation of complex Fourier series. *J. of American Math. Soc.*, pages 297–301, 1964.

[3] Jacques Morgenstern. Note on a lower bound on the linear complexity of the fast Fourier transform. *J. ACM*, 20(2):305–306, April 1973.

[4] Christos H. Papadimitriou. Optimality of the fast Fourier transform. *J. ACM*, 26(1):95–102, January 1979.

[5] S. Winograd. On computing the discrete Fourier transform. *Proc. Nat. Assoc. Sci.*, 73(4):1005–1006, 1976.