The Fast Fourier Transform

Zan Ahmad Adam Moritz

May 12, 2020

1 Introduction

The fast Fourier Transform (FFT) remains one of the most widely used and important algorithms of the 20th Century. Today, it is used in digital recording, signal processing, medical image reconstruction, pitch correction tools, additive synthesis, and more. It is a mathematical method that essentially converts a discrete function of time or space into a function of frequency using Fourier Analysis. Most of the modern implementations are based off of the method developed by James W. Cooley and John W. Tukey in 1965 [1], but its original formulations date back to unpublished work from Gauss in 1805 [6]. This version of the FFT provides an efficient algorithm for computing the Discrete Fourier Transform (DFT) of a sequence of N points.

In this report we introduce the notion of the Discrete Fourier Transform and provide a precise mathematical definition. Although the DFT is a useful operation with applications in many fields, a direct computation following the definition proves to be impractical. We consider Cooley and Tukey's FFT algorithm and how it implements a more efficient divide-and-conquer method that significantly reduces the order of complexity and speed of the computation. Finally, we move to an interesting discussion regarding the Entropic Uncertainty Principle for the Discrete Fourier Transform and give an outline for two proofs, one that is more mathematically involved [4] and one that highlights the relationship between the Entropic Uncertainty Principle and FFT[8].

2 Discrete Fourier Transform

In the case of continuous functions, the Fourier Transform is a generalization of the complex Fourier series, taking the form

$$F(t) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i t x} dx \tag{1}$$

where $f \in L^1(\mathbb{R})$. However, in many applications, f is not continuous, but instead is a finite sequence of equally spaced samples of points from a continuous function.

2.1 Definitions and Representations

In the discrete case, the Fourier Transform takes the form of a linear transform between vectors, with the function f represented as a complex vector of N points, taken as equally distanced samples of the original periodic function $(f_0, f_1, \ldots, f_{N-1})$. There has been research in the area of nonuniform sampling of the function f [5], but it is outside the scope of this report. Written in summation form, the DFT of a given discrete function f can be computed as

$$F_n = \sum_{k=0}^{N-1} f_k e^{-2\pi i n k / N} \tag{2}$$

As the DFT can be thought of as an approximation of the Fourier coefficients of a given function by means of sampling at equivalent intervals, it is sensible to wonder under what conditions the DFT well approximates its original function's Fourier coefficients. It can be seen that if c_n is the nth Fourier coefficient of a function f, then F_n/N well approximates c_n when f is smooth, with an even finer approximation when N is large. Often, as in Cooley and Tukey's formulation, $\omega_N = e^{-2\pi i/N}$ is taken as the principal Nth root of unity, and as such we can rewrite (2) as

$$F_n = \sum_{k=0}^{N-1} f_k \cdot \omega_N^{nk} \tag{3}$$

Another way of looking at the DFT is through the use of the counting measure μ such that

$$\mu(f) = \sum_{k=0}^{N-1} f(x_k)$$

Where $(x_0, x_1, \ldots, x_{N-1})$ is a mesh of equally spaced points in a function with period N.We can then look at the Fourier transform of f with respect to this counting measure

$$\hat{f}(n) = \int_{-\infty}^{\infty} \mu(f) e^{-2\pi itx} dx = \sum_{k=0}^{N-1} f(x_k) e^{-2\pi ink/N} = \sum_{k=0}^{N-1} f_k \omega_N^{nk} \tag{4}$$

Where the DFT becomes the value of this Fourier transform taken at discrete values n in the range $0 \leq n \leq N-1$, with $F_n = \hat{f}(n)$. It can be becomes useful to refer to the DFT as a Vandermonde matrix of ω_N:

$$W = \begin{bmatrix} \omega_N^{0\cdot 0} & \omega_N^{0\cdot 1} & \ldots & \omega_N^{0\cdot(N-1)} \\ \omega_N^{1\cdot 0} & \omega_N^{1\cdot 1} & \ldots & \omega_N^{1\cdot(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_N^{(N-1)\cdot 0} & \omega_N^{(N-1)\cdot 1} & \ldots & \omega_N^{(N-1)\cdot(N-1)} \end{bmatrix} \tag{5}$$

where the DFT of f can be calculated as a matrix vector product $F = Wf$.

2.2 Properties of the DFT

Many interesting properties of the DFT can be seen using this matrix representation. An important aspect of the continuous Fourier transform is the Inversion Theorem, or that if $f \in L^1$ and $\hat{f} \in L^1$, then the function g defined as

$$g(x) = \int_{-\infty}^{\infty} \hat{f} e^{-2\pi ixt} dt \tag{6}$$

is C_0 and $f(x) = g(x)$ almost everywhere. In the discrete case, this Inversion is represented as the inverse of the Vandermonde matrix W, which is equal to

$$W^{-1} = \frac{1}{N} \begin{bmatrix} (\omega_N^*)^{0\cdot 0} & (\omega_N^*)^{0\cdot 1} & \ldots & (\omega_N^*)^{0\cdot(N-1)} \\ (\omega_N^*)^{1\cdot 0} & (\omega_N^*)^{1\cdot 1} & \ldots & (\omega_N^*)^{1\cdot(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ (\omega_N^*)^{(N-1)\cdot 0} & (\omega_N^*)^{(N-1)\cdot 1} & \ldots & (\omega_N^*)^{(N-1)\cdot(N-1)} \end{bmatrix} \tag{7}$$

Where ω_N^* is the complex conjugate of ω_N. As such, it can be seen that $W^{-1} = \frac{1}{N} W^*$, and that if we take $U = \frac{1}{\sqrt{N}} W$, then the DFT becomes a Unitary transformation such that $U^{-1} = U^*$. This

provides a discrete analog to Plancherel's theorem, which states that the Fourier transform can be seen as a unitary isomorphism of L^2 onto L^2, or that for every $f \in L^2$, $||\hat{f}||_2 = ||f||_2$.
In the discrete case, as we know that the DFT is unitary, or that it preserves the inner product between two vectors f_n and g_n such that

$$\sum_{k=0}^{N-1} f_k g_k^* = \sum_{k=0}^{N-1} F_k G_k^* \tag{8}$$

where F is the DFT of f and G is the DFT of g, we can take the special case where $f = g$ and arrive at the analog of Plancherel's theorem, where

$$\sum_{k=0}^{N-1} |f_k|^2 = \sum_{k=0}^{N-1} |F_k|^2 \tag{9}$$

2.3 Computational Drawbacks

It can be seen from Section 2.1 that calculating the corresponding DFT of a sequence of N points takes N^2 operations, where each operation comprises a complex multiplication and a complex addition, producing an $\mathrm{O}(N^2)$ running time [1]. For large N, this is computationally prohibitive, and diminishes DFT's effectiveness in most modern applications. As such, the problem of searching for ways to compute a given signal's DFT in a computationally efficient way has been of great importance.

3 Cooley–Tukey FFT

The paper by Cooley and Tukey is considered the seminal work in this field, regarded for its simplicity in implementation, as well as its much faster running time of $\mathrm{O}(n \log n)$. In fact, most modern FFT algorithms are based off of their original formulation in some way. We will now walk through a detailed overview of Cooley and Tukey's algorithm, and explain how this efficiency is achieved.

3.1 Assumptions

Cooley and Tukey's algorithm is based off a divide and conquer approach. While they were not the first to attempt to create an efficient algorithm for computing DFT's using divide and conquer, their approach combined an assumption about the size of the input vector. Their work relied on the assumption that the size N was a power of two. Although there has been further work done in optimizing the algorithm in other special cases, such as when N is a power of 4 or 8, the radix-2 case remains one of the simplest and easiest to implement algorithms in computing the DFT of a sequence of values.

3.2 Radix-2 Case

As said above, the key part of the radix-2 case of Cooley and Tukey's FFT algorithm relies on splitting up the sequence of size N into two sequences of size $N/2$ recursively. The algorithm splits up the sequence into the even indexed inputs and the odd indexed inputs as follows:

$$F_n = \sum_{k=0}^{N/2-1} f_{2k} e^{\frac{-2\pi i}{N}(2k)n} + \sum_{k=0}^{N/2-1} f_{2k+1} e^{\frac{-2\pi i}{N}(2k+1)n} \tag{10}$$

We can then factor out $e^{\frac{-2\pi i}{N}n}$ from the second term yielding

$$F_n = \sum_{k=0}^{N/2-1} f_{2k} e^{\frac{-2\pi i}{N/2}kn} + e^{\frac{-2\pi i}{N}n} \sum_{k=0}^{N/2-1} f_{2k+1} e^{\frac{-2\pi i}{N/2}(kn)} \tag{11}$$

From this it can be seen that the first term is the DFT for the even indexed inputs, and the term on the right is the DFT for the odd indexed inputs

$$F_n = \sum_{k=0}^{N/2-1} f_{2k} \cdot \omega_{N/2}^{nk} + \omega_N^n \sum_{k=0}^{N/2-1} f_{2k+1} \cdot \omega_{N/2}^{nk} \tag{12}$$

Let

$$F_{n,even} = \sum_{k=0}^{N/2-1} f_{2k} \cdot \omega_{N/2}^{nk} \text{ and } F_{nodd} = \sum_{k=0}^{N/2-1} f_{2k+1} \cdot \omega_{N/2}^{nk} \tag{13}$$

so (12) becomes

$$F_n = F_{n,even} + \omega_N^n F_{n,odd} \tag{14}$$

The time saving aspect of Cooley and Tukey's algorithm comes into play when we can express other entries of F in terms of $F_{n,even}$ and $F_{n,odd}$, namely $F_{n+N/2}$ We write $F_{n+N/2}$ as

$$F_{n+N/2} = \sum_{k=0}^{N/2-1} f_{2k} e^{\frac{-2\pi i}{N/2}k(n+N/2)} + e^{\frac{-2\pi i}{N}(n+N/2)} \sum_{k=0}^{N/2-1} f_{2k+1} e^{\frac{-2\pi i}{N/2}(k(n+N/2))} \tag{15}$$

Which is equal to

$$\sum_{k=0}^{N/2-1} f_{2k} e^{\frac{-2\pi i}{N/2}kn} e^{-2\pi ki} + e^{\frac{-2\pi i}{N}n} e^{-\pi i} \sum_{k=0}^{N/2-1} f_{2k+1} e^{\frac{-2\pi i}{N/2}kn} e^{-2\pi ki} \tag{16}$$

Which by Euler's identity is equal to

$$\sum_{k=0}^{N/2-1} f_{2k} e^{\frac{-2\pi i}{N/2}kn} - e^{\frac{-2\pi i}{N}n} \sum_{k=0}^{N/2-1} f_{2k+1} e^{\frac{-2\pi i}{N/2}kn} \tag{17}$$

It can be seen
that these two terms are identical to $F_{n,even}$ and $F_{n,odd}$. So $F_{n+N/2}$ can be rewritten as

$$F_{n+N/2} = F_{n,even} - \omega_N^n F_{n,odd}$$

Now that we know this, we can
create an algorithm that computes the FFT much more efficiently than direct computation. The pseudocode for the algorithm proposed by Cooley and Tukey can be seen in Figure 1.

While this result is often attributed to Cooley and Tukey, a very similar method can be seen in a paper written by Gordon C. Danielson and Cornelius Lanczos published in 1942 [2], as well as in the paper by Gauss we previously

FFT(f):
  $N \leftarrow \text{len}(f)$
  if $N = 1$ **then**
    return N
  else
    $\omega_N \leftarrow e^{-2\pi i/N}$
    $f_{n,even} \leftarrow (f_0, f_2, \ldots, f_{n-2})$
    $f_{n,odd} \leftarrow (f_1, f_3, \ldots, f_{n-1})$
    $F_{n,even} \leftarrow \text{FFT}(f_{n,even})$
    $F_{n,odd} \leftarrow \text{FFT}(f_{n,odd})$
    $\omega_0 = 1$
    for $i = 0$ to $N/2 - 1$ **do**
      $F[i] \leftarrow F_{n,even}[i] + \omega_0 * F_{n,odd}[i]$
      $F[i + N/2] \leftarrow F_{n,even}[i] - \omega_0 * F_{n,odd}[i]$
      $\omega_0 \leftarrow \omega_0 * \omega_N$
    end for
  end if
  return F

Figure 1: Pseudocode of the Radix-2 FFT

mentioned [6]. As each problem of size N can be broken down into two of size $N/2$ and the intermediate operations can be done in time O(N), it can be seen that this algorithm can be represented by the recurrence relation

$$T(N) = 2T(N/2) + O(N)$$

Using recurrence relation analysis, it can be derived that this algorithm runs in time $O(n \log n)$.

3.3 Highly Composite Case

In their paper, Cooley and Tukey presented a more general time-saving algorithm in the case that N is not a power of two, but instead highly composite, with $N = r_1 \cdot r_2$. In this case, Cooley and Tukey outline an algorithm that decomposes the one dimensional DFT of size N into a two dimensional DFT of size r_1 by r_2. In doing this, they were able to compute the DFT of f in time $O(N(r_1 + r_2))$. They also outlined a procedure to generalize the algorithm further through recursion, showing that if N can be written

$$N = r \cdot r_2 \cdots r_m$$

then the DFT of f can be computed in

$$O(N(r_1 + r_2 + \cdots + r_m))$$

time. Further research has been done in expanding and optimizing Cooley and Tukey's algorithms, and they still are considered by many to be foundational in the field of FFTs.

3.4 Applications of DFT/FFT

In order to show how wide reaching these principles are in a qualitative manner, we will provide in this subsection a brief overview of a various fields where FFT is of great importance. Since FFT is an algorithm that quickly computes discrete Fourier Transforms, anywhere that analysis of sampled signals is performed, the FFT is often utilized to aid in this analysis. In the field of medicine, the Fourier Transform is used in the signal processing of images. The MRI (magnetic resonance imaging), for example, takes an input which is a complex superposition of periodic signals received by the detector (receiver coils) which may be decomposed into a simple signals. The Fourier transform resolves the frequency and phase-encoded MR signals that compose "k-space," the spatial frequency information, (two-dimensional in this case) defined by the space covered by the phase and frequency encoding data. The MR image that is produced is the 2D inverse Fourier transform of k-space. Fourier transforms can also be used to perform signal analysis regarding musical applications such as pitch tracking, noise reduction, and other real-time implementations regarding controls and leveling of sound characteristics. An example of FFT being used as a diagnostic tool can be seen in a previous study performed at the University of Pittsburgh in which FFT was used in conjunction with machine learning methods as a means of diagnosing throat-related conditions. The study sought to remove the need for x-ray imaging by replacing it with analyzing the signals produced by various devices such as microphones and accelerometers placed on the exterior of the throat, from which features were extracted through the use of FFT to be used in machine learning classification [7].

4 Entropic Uncertainty Principle

We now shift focus to a discussion about the entropic uncertainty principles. To begin, the original statement of the uncertainty principle introduced by Heisenberg in quantum mechanics states that

the greater the precision at which a particle's position is known, the less precisely the momentum of the particle may be determined and vise-versa and this holds in general with any pair of canonically conjugate variables.[3]. In other words, the product of the variables is bounded below. Within the context of harmonic analysis, a function and its Fourier transform cannot both be made arbitrarily small simultaneously and the product of the two is bounded below.

The entropic uncertainty principle gives a stronger statement than the original uncertainty principle. The relationship between the two principles comes from the fact that the uncertainty principle can be expressed as a lower bound of the sum of the Shannon entropies of conjugate variables. The entropic uncertainty principle states that this sum is bounded below.

4.1 Definitions

Let the entropy H of a discrete random variable X: $\{x_1, x_2, ...x_n\}$ be defined as it is in information theory by Shannon[9]:

$$H(X) = -\sum_{i=1}^{n} P(x_i)\log_b P(x_i) \tag{18}$$

where b is the base of the logarithm (we will assume a base of e in this section) and P is the probability mass function of X.

We may also consider the Shannon entropy of a continuous random variable. Let $|f|^2$ be the density of a probability measure on $\mathbb{R}$ so that $\int |f|^2 dx = 1$. Then the Shannon entropy for $|f|^2$ replaces the sum in the discrete case by integrating over $\mathbb{R}$:

$$H(|f|^2) = -\int_{-\infty}^{\infty} |f(x)|^2 \log_b |f(x)|^2 dx \tag{19}$$

If we take $\hat{f}$ to be the Fourier transform of f with the Shannon entropies from our continuous definition above, then the statement for the entropic uncertainty principle is as follows:

$$H(|f|^2) + H(|\hat{f}|^2) \geq \log(e/2) \tag{20}$$

and

$$H(|f|^2) + H(|\hat{f}|^2) = \log(e/2) \iff f \text{ is Gaussian [3]} \tag{21}$$

Let X and Y be discrete random variables with probability distribution functions $P(X = i) = |x_i|^2/||\mathbf{x}||_2^2$ and $P(Y = i) = |Ux_i|^2/||U\mathbf{x}||_2^2$ where $\mathbf{x}$ is a finite vector and U is a unitary $n \times n$ matrix (a matrix which has its conjugate transpose equal to its inverse). Then using the definition of Shannon entropy provided above for the discrete case, the entropic uncertainty principle is the following inequality:

$$H(X) + H(Y) \geq 2\log(\frac{1}{M}) \tag{22}$$

In the subsequent subsections, we will outline two proofs regarding the discrete case. For the general statement above we very closely follow a short proof by Amir Dembo et al. [4] that is quite mathematically involved. There is also an inductive proof we will carry out for a specific case following steps from notes on a talk given by Charles Peskin [8] that is much more elementary in the tools it utilizes as it is strictly concerned with a special case in finite-dimensional vector space.

4.2 Proof by Dembo et al.

U is a unitary matrix and thus for all x, $||U\mathbf{x}||_2 = ||\mathbf{x}||_2$. It is obvious given this, that for $||x||_p = [\sum_{i=1} |x_i|^P]^{1/p}$ and $||x||_\infty = \sup_{i=1}\{|x_i|\}$, $||U\mathbf{x}||_\infty \leq M||\mathbf{x}||_1$. By the Riesz-Thorin Interpolation Theorem from $1 \leq p \leq 2$ gives the following Hausdorff-Young inequality (see Theorem 22 in [4]) for an arbitrary $\mathbf{x}$ and value of p within the range:

$$||U\mathbf{x}||_q \leq M^{(2-p)/p}||\mathbf{x}||_p \tag{23}$$

where $\frac{1}{p} + \frac{1}{q} = 1$. Using the Renyi entropies for generalized dimensions we have that for X and Y defined in Section 4.1:

$$H_{p/2}(X) = \frac{1}{1-(p/2)}\log\sum_i P(X=i)^{p/2} \tag{24}$$

$$= \frac{p}{1-p/2}\log(||\mathbf{x}||_p/||\mathbf{x}||_2), \tag{25}$$

$$H_{q/2}(Y) = \frac{1}{1-(q/2)}\log\sum_i P(Y=i)^{q/2} \tag{26}$$

$$= \frac{q}{1-q/2}\log(||U\mathbf{x}||_q/||U\mathbf{x}||_2), \tag{27}$$

This Hausdorff-Young inequality from above (23) can be expressed as:

$$(\frac{1}{p}-\frac{1}{2})H_{p/2}(X) + (\frac{1}{2}-\frac{1}{q})H_{q/2}(Y) \leq (\frac{1}{p}-\frac{1}{2})2\log(\frac{1}{M}) \tag{28}$$

Setting $(1/p) = (1/2) + \epsilon$ and $(1/q) = (1/2) - \epsilon$ and dividing the above inequality (28) by ϵ and then taking $\epsilon \to 0$ gives:

$$H(X) + H(Y) \geq 2\log(\frac{1}{M}) \tag{29}$$

4.3 Inductive Proof by Peskin

Here we shall consider the special case of the entropic uncertainty principle taking $U = F_n$ such that F_n is the discrete Fourier transform of order $n = 2^p$. $F_n : \mathbb{C}^n \to \mathbb{C}^n$ is given by:

$$(F_n u)_k = \frac{1}{\sqrt{n}}\sum_{i=0}^{n-1} e^{-i2\pi jk/n}u_i \tag{30}$$

for $k = 0, 1, ..., n-1$. F_n is unitary with respect to the Euclidean norm. Given a vector $u \in \mathbb{C}$, we normalize it so that $V_i = |u_i|/||u||$. The Shannon entropy of u thus follows the definition from Section 4.1 (18):

$$H(u) = -\sum_{i=0}^{n-1} V_i^2 \log(V_i^2) \tag{31}$$

We may interpret $P_i = V_i^2$ to be the probability of the ith event and thus $H(u)$ is the entropy of the probability distribution of the discrete random variable P which has possible values $\{P_i\}_{i=0}^{n-1}$. The entropic uncertainty principle states:

$$H(u) + H(F_n u) \leq 2\log(1/M) \tag{32}$$

where $M = \max_{ik}|U_{ik}|$ and since we can verify easily that F_n in its matrix representation is unitary, every column/row is a unit vector which further implies that $1/\sqrt{n} \leq M \leq 1$. The greater we make the lower bound on the right hand side, the stronger the statement of the inequality. Thus, let us take $M \to 1/\sqrt{n}$ and the right handside becomes $2\log(\sqrt{n}) = \log(\sqrt{n}^2) = \log(n)$. This extreme case is what yields the statement for the entropic uncertainty principle for the discrete Fourier transform:

$$H(u) + H(F_n u) \leq \log(n) \tag{33}$$

Let the order of the discrete Fourier transform F_n be $n = 2^p$. We proceed in the same way as section 3.2 for the Radix-2 case to set up the recursion relation for the FFT algorithm and substitute the notation used in this section:

$$(F_n u)_k = \frac{1}{\sqrt{n}} \sum_{i=0}^{\frac{n}{2}-1} e^{-j2\pi 2ik/n} u_{2i} + \frac{1}{\sqrt{n}} \sum_{i=0}^{\frac{n}{2}-1} e^{-j2\pi(2i+1)k/n} u_{2i+1} \tag{34}$$

$$= \frac{1}{\sqrt{2}} \frac{1}{\sqrt{n/2}} \sum_{i=0}^{\frac{n}{2}-1} e^{-j2\pi ik/(n/2)} u_{2i} + \frac{e^{-j2\pi k/n}}{\sqrt{n/2}} \frac{1}{\sqrt{2}} \sum_{i=0}^{\frac{n}{2}-1} e^{-j\pi(2i)k/(n/2)} u_{2i+1} \tag{35}$$

Given the possible values of $i, k : \{0, 1, 2, ..., (n/2) - 1\}$, we can set

$$v_k^0 = (F_n u)_k, \tag{36}$$

$$v_k^1 = (F_n u)_{k+n/2} \tag{37}$$

$$u_{2i} = u_i^{\text{even}} \tag{38}$$

$$u_{2i+1} = u_i^{\text{odd}} \tag{39}$$

$$w^0 = F_{n/2} u^{\text{even}} \tag{40}$$

$$w^1 = F_{n/2} u^{odd} \tag{41}$$

for $i = 0, 1, ..., n/2 - 1$ and let D be an $\frac{n}{2} \times \frac{n}{2}$ matrix with entries at the ik position to be zero when $i \neq k$ (off-diagonal entries) and $e^{-i2\pi k/n}$ when $i = k$ (diagonal entries). D is unitary and the following equations are derived:

$$v^0 = \frac{1}{\sqrt{2}}(w^0 + Dw^1), \tag{42}$$

$$v^1 = \frac{1}{\sqrt{2}}(w^0 - Dw^1), \tag{43}$$

$$v_k^0 = \frac{1}{\sqrt{2}}(w_k^0 + D_{kk} w^1), \tag{44}$$

$$v_k^1 = \frac{1}{\sqrt{2}}(w_k^0 - D_{kk} w^1), \tag{45}$$

$$\tag{46}$$

Computing the squares of the last two component equations above yields:

$$|v_k^0|^2 = \frac{1}{2}(|w_k^0|^2 + |w_k^1|^2 + w_k^0 \overline{D_{kk} w_k^1} + \overline{w_k^0} D_{kk} w_k^1) \tag{47}$$

$$|v_k^1|^2 = \frac{1}{2}(|w_k^0|^2 + |w_k^1|^2 - w_k^0 \overline{D_{kk} w_k^1} + \overline{w_k^0} D_{kk} w_k^1) \tag{48}$$

and summing these together gives the equality:

$$|v_k^0|^2 + |v_k^1|^2 = |w_k^0|^2 + |w_k^1|^2 \tag{49}$$

Therefore if we define vectors $v, w \in \mathbb{C}^n$ with the following components:

$$v_k = v_k^0, v_{k+n/2} = v_k^1, \tag{50}$$

$$w_k = w_k^0, w_{k+n/2} = w_k^1, \tag{51}$$

for $k = 0, 1, ..., (n/2) - 1$ we can easily see that their norms are the same: $||v|| = ||w||$. This information can be used to derive a lower bound for the sum of the Shannon entropies of these vectors $H(v) + H(w)$ and for the sake of concision we will leave the details of the derivation out of this report and simply state it. The steps may be found in $(29) - (36)$ of [8]:

$$H(v) + H(w) \geq \log(2) + 2H_0(w) \tag{52}$$

where

$$H_0(w) = -\sum_{k=0}^{\frac{n}{2}-1} \frac{|w_k^0|^2 + |w_k^1|^2}{||w||^2} \log(\frac{|w_k^0|^2 + |w_k^1|^2}{||w||^2}) \tag{53}$$

Using this lower bound we may now begin a proof for the entropic uncertainty principle for the discrete Fourier transform which states that the lower bound on the sum of the Shannon entropies of a function and its Fourier transform is $\log(n)$ and we focus on the specific case where the order n of the Fourier transform are all the powers of $2 \rightarrow n = 2^p$:

$$H(u) + H(F_n u) \geq \log(n) \tag{54}$$

We proceed by induction on p. Consider the base case for $p = 0$. This gives us $n = 1$ which results in a trivial computation yielding the equality of 0 of on both sides since $H(u) = 0$, $H(F_1 u) = 0$ and $\log(1) = 0$. Now for the inductive case, let us assume that the statement for the entropic uncertainty principle for the discrete Fourier transform is true for $p = k - 1$ and we want to show that the statement is true for $p + 1 = k$ such that $n = 2^{p+1} = 2^k$. Therefore our starting point is at the order $2^{k-1} = 2^k/2 = n/2$. Our inductive hypothesis is as follows:

$$H(x) + H(F_{n/2} x) \geq \log(\frac{n}{2}) \tag{55}$$

for all $x \in \mathbb{C}$. Let $x_1 = u^{\text{even}} = u_{2i}$ and $x_2 = u^{\text{odd}} = u_{2i+1}$ and take $w^0 = F_{n/2} u_{2i}$ and $w^1 = F_{n/2} u_{2i+1}$. We can write the following inequality:

$$H(u_{2i}) + H(w^0) \geq \log(n/2) \tag{56}$$

$$H(u_{2i+1}) + H(w^1) \geq \log(n/2) \tag{57}$$

The components of $u, w \in \mathbb{C}^n$ are expressed below:

$$||u||^2 = ||u_{2i}||^2 + ||u_{2i+1}||^2 \tag{58}$$

$$||w||^2 = ||w^0||^2 + ||w^1||^2 \tag{59}$$

Now we may write the entropies of the vectors u and w as follows:

$$H(u) = \frac{||u_{2i}||^2}{||u||^2} H(u_{2i}) + \frac{||u_{2i+1}||^2}{||u||^2} H(u_{2i+1}) + H((||u_{2i}||, ||u_{2i+1}||)) \tag{60}$$

$$H(w) = \frac{||w^0||^2}{||w||^2} H(w^0) + \frac{||w^1||^2}{||w||^2} H(w^1) + H((||w^0||, ||w^1||)) \tag{61}$$

$$\tag{62}$$

Note that $||u_{2i}|| = ||w^0||$ and $||u_{2i+1}|| = ||w^1||$ and $||w|| = ||u||$ as a result of $F_{n/2}$ being unitary. We can rewrite the inequality from above:

$$H(u_{2i}) + H(w^0) = 2H(w^0) \geq \log(n/2) \tag{63}$$

$$H(u_{2i+1}) + H(w^1) = 2H(w^1) \geq \log(n/2) \tag{64}$$

Adding the entropies now results in the following:

$$H(u) + H(w) = \frac{||u_{2i}||^2}{||u||^2} H(u_{2i}) + \frac{||w^0||^2}{||w||^2} H(w^0) + \frac{||u_{2i+1}||^2}{||u||^2} H(u_{2i+1}) + \frac{||w^1||^2}{||w||^2} H(w^1) \tag{65}$$

$$+ H((||u_{2i}||, ||u_{2i+1}||)) + H((||w^0||, ||w^1||)) \tag{66}$$

$$= 2H(w^0) \frac{||w^0||^2}{||w||^2} + 2H(w^1) \frac{||w^1||}{||w||^2} + 2H((||w^0||, ||w^1||)) \tag{67}$$

Using this and the inequality that last inequality, we note that

$$H(u) + H(w) \geq \log(n/2) + 2H((||w^0||, ||w^1||)) \tag{68}$$

Since we know that

$$H(v) + H(w) \geq \log(2) + 2H_0(w) \tag{69}$$

we can add this to the previous statement to get the new inequality:

$$H(u) + H(v) + 2H(w) \geq \log(n/2) + \log(2) + 2[H((||w^0||, ||w^1||)) + H_0(w)] \tag{70}$$

$$H(u) + H(v) \geq \log(n) + 2[H((||w^0||, ||w^1||)) + H_0(w) - H(w)] \tag{71}$$

Substituting $H(w) = \frac{||w^0||^2}{||w||^2} H(w^0) + \frac{||w^1||^2}{||w||^2} H(w^1) + H((||w^0||, ||w^1||))$ gives:

$$H(u) + H(v) \geq \log(n) + 2[H_0(w) - \frac{||w^0||^2}{||w||^2} H(w^0) - \frac{||w^1||^2}{||w||^2} H(w^1)] \tag{72}$$

We now have our $\log(n)$ term that we require as our lower bound in the statement we want to prove. As long as

$$H_0(w) - \frac{||w^0||^2}{||w||^2} H(w^0) - \frac{||w^1||^2}{||w||^2} H(w^1) \geq 0 \tag{73}$$

it is true that

$$H(u) + H(v) \geq \log(n) \tag{74}$$

Let $P_k^0 = |w_k^0|^2 / ||w^0||^2$ and $P_k^1 = |w_k^1|^2 / ||w^1||^2$. Then

$$P_k = P_k^0 \frac{||w^0||^2}{||w||^2} + P_k^1 \frac{||w^1||^2}{||w||^2} \tag{75}$$

$$= \frac{|w_k^0|^2 + |w_k^1|^2}{||w||^2} \tag{76}$$

Note that if we take $\frac{||w^0||^2}{||w||^2} \geq 0$ and $\frac{||w^1||^2}{||w||^2} \geq 0$, then $\frac{||w^0||^2}{||w||^2} + \frac{||w^1||^2}{||w||}^2 = 1$ and P is a discrete probability distribution evidently. Let $\mathbb{H}(P)$ be the entropy of P. Then we can say that

$$H(w^0) = \mathbb{H}(P^0) \tag{77}$$

$$H(w^1) = \mathbb{H}(P^1) \tag{78}$$

$$H_0(w) = \mathbb{H}(P^0 \frac{||w^0||^2}{||w||^2} + P^1 \frac{||w^1||^2}{||w||^2} = 1) \tag{79}$$

Subsequently

$$[H((||w^0||, ||w^1||)) + H_0(w) - H(w)] \tag{80}$$

$$= \mathbb{H}(P^0 \frac{||w^0||^2}{||w||^2} + P^1 \frac{||w^1||^2}{||w||^2}) - (\mathbb{H}(P^0) + \mathbb{H}(P^1)) \tag{81}$$

$$\tag{82}$$

Of course, this is nonnegative since $-\mathbb{H}$ is convex and we can check this by simply noting that the Shannon entropy of a discrete random variable $\mathbb{H}(P)$ which we defined in section 4.1 is known to be concave. We have thus proven the result:

$$H(u) + H(v) \geq \log(n) \tag{83}$$

$$v = F_n u \tag{84}$$

5 Conclusion

After a brief introduction to Discrete Fourier Transform and its properties relating to the continuous Fourier Transform, we laid out one of the proofs outlined by Cooley and Tukey, showing how the DFT can be computed in $O(n \log n)$ time in the special case when N is a power of two, and in $O(N(r_1 + r_2))$ time when N is highly composite with $N = r_1 \cdot r_2$. In our discussion of entropic uncertainty principle, we showed two proofs. One for the general case and one of a special case for the DFT with the orders as only the powers of 2. The significance of providing this latter proof as opposed to the former one is that by proceeding this way we highlight a connection between entropic uncertainty and the fast Fourier transform that is not evident in the original proof of the general statement and we utilize more elementary tools appropriate for the confinement to a finite dimensional vector space that we are working in. The bound of $\log(n)$ is suggestive of the order of complexity $O(n\log(n))$ of the fast Fourier Transform where for $n = 2^p$ each step of the algorithm contributes to $\log(2)$ of the total entropy.

References

[1] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[2] G.C. Danielson and C. Lanczos. Some improvements in practical fourier analysis and their application to x-ray scattering from liquids. *Journal of the Franklin Institute*, 233(4):365 – 380, 1942.

[3] Kiril Datcheb and Long Jin. Discrete fourier transform and uncertainty principles.

[4] A. Dembo, T. M. Cover, and J. A. Thomas. Information theoretic inequalities. *IEEE Transactions on Information Theory*, 37(6):1501–1518, 1991.

[5] Leslie Greengard and June-Yub Lee. Accelerating the nonuniform fast fourier transform. *SIAM review*, 46(3):443–454, 2004.

[6] Michael T. Heideman, Don H. Johnson, and C. Sidney Burrus. Gauss and the history of the fast fourier transform. *Archive for History of Exact Sciences*, 34(3):265–277, Sep 1985.

[7] Adam Moritz, Yassin Khalifa, and Ervin Sejdic. *Material Viscosity Prediction under Normal Swallowing Conditions via High Resolution Cervical Auscultation.* IEEE 2018 Conference on Biomedical Health Informatics, 2018.

[8] Charles S Peskin. The entropic uncertainty principle and the fast fourier transform, Mar 2020.

[9] Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.