Derivations and Applications of the Discrete, Fast, and Short-Time Fourier Transforms

APPM 4600 Fall 2024 Section 001

Elizabeth Cutting, Edward Wawrzynek, Erick White

15 December 2024

1 Introduction

In this project, we investigate the Fast Fourier Transform (FFT), a method for efficient computation of the trigonometric interpolation of a function along an interval. The Fourier transform has numerous applications in signal filtering, communication, and data processing, as it reveals information in the frequency domain. Without the insight from the FFT, this would go unnoticed and thus be nearly useless. This property pushes the limits of the possible applications, which extend to speech analysis and recognition, image processing and filtration, and financial analysis.

We begin with a derivation of the Discrete Fourier Transform (DFT) by drawing on the theory of Fourier series, then show its drawbacks, focusing on its $\mathcal{O}(n^2)$ computational complexity. We introduce the Cooley-Tukey FFT algorithm and show how it can reduce this complexity to $\mathcal{O}(n \log n)$.

We explore the properties of the FFT, including its restrictions on the sample points and spectral leakage, and compare its performance to that of the DFT.

Then, we continue with an exploration of the Short-Time Fourier Transform (STFT), a common tool in time-dependent signal analysis and processing due to its ability to incorporate information relating to a function's behavior over time. The STFT goes beyond the FFT, introducing new capabilities, including time-dependent sonar analysis and musical study.

We motivate this method's utility with a focus on the discrete STFT, focusing specifically on its use in analyzing scientific data; in doing so, we illustrate why it is necessary (and why the FFT is not sufficient) for many real-world signals. We discuss how the STFT can be used to generate spectrograms and we discuss the parameters used in the STFT. We focus especially on the windowing function, considering several common windowing functions used in the STFT and discuss their benefits and drawbacks. We then use magnetometer data from NASA's Van Allen Probe A to illustrate the use of the STFT in a real-world scientific context. We highlight how spectrograms can be used to illustrate otherwise hidden patterns in the data, focusing on EMIC waves in the Earth's magnetosphere, and discuss why having such a tool is crucial to scientific discovery.

2 The Discrete Fourier Transform

2.1 Derivation

Information is frequently contained in signals which are dense or tricky to interpret in the time domain. By applying the methods discussed in the following sections, a signal can be decomposed into its constituent frequency components, revealing the amplitude and phase of each frequency present. Such an insight is the bedrock of the modern approach to signal processing. Below, we derive the Discrete Fourier Transform (DFT) from the mathematical foundation of Fourier series.

2.1.1 Finite Fourier Series

Consider some piecewise smooth square-integrable function $f(t) \in \mathbb{C}$, $f \in L^2$ defined over the interval $t \in [-\pi, \pi]$. Over that interval, we are interested in approximating f with a trigonometric basis; that is, a basis composed of sines and cosines. This approximation is called the Fourier series for f, and is given as

$$\hat{f}(t) = A_0 + \sum_{n=1}^{N} A_n \cos(nt) + \sum_{n=1}^{N} B_n \sin(nt). \tag{1}$$

Euler's identity allows us to write an individual term in the series as

$$A_n \cos(nt) + B_n \sin(nt) = \frac{A_n - iB_n}{2} e^{int} + \frac{A_n + iB_n}{2} e^{-int},$$

so we can write the full series (1) as

$$\begin{aligned}\hat{f}(t) = A_0 + \sum_{n=1}^{N} \left(c_n e^{int} + c_{-n} e^{-int}\right) \\ = \sum_{n=-N}^{N} c_n e^{int}, \end{aligned} \tag{2}$$

where the coefficients are given by

$$c_n = \begin{cases} \frac{1}{2}\left(A_{-n} + iB_{-n}\right) & n < 0 \\ A_0 & n = 0 \\ \frac{1}{2}\left(A_n - iB_n\right) & n > 0 \end{cases}.$$

We can invert this relationship to give the coefficients

$$A_0 = c_0, \; A_n = c_n + c_{-n}, \text{ and } B_n = ic_n + ic_{-n}. \tag{3}$$

We wish to find these coefficients such that they minimize the mean squared error.

Theorem 1. *Let $f(t) \in \mathbb{C}$ be a piecewise smooth function defined over $t \in [-\pi, \pi]$, and let $\hat{f}$ be the Fourier series*

$$\hat{f}(t) = \sum_{n=-N}^{N} c_n e^{int}.$$

Then the mean square error between f and $\hat{f}$,

$$E = \int_{-\pi}^{\pi} \left| f(t) - \hat{f}(t) \right|^2 \mathrm{d}t,$$

is minimized when c_n is chosen as

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} \mathrm{d}t. \tag{4}$$

Proof. See Section 5.1. □

By substitution of (4) into (3), we obtain the usual synthesis equations for the series,

$$\begin{aligned} A_0 &= c_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) \mathrm{d}t, \\ A_n &= c_n + c_{-n} = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(nt) \mathrm{d}t, \\ B_n &= c_n + ic_{-n} = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(nt) \mathrm{d}t. \end{aligned}$$

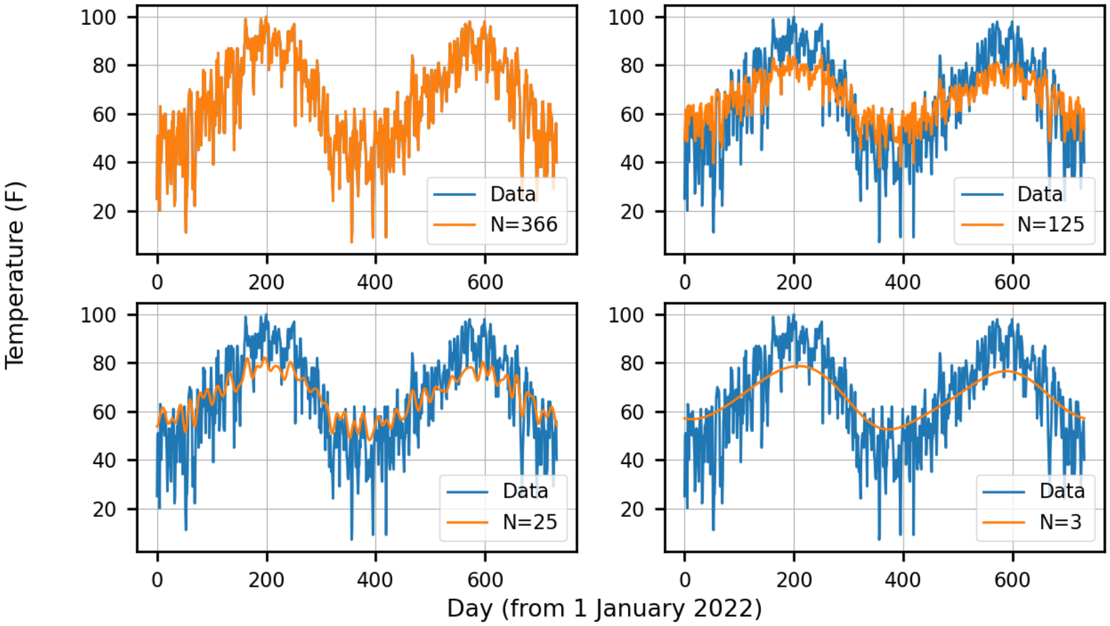


Figure 1: The daily high temperature in Boulder, CO over the period 2022-2023 [12] (blue), interpolated with a truncated Fourier series (orange). The data is perfectly approximated by the series with $N = 366$ terms, while a limited number of terms captures the low frequency behavior of the data but misses the high frequency behavior.

For numerical applications, we are restricted to a finite number of coefficients. Indeed, since the magnitudes of the coefficients often decrease with n, truncation of the series can still lead to accurate approximations. Figure 1 presents an example of the effect of truncation of the series, where more terms in the series corresponds to an improved approximation.

2.1.2 Discrete Signals and Coefficients

When measuring signals, it is impossible to capture perfect continuous data. As such, we often obtain results that are composed of discrete points which form samples of a signal. An example of a continuous-time signal and its discrete-time sampling is shown in Figure 2. We wish to extend our earlier results on continuous-time functions to discretely sampled functions. We will find that this can be done in a straightforward manner through discrete sums.

A discrete signal f with uniformly spaced samples $f_0, f_1, \ldots, f_{N-1}$ has a discrete Fourier transform (DFT) given by

$$\hat{f}_k = \sum_{j=0}^{N-1} f_j e^{-\frac{i2\pi jk}{N}}. \tag{5}$$

Equation 5 plays the same role as (4) did for the continuous-time series. Indeed, (5) minimizes the mean squared error in a similar manner as the continuous case.

Theorem 2. *Let $\{f_j\}_{j=0}^{N-1}$ be a discrete-time function, and let $\{\hat{f}_k\}_{k=0}^{N-1}$ be the Fourier series given by*

$$\hat{f}_k = \sum_{n=-N}^{N} c_n e^{ink}.$$

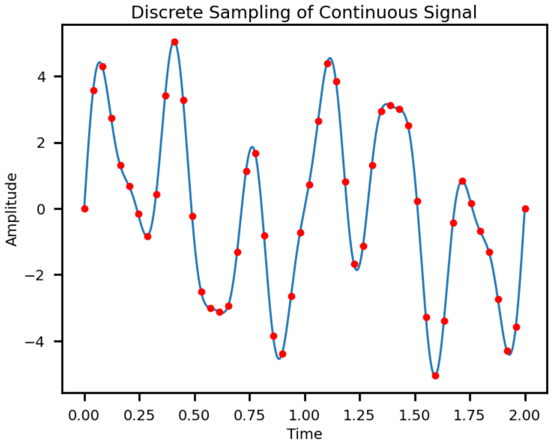


Figure 2: A continuous-time function $f(t) = 2\sin(2\pi t) + 3\sin(6\pi t) + \sin(11\pi t)$ (blue) and its discrete-time sampling (red) at a rate of 25 samples per second. Note the discrete nature of the sampled points introduces the necessity of well-sampled data, else one risks losing information.

Then the mean squared error

$$E = \sum_{j=0}^{N-1} \left| f_j - \hat{f}_j \right|^2$$

is minimized when the coefficients c_n are chosen as (5)*.*

Proof. Omitted. Similar to the continuous time case (Theorem 1) [11]. □

By analogy to the relationship in (3), we have

$$A_0 = \hat{f}_0,\ A_n = \hat{f}_n + \hat{f}_{-n},\ \text{and}\ B_n = i\hat{f}_n - i\hat{f}_{-n}.$$

To compute the full transform, we wish find the coefficients $\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_{N-1}$ in terms of the samples $f_0, f_1, \ldots, f_{N-1}$. Equation (5) gives us a system of N equations for our N unknown coefficients. We observe that the fundamental frequency in the DFT is $\omega_N = e^{-i2\pi/N}$. We can write the system as $\hat{\mathbf{f}} = F_N \mathbf{f}$, where the system matrix is

$$F_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & & \omega_N^{2(N-1)} \\ \vdots & & & \ddots & \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & & \omega_N^{(N-1)^2} \end{bmatrix}.$$

Evaluating this system directly is expensive. F_N is of size $N \times N$, which means that the system requires $\mathcal{O}(N^2)$ time to evaluate, which can be severely limiting when trying to process a large number of sample points.

2.1.3 The Cooley-Tukey FFT Algorithm

To address the limitations of the costly DFT, in 1965, Cooley and Tukey introduced an algorithm for computation of the DFT that reduced its complexity to $\mathcal{O}(n \log n)$, now known as Cooley-Tukey Fast Fourier Transform [2]. We describe their algorithm below.

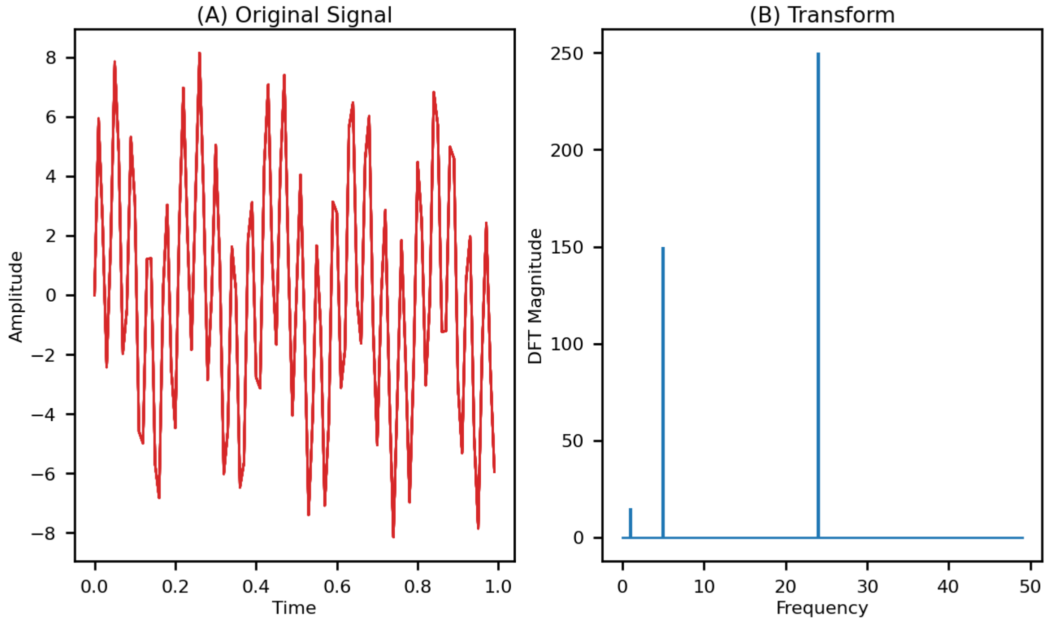


Figure 3: (A) represents the original signal and is a graph of $f(t) = 0.3\sin(2\pi t) + 5\sin(24 \cdot 2\pi t) + 3\sin(5 \cdot 2\pi t)$ with 100 equispaced sampling points. (B) denotes the transform into the frequency domain. Note the frequencies of the sine wave appear in the transform with the corresponding magnitude, which is the information sought after when applying the DFT.

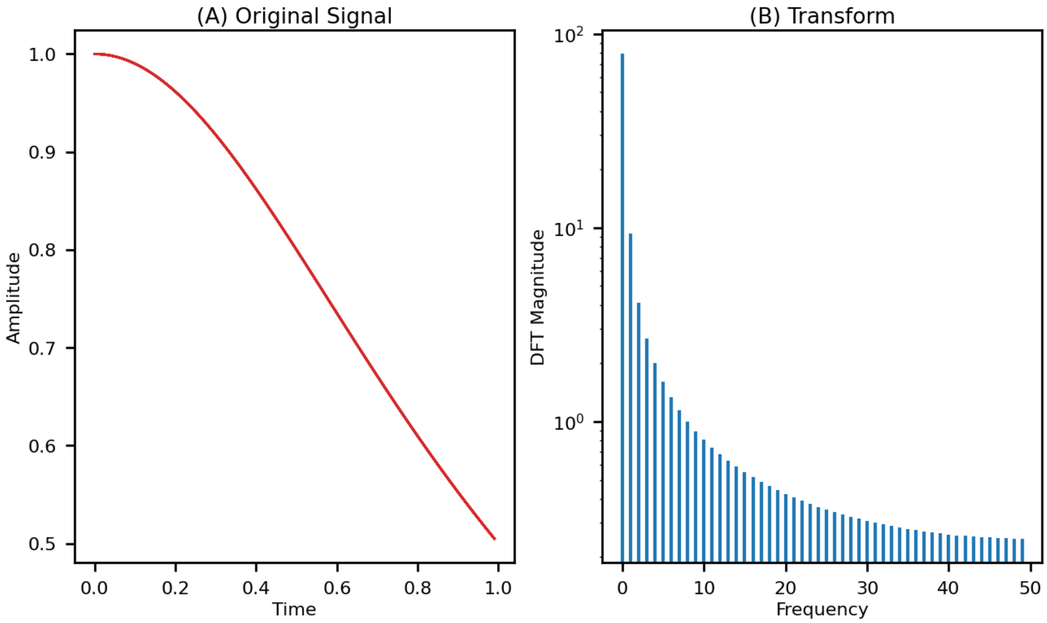


Figure 4: (A) depicts the original signal, which here is $f(t) = \frac{1}{1+x^2}$ with 100 equispaced sampling points. (B) is the DFT on logarithmic scale. Here, (B) has many contributing frequencies, many of which are small but non-negligible.

We consider a signal with a number of samples which is a power of two, that is, we restrict ourselves to $N = 2^k, k \in \mathbb{N}$. As a first step towards improving the performance of our DFT calculation, we will split the problem in two. Let $n = N/2$. By breaking (5) into sums over even and odd indices, we have

$$\begin{aligned}\hat{f}_k &= \sum_{j=0}^{n-1} f_{2j} e^{\frac{-i2\pi k}{N}(2j)} + \sum_{j=0}^{n-1} f_{2j+1} e^{\frac{-i2\pi k}{N}(2j+1)} \\ &= \sum_{j=0}^{n-1} f_{2j} e^{\frac{-i2\pi k}{N/2}j} + e^{\frac{-i2\pi k}{N}} \sum_{j=0}^{n-1} f_{2j+1} e^{\frac{-i2\pi k}{N/2}j} \\ &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{kj} + \omega_n^{k/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{kj}.\end{aligned} \tag{6}$$

For $k = 0, 1, \ldots, n-1$, these two sums bear a resemblance to our original DFT; the sum over the even indices looks like a DFT of the even samples, while the sum over the odd indices looks like a DFT of the odd samples. However, we want to compute the coefficients for the entire range $k = 0, 1, \ldots, 2n-2$. Observe that from periodicity of the exponential, we have

$$\omega_n^{\frac{1}{2}n} = e^{\frac{-i2\pi(n/2)}{n}} = e^{-i\pi} = -1 \quad \text{and} \quad \omega_n^n = e^{\frac{-i2\pi n}{n}} = e^{-i2\pi} = 1.$$

Thus,

$$\begin{aligned}\hat{f}_{k+n} &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{(k+n)j} + \omega_n^{(k+n)/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{(k+n)j} \\ &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{kj} \left(\omega_n^n\right)^j + \omega_n^{k/2} \omega_n^{n/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{kj} \left(\omega_n^n\right)^j \\ &= \sum_{j=0}^{n-1} f_{2j} \omega_n^{kj} - \omega_n^{k/2} \sum_{j=0}^{n-1} f_{2j+1} \omega_n^{kj}.\end{aligned} \tag{7}$$

Together, (6) and (7) allow us to compute the coefficients $\hat{f}_k$ in terms of the DFT of the even sample terms and the DFT of the odd sample terms.

We wish to express this computation in matrix form. We rearrange our sample vector $\mathbf{f}$ to place even indexed samples before odd indexed samples,

$$\mathbf{f}^* = \begin{bmatrix} f_0 \\ f_2 \\ \vdots \\ f_{2n-2} \\ f_1 \\ f_3 \\ \vdots \\ f_{2n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{f_{even}} \\ \mathbf{f_{odd}} \end{bmatrix}.$$

Our system now becomes $\hat{\mathbf{f}} = F^*\mathbf{f}^*$, where

$$F^* = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \dots & \omega_n^{n-1} & \omega_n^{\frac{1}{2}} & \omega_n^{\frac{1}{2}+1} & \dots & w_n^{\frac{1}{2}+(n-1)} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} & \omega_n^{\frac{n-1}{2}} & \omega_n^{\frac{n-1}{2}+n-1} & \dots & \omega_n^{\frac{n-1}{2}+(n-1)^2} \\ 1 & 1 & \dots & 1 & -1 & -1 & \dots & -1 \\ 1 & \omega_n & \dots & \omega_n^{n-1} & -\omega_n^{\frac{1}{2}} & -\omega_n^{\frac{1}{2}+1} & \dots & -w_n^{\frac{1}{2}+(n-1)} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} & -\omega_n^{\frac{n-1}{2}} & -\omega_n^{\frac{n-1}{2}+n-1} & \dots & -\omega_n^{\frac{n-1}{2}+(n-1)^2} \end{bmatrix}.$$

The top half of the F^* matrix is from (6) while the bottom half is from (7). We can factor F^* as

$$\begin{aligned} F^* &= \begin{bmatrix} 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \omega_n^{\frac{1}{2}} & \dots & 0 \\ \vdots & & \ddots & & \vdots & & \ddots & \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & w_n^{\frac{n-1}{2}} \\ 1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & -\omega_n^{\frac{1}{2}} & \dots & 0 \\ \vdots & & \ddots & & & \vdots & & \ddots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & -w_n^{\frac{n-1}{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 \\ 1 & \omega_n & \dots & \omega_n^{n} & 0 & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 & 1 & \omega_n & \dots & \omega_n^{n} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 & \omega_n^{n-1} & \dots & \omega_n^{(n-1)^2} \end{bmatrix} \\ &= \begin{bmatrix} I_n & D_n \\ I_n & -D_n \end{bmatrix} \begin{bmatrix} F_n & 0 \\ 0 & F_n \end{bmatrix}, \end{aligned}$$

where D_n is the diagonal matrix multiplied before the odd part of the DFT,

$$D_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & \omega_n^{\frac{1}{2}} & 0 & \dots & 0 \\ 0 & 0 & \omega_n^{\frac{2}{2}} & \dots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \dots & \omega_n^{\frac{n-1}{2}} \end{bmatrix}.$$

We now have a way to split the DFT into two computations each on half the number of samples. Each of these computations (the DFT on f_{even} and f_{even}) can be done by recursive application of the FFT algorithm. Once we reach a sufficiently small system, which depends on the best optimization at the machine level, we can directly evaluate the DFT via (5). This process is shown schematically in Figure 5. This method reduces the time needed to evaluate the full DFT to $\mathcal{O}(N \log N)$.

To demonstrate the impact of this complexity reduction in real world analysis, we test an example signal of three superimposed sine waves, shown in Figure 6. We measure the computation time for the DFT and FFT for a number of points ranging from 1000 to 10,000, with the computation time plotted in Figure 8.

In order for the FFT to take just as much time as the DFT processing a signal with 10,000 samples, the FFT must process a signal of roughly 1,000,000 samples.

One important consequence of this breakdown is that in order to perfectly divide the points until we reach a set of 2×2 systems, we require, as noted above, that our original signal consists of 2^n, $n \in \mathbb{N}$ points. In real world applications, this often requires us to truncate or pad data. However, since the transform will only give correct results on a commensurate periodic function, or a function sampled over a whole number of periods, truncating data can be extremely problematic [3]. A fractional number of periods will result

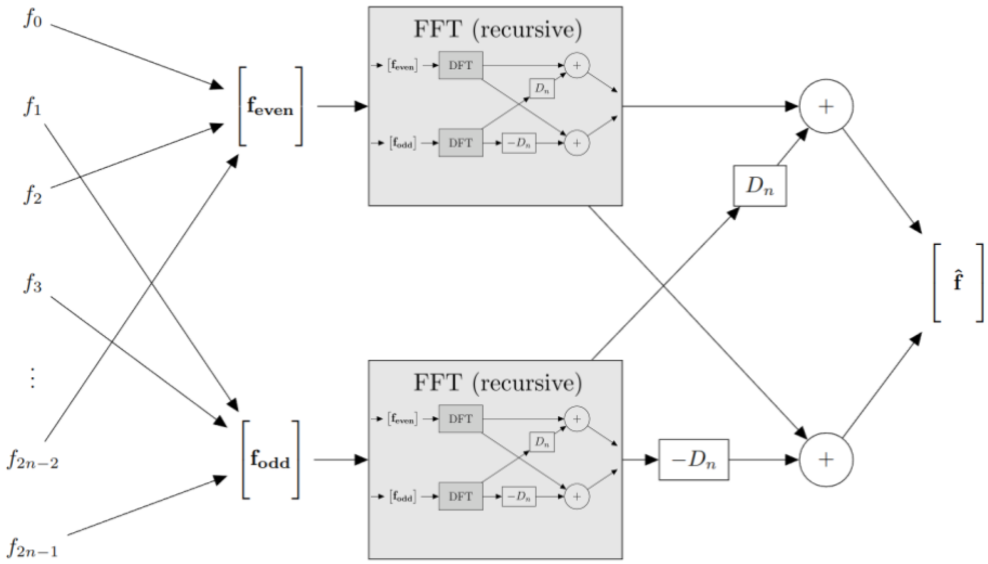


Figure 5: A single step of computation for the Cooley-Tukey FFT on samples f. The input data is split into two components f_{even} and f_{odd}, over which the DFT is computed. This is done by recursive application of the FFT to f_{even} and f_{odd}. The Fourier coefficients $\hat{f}$ are formed from a linear combination of the two DFT results.

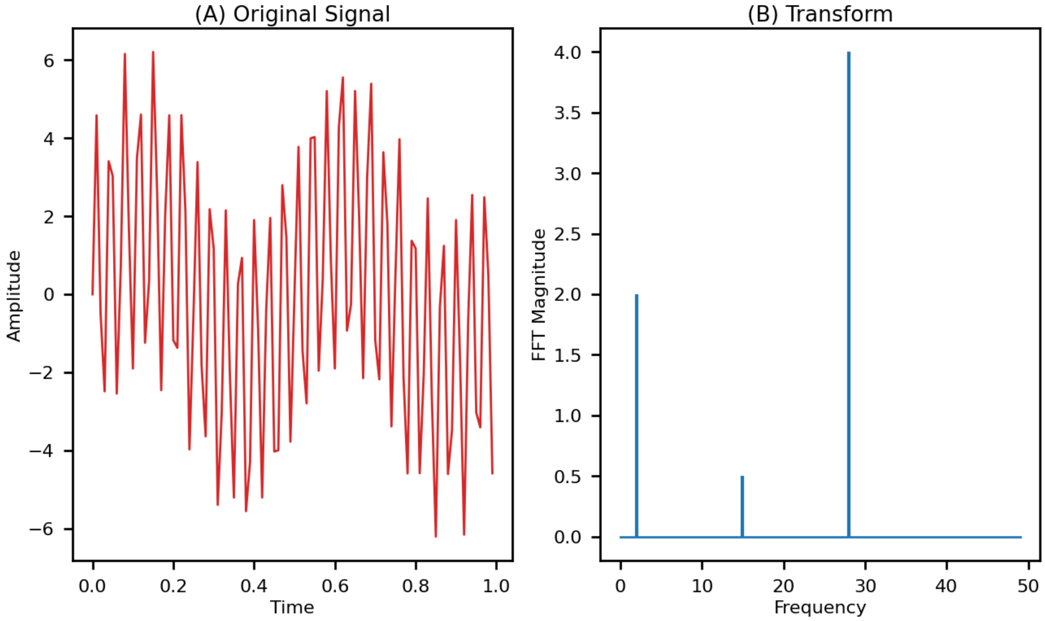


Figure 6: (A) plots the original signal, given as $f(t) = 2\sin(2 \cdot 2\pi t) + 4\sin(28 \cdot 2\pi t) + 0.5\sin(15 \cdot \pi t)$ with 100 equispaced sampling points. (B) illustrates the transform into the frequency domain. The FFT gives the same style of result as the DFT in Figure 3, where the transform picks out the frequencies and magnitudes of the individual sine waves in the original signal.

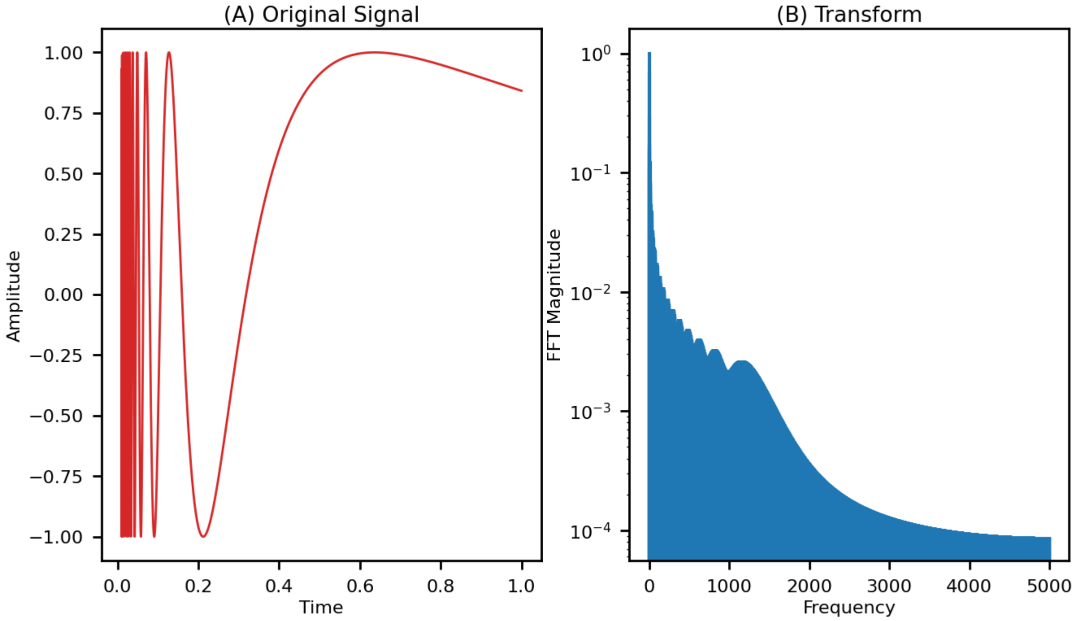


Figure 7: (A) graphs the signal $f(t) = \sin\left(\frac{1}{t}\right)$ from 0.01 to 1.01 with 10,000 equispaced sample points. (B) depicts the frequency domain, which here has much finer resolution due to the higher number of sample points.

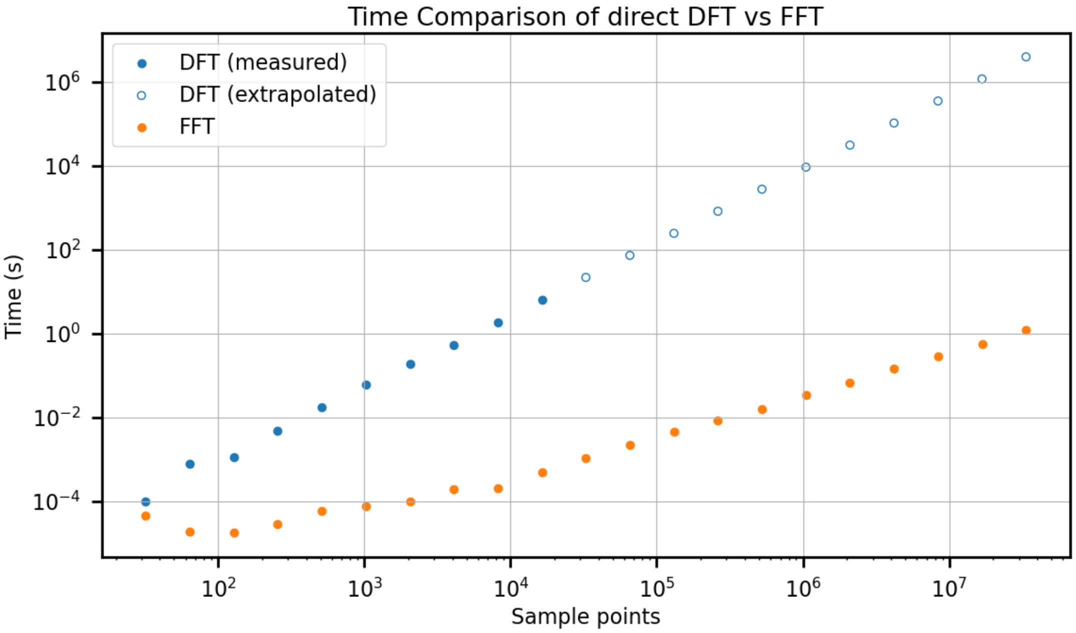


Figure 8: Computational time of the DFT and FFT versus the number of sample points. Beyond 2^{14} samples, the runtime of the DFT grows longer than is practical to measure and is extrapolated. The FFT implementation is from `scipy`.

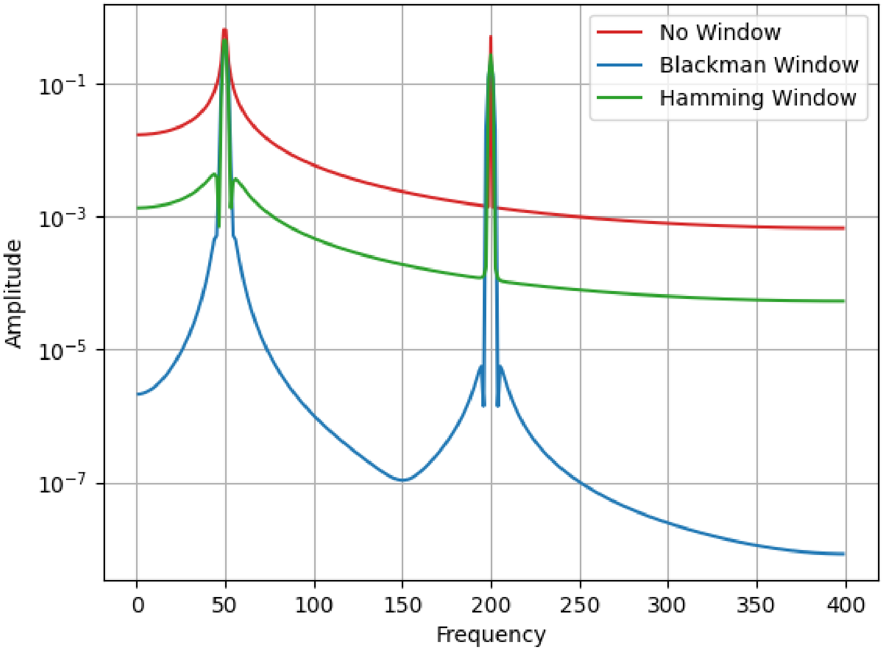


Figure 9: FFT of $f(t) = \sin(100\pi t) + 0.5\sin(400\pi t)$ with no window (red), the Blackman window (blue), and the Hamming window (green). The presence of a window function corresponds to significantly less spectral leakage.

in spectral leakage, where the result of the transform takes on nonzero values around a frequency. We can rectify this by introducing a window function, which acts to reduce the amplitude of the discontinuities at the boundaries of the input data. This is demonstrated in Figure 9, where different window functions distribute the spectral leakage differently.

3 Short-Time Fourier Transform

The FFT gives us the spectrum of a signal over its entire sampled range. Since it is defined only over the frequency domain, it can provide no information as to time variance in a signal's spectrum. Often, we desire to know information about the time domain behavior of a spectrum. To do this, we introduce the Short-Time Fourier Transform (STFT).

3.1 Derivation

As in the DFT, the STFT is defined in terms of a frequency index k. To express time-dependence, the STFT is also defined over a time index $m \in \mathbb{Z}$. We want only the contribution of the sample points near the time of interest, so we center the window function at $n_t = mR$, where the hop size $R \in \mathbb{R}$ is the time step for successive time indices. By a straightforward introduction of the window into the summation formula (5) for the DFT, we have

$$\hat{f}_{k,m} = \sum_{n=0}^{N-1} f(n)g(n - mR)e^{-i\frac{2\pi}{N}kn}. \tag{8}$$

The process of evaluating this sum is illustrated in Figure 10. For a given m, the STFT can be evaluated through the FFT instead of through the direct summation (8), with the same improvements to time com-

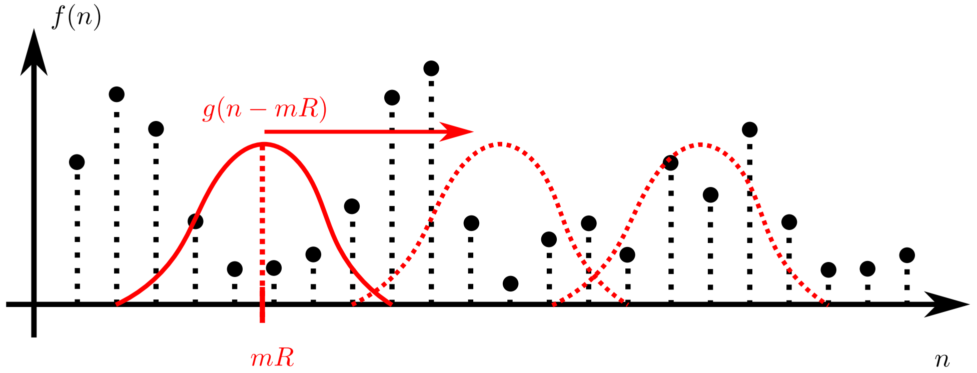


Figure 10: Evaluation of the STFT over a sampled signal $f(n)$ (black). A window function g (red) is moved over the signal to select a short interval of time over which to evaluate the transform. At a time index m, the STFT is the DFT of the signal multiplied by the window, given by Equation (8).

plexity as in the previous section. For N sample points and M time indices, computation of the STFT has complexity $\mathcal{O}(MN \log N)$.

3.1.1 Window Functions

Window functions are not only used in the standard FFT. Indeed, because the signal is further broken down in the STFT, window functions become essential to the reliability of the results. Recall that window functions are used to minimize error caused by non-commensurate periodic functions. When we subdivide a signal, we lose any guarantee of the endpoints aligning with the function's period, and as such, window functions become a necessity. Below, we introduce several key window functions, which are plotted with their Fourier transforms in Figure 11. Applying any window function changes the frequency content of the signal, an effect called spectral leakage. Different choices of window functions distribute this leakage differently, making them appropriate for different applications. The Fourier transforms in Figure 11(b) illustrate how each window function distributes spectral leakage.

We begin with the Hamming window function [8], which has the formula

$$w(n) = 0.54 - 0.46 \cos(2\pi n/N), \ 0 \leq n \leq N, \tag{9}$$

where $N + 1$ is the length of the window.

The Hamming window is best known for reliability with most functions because it has good frequency selectivity for larger signals. From Figure 11(b), the Hamming window has a relatively narrow main lobe, which gives it good selectivity for signals that are close in frequency. The selection of the coefficients 0.54 and 0.46 in (9) suppresses the side lobes immediately next to the main lobe. However, the side lobes have relatively slow roll-off, which can limit detection of signals with a wide range of amplitudes.

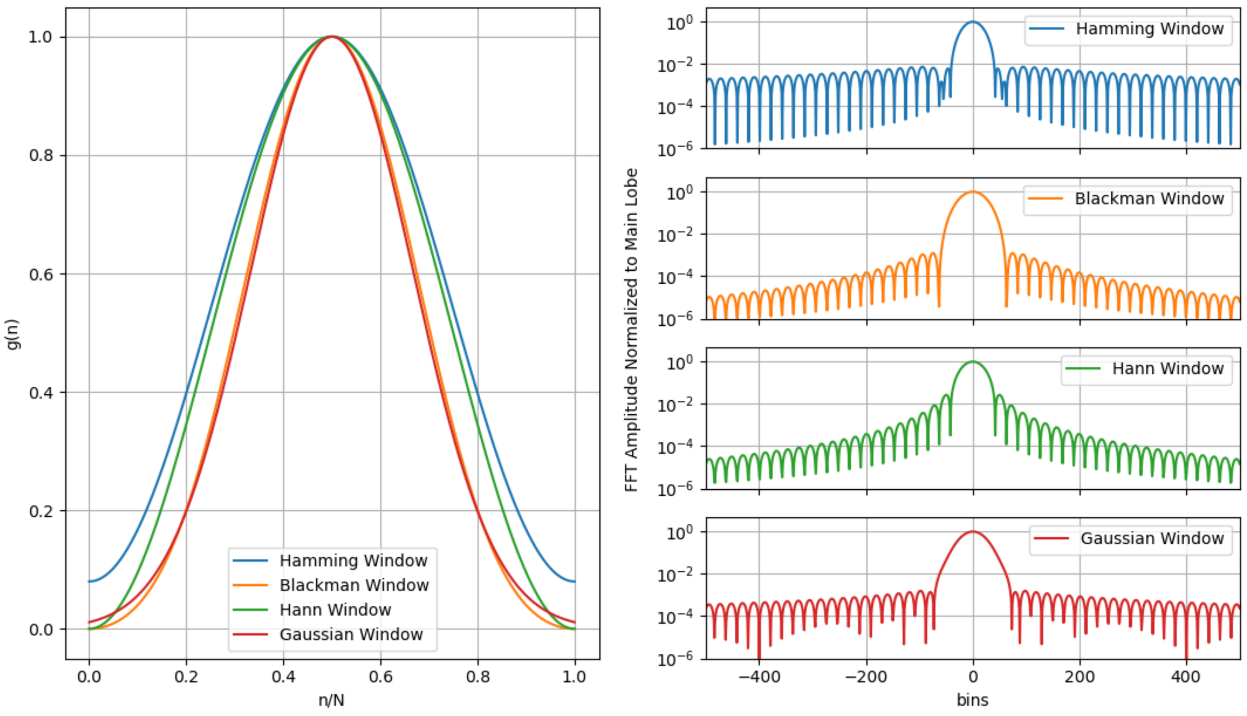
Another common function is the Blackman window [6], which takes the form

$$w(n) = 0.42 - 0.5 \cos\left(\frac{2\pi n}{L-1}\right) + 0.08 \cos\left(\frac{4\pi n}{L-1}\right), \ 0 \leq n \leq M-1, \tag{10}$$

where L is the length of the window, $M = N/2$ for even N, and $M = (N+1)/2$ for odd N. The Blackman window has faster sidelobe roll-off but a fatter main lobe than the Hamming window, making it less selective in frequency but more sensitive in amplitude.

For a middle ground, we introduce the Hann window [9] with formula

$$w(n) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi n}{N}\right)\right). \tag{11}$$



(a)

(b)

Figure 11: A comparison of several notable window functions used in the STFT, plotting (a) the window functions themselves and (b) their Fourier transforms. The Gaussian window selected has $\sigma = 0.2$. We note the properties of the various windows are visually demonstrated here.

The Hann window is notable for maintaining a reliable middle ground between the Hamming and Blackman window, with lower sidelobes than Hamming and a narrower main lobe than Blackman. If nothing is known about the input signal, the Hann function is a safe choice.

The final window function we analyze is the Gaussian [7], which is the most complicated. It can be expressed as

$$w(n) = e^{-\frac{1}{2}\left(\alpha\frac{2n}{L-1}\right)^2} = e^{-\frac{n^2}{2\sigma^2}} \tag{12}$$

where $-(L-1)/2 \leq n \leq (L-1)/2$. The standard deviation of the Gaussian is σ and can be expressed as $\sigma = \frac{L-1}{2\alpha}$.

The Gaussian is most notable for its superior joint time-frequency resolution. This originates from the fact that a pure Fourier transform of a Gaussian window is a Gaussian; a bound on the resolution is given in Theorem 3.

3.1.2 Parameters

Besides the windowing function itself, there are three key parameters to consider when calculating a STFT: window size, window overlap, and number of FFT points.

The number of FFT points is the most straightforward of these three parameters. As discussed in Section 2.1.3, the Cooley-Tukey FFT algorithm, which takes the time complexity of the DFT from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$, requires a number of points that is a power of two. The STFT algorithm, therefore, performs best when each individual FFT taken is sampled on $k = 2^n$ points, $n \in \mathbb{N}$ [1]. The number of FFT points, in turn, determines the frequency domain resolution of the STFT.

Whereas the number of FFT points determines the *frequency-domain resolution*, the window size determines the *time-domain resolution.* Numerically, the window size corresponds to the width (in number of samples) of the windowing function used in the STFT; therefore, a smaller window size means that each FFT is taken for a shorter duration of time, while a larger window size means that each FFT is taken for a longer duration of time. Localizing a signal in the time domain requires a small window size; additionally, a quicker-varying signal will requier a smaller window size to properly detect changing frequencies.

The final parameter is the window overlap. Since the STFT analyzes a time-varying signal, it is necessary to shift the window for each FFT (as shown in Figure 10); however, the amount to move this window by is once again dependent on the signal itself. A rapidly-changing signal, if analyzed using no window overlap (*i.e.*, the window moves by its width for each FFT), may result in apparent discontinuities wherein the frequencies change suddenly between two neighboring strips of the STFT; including window overlap prevents this phenomenon by ensuring that changes in the constituent frequencies of a signal over time are not erroneously given less weight if they happen to lie only on the endpoints of a window. Note that the window overlap is equivalent to another parameter, the *hop size*, which is a numerical value equal to the number of samples by which to move the window each time; for instance, a 1024-sample window with a window overlap of 0.5 would have a hop size of 512.

An important question arises from this investigation: why not always pick the parameters that lead to the highest possible frequency resolution (*i.e.*, the most FFT points) *and* the highest possible time resolution (*i.e.*, the smallest window size)? Unfortunately, this is impossible thanks to the well-known *Heisenberg Uncertainty Principle.*

In its most well-known form (*i.e.*, in the context of quantum mechanics), the Heisenberg Uncertainty Principle states that the product of the uncertainty in position and the uncertainty in momentum of a measurement of a particle must be greater than or equal to a certain physical constant (half the reduced Planck constant) [14]:

$$\Delta x \Delta p \geq \frac{\hbar}{2} \tag{13}$$

When taking a STFT, the t-ω plane is essentially divided into a series of grid cells (often called atoms), where the width of each cell in the t-direction dictates the time resolution, while the width of each cell in the ω-direction dictates the frequency resolution; the question is then, just how small can these cells be?

Theorem 3. *Let $w \in L^2(\mathbb{R})$ be a windowing function centered around $t = \mu$ with $\|w\| = 1$; furthermore, let the Fourier-transformed windowing function $\hat{w}$ be centered around $\omega = \xi$. The time-frequency resolution of $w(t)$ is represented by a rectangle in the $t - \omega$ plane centered on (μ, ξ) with side lengths σ_t and σ_ω and obeys*

$$\sigma_t \sigma_\omega \geq \frac{1}{2} \tag{14}$$

where σ_t and σ_ω are the standard deviations

$$\sigma_t = \sqrt{\int_{-\infty}^{\infty} (t-\mu)^2 |w(t)|^2 \mathrm{d}t}, \qquad \sigma_\omega = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} (\omega - \xi)^2 |\hat{w}(\omega)|^2 \mathrm{d}\omega}.$$

Proof. See Section 5.2. □

It turns out that the window with the best joint time-frequency resolution is the Gaussian window; the above inequality can be an equality strictly for a Gaussian window and is an inequality for all other windows [1].

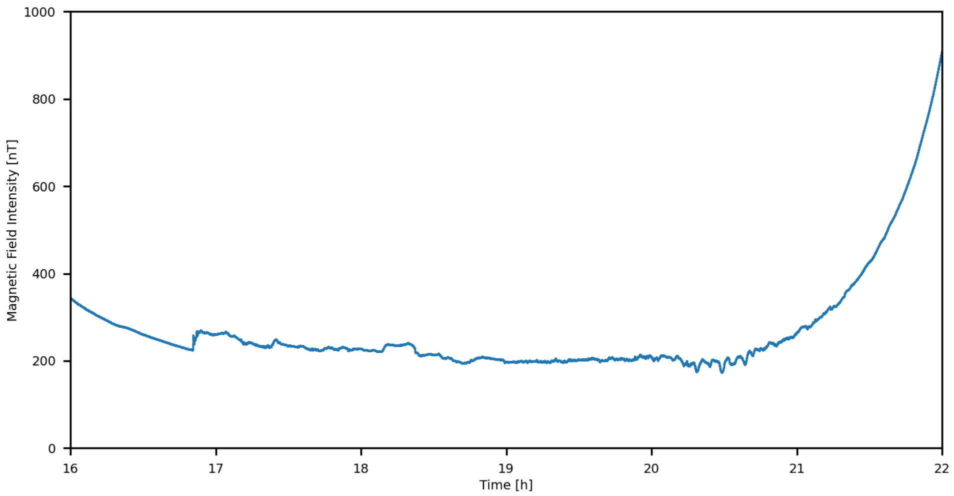


Figure 12: Time series plot of magnetic field intensity over a six-hour period from Van Allen Probe A magnetometer data. This data was taken on Feburary 27th, 2014 at a rate of 64 Hz. We note the presence of an EMIC wave, which makes this data interesting and useful.

3.2 Application

The Earth's magnetosphere is the region of space permeated by the Earth's magnetic field. Interactions between the magnetosphere and either solar wind or Earth's own weather lead to a variety of charged particle events known as plasma waves [13]. Plasma waves can be used to understand space weather and make informed decisions on spaceflight topics including human spaceflight missions, satellite deployments, and space situational awareness. Unfortunately, detecting plasma waves is a complex task often requiring direct measurements of the Earth's magnetic field; to deal with this task, NASA developed the Van Allen Probes [4].

The Van Allen Probes include the EMFISIS suite of instruments, which contains a dedicated magnetometer capable of measuring the Earth's magnetic field strength over time in three dimensions, thereby measuring fluctuations which may be correlated with plasma wave events [4]. Figure 12 shows raw data of the strength of the magnetic field, in nT, for a six-hour period. This data was selected because it has an electromagnetic ion cyclotron (EMIC) wave, a type of plasma wave event, spanning much of its width [15].

Unfortunately, from the data in Figure 12 alone, it is not immediately evident that there is a significant event, as the spike on the latter half of the interval is periodic over the full day of data and may be due to some interaction with a given region of the magnetic field during a certain portion of the orbit. We first try taking the FFT of this signal, as shown in Figure 13.

Once again, looking at solely the data in Figure 13, it is difficult to tell that there is an event of interest; although there appears to be some sort of spike in frequency intensity around 14 Hz, looking at the full range of data once again suggests that this is something other than a plasma wave. Since it is a consistent signal throughout the day, it is likely DC spacecraft noise from some onboard instrument. The problem here is that the signal of interest is not constant over time; however, the FFT analyzes the *entire* signal at once, thereby losing useful information. Finally, we proceed with taking the STFT of this data, using 8192 FFT points, a Hamming window of size 8192, and a hop size of 4096, shown in Figure 14.

This figure shows more apparent structure to the data; indeed, the lighter regions in this spectrogram show the EMIC wave of interest! The dark bands represent gyrofrequencies of different charged particles, thereby confirming that this is, in fact, an EMIC wave.

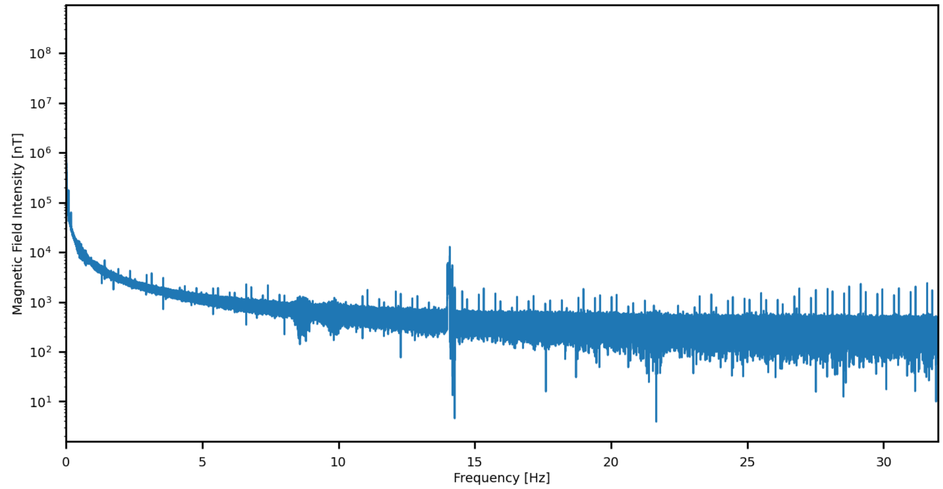


Figure 13: FFT of EMIC wave seen in Van Allen Probe A data. Although this data shows some interesting features, including a spike around 14 Hz which is probably due to spacecraft noise, it lacks (at least visually) the signatures of a plasma wave event. Noticeably, there is no trace of an EMIC wave occuring during this time frame.

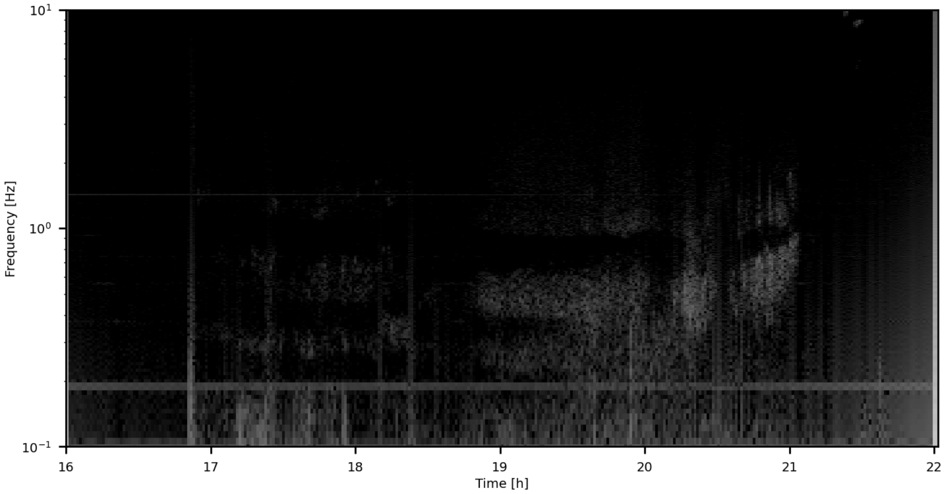


Figure 14: Spectrogram of EMIC wave seen in Van Allen Probe A data. Note the clear visibility of the EMIC wave which was not visible in Figures 12 and 13. The STFT allows researchers to study time-varying signals and detect features not otherwise visible using only time-invariant analysis methods.

However, the choice of FFT points, window type/size, and hop size were chosen here somewhat arbitrarily; how do we go about choosing these parameters optimally?

Unfortunately, it turns out that these decisions are, in general, highly dependent on the signal and desired analysis. There is no single correct answer as to which parameters are optimal; rather, different studies use different paramters to bring out different features of a signal. A study which seeks to highlight rapid changes in frequency of a signal without caring too much about the frequencies themselves, for instance, would likely want to optimize the STFT for time resolution; a study using a known slowly-varying signal for which the frequencies are more important, on the other hand, would likely seek to optimize frequency resolution. The parameters used to create Figure 14 were chosen to visually highlight the EMIC wave; thus, they clearly show the presence of the wave, as well as the charged particle gyrofrequency bands, without sacrificing too much in either time- or frequency-resolution.

4 Conclusions

We begin with a fundamental introduction to the Fast Fourier Transform via a journey that begins with a Fourier approximation, explores the Discrete Fourier Transform for discretely sampled signals, and analyzes the limitations of the DFT before converging on the Fast Fourier Transform as an optimized solution. We then demonstrate the necessity of decoding time-dependent signals and propose the Short Time Fourier Transform as a methodology. Finally, we apply the STFT to scientific data for detection of plasma wave events in the Earth's magnetosphere. Ultimately, we have developed an arsenal of useful signal analysis tools that enable us to better understand, and therefore better utilize, the information contained within various real-world signals.

4.1 Analysis of Results

Our introductory results demonstrate the utility of the Fourier approximation for constructing the DFT and FFT by relying upon the process of complex-valued sums. We prove the minimization of the error in Theorem 1 and extend the idea to discrete signals.

We then show the efficiency of the FFT, a notable improvement over the DFT. Crucially, we prove the time complexity of both algorithms ($\mathcal{O}(n^2)$ vs. $\mathcal{O}(n \log n)$) and demonstrated their runtimes on signals of various lengths in Figure 8. This experiment allowed us to justify the use of the FFT in modern signal processing algorithms.

We show also that the STFT is a necessary tool for analyzing time-varying signals. Note once again the difference between Figures 12, 13, and 14; these figures illustrate a use case for the STFT where the FFT is insufficient and provide valuable insight into otherwise unintelligible data. Furthermore, our results justify the use of various windows based on parameters such as spectral leakage (as illustrated in Figure 9); we also state the optimality of the Gaussian window (with a bound on joint time-frequency resolution given in Theorem 3).

4.2 Future Work

While the STFT is one approach to addressing the lack of time-dependency in the FFT, several other approaches exist. Most notably, the method of wavelets is a useful alternative and would make for an interesting study and comparison with the methods introduced and applied above [1]. Whereas the STFT is restricted to uniformly-sized time-frequency atoms on the t-ω plane, the wavelet method allows for higher frequencies to have smaller atoms than lower frequencies; this, in turn, allows for more detail to be resolved at both higher and lower frequencies [1]. In contrast, the STFT would have to optimize for one or the other.

Furthermore, the FFT has other applications in data analysis, including speech recognition and musical engineering. These applications are also promising, and such an exploration would be beneficial for demonstrating the diverse applications of the FFT and STFT.

References

[1] Ronald L. Alle and Duncan W. Mills. *Signal Analysis: Time, Frequency, Scale, and Structure*. John Wiley & Sons, 2004.

[2] James W Cooley and John W Tukey. "An algorithm for the machine calculation of complex Fourier series". In: *Mathematics of computation* 19.90 (1965), pp. 297–301.

[3] Dataq Instruments. *FFT (Fast Fourier Transform) Waveform Analysis*. 2024. URL: `https://www.dataq.com/data-acquisition/general-education-tutorials/fft-fast-fourier-transform-waveform-analysis.html`.

[4] C. A. Kletzing et al. "The Electric and Magnetic Field Instrument Suite and Integrated Science (EMFISIS) on RBSP". In: *Space Science Review* (2012).

[5] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Elsevier, 2009.

[6] MathWorks. *Blackman Window*. URL: `https://www.mathworks.com/help/signal/ref/blackman.html`.

[7] MathWorks. *Gaussian Window*. URL: `https://www.mathworks.com/help/signal/ref/gausswin.html`.

[8] MathWorks. *Hamming Window*. URL: `https://www.mathworks.com/help/signal/ref/hamming.html`.

[9] MathWorks. *Hann Window*. URL: `https://www.mathworks.com/help/signal/ref/hann.html`.

[10] Peter J. Olver and Chehrzad Shakiban. *Applied Linear Algebra*. Springer International Publishing, 2018.

[11] Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals & systems*. Pearson Education, 1997.

[12] NOAA Regional Climate Centers & The National Weather Service. *Boulder Climate*. `https://www.weather.gov/wrh/climate?wfo=bou`. Accessed: 18 Nov 2024. 2024.

[13] Richard Mansergh Thorne. "Radiation belt dynamics: The importance of wave-particle interactions". In: *Geophysical Research Letters* (2010).

[14] Biao Wu. *Quantum Mechanics: A Concise Introduction*. Peking University Press, 2022.

[15] Zheng Xiang et al. "Understanding the Mechanisms of Radiation Belt Dropouts Observed by Van Allen Probes". In: *Journal of Geophysical Research: Space Physics* (2017).

5 Appendix A: Proofs of Theorems

5.1 Proof of Theorem 1: Truncated Fourier Coefficients Minimize RMSE

Proof. Observe that we can expand E as

$$\begin{aligned} E &= \int_{-\pi}^{\pi} \left| f(t) - \sum_{n=-N}^{N} c_n e^{int} \right|^2 \mathrm{d}t \\ &= \int_{-\pi}^{\pi} \left(f(t) - \sum_{n=-N}^{N} c_n e^{int} \right) \overline{\left(f(t) - \sum_{n=-N}^{N} c_n e^{int} \right)} \mathrm{d}t \\ &= \int_{-\pi}^{\pi} \left(|f(t)|^2 - f(t) \sum_{n=-N}^{N} \overline{c_n e^{int}} - \overline{f(t)} \sum_{n=-N}^{N} c_n e^{int} + \sum_{n=-N}^{N} \sum_{m=-N}^{N} c_n \overline{c_m} e^{int} \overline{e^{imt}} \right) \mathrm{d}t. \end{aligned} \tag{15}$$

Notice that our set of basis functions $\{e^{imt}\}$ is orthogonal under the complex L^2 inner product, that is,

$$\langle e^{imt}, e^{int} \rangle = \int_{-\pi}^{\pi} e^{imt} \overline{e^{int}} \mathrm{d}t = \int_{-\pi}^{\pi} e^{i(m-n)t} \mathrm{d}t = 2\pi \delta_{nm}.$$

Thus, the rightmost sum of (15) simplifies and we have

$$\begin{aligned} E &= \int_{-\pi}^{\pi} \left(|f(t)|^2 - f(t) \sum_{n=-N}^{N} \overline{c_n e^{int}} - \overline{f(t)} \sum_{n=-N}^{N} c_n e^{int} + \sum_{n=-N}^{N} c_n \overline{c_n} \right) \mathrm{d}t \\ &= \int_{-\pi}^{\pi} |f(t)|^2 \mathrm{d}t + \sum_{n=-N}^{N} \left(- \int_{-\pi}^{\pi} \left(f(t) \overline{c_n e^{int}} + \overline{f(t)} c_n e^{int} \right) \mathrm{d}t + 2\pi c_n \overline{c_n} \right) \\ &= \int_{-\pi}^{\pi} |f(t)|^2 \mathrm{d}t + \sum_{n=-N}^{N} \left(- \int_{-\pi}^{\pi} \left(f(t) \overline{c_n} e^{-int} + \overline{f(t)} c_n e^{int} \right) \mathrm{d}t + 2\pi c_n \overline{c_n} \right) \end{aligned}$$

We can factor the sum on the right to get

$$\begin{aligned} E &= \int_{-\pi}^{\pi} |f(t)|^2 \mathrm{d}t + \sum_{n=-N}^{N} \left[2\pi \left(c_n - \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} \mathrm{d}t \right) \left(\overline{c_n} - \frac{1}{2\pi} \int_{-\pi}^{\pi} \overline{f(t)} e^{int} \mathrm{d}t \right) - \frac{1}{2\pi} \int_{-\pi}^{\pi} |f(t)|^2 \mathrm{d}t \right] \\ &= \int_{-\pi}^{\pi} |f(t)|^2 \mathrm{d}t + \sum_{n=-N}^{N} \left[2\pi \left| c_n - \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} \mathrm{d}t \right|^2 - \frac{1}{2\pi} \int_{-\pi}^{\pi} |f(t)|^2 \mathrm{d}t \right]. \end{aligned}$$

Notice that the only term in E that depends on c_n is $\left| c_n - \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} \mathrm{d}t \right|^2$, which is non-negative and is minimized when

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} \mathrm{d}t.$$

Thus, E is minimized under this choice. □

5.2 Proof of Theorem 3

Proof (adapted from [5]). We prove here the simplified case in which $\mu = 0$ and $\xi = 0$.

Assume that $w \in L^2$ with $\|w\| = 1$ and $\lim_{|t| \to \infty} \sqrt{t} w(t) = 0$. Observe that

$$\sigma_t^2\sigma_\omega^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty}|tw(t)|^2\mathrm{d}t\int_{-\infty}^{\infty}|\omega\hat{w}(\omega)|^2\mathrm{d}\omega$$

Since $i\omega\hat{w}(\omega)$ is the Fourier transform of $w'(t)$,

$$\sigma_t^2\sigma_\omega^2 = \int_{-\infty}^{\infty}|tw(t)|^2\mathrm{d}t\int_{-\infty}^{\infty}|w'(t)|^2\mathrm{d}t$$

by the Plancherel identity. Invoking the Cauchy-Schwarz inequality,

$$\begin{aligned}\sigma_t^2\sigma_w^2 &\geq \left(\int_{-\infty}^{\infty}|tw'(t)w^*(t)|\mathrm{d}t\right)^2\\ &\geq \left(\int_{-\infty}^{\infty}\frac{t}{2}\left(w'(t)w^*(t)+w'^*(t)w(t)\right)\mathrm{d}t\right)^2\\ &\geq \frac{1}{4}\left(\int_{-\infty}^{\infty}t\left(|w(t)|^2\right)'\,\mathrm{d}t\right)^2\end{aligned}$$

Since $\lim\limits_{|t|\to\infty}\sqrt{t}w(t) = 0$,

$$\sigma_t^2\sigma_\omega^2 \geq \frac{1}{4}\left(\int_{-\infty}^{\infty}|w(t)|^2\mathrm{d}t\right)^2$$

Finally, since $\|w\| = 1$,

$$\sigma_t^2\sigma_\omega^2 \geq \frac{1}{4}$$

and square-rooting both sides yields the desiring result:

$$\sigma_t\sigma_\omega \geq \frac{1}{2}$$

□

Note: We omit here the statements and proofs of the Plancherel identity and the Cauchy-Schwarz inequality; for the Plancherel identity, we refer the reader to [5], Theorem 2.3, while for the Cauchy-Schwarz inequality, we refer the reader to [10], Theorem 3.5.

6 Appendix B: Code and Data

6.1 Code

All figures in this report were created using code available at https://github.com/ErickWhiteDev/APPM-4600-Final-Project. All code in this repository makes use of Python 3 and free and open-source packages; all data used is publicly available (see Section 6.2 and the `README.md` file in the repository for more details). The code can be run as-is (assuming the proper packages are installed and the data is downloaded).

6.2 Data

The data for this report were provided by the National Weather Service (Figure 1) and by the University of Iowa (Figures 12, 13, and 14). All data used is publicly available through the National Weather Service website at https://www.weather.gov/wrh/climate?wfo=bou and through the University of Iowa website at https://emfisis.physics.uiowa.edu/data/index.

Figure 1 made use of the temperature data for Boulder, CO over the period 1 January, 2022 to 31 December, 2023.

Figures 12, 13, and 14 made use of the L2 RBSP-A magnetometer data from February 27th, 2014.