



Bangladesh University of Engineering and Technology

CSE 300

Report on

**Fast Fourier Transform (FFT):
A Revolutionizing Algorithm**

Authors:

Tanvirul Islam Turad (2005011)

Tanvir Hossain (2005014)

MD. Jakaria Hossain

Date: March 9, 2024

Contents

1	Applications	2
1.1	FFT in signal processing	2
1.2	FFT in Communications	3
1.3	FFT in Instrumentation	3
1.4	FFT in Control Systems	3
1.5	FFT in Geophysics and Seismology	3
1.6	FFT in Medical Imaging	4
1.7	FFT in Astrophysics and Cosmology	4
1.8	FFT in Computer Science	4
1.8.1	All possible sums	5
1.8.2	All possible scalar products	5
1.8.3	Two stripes	5
1.8.4	String matching	5
1.8.5	String matching with wildcards	6
2	Conclusion	7

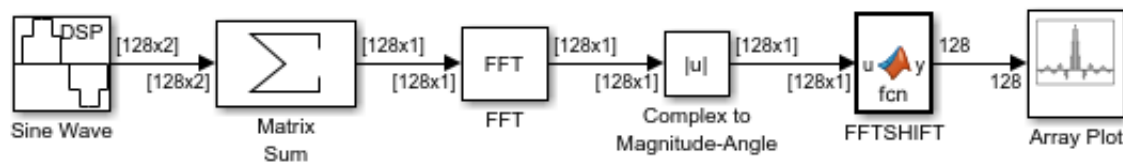
Chapter 1

Applications

FFT is the core component of frequency domain analysis, also known as spectrum analysis, in signal processing. It is widely used in data compression, spectral estimation, signal filtering, and other applications. Analysis in both the frequency and time domains can be done simultaneously using FFT variations like the short-time Fourier transform. These methods can be applied to a wide range of signals, including communication, radar, audio and speech, and other sensor data signals. Additionally, FFT is occasionally employed as a transitional step before more sophisticated signal processing methods. FFT is used for filtering and picture compression in image processing. In mathematics and physics, partial differential equations (PDEs) are also solved using FFT.

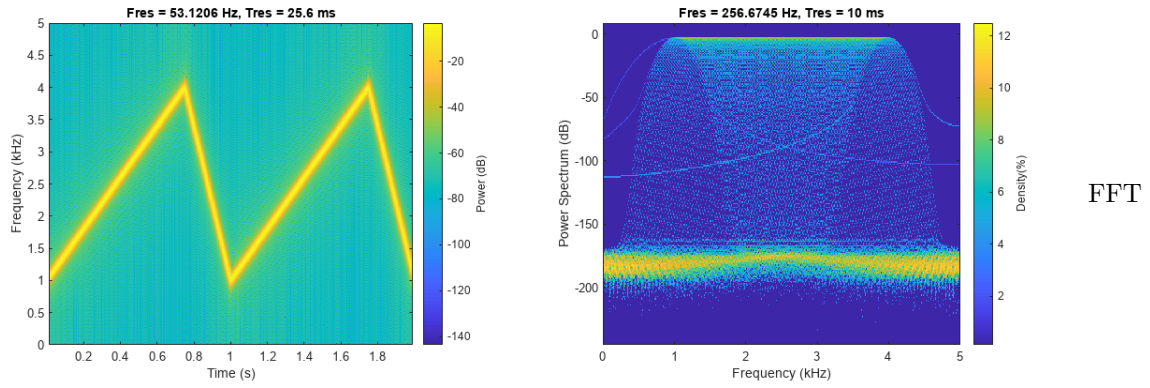
1.1 FFT in signal processing

FFT is most frequently used to convert signals from the time domain to the frequency domain.



Copyright 2018 The MathWorks, Inc.

The fundamental algorithm used for time-frequency based signal analysis in many simulation programs, such as MATLAB, is FFT. The FFT algorithm was utilized by MATLAB's Signal Processing *Toolbox*TM to display spectrograms and persistent spectrums, which are time-frequency views that display the proportion of a signal's duration that a certain frequency is present.



helps in radar signal analysis, target detection, and tracking. FFT aids in underwater acoustics, target detection, and classification.

1.2 FFT in Communications

- Digital communication systems: FFT is utilized in modulation and demodulation schemes such as OFDM (Orthogonal Frequency Division Multiplexing) used in Wi-Fi, LTE, and digital television.
- Channel estimation and equalization: FFT helps in estimating the channel characteristics and compensating for distortions in communication channels.

1.3 FFT in Instrumentation

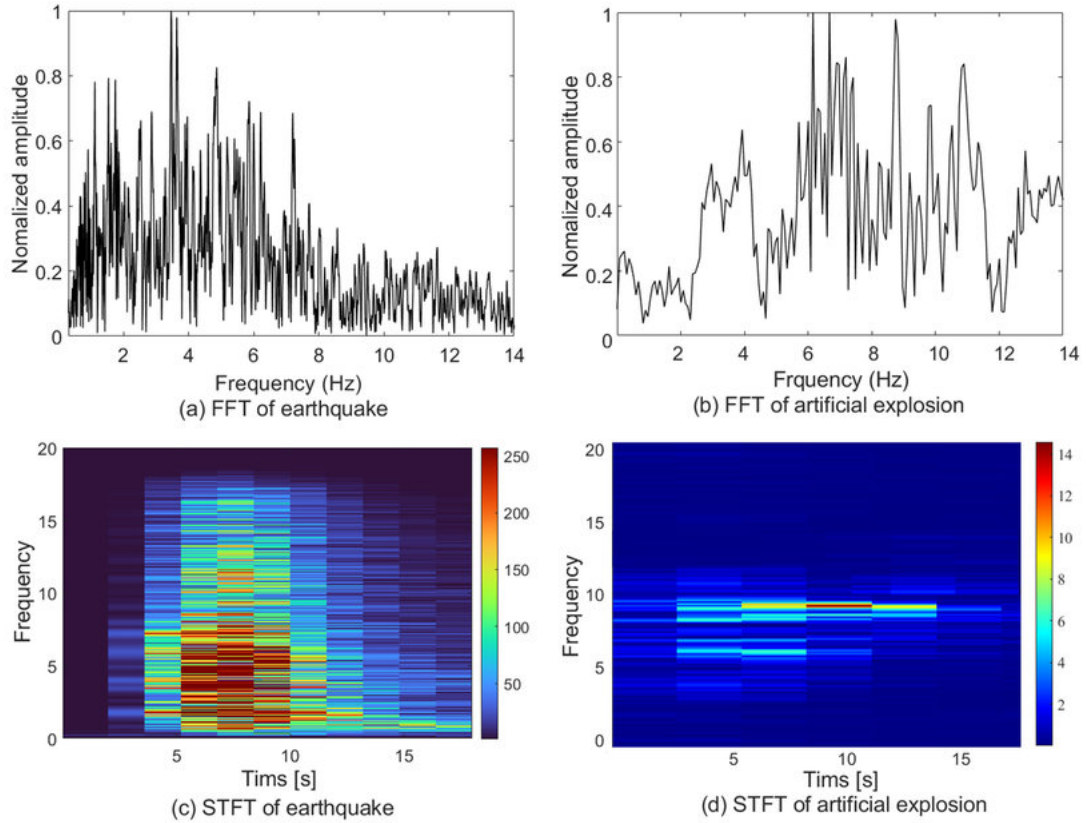
- Spectrum analysis: FFT is employed in spectrum analyzers for measuring frequency components of signals.
- Biomedical signal analysis: FFT is used in the analysis of EEG (Electroencephalography), ECG (Electrocardiography), and other biomedical signals for diagnosis and monitoring.

1.4 FFT in Control Systems

- Control system analysis: FFT is used in identifying system dynamics, frequency response analysis, and stability analysis.
- Control system design: FFT helps in designing controllers and compensators for systems with frequency-dependent characteristics.

1.5 FFT in Geophysics and Seismology

- Earthquake analysis: FFT is used in seismic data processing for earthquake detection, location, and magnitude estimation.



- Exploration geophysics: FFT assists in the interpretation of seismic data for oil and gas exploration.

1.6 FFT in Medical Imaging

- MRI (Magnetic Resonance Imaging): FFT is utilized in MRI for image reconstruction from raw data acquired during scans.
- CT (Computed Tomography): FFT is used in image reconstruction algorithms to convert raw projection data into cross-sectional images.

1.7 FFT in Astrophysics and Cosmology

- Radio astronomy: FFT is employed in processing radio signals received from celestial objects for analysis and imaging.
- Cosmology: FFT assists in analyzing cosmic microwave background radiation data for studying the early universe.

1.8 FFT in Computer Science

DFT, a special variation of FFT can be used in a huge variety of other problems, which at the first glance have nothing to do with multiplying polynomials.

1.8.1 All possible sums

We are given two arrays $a[]$ and $b[]$. We have to find all possible sums $a[i] + b[j]$, and for each sum count how often it appears.

For example for $a = [1, 2, 3]$ and $b = [2, 4]$ we get: then sum 3 can be obtained in 1 way, the sum 4 also in 1 way, 5 in 2, 6 in 1, 7 in 1.

We construct for the arrays a and b two polynomials A and B . The numbers of the array will act as the exponents in the polynomial ($a[i] \Rightarrow x^{a[i]}$); and the coefficients of this term will be how often the number appears in the array.

Then, by multiplying these two polynomials in $O(n \log n)$ time, we get a polynomial C , where the exponents will tell us which sums can be obtained, and the coefficients tell us how often. To demonstrate this on the example:

$$(1x^1 + 1x^2 + 1x^3)(1x^2 + 1x^4) = 1x^3 + 1x^4 + 2x^5 + 1x^6 + 1x^7$$

1.8.2 All possible scalar products

We are given two arrays $a[]$ and $b[]$ of length n . We have to compute the products of a with every cyclic shift of b .

We generate two new arrays of size $2n$: We reverse a and append n zeros to it. And we just append b to itself. When we multiply these two arrays as polynomials, and look at the coefficients $c[n-1]$, $c[n]$, ..., $c[2n-2]$ of the product c , we get:

$$c[k] = \sum_{i+j=k} a[i]b[j]$$

And since all the elements $a[i] = 0$ for $i \geq n$:

$$c[k] = \sum_{i=0}^{n-1} a[i]b[k-i]$$

It is easy to see that this sum is just the scalar product of the vector a with the $(k - (n - 1))$ -th cyclic left shift of b . Thus these coefficients are the answer to the problem, and we were still able to obtain it in $O(n \log n)$ time. Note here that $c[2n-1]$ also gives us the n -th cyclic shift but that is the same as the 0-th cyclic shift so we don't need to consider that separately into our answer.

1.8.3 Two stripes

We are given two Boolean stripes (cyclic arrays of values 0 and 1) a and b . We want to find all ways to attach the first stripe to the second one, such that at no position we have a 1 of the first stripe next to a 1 of the second stripe.

The problem doesn't actually differ much from the previous problem. Attaching two stripes just means that we perform a cyclic shift on the second array, and we can attach the two stripes, if scalar product of the two arrays is 0.

1.8.4 String matching

We are given two strings, a text T and a pattern P , consisting of lowercase letters. We have to compute all the occurrences of the pattern in the text.

We create a polynomial for each string ($T[i]$ and $P[i]$ are numbers between 0 and 25 corresponding to the 26 letters of the alphabet):

$$A(x) = a_0x^0 + a_1x^1 + \dots + a_{n-1}x^{n-1}, \quad n = |T|$$

with

$$a_i = \cos(\alpha_i) + i \sin(\alpha_i), \quad \alpha_i = \frac{2\pi T[i]}{26}.$$

And

$$B(x) = b_0x^0 + b_1x^1 + \dots + b_{m-1}x^{m-1}, \quad m = |P|$$

with

$$b_i = \cos(\beta_i) - i \sin(\beta_i), \quad \beta_i = \frac{2\pi P[m-i-1]}{26}.$$

Notice that with the expression $P[m-i-1]$ explicitly reverses the pattern.

The $(m-1+i)$ th coefficients of the product of the two polynomials $C(x) = A(x) \cdot B(x)$ will tell us, if the pattern appears in the text at position i .

$$c_{m-1+i} = \sum_{j=0}^{m-1} a_{i+j} \cdot b_{m-1-j} = \sum_{j=0}^{m-1} (\cos(\alpha_{i+j}) + i \sin(\alpha_{i+j})) \cdot (\cos(\beta_j) - i \sin(\beta_j))$$

with $\alpha_{i+j} = \frac{2\pi T[i+j]}{26}$ and $\beta_j = \frac{2\pi P[j]}{26}$

If there is a match, then $T[i+j] = P[j]$, and therefore $\alpha_{i+j} = \beta_j$. This gives (using the Pythagorean trigonometric identity):

$$\begin{aligned} c_{m-1+i} &= \sum_{j=0}^{m-1} (\cos(\alpha_{i+j}) + i \sin(\alpha_{i+j})) \cdot (\cos(\alpha_{i+j}) - i \sin(\alpha_{i+j})) \\ &= \sum_{j=0}^{m-1} \cos^2(\alpha_{i+j}) + \sin^2(\alpha_{i+j}) = \sum_{j=0}^{m-1} 1 = m \end{aligned}$$

If there isn't a match, then at least a character is different, which leads that one of the products $a_{i+1} \cdot b_{m-1-j}$ is not equal to 1, which leads to the coefficient $c_{m-1+i} \neq m$.

1.8.5 String matching with wildcards

This is an extension of the previous problem. This time we allow that the pattern contains the wildcard character `*`, which can match every possible letter. E.g. the pattern $a * c$ appears in the text `abccaacc` at exactly three positions, at index 0, index 4 and index 5.

We create the exact same polynomials, except that we set $b_i = 0$ if $P[m-i-1] = *$. If x is the number of wildcards in P , then we will have a match of P in T at index i if $c_{m-1+i} = m - x$.

More problems to explore

[POLYMUL - Polynomial Multiplication](#)
[MAXMATCH - Maximum Self-Matching](#)
[ADAMATCH - Ada and Nucleobase](#)
[Yet Another String Matching Problem](#)
[Lightsabers \(hard\)](#)
[Running Competition](#)
[Dasha and cyclic table](#)
[Centroid Probabilities](#)
[Expected Number of Customer](#)
[Power Sum](#)
[A+B Problem](#)
[K-Inversions](#)

Chapter 2

Conclusion

In conclusion, the Fast Fourier Transform (FFT) technique is a cornerstone in signal processing and beyond. Its extraordinary efficiency in computing the Discrete Fourier Transform (DFT) has enabled advances in a wide range of applications.

Throughout this report, we have explored the diverse applications of the FFT algorithm, spanning signal processing, communications, instrumentation, control systems, geophysics, medical imaging, finance, astrophysics, and cosmology. From audio and image processing to radar and sonar signal analysis, from medical diagnostics to seismic data interpretation, the FFT algorithm finds itself at the heart of countless innovative solutions.

The versatility of the FFT algorithm supports vital technologies like digital communication networks, medical imaging equipment, and astronomical observatories in addition to making signal analysis and modification easier. Its widespread use highlights its significance as a vital instrument in contemporary research and engineering.

Looking ahead, as technology continues to evolve, the FFT algorithm is poised to remain an indispensable asset, enabling further breakthroughs in fields ranging from data science to space exploration. Its impact on our understanding of the world and our ability to manipulate and interpret signals is undeniable, making it a subject of continued research and application for years to come.

In essence, the FFT algorithm exemplifies the power of computational methods to unlock new insights and capabilities across a broad spectrum of disciplines, shaping the landscape of scientific inquiry and technological innovation.