OLD DOMINION UNIVERSITY

Fourier Series and Transform

A. Godunov

1. Introductions
2. Fourier series and Fourier transform (brief review)
3. The Discrete Fourier Transform (DFT)
4. The Fast Fourier Transform (FFT)
5. Filtering noisy signals

1

Part 1:

Introductions

2

Familiar equations

Fourier series

$$f(f) = a_0 + \sum_{n=1}^{\infty}[a_n \cos(nx) + b_n \sin(nx)]$$

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(f)\cos kx\, dx \qquad (k = 0, 1, 2, \dots)$$

$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(f)\sin kx\, dx \qquad (k = 1, 2, \dots)$$

3

3

Familiar equations

Fourier transform:

$$F(\omega) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(x)e^{i\omega x}\, dx$$

$$f(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} F(\omega)e^{-i\omega x}\, d\omega$$

4

4

Three important parts

1. Why it’s so important?
2. Understanding Fourier series and Fourier Transform
3. Implementation

5

5

Jean-Baptiste Joseph Fourier (1768 –1830)

Jean-Baptiste Joseph Fourier was a French mathematician and physicist born in Auxerre and best known for initiating the investigation of Fourier series, which eventually developed into Fourier analysis.

He also made significant contributions to the study of heat transfer and vibrations.

His work has had a profound impact on many fields, including physics, engineering, and mathematics.

6

What is so special about it?

Fourier analysis is a powerful mathematical tool widely used in various fields of science and engineering *to analyze and process signals, and functions*.

It involves decomposing a complex signal into simpler components, *typically* represented as a sum of sinusoids (sines and cosines) waves with different frequencies, amplitudes, and phases.

This decomposition is extremely useful for identifying and isolating the frequency content within a signal.

7

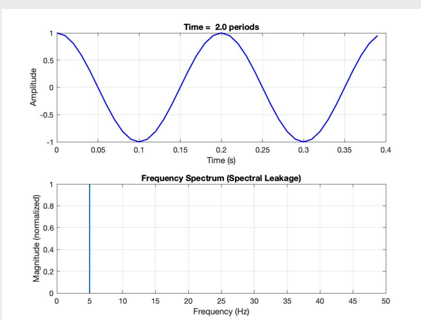
Physics and Mathematics

- Fourier analysis describes the relationship between time-domain and frequency-domain information, helping physicists understand wave-particle behavior.
- It is also used in analyzing vibrations, acoustics, and optical phenomena, which are inherently periodic or wave-like.
- Solving ODE and PDE

Example: In optics, the Fourier transform explains how diffraction patterns arise from the physical structure of objects.

8

Physics (example)



9

Application: Signal Processing

- Frequency Decomposition: Fourier analysis helps break down signals into constituent frequencies, making it easier to analyze specific components.
- Noise Reduction: By filtering out unwanted frequencies (like high-frequency noise), Fourier analysis can help clean up signals.
- Compression: Techniques like JPEG and MP3 compression rely on Fourier analysis to represent data in a more compact form.

Example: In audio processing, the Fourier transform helps equalize or remove noise by filtering certain frequency bands.

10

Application: Image Processing

- Fourier transforms can be applied to images to emphasize or suppress certain frequency components, aiding in tasks like edge detection, pattern recognition, and image filtering.
- In digital imaging, the analysis can help reduce blurring and sharpen images by filtering out unwanted frequencies.

Example: Medical imaging (e.g., MRI, CT scans), heavily relies on the Fourier transform in its image processing.

11

Communication Systems

- In wireless communication and network systems, Fourier analysis aids in understanding how signals propagate, interact, and overlap.
- Modulation and demodulation processes in radio, television, and digital communication depend on Fourier concepts.

Example: Fourier transform is used in designing antennas and analyzing their frequency response.

12

Mechanical and Civil Engineering

- Vibration Analysis: Identifies natural frequencies of structures and machines to prevent resonance and optimize performance.
- Structural Health Monitoring: Detects changes in a structure by analyzing frequency shifts caused by damage or wear.

Example: In bridge engineering, Fourier analysis identifies vibration frequencies to assess structural integrity.

13

13

Medical and Biological Sciences

- Medical Imaging: MRI and CT use Fourier transform to reconstruct spatial images from frequency-domain data.
- Neuroscience: Analyzes brain activity by decomposing EEG or fMRI signals into frequency components.
- Bioinformatics: Examines periodic patterns in DNA or protein sequences.

Example: In EEG analysis, the Fourier transform identifies the frequency bands associated with different brain states (e.g., sleep or attention).

14

14

How Fourier Analysis Works

The most common forms of Fourier analysis include:

- Fourier Series: Used for periodic signals, breaking them down into sines and cosines.
- Fourier Transform (FT): Extends this analysis to non-periodic signals.
- Discrete Fourier Transform (DFT): Applicable in digital settings, like digital signal processing.
- Fast Fourier Transform (FFT): A computationally efficient way to perform DFT, crucial for real-time applications.

15

15

Our life without Fourier Analysis

Without the Fourier transform, modern life would look vastly different, as it underpins much of the technology and scientific understanding we rely on daily

- Communication and media: no internet or cell phones, poor-quality audio and video …
- Healthcare: no MRI or CT scans, ultrasound and EEG analysis would be less effective
- Science and engineering: slower scientific progress, technologies like GPS, and sonar would be rudimentary
- Entertainment: no advanced animations, no streaming platforms - data compression needed for Netflix, YouTube, etc., would not work
- …

16

16

Part 2A:

Fourier Series (brief review)

17

Reference

see “Pragmatic Mathematics for Scientists and Engineers” by Alexander Godunov and John A. Adam, World Scientific (2024)

Chapter 12 “Fourier series and transform”, Sections 7.1-7.3

18

18

Part 2B:

Fourier Transform (brief review)

19

Reference

see “Pragmatic Mathematics for Scientists and Engineers” by Alexander Godunov and John A. Adam, World Scientific (2024)

Chapter 12 “Fourier series and transform”, Section 7.4

20

20

Part 3:

Calculating Fourier series

21

Fourier series

Calculating a Fourier series involves determining the coefficients a_0, a_n, b_n that define the series.

$$f(f) = a_0 + \sum_{n=1}^{\infty} [a_n \cos(nx) + b_n \sin(nx)]$$

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(f) \cos kx \, dx \qquad (k = 0, 1, 2, \dots)$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(f) \sin kx \, dx \qquad (k = 1, 2, \dots)$$

22

22

Brief outline

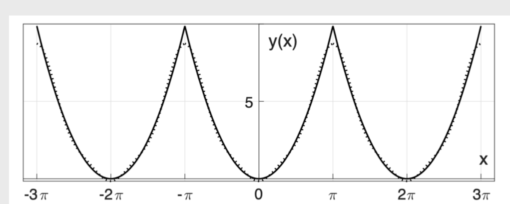
1. Analytical Method: This involves directly applying the Fourier series formulas through integration
2. Numerical Integration: when analytical integration is too complex, numerical methods can estimate the coefficients.
   One needs to be very careful with oscillating functions.
3. Computational Methods (Using Software) : Tools such as MATLAB, Mathematica, Python (with NumPy and SciPy), C++, FORTRAN (with general and special libraries) can calculate Fourier series coefficients efficiently, especially for complex or numerical problems.

23

23

Example

Extension of $f(x) = x^2$ as a periodic function with a period of 2π (solid line), the Fourier series with three terms (dotted line).



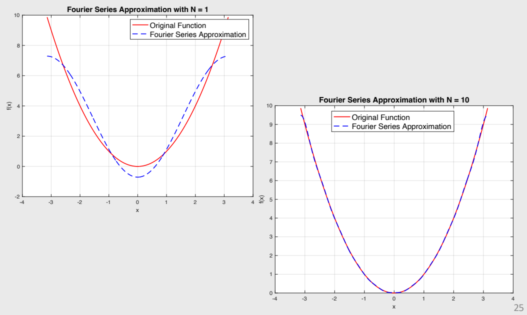
$$x^2 = \frac{\pi^2}{3} + 4 \sum_{k=1}^{\infty} (-1)^k \frac{\cos kx}{k^2} \qquad (-\pi \le x \le \pi).$$

24

24

Example (cont.)

Extension of $f(x) = x^2$ as a periodic function with a period of 2π into the Fourier series with various number of terms

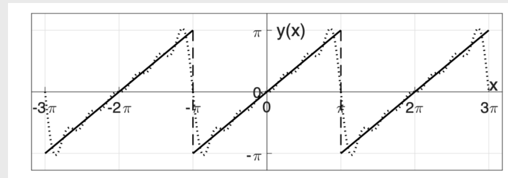


25

25

Example

Extension of $f(x) = x$ as a periodic function with a period of 2π (solid line), the Fourier series with six terms (dotted line).



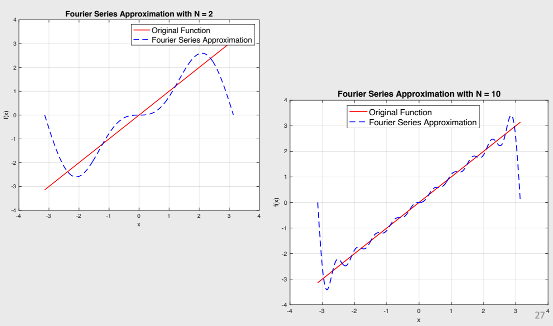
$$2\left(\frac{\sin x}{1}-\frac{\sin 2x}{2}+\dots+\frac{(-1)^{k-1}\sin kx}{k}+\dots\right)=\begin{cases} x & \text{for} \quad -\pi<x<\pi \\ 0 & \text{for} \quad x=\pm\pi \end{cases}$$

26

26

Example (cont.)

Extension of $f(x) = x$ as a periodic function with a period of 2π into the Fourier series with various number of terms

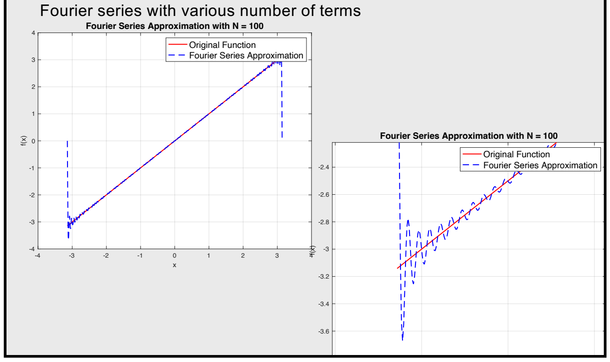


27

27

Example (cont.)

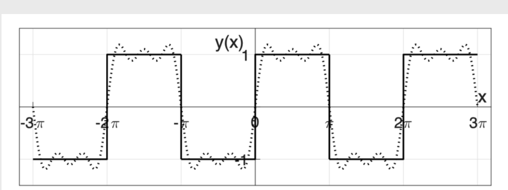
Extension of $f(x) = x$ as a periodic function with a period of 2π into the Fourier series with various number of terms



28

Example

Extension of step function as a periodic function with a period of 2π (solid line), the Fourier series with six terms (dotted line).



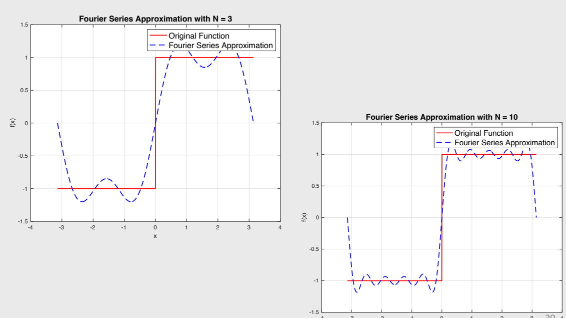
$$f(x)=\frac{1}{2}+\frac{2}{\pi}\left(\frac{\sin x}{1}+\frac{\sin 3x}{3}+\frac{\sin 5x}{5}+\dots\right)=\begin{cases} c_1 & -\pi<x<0 \\ c_2 & x<x<\pi \\ (c_1+c_2)/2 & x=0,\pm\pi \end{cases}\ .$$

29

29

Example (cont.)

Extension of the step function as a periodic function with a period of 2π into the Fourier series with various number of terms

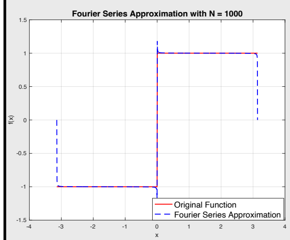


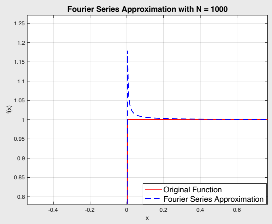
30

30

Example (cont.)

Extension of the step function as a periodic function with a period of 2π into the Fourier series with various number of terms





31

31

Part 3:

The Discrete Fourier Transform

32

Fourier transform

For a given function $f(x)$ the Fourier transform is

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{i\omega x}\, dx$$

The Fourier transform converts the signal $f(x)$ to its transform $F(\omega)$.

And then the inverse Fourier transform

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega) e^{-i\omega x}\, d\omega$$

33

33

The Discrete Fourier Transform (DFT)

- If $f(t)$ or $F(\omega)$ is known analytically, the integrals can be evaluated either analytically or numerically by the numerical integration techniques.
- In practice, the signal $f(t)$ is measured at just a finite number N of times t, and these are all we have as input to approximate the transform.
- The resultant discrete Fourier transform is an approximation both because the signal is not known for all times, and *because* we must integrate numerically.
- Once we have a discrete set of (approximate) transform values, they can be used to reconstruct the signal for any value of the time.
- In this way, the DFT can be thought of as a technique for *interpolating, compressing, and extrapolating* the signal.

34

34

DFT: Discrete times

We assume that the signal $f(t)$ is sampled at $(N + 1)$ discrete times (N time intervals), with a constant spacing $\Delta t = h$ between times:

$$f_k \equiv f(t_k), \qquad k = 0, 1, 2, \dots N$$

$$t_k \equiv kh, \qquad h = \Delta t$$

In other words, we measure the signal $f(t)$ once every h-th of a second for a total time of T. This correspondingly define the signal's period T and the sampling rate s:

$$T \equiv Nh, \qquad s = \frac{N}{T} = \frac{1}{h}$$

Regardless of the true periodicity of the signal, when we choose a period T over which to sample the signal, the mathematics will inevitably produce a $f(t)$ that is periodic with period T, $y(t + T) = y(t)$

Attention: Since the DFT assumes periodic repetition, truncating signals can lead to leakage (spread of energy across frequencies).

35

35

DFT: frequencies and the period

If we are analyzing a truly periodic function, then the N points should span one complete period, but not more. This guarantees their independence.

Unless we make further assumptions, the N independent data $f(t_k)$ can determine no more than N independent transform values $F(\omega_k)\ k = 0, \dots, N$.

The time interval T (which should be the period for periodic functions) is the largest time over which we measure the variation of $f(t)$. Consequently, it determines the lowest frequency contained in our Fourier representation of $f(t)$,

$$\omega_1 = \frac{2\pi}{T}$$

The full range of frequencies in the spectrum ω_n are determined by the number of samples taken, and by the total sampling time $T = Nh$ as

$$\omega_n = n\omega_1 = n\frac{2\pi}{Nh}, \qquad n = 0, 1, \dots, N$$

36

36

DFT: Two basic ideas

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt,$$

The discrete Fourier transform (DFT) algorithm follows from two approximations.

First, we evaluate the integral from time 0 to time T, over which the signal is measured, and not from $-\infty$ to $+\infty$.

Second, the trapezoid rule is used for the integration

$$F(\omega_n) \approx \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{n} h f(t_k) e^{-i\omega_n t_k} = \frac{1}{\sqrt{2\pi}} h \sum_{k=1}^{n} f(t_k) e^{-\frac{i2\pi k n}{N}}$$

then

$$f(t) \approx \frac{1}{\sqrt{2\pi}} \sum_{n=1}^{N} \frac{2\pi}{Nh} F(\omega_n) e^{i\omega_n t}$$

Trapezoid approximation:

$$\int_a^b f(x)dx \approx \frac{1}{2} \Delta x (f_0 + 2f_1 + 2f_2 + \cdots + 2f_{n-1} + f_n)$$

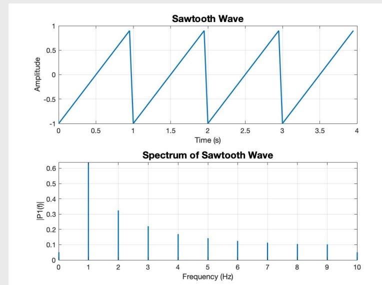
37

37

Example

f = 1; % Fundamental frequency of the sawtooth wave (Hz)

Fs = 20; % Sampling frequency (Hz)



38

38

DFT: smoother frequency spectrum

We see from

$\omega_n = n\omega_1 = n\frac{2\pi}{Nh}$

that the larger we make the time $T = Nh$ over which we sample the function, the smaller will be the frequency steps or resolution.

Accordingly, if you want a smooth frequency spectrum, you need to have a small frequency step $2\pi / T$, which means a longer observation time T.

While the best approach would be to measure the input signal for all times, in practice a measured signal $f(t)$ is often extended in time ("padded") by adding zeros for times beyond the last measured signal, which thereby increases the value of T artificially.

Although this does not add new information to the analysis, it does build in the experimentalist's view that the signal has no existence, or no meaning, at times after the measurements are stopped.

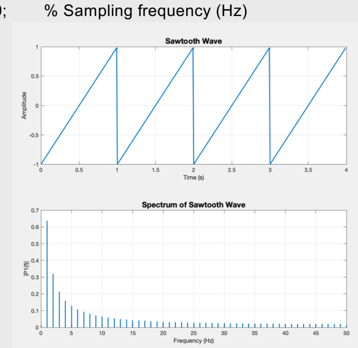
39

39

Example

f = 1; % Fundamental frequency of the sawtooth wave (Hz)

Fs = 100; % Sampling frequency (Hz)



40

40

DFT: problem with periodicity

Periodicity is expected for a Fourier series.

However, if we input values of the signal for longer lengths of time, then the inherent period becomes longer, and if the repeat period T is very long, it may be of little consequence for times short compared to the period.

If $f(t)$ is actually periodic with period Nh, then the DFT is an excellent way of obtaining Fourier series.

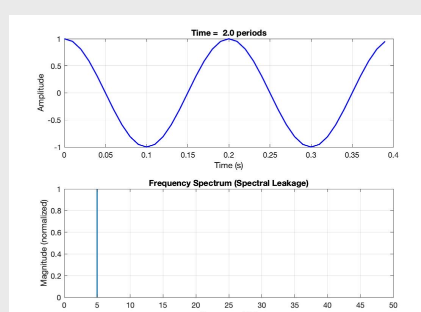
If the input function is not periodic, then the DFT can be a bad approximation near the endpoints of the time interval (after which the function will repeat) or, correspondingly, for the lowest frequencies.

41

41

Spectral leakage I

Suppose we have a signal $y(t) = \cos(2\pi f_0 t)$ with frequency $f_0 = 5$ Hz (or the period of oscillations is $T = 0.2$. Here $Time = 2T$. NO leakage

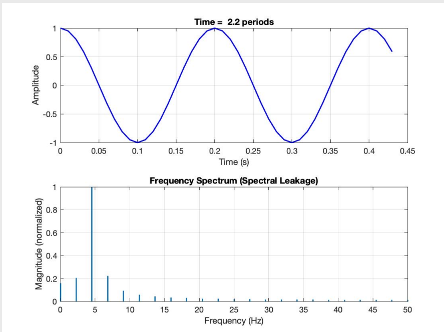


42

42

Spectral leakage II

the period of oscillations is $T = 0.2$. Here $Time = 2.2T$. Leakage

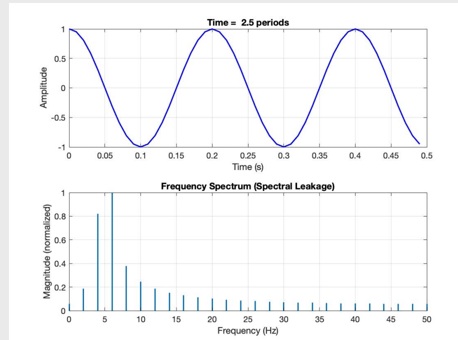


43

43

Spectral leakage III

the period of oscillations is $T = 0.2$. Here $Time = 2.5T$. Leakage even more

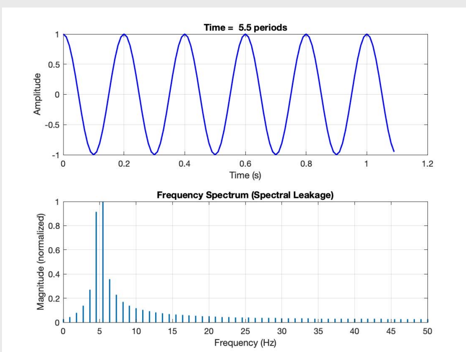


44

44

Spectral leakage IV

the period of oscillations is $T = 0.2$. Here $Time = 5.5T$. Still leakage

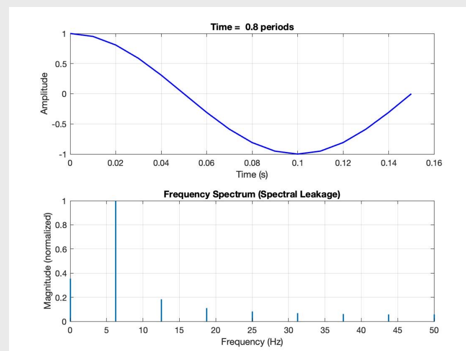


45

45

Spectral leakage V

the period of oscillations is $T = 0.2$. Here $Time = 0.8T$. Leakage again



46

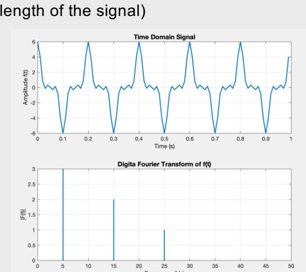
46

DFT: example

Analyze the signal

$$f(t) = 3\cos(\omega t) + 2\cos(3\omega t) + \cos(5\omega t)$$

Let's set $f_0 = 5, F_s = 100$ (sampling frequency), $T = 1/F_s$ (sampling period), $L = 100$ (length of the signal)



47

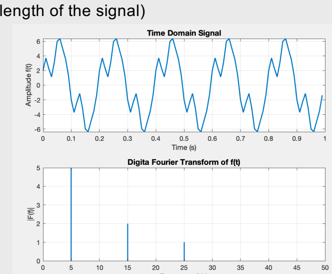
47

DFT: another example

Analyze the signal

$$f(t) = 5\sin(\omega t) + 2\cos(3\omega t) + \sin(5\omega t)$$

Let's set $f_0 = 5, F_s = 100$ (sampling frequency), $T = 1/F_s$ (sampling period), $L = 100$ (length of the signal)



48

48

DFT: Compact form

The DFT and its inverse can be written in a concise and insightful way, and be evaluated efficiently, by introducing a complex variable Z for the exponential and then raising Z to various powers:

$$F_n = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{N} Z^{nk} f_k , \qquad Z = e^{-2\pi i/N}, \qquad Z^{nk} \equiv [(Z)^n]^k$$

$$f_k = \frac{\sqrt{2\pi}}{N} \sum_{n=1}^{N} Z^{-nk} F_n ,$$

With this formulation, the computer needs to compute only powers of Z.

49

49

DFT: Without working with complex numbers

If your preference is to avoid complex numbers, we can rewrite DFT in terms of separate real and imaginary parts by applying Euler's theorem with $\theta = 2\pi \, / \, N$

$$Z = e^{-i\theta}, \qquad Z^{\pm nk} = e^{\mp inkϑ} = \cos(nkϑ) \mp i \sin(nkϑ)$$

$$F_k = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{N} [\cos(nkϑ)\, Re\, f_k + \sin(nkϑ)\, Im\, f_k + i(\cos(nkϑ)\, Im\, f_k - \sin(nkϑ)\; Re\, f_k)]$$

Readers new to DFTs are often surprised when they apply these equations to practical situations and end up with transforms F_k having imaginary parts, despite the fact that the signal f is real.

A real signal ($Im\, f_k \;\equiv\; 0$) will yield an imaginary transform unless $\sum_{k=1}^{N} \sin(nkϑ)\;\; Re\, f_k = 0$. This occurs only if $f(t)$ is an even function and we integrate exactly. Because neither condition holds, the DFTs of real, even functions may have small imaginary parts. This is a good measure of the approximation error in the entire procedure.

50

50

DFT: Aliasing

The sampling of a signal by DFT for only a finite number of times (large Δt) limits the accuracy of the deduced high-frequency components present in the signal.

Clearly, good information about very high frequencies requires sampling the signal with small time steps so that all the wiggles can be included.

While a poor deduction of the high-frequency components may be tolerable if all we care about are the low-frequency components, the inaccurate high-frequency components remain present in the signal and may contaminate the low-frequency components that we deduce.

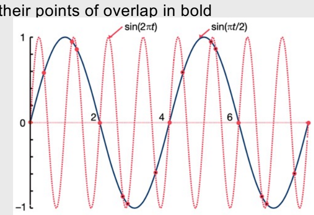
This effect is called aliasing and is the cause of the *Moiré pattern* distortion in digital images.

51

51

DFT: Aliasing: example

As an example, consider the two functions $\sin(\pi t \, / \, 2)$ and $\sin(2\pi t)$ for $0 \;\leq\; t \;\leq\; 8$, with their points of overlap in bold



If we were to sample a signal containing these functions at the times t = 0, 2, 4, 6, 8, then we would measure $f \;\equiv\; 0$ and assume that there was no signal at all.

52

52

DFT: Aliasing: example

As an example, consider the two functions $\sin(\pi t \, / \, 2)$ and $\sin(2\pi t)$ for $0 \;\leq\; t \;\leq\; 8$, with their points of overlap in bold

As an example, consider the two functions $\sin(\pi t \, / \, 2)$ and $\sin(2\pi t)$ for $0 \;\leq\; t \;\leq\; 8$, with their points of overlap in bold

However, if we were to measure the signal at the filled dots where sin (πt/2) = sin(2πt), then our Fourier analysis would completely miss the high-frequency component. In

DFT jargon, we would say that the high-frequency component has been *aliased* by the low-frequency component

However, if we were to measure the signal at the filled dots where sin (πt/2) = sin(2πt), then our Fourier analysis would completely miss the high-frequency component. In

DFT jargon, we would say that the high-frequency component has been *aliased* by the low-frequency component

53

53

DFT: Aliasing (more)

In other cases, some high-frequency values may be included in our sampling of the signal, but our sampling rate may not be high enough to include enough of them to separate the high-frequency component properly.

In this case some high-frequency signals would be included spuriously as part of the low-frequency spectrum, and this would lead to spurious low-frequency oscillations when the signal is synthesized from its Fourier components.

More precisely, aliasing occurs when a signal containing frequency f that is sampled at a rate of $s \;=\; N \, / \, T$ measurements per unit time, with $s \leq f \, / \, 2$. In this case, the frequencies f and $f \;-\; 2s$ yield the same DFT, and we would not be able to determine that there are two frequencies present

54

54

DFT: Nyquist criterion

To avoid aliasing we want NO frequencies $f > s / 2$ to be present in our input signal. This is known as the *Nyquist criterion*.

Although filtering eliminates some high-frequency information, it lessens the distortion of the low-frequency components, and so may lead to improved reproduction of the signal.

If accurate values for the high frequencies are required, then we will need to increase the sampling rate s by increasing the number N of samples taken within the fixed sampling time $T = N\,h$

If we increase the total time sampling time $T = N\,h$ and keep h the same, then the sampling rate $s = N / T = 1 / h$ remains the same. Because $\omega_1 = 2\pi / T$, this makes ω_1 smaller, which means we have more low frequencies recorded and a smoother frequency spectrum.

55

Nyquist Sampling Theorem

The Nyquist theorem states that a continuous signal must be sampled at least twice its highest frequency component to avoid aliasing. If the signal's highest frequency is f_{max}, the sampling frequency f_s must satisfy:

$$f_s \geq 2 f_{max}$$

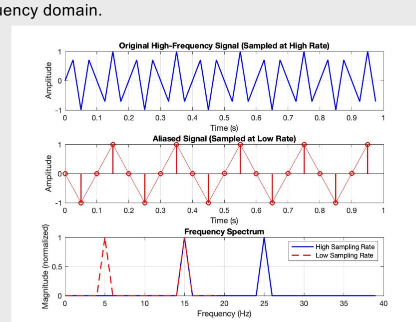
If this condition is not met, aliasing will occur, leading to distortion in the frequency spectrum.

56

Nyquist Sampling Theorem (example)

In this example, we create a high-frequency sine wave, sample it at a frequency below the Nyquist rate, and use the DFT to observe the effect of aliasing in the frequency domain.

Due to sampling below the Nyquist rate, the 15 Hz component "folds" down to 5 Hz in the spectrum.



57

Part 4:

Fast Fourier Transform

58

DFT challenge

The DFT

$$F_n = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{N} Z^{nk} f_k\,, \qquad Z = e^{-2\pi i/N}, \qquad \mathrm{n} = 0, 1, \ldots \mathrm{N} - 1$$

Even if the signal elements f_k to be transformed are real, Z is complex, and therefore we must process both real and imaginary parts when computing transforms.

Because both n and k range over N integer values, the $Z^{nk} f_k$ multiplications require some N^2 multiplications and additions of complex numbers.

As N gets large, as happens in realistic applications, this geometric increase in the number of steps leads to long computation times.

59

FFT: history

- First step: Carl Friedrich Gauss's unpublished 1805 work on the orbits of asteroids Pallas and Juno.
- Next step: Joseph Fourier in 1822 (without analyzing complexity)
- pre-computer era: Danielson and Lanczos , 1942
- Need for speed: Cooley and Tukey, 1965 (they got the most credit)

FFT reduces the number of operations necessary to perform a DFT from N^2 to roughly $N \log 2\, N$

N	N^2	$N \log 2\, N$	
10	100	33	
100	10,000	664	
1000	1000,000	9,965	hundred times faster

for 10,000 – almost 1000 times faster

Because of its widespread use (including cell phones), the fast Fourier transform algorithm is considered one of the 10 most important algorithms of all time.

60

Top 10 most important algorithms

1946: The Metropolis Algorithm

1947: Simplex Method

1950: Krylov Subspace Method

1951: The Decompositional Approach to Matrix Computations

1957: The Fortran Optimizing Compiler

1959: QR Algorithm

1962: Quicksort

1965: Fast Fourier Transform

1977: Integer Relation Detection

1987: Fast Multipole Method

61

61

The idea behind the FFT

The idea behind the FFT is to utilize the periodicity inherent in the definition of the DFT

$$F_n = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{N} Z^{nk} f_k , \qquad Z = e^{-2\pi i/N}, \qquad \mathrm{n} = 0, 1, \ldots \mathrm{N} - 1$$

to reduce the total number of computational steps.

Essentially, the algorithm divides the input data into two equal groups and transforms only one group, which requires $\sim\ (N/2)^2$ multiplications.

It then divides the remaining (nontransformed) group of data in half and transforms them, continuing the process until all the data have been transformed.

The total number of multiplications required with this approach is approximately $N\ \log 2\ N$.

62

62

More specific

Let's change notations (for the sake of saving time)

$$Y_n = \frac{1}{\sqrt{2\pi}} \sum_{k=1}^{N} Z^{nk} y_k , \qquad Z = e^{-2\pi i/N}, \qquad \mathrm{n} = 0, 1, \ldots \mathrm{N} - 1$$

Specifically, the FFT's time economy arises from the computationally expensive complex factor $Z^{nk} \equiv [(Z)^n]^k$ having values that are repeated as the integers n and k vary sequentially.

63

63

More specific

For instance, for $N = 8$,

$$\begin{aligned}
Y_0 &= Z^0 y_0 + Z^0 y_1 + Z^0 y_2 + Z^0 y_3 + Z^0 y_4 + Z^0 y_5 + Z^0 y_6 + Z^0 y_7 ,\\
Y_1 &= Z^0 y_0 + Z^1 y_1 + Z^2 y_2 + Z^3 y_3 + Z^4 y_4 + Z^5 y_5 + Z^6 y_6 + Z^7 y_7 ,\\
Y_2 &= Z^0 y_0 + Z^2 y_1 + Z^4 y_2 + Z^6 y_3 + Z^8 y_4 + Z^{10} y_5 + Z^{12} y_6 + Z^{14} y_7 ,\\
Y_3 &= Z^0 y_0 + Z^3 y_1 + Z^6 y_2 + Z^9 y_3 + Z^{12} y_4 + Z^{15} y_5 + Z^{18} y_6 + Z^{21} y_7 ,\\
Y_4 &= Z^0 y_0 + Z^4 y_1 + Z^8 y_2 + Z^{12} y_3 + Z^{16} y_4 + Z^{20} y_5 + Z^{24} y_6 + Z^{28} y_7 ,\\
Y_5 &= Z^0 y_0 + Z^5 y_1 + Z^{10} y_2 + Z^{15} y_3 + Z^{20} y_4 + Z^{25} y_5 + Z^{30} y_6 + Z^{35} y_7 ,\\
Y_6 &= Z^0 y_0 + Z^6 y_1 + Z^{12} y_2 + Z^{18} y_3 + Z^{24} y_4 + Z^{30} y_5 + Z^{36} y_6 + Z^{42} y_7 ,\\
Y_7 &= Z^0 y_0 + Z^7 y_1 + Z^{14} y_2 + Z^{21} y_3 + Z^{28} y_4 + Z^{35} y_5 + Z^{42} y_6 + Z^{49} y_7 ,
\end{aligned}$$

where we include $Z^0\ (\equiv\ 1)$ for clarity.

64

64

More specific (cont.)

When we actually evaluate these powers of Z, we find only four independent values:

$$\begin{aligned}
Z^0 &= \exp(0) = +1 , & Z^1 &= \exp(-\frac{2\pi}{8}) = +\frac{\sqrt{2}}{2} - \mathrm{i}\frac{\sqrt{2}}{2} ,\\
Z^2 &= \exp\left(-\frac{2 \cdot 2\mathrm{i}\pi}{8}\right) = -\mathrm{i} , & Z^3 &= \exp\left(-\frac{2\pi \cdot 3\mathrm{i}}{8}\right) = -\frac{\sqrt{2}}{2} - \mathrm{i}\frac{\sqrt{2}}{2} ,\\
Z^4 &= \exp\left(-\frac{2\pi \cdot 4\mathrm{i}}{8}\right) = -Z^0 , & Z^5 &= \exp\left(-\frac{2\pi \cdot 5\mathrm{i}}{8}\right) = -Z^1 ,\\
Z^6 &= \exp\left(-\frac{2 \cdot 6\mathrm{i}\pi}{8}\right) = -Z^2 , & Z^7 &= \exp\left(-\frac{2 \cdot 7\mathrm{i}\pi}{8}\right) = -Z^3 ,\\
Z^8 &= \exp\left(-\frac{2\pi \cdot 8\mathrm{i}}{8}\right) = +Z^0 , & Z^9 &= \exp\left(-\frac{2\pi \cdot 9\mathrm{i}}{8}\right) = +Z^1 ,\\
Z^{10} &= \exp\left(-\frac{2\pi \cdot 10\mathrm{i}}{8}\right) = +Z^2 , & Z^{11} &= \exp\left(-\frac{2\pi \cdot 11\mathrm{i}}{8}\right) = +Z^3 ,\\
Z^{12} &= \exp\left(-\frac{2\pi \cdot 11\mathrm{i}}{8}\right) = -Z^0 , & &\cdots
\end{aligned} \tag{12.82}$$

65

65

More specific (cont.)

When substituted into the definitions of the transforms, we obtain

$$\begin{aligned}
Y_0 &= Z^0 y_0 + Z^0 y_1 + Z^0 y_2 + Z^0 y_3 + Z^0 y_4 + Z^0 y_5 + Z^0 y_6 + Z^0 y_7 ,\\
Y_1 &= Z^0 y_0 + Z^1 y_1 + Z^2 y_2 + Z^3 y_3 - Z^0 y_4 - Z^1 y_5 - Z^2 y_6 - Z^3 y_7 ,\\
Y_2 &= Z^0 y_0 + Z^2 y_1 - Z^0 y_2 - Z^2 y_3 + Z^0 y_4 + Z^2 y_5 - Z^0 y_6 - Z^2 y_7 ,\\
Y_3 &= Z^0 y_0 + Z^3 y_1 - Z^2 y_2 + Z^1 y_3 - Z^0 y_4 - Z^3 y_5 + Z^2 y_6 - Z^1 y_7 ,\\
Y_4 &= Z^0 y_0 - Z^0 y_1 + Z^0 y_2 - Z^0 y_3 + Z^0 y_4 - Z^0 y_5 + Z^0 y_6 - Z^0 y_7 ,\\
Y_5 &= Z^0 y_0 - Z^1 y_1 + Z^2 y_2 - Z^3 y_3 - Z^0 y_4 + Z^1 y_5 - Z^2 y_6 + Z^3 y_7 ,\\
Y_6 &= Z^0 y_0 - Z^2 y_1 - Z^0 y_2 + Z^2 y_3 + Z^0 y_4 - Z^2 y_5 - Z^0 y_6 + Z^2 y_7 ,\\
Y_7 &= Z^0 y_0 - Z^3 y_1 - Z^2 y_2 - Z^1 y_3 - Z^0 y_4 + Z^3 y_5 + Z^2 y_6 + Z^1 y_7 ,\\
Y_8 &= Y_0 .
\end{aligned}$$

66

66

More specific (cont.)

We see that these transforms now require 8 × 8 = 64 multiplications of com- plex numbers, in addition to some less time-consuming additions. We place these equations in an appropriate form for computing by regrouping the terms into sums and differences of the y's:

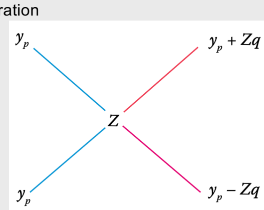
(see next slide)

More specific (cont.)

$$Y_0 = Z^0(y_0 + y_4) + Z^0(y_1 + y_5) + Z^0(y_2 + y_6) + Z^0(y_3 + y_7)\,,$$
$$Y_1 = Z^0(y_0 - y_4) + Z^1(y_1 - y_5) + Z^2(y_2 - y_6) + Z^3(y_3 - y_7)\,,$$
$$Y_2 = Z^0(y_0 + y_4) + Z^2(y_1 + y_5) - Z^0(y_2 + y_6) - Z^2(y_3 + y_7)\,,$$
$$Y_3 = Z^0(y_0 - y_4) + Z^3(y_1 - y_5) - Z^2(y_2 - y_6) + Z^1(y_3 - y_7)\,,$$
$$Y_4 = Z^0(y_0 + y_4) - Z^0(y_1 + y_5) + Z^0(y_2 + y_6) - Z^0(y_3 + y_7)\,,$$
$$Y_5 = Z^0(y_0 - y_4) - Z^1(y_1 - y_5) + Z^2(y_2 - y_6) - Z^3(y_3 - y_7)\,,$$
$$Y_6 = Z^0(y_0 + y_4) - Z^2(y_1 + y_5) - Z^0(y_2 + y_6) + Z^2(y_3 + y_7)\,,$$
$$Y_7 = Z^0(y_0 - y_4) - Z^3(y_1 - y_5) - Z^2(y_2 - y_6) - Z^1(y_3 - y_7)\,,$$
$$Y_8 = Y_0\,.$$

More specific (cont.)

Note the repeating factors inside the parentheses, with combinations of the form $y_p \pm y_q$. These symmetries are systematized by introducing the butterfly operation



This operation takes the y_p and y_q data elements from the left wing and converts them to the $y_p + Zy_q$ elements in the right wings.

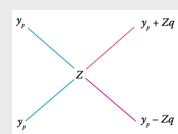
We apply the butterfly operations to an entire FFT process, specifically to the pairs (y_0, y_4), (y_1, y_5), (y_2, y_6), and (y_3, y_7).

More specific (cont.)

Note how the number of multiplications of complex numbers has been reduced:

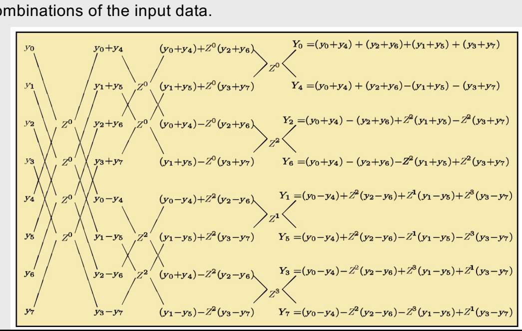
For the first butterfly operation there are 8 multiplications by Z^0; for the second butterfly operation there are 8 multiplications, and so forth, until a total of 24 multiplications are made in four butterflies.

In contrast, 64 multiplications are required in the original DFT



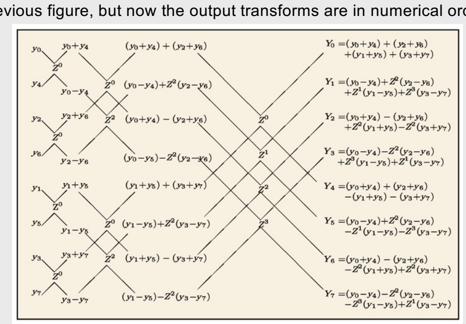
More specific (cont.)

The butterfly operations performing a FFT on the eight data on the left leading to eight transforms on the right. The transforms are different linear combinations of the input data.



More specific (cont.)

A modified FFT in which the eight input data on the left are transformed into eight transforms on the right. The results are the same as in the previous figure, but now the output transforms are in numerical order



FFT - implementations

There are very many programs written in many languages for doing FFT

Check what is available for you!!!

73

Part 4b:

Filtering noisy signals

74

Filtering noisy signals (basics)

Filtering noisy signals is a common problem in signal processing.

Noise can distort a signal, making it challenging to interpret or analyze.

Filtering helps to reduce unwanted noise while preserving the important features of the signal.

Types of Noise:

- Random (white noise),
- Periodic (e.g., hum from power lines), or
- Structured (e.g., interference from other signals).

75

Types of Filters

- Low-Pass Filter:
  Allows low-frequency components to pass while attenuating high-frequency noise.
- High-Pass Filter:
  Allows high-frequency components to pass, useful for removing low-frequency trends.
- Band-Pass Filter:
  Allows a specific range of frequencies to pass, rejecting frequencies outside this range.
- Notch Filter:
  Removes specific frequencies, useful for removing periodic interference.

76

Common Filtering Techniques

A. Fourier Transform Filtering

- Use FFT to transform the signal into the frequency domain.
- Remove unwanted frequency components by setting their values to zero.
- Apply the inverse FFT to get the filtered signal.

B. Digital Filtering

- Use designed digital filters (like Butterworth, Chebyshev, or FIR filters) to remove specific frequencies directly in the time domain.

C. Moving Average Filter

- Simple and effective for removing high-frequency noise by averaging neighboring points.

D. Wavelet Denoising

- Wavelet transforms are useful for decomposing a signal into various frequency bands and selectively removing noise.

77

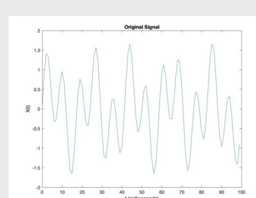
Example – using FFT

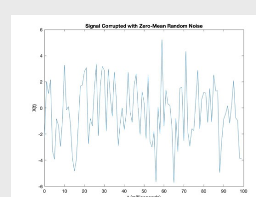
A signal containing a 50 Hz sinusoid of amplitude 0.7 and a 120 Hz sinusoid of amplitude 1.

$$S = 0.7\sin(2\pi * 50 * t) + \sin(2\pi * 120 * t)$$

and Corrupt the signal with zero-mean white noise with a variance of 4.

$$X = S + 2 * randn(size(t))$$

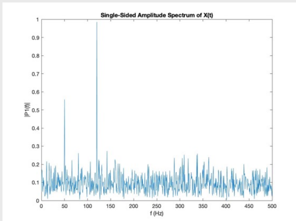




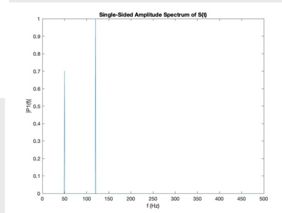
78

After using FFT

After applying FFT



if there was no noise

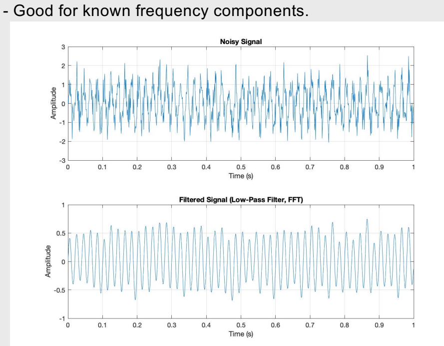


79

Example – Fourier Transform Filtering (Low-Pass)

FFT-based low-pass filter to remove high-frequency noise.

FFT - Good for known frequency components.



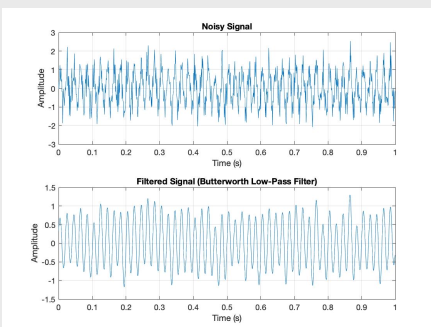
80

80

Example – Butterworth Low-Pass Filter

Using a Butterworth filter to remove high-frequency noise.

Smooth filtering for general use; commonly used.



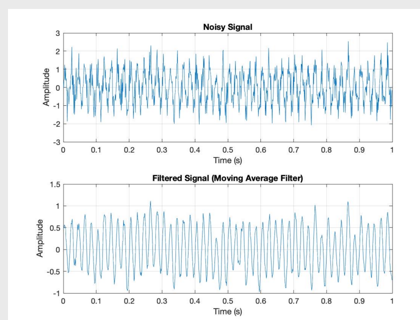
81

81

Example – Moving Average Filter

A moving average filter smooths the signal by averaging points.

Simple, effective for reducing high-frequency noise.



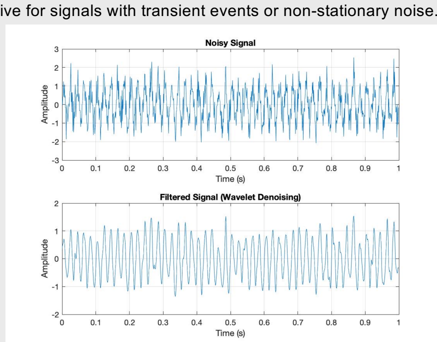
82

82

Example – Wavelet Denoising

Wavelet denoising can remove noise while preserving sharp changes

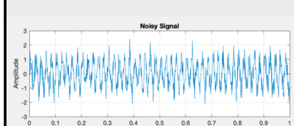
Effective for signals with transient events or non-stationary noise.

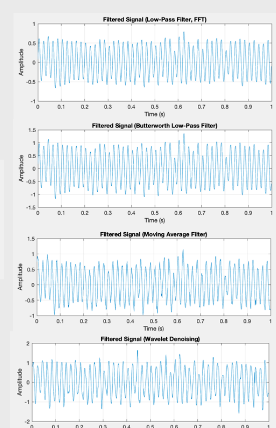


83

83

All four:
same slide





84