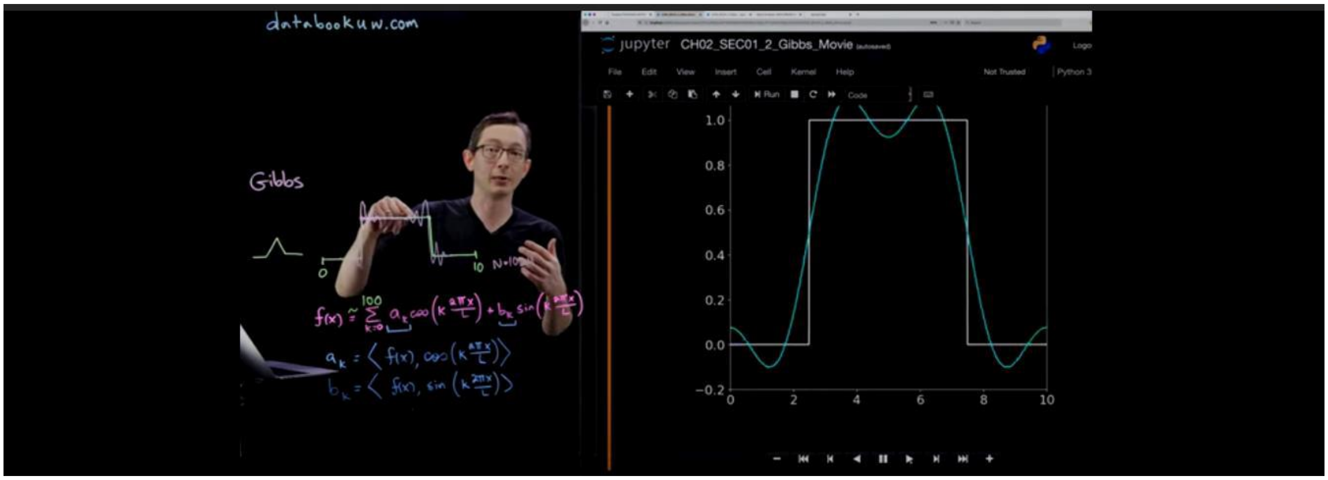
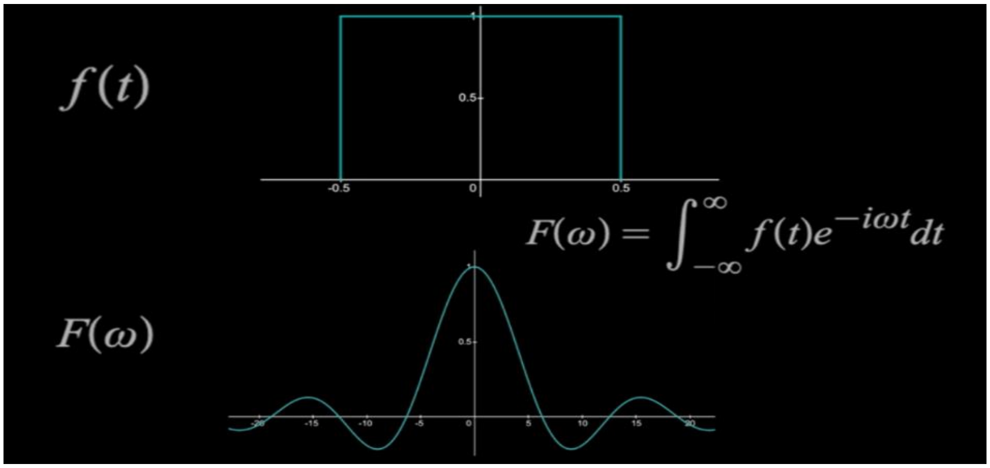
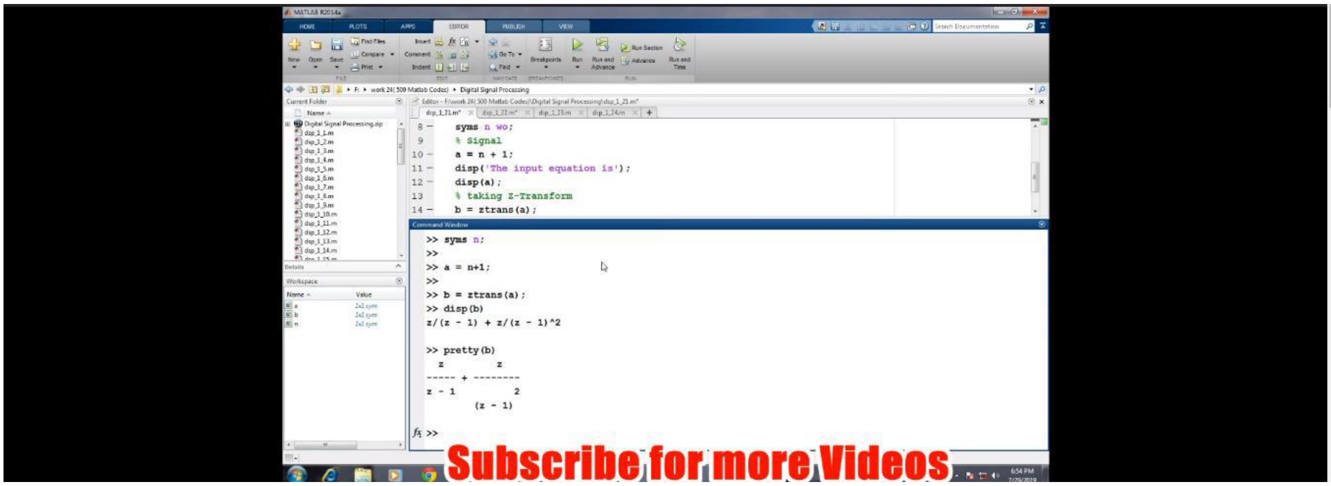
DAILY ASSESSMENT

Date:	26-May-2020	Name:	Swastik R Gowda
Course:	Digital Signal Processing	USN:	4AL17EC091
Topic:	❖ Fourier Series & Gibbs Phenomena using Python ❖ Fourier Transform and derivatives ❖ Convolution ❖ Laplace transform and z- transform	Semester & Section:	6th Sem 'B' Sec
Github Repository:	Swastik-gowda		

FORENOON SESSION DETAILS
Image of session







Report – Report can be typed or hand written for up to two pages.

Given the Fourier transform of a general function, find the Fourier transform of its derivative. Use this result to find the Fourier transform of a window function out of the Fourier transform of an asymmetric pair of delta functions. (t)eiωtdt. (t)] = −iωF(ω).
Each of these sinusoidal terms has a magnitude (scale factor) and a phase (shift). – Note that in a computer, we can represent a function as an array of numbers giving the values of that function at equally spaced points Fourier and Laplace transforms are very much related to each other, and both of them can help you solve differential equations, but the intuitive difference is that: The Fourier Transform is more useful for understanding the steady state response of a system.
Z transform is used to convert discrete time domain signal into discrete frequency domain signal. It has wide range of applications in mathematics and digital signal processing. It is mainly used to analyze and process digital data.

Derivatives of functions

The Fourier transform of the derivative of a function is given by:

$$\begin{aligned}
\mathcal{F}\left(\frac{d}{dx}f(x)\right) &= \int_{-\infty}^{\infty} \overbrace{f'(x)}^{dv}\,\overbrace{e^{-i\omega x}}^{u}\,dx \\
&= \Big[\underbrace{f(x)e^{-i\omega x}}_{uv}\Big]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} \underbrace{f(x)}_{v}\Big[\underbrace{-i\omega e^{-i\omega x}}_{du}\Big]\,dx \\
&= i\omega \int_{-\infty}^{\infty} f(x)e^{-i\omega x}\,dx \\
&= i\omega \mathcal{F}(f(x)).
\end{aligned}$$

This is an extremely important property of the Fourier transform, as it will allow us to turn PDEs into ODEs, closely related to the separation of variables:

$$\underset{\text{(PDE)}}{u_{tt} = cu_{xx}} \quad \overset{\mathcal{F}}{\Longrightarrow} \quad \underset{\text{(ODE)}}{\hat{u}_{tt} = -c\omega^2\hat{u}.}$$

Linearity of Fourier transforms

The Fourier transform is a linear operator, so that:

$$\mathcal{F}(\alpha f(x) + \beta g(x)) = \alpha \mathcal{F}(f) + \beta \mathcal{F}(g).$$

$$\mathcal{F}^{-1}(\alpha \hat{f}(\omega) + \beta \hat{g}(\omega)) = \alpha \mathcal{F}^{-1}(\hat{f}) + \beta \mathcal{F}^{-1}(\hat{g}).$$

Parseval's theorem

$$\int_{-\infty}^{\infty} |\hat{f}(\omega)|^2 \, d\omega = 2\pi \int_{-\infty}^{\infty} |f(x)|^2 \, dx.$$

Convolution

The convolution of two functions is particularly well-behaved in the Fourier domain, being the product of the two Fourier transformed functions. Define the convolution of two functions f(x) and g(x) as f * g:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(x - \xi) g(\xi) \, d\xi.$$

If we let ^f = F(f) and ^g = F(g), then:

$$\begin{aligned}
\mathcal{F}^{-1}\left(\hat{f}\hat{g}\right)(x) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega)\hat{g}(\omega) e^{i\omega x} \, d\omega \\
&= \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega x} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} g(y) e^{-i\omega y} \, dy \right) d\omega \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(y) \hat{f}(\omega) e^{i\omega(x-y)} \, d\omega \, dy \\
&= \int_{-\infty}^{\infty} g(y) \underbrace{\left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega(x-y)} \, d\omega \right)}_{f(x-y)} dy \\
&= \int_{-\infty}^{\infty} g(y) f(x-y) \, dy = g * f = f * g.
\end{aligned}$$

Laplace Transform

Laplace transform of a signal (function) f is the function F = L(f) defined by:

$$F(s) = \int_{0}^{\infty} f(t) e^{-st} \, dt$$

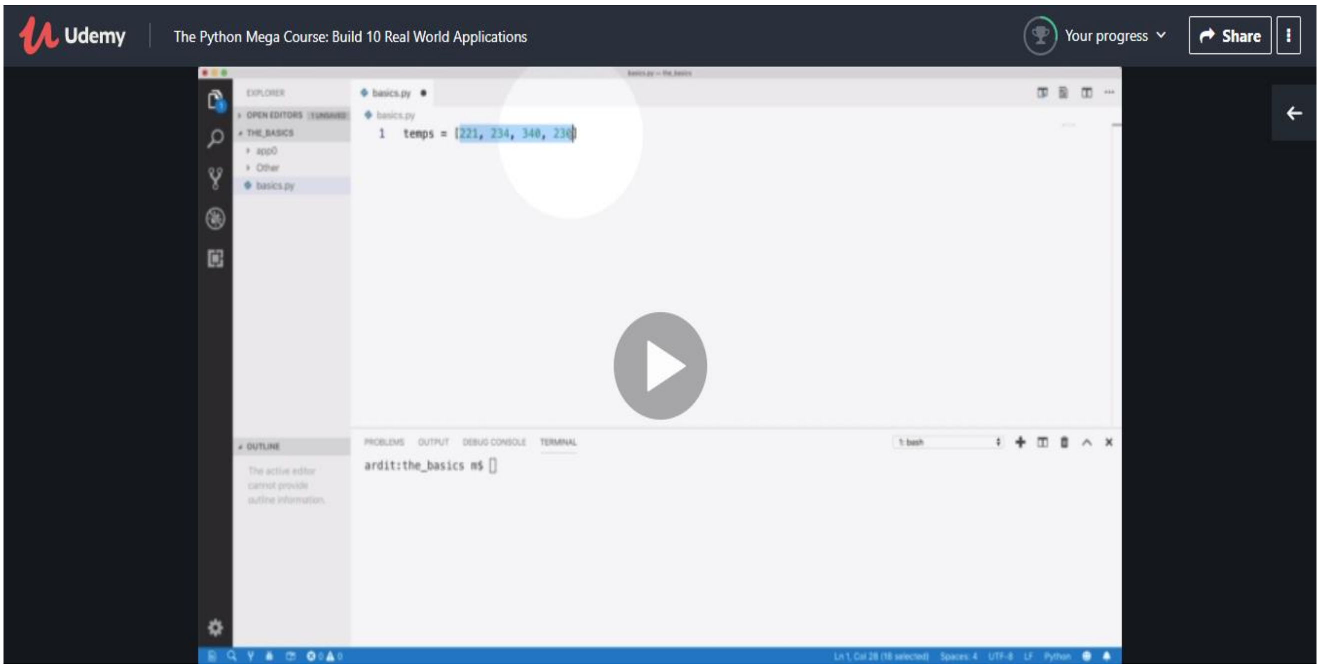
For those $s \in C$ for which the integral makes sense.

- F is a complex-valued function of complex numbers.
- s is called the (complex) frequency variable, with units sec^{-1}.
- t is called the time variable (in sec).
- Assume f contains no impulses at t = 0.

Date:	**26-May-2020**	**Name:**	**Swastik R Gowda**
Course:	**Python**	**USN:**	**4AL17EC091**
Topic:	• **List comprehensions** • **More on functions** • **File processing** • **Imported values**	**Semester & Section:**	**6th Sem 'B' Sec**

AFTERNOON SESSION DETAILS

Image of session



Report – Report can be typed or hand written for up to two pages.

List Comprehensions

We have learnt basic list comprehension. It's an expression that creates a list by iterating over another container.

EXAMPLE: [i*2 for i in [1,5,10]]

This multiplies every value in the list with 2 and output is given as [2,10,20]

- This was using for loop.
- We can do this using if also along with for

[i * 2 for i in [1 , -2 , 10] if i > 0]

- This multiplies every value in list which is greater than zero with 2. Thus gives output as [2 , 20]
- Using if and else condition:

[i * 2 if i > 0 else o for i in [1 , -2 , 10]]

This gives output as [2, 0, 20]

Functions:

- Functions can have more than one parameter.
  Def foo(a,b)
- Functions can also have default parameter
  Def foo (a,b=9)
- Arguments can be passed as non-Keyword arguments or keyword arguments.
- An *args parameter allows the function to be called with an arbitrary number of non-keyword arguments.
- Keywords allow the function to be called with an arbitrary number of keyword arguments.

File Processing

- Any file can be read using python
- We can create a new file with python & write text in it.
    - For reading we use “r”
    - For writing we use “w”
- We can append text also in a file
  For appending we use “a”
    - With open (“file.txt”,”r”) as file
    - With open (“file.txt”,”w”) as file
    - With open (“file.txt”,”a”) as file
- We can both append & read a file by
  With open (“file.txt”,”at”)

Imported Modules

- Builtin objects are all objects that are written inside the python interpreter in ‘C’ language
- Builtin modules contains builtins object
- A list of all builtin modules can be created & printed import sys
  Sys.builtin_module_names
- Standard libraries is a jargon that includes both builtin modules written in C and also modules written in python.
- Third party libraries can be installed from terminals.