**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **26 MAY 2020** | **Name:** | **MANAVI** |
| **Course:** | **Signals and system** | **USN:** | **4AL18EC031** |
| **Topic:** | **Day 2: fourier series & gibbs phenomena using python.**  **fourier transform**  **fourier transform derivatives**  **fourier transform & convolutions**  **intuition of fourier transform and laplace transform**  **laplace transform of 1st order**  **implementation of 1st order**  **implementation of laplacetransform using matlab**  **application of z-transform**  **find the z-transform of sequence using matlab** | **Semester & Section:** | **4TH SEM**  **& A SEC** |
| **Github Repository:** | **Manavi-test** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **FOURIER SERIES AND GIBBS PHENOMENA:-**   * Bessel functions come up naturally when working in polar coordinates, just as sines and cosines come up naturally when working in rectangular coordinates. You can think of Bessel functions as a sort of variation on sine waves. Or even more accurately, a variation on sinc functions, where sinc(z) = sin(z)/z. [1] * A Fourier series represents a function as a sum of sines and cosines of different frequencies. To make things a little simpler here, I’ll only consider Fourier sine series so I don’t have to repeatedly say “and cosine.” * f(z) = \sum\_{n=1}^\infty c\_n \sin(n \pi z) * A Fourier-Bessel function does something similar. It represents a function as a sum of rescaled versions of a particular Bessel function. We’ll use the Bessel J0 here, but you could pick some other Jν. * Fourier series scale the sine and cosine functions by π times integers, i.e. sin(πz), sin(2πz), sin(3πz), etc. Fourier-Bessel series scale by the zeros of the Bessel function: J0(λ1z), J0(λ2z), J0(λ3z), etc. where λn are the zeros of J0. This is analogous to scaling sin(πz) by its roots: π, 2π, 3π, etc. So a Fourier-Bessel series for a function f looks like * f(z) = \sum\_{n=1}^\infty c\_n J\_0(\lambda\_n z). * The coefficients cn for Fourier-Bessel series can be computed analogously to Fourier coefficients, but with a couple minor complications. First, the basis functions of a Fourier series are orthogonal over [0, 1] without any explicit weight, i.e. with weight 1. And second, the inner product of a basis function doesn’t depend on the frequency. In detail, * Here δmn equals 1 if m = n and 0 otherwise. * Fourier-Bessel basis functions are orthogonal with a weight z, and the inner product of a basis function with itself depends on the frequency. In detail * So whereas the coefficients for a Fourier sine series are given by * the coefficients for a Fourier-Bessel series are given by * Fourier and Fourier-Bessel series are examples of orthogonal series, and so by construction they converge in the norm given by their associated inner product. That means that if SN is the Nth partial sum of a Fourier series      |  |  |  |  | | --- | --- | --- | --- | | **DATE:** | **26 MAY 2020** | **NAME:** | **MANAVI** | | **COURSE:** | **PYTHON** | **USN:** | **4AL18EC031** | | **TOPIC:** | **DAY 8:graphical user interfaces with tkinter**  **interacting with data bases.** | **SEMESTER & SECTION:** | **4TH SEM & A SEC** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages**  **Graphical User Interfaces With Tkinter Interacting With Data Bases:-**   * GUI is nothing but a desktop app that provides you with an interface that helps you to interact with the computers and enriches your experience of giving a command (command-line input) to your code. They are used to perform different tasks in desktops, laptops, and other electronic devices, etc. * Some of the applications where the power of GUI is utilized are: * Creating a Calculator which would have a user-interface and functionalities that persists in a calculator. * Text-Editors, IDE's for coding are on a GUI app. * Sudoku, Chess, Solitaire, etc.., are games that you can play are GUI apps   .   * Chrome, Firefox, Microsoft Edge, etc. used to surf the internet is a GUI app. * Another interesting use-case would be - A GUI for controlling a Drone from your laptop, and the GUI would probably have buttons to maneuver the Drone along with a screen that would show the camera feed captured by the Drone in a real-time. * Let's see some of the frameworks that Python provides to develop a GUI: * PyQT is one of the favored cross-platform Python bindings implementing the Qt library for the Qt application development framework. Nokia primarily owns Qt. Currently, PyQT is available for almost all operating systems like Unix/Linux, Windows, Mac OS X. It blends the best of Python and Qt and provides flexibility to the programmer to decide whether to create a program by writing a pure python code or use Qt Designer to create visual dialogs. * Kivy is for the creation of new user interfaces and is an OpenGL ES 2 accelerated framework. Much like PyQt, Kivy also supports almost all platforms like Windows, MacOSX, Linux, Android, iOS. It is an open-source framework and comes with over 20 pre-loaded widgets in its toolkit. * Jython is a Python port for Java, which gives Python scripts seamless access to Java class libraries on the local machine. * WxPython, initially known as WxWindows (now as a WxWidgets library), is an open-source abstract-level wrapper for cross-platform GUI library. It is implemented as a Python expansion module. With WxPython, you, as a developer, can create native applications for Windows, Mac OS, and Unix. | | | |