**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 27 May 2020 | **Name:** | Anupama J S |
| **Course:** | DSP | **USN:** | 4AL16EC005 |
| **Topic:** | 1. Fourier Series & Gibbs Phenomena using Python 2. Fourier Transform 3. Fourier Transform Derivatives 4. Fourier Transform and Convolution | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | AnupamaJS |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**    C:\Users\User\Downloads\WhatsApp Image 2020-05-27 at 6.42.42 PM.jpeg\  **C:\Users\User\Downloads\WhatsApp Image 2020-05-27 at 6.42.42 PM (1).jpeg** |
| **Report – Report can be typed or hand written for up to two pages.**  **FOURIER SERIES & GIBBS PHENOMENA USING PYTHON**  In [mathematics](https://en.wikipedia.org/wiki/Mathematics), the Gibbs phenomenon, discovered by [Henry Wilbraham](https://en.wikipedia.org/wiki/Henry_Wilbraham) ([1848](https://en.wikipedia.org/wiki/Gibbs_phenomenon#CITEREFWilbraham1848)) and rediscovered by [J. Willard Gibbs](https://en.wikipedia.org/wiki/Willard_Gibbs) ([1899](https://en.wikipedia.org/wiki/Gibbs_phenomenon#CITEREFGibbs1899)) is the peculiar manner in which the [Fourier series](https://en.wikipedia.org/wiki/Fourier_series) of a [piecewise](https://en.wikipedia.org/wiki/Piecewise) continuously differentiable [periodic function](https://en.wikipedia.org/wiki/Periodic_function) behaves at a [jump discontinuity](https://en.wikipedia.org/wiki/Jump_discontinuity). The *n*th [partial sum](https://en.wikipedia.org/wiki/Partial_sum) of the Fourier series has large oscillations near the jump, which might increase the maximum of the partial sum above that of the function itself. The overshoot does not die out as *n* increases, but approaches a finite limit. This sort of behavior was also observed by experimental physicists, but was believed to be due to imperfections in the measuring apparatus. The Gibbs phenomenon involves both the fact that Fourier sums overshoot at a jump discontinuity, and that this overshoot does not die out as more terms are added to the sum. The three pictures on the right demonstrate the phenomenon for a square wave (of height {\displaystyle \pi /4}\pi /4) whose Fourier expansion is As can be seen, as the number of terms rises, the error of the approximation is reduced in width and energy, but converges to a fixed height. A calculation for the square wave (see Zygmund, chap. 8.5., or the computations at the end of this article) gives an explicit formula for the limit of the height of the error. It turns out that the Fourier series exceeds the height {\displaystyle \pi /4}\pi /4 of the square wave by about 9 percent of the jump. More generally, at any jump point of a piecewise continuously differentiable function with a jump of a, the nth partial Fourier series will (for n very large) overshoot this jump by approximately at one end and undershoot it by the same amount at the other end; thus the "jump" in the partial Fourier series will be about 18% larger than the jump in the original function. At the location of the discontinuity itself, the partial Fourier series will converge to the midpoint of the jump (regardless of what the actual value of the original function is at this point).  **FOURIER TRANSFORM DERIVATIVES AND CONVOLUTION**  C:\Users\User\Downloads\WhatsApp Image 2020-05-27 at 6.42.42 PM (2).jpeg  C:\Users\User\Downloads\WhatsApp Image 2020-05-27 at 6.42.43 PM.jpeg  We’re given an array of numerical values – We can think of this array as specifying values of a function at regularly spaced intervals • To compute a moving average, we replace each value in the array with the average of several values that precede and follow it (i.e., the values within a window) • We might choose instead to calculate a weighted moving average, where we again replace each value in the array with the average of several surrounding values, but we weight those values differently • We can express this as a convolution of the original function (i.e., array) with another function (array) that specifies the weights on each value in the window If f and g are functions defined at evenly spaced points, their convolution is given by: ( f ∗ g)[n] = f [m] m=−∞ ∞ ∑ g[n − m]  **Two-dimensional convolution** • In two-dimensional convolution, we replace each value in a two-dimensional array with a weighted average of the values surrounding it in two dimensions – We can represent two-dimensional arrays as functions of two variables, or as matrices, or as images  **Multidimensional convolution •** The concept generalizes to higher dimensions • For example, in three-dimensional convolution, we replace each value in a three-dimensional array with a weighted average of the values surrounding it in three dimensions |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 27 May 2020 | **Name:** | Anupama J S |
| **Course:** | Python | **USN:** | 4AL16EC005 |
| **Topic:** | 1. Graphical user interfaces with tkinter | **Semester & Section:** | 8th sem “A”section |
| **Github Repository:** | AnupamaJS |  |  |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session**  **C:\Users\User\Downloads\WhatsApp Image 2020-05-27 at 6.17.28 PM.jpeg** | | | |
| **Report – Report can be typed or hand written for up to two pages.**  **INTRODUCTION TO TKINTER**  In this part of the Tkinter tutorial, we introduce the Tkinter toolkit and create our first programs. The purpose of this tutorial is to get you started with the Tkinter toolkit.  **Tkinter**  Tkinter is a Python binding to the Tk GUI toolkit. Tk is the original GUI library for the Tcl language. Tkinter is implemented as a Python wrapper around a complete Tcl interpreter embedded in the Python interpreter. There are several other popular Python GUI toolkits. Most popular are wxPython, PyQt, and PyGTK.  **Python**  python logoPython is a general-purpose, dynamic, object-oriented programming language. The design purpose of the Python language emphasizes programmer productivity and code readability. Python was initially developed by Guido van Rossum. It was first released in 1991. Python was inspired by ABC, Haskell, Java, Lisp, Icon, and Perl programming languages. Python is a high-level, general-purpose, multiplatform, interpreted language. Python is well suited for learning about GUI programming.  The official web site for the Python programming language is python.org  **Pillow**  Pillow is a Python library for for opening, manipulating, and saving many different image file formats. Some of the examples in this tutorial use Pillow.  $ sudo apt-get install python-pil.imagetk  On Debian Linux, we can install Pillow with its package manager.  $ yum install python-imaging  On systems using RPM package format, we use the above command to install Pillow.  $ pip install pillow  Alternatively, we can use pip to install Pillow.  Tkinter simple example  In our first example, we show a basic window on the screen.  simple.py  #!/usr/bin/env python3  ZetCode Tkinter tutorial  from tkinter import Tk, BOTH  from tkinter.ttk import Frame  class Example(Frame):  def \_init\_(self):  super().\_init\_()  self.initUI()  def initUI(self):  self.master.title("Simple")  self.pack(fill=BOTH, expand=1)  def main():  root = Tk()  root.geometry("250x150+300+300")  app = Example()  root.mainloop()  if \_name\_ == '\_main\_':  main()  While this code is very small, the application window can do quite a lot. It can be resized, maximized, or minimized. All the complexity that comes with it has been hidden from the application programmer.  from tkinter import Tk, BOTH  from tkinter.ttk import Frame  Here we import Tk and Frame classes, and BOTH constant. Tk class is used to create a root window. Frame is a container for other widgets.  class Example(Frame):  def \_init\_(self):  super().\_init\_()  Our example class inherits from the Frame container widget. In the \_init\_ constructor method we call the constructor of our inherited class.  self.initUI()  **C:\Users\User\Downloads\WhatsApp Image 2020-05-27 at 6.17.28 PM (1).jpeg**  **INTRODUCTION TO GUI WITH TKINTER IN PYTHON**  In this tutorial, you are going to learn how to create GUI apps in Python. You'll also learn about all the elements needed to develop GUI apps in Python.  Before you begin, you should be familiar with Python to learn Tkinter. If you're new to Python, check out DataCamp's Introduction to Python course.  Introduction  missing  Drone Feed on GUI using Tkinter  Most of you write a code and run it in a command-line terminal or an IDE (Integrated Development Environment), and the code produces an output based on what you expect out of it either on the terminal or on the IDE itself. However, what if you want your system to have a fancy looking user-interface or maybe your application (use-case) requires you to have a GUI.  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 | | | |