**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | 25th May 2020 | **Name:** | Rashmi K B |
| **Course:** | Digital signal processing | **USN:** | 4AL16EC056 |
| **Topic:** | Introduction to Fourier Series & Fourier Transform, Fourier Series using Matlab and python | **Semester & Section:** | 8th B |
| **Github Repository:** | rashmikb |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**    **d2.PNG** |
| **Report**  **Introduction to Fourier Series & Fourier Transform:**  The fast Fourier transform has become the cornerstone of computational mathematics, enabling real-time image and audio compression, global communication networks, modern devices and hardware, numerical physics and engineering at advanced data analysis.  The fast Fourier transform has had a more significant and profound role in shaping the modern world than any other algorithm to date. With increasingly complex problems, data sets, and computational geometries, simple Fourier sine and cosine bases have given way to tailored bases, such as the data-driven SVD. In fact, the SVD basis can be used as a direct analogue of the Fourier basis for solving PDEs with complex geometries. In addition, related functions, called wavelets, have been developed for advanced signal processing and compression efforts.  **Fourier series:**  A fundamental result in Fourier analysis is that if f(x) is periodic and piece wise smooth, then it can be written in terms of a Fourier series, which is an infinite sum of cosines and sines of increasing frequency. In particular, if f(x) is 2\_periodic, it may be written as:    Fourier series approximation to a hat function  % Define domain  dx = 0.001;  L = pi;  x = (-1+dx:dx:1)\*L;  n = length(x); nquart = floor(n/4);  % Define hat function  f = 0\*x;  f(nquart:2\*nquart) = 4\*(1:nquart+1)/n;  f(2\*nquart+1:3\*nquart) = 1-4\*(0:nquart-1)/n;  plot(x,f,’-k’,’LineWidth’,1.5), hold on  % Compute Fourier series  CC = jet(20);  A0 = sum(f.\*ones(size(x)))\*dx;  fFS = A0/2;  for k=1:20  A(k) = sum(f.\*cos(pi\*k\*x/L))\*dx; % Inner product  B(k) = sum(f.\*sin(pi\*k\*x/L))\*dx;  fFS = fFS + A(k)\*cos(k\*pi\*x/L) + B(k)\*sin(k\*pi\*x/L);  plot(x,fFS,’-’,’Color’,CC(k,:),’LineWidth’,1.2)  end  **Fourier series using python:**  test that it works with real coefficients:  from numpy import linspace, allclose, cos, sin, ones\_like, exp, pi, \  complex64, zeros    def series\_real\_coeff(a0, a, b, t, T):  """calculates the Fourier series with period T at times t,  from the real coeff. a0,a,b"""  tmp = ones\_like(t) \* a0 / 2.  for k, (ak, bk) in enumerate(zip(a, b)):  tmp += ak \* cos(2 \* pi \* (k + 1) \* t / T) + bk \* sin(  2 \* pi \* (k + 1) \* t / T)  return tmp  t = linspace(0, T, 100)  f\_values = f(t)  a0, a, b = fourier\_series\_coeff\_numpy(f, T, 52)  # construct the series:  f\_series\_values = series\_real\_coeff(a0, a, b, t, T)  # check that the series and the original function match to numerical precision:  assert allclose(f\_series\_values, f\_values, atol=1e-6)  # #### test similarly that it works with complex coefficients:  def series\_complex\_coeff(c, t, T):  """calculates the Fourier series with period T at times t,  from the complex coeff. c"""  tmp = zeros((t.size), dtype=complex64)  for k, ck in enumerate(c):  # sum from 0 to +N  tmp += ck \* exp(2j \* pi \* k \* t / T)  # sum from -N to -1  if k != 0:  tmp += ck.conjugate() \* exp(-2j \* pi \* k \* t / T)  return tmp.real  f\_values = f(t)  c = fourier\_series\_coeff\_numpy(f, T, 7, return\_complex=True)  f\_series\_values = series\_complex\_coeff(c, t, T)  assert allclose(f\_series\_values, f\_values, atol=1e-6) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | 25th May 2020 | **Name:** | Rashmi K B | |
| **Course:** | Udemy | **USN:** | 4AL16EC056 | |
| **Topic:** | Python | **Semester & Section:** | 8th B | |
| **AFTERNOON SESSION DETAILS** | | | |
| **To fix programming errors:**  **Working of host file in website blocker:**  # Run this script as root  import time  from datetime import datetime as dt  # change hosts path according to your OS  hosts\_path = "/etc/hosts"  # localhost's IP  redirect = "127.0.0.1"  # websites That you want to block  website\_list =  ["www.facebook.com","facebook.com",  "dub119.mail.live.com","www.dub119.mail.live.com",  "www.gmail.com","gmail.com"]  while True:  # time of your work  if dt(dt.now().year, dt.now().month, dt.now().day,8)  < dt.now() < dt(dt.now().year, dt.now().month, dt.now().day,16):  print("Working hours...")  with open(hosts\_path, 'r+') as file:  content = file.read()  for website in website\_list:  if website in content:  pass  else:  # mapping hostnames to your localhost IP address  file.write(redirect + " " + website + "\n")  else:  with open(hosts\_path, 'r+') as file:  content=file.readlines()  file.seek(0)  for line in content:  if not any(website in line for website in website\_list):  file.write(line)  # removing hostnmes from host file  file.truncate()  print("Fun hours...")  time.sleep(5) | | | |