**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **25 May 2020** | **Name:** | **Veronica gudagur** |
| **Course:** | **Digital Signal Processing** | **USN:** | **4al16ec091** |
| **Topic:** | |  | | --- | | Introduction to Fourier Series & Fourier Transform | | Fourier Series – Part 1 | | Fourier Series – Part 2 | | Inner Product in Hilbert Transform | | Complex Fourier Series | | Fourier Series using Mat lab  (Use Octave to execute the code) | | Fourier Series using Python  (Experience implementation using Python) | | Fourier Series and Gibbs Phenomena Using Mat lab | | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Veronica-g** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **A close up of a map  Description automatically generatedA close up of a logo  Description automatically generatedA close up of a map  Description automatically generatedA screenshot of a cell phone  Description automatically generatedA picture containing drawing, sitting  Description automatically generatedA close up of a map  Description automatically generated**  **A picture containing text, blackboard, man, swinging  Description automatically generatedA screen shot of a person  Description automatically generated**  **A screen shot of a person  Description automatically generatedA screen shot of a person  Description automatically generated** |
| **Report -**  Fourier’s seminal work provided the mathematical foundation for Hilbert spaces, operator theory, approximation theory, and the subsequent revolution in analytical and computational mathematics. Fast forward two hundred years, and 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 scale, and advanced data analysis. Simply put, the fast Fourier transform has had a more signiﬁcant and profound role in shaping the modern world than any other algorithm to date.  **Fourier series and Fourier transforms**  The discrete and continuous formulations should match in the limit of data with inﬁnitely ﬁne resolution. The Fourier series and transform are intimately related to the geometry of inﬁnite-dimensional function spaces, or Hilbert spaces, which generalize the notion of vector spaces to include functions with inﬁnitely many degrees of freedom. Thus, we begin with an introduction to function spaces.  **Inner products of functions and vectors**  In this section, we will make use of inner products and norms of functions. In particular, we will use the common Hermitian inner product for functions f(x) and g(x) deﬁned for x on a domain x ∈ [a, b]: <f(x),g(x )> = --------------🡪 (2.1)  where ¯ g denotes the complex conjugate. The inner product of functions may seem strange or unmotivated at ﬁrst, but this deﬁnition becomes clear when we consider the inner product of vectors of data. If we discretize the functions f(x) and g(x) into vectors of data, as in we would like the vector inner product to converge to the function inner product as the sampling resolution is increased. The inner product of the data vectors f = [f1 f2 ··· fn]T and g =[g1 g2 ··· gn]T is  deﬁned by:  <f,g> = g∗f = ∑nk=1 fkgk= ∑ nk=1f(xk) g(xk).-------------------🡪 (2.2)  The magnitude of this inner product will grow as more data points are added; i.e., as n increases. Thus, we may normalize by ∆x = (b−a)/(n−1):  b−a /n−1 <f,g> = ∑n  k=1 f(xk)¯ g(xk)∆x, ---------------------🡪(2.3) whichistheRiemannapproximationtothecontinuousfunctioninnerproduct. It is now clear that as we take the limit of n → ∞ (i.e., inﬁnite data resolution, with ∆x → 0), the vector inner product converges to the inner product of functions in (2.1).  This inner product also induces a norm on functions, given by  ||f||2 = (<f,f>)1/2 = =( )1/2 .-------------------🡪 (2.4)  The set of all functions with bounded norm deﬁne the set of square integrable functions, denoted by L2([a,b]); this is also known as the set of Lebesgue integrable functions. The interval [a,b] may also be chosen to be inﬁnite (e.g., (−∞,∞)), semi-inﬁnite (e.g., [a,∞)), or periodic (e.g., [−π,π)). A fun example of a function in L2([1,∞)) is f(x) = 1/x. The square of f has ﬁnite integral from 1 to ∞, although the integral of the function itself diverges. The shape obtained by rotating this function about the x-axis is known as Gabriel’s horn, as the volume is ﬁnite (related to the integral of f2), while the surface area is inﬁnite (related to the integral of f). As in ﬁnite-dimensional vector spaces, the inner product may be used to project a function into an new coordinate system deﬁned by a basis of orthogonal functions. A Fourier series representation of a function f is precisely a projection of this function onto the orthogonal set of sine and cosine functions with integer period on the domain [a,b]. This is the subject of the following sections.  **Fourier series**  f(x) =a0/ 2+∑∞ k=1(ak cos(kx) + bk sin(kx))  Thus, the functions ψk = eikx for k ∈ Z (i.e., for integer k) provide a basis for periodic, complex-valued functions on an interval [0,2π). It is simple to see that these functions are orthogonal:    Thecoefﬁcientsaregivenbyck = 1 2πhf(x),ψk(x)i. Thefactorof1/2π normalizestheprojectionbythesquareofthenormof ψk;i.e.,kψkk2 = 2π. Thisisconsistent withourstandardﬁnite-dimensionalnotionofchangeofbasis,asinFig.2.2. A vector \* f maybewritteninthe (\* x, \* y) or (\* u,\* v) coordinatesystems,viaprojection onto these orthogonal bases:  A picture containing object, clock  Description automatically generated   * 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   * The truncated Fourier series is plagued by ringing oscillations, known as Gibbs phenomena, around the sharp corners of the step function. This example highlights the challenge of applying the Fourier series to discontinuous functions:   dx = 0.01; L = 10;  x = 0:dx:L;  n = length(x); nquart = floor(n/4);  f = zeros(size(x)); f(nquart:3\*nquart) = 1;  A0 = sum(f.\*ones(size(x)))\*dx\*2/L;  fFS = A0/2;  for k=1:100  Ak = sum(f.\*cos(2\*pi\*k\*x/L))\*dx\*2/L;  Bk = sum(f.\*sin(2\*pi\*k\*x/L))\*dx\*2/L;  fFS = fFS + Ak\*cos(2\*k\*pi\*x/L) + Bk\*sin(2\*k\*pi\*x/L);  end  plot(x,f,’k’,’LineWidth’,2), hold on  plot(x,fFS,’r-’,’LineWidth’,1.2)  **A screenshot of a cell phone  Description automatically generated** |

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