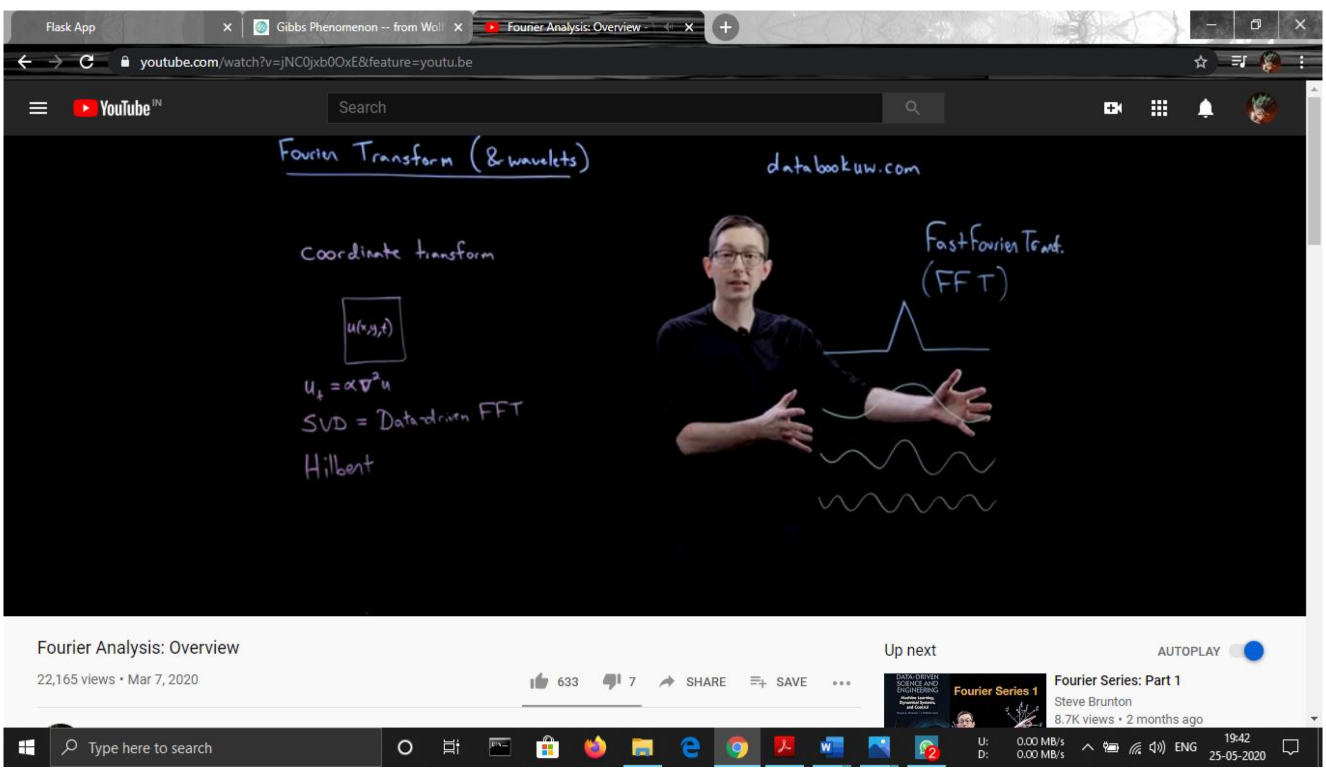
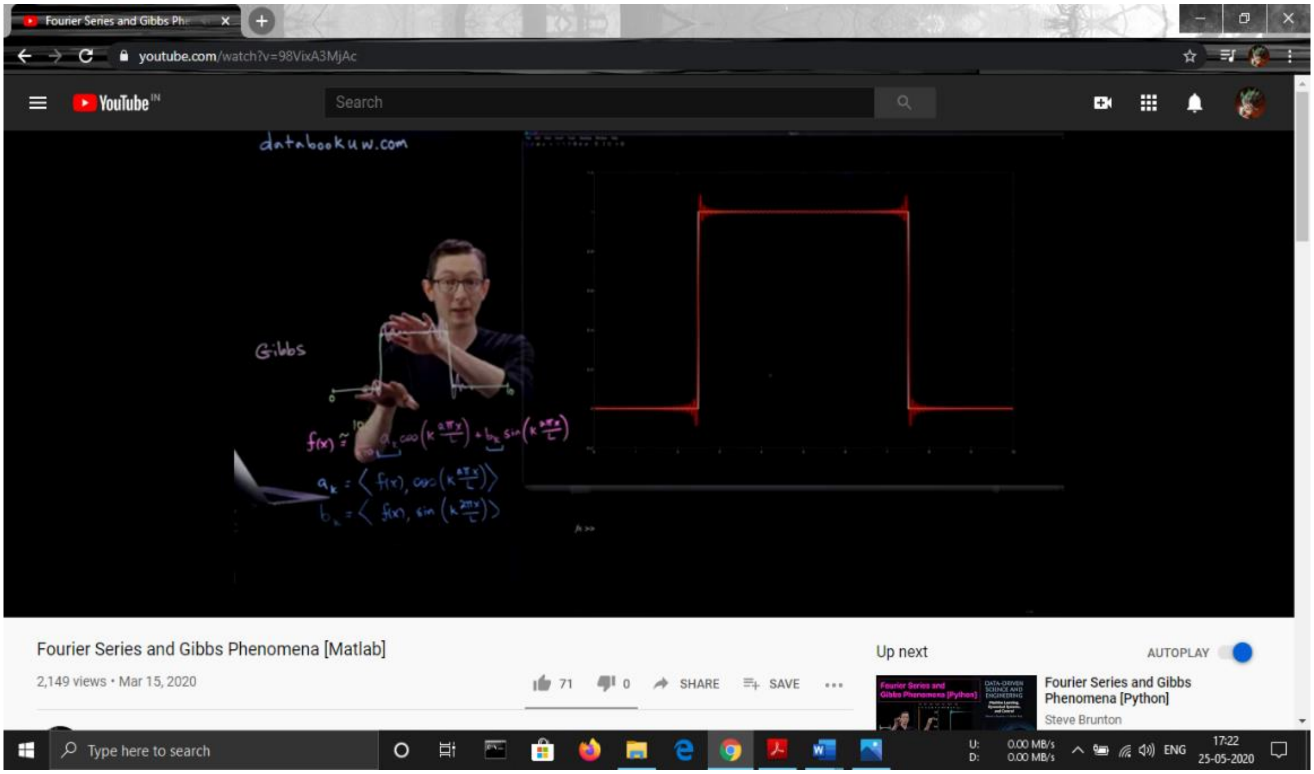
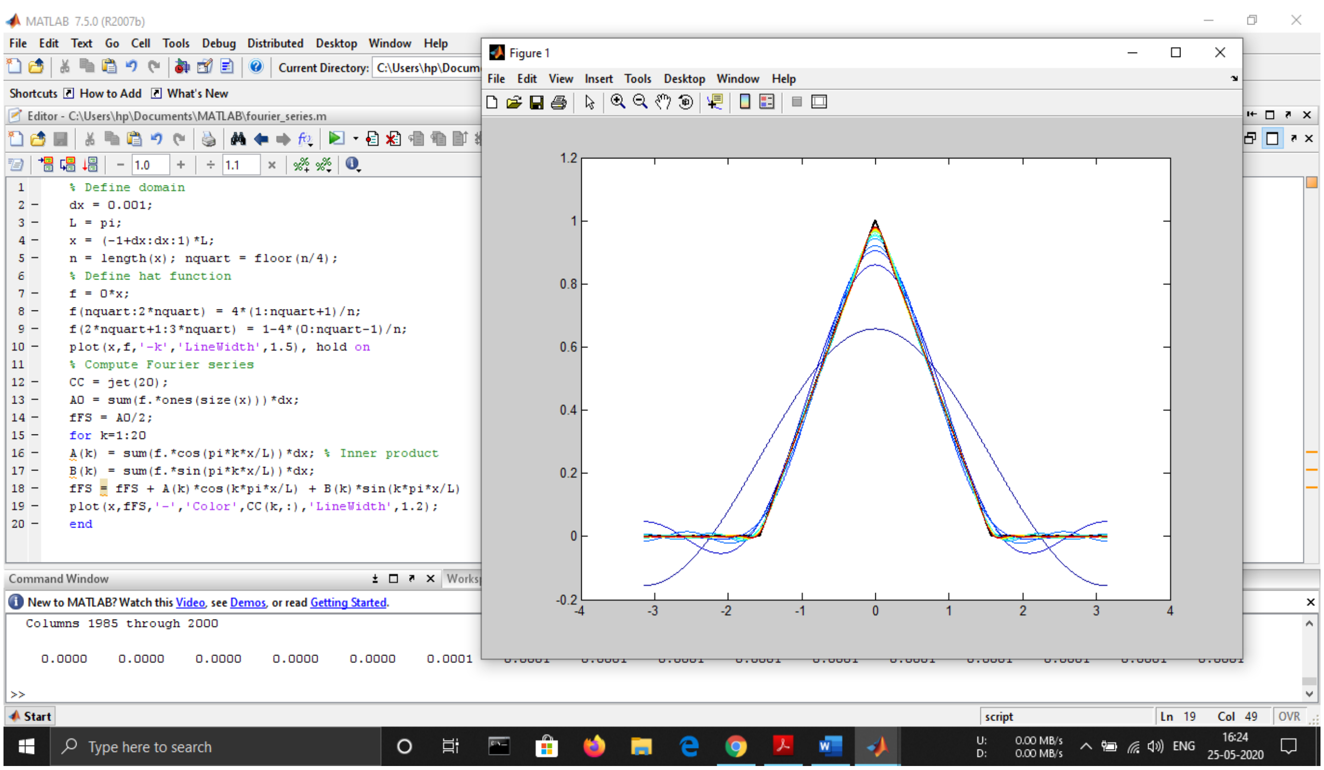
DAILY ASSESSMENT REPORT

Date:	**25/05/2020**	**Name:**	**Abhishek M Shastry K**
Subject:	**Digital Signal Processing**	**USN:**	**4AL17EC002**
Topic:	**1] Introduction to Fourier Series & Fourier Transform** **2] Fourier Series – Part 1** **3] Fourier Series – Part 2** **4] Inner Product in Hilbert Transform** **5] Complex Fourier Series** **6] Fourier Series using Matlab** **7] Fourier Series using Python** **8] Fourier Series and Gibbs Phenomena Using Matlab**	**Semester & Section:**	**6^{th} 'A'**
Github Repository:	**AbhishekShastry-Courses**		

FORENOON SESSION DETAILS
Image of session




Fourier Series and Gibbs Phe
youtube.com/watch?v=98VixA3MjAc
YouTube IN
Search
databookuw.com
Gibbs
Fourier Series and Gibbs Phenomena [Matlab]
2,149 views • Mar 15, 2020
71
0
SHARE
SAVE
Up next
AUTOPLAY
Fourier Series and Gibbs Phenomena [Python]
Steve Brunton
Type here to search
ENG
17:22
25-05-2020


MATLAB 7.5.0 (R2007b)
File Edit Text Go Cell Tools Debug Distributed Desktop Window Help
Current Directory: C:\Users\hp\Docum
Shortcuts How to Add What's New
Editor - C:\Users\hp\Documents\MATLAB\fourier_series.m
% 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
Figure 1
File Edit View Insert Tools Desktop Window Help
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Columns 1985 through 2000
0.0000 0.0000 0.0000 0.0000 0.0000 0.0001
>>
Start
script
Ln 19 Col 49 OVR
Type here to search
ENG
16:24
25-05-2020

Report

Fourier Series

A fundamental result in Fourier analysis is that if f(x) is periodic and piecewise 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 2π periodic, it may be written as:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos(kx) + b_k \sin(kx)\right).$$

The coefficients a_k and b_k are given by,

$$a_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(kx)dx$$
$$b_k = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(kx)dx,$$

Fourier Transform

The Fourier transform integral is essentially the limit of a Fourier series as the length of the domain goes to infinity, which allows us to define a function defined on (-∞,∞) without repeating. We will consider the Fourier series on a domain x belongs to [-L; L), and then let L → ∞. On this domain, the Fourier series is:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}\left[a_k \cos\left(\frac{k\pi x}{L}\right) + b_k \sin\left(\frac{k\pi x}{L}\right)\right] = \sum_{k=-\infty}^{\infty} c_k e^{ik\pi x/L}$$

With the coefficients given by:

$$c_k = \frac{1}{2L}\langle f(x), \psi_k\rangle = \frac{1}{2L}\int_{-L}^{L} f(x)e^{-ik\pi x/L}\,dx.$$

Using previous results of f(x) in addition, the summation with weight ▲w becomes a Riemann integral, resulting in the following:

$$f(x) = \mathcal{F}^{-1}\left(\hat{f}(\omega)\right) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)e^{i\omega x}\,d\omega$$
$$\hat{f}(\omega) = \mathcal{F}\left(f(x)\right) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x}\,dx.$$

These two integrals are known as the Fourier transform pair.

Complex Fourier Series

The complex Fourier series is presented first with period 2π, then with general period. The expression for complex fourier series is given by,

$$f(t) = d + \sum_{n=1}^{\infty} [a_n \cos(nt) + b_n \sin(nt)]$$

$$= d + \sum_{n=1}^{\infty} \left[a_n \left(\frac{e^{int} + e^{-int}}{2} \right) + b_n \left(\frac{e^{int} - e^{-int}}{2i} \right) \right]$$

$$= d + \sum_{n=1}^{\infty} \frac{(a_n - ib_n)}{2} e^{int} + \sum_{n=1}^{\infty} \frac{(a_n + ib_n)}{2} e^{-int}$$

$$= \sum_{n=-\infty}^{\infty} c_n e^{int}$$

Where,

$$c_n = \begin{cases} d & , n = 0 \\ (a_n - ib_n)/2 & , n = 1, 2, 3, \ldots \\ (a_{-n} + ib_{-n})/2 & , n = -1, -2, -3, \ldots \end{cases}$$

Note that a_{-n} and b_{-n} are only defined when n is negative.

Hilbert Transform

In mathematics and in signal processing, the Hilbert transform is a specific linear operator that takes a function, $u(t)$ of a real variable and produces another function of a real variable $H(u)(t)$. This linear operator is given by convolution with the function $1/(\pi t)$:

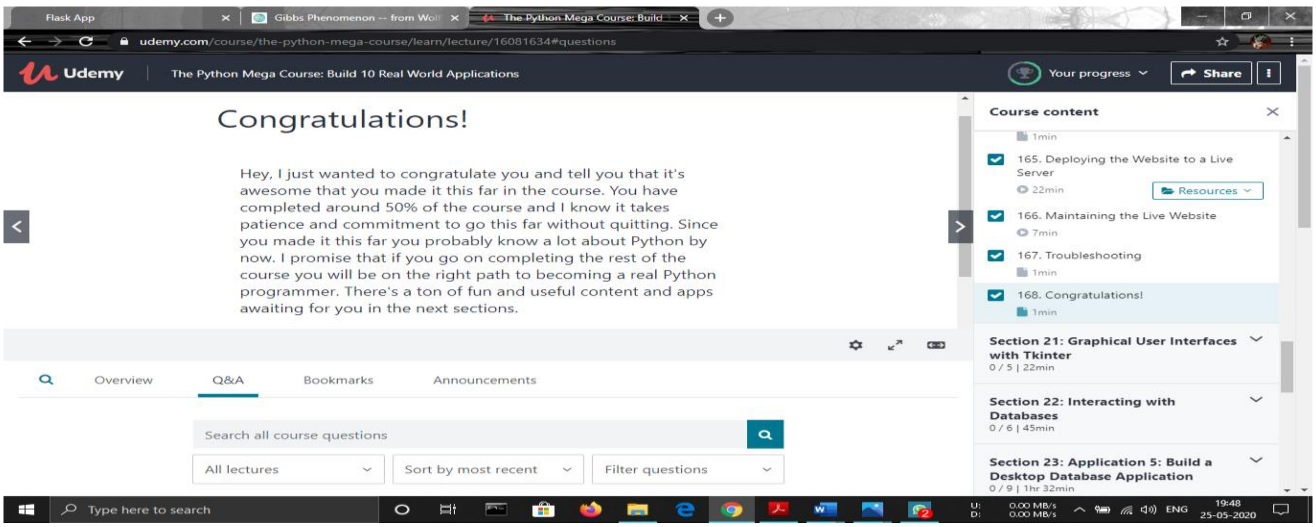
$$H(u)(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{u(\tau)}{t - \tau} \, d\tau$$

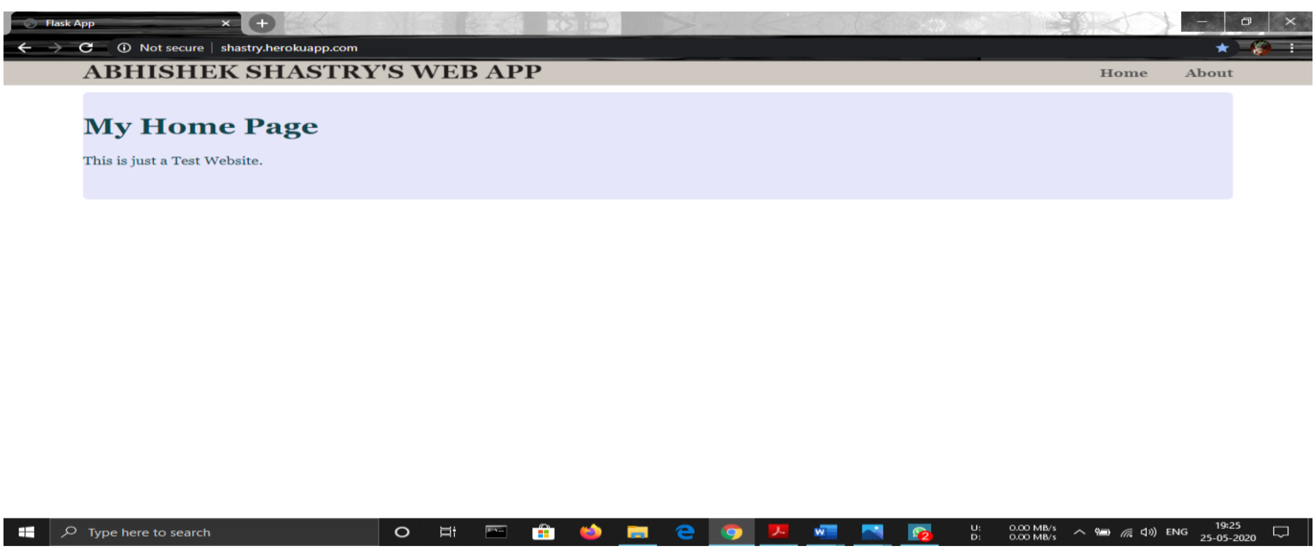
- The Gibbs phenomenon is an overshoot (or "ringing") of Fourier series and other eigen function series occurring at simple discontinuities.
- It can be reduced with the Lanczos sigma factor. The phenomenon is illustrated above in the Fourier series of a square wave.

Date:	**25/05/2020**	**Name:**	**Abhishek M Shastry K**
Course:	**The Python Mega Course: Build 10 Real World Applications**	**USN:**	**4AL17EC002**
Topic:	**1] Application 4: Build a Personal Website with Python and Flask**	**Semester & Section:**	**6th 'A'**
Github Repository:	**AbhishekShastry-Courses**		

AFTERNOON SESSION DETAILS

Image of session





Report

Application 4: Build a Personal Website with Python and Flask

- A local Website can be created using Flask package under flask library in just seven lines of code.
- The html files which are required to design the webpage are saved in the folder named **templates**.
- The html files **home.html** and **about.html** files (child template) are linked to **layout.html** (parent template) file for navigation menu using **extends** tag.
- For **css styling** of the webpage **main.css** file is created under the folder **static\css**.
- To deploy the web app into a live server, Git software is used. Git is a version control system allowing you to upload the project files to a server and helps track your changes while maintaining the web app.
- **Steps to deploy a static Flask website to Heroku:**
    1. Create an account on www.heroku.com, if you have one already then login to Heroku.
    2. Download and install Heroku Toolbelt from https://devcenter.heroku.com/articles/heroku-cli.
    3. Install **gunicorn** with "**pip install gunicorn**", **gunicorn** is a http server which Heroku needs to run web application.
    4. Create a virtual environment in python using **virtualenv** package.
    5. Create a **requirement.txt file** in the main app directory where the main Python app file is located. You can create that file by running "pip freeze > requirements.txt" in the command line. Make sure you're using pip from your virtual environment if you have one. The requirement.txt file should now contain a list of Python packages.
    6. Create a file named "**Procfile**" in the main app directory. The file should not contain any extension. Then type in this line inside: "**web: gunicorn app4:app**" where "**app4**" should be replaced with the name of your Python script and "**app**" with the name of the variable holding your Flask app.
    7. Create a runtime.txt file in the main app directory and type "**python-3.7.7**" inside. By default, Heroku takes **python-3.6.10** into consideration.

8. Open your computer terminal/command line to point to the directory where the Python file containing your app code is located.
9. Using the terminal, log in to Heroku with command "**heroku login**"
10. Enter your Heroku email address and password.
11. Create a new Heroku app with "**heroku create my_app_name**"
12. Initialize a local git repository with "**git init**"
13. Add your local application files to git with "**git add .**"
14. Tell git your email address with "**git config --global user.email "myemail@gmail.com"**". Make sure the email address is inside quotes here.
15. Tell git your username (just pick whatever username) with "**git config --global user.name "what_ever_username"**”. The username should be in quotes.
16. Commit the changes with "**git commit -m "first commit"**". Make sure "**first commit**" is inside quotes.
17. Before pushing the changes to Heroku, tell Heroku the name of the app you want to use with "**heroku git:remote --app my_app_name**"
18. Push the changes to Heroku with "**git push heroku master**"
19. Open your app with "**heroku open**" command.

- After deploying the web app to Heroku, when you visit the website on the browser you see an error, probably something went wrong during the deployment.
- You can see what you what went wrong during deployment by looking at the server logs. You can access the server logs by running “**heroku logs**” command in the terminal.
- This command will show a series of messages. Carefully read the logs to understand what went wrong.
- If there are any future changes to made (maintenance) can be completed with the help of git software.