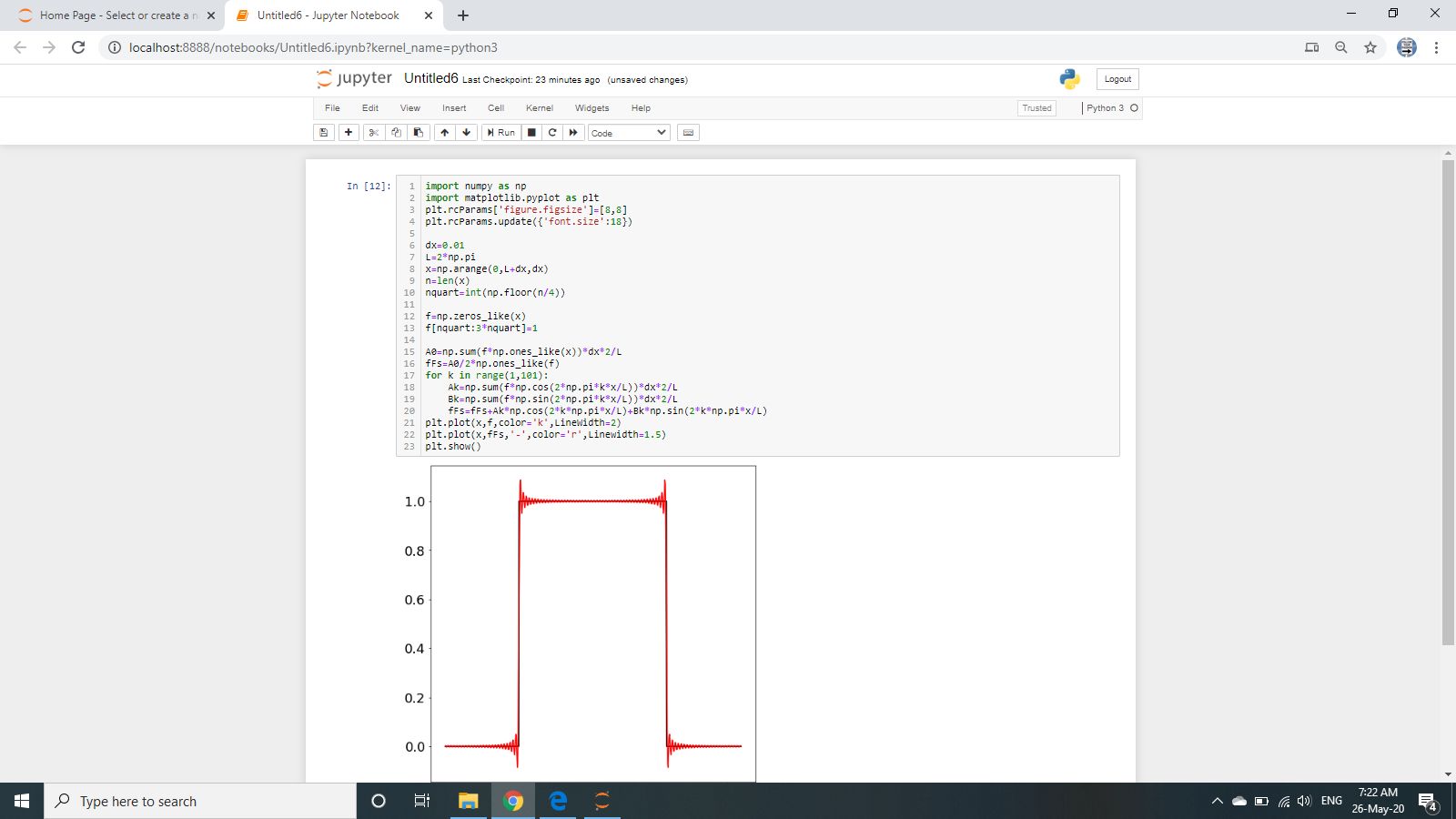
DAILY ASSESSMENT FORMAT

|  |  |  |  |
| --- | --- | --- | --- |
| Date | 26/05/2020 | Name: | Prajna |
| Course: | Digital signal processing | USN: | 4AL16EC047 |
| Topic: | 1.Fourier Series & Gibbs Phenomena Using Python  2.Fourier Transform  3.Fourier Transform and Its Convolution and Derivatives 4.Institution of Fourier Transform and Laplace Transform  5.Laplace Transform of First Order  6. Implementation Using MATLAB, 7. Application Using Z Transform 8.Find the Z-transform of sequence Using MATLAB . | Semester &  Section: | 8 “A” |
| FORENOON SESSION DETAILS | | | |

****

Fourier Series and Fourier Transform Fourier Series:

Fourier Transform:

∞  
*f*(*x*)=1*a*0 +∑(*akcos*2*kt*+*bksin*2*kt*)

−∞

∞

*X*(*F*) = ∫ *x*(*t*)*e*−*j*2*Ftdt* −∞

2

Fourier Series and Gibbs Phenomena Using Python:

import num py as np

import matplotlib.pyplot as pltplt.rcParams['figure.figsize']=[8,8]

plt.rcParams.update({'font.size':18}) dx=0.01  
L=2\*np.pi  
x=np.arange(0,L+dx,dx)  
n=len(x)  
nquart=int(np.floor(n/4)) f=np.zeros\_like(x) f[nquart:3\*nquart]=1 A0=np.sum(f\*np.ones\_like(x))\*dx\*2/L fFs=A0/2\*np.ones\_like(f)  
for k in range(1,101):

Ak=np.sum(f\*np.cos(2\*np.pi\*k\*x/L))\*dx\*2/L Bk=np.sum(f\*np.sin(2\*np.pi\*k\*x/L))\*dx\*2/L fFs=fFs+Ak\*np.cos(2\*k\*np.pi\*x/L)+Bk\*np.sin(2\*k\*np.pi\*x/L)

plt.plot(x,f,color='k',LineWidth=2) plt.plot(x,fFs,'-',color='r',Linewidth=1.5) plt.show()

Laplace Transform [Matlab] clear all;  
close all;  
syms L f t; f=(exp(-3\*t)\*sin(2\*t))/t

L=laplace(f​)

Inverse Laplace Transform clear all;

close all;

symsF,s,x; F=(s+29)/(s^3+4\*s^2+9\*s+36) ilaplace(F,x)

Z Transform Using Matlab

clear all;  
close all;  
symsn,w;

a=sin(w\*n)  
b=ztrans(a)  
disp(b)  
(z\*sin(w))/(z^2 -2\*cos(w)\*z+1) pretty(b)

MATLAB software saves and reduces a lot of time in routine calculations and application for mathematicians, physicist, engineers and scientists.

• In this work Laplace transform are defined and applied for solving an ODE example classically, MATLAB program are used to programming Laplace transformation for initial value problem, second order ordinary differential equations with constant coefficient.

• MATLAB function are constructed, estimates exact solution in less than a second, which makes easy and useful for researcher, and generates the graph of exact solution.

• MATLAB program is possible to simulate the Laplace transformable equations directly which has made a good advancement in the research.

Application of z-transform:

* Z transform is used to convert discrete time domain signal into discrete frequency domain signal.
* It has wide range of applications in mathematics and digital signal processing. It is mainly used to analyze and process digital data.
* For example, to analyze JPEG images, MP3 and MP4 songs, ZIP files etc., we can make use of Z transform.

Applications of Z transform in digital signal processing :

1. Used in system designing

2. Used to find out stability of a system

3. Used to find frequency response of a signal

4. Analysis of linear discrete system

5. For designing digital filters.

MATLAB Program For Z-Transform of Finite Duration Sequence:

Program Code :

%transform of finite duration sequence

clc;

close all;

clear all;

syms 'z';

disp ('If you input a finite duration sequence x(n), we will give you its z-transform');

nf=input ('Please input the initial value of n = ');

nl=input ('Please input the final value of n = ');

x= input ('Please input the sequence x(n)= '); syms 'm';

syms 'y';

f (y, m) =(y\*(z^(-m)));

disp ('Z-transform of the input sequence is displayed below');

k=1;

for n=nf:1:nl

answer(k)=(f((x(k)), n));

k=k+1;

end

disp(sum(answer));

Example of Output :

If you input a finite duration sequence x(n), we will give you its z-transform

Please input the initial value of n = 0

Please input the final value of n = 4

Please input the sequence x(n)= [1 0 3 -1 2]

Z-transform of the input sequence is displayed below:

3/z^2 - 1/z^3 + 2/z^4 + 1

|  |  |  |  |
| --- | --- | --- | --- |
| Date | 26/05/2020 | Name: | Prajna |
| Course: | PYTHON | USN: | 4AL16EC047 |
| Topic: | Loops | Semester &  Section: | 8 “A” |
| AFTERNOON SESSION DETAILS | | | |

If else is common in all languages so Python is the best to learn it.

Orphan if command without else:

* They use elif and not elseif to keep your life easy in typing!

python = 20

anaconda = 30

rattlesnake = 15

if python < anaconda:

print("Too many ana

conda! The world is doomed!")

if python > rattlesnake:

print("Not many cats! The world is saved!")

if rattlesnake < rattlesnake:

print("The world is drooled on!")

if should come with else

python = 30

anaconda = 40

rattlesnake = 15

if anaconda > python:

print("We should take the anaconda.")

elif anaconda < python:

print("We should not take the python.")

else:

print("We can't decide.")

When would the else trigger, search the value for python and anaconda.

One more variation:

python = 20

anaconda = 50

rattlesnake = 15

if python < anaconda:

print("Too many anaconda! The world is doomed!")

print("Again Too many anaconda! The world is doomed!")

if python > rattlesnake:

print("Not many rattlesnake! The world is saved!")

if rattlesnake < anaconda:

print("The world is anaconda on!")

Another short cut to write if else - only in Python!

x = 5

'Non-negative' if x >= 0 else 'Negative'

Trying to extend the if else in one line to its limit:

i=51

a = 1 if i<50 else 2 if i>50 else 0

a

x = 6

y = 7

if x > 5:

x += 1

y = 8

a = 5

b = 7

c = 8

d = 4

if (a < b) | (c > d):

print('Made it')

Or is written in two ways

if a < b or c > d:

print('Made it')

words = stuff.split(' ')

#What will split do?

 a\_list = ['foo', 2, [4, 5]]

a\_list[2] = (3, 4)

a\_list

What is twin and copy?

When your twin is in pain do you feel it?

a = [1, 2, 3]

b=a.copy()

a.append(4)

b

def isiterable(obj):

try:

iter(obj)

return True

except TypeError: # not iterable

return False

isiterable('a string')

isiterable([1, 2, 3])

iter(5)

Difference between is and ==?

a = [1, 2, 3]

b = a

c = list(a)

a is b

a is c

a==c

What is range?

What do you think when you hear this name?

# This creates a list using a range

list(range(10))

Adding two list, and extending a list as you can see.

Let is "BREAK THE CODE"

lst1 =[4, None, 'foo'] + [7, 8, [2, 3]]

lst1[5][1]

x = [[4, None, 'foo']

x.extend([7, 8, (2, 3)])

x

Looping through the key value pairs.

# create a mapping of state to abbreviation

states = {

'Oregon': 'OR',

'Florida': 'FL',

'California': 'CA',

'New York': 'NY',

'Michigan': 'MI'

}

# create a basic set of states and some cities in them

cities = {'CA': 'San Francisco',

'MI': 'Detroit',

'FL': 'Jacksonville'}

# add some more cities

cities['NY'] = 'New York'

cities['OR'] = 'Portland'

# print out some cities

print('-' \* 10)

print("NY State has: ", cities['NY'])

print("OR State has: ", cities['OR'])

# print some states

print('-' \* 10)

print("Michigan's abbreviation is: ", states["Michigan"])

print("Florida's abbreviation is: ", states['Florida'])

# do it by using the state then cities dict

print('-' \* 10)

print("Michigan has: ", cities[states['Michigan']])

print("Florida has: ", cities[states['Florida']])

# print every state abbreviation

print('-' \* 10)

for state, abbrev in states.items():

print(f"{state} is abbreviated {abbrev}")

# print every city in state

print('-' \* 10)

for abbrev, city in cities.items():

print(f"{abbrev} has the city {city}")

# now do both at the same time

print('-' \* 10)

for state, abbrev in states.items():

print(f"{state} state is abbreviated {abbrev}")

print(f"and has city {cities[abbrev]}")

print('-' \* 10)

# safely get an abbreviation by state that might not be there

state = states.get('Texas')

if not state:

print("Sorry, no Texas.")

# get a city with a default values

city = cities.get('FL', 'Does not Exist')

print(f"The city for the state 'TX' is: {city}")