# EML 4535, EML 5530

# MECHANICAL COMPUTER AIDED DESIGN,

# INTERMEDIATE CAD/CAE

# FALL 2022

# PROJECT 2

**Please submit your programs to the following G-Mail address:**

[**fiucad1@gmail.com**](mailto:fiucad1@gmail.com)

**Please use the subject like this:**

**Last name – First name - CAD – Project #**

**Example:**

**Tansel Ibrahim – CAD – Project 2**

Prepare a plot program with the following capabilities:

|  |  |  |
| --- | --- | --- |
| **Segment** | **Value** | **Capability** |
| Entry | 50 | Draw and put the ticks for Y axis |
| Mid | 60 | All the above and activate the Cos button |
| Upper class | 90 | All the above and add tangent button. Tangent may go to infinity. Adjust the maximum and minimum values of the Y axis to see the tangent plot at the screen. Also, let the user to give the inputs as degree and radian |
| Luxury | 100 | All the above and add one button to plot  A sin (2 x) + B sin (4 x)  option. Where A, and B, are the arbitrary coefficients user would give to the program by writing them into the assigned text box you prepared. Program will plot these functions when these coefficients are given and an execute button is pushed. |

Adjustment of the max and min values of the Y-Axis when you plot the tan curve:

For highest credit:  You generate the y values for the tangent.  Search all the values to select the minimum and maximum.  You put the minimum value to the bottom and maximum to the top of the Y axis.  When you generate the tangent values you don’t know what will be the maximum or minimum.  It depends which angle value you will come up when you divide the x axis to 400 or 800 values to make the plot.  So, the ideal is to find them.

Compromise if you can’t make the above approach work:  You may fix the max. and min. of the Y Axis with two values such as 100 to -100 or 10 to -10.  You can find the one which you like by trial and error.  When you make the tangent plot you may change the Y Axis.

**LIST OF THE EXAMPLE PROGRAM**

**# Graphics Program**

**from tkinter import \* # We read the graphic library**

**import tkinter as tk**

**import math**

**#Initilize the TKInter**

**root = Tk()**

**root.title('Plot 1') # This is the title of the window**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Define the Window sizes and draw 2 frames (plot and control windows) inside a large window**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Window size**

**window\_width=int(800)**

**window\_height=int(600)**

**# Window - setup and draw**

**canvas = Canvas(root, width =window\_width, height=window\_height)**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Window 1 - Large external frame**

**x\_and\_y\_offsets=float(0.05) # Space between window and external frame**

**x\_left\_top\_large\_frame=window\_width \* x\_and\_y\_offsets**

**y\_left\_top\_large\_frame=window\_height \* x\_and\_y\_offsets**

**x\_right\_bottom\_large\_frame=window\_width - (window\_width \* x\_and\_y\_offsets)**

**y\_right\_bottom\_large\_frame=window\_height - (window\_height \* x\_and\_y\_offsets)**

**#Draw the frame with a rectangle**

**canvas.create\_rectangle(x\_left\_top\_large\_frame, y\_left\_top\_large\_frame,**

**x\_right\_bottom\_large\_frame, y\_right\_bottom\_large\_frame,**

**outline="blue", fill="grey", width='2')**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Window 2 - Plot frame in the large window - A function is used to redraw this window to clean the area**

**# Plot the "PLOT WINDOW" - Use this function to clean the picture in it - We will erase the curve and both Axes with labels**

**def draw\_plot\_window(x\_left\_offset, x\_right\_offset, y\_top\_bottom\_offsets):**

**x\_left\_top\_plot\_frame=window\_width \* x\_left\_offset**

**y\_left\_top\_plot\_frame=window\_height \* y\_top\_bottom\_offsets**

**x\_right\_bottom\_plot\_frame=window\_width - (window\_width \* x\_right\_offset)**

**y\_right\_bottom\_plot\_frame=window\_height - (window\_height \* y\_top\_bottom\_offsets)**

**#Draw the frame with a rectangle**

**canvas.create\_rectangle(x\_left\_top\_plot\_frame, y\_left\_top\_plot\_frame,**

**x\_right\_bottom\_plot\_frame, y\_right\_bottom\_plot\_frame,**

**outline="red", fill="white", width='2')**

**# Define boundaries and call the plot window drawing function**

**x\_left\_plot\_frame\_offset=float(0.1) # Space between window and plot frame**

**x\_right\_plot\_frame\_offset=float(0.3) # Space between window and plot frame**

**y\_top\_bottom\_plot\_frame\_offsets=float(0.1) # Space between window and plot frame**

**# Draw the window with above boundaries by using the above function**

**draw\_plot\_window(x\_left\_plot\_frame\_offset, x\_right\_plot\_frame\_offset, y\_top\_bottom\_plot\_frame\_offsets)**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Window 3 - Control button frame in the window**

**x\_left\_control\_offset=float(0.72) # Space between window and plot frame**

**x\_right\_control\_offset=float(0.08) # Space between window and plot frame**

**y\_top\_bottom\_control\_offsets=float(0.1) # Space between window and plot frame**

**x\_left\_top\_control\_frame=window\_width \* x\_left\_control\_offset**

**y\_left\_top\_control\_frame=window\_height \* y\_top\_bottom\_control\_offsets**

**x\_right\_bottom\_control\_frame=window\_width - (window\_width \* x\_right\_control\_offset)**

**y\_right\_bottom\_control\_frame=window\_height - (window\_height \* y\_top\_bottom\_control\_offsets)**

**#Draw the frame with a rectangle**

**canvas.create\_rectangle(x\_left\_top\_control\_frame, y\_left\_top\_control\_frame,**

**x\_right\_bottom\_control\_frame, y\_right\_bottom\_control\_frame,**

**outline="red", fill="white", width='2')**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Set the default values to draw the geometric functions**

**# Put the range of the x axis in radian**

**# Define the boundaries of the x variable and display them on the screen**

**x\_start=int(0)**

**x\_end=float(4\*3.14)**

**x\_button\_offset= x\_left\_control\_offset + 0.02 # Percentage of the width of the window**

**y\_button\_offset= y\_top\_bottom\_control\_offsets + 0.02 # Percentage of the height of the window**

**y\_button\_space= 30 # Exact distance between the buttons and input boxes in the control frame**

**x\_button\_location=window\_width\*x\_button\_offset**

**y\_button\_location=window\_height\*(y\_button\_offset+0.02)**

**canvas.create\_text(x\_button\_location, y\_button\_location,**

**anchor=W, font=('verdana', 12),fill='blue',**

**text="X range")**

**textBox1=Text(root, height=1, width=8)**

**textBox1.place(x = x\_button\_location, y = y\_button\_location+y\_button\_space)**

**#textBox1.pack()**

**textBox1.insert(tk.END, x\_start)**

**textBox2=Text(root, height=1, width=8)**

**textBox2.place(x = x\_button\_location, y = y\_button\_location+y\_button\_space\*2)**

**#textBox1.pack()**

**textBox2.insert(tk.END, x\_end)**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Functions to plot the X and Y axes and to put the labels**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Draw X axis**

**def draw\_x\_axis(x\_left\_point,y\_left\_point,x\_right\_point,y\_right\_point):**

**x\_left\_offset=float(0.2) # Space between window and plot frame**

**x\_right\_offset=float(0.35) # Space between window and plot frame**

**y\_top\_bottom\_offsets=float(0.5) # Space between window and plot frame**

**x\_left\_point=window\_width \* x\_left\_offset**

**y\_left\_point=window\_height \* y\_top\_bottom\_offsets**

**x\_right\_point=window\_width - (window\_width \* x\_right\_offset)**

**y\_right\_point=y\_left\_point**

**#Draw the line of the X axis**

**canvas.create\_line(x\_left\_point, y\_left\_point,**

**x\_right\_point, y\_right\_point,**

**fill="black", width='1')**

**# Boundaries of the X axis in the external window coordinates - Use them to make the plots**

**x\_axis\_left=x\_left\_point**

**x\_axis\_right=x\_right\_point**

**x\_axis\_range=x\_right\_point - x\_left\_point**

**x\_axis\_location\_along\_window\_height=y\_left\_point**

**# Put 10 division lines along the X axis**

**n\_divisions=int(10) # Defines how many divisions will be drawn**

**n\_division\_length=int(x\_axis\_range/n\_divisions)**

**x\_division\_height\_ratio=float(0.02) # Division heights will be 3% of the full height of the window**

**x\_division\_height=int(window\_height\*x\_division\_height\_ratio)**

**x\_division\_coordinates=[]**

**print ("X axis has been drawn, coordinates of the divisions along the axis:")**

**for i in range (1, 11):**

**x=x\_axis\_left+n\_division\_length\*i**

**y=x\_axis\_location\_along\_window\_height**

**canvas.create\_line(x,y+x\_division\_height,x,y-x\_division\_height,fill='black',width='1')**

**print(i,x,y)**

**return(x\_axis\_left, x\_axis\_right, x\_axis\_range, x\_axis\_location\_along\_window\_height)**

**# Print the labels of the X - Axis**

**def print\_labels\_x\_axis(x\_axis\_left,x\_axis\_right, x\_axis\_min\_value,x\_axis\_max\_value,x\_axis\_location\_along\_window\_height):**

**# Put the labels along the Y - Axis**

**# Calculate the values to be put as a label**

**number\_labels=int(6)**

**x\_label\_interval=float((x\_axis\_max\_value-x\_axis\_min\_value)/(number\_labels-1))**

**x\_axis\_labels=[]**

**for i in range (0, number\_labels):**

**x\_axis\_labels.append(float(x\_axis\_min\_value+float(x\_label\_interval\*i)))**

**print(x\_axis\_labels)**

**# Calculate the location of the labels in the window coordinates**

**x\_label\_distance=float(x\_axis\_right-x\_axis\_left)**

**x\_label\_interval=float(x\_label\_distance/(number\_labels-1))**

**x\_axis\_label\_location\_along\_y =x\_axis\_location\_along\_window\_height+ int(0.035\*window\_height)**

**x\_axis\_locations=[]**

**print (" The X Axis labels were printed to the following addresses:")**

**for i in range (0, number\_labels):**

**x\_axis\_label\_location\_along\_x= (int(float(x\_axis\_left+float(x\_label\_interval\*i)-0.035\*window\_width)))**

**canvas.create\_text(x\_axis\_label\_location\_along\_x,x\_axis\_label\_location\_along\_y,**

**anchor=W, font=('verdana', 12),fill='blue',**

**text=str(x\_axis\_labels[i]))**

**print(x\_axis\_label\_location\_along\_x, x\_axis\_label\_location\_along\_y)**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Plot a sine function with the defined boundaries of the X axis**

**def plot\_sine\_first(x\_axis\_right,x\_axis\_left,y\_axis\_top,y\_axis\_bottom,amplitude):**

**points=400**

**x\_increment = float(x\_axis\_right-x\_axis\_left)/points # For adjusting for the scale if number of points and window should be matchedx\_factor= 4\*3.14/points # 400 points will bring us from 0 to 4 pi**

**x\_factor= (x\_end-x\_start)/points # 400 points will cover from 0 to 2 pi**

**xy=[]**

**for x in range(points): # means range (0,400) Generate 400 values for the x variable - Start from ZERO**

**# put x coordinates in the LIST they are integers**

**xy.append(int(float(x\_axis\_left)+float(x \* x\_increment))) # to move the axis to the right - to position in frame**

**# y coordinates**

**# Put the second column - these are the y coordiantes of the 2 dimensional list**

**# Since the coordinates increase when we go to bottom we put a minus to correct its effect**

**y=math.sin(x \* x\_factor) \* amplitude**

**y\_scale=float(float(-(y\_axis\_top-y\_axis\_bottom)/float(amplitude\*2)))**

**y\_plot=int(float(y)\*y\_scale)**

**xy.append(-int(y\_plot) + x\_axis\_location\_along\_window\_height)**

**#Plot the sine curve**

**global plot\_curve # Very important to pass the "plot\_curve" to the main program**

**#- you can't erase the old plot without it**

**plot\_curve = canvas.create\_line(xy, fill='red') # We gave a name to this line**

**print ("Sine function is drawn from ",x\_start," to ",x\_end," rad")**

**#print the labels of the y axis**

**y\_axis\_top=y\_axis\_top**

**y\_axis\_bottom=y\_axis\_bottom**

**y\_axis\_max\_value =amplitude**

**y\_axis\_min\_value= - amplitude**

**# Draw the sine function when it is wanted for the second time - cos and tan are the first time**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Erase the existing plot first - Call the function to draw the sine second**

**def prepare\_plot\_area():**

**(x\_start,x\_end)=retrieve\_input()**

**draw\_plot\_window(x\_left\_plot\_frame\_offset, x\_right\_plot\_frame\_offset, y\_top\_bottom\_plot\_frame\_offsets)**

**draw\_x\_axis(x\_left\_point,y\_left\_point,x\_right\_point,y\_right\_point) # We will not get anything back from the function**

**print\_labels\_x\_axis(x\_axis\_left,x\_axis\_right, x\_start,x\_end,x\_axis\_location\_along\_window\_height)**

**def plot\_sine\_multiple(): # Will erase the existing sine wave**

**# canvas.delete(plot\_curve) Deletes the plot of the function - May be used if changing the X axis isn't needed**

**prepare\_plot\_area()**

**plot\_sine\_first(x\_axis\_right,x\_axis\_left,y\_axis\_top,y\_axis\_bottom,amplitude)**

**# Erase the existing plot first - Call the function to draw the sine second**

**def plot\_cos\_multiple(): # Will erase the existing sine wave**

**prepare\_plot\_area()**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Read the values in the textBox1 and textBox2**

**def retrieve\_input():**

**inputValue1=textBox1.get("1.0","end-1c") #It will read whatever is in the box - will read as string**

**inputValue2=textBox2.get("1.0","end-1c") #It will read whatever is in the box - will read as string**

**x\_start=float(inputValue1)**

**x\_end=float(inputValue2)**

**print('X Axis boundaries are ',inputValue1,inputValue2) #Convert to float or int if you will use the read strings in calculation**

**return(x\_start,x\_end)**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MAIN ACTION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Draw the sine wave with defined boundaries first time**

**# Define the X - axis locations and call the function to plot the X axis**

**x\_left\_offset=float(0.2) # Space between window and plot frame**

**x\_right\_offset=float(0.35) # Space between window and plot frame**

**y\_top\_bottom\_offsets=float(0.5) # Space between window and plot frame**

**# Calculate the coorinates of the left and right points of the X axis**

**x\_left\_point=window\_width \* x\_left\_offset**

**y\_left\_point=window\_height \* y\_top\_bottom\_offsets**

**x\_right\_point=window\_width - (window\_width \* x\_right\_offset)**

**y\_right\_point=y\_left\_point**

**# Plot the X Axis**

**(x\_axis\_left, x\_axis\_right, x\_axis\_range, x\_axis\_location\_along\_window\_height)=draw\_x\_axis(x\_left\_point,y\_left\_point,x\_right\_point,y\_right\_point)**

**# Define the Y - axis locations and call the function to plot the Y- Axis**

**x\_left\_offset=float(0.2) # Vertical line we calculate only the X for the Y Axis**

**y\_top\_bottom\_offsets=float(0.2) # Space between plot area and Y- Axis' top and bottom points**

**x\_top\_point=window\_width \* x\_left\_offset**

**y\_top\_point=window\_height \* y\_top\_bottom\_offsets**

**x\_bottom\_point=x\_top\_point**

**y\_bottom\_point=window\_height - (window\_height \* y\_top\_bottom\_offsets)**

**amplitude=50**

**x\_axis\_max\_value=amplitude**

**x\_axis\_min\_value=-amplitude**

**y\_axis\_top=y\_top\_point**

**y\_axis\_bottom=y\_bottom\_point**

**print\_labels\_x\_axis(x\_axis\_left,x\_axis\_right, x\_start,x\_end,x\_axis\_location\_along\_window\_height)**

**plot\_sine\_first(x\_axis\_right, x\_axis\_left,y\_axis\_top,y\_axis\_bottom,amplitude)**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Button to plot the sine wave**

**sine\_button= Button(canvas,text="Sine", command=plot\_sine\_multiple, height=1, width=8, bg="yellow",compound=LEFT)**

**sine\_button.place(x = x\_button\_location, y = y\_button\_location+y\_button\_space\*4)**

**# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**# Button to plot the cos wave**

**cos\_button= Button(canvas,text="Cos", command=plot\_cos\_multiple, height=1, width=8, bg="yellow",compound=LEFT)**

**cos\_button.place(x = x\_button\_location, y = y\_button\_location+y\_button\_space\*5)**

**canvas.pack()**

**root.mainloop()**