FourierSeriesPlotting.py

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import argrelextrema

def fourierSumAtX(x, numberOfFourierSum):
    f = np.array([np.sin((2*i-1)*x) / (2*i-1) for i in range(1, numberOfFourierSum)])
    return f.sum()*4/np.pi

def functionOfX(x):
    if x < np.pi:
        return 1
    else:
        return -1

def getXList(stepSize, upperLimitOfX):
    return [i * stepSize for i in range(int(upperLimitOfX // stepSize) + 2)]

def getFunctionOfXList(stepSize, upperLimitOfX):
    return [functionOfX(i * stepSize) for i in range(int(upperLimitOfX // stepSize) + 2)]

def getFourierList(stepSize, upperLimitOfX, numberOfFourierSum):
    return [fourierSumAtX(i * stepSize, numberOfFourierSum) for i in range(int(upperLimitOfX // stepSize) + 2)]

def plotFunctionAndFourier(stepSize, upperLimitOfX, numberOfFourierSum):
    xList = getXList(stepSize, upperLimitOfX)
    functionList = getFunctionOfXList(stepSize, upperLimitOfX)
    fourierList = getFourierList(stepSize, upperLimitOfX, numberOfFourierSum)
    localMaxima = max(fourierList)
    locationOfLocalMaxima = xList[fourierList.index(localMaxima)-1]
    deltaN = localMaxima-1
    print ("delta N = " + str(deltaN))
    plt.plot(xList, fourierList, label="Fourier Series Approximation")
    plt.plot(xList, functionList, label="Function of X", ls="--", lw = 3.0)
    plt.plot([locationOfLocalMaxima], [localMaxima], marker='o', markersize=5, color="red", label = "Local Maxima = "+str(localMaxima))
    plt.title("Fourier Series (N = " + str(numberOfFourierSum)+")")
    plt.xlabel("x")
    plt.ylabel("f (x)")
    plt.legend()
    plt.savefig("Graph of Fourier Series for N = "+str(numberOfFourierSum))
    plt.show()


plotFunctionAndFourier(0.001, np.pi/2, 40000)
```