191IT234_Niraj_Nandish

September 2, 2020

1 Lab Week 3

1.0.1 Name: Niraj Nandish

1.0.2 Roll no.: 191IT234

1.0.3 Semester: 3

[1]:
```
# Import all required libraries
import numpy as np                # Contains built-in functions to work on arrays
import matplotlib.pyplot as plt   # Use to plot graphs
from scipy import signal          # Generate signals of various types
```

1.1 Question 1

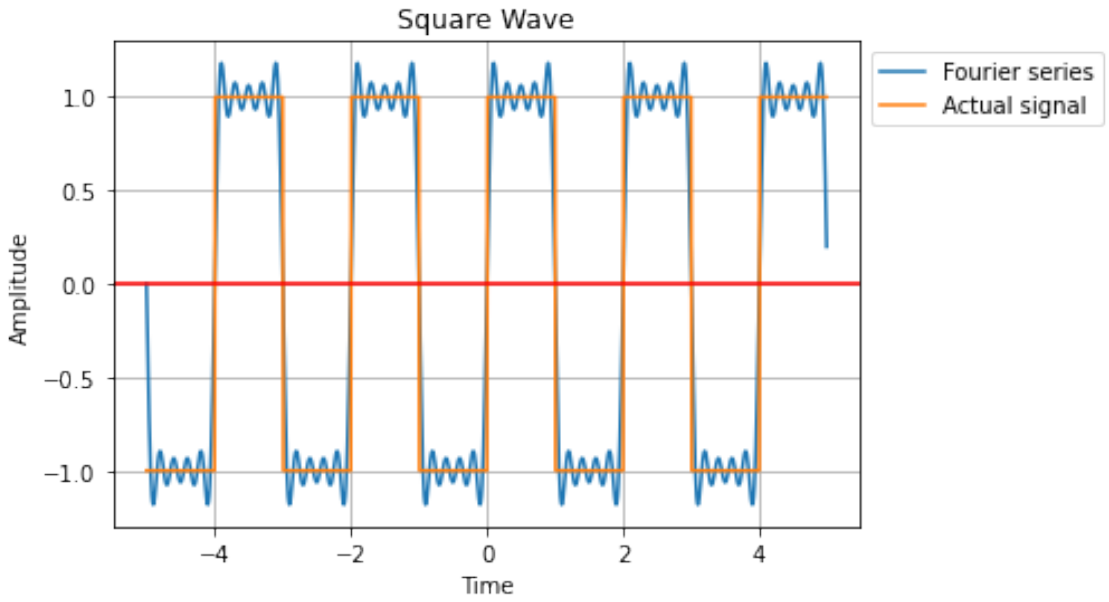
Write the program in python to compute Fourier series of the following periodic signals.

Find the Fourier Coefficients. Plot the amplitude and phase spectrum.

1.1.1 a) Square Wave

[2]:
```
def square(x, num):
    val = 0
    const = 4/(np.pi)
    for i in range(1, x, 2):
        val = val + np.sin(i*np.pi*num)/i
    val = val*const
    return val

time = np.arange(-5, 5, 0.01)
amp = square(10, time)

plt.figure()
plt.plot(time, amp, label="Fourier series")
plt.plot(time, signal.square(time*np.pi), label="Actual signal")
plt.axhline(color='r')
plt.xlabel("Time")
plt.grid()
plt.legend(bbox_to_anchor=(1, 1))
```

```
plt.ylabel("Amplitude")
plt.title("Square Wave")
plt.show()
```

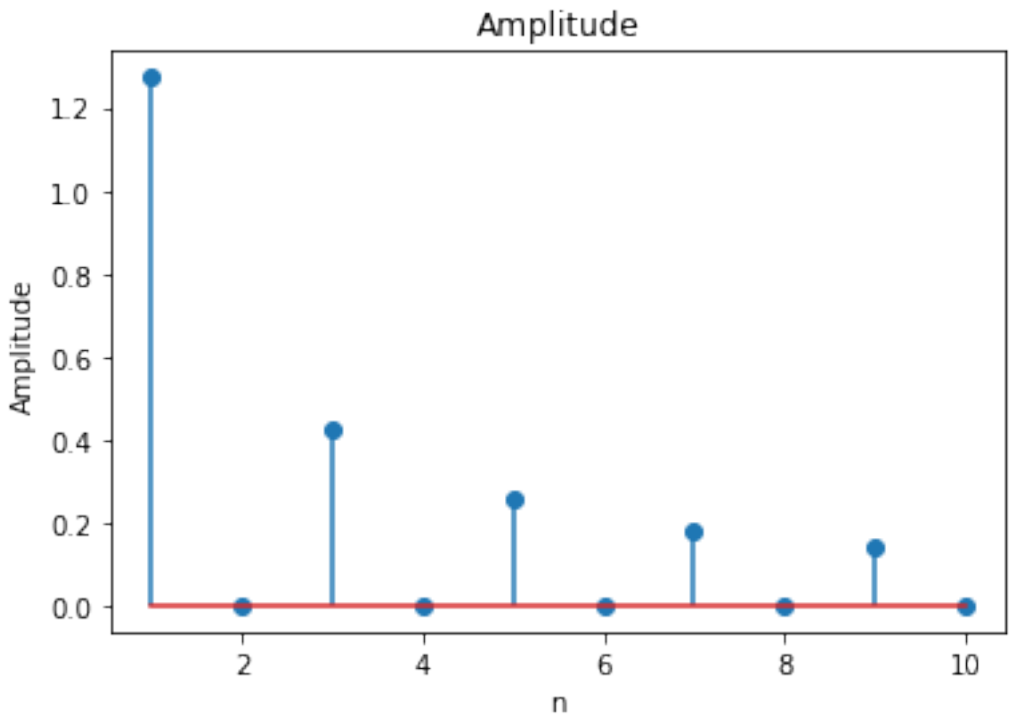


[3]:
```
def square_coeff(x):
    a0 = 0
    an = 0
    bn = []
    for i in range(1, x+1):
        if i % 2 == 1:
            res = 4/(i*np.pi)
            bn.append(res)
        else:
            bn.append(0)
    return a0, an, bn


a0, an, bn = square_coeff(10)
print("The Fourier Coeffients of the square wave are:")
print("a0 = 0\nan = 0")
for i in range(len(bn)):
    print(f"b{i+1} = {bn[i]}")
```

```
The Fourier Coeffients of the square wave are:
a0 = 0
an = 0
b1 = 1.2732395447351628
```

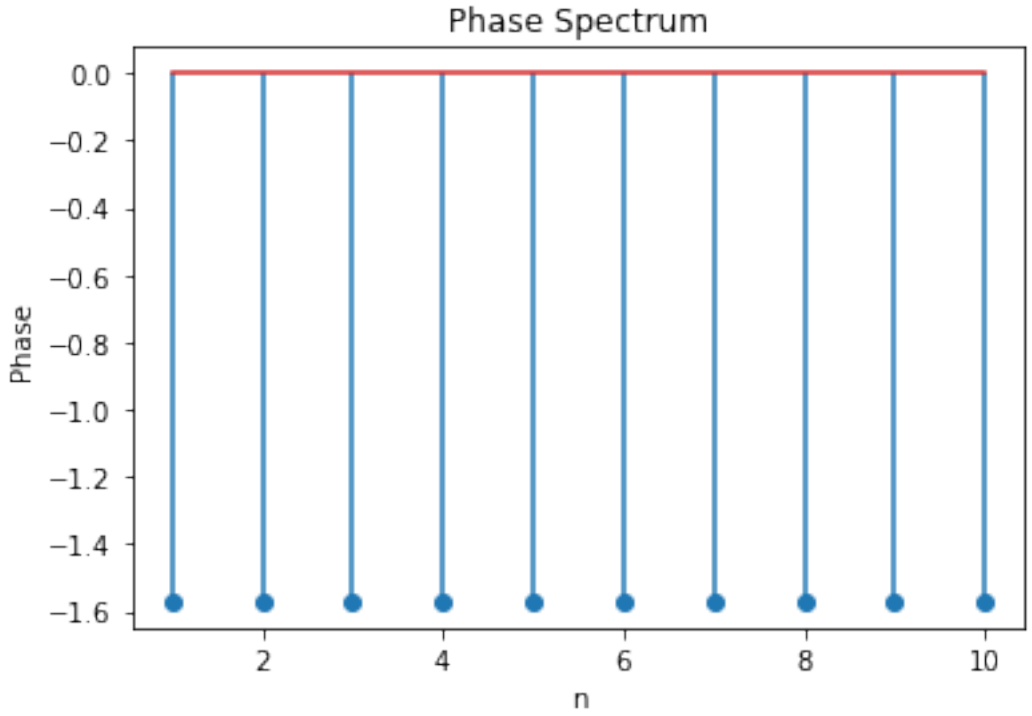
```
b2 = 0
b3 = 0.4244131815783876
b4 = 0
b5 = 0.25464790894703254
b6 = 0
b7 = 0.18189136353359467
b8 = 0
b9 = 0.1414710605261292
b10 = 0
```

[4]:
```
n = np.arange(1,11)
amp = [abs(i) for i in bn]

plt.figure()
plt.stem(n, amp, use_line_collection=True)
plt.xlabel("n")
plt.ylabel("Amplitude")
plt.title("Amplitude")
plt.show()
```



[5]:
```
phase = []
an = np.exp(-50)

for i in bn:
```

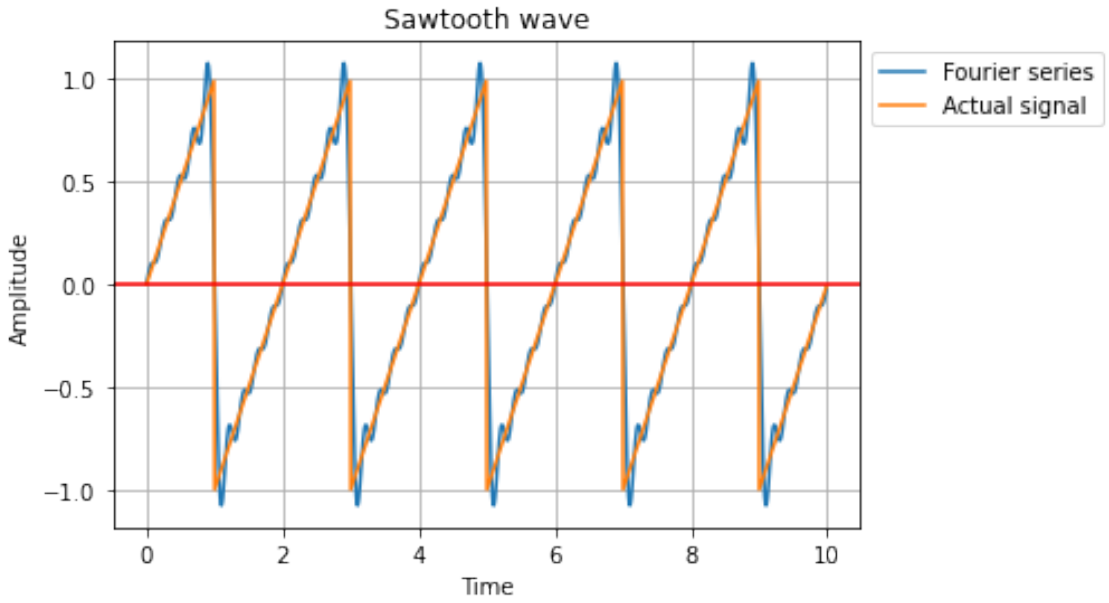
```
    if i == 0:
        i = np.exp(-20)
    phase.append(-np.arctan(i/an))

plt.figure()
plt.stem(n, phase, use_line_collection=True)
plt.xlabel("n")
plt.ylabel("Phase")
plt.title("Phase Spectrum")
plt.show()
```



1.1.2 b) Sawtooth Wave

[6]:
```
def sawtooth(x, num):
    val = 0
    const = -2/(np.pi)
    for i in range(1, x):
        if i%2 == 1:
            val = val - const*np.sin(i*np.pi*num)/i
        else:
            val = val + const*np.sin(i*np.pi*num)/i
    return val
```

```
time = np.arange(0, 10, 0.01)
saw = sawtooth(10, time)

plt.figure()
plt.plot(time, saw, label="Fourier series")
plt.plot(time, signal.sawtooth(time*np.pi+np.pi), label="Actual signal")
plt.axhline(color='r')
plt.grid()
plt.legend(bbox_to_anchor=(1,1))
plt.xlabel("Time")
plt.ylabel("Amplitude")
plt.title("Sawtooth wave")
plt.show()
```

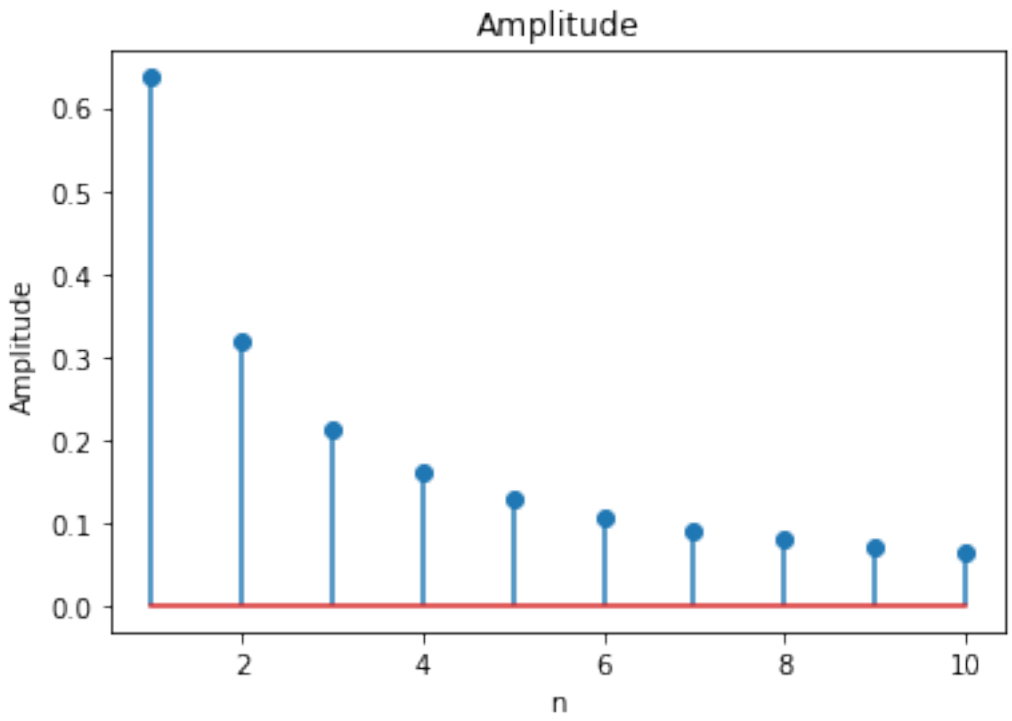


[7]:
```
def saw_coeff(x):
    a0 = 0
    an = 0
    bn = []
    const = -2/(np.pi)
    for i in range(1, x+1):
        if i%2 == 0:
            val = const/i
        else:
            val = -const/i
        bn.append(val)
    return a0, an, bn
```

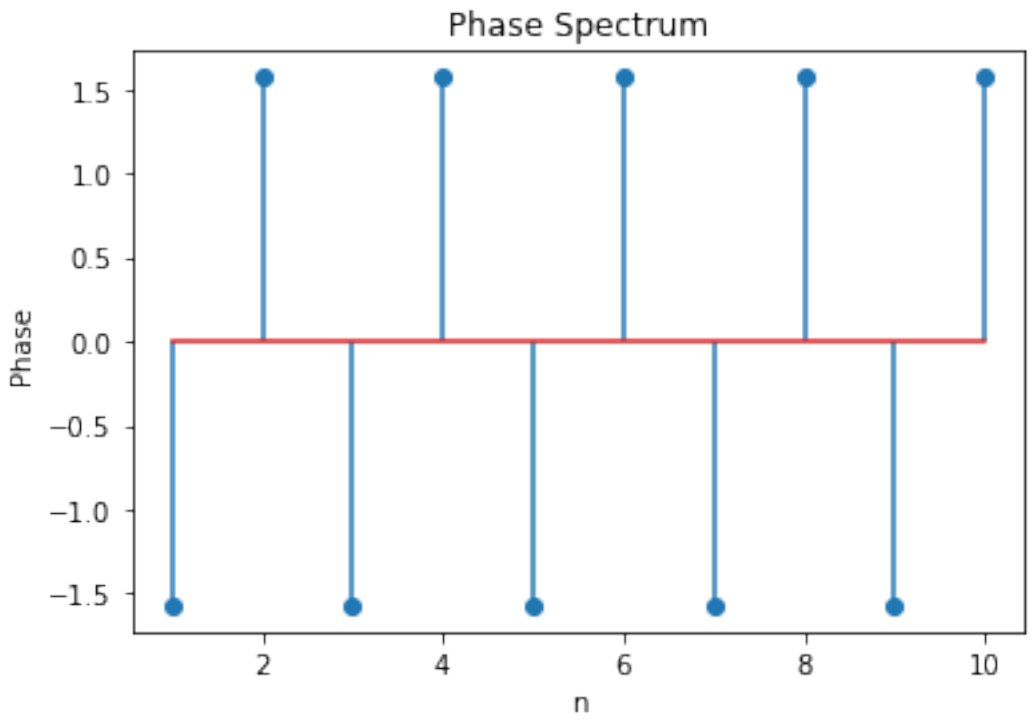
```
a0, an, bn = saw_coeff(10)
print("The Fourier Coeffients of the sawtooth wave are:")
print("a0 = 0\nan = 0")
for i in range(len(bn)):
    print(f"b{i+1} = {bn[i]}")
```

```
The Fourier Coeffients of the sawtooth wave are:
a0 = 0
an = 0
b1 = 0.6366197723675814
b2 = -0.3183098861837907
b3 = 0.2122065907891938
b4 = -0.15915494309189535
b5 = 0.12732395447351627
b6 = -0.1061032953945969
b7 = 0.09094568176679733
b8 = -0.07957747154594767
b9 = 0.0707355302630646
b10 = -0.06366197723675814
```

[8]:
```
n = np.arange(1,11)
amp = [abs(i) for i in bn]

plt.figure()
plt.stem(n, amp, use_line_collection=True)
plt.xlabel("n")
plt.ylabel("Amplitude")
plt.title("Amplitude")
plt.show()
```



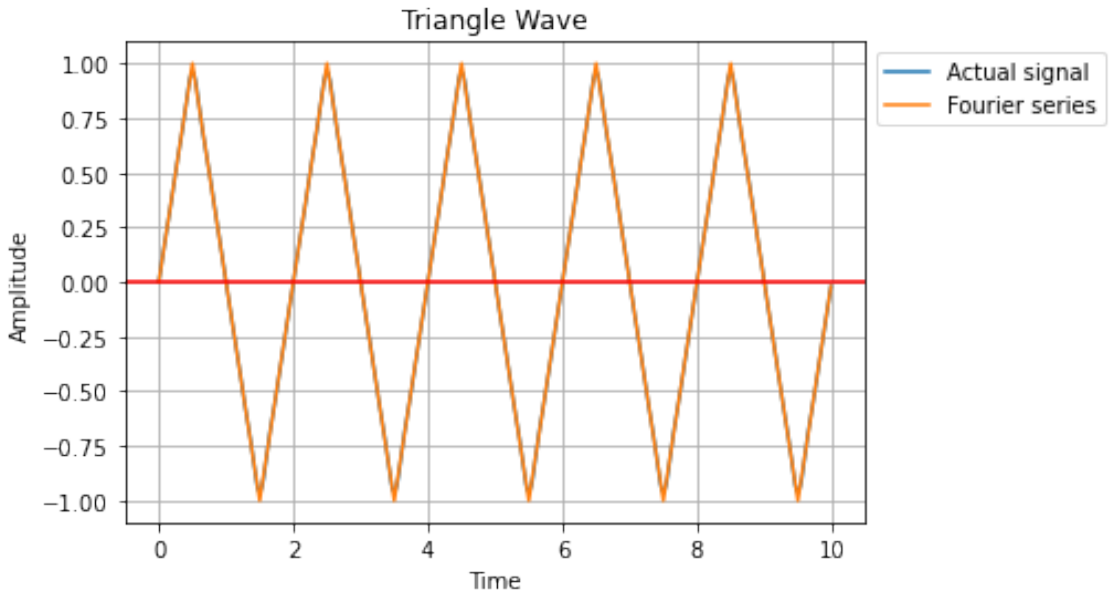
[9]:
```
phase = []
an = np.exp(-50)

for i in bn:
    phase.append(-np.arctan(i/an))

plt.figure()
plt.stem(n, phase, use_line_collection=True)
plt.xlabel("n")
plt.ylabel("Phase")
plt.title("Phase Spectrum")
plt.show()
```



1.1.3 c) Triangular Wave

[10]:
```
def triangle(x, num):
    val = 0
    const = 8/(np.pi)**2
    for i in range(1, x, 2):
        if (i-1)%4 == 0:
            val = val + const*np.sin(i*np.pi*num)/(i*i)
        else:
            val = val - const*np.sin(i*np.pi*num)/(i*i)
    return val

time = np.arange(0, 10, 0.01)
tri = triangle(10, time)

plt.figure()
plt.plot(time, tri, label="Actual signal")
plt.plot(time, signal.sawtooth(time*np.pi+np.pi/2, 0.5), label="Fourier series")
plt.axhline(color='r')
plt.grid()
plt.legend(bbox_to_anchor=(1,1))
plt.xlabel("Time")
plt.ylabel("Amplitude")
```

```
plt.title("Triangle Wave")
plt.show()
```

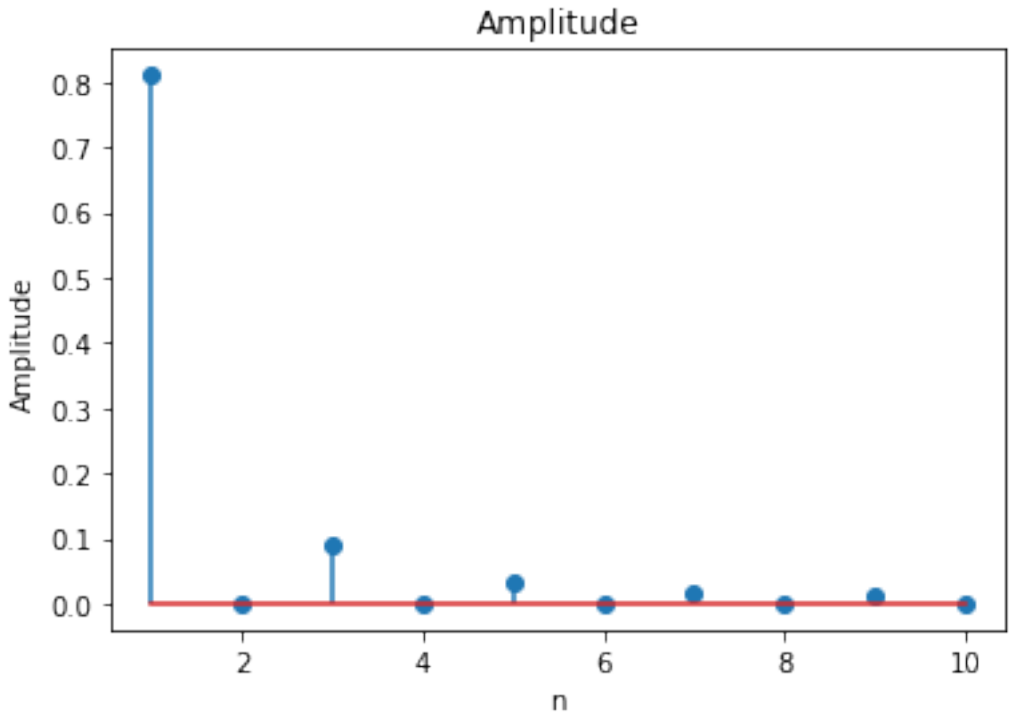


[11]:
```
def triag_coeff(x):
    a0 = 0
    an = 0
    bn = []
    for i in range(1, x+1):#
        if i % 2 == 1:
            if (i-1) % 4 == 0:
                temp = -8/((i**2)*(np.pi**2))
            else:
                temp = 8/((i**2)*(np.pi**2))
            bn.append(temp)
        else:
            bn.append(0)
    return a0, an, bn

a0, an, bn = triag_coeff(10)
print("The Fourier Coeffients of the triangle wave are:")
print("a0 = 0\nan = 0")
for i in range(len(bn)):
    print(f"b{i+1} = {bn[i]}")
```

```
The Fourier Coeffients of the triangle wave are:
a0 = 0
an = 0
```

```
b1 = -0.8105694691387022
b2 = 0
b3 = 0.09006327434874468
b4 = 0
b5 = -0.03242277876554809
b6 = 0
b7 = 0.016542234064055146
b8 = 0
b9 = -0.010007030483193855
b10 = 0
```

[12]:
```
n = np.arange(1,11)
amp = [abs(i) for i in bn]

plt.figure()
plt.stem(n, amp, use_line_collection=True)
plt.xlabel("n")
plt.ylabel("Amplitude")
plt.title("Amplitude")
plt.show()
```



[13]:
```
phase = []
an = np.exp(-50)
```

```
for i in bn:
    phase.append(-np.arctan(i/an))

plt.figure()
plt.stem(n, phase, use_line_collection=True)
plt.xlabel("n")
plt.ylabel("Phase")
plt.title("Phase Spectrum")
plt.show()
```