Fourier Series

A brief explaination of the series that can take form of any periodic function

A Fourier series is the representation of a periodic function using trignometric function series. The Fourier series can be used only to represent periodic functions, non periodic functions are represented using Fourier transform. Our main focus is to study Fourier series and how it can encapsulate various signals within its trignometric curves.

Jean-Baptiste Joseph Fourier came up with the Fourier series while trying to find solutions to heat equation. He was able to represent a complicated heat source as a superposition of simple functions, and therefore be to  derive a general solution.

$$JosephFourier$$

The most basic form of Fourier series is represented as given below:

$$\sum_{n=1}^{N} A_n \sin(2\pi nt + \phi_n)$$

The above form explicitly displays the amplitude and phase of each harmonic, but it is difficult to calculate. Different representations of Fourier series are given below:

$$\sum_{n=1}^{N} (a_n \cos(2\pi nt) + b_n \sin(2\pi nt))$$

Including the a constant term for n = 0, we get,

$$\frac{a_0}{2} + \sum_{n=1}^{N} (a_n \cos(2\pi nt) + b_n \sin(2\pi nt))$$

The combined term has a time period of 1. The above term can still be changed by replacing cos and sin terms with complex exponentials

$$\cos t = \frac{e^{it} + e^{-it}}{2}, \sin t = \frac{e^{it} - e^{-it}}{2}$$

Replacing the original equation we above terms, we get

$$\sum_{n=-N}^{N} c_n e^{2\pi int}$$

Here the coefficient c_n can be calculated as

$$c_n = \int_0^1 e^{-2\pi int} f(t) dt$$

If the time period of is not 1, Fourier series if given by,

$$\sum_{n=-N}^{N} c_n e^{2\pi int/T}$$

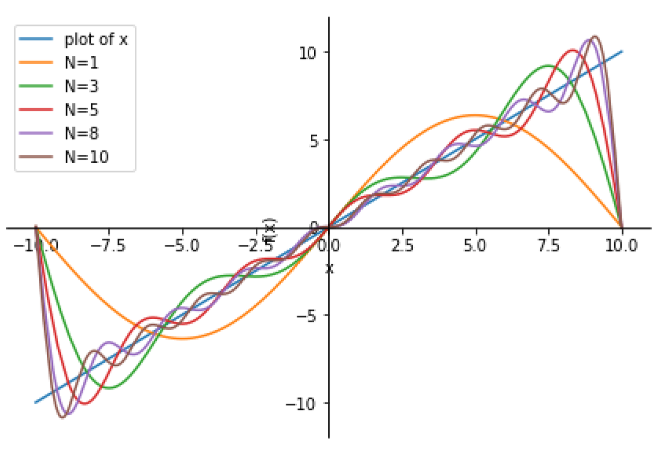
where T = time period and,

$$c_n = \frac{1}{T} \int_0^T e^{-2\pi int/T} f(t) dt$$

The accuracy of approximation of Fourier series depends on the number of terms it uses. Idealy to produce the exact same function using Fourier series number of terms should be infinite. But in real life conditions very few number of terms can approximate Fourier series very close to the function

Now we are going to approximate Fourier series for some simple curves and analyse the results. We would compare the approximation for different number of terms used in the series

In [30]:
```
from sympy import fourier_series, pi, plot
from sympy.abc import x
f = x
s = fourier_series(f, (x, -10, 10))
s1 = s.truncate(n = 1)
s2 = s.truncate(n = 3)
s3 = s.truncate(n = 5)
s4 = s.truncate(n = 8)
s5 = s.truncate(n = 10)
p = plot(f, s1, s2, s3, s4, s5, (x, -10, 10), show=False, legend=True)
p[0].label = 'plot of x'
p[1].label = 'N=1'
p[2].label = 'N=3'
p[3].label = 'N=5'
p[4].label = 'N=8'
p[5].label = 'N=10'
p.show()
```

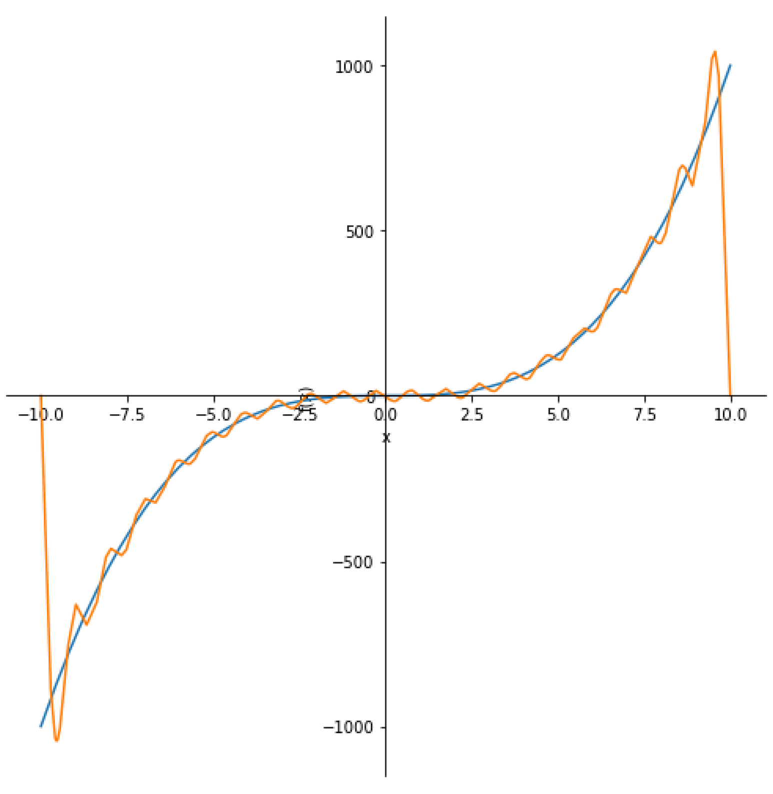


The above grpah shows comparison between the accuracy of the Fourier series approximation of a striaght line for different number of terms

In [24]:
```
from sympy import *
from sympy.abc import x
from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets
%matplotlib inline


def f(n):
    g = x**3
    s = fourier_series(g, (x, -10, 10)).truncate(20)
    p = plot(g, s, size=(7, 7))

w = interactive(f, n=widgets.IntSlider(min=1, max=100, step=1, value=5, description='N'))
display(w)
```

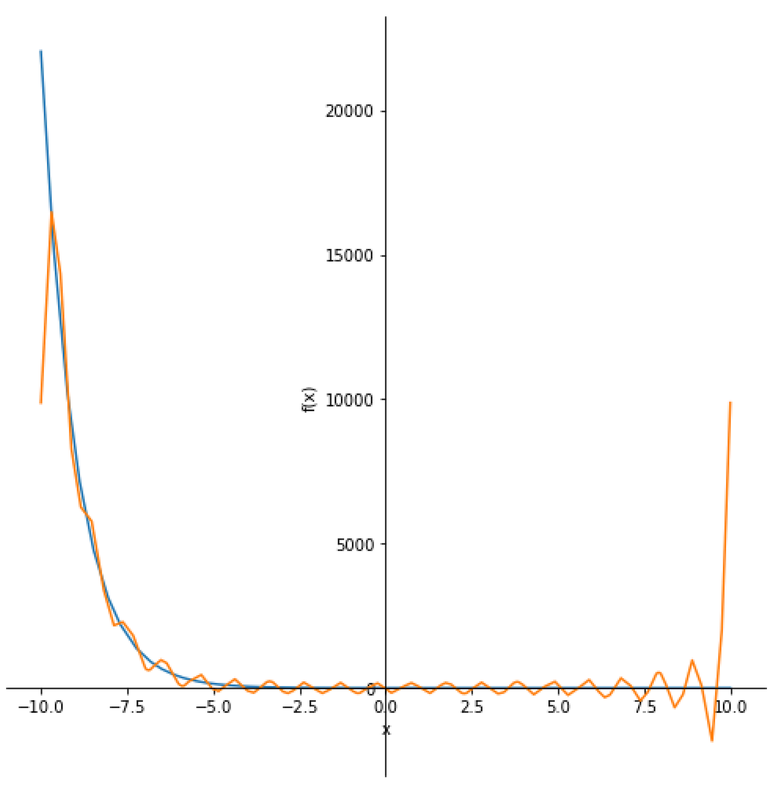


The above graph shows the comparison between a signal following a cubic graph and a fourier series approximation of that function from (-10, 10). The slider for N can be used to increase or decrease the number of terms. Notice that higher the N, better the approximation of the curve. Similar fourier series approximations are given for curves below

In [23]:
```
import numpy as np
import sympy as sy
from sympy import *
from sympy.abc import x
from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets
%matplotlib inline

def f(n):
    g = sy.exp(-x)
    s = fourier_series(g, (x, -10, 10)).truncate(20)
    plot(g, s, size=(7, 7))

w = interactive(f, n=widgets.IntSlider(min=1, max=100, step=1, value=5, description='N'))
display(w)
```



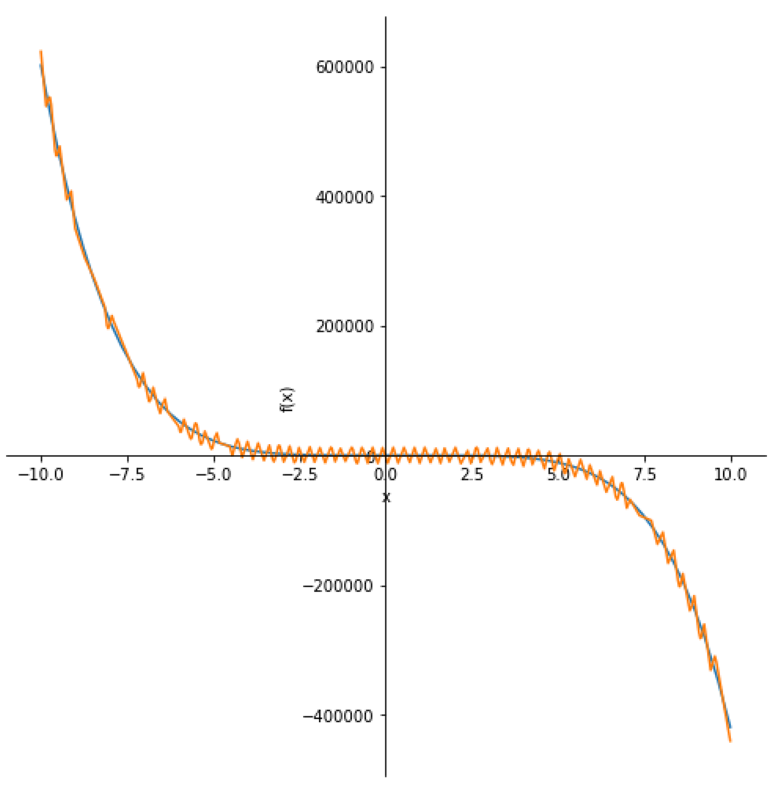
Out[23]:
```
<sympy.plotting.plot.Plot at 0x2c2f597b820>
```

Fourier series approximation of exponent of x

In [26]:
```
import numpy as np
import sympy as sy
from sympy import *
from sympy.abc import x
from ipywidgets import interact, interactive, fixed
import ipywidgets as widgets
%matplotlib inline

def f(n):
    g = 10*x**2-10*x**3+9*x**4-5*x**5
    s = fourier_series(g, (x, -15, 15)).truncate(100)
    plot(g, s, size=(7, 7))

w = interactive(f, n=widgets.IntSlider(min=1, max=1000, step=1, value=5, description='N'))
display(w)
```



Out[26]:
```
<sympy.plotting.plot.Plot at 0x2c2f5ed8ca0>
```

The Fourier approximation of a polynomial

We can observe that while certain signals can be approximated to a very close value for a small number of terms, some signals require high number of terms to be approximated closely.

This was a brief introduction the the Fourier series. These series are used in almost every field due to its property of taking shape of any function. Hence using these series we can easily study different types of periodic signals.