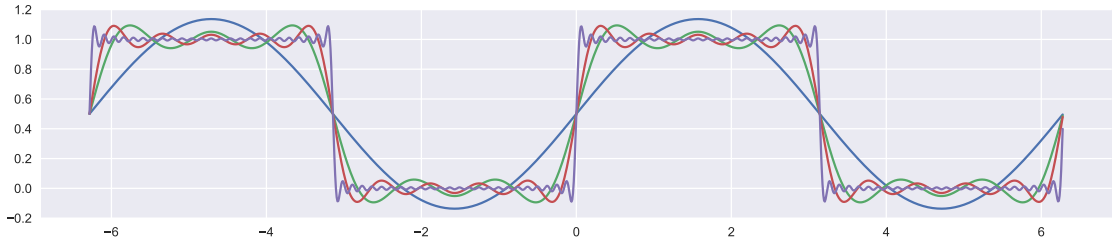
square wave

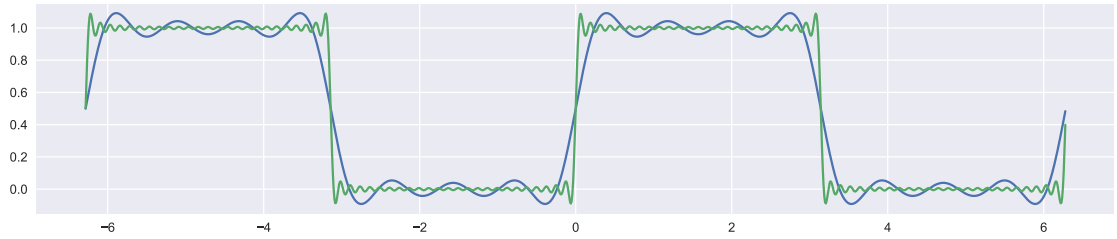
March 11, 2021

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
from numpy import pi

from IPython.display import set_matplotlib_formats
plt.style.use("seaborn")
set_matplotlib_formats('svg', 'pdf')
plt.rc('figure', figsize=(15, 3))
```

[2]:
```
def fun(x, TERMS=10):
    sum_ = 0
    N = np.arange(1, TERMS + 1, 2)
    for i in N:
        sum_ += np.sin(i * x) / i

    return 1 / 2 + 2 / np.pi * sum_


x = np.arange(-2*np.pi, 2 * np.pi, 0.01)

plt.plot(x, fun(x, 1),)
plt.plot(x, fun(x, 5),)
plt.plot(x, fun(x, 10),)
plt.plot(x, fun(x, 50),)
plt.show()
```



[3]:
```
# Evaluate the value of `x` for the integral `LIMIT`
def squareFourier(x, LIMIT=10):
    result = 0
    for n in np.arange(-LIMIT, LIMIT + 1, 1):
        if (n == 0):
            result += 1/2

        # for even `n`
        elif (n % 2 == 0):
            result += 0

        # for odd `n`
        else:
            result += 1 / (pi * 1j * n) * np.exp(1j * n * x)

    return result



x = np.arange(-2*np.pi, 2 * np.pi, 0.01)
y1 = [squareFourier(x_, 7).real for x_ in x]
y2 = [squareFourier(x_, 50).real for x_ in x]

plt.plot(x, y1)
plt.plot(x, y2)
plt.show()
```



[ ]: