Problem 1: Basic Matlab Concepts

The Taylor series for the exponential function e^x is:

$$1+\frac{x^1}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots$$

Write a MATLAB script to compute the exponential for $x = 1, 3, 9$ using the Taylor series. Also compute the exponential using the built-in MATLAB function `exp`. What is the number of terms required for the approximation to be accurate to within 0.01? Run your program several times with the number of terms N=1, 10, 100 in the Taylor series (consider the term 1 is for N=0) to see the effect this has on the closeness of the approximation. Discuss this effect briefly in your report. (Hint: The MATLAB function for calculating factorial is `factorial`.)

My MatLab script to compute the Taylor series:

```
x=9;
n=100;
Series(t) = (x^t)/(factorial(t));
value=0;
while (n>=0)
value = value + Series(n);
n = n-1;
end
double(value)
```

Where I changed the values of x and n.

For x=1

N=1, value = 2
N=10, value = 2.7183
N=100, value = 2.7183

For x=3

N=1, value = 4
N=10, value = 20.0797
N=100, value = 20.0855

For x=9

N=1, value = 10
N=10, value = 5.7207e3
N=100, value = 8.1031e3

Using the exp(x) function to confirm these findings:

```
%double(value)
exp(1)
exp(3)
exp(9)
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> problem1

ans =

    2.7183


ans =

   20.0855


ans =

   8.1031e+03

>>
```

Shows that increasing N increases precision. This is due to the way Taylor series works in that the infinite summation:

$$\sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Where for any n, the maximum error is the value of the equation when n=n+1

Problem 2:

Suppose for the following function

$$f(x) = \begin{cases} 1, & -1 < x < 0 \\ x, & 0 \le x < 1 \end{cases}$$

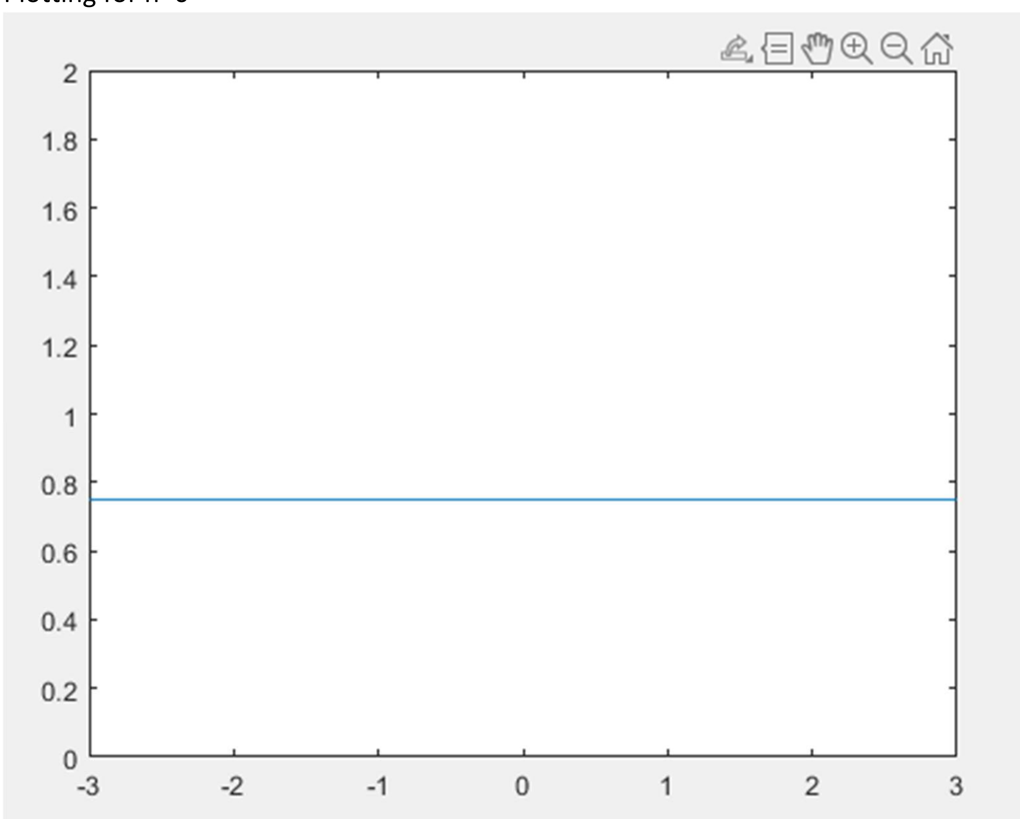
you have expanded in a Fourier series:

$$f(x) = \frac{3}{4} + \sum_{n=1}^{\infty} \left[\frac{(-1)^n - 1}{n^2 \pi^2} \cos n\pi x - \frac{1}{n\pi} \sin n\pi x \right]$$

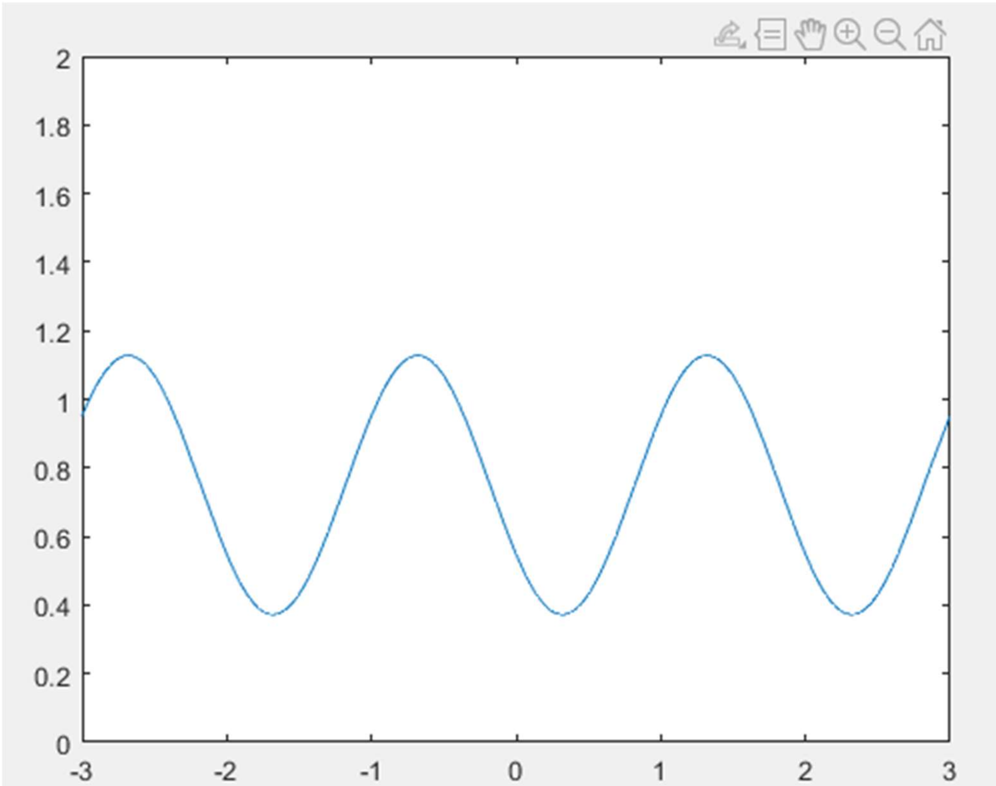
Using MATLAB, plot the periodic extensions of the function *f(x)* in the Fourier series in the x range from -3 to 3 for

1) n goes to 0 (the first term that is outside the Σ)
2) n goes to 1
3) n goes to 2
4) n goes to 3
5) n goes to 100

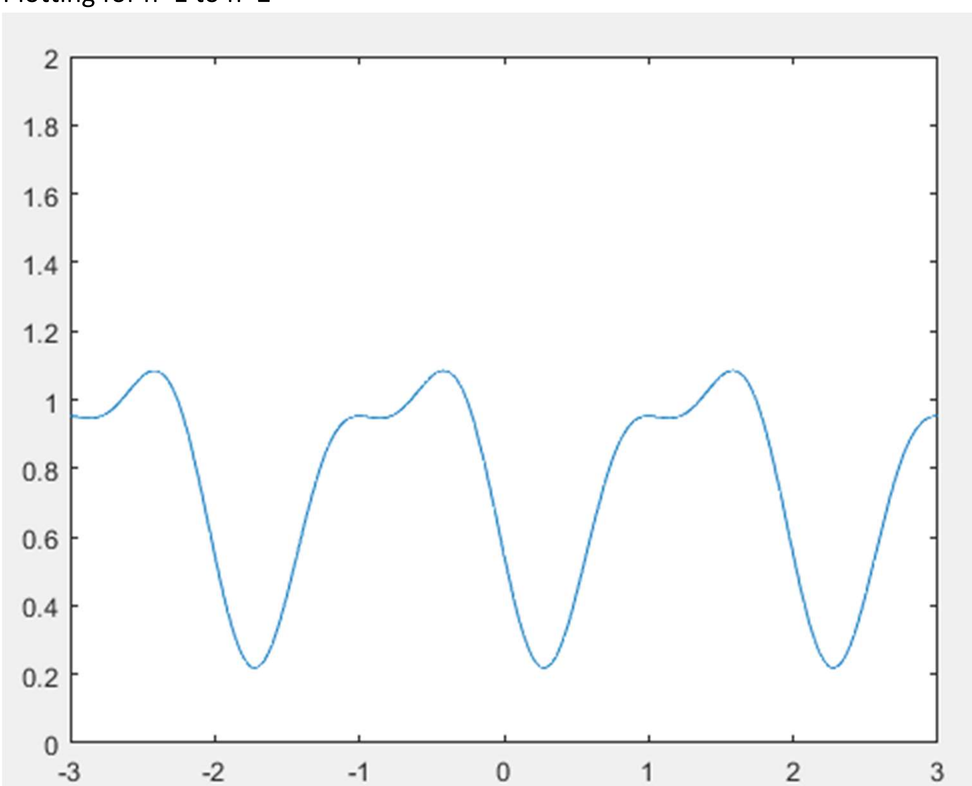
1) Plotting for n=0



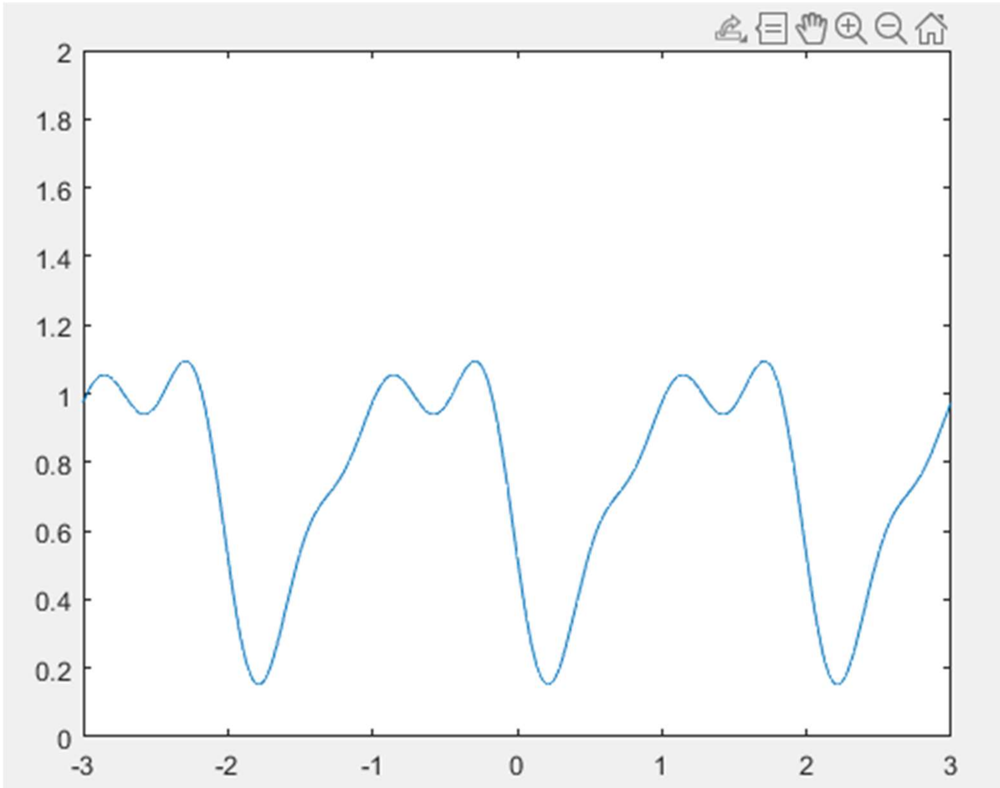
2) Plotting for n=0 and n=1



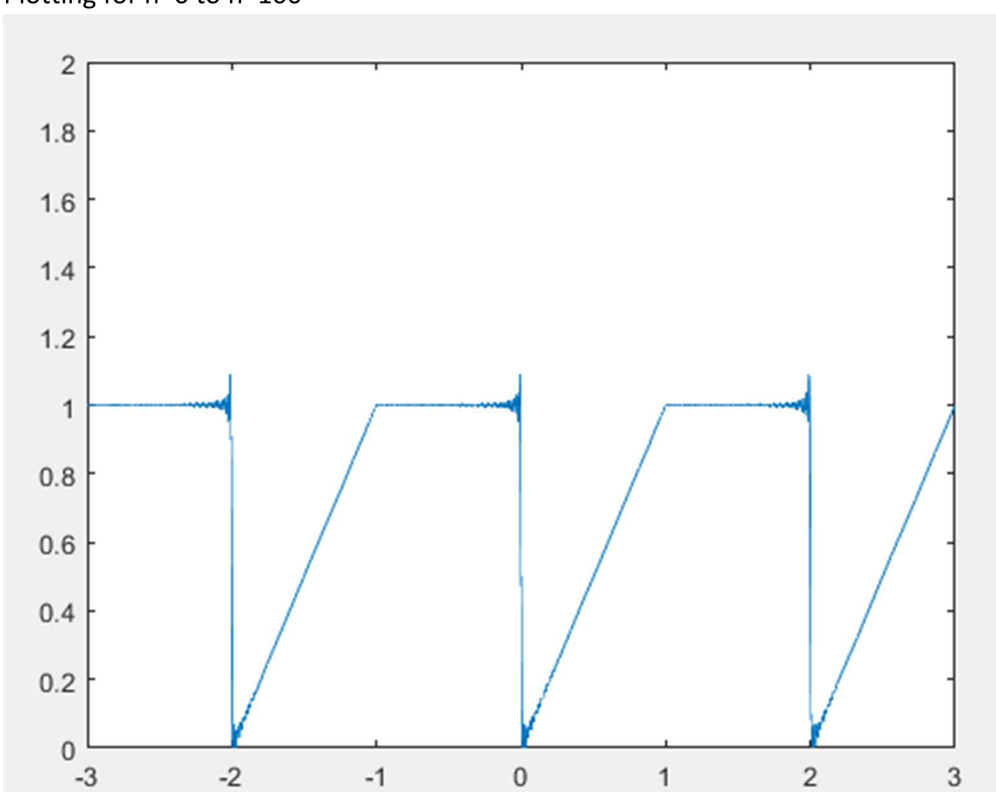
3) Plotting for n=1 to n=2



4) Plotting for n=0 to n=3



5) Plotting for n=0 to n=100



problem1.m | problem2.m

```
syms x n
r(n) = (((((-1)^n)-1)/((n^2)*(pi^2)))*cos(n*pi*x)-(1/(n*pi))*sin(n*pi*x));

g = symsum(r(n),n,1,100);
f = (3/4) + g;

fplot(f);
axis([-3,3,0,2]);
```

Figure 1. Code used to plot the summation for different bounds of N.