**For the octave online**

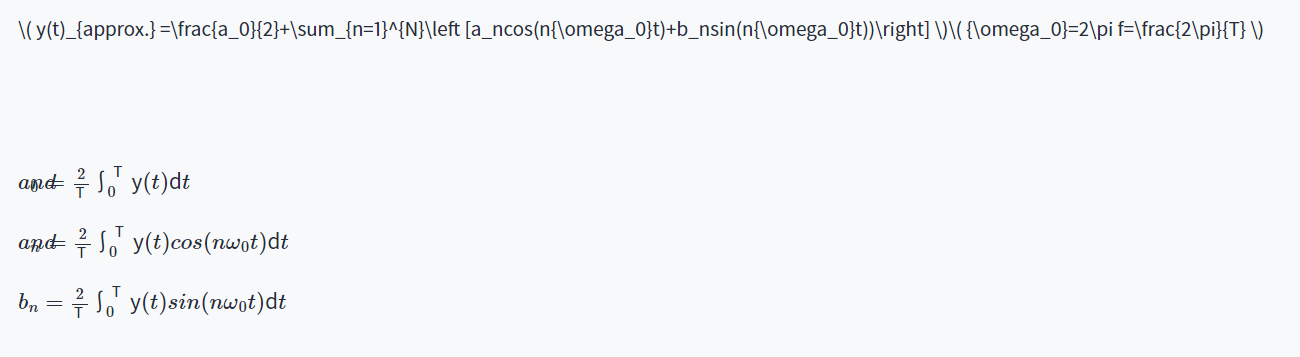
**Approximate the given periodic signal y(t) by a Fourier series, it will be a signal y(t) = t, where t = <-pi, + pi>.**

**To display the resulting approximated waveform, select the appropriate sampling frequency.**

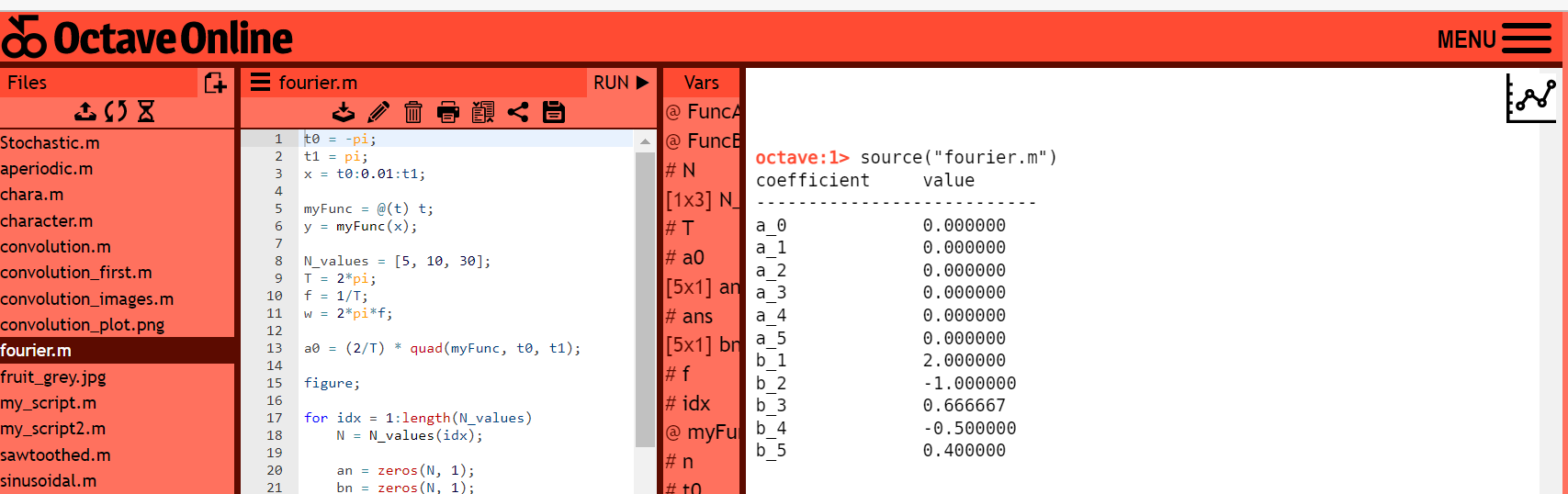
**1) Create an algorithm for calculating the coefficients of the Fourier series according to the assignment.**

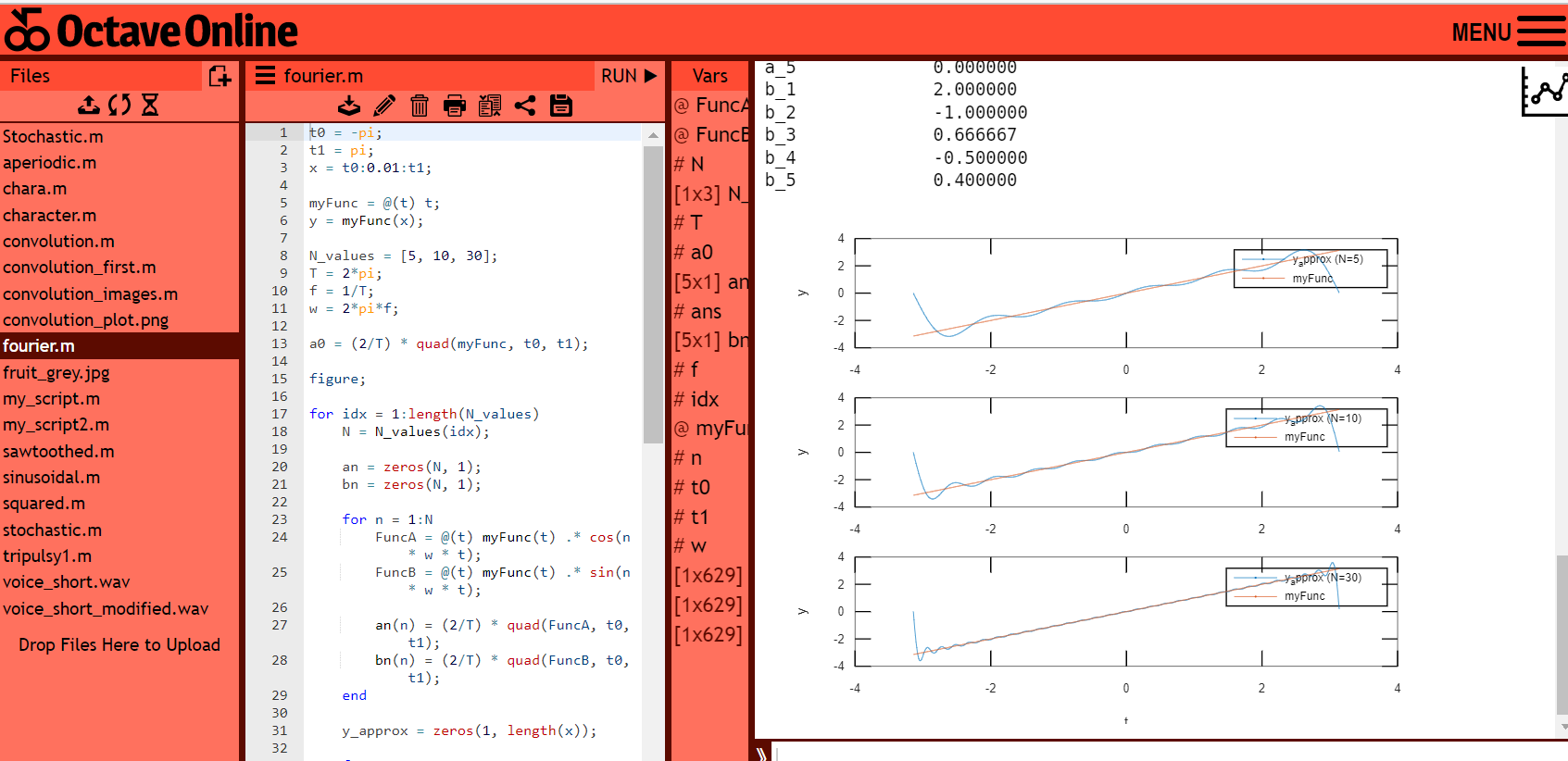
**2) Approximate the Fourier series sequentially for the number of members N = 5, 10 and 30 and graphically display 3 signal periods.**

**3) Write the coefficients of the Fourier series in a clear table (only for N = 5).**



**Screenshots:**





**Code in Octave online:**

t0 = -pi;

t1 = pi;

x = t0:0.01:t1;

myFunc = @(t) t;

y = myFunc(x);

N\_values = [5, 10, 30];

T = 2\*pi;

f = 1/T;

w = 2\*pi\*f;

a0 = (2/T) \* quad(myFunc, t0, t1);

figure;

for idx = 1:length(N\_values)

N = N\_values(idx);

an = zeros(N, 1);

bn = zeros(N, 1);

for n = 1:N

FuncA = @(t) myFunc(t) .\* cos(n \* w \* t);

FuncB = @(t) myFunc(t) .\* sin(n \* w \* t);

an(n) = (2/T) \* quad(FuncA, t0, t1);

bn(n) = (2/T) \* quad(FuncB, t0, t1);

end

y\_approx = zeros(1, length(x));

for n = 1:N

y\_approx = y\_approx + an(n) \* cos(n \* w \* x) + bn(n) \* sin(n \* w \* x);

end

subplot(length(N\_values), 1, idx);

plot(x, y\_approx);

hold on;

plot(x, y);

legend(['y\\_approx (N=' num2str(N) ')'], 'myFunc');

xlabel('t');

ylabel('y');

end

% Table for N = 5

N = 5;

an = zeros(N, 1);

bn = zeros(N, 1);

for n = 1:N

FuncA = @(t) myFunc(t) .\* cos(n \* w \* t);

FuncB = @(t) myFunc(t) .\* sin(n \* w \* t);

an(n) = (2/T) \* quad(FuncA, t0, t1);

bn(n) = (2/T) \* quad(FuncB, t0, t1);

end

disp('coefficient value');

disp('---------------------------');

fprintf('a\_0\t\t%f\n', a0);

for n = 1:N

fprintf('a\_%d\t\t%f\n', n, an(n));

end

for n = 1:N

fprintf('b\_%d\t\t%f\n', n, bn(n));

end

**Comment:**

The code above demonstrates the use of Fourier series to approximate a given function by summing up an infinite series of sine and cosine functions with different coefficients. The given function is defined using an anonymous function, and its domain is defined by the values of t0, t1, and the step size of 0.01. The code then defines the number of coefficients to be used in the approximation, which is stored in the N\_values array. The code then calculates the Fourier coefficients for each value of N using the quad function, which performs numerical integration. The calculated coefficients are then used to construct the Fourier series approximation of the given function, and the resulting plots are displayed using subplots. Finally, a table is printed for the coefficients for the case where N is equal to 5. The table lists the values of the coefficients a0, an, and bn, which represent the constant, cosine, and sine terms in the Fourier series expansion, respectively.