```matlab
% Abhilash Gudgunti
% 11th November, 2024
% ECE 202 Project 1: Phase 6
% Power series expansion of function of form Acos(wt)

clear % clear registers
clf % clear figures
format shortG

% Setting up Givens
A = 12; % Amplitude of the wave
w = 40; % Frequency of the wave

% Asking User for Inputs
% No. of non-zero terms in the truncated series
N = input("Enter the no. of non-zero terms for the truncated series: ");

ti = input("Enter the intial time (in ms): "); % Initial time (in ms)
tf = input("Enter the final time (in ms)" + ...
    "(should be greater than intitial time): "); % Final time (in ms)
% No. of Intervals between each time value
int = input("Enter the number of intervals between the time" + ...
    "(recommended >400): ");

% Defining arrays
n = (0:2:2*(N-1))'; % values of n for the first N non-zero terms. (up by 2)
an = A*(-1).^(n./2) .* w.^n ./ factorial(n); % Array of values of a_n

% Outputting a table of n and a_n
T = table(n, an, 'VariableNames',{'n','Coefficients (a_n)'})

% Setting up array of time
tms = linspace(ti, tf, int+1); % time between 0 - 200 ms
t = tms/1000; % time in seconds to compute in functions

% Initialize the function array f as zeros
f = zeros(size(t)); % initialize f as an array of zeros
p = zeros(size(n)); % initialize p as an array for plot objects

%FOR Loop to define funciton efficiently
hold on
for k = 1:N
    f = f + an(k)*t.^n(k); % adding each term to the series

    % Plotting each function
    if k == N
        % thicker line for the last function
        p(k) = plot(tms, f, 'LineWidth', 3);
    else
        p(k) = plot(tms, f, 'LineWidth', 1.5);
    end
end

```

```
% Computing average Deviation
Avg_dev = sum(abs(A*cos(w*t) - f))/(int+1)

% ====PLOTTING====

plot([ti,tf], [0,0], 'k', 'LineWidth', 1) % x-axis (not shown in legend)

% Setting legends for the figure
legend(p, "n = " + n, 'Location', 'bestoutside');

% Figure components
ax = gca; ax.FontSize = 16;
title(sprintf(['ECE 202 Project 1 Phase 6:\nApproximation of ' ...
    'f(x) = %gcos(%gt) \nfor %g non-zero terms\n ' ...
    'with an average deviation of %.2f from the function'], ...
    A, w , N, Avg_dev),"FontSize", 19);
xlabel('Time t (in ms)', 'FontSize', 17);
ylabel('f(t)', 'FontSize', 17);
ylim([-1.2*A , 1.2*A]);
xlim([ti, tf]);
ax.GridAlpha = 0.4; % making the grid darker
grid on
hold off

% ====CHECK====

% Check difference between new and old approach for the final function
if N == 6
    %Inefficiently represented function
    f1 = an(1) * t.^n(1); % The first non zero term
    f2 = f1 + an(2) * t.^n(2); % The second non zero term
    f3 = f2 + an(3) * t.^n(3); % The third non zero term
    f4 = f3 + an(4) * t.^n(4); % The fourth non zero term
    f5 = f4 + an(5) * t.^n(5); % The fifth non-zero term
    f6 = f5 + an(6) * t.^n(6); % The sixth non zero term
    check = max(abs(f - f6)) % should be zero for the final function
end

%Yes, The graph continues to look the same visually from phase 2 and
%nothing has been changed (from  Phase 3)

%(Phase 5) While computing the average deviation, It is seen that with an
%increase in the number of non-zero terms for the truncated series, the
%average deviation approaches 0. This is correct because, as we increase.
%the number of non-zero terms, we are increasing a better approximation for
%our function.

% ====Phase 6====

% a.) The number of non-zero terms that is close to the actual function and
% has a average magnitude of deviation of less than 0.05 is 11

% b.) Yes, the average magnitude of deviation does not change appreciably
```

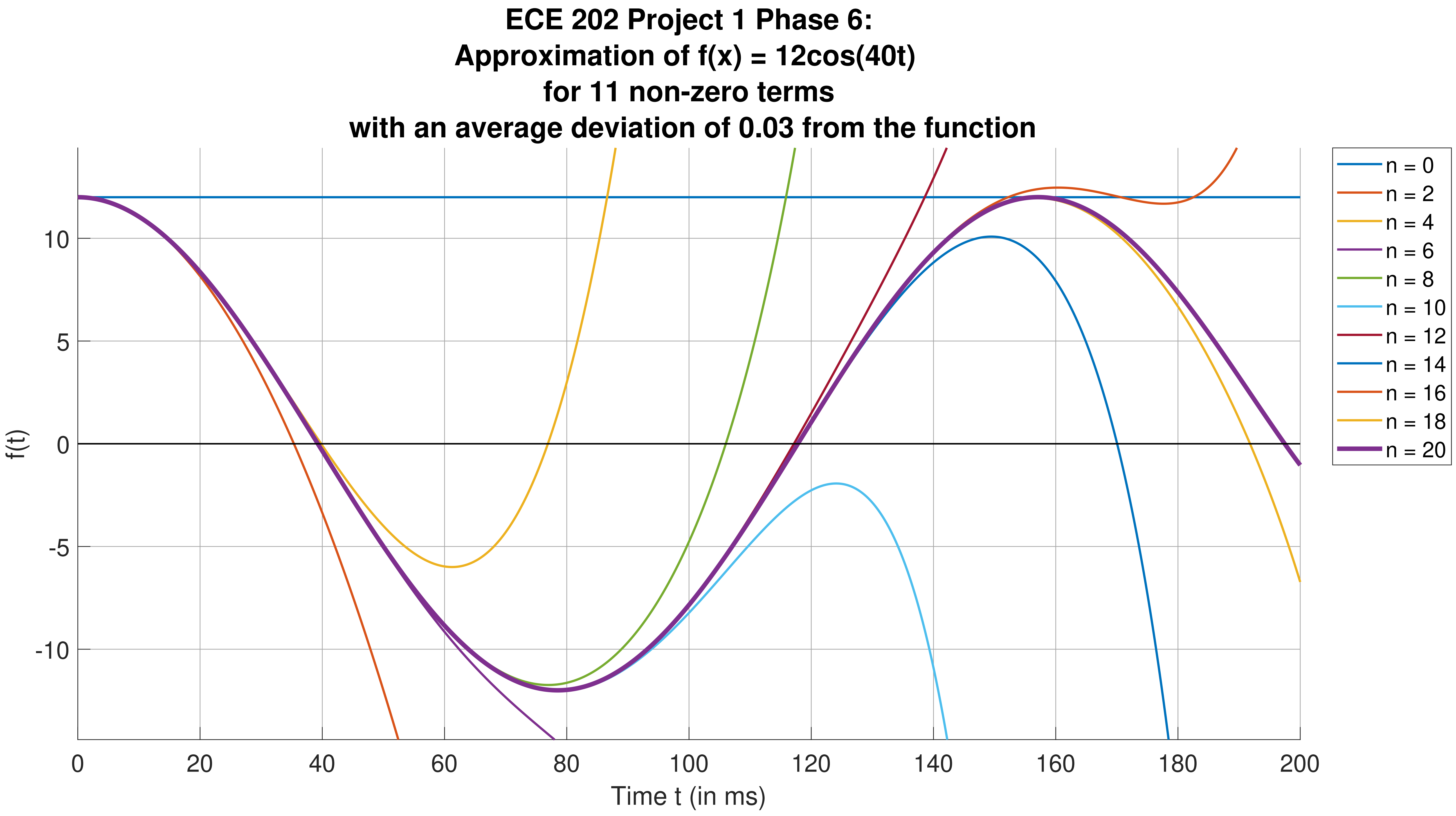
```
% when we double the number of intervals. (here 1000 to 2000). We can say
% that from looking the values themselves as well:
% 0.031228 for Run with 1000 intervals and 0.031067 for run with 2000
% intervals

% c.) I think the average magnitude of deviation is going to be the same,
% because the range from t0, (that is 0ms here) and to 200 and -200 is the
% same. taylor's approximation is affected by how far you go from this t0
% value and not by the number of intervals. Hence the average deviation
% shouldn't change by a lot.

% d.) The output value of the average magintude of deviation after running
% for -200-0 with 1000 intervals is 0.031584. This is about the same as
% the runs we did for a. and b. The reasoning is the same as explained in
% part c.

% e.) The average magnitude of deviation will still be the same if we move
% t0 to be 200ms instead of 0ms because the number of non-zero terms is
% what really impacts the average deviation value. What we aer doing here
% is just shifting it from 0 to 200ms. This shouldn't affect the magnitude
% of average deviation.

% f.) at 500ms, the value of the functions is a really big number. This is
% because the average deviation value is 1.0024*10^5. This means the value
% for the taylor function vs the actual function has a big difference. This
% also happens as the approximation should be needing more non-zero terms
% to better approximate the value. t0 is important here, because it
% basically establishes the starting point for the taylor function. The
% further we go from t0, the larger the average deviation we are gonna get.
% If for instance our t0 would have been 200ms, then our average deviation
% value would have been way less due as we are moving only 200ms front or
% back and the interval is same.

% g.) The minimum number of non-zero terms for the function to look like
% and have a magnitude of average deviation less than 0.05 is 22.
```

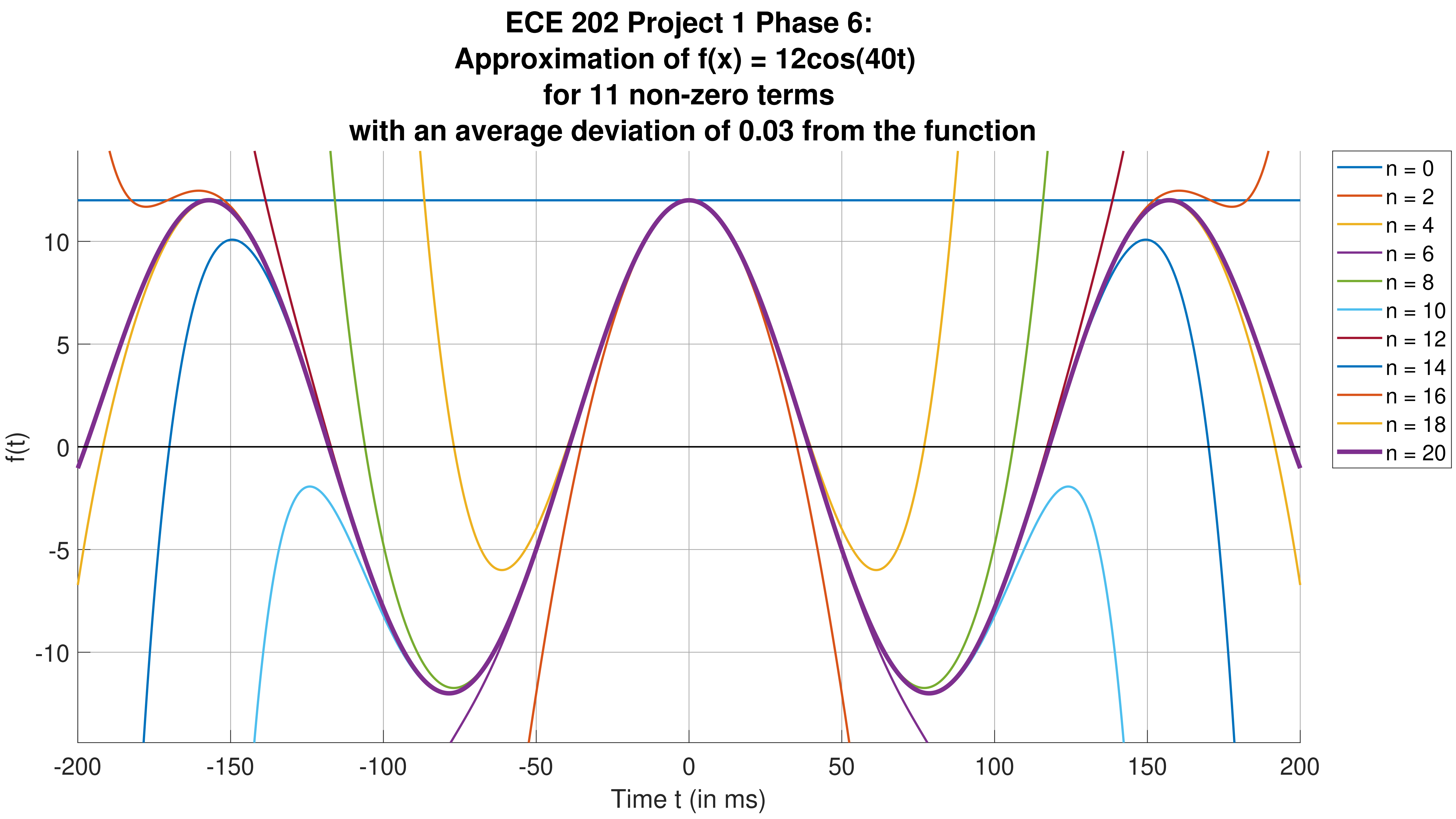
```
>> ECE202_P1_Phase6
Enter the no. of non-zero terms for the truncated series: 11
Enter the intial time (in ms): 0
Enter the final time (in ms)(should be greater than intitial time): 200
Enter the number of intervals between the time(recommended >400): 1000

T =

  11×2 table

    n     Coefficients (a_n)
    __    __________________

     0              12
     2           -9600
     4        1.28e+06
     6     -6.8267e+07
     8      1.9505e+09
    10     -3.4675e+10
    12       4.203e+11
    14      -3.695e+12
    16      2.4633e+13
    18      -1.288e+14
    20      5.4232e+14


Avg_dev =

     0.031228

>> ECE202_P1_Phase6
Enter the no. of non-zero terms for the truncated series: 11
Enter the intial time (in ms): 0
Enter the final time (in ms)(should be greater than intitial time): 200
Enter the number of intervals between the time(recommended >400): 2000

T =

  11×2 table

    n     Coefficients (a_n)
    __    __________________

     0              12
     2           -9600
     4        1.28e+06
     6     -6.8267e+07
     8      1.9505e+09
    10     -3.4675e+10
    12       4.203e+11
```

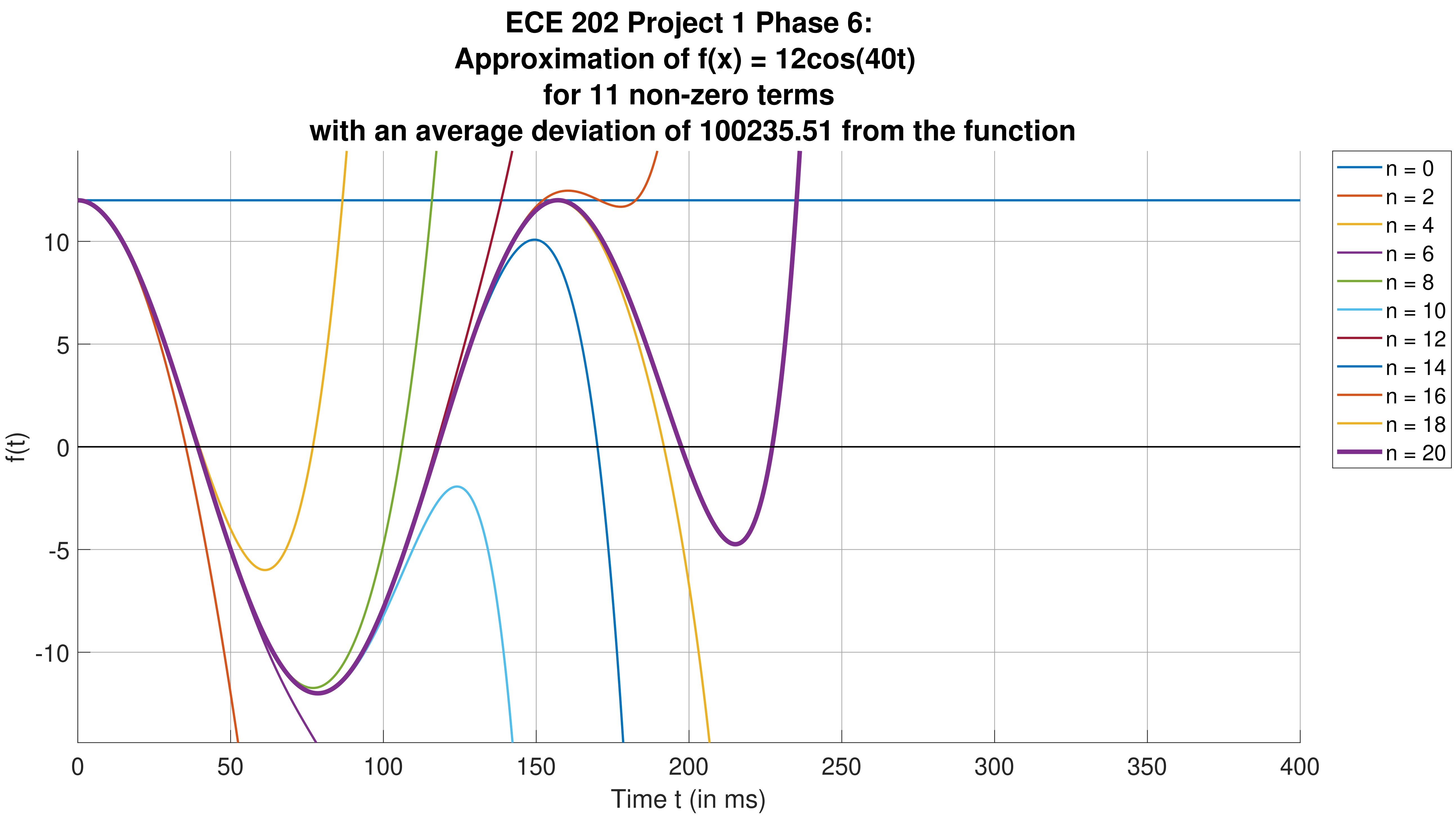
```
    14        -3.695e+12
    16        2.4633e+13
    18        -1.288e+14
    20        5.4232e+14


Avg_dev =

     0.031067

>> ECE202_P1_Phase6
Enter the no. of non-zero terms for the truncated series: 11
Enter the intial time (in ms): -200
Enter the final time (in ms)(should be greater than intitial time): 200
Enter the number of intervals between the time(recommended >400): 1000

T =

  11×2 table

    n     Coefficients (a_n)
    __    __________________

     0                12
     2             -9600
     4           1.28e+06
     6        -6.8267e+07
     8         1.9505e+09
    10        -3.4675e+10
    12          4.203e+11
    14         -3.695e+12
    16         2.4633e+13
    18         -1.288e+14
    20         5.4232e+14


Avg_dev =

     0.031584

>> ECE202_P1_Phase6
Enter the no. of non-zero terms for the truncated series: 11
Enter the intial time (in ms): 0
Enter the final time (in ms)(should be greater than intitial time): 400
Enter the number of intervals between the time(recommended >400): 1000

T =

  11×2 table
```

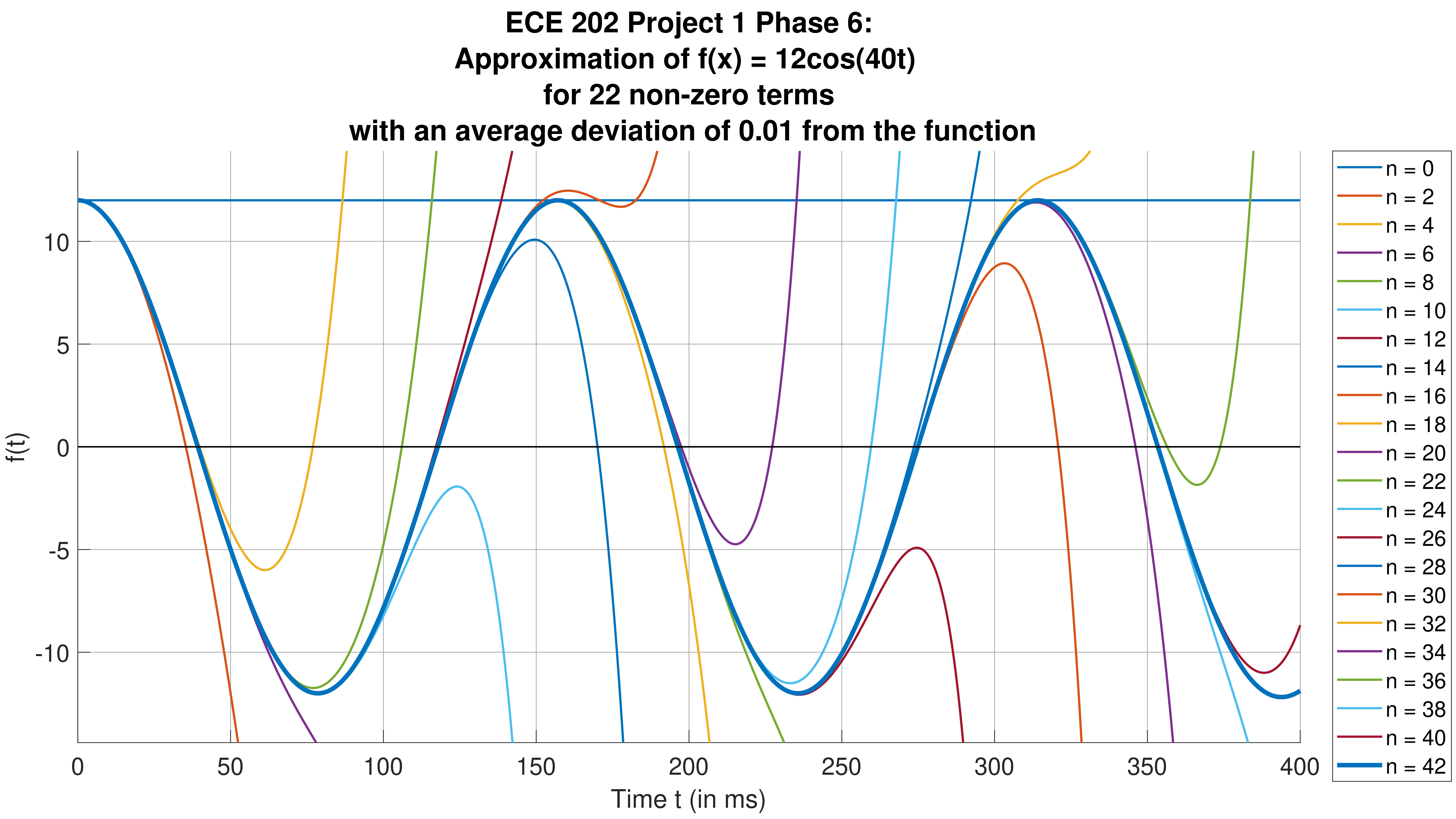
```
    n     Coefficients (a_n)
    __    __________________

     0                12
     2             -9600
     4          1.28e+06
     6       -6.8267e+07
     8        1.9505e+09
    10       -3.4675e+10
    12          4.203e+11
    14        -3.695e+12
    16        2.4633e+13
    18        -1.288e+14
    20        5.4232e+14


Avg_dev =

   1.0024e+05

>> ECE202_P1_Phase6
Enter the no. of non-zero terms for the truncated series: 22
Enter the intial time (in ms): 0
Enter the final time (in ms)(should be greater than intitial time): 400
Enter the number of intervals between the time(recommended >400): 1000

T =

  22×2 table

    n     Coefficients (a_n)
    __    __________________

     0                12
     2             -9600
     4          1.28e+06
     6       -6.8267e+07
     8        1.9505e+09

    :             :

    34       -1.1997e+17
    36        1.5234e+17
    38       -1.7336e+17
    40         1.778e+17
    42        -1.652e+17

    Display all 22 rows.
```

```
Avg_dev =

    0.0087679


T =

  22×2 table

    n     Coefficients (a_n)
    __    __________________

     0                12
     2             -9600
     4          1.28e+06
     6       -6.8267e+07
     8        1.9505e+09
    10       -3.4675e+10
    12         4.203e+11
    14        -3.695e+12
    16        2.4633e+13
    18        -1.288e+14
    20        5.4232e+14
    22       -1.8782e+15
    24         5.444e+15
    26       -1.3401e+16
    28        2.8361e+16
    30       -5.2158e+16
    32        8.4126e+16
    34       -1.1997e+17
    36        1.5234e+17
    38       -1.7336e+17
    40         1.778e+17
    42        -1.652e+17

>>
```


ECE 202 Project 1 Phase 6:
Approximation of f(x) = 12cos(40t)
for 11 non-zero terms
with an average deviation of 0.03 from the function
f(t)
Time t (in ms)
0
20
40
60
80
100
120
140
160
180
200
10
5
0
-5
-10
n = 0
n = 2
n = 4
n = 6
n = 8
n = 10
n = 12
n = 14
n = 16
n = 18
n = 20


ECE 202 Project 1 Phase 6:
Approximation of f(x) = 12cos(40t)
for 11 non-zero terms
with an average deviation of 0.03 from the function
f(t)
Time t (in ms)
10
5
0
-5
-10
-200
-150
-100
-50
0
50
100
150
200
n = 0
n = 2
n = 4
n = 6
n = 8
n = 10
n = 12
n = 14
n = 16
n = 18
n = 20


ECE 202 Project 1 Phase 6:
Approximation of f(x) = 12cos(40t)
for 11 non-zero terms
with an average deviation of 100235.51 from the function
f(t)
Time t (in ms)
0
50
100
150
200
250
300
350
400
10
5
0
-5
-10
n = 0
n = 2
n = 4
n = 6
n = 8
n = 10
n = 12
n = 14
n = 16
n = 18
n = 20


ECE 202 Project 1 Phase 6:
Approximation of f(x) = 12cos(40t)
for 22 non-zero terms
with an average deviation of 0.01 from the function
f(t)
Time t (in ms)
10
5
0
-5
-10
0
50
100
150
200
250
300
350
400
n = 0
n = 2
n = 4
n = 6
n = 8
n = 10
n = 12
n = 14
n = 16
n = 18
n = 20
n = 22
n = 24
n = 26
n = 28
n = 30
n = 32
n = 34
n = 36
n = 38
n = 40
n = 42