

UWTSD

MSc Motorsport Engineering

Task 1

ACHA7002

2504057

12/02/2025

Contents

1	Plotting and differentiation.....	1
2	Trapezoidal Method.....	3
3	Composite	3
4	Simpsons.....	4
5	Integration results tables.....	5
5.1	Composite convergence iterations	5
5.2	Simpsons Convergence iterations.....	9
6	Appendices.....	10
6.1	Your code script and functions.....	10

1 Plotting and differentiation

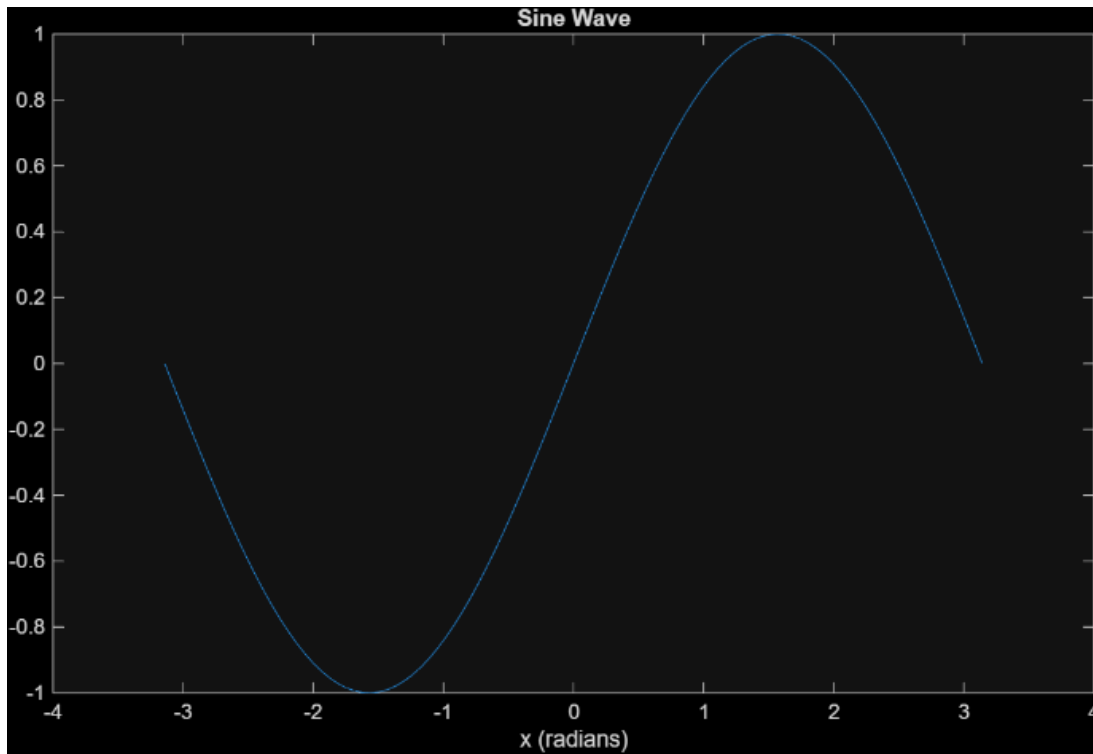


Figure 1 $y = \sin(x)$ plot

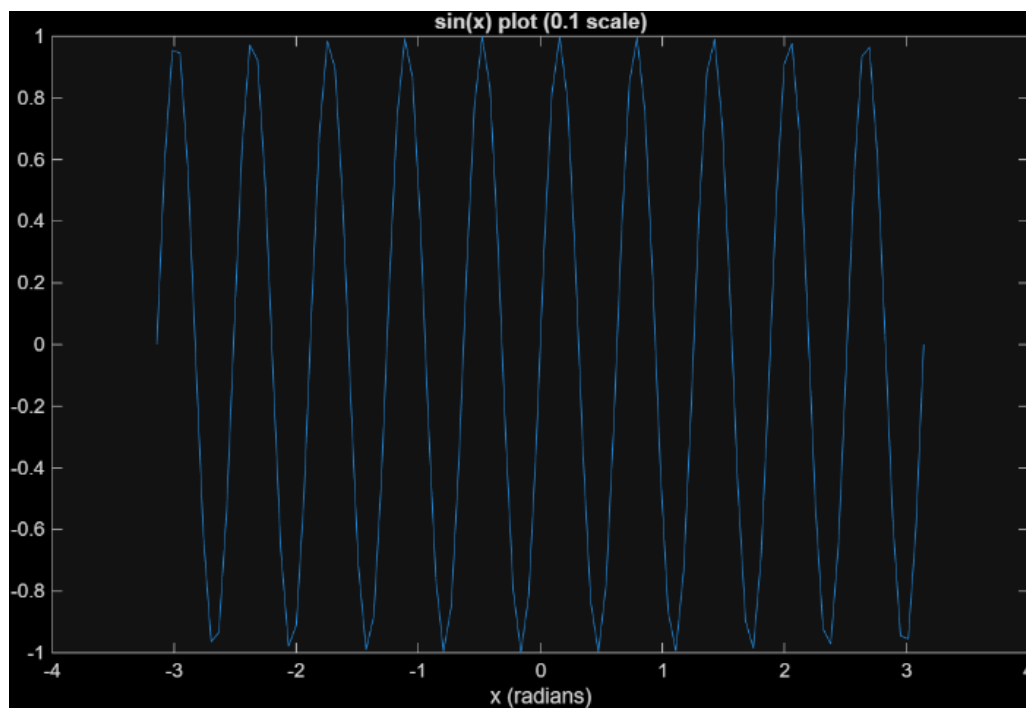


Figure 2 $y = \sin_scale(x, sf)$ plot (0.1 steps)

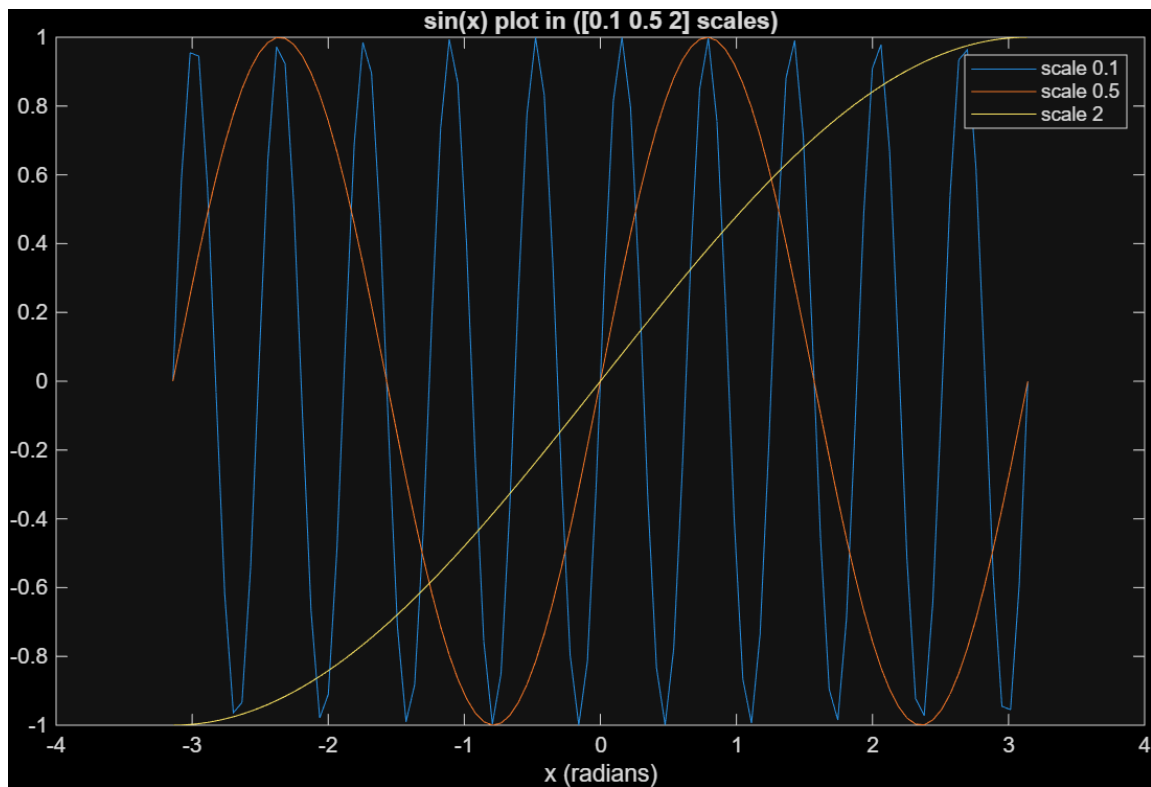


Figure 3 $y = \sin_scale(x, sf)$ plot ([0.1 0.5 2] steps)

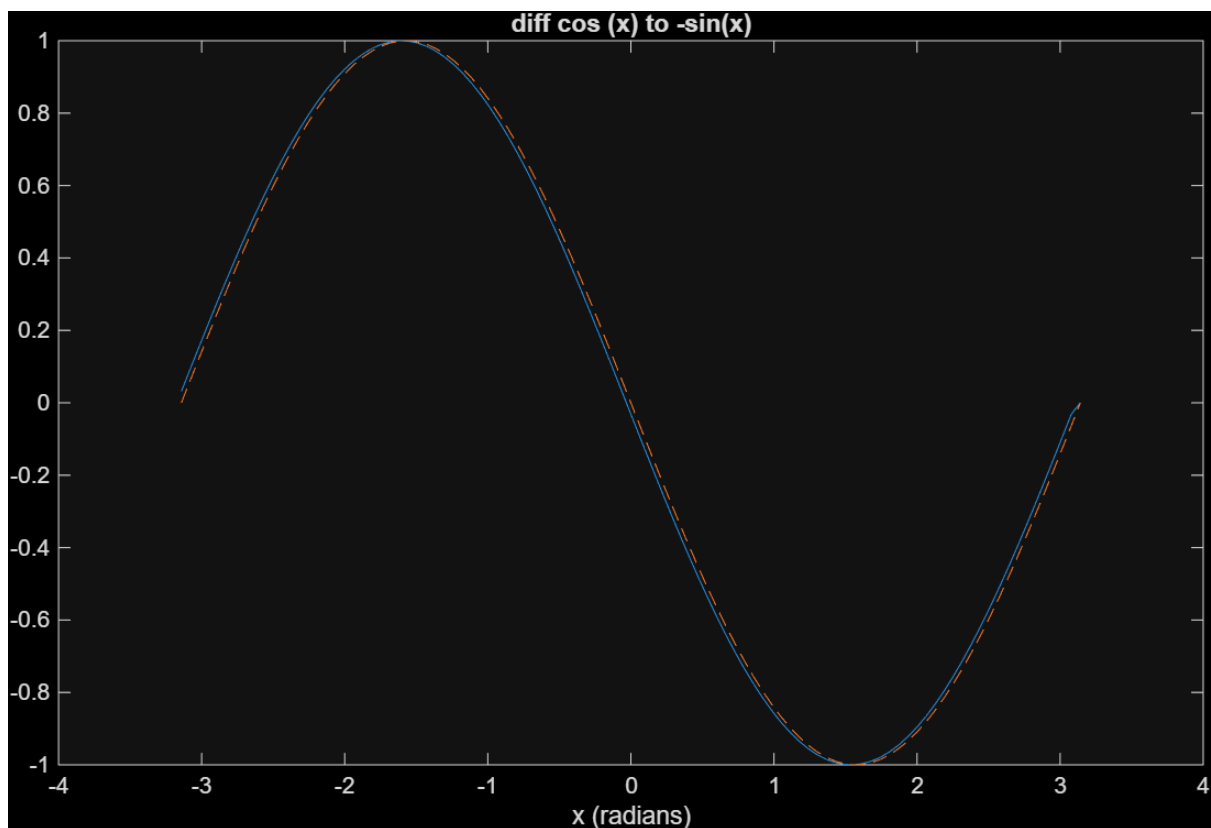


Figure 4 diff cos (x) to $-\sin(x)$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h_i}$$

Equation 1-1 numerical differentiation

2 Trapezoidal Method

$$I \approx \frac{1}{2}(f_a + f_b)(b - a)$$

Equation 2-1 trapezoidal method

A function was created to integrate 'x sin(x)' between 0 and pi/2 the code is in appendix

$$I = \int_0^{\pi/2} x \sin(x) dx ;$$

IntTrap (your code result)	Analytical (result)	percentage_error_Trap
1.2337	1	23.37

3 Composite

A function was created to integrate further functions in a discretised way from equation 3.1.

$$I \approx \frac{1}{2} \Delta x_n (f_a + 2 \sum_{i=1}^{n-1} f_i + f_b)$$

Equation 3-1 composite

Table 3-1 results for integration of $f(x) = 1/x$ using composite method a while loop was used to find when the result was within $1e-6$

	Analytical	Composite Trapezium			Converge 1e-6
'Result'	0.693147	0.695634	0.693172	0.693153	0.693148
'n iterations'	Na	5	50	100	250
'Percentage difference'	Na	0.358905%	0.003607%	0.000902%	0.000001%

4 Simpsons

A function was created to integrate further functions in a discretised separating odd and even crossing over in a quadratic manner way from equation 4.1

$$I \approx \frac{1}{3} \Delta x_n (f_a + 4 \sum_{i=1, \text{ odd only}}^{n-1} f_i + 2 \sum_{i=1, \text{ even only}}^{n-1} f_i + f_b)$$

Equation 4-1 Simpsons

Table 4-1 results for integration of $f(x) = 1/x$ using Simpson's method a while loop was used to find when the result was within $1e-6$ of the analytical

	Analytical	Simpsons Trapezium			Converge 1e-6
'Result'	0.693147	0.584126	0.693147	0.693147	0.693147
'n iterations'	NA	5	50	100	14
'Percentage difference'	NA	15.728290%	0.000001%	0.0000001%	0.000001%

5 Integration results tables

5.1 Composite convergence iterations

The difference is = 0.056853 for = 1 Number of intervals with a integrate value of = 0.750000
The difference is = 0.015186 for = 2 Number of intervals with a integrate value of = 0.708333
The difference is = 0.006853 for = 3 Number of intervals with a integrate value of = 0.700000
The difference is = 0.003877 for = 4 Number of intervals with a integrate value of = 0.697024
The difference is = 0.002488 for = 5 Number of intervals with a integrate value of = 0.695635
The difference is = 0.001730 for = 6 Number of intervals with a integrate value of = 0.694877
The difference is = 0.001272 for = 7 Number of intervals with a integrate value of = 0.694419
The difference is = 0.000975 for = 8 Number of intervals with a integrate value of = 0.694122
The difference is = 0.000770 for = 9 Number of intervals with a integrate value of = 0.693918
The difference is = 0.000624 for = 10 Number of intervals with a integrate value of = 0.693771
The difference is = 0.000516 for = 11 Number of intervals with a integrate value of = 0.693663
The difference is = 0.000434 for = 12 Number of intervals with a integrate value of = 0.693581
The difference is = 0.000370 for = 13 Number of intervals with a integrate value of = 0.693517
The difference is = 0.000319 for = 14 Number of intervals with a integrate value of = 0.693466
The difference is = 0.000278 for = 15 Number of intervals with a integrate value of = 0.693425
The difference is = 0.000244 for = 16 Number of intervals with a integrate value of = 0.693391
The difference is = 0.000216 for = 17 Number of intervals with a integrate value of = 0.693363
The difference is = 0.000193 for = 18 Number of intervals with a integrate value of = 0.693340
The difference is = 0.000173 for = 19 Number of intervals with a integrate value of = 0.693320
The difference is = 0.000156 for = 20 Number of intervals with a integrate value of = 0.693303
The difference is = 0.000142 for = 21 Number of intervals with a integrate value of = 0.693289
The difference is = 0.000129 for = 22 Number of intervals with a integrate value of = 0.693276
The difference is = 0.000118 for = 23 Number of intervals with a integrate value of = 0.693265
The difference is = 0.000108 for = 24 Number of intervals with a integrate value of = 0.693256
The difference is = 0.000100 for = 25 Number of intervals with a integrate value of = 0.693247
The difference is = 0.000092 for = 26 Number of intervals with a integrate value of = 0.693240
The difference is = 0.000086 for = 27 Number of intervals with a integrate value of = 0.693233
The difference is = 0.000080 for = 28 Number of intervals with a integrate value of = 0.693227
The difference is = 0.000074 for = 29 Number of intervals with a integrate value of = 0.693221
The difference is = 0.000069 for = 30 Number of intervals with a integrate value of = 0.693217
The difference is = 0.000065 for = 31 Number of intervals with a integrate value of = 0.693212
The difference is = 0.000061 for = 32 Number of intervals with a integrate value of = 0.693208
The difference is = 0.000057 for = 33 Number of intervals with a integrate value of = 0.693205
The difference is = 0.000054 for = 34 Number of intervals with a integrate value of = 0.693201
The difference is = 0.000051 for = 35 Number of intervals with a integrate value of = 0.693198
The difference is = 0.000048 for = 36 Number of intervals with a integrate value of = 0.693195
The difference is = 0.000046 for = 37 Number of intervals with a integrate value of = 0.693193
The difference is = 0.000043 for = 38 Number of intervals with a integrate value of = 0.693190
The difference is = 0.000041 for = 39 Number of intervals with a integrate value of = 0.693188
The difference is = 0.000039 for = 40 Number of intervals with a integrate value of = 0.693186
The difference is = 0.000037 for = 41 Number of intervals with a integrate value of = 0.693184
The difference is = 0.000035 for = 42 Number of intervals with a integrate value of = 0.693183
The difference is = 0.000034 for = 43 Number of intervals with a integrate value of = 0.693181
The difference is = 0.000032 for = 44 Number of intervals with a integrate value of = 0.693179
The difference is = 0.000031 for = 45 Number of intervals with a integrate value of = 0.693178
The difference is = 0.000030 for = 46 Number of intervals with a integrate value of = 0.693177
The difference is = 0.000028 for = 47 Number of intervals with a integrate value of = 0.693175
The difference is = 0.000027 for = 48 Number of intervals with a integrate value of = 0.693174
The difference is = 0.000026 for = 49 Number of intervals with a integrate value of = 0.693173
The difference is = 0.000025 for = 50 Number of intervals with a integrate value of = 0.693172

[illegible]

[illegible]

[illegible]

[illegible]

5.2 Simpsons Convergence iterations

The difference is = 0.193147 for = 1 Number of intervals with a integrate value of = 0.693148
The difference is = 0.001297 for = 2 Number of intervals with a integrate value of = 0.693148
The difference is = 0.193147 for = 3 Number of intervals with a integrate value of = 0.693148
The difference is = 0.000107 for = 4 Number of intervals with a integrate value of = 0.693148
The difference is = 0.109020 for = 5 Number of intervals with a integrate value of = 0.693148
The difference is = 0.000023 for = 6 Number of intervals with a integrate value of = 0.693148
The difference is = 0.075927 for = 7 Number of intervals with a integrate value of = 0.693148
The difference is = 0.000007 for = 8 Number of intervals with a integrate value of = 0.693148
The difference is = 0.058243 for = 9 Number of intervals with a integrate value of = 0.693148
The difference is = 0.000003 for = 10 Number of intervals with a integrate value of = 0.693148

The difference is = 0.047239 for = 11 Number of intervals with a integrate value of = 0.693148
 The difference is = 0.000001 for = 12 Number of intervals with a integrate value of = 0.693148
 The difference is = 0.039732 for = 13 Number of intervals with a integrate value of = 0.693148
 The difference is = 0.000001 for = 14 Number of intervals with a integrate value of = 0.693148
Number of intervals for the desire accuracy is = 14 and difference is = 0.000001 for an integrate value of =0.693148

6 Appendices

6.1 Your code script and functions

```
% Using a script file (I)
% b) First clear all the spaces
clc, clear, close all

format short g %formats the outputted numerical data

% d) Define variables and inputs
N = 100;
z = linspace(-pi,pi,N);
sf = 0.1;
%% e)
figure (1)
y1 = sin(z);
plot(z,y1)
title('Sine Wave');
xlabel('x (radians)');

%% Defining a function (II)
% Exercise e)
y2 = sin_scale(z,sf);
figure(2),
plot(z,y2)
title('sin(x) plot (0.1 scale)');
xlabel('x (radians)');

%% Exercise f)
sf_f=[0.1 0.5 2]; % Scale factor values in an array named sf for the exercise
"f" in the task 1
N_sf = numel(sf_f); % Number of elements within sf_f (ML intrinsic function)
sin_array = zeros(N,N_sf); %Initialise an array of zeros to store data.

for ndx = 1: N_sf
sin_array(:,ndx) = sin_scale(z,sf_f(ndx));
end

figure(3)
plot(z,sin_array)
title('sin(x) plot in ([0.1 0.5 2] scales)')
xlabel('x (radians)');
legend('scale 0.1', 'scale 0.5', 'scale 2')

%% Numerical Differentiation
```

```

x_col = z'; % a) we define our row vector x as a column vector
y3 = cos(x_col); % b) we calculate our cos(x)
f_diff = MyDiff(y3,x_col); %d) estimate cos derivative with our function using
equation 1

% e) Plotting f_diff against -sin function
figure(4)
plot(z,f_diff)
hold on
plot (z, -sin(x_col), '--')
title('diff cos (x) to -sin(x)')
xlabel('x (radians)');
legend('f_diff','-sin(x_col)')
hold off

%% Numerical Integration

% 1. Trapezoidal Method

% a)
a = 0; % lower limit
b = pi/2; %upper limit
I_trap = trap1('I',a,b);

% b) Analytical result of the integration x*sin(x) between [0 pi/2)
syms z
expr = z.*sin(z);
True_value = double(int(expr,[0 pi/2]));
fprintf('True_value = %f \n',True_value);

% Calculate the error for the trapezoidal method
error_trap = (abs(True_value - I_trap)/True_value)*100;
fprintf('Error = %f \n',error_trap);

%% 2. Composite Trapezium Rule
% Exercise b) Calculate the integral for I function
I_CompTrap = CompTrap('I', a,b,N);

% calculate the error
error_trap2 = (abs(True_value - I_CompTrap)/True_value)*100;
fprintf('Error = %f \n',error_trap2);

%% Exercise c)
% Create the different intervals to evaluate
x5 = linspace(1,2,5);
x50 = linspace(1,2,50);
x100 = linspace(1,2,100);

F2_ComTrap_5 = CompTrap('F2',1,2,5);
F2_ComTrap_50 = CompTrap('F2',1,2,50);
F2_ComTrap_100 = CompTrap('F2',1,2,100);
% d) calculate the analitical solution and error for N=5
syms z
expr = 1./z;
True_value2 = double(int(expr,[1 2]));
fprintf('True_value2 = %f \n',True_value2);
error_ComTrap5 = (abs(True_value2 - F2_ComTrap_5)/True_value2)*100;
fprintf('Error = %f \n',error_ComTrap5);

```

```

% d) calculate the error for N=50
error_ComTrap50 = (abs(True_value2 - F2_ComTrap_50)/True_value2)*100;
fprintf('Error = %f \n',error_ComTrap50);

% d) calculate the error for N=100
error_ComTrap100 = (abs(True_value2 - F2_ComTrap_100)/True_value2)*100;
fprintf('Error = %f \n',error_ComTrap100);

%% e) Determine the number of sub-intervals needed to achieve an accuracy to 6
decimal places.
tol = 1e-6;
N_interval = 1;
diference = 50; % Initialize error to a large value for the while loop
%Analytic_value = True_value2

while diference > tol

    F2_ComTrap = CompTrap('F2',1,2,N_interval);
    diference = abs(True_value2 - F2_ComTrap);
    fprintf('The diference is = %f for = %d Number of intervals with a integrate
value of = %f \n',diference, N_interval, F2_ComTrap )
    N_interval = N_interval + 1;

end
N_interval = N_interval-1;
fprintf('Number of intervals for the desire accuracy is = %d and diference is =
%f for an integrate value of =%f \n',N_interval, diference, F2_ComTrap)

%% 3. Simpson's Rule

% b) Calculate the value of the integral for F2 with SimsRule between [1,2]
% For 5 intervals
F2_SimsRule_5 = SimsRule('F2', 1, 2, 5);

% For 50 intervals
F2_SimsRule_50 = SimsRule('F2', 1, 2, 50);

% For 100 intervals
F2_SimsRule_100 = SimsRule('F2', 1, 2, 100);

%% c) Calculate the error for each case
% 5 intervals
error_SimsRule5 = (abs(True_value2 - F2_SimsRule_5)/True_value2)*100;
fprintf('Error = %f \n',error_SimsRule5);

% 50 intervals
error_SimsRule50 = (abs(True_value2 - F2_SimsRule_50)/True_value2)*100;
fprintf('Error = %f \n',error_SimsRule50);

% 100 intervals
error_SimsRule100 = (abs(True_value2 - F2_SimsRule_100)/True_value2)*100;
fprintf('Error = %f \n',error_SimsRule100);

%% d) Determine the number of sub-intervals needed to achieve an accuracy to 6
decimal places for Simpson's Rule.
tol = 1e-6;
N_interval = 1;
diference = 50; % Initialize error to a large value for the while loop

```

```

%Analytic_value = True_value2;

while difference > tol

    F2_SimsRule = SimsRule('F2',1,2,N_interval);
    difference = abs(True_value2 - F2_SimsRule);
    fprintf ('The difference is = %f for = %d Number of intervals with a integrate
value of = %f \n',difference, N_interval, F2_ComTrap )
    N_interval = N_interval + 1;

end
N_interval = N_interval - 1;
fprintf ('Number of intervals for the desire accuracy is = %d and difference is =
%f for an integrate value of =%f \n',N_interval, difference, F2_SimsRule)

-----

% Exercise b), c), d).
function y = sin_scale (x,sf)
% Input: scalar or vector x, sf scale factor
% Output: scalar or vector containing corresponding sine values.

% y = sin(x); ML intrinsic sine function
y = sin(x./sf);

% The '.' before an operator implies a vector/array operation

end

-----

% Exercise c)
function dy_dx = MyDiff(y, x)
%Inputs:
% x --> vector with all the point in the interval
% y --> vector with all the x points evaluated in the interval

dy_dx= zeros(length(y),1);
for loop = 1: length(y)-1
    h = x(loop+1) - x(loop); % Calculate the step size for differentiation
    dy_dx(loop) = (y(loop+1) - y(loop)) / h;

end

-----

% Exercise a) Write function  $I = x \sin(x)$ 
function y = I(x)

y = x.*sin(x);

end

-----

```

```

function int=trap1(f,a,b)
%Input:
% f --> is the integrand input as a string 'f'
% a --> lower limit of integration
% b --> upper limit of integration
%Output:
% int --> is the trapezoidal rule sum.
if a>b
error('b must be greater than a')
end
h=(b-a);
% Equation 3
int=(h*(feval(f,a)+feval(f,b)))/2;

-----

% Exercise a), build our function to resolve equation 4
function Int = CompTrap(fun, a, b, n)
% Inputs:
% fun --> Function to evaluate
% a --> lower limit
% b --> upper limit
% n --> number of intervals
%Output:
% int --> is the Integration.

h = (b-a)/n; % delta Xn
x = linspace(a,b,n+1);
y_values= feval(fun,x);

% Equation 4
Int = 0.5*h*(y_values(1) + y_values(end) + 2* sum(y_values(2:end-1)));

end

-----

% c) create de function f(x) = 1/x
function y = F2(x)

y = 1./x;

end

-----

% a) section, build our function to resolve equation 5
function Int = SimsRule(fun, a, b, n)
% Inputs:
% fun --> Function to evaluate
% a --> lower limit
% b --> upper limit
% n --> number of intervals
%Output:

```

```
% int --> is the Integration.

h = (b-a)/n; % delta Xn
x = linspace(a,b,n+1);
y_values= feval(fun,x);

% Equation 5
Int = (1/3).*h.*((y_values(1) + y_values(end)) + 4.*sum(y_values(2:2:end-1)) +
2.*sum(y_values(3:2:end-2))));

end
```