

Task 1

please convert this to pdf for submission

Table of Contents

(i) (e) $y = \sin(x)$ plot [1].....	1
(ii) (e) $y = \sin_scale(x,sf)$ plot [1].....	1
(ii) (e) $y = \sin$ array plot [3].....	1
Numerical Differentiation $y=\cos(x)$ plot [3].....	2
Numerical Intergration (a) $\int \sin(x) dx$ [3].....	3
Numerical Intergration (b) compare [3].....	3
Comp Trap (c) $1/x$ for 5,50,100 [4].....	4
Comp Trap (d) compare [1].....	4
Comp Trap (e) to tolerance [3].....	5
Simpsons (b) $1/x$ for 5,50,100 [4].....	5
Simpsons (c) compare [1].....	5
Simpsons (d) to tolerance [3].....	5

(i) (e) $y = \sin(x)$ plot [1]

```
clear; clc; close all;
format short g;

N = 100;
x = linspace(-pi, pi, N)';
x = (-pi:2*pi/(N-1):pi)';

y = sin(x);
figure(1);
plot(x, y, 'b', 'LineWidth', 1.5);
xlabel('x-axis');
ylabel('y = sin(x)');
title('Original Sine Function');
grid on;
```

(ii) (e) $y = \sin_scale(x,sf)$ plot [1]

```
sf = 0.1;
y = sin_scale(x, sf);

figure;
plot(x, y, 'r', 'LineWidth', 1.5);
xlabel('x-axis');
ylabel('Scaled Sin(x)');
title(['Sine Function with Scale Factor = ', num2str(sf)]);
grid on;
```

(ii) (e) $y = \sin$ array plot [3]

```
sf = [0.1, 0.5, 2];
```

```

N_sf = numel(sf);
sin_array = zeros(N, N_sf);

for ndx = 1:N_sf
    sin_array(:, ndx) = sin_scale(x, sf(ndx));
end

figure;
plot(x, sin_array, 'LineWidth', 1.5);
xlabel('x-axis');
ylabel('Scaled Sin(x)');
title(['A Sine Function; Limits = [-\pi, \pi]; Scale Factors = ', num2str(sf)]);
legend(arrayfun(@(s) ['sf = ', num2str(s)], sf, 'UniformOutput', false),
'Location', 'Best');
grid on;

```

Numerical Differentiation $y=\cos(x)$ plot [3]

```

clear,clc,close all
% Define the number of points
N = 100;

% Create the vector x using linspace
x = linspace(-pi, pi, N)';

% Calculate y = cos(x)
y = cos(x);

% Estimate the derivative using FDT (forward difference)
dy_dx = zeros(N, 1); % Preallocate for speed

for i = 1:N-1
    dx = x(i+1) - x(i);
    dy = y(i+1) - y(i);
    dy_dx(i) = dy/dx;
end

% Handle the last point by copying the second last derivative
dy_dx(N) = dy_dx(N-1);

% Plot the estimated derivative vs actual derivative
figure;
plot(x, dy_dx, 'b', 'LineWidth', 1.5); % Estimated derivative
hold on;
plot(x, cos(x), 'r--', 'LineWidth', 1.5); % Actual derivative of sin(x) is cos(x)

% Add title, labels, and legend
title('Derivative of sin(x) using Forward Difference Method');
xlabel('x-axis');
ylabel('dy/dx');

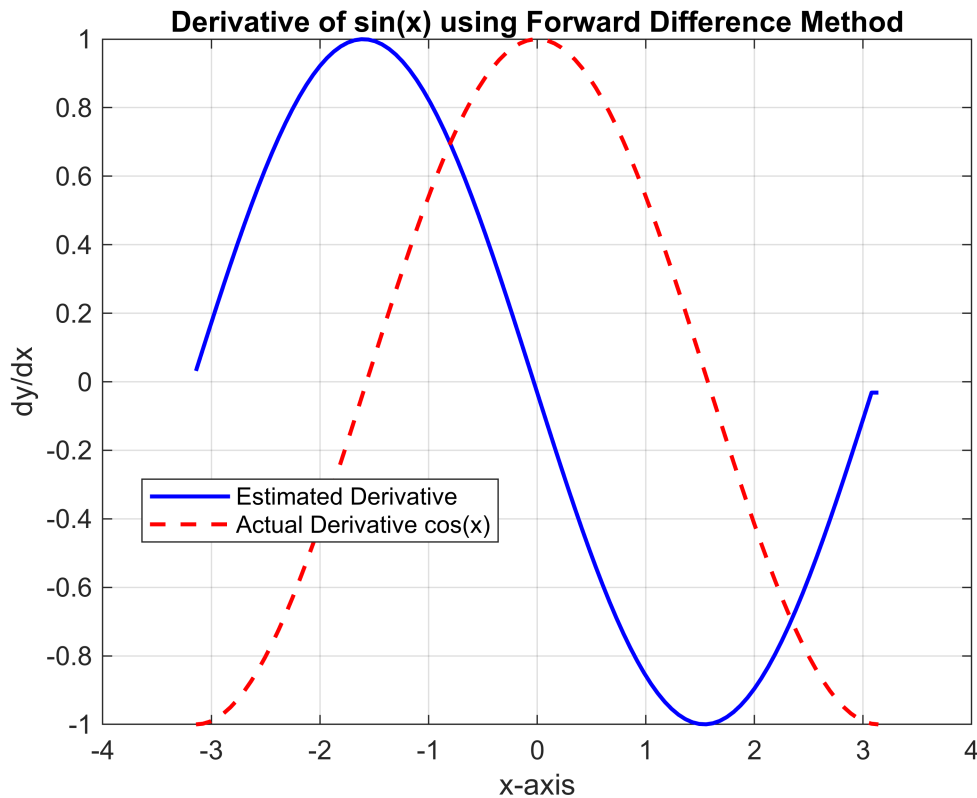
```

```

legend('Estimated Derivative', 'Actual Derivative cos(x)', 'Location', 'Best');

grid on;
hold off;

```



Numerical Intergration (a) $\diamond = \int \diamond \sin(\diamond) \diamond x$ [3]

```

% Define the symbolic variable
syms x

% Define the function to integrate
f = x * sin(x);

% Compute the definite integral
I = double(int(f, 0, pi/2));

sinx = sinx(x);
a = 0;
b = pi/2;
T1 = trap1('sinx',a,b);

```

Numerical Intergration (b) compare [3]

```

PE = (T1-I/I)*100;

```

```
% Display the result
table1 = table(["Analytical Integration";"Trapez Method";"Percentage";],
[I;T1;PE], 'VariableNames',{ 'Description', 'Value'});
disp(table1);
```

Description	Value
"Analytical Integration"	1
"Trapez Method"	1.2337
"Percentage"	23.37

Comp Trap (c) 1/x for 5,50,100 [4]

```
f = @(x)x.*sin(x);
comptrap = composite_trapezium_rule(f,a,b,100);

PE2 = (comptrap-I/I)*100;
table2 = table(["Analytical Integration";"comptrap";"Percentage";],
[I;comptrap;PE2], 'VariableNames',{ 'Description', 'Value'});
disp(table2);
```

Description	Value
"Analytical Integration"	1
"comptrap"	1
"Percentage"	0.0020562

Comp Trap (d) compare [1]

```
f1=@(x)1/x;
a1 = 1;
b2 = 2;
comptrap5 = composite_trapezium_rule(f1,a1,b2,5);
comptrap50 = composite_trapezium_rule(f1,a1,b2,50);
comptrap100 = composite_trapezium_rule(f1,a1,b2,100);

PE5 = (comptrap5-log(2)/log(2))*100;
PE50 = (comptrap50-log(2)/log(2))*100;
PE100 = (comptrap100-log(2)/log(2))*100;

table3 = table([5;50;100],
[comptrap5;comptrap50;comptrap100],repmat(log(2),3,1), 'VariableNames',{ 'Number of
Intervals', 'CompTrap', 'Analytical Values'});
disp(table3);
```

Number of Intervals	CompTrap	Analytical Values
5	0.69563	0.69315
50	0.69317	0.69315
100	0.69315	0.69315

Comp Trap (e) to tolerance [3]

```
accuracycomp =  
Accuracycheck(log(2),1e-6,@composite_trapezium_rule,"comptrap",f1,a1,b2);
```

Using comptrap:

Number of subintervals needed to match analytical value to 1.000000e-06 decimal places: 250

Analytical Value: 0.6931471806

Simpsons (b) 1/x for 5,50,100 [4]

```
simrule5 = simpsons_rule(f1,a1,b2,5);  
simrule50 = simpsons_rule(f1,a1,b2,50);  
simrule100 = simpsons_rule(f1,a1,b2,100);
```

Simpsons (c) compare [1]

```
table4 = table([5;50;100],  
[simrule5;simrule50;simrule100],repmat(log(2),3,1),'VariableNames',{'Number of  
Intervals','Simrule','Analytical Values'});  
disp(table4);
```

Number of Intervals	Simrule	Analytical Values
5	0.6582	0.69315
50	0.69315	0.69315
100	0.69315	0.69315

Simpsons (d) to tolerance [3]

```
accuracysim = Accuracycheck(log(2),1e-6,@simpsons_rule,"simrule",f1,a1,b2);
```

Using simrule:

Number of subintervals needed to match analytical value to 1.000000e-06 decimal places: 14

Analytical Value: 0.6931471806