

UWTSD

MSc

Task 1

Module code: ACHA7002

Anthony Yin Wai Hau

Student Number:1550559

5-12-2025

Contents

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

1 Plotting and differentiation

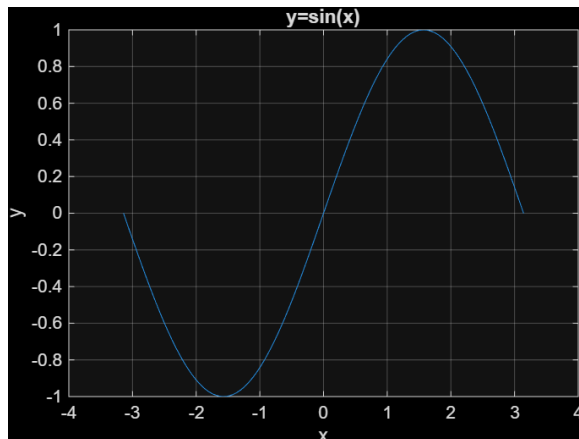


Figure 1 $y = \sin(x)$ plot

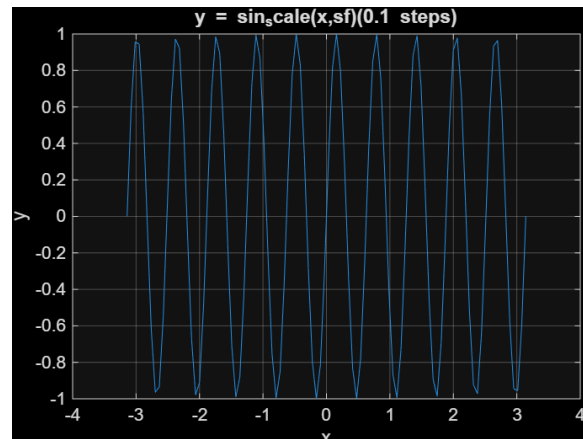


Figure 2 $y = \sin_scale(x, sf)$ plot (0.1 steps)

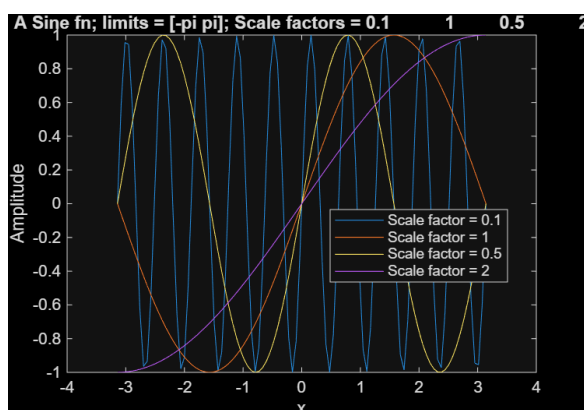


Figure 3 $y = \sin_scale(x, sf)$ plot ([0.1 0.5 2] steps)

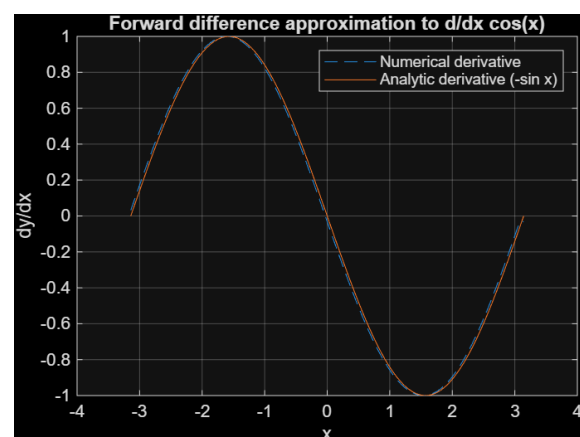


Figure 4 diff cos (x) to $-\sin(x)$

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

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
1.233701	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'	0.69315	0.69563	0.69317	0.69315	354
'n iterations'	Na	5	50	100	
'percentage difference'	Na	0.35891	0.0036066	0.00090167	

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'	0.69315	0.69317	0.69315	0.69315	14
'n iterations'	NA	5	50	100	
'percentage difference'	NA	0.0032623	7.2063e-07	4.5073e-08	

5 Integration results tables

5.1 Composite convergence iterations

5.2 Simpsons Convergence iterations

6 Appendices

6.1 Your code script and functions

6.1.1 Lab1_sin

6.1.2 F

6.1.3 Trap1

6.1.4 Sin_scale

6.1.5 my_funct

6.1.6 compTrap

```
function I = compTrap(a,b,n)
% Composite trapezium rule for integral of my_funct(x) from a to b
% n - number of sub-intervals
% my_funct(x) is defined in a separate file f.m

h = (b - a)/n;      % step size Δx
sum_mid = 0;        % accumulate interior values Σ f_i, i = 1..n-1

for i = 1:n-1
    x_i = a + i*h;
    sum_mid = sum_mid + my_funct(x_i);
end

I = 0.5*h*( my_funct
(a) + 2*sum_mid + my_funct
(b) );
end
```

6.1.7 fdt_forward

```
function dy = fdt_forward(x, y)
%FDT_FORWARD Forward difference derivative of y(x) using a loop.

% assume uniform spacing
h = x(2) - x(1);

% preallocate output
dy = zeros(size(y));
```

```
% forward-difference loop
for i = 1:length(y)-1
    dy(i) = (y(i+1) - y(i)) / h;
end

% for the final point, copy the previous derivative
dy(end) = dy(end-1);

end
```

6.1.8 SimpsonRule

```
function I = SimpsonRule(a,b,n)
% SimpsonRule composite Simpson's rule on [a,b] with n sub-intervals
% n MUST be even

if mod(n,2) ~= 0
    error('n must be even for Simpson''s rule');
end

h = (b-a)/n;           % step size
x = a:h:b;             % vector of x values
fx = my_funct(x);      % *** no @(x) here, just call the function below

I = h/3 * ( fx(1) + fx(end) + ...
            4*sum(fx(2:2:end-1)) + ...
            2*sum(fx(3:2:end-2)) );

end
```