



prifysgol cymru trinity saint devid

University of Wales Trinity St David

**Student name: Anandu Unnikrishanan Nair**

**Student ID number : 2411630**

**Programme : MSc – Automotive Engineering**

**Module : Alternative fuels and Combustion Modelling**

**Module code: ACHA7002**

## Contents

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

## 1 Plotting and differentiation

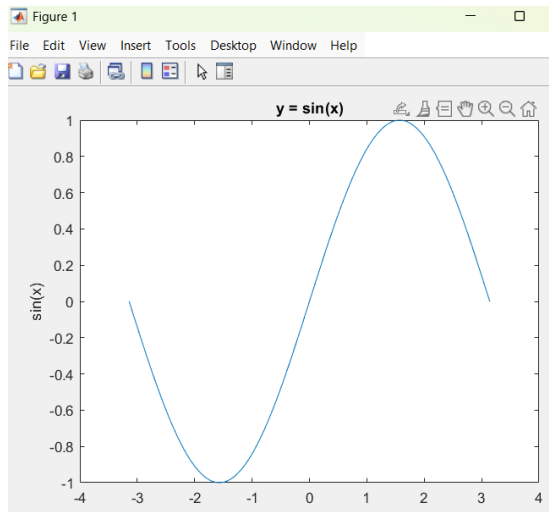


Figure 1  $y = \sin(x)$  plot

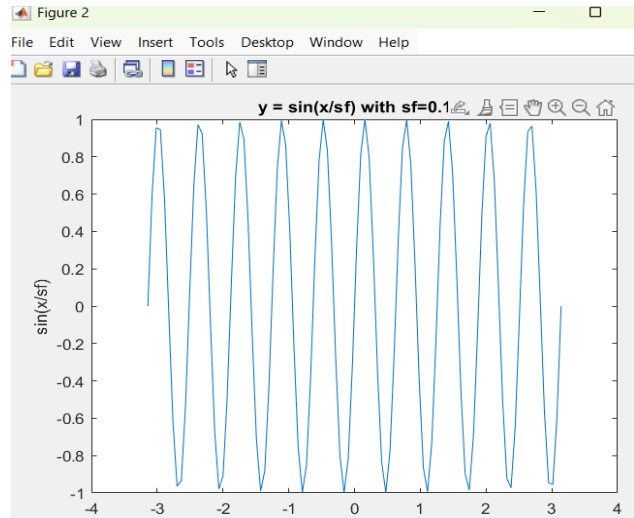


Figure 2  $y = \sin\_scale(x, sf)$  plot (0.1 steps)

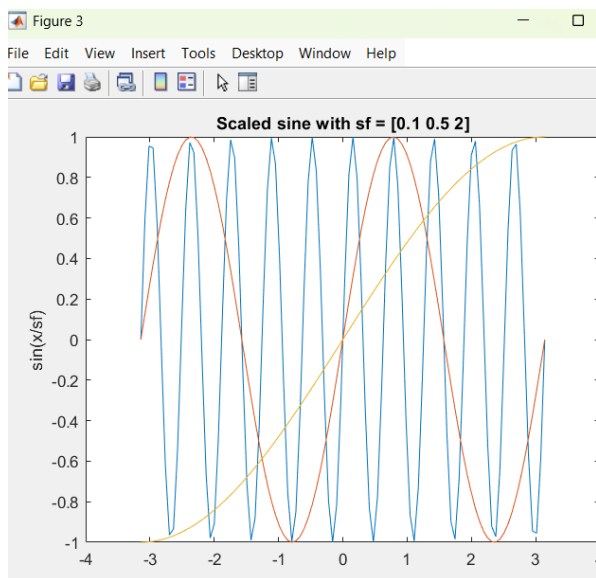


Figure 3  $y = \sin\_scale(x, sf)$  plot ([0.1 0.5 2] steps)

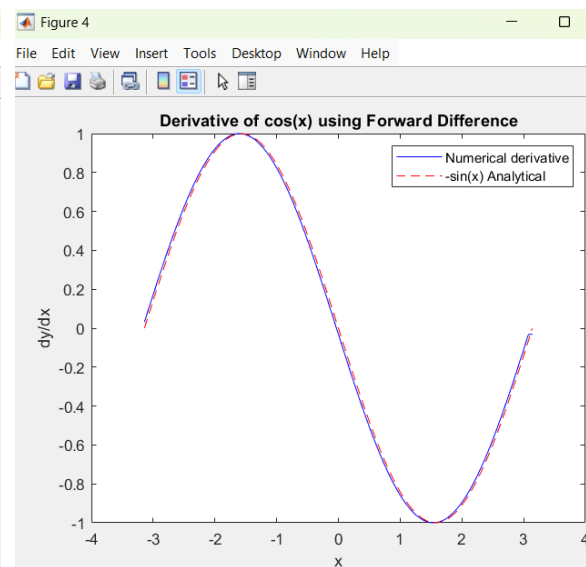


Figure 4 diff cos (x) to  $-\sin(x)$

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

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                   | 23.3701%              |

### Command Window

```
>> lab1_sin  
Trapezoidal result = 1.2337  
Percentage error = 23.3701%
```

### 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<br>1e-6 |
|-------------------------|------------|---------------------|------------|-------------|------------------|
| 'Result'                | 0.693147   | 0.69563             | 0.69317    | 0.69315     | 0.69315          |
| 'n iterations'          | Na         | 5                   | 50         | 100         | 250              |
| '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'                | 0.693147   | 0.69317            | 0.69315     | 0.69315     | 0.69315       |
| 'n iterations'          | NA         | 5                  | 50          | 100         | 14            |
| 'percentage difference' | NA         | 0.0032623%         | 7.2063e-07% | 4.5073e-08% | 8.0332e-07    |

Integration results tables

## 4.1 Composite convergence iterations

### Command Window

```
>> lab1_sin
Trapezoidal result = 1.2337
Percentage error = 23.3701%
Composite trap result (x*sin(x), n=100) = 1
Composite % error = 0.0020562%
Composite trapezoid results for f(x)=1/x:
    0.69563    0.69317    0.69315

Percentage differences:
    0.35891    0.0036066    0.00090167

Composite trapezoid converged at n = 250
Result = 0.69315
Accuracy = 1e-06
```

## 4.2 Simpsons Convergence iterations

### Command Window

```
Percentage differences:
    0.35891    0.0036066    0.00090167

Composite trapezoid converged at n = 250
Result = 0.69315
Accuracy = 1e-06
Simpson results:
    0.69317    0.69315    0.69315

Simpson percentage differences:
    0.0032623    7.2063e-07    4.5073e-08

Simpson converged at n = 14
Result = 0.69315
Accuracy = 8.0332e-07
```

*fx* >>

## 5 Appendices

### 5.1 Your code script and functions

Lab1\_sin.m( main script)

```
clear
format short g

N = 100;
x = linspace(-pi, pi, N)';
y = sin(x);
figure(1)
plot(x, y)
title('y = sin(x)')
xlabel('x')
ylabel('sin(x)')

sf = 0.1;
y2 = sin_scale(x, sf);

figure(2)
plot(x, y2)
title('y = sin(x/sf) with sf=0.1')
xlabel('x')
ylabel('sin(x/sf)')

sf = [0.1 0.5 2];
N_sf = numel(sf);

sin_array = zeros(N, N_sf); % allocate space

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

figure(3)
plot(x, sin_array)
title('Scaled sine with sf = [0.1 0.5 2]')
xlabel('x')
ylabel('sin(x/sf)')

% Numerical differentiation of cos(x)
y = cos(x);
h = x(2) - x(1);
dy = zeros(N,1);

for i = 1:N-1
    dy(i) = (y(i+1) - y(i)) / h;
end

dy(N) = dy(N-1);

figure(4)
plot(x, dy, 'b', x, -sin(x), 'r--')
legend('Numerical derivative', '-sin(x) Analytical')
```

```

title('Derivative of cos(x) using Forward Difference')
xlabel('x')
ylabel('dy/dx')

% TRAPEZOIDAL INTEGRATION of x*sin(x) from 0 to pi/2
a = 0;
b = pi/2;

I_trap = trap1('f_xsinx', a, b);
disp(['Trapezoidal result = ', num2str(I_trap)]);

I_analytic = 1;
percentage_error_Trap = abs((I_analytic - I_trap) / I_analytic) * 100;

disp(['Percentage error = ', num2str(percentage_error_Trap), '%']);

% Composite trapezoid for x*sin(x)
I_comp = comp_trap('f_xsinx', 0, pi/2, 100);
disp(['Composite trap result (x*sin(x), n=100) = ', num2str(I_comp)]);

error_comp = abs((1 - I_comp)/1) * 100;
disp(['Composite % error = ', num2str(error_comp), '%']);

n_values = [5 50 100];
results = zeros(1,3);

for i = 1:length(n_values)
    results(i) = comp_trap('f_invx', 1, 2, n_values(i));
end

disp('Composite trapezoid results for f(x)=1/x:');
disp(results);

analytic = log(2);
percent_diff = abs((results - analytic) / analytic) * 100;

disp('Percentage differences:');
disp(percent_diff);

target = log(2);
n = 1;
accuracy = 1;

while accuracy > 1e-6
    n = n + 1;
    I_test = comp_trap('f_invx', 1, 2, n);
    accuracy = abs(I_test - target);
end

disp(['Composite trapezoid converged at n = ', num2str(n)]);
disp(['Result = ', num2str(I_test)]);
disp(['Accuracy = ', num2str(accuracy)]);

% SIMPSON'S RULE
n_vals = [6 50 100];
simp_results = zeros(1,3);

```

```
for i = 1:length(n_vals)
    simp_results(i) = simpson('f_invx', 1, 2, n_vals(i));
end

disp('Simpson results:');
disp(simp_results);

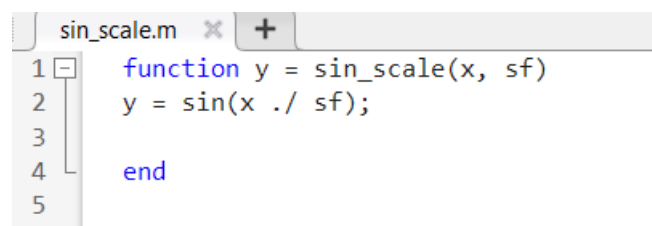
analytic = log(2);
simp_percent_diff = abs((simp_results - analytic) / analytic) * 100;

disp('Simpson percentage differences:');
disp(simp_percent_diff);

target = log(2);
n = 2; % must be even
accuracy = 1;

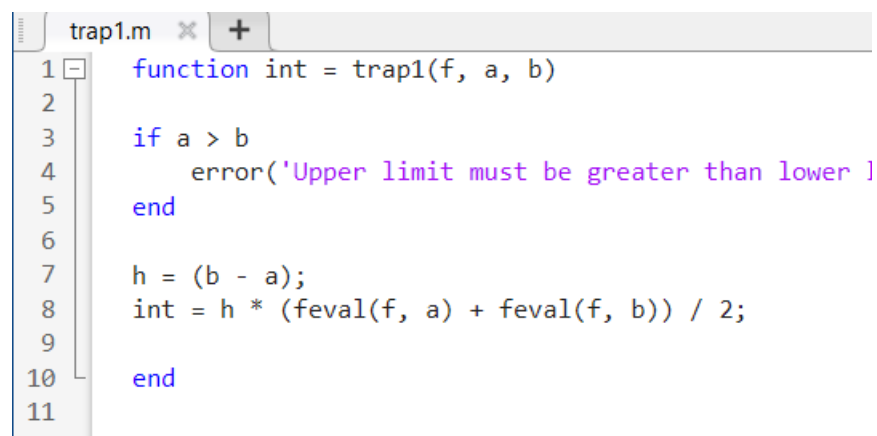
while accuracy > 1e-6
    n = n + 2; % increment by 2 to stay even
    I_test = simpson('f_invx', 1, 2, n);
    accuracy = abs(I_test - target);
end

disp(['Simpson converged at n = ', num2str(n)]);
disp(['Result = ', num2str(I_test)]);
disp(['Accuracy = ', num2str(accuracy)]);
```



The screenshot shows a MATLAB script editor window titled 'sin\_scale.m'. The script contains a function definition for 'sin\_scale' that takes two inputs, 'x' and 'sf', and returns 'y'. The function calculates 'y' as 'sin(x ./ sf)'. The script is 5 lines long.

```
1 function y = sin_scale(x, sf)
2   y = sin(x ./ sf);
3
4   end
5
```



The screenshot shows a MATLAB script editor window titled 'trap1.m'. The script contains a function definition for 'trap1' that takes three inputs, 'f', 'a', and 'b', and returns 'int'. The function checks if 'a' is greater than 'b' and throws an error if so. It then calculates 'h' as '(b - a)' and 'int' as 'h \* (feval(f, a) + feval(f, b)) / 2'. The script is 11 lines long.

```
1 function int = trap1(f, a, b)
2
3   if a > b
4       error('Upper limit must be greater than lower ')
5   end
6
7   h = (b - a);
8   int = h * (feval(f, a) + feval(f, b)) / 2;
9
10  end
11
```

```
trap1.m x simpson.m +
1 function I = simpson(f, a, b, n)
2
3     if mod(n,2) ~= 0
4         error('n must be even for Simpson''s rule')
5     end
6
7     h = (b - a) / n;
8     x = a:h:b;
9     y = feval(f, x);
10
11     odd_sum = sum(y(2:2:end-1));
12
13     even_sum = sum(y(3:2:end-2));
14
15     I = (h/3) * (y(1) + 4*odd_sum + 2*even_sum + y(end));
16
17 end
```

```
trap1.m x simpson.m x f_xsinx.m +
1 function y = f_xsinx(x)
2     y = x .* sin(x);
3 end
4
```

```
f_invx.m +
1 function y = f_invx(x)
2     y = 1 ./ x;
3 end
4
```

```
comp_trap.m +
1 function I = comp_trap(f, a, b, n)
2
3     h = (b - a) / n;
4     x = a:h:b;
5     y = feval(f, x);
6
7     I = h * (y(1) + 2*sum(y(2:end-1)) + y(end)) / 2;
8
9 end
10 |
```