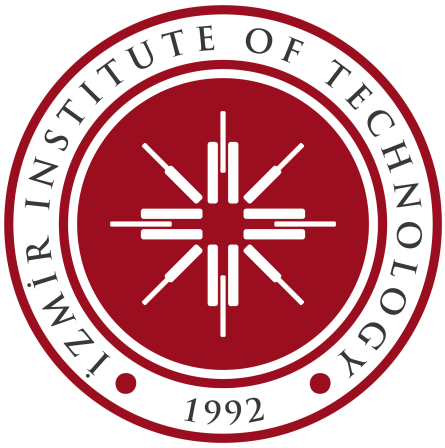

İzmir Institute of Technology

Numerical Methods in Engineering

CE301

Assignment 1

Section B

Mert Emrem - 250203015
Mechanical Engineering

Summer 2021

To compare for error, the value of the integral that is accurate to more decimal places than are required is obtained via WolframAlpha:

$$\int_{\pi/6}^{\pi/2} \frac{\cos(x)(e^x + x)}{x^2 - \ln(x)} \, \mathrm{d}x = 1.6724695$$

The MATLAB code (see Appendix A.1) output is as follows:

```
The number of subintervals needed for:

----> Compound mid-point: 173
----> Compound trapezoid: 244
----> Simpson's rule:     17
```

The code can be further deliberated upon at instructors' request.

Appendix

A.1 Computer Code

```
%%%%%%%%%%%%%%%%%%%%    Mert Emrem - 250203015    %%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%    CE301 - Assignment #1     %%%%%%%%%%%%%%%%%%%%

clc; clear all;

[a, b, err_goal, h_n] = deal(pi/6, pi/2, 1e-5, 1);

err = err_goal + 1;

true_int = 1.6724695;

f = @(x)(cos(x)*(exp(x)+x))/(x^2-log(x));

%%%% Compound mid-point:

while err > err_goal

    h = (b-a)/h_n;

    sum = 0;

    for k = 1:h_n

        z_i = a +((k-1)*h)+(h/2);
        nth_section = h*f(z_i);
        sum = sum + nth_section;

    end

    err = abs(sum - true_int);
    h_n = h_n + 1;

end

comp_mp_h = h_n;

%%%% Reset starting conditions
```

```
38
39 h_n = 1;
40 err = err_goal + 1;
41
42
43
44 %%%% Compound trapezoid:
45
46 %       (a+h*(k-1))+(a+h*k)
47 %  h * --------------------
48 %                2
49
50 while err > err_goal
51
52     h = (b-a)/h_n;
53
54     sum = 0;
55
56     for k = 1:h_n
57
58         x_alpha = a + h*(k-1);
59         x_beta = a + h*k;
60         nth_section = h * (f(x_alpha)+f(x_beta))/2;
61         sum = sum + nth_section;
62
63     end
64
65     err = abs(sum - true_int);
66     h_n = h_n + 1;
67
68 end
69
70 comp_trap_h = h_n;
71
72 %%%% Reset starting conditions
73
74 h_n = 1;
75 err = err_goal + 1;
76
77
78 %%%% Simpson's rule:
79
80 % (h/3)(A + B + C + D), where A & D are f(x_1), f(x_n)
81
82
83 A = f(a);
84 D = f(b);
```

```
85
86
87  while err > err_goal
88
89      h = (b-a)/h_n;
90
91      sum = 0;
92
93      B = 0;
94      C = 0;
95
96      for k = 1:2:(h_n-1)
97
98          B_section = 4*f(a+h*k);
99          B = B + B_section;
100
101     end
102
103
104     for k = 2:2:(h_n-1)
105
106         C_section = 2*f(a+h*k);
107         C = C + C_section;
108
109     end
110
111     sum = (h/3)*(A + B + C + D);
112
113     err = abs(sum - true_int);
114     h_n = h_n + 1;
115
116 end
117
118 simpson_h = h_n;
119
120
121
122 disp("The number of subintervals needed for:");
123
124 line_1 = ['----> Compound mid-point: ', num2str(comp_mp_h)];
125 line_2 = ['----> Compound trapezoid: ', num2str(comp_trap_h)];
126 line_3 = ['----> Simpson''s rule:     ', num2str(simpson_h)];
127
128 disp(line_1);
129 disp(line_2);
130 disp(line_3);
```