BME 313L: INTRODUCTION TO NUMERICAL METHODS IN BIOMEDICAL ENGINEERING

LAB REPORT

Lab_10: Numerical Integration

Last name, First name: Kirk, Andrew

EID: alk2488

Lab Section: Friday

Problem 1.

Evaluate the following integral:

$$\int_{-2}^{4}(1 - x - 4x^3 + 2x^5)dx$$

(a) Analytically
(b) Single application of the trapezoidal rule
(c) Composite trapezoidal rule with n=10 & 20 points
(d) Single application of Simpson’s 1/3 rule
(e) Composite Simpson’s 3/8 rule
(f) Boole’s rule

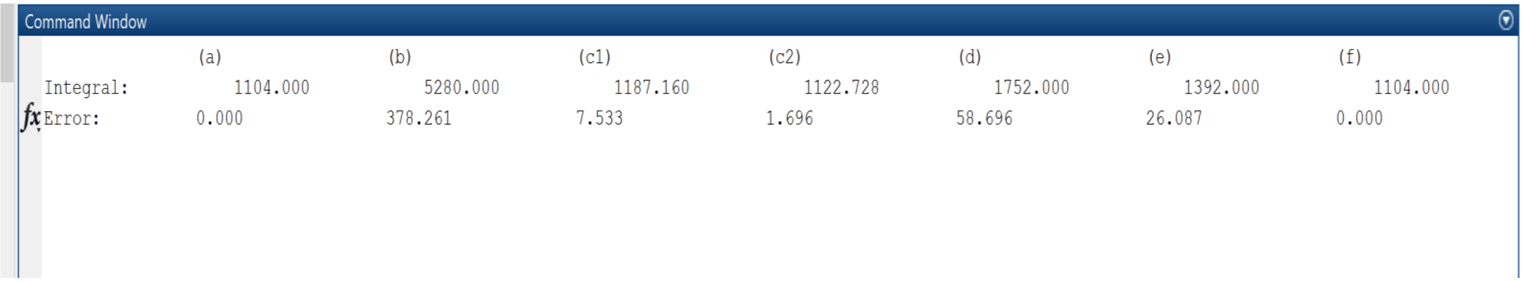
For each of the numerical estimates (b) through (f), determine the true percent relative error based on (a).

MATLAB code:

```
clear all
close all
clc ;

low = -2; %given values
high = 4;
func =@(x) 1-x-4*x.^3+2*x.^5;

integral =@(x) x - (x^2)/2 - x^4 + (x^6)/3; %part a
ans(1) = integral(4) - integral(-2);

ans(2) = (high - low)*(func(low) + func(high))/2; % part b

range1 = linspace(low,high,10);
range2 = linspace(low,high,20);
ans(3) = trapz(range1,func(range1)); %part c n=10
ans(4) = trapz(range2,func(range2)); %part c n=20

middle = 1; % Find the third, middle point
ans(5) = (high - low)*(func(low) + 4*func(middle) + func(high))/6;    %    par d Simpson's 1/3 formula
```

```
r = linspace(-2,4,4); % Simpson's 3/8 part e
ans(6) = (high - low)*(func(r(1)) + 3*func(r(2)) + 3*func(r(3)) + func(r(4)))/8;

r = linspace(-2,4,5); % Boole's part f
ans(7) = (high - low)*(7*func(r(1)) + 32*func(r(2)) + 12*func(r(3)) + 32*func(r(4)) + 7*func(r(5)))/90; % formula

error = zeros(1,length(ans));
for i=1:length(ans)
    error(i) = (abs(ans(1) - ans(i))/ans(1)*100); % finding error compared to analytical
end

fprintf('            \t\t(a)\t\t\t\t\t(b)\t\t\t\t\t(c1)\t\t\t\t(c2)\t\t\t\t(d)\t\t\t\t\t(e)\t\t\t\t\t(f)\n') %show answer
fprintf('Integral:\t\t\t')
for i = 1:length(ans)
fprintf('%.03f\t\t\t',ans(i))    % answer
end
fprintf('\nError:\t\t\t')
for i = 1:length(ans)
fprintf('%.03f\t\t\t\t',error(i)) % errors
```

Results:



```
                (a)             (b)             (c1)            (c2)            (d)             (e)             (f)
Integral:          1104.000        5280.000        1187.160        1122.728        1752.000        1392.000        1104.000
Error:          0.000           378.261         7.533           1.696           58.696          26.087          0.000
```

Discussion:

This was a very straight forward question. First the analytical answer was found by doing normal intergration techniques. Then the other methods were solved for by using their equations and rules. The error was found for each of these by comparing to the analytical solution. This was good practice to learn how to use the various intergration techniques and see how accurate/innacurate they can be.

Problem 2.

The force on a sailboat mast can be represented by the following function:

$$F = \int_0^H 100\left(\frac{z}{5+z}\right)e^{-2z/H}\,dz$$

where z = the elevation above the deck and H = the height of the mast. Compute F for the case where H = 30 using:

(a) Romberg integration to a tolerance of $\varepsilon_s = 0.5\%$ using the provided MATLAB functions *romberg.m*. and *trap.m*. Report the integral estimates and the approximate percentage errors at every iteration.

(b) Two-point Gauss-Legendre formula. Besides the integral value, also report the variable transformation used for the two-point Gauss-Legendre formula.

(c) MATLAB *quad* and *quadl* function

Things to discuss:

1. Discuss Romberg Integration technique in brief. Discuss its efficiency over Trapezoidal rule.
2. Why do we need transformation to apply Two-point Gauss-Legendre formula? Discuss the transformation.
3. What is the difference between *quad* and *quadl* function?

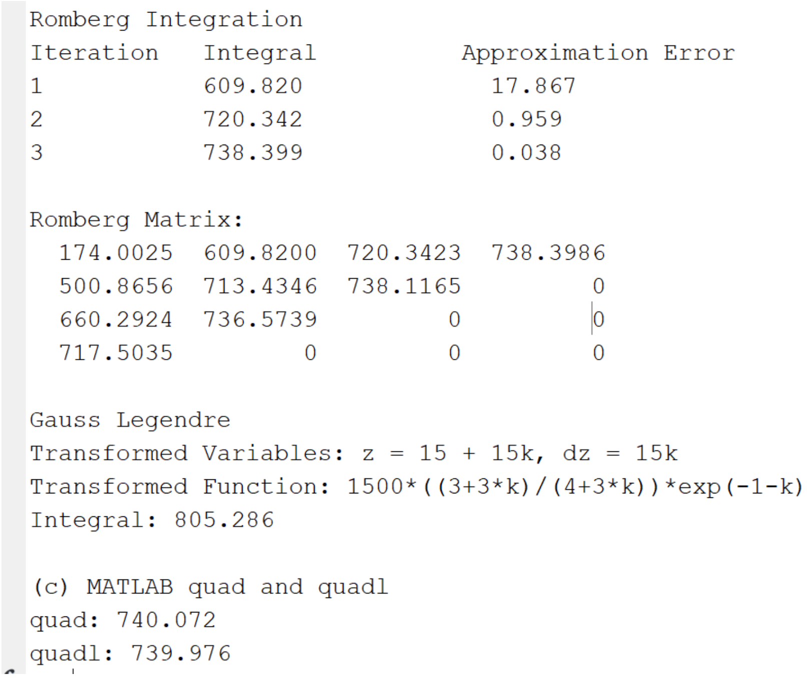
Things to discuss: (100 word minimum for each question, 50 word minimum for discussing what you learned, what was reinforced)

MATLAB code:

```
clear all
close all
clc ;

H = 30;    % given
low = 0;
high = H;
func =@(z) (100.*(z./(5+z)).*exp(-2.*z./H));

[q,Error,iter] = romberg_AK(func,low,high,0.5); %part a romberg
fprintf('Romberg Integration\nIteration      Integral                Approximation Error\n')    % Labels
for i = 1:iter
fprintf('%0.0f                    %0.3f                    %0.3f\n',i,q(1,i+1),Error(i))
end
fprintf('\n')
disp('Romberg Matrix:')
disp(q)

disp('Gauss Legendre') %part b Two point Gauss Legendre
```

```
trans =@(k) 1500*((3+3*k)/(4+3*k))*exp(-1-k); % transformed equation
disp('Transformed Variables: z = 15 + 15k, dz = 15k')
disp('Transformed Function: 1500*((3+3*k)/(4+3*k))*exp(-1-k)')
I = 1*trans(-1/sqrt(3)) + 1*trans(1/sqrt(3));    % Gauss Legendre for two point
fprintf('Integral: %.03f\n\n', I)

C1 = quad(func,low,high,0.5); % part c quad and quadl
C2 = quadl(func,low,high,0.5);    % tolerance of 0.5 because default tolerance was giving error
fprintf('(c) MATLAB quad and quadl\nquad: %0.3f \nquadl: %0.3f\n',C1,C2)
```

Results:



```
Romberg Integration
Iteration   Integral          Approximation Error
1           609.820             17.867
2           720.342             0.959
3           738.399             0.038

Romberg Matrix:
  174.0025  609.8200  720.3423  738.3986
  500.8656  713.4346  738.1165         0
  660.2924  736.5739         0         0
  717.5035         0         0         0

Gauss Legendre
Transformed Variables: z = 15 + 15k, dz = 15k
Transformed Function: 1500*((3+3*k)/(4+3*k))*exp(-1-k)
Integral: 805.286

(c) MATLAB quad and quadl
quad: 740.072
quadl: 739.976
```

Discussion:

Romberg technique uses two different step sizes with the smaller one being atleast half the bigger one. This is then added together and averaged in a way to give precedence to the more accurate step size. This makes sure the error of the combined value is smaller than both step size errors individually. It is better than the trapezoidal rule because you don't have to keep finding the area of the trapezoid under the curve which is very tedious.

Th Gauss Legendre technique requires you to fit the formula with limits of -1 and 1. This helps limit errors that you would normally get in trapezoidal rule. You transform the equation by assuming it acts as a line and solve for x by the equation. You use the actual given limits to calculate the new formula that is fitted to 1 and -1 and then use the Gauss Legendre formula to solve.

The quad function is better for low accuracy and non smooth integrands. The reverse is true for the quadl function. The quad function uses recursive adaptive simpson formula while quadl uses recursive adaptive Labatto technique.