MATH 446/OR 481
Project 1

January 31, 2023

Project Conclusion - The Bisection Method is able to find a root accurate to 6 decimal places for the first function but not for the second function. In theory, the bisection method can give as much precision as we need, but in case of the second function, as the method approaches the real root, the function evaluates to a number less than the machine precision, causing errors in machine arithmetic, and never ending loops. Hence, we must stop the computation when the function evaluates to a number less than the machine precision, but that doesn't satisfy our accuracy requirement. However, considering that the "less accurate" computed root evaluates to a very small number close to zero (the machine precision and less), it should be acceptable as a good enough approximation to the actual root.

Q1. For the first function

$$f(x) = e^{3x} - 6e^{2x} + 12e^{x} - \frac{15}{2}$$

we obtain the following results from MATLAB.

```
f =

  function_handle with value:

    @(x)exp(3*x)-6*exp(2*x)+12*exp(x)-15/2

Root Estimate 1 = 0.18755740
Root Estimate 2 = 0.18755732
Root Estimate 3 = 0.18755738
Hence, computed root upto 6 decimal places = 0.187557
```

All the three roots agree up to 6 decimal places. We use a tolerance level of 10^{-7} for the computations. The computed root, accurate to 6 decimal places is $\boxed{0.187557}$.

Q2. The tables of checking errors is shown below.

```
f =

  function_handle with value:

    @(x)exp(3*x)-6*exp(2*x)+12*exp(x)-15/2

Root Estimate 1 = 0.18755740
Root Estimate 2 = 0.18755732
Root Estimate 3 = 0.18755738
Hence, computed root upto 6 decimal places = 0.187557

Table of checking errors

            x                      f
    ___________________    ____________________

    0.187552375431061    -1.14265211266229e-05
    0.187553375431061    -9.14673699714541e-06
    0.187554375431061    -6.86695751550559e-06
    0.187555375431061     -4.5871826834798e-06
    0.187556375431061    -2.30741250195621e-06
    0.187557375431061    -2.76469744875385e-08
    0.187558375431061     2.25211391047253e-06
    0.187559375431061     4.53187014226586e-06
    0.187560375431061     6.81162172000427e-06
    0.187561375431061     9.09136865345772e-06
    0.187562375431061     1.13711109328563e-05
```

We can see that we get the least error with the computed root, and all other values give higher errors. Hence, we can be confident about the accuracy of our computed root. Our best guess for the root is $\boxed{0.187557}$ and we are confident about all the digits.

Q3. For the second function

$$g(x) = e^{3x} - 6e^{2x} + 12e^{x} - 8$$

we obtain the following results from MATLAB.

```
g =

  function_handle with value:

    @(x)exp(3*x)-6*exp(2*x)+12*exp(x)-8

Root Estimate 1 = 0.69314194
Computed root upto 6 decimal places = 0.693142

Root Estimate 2 = 0.69315338
Computed root upto 6 decimal places = 0.693153

Root Estimate 3 = 0.69315491
Computed root upto 6 decimal places = 0.693155
```

The three roots do not agree to 6 decimal places. We use a tolerance level of 10^{-7} for the computations. We have three different approximations to the root based on 3 different starting intervals - $\boxed{0.693142, 0.693153, 0.693155}$.

Q4. The tables of checking errors is shown below for each of the computed root approximations.

```
g =

  function_handle with value:

    @(x)exp(3*x)-6*exp(2*x)+12*exp(x)-8

Root Estimate 1 = 0.69314194
Computed root upto 6 decimal places = 0.693142

Table of checking errors

           x                  g
    _______________    ____________________

    0.693136937255859     -7.105427357601e-15
    0.693137937255859     -3.5527136788005e-15
    0.693138937255859                        0
    0.693139937255859    -5.32907051820075e-15
    0.693140937255859                        0
    0.693141937255859                        0
    0.693142937255859      3.5527136788005e-15
    0.693143937255859    -1.77635683940025e-15
    0.693144937255859     1.77635683940025e-15
    0.693145937255859                        0
    0.693146937255859     1.77635683940025e-15

Root Estimate 2 = 0.69315338
Computed root upto 6 decimal places = 0.693153

Table of checking errors

           x                  g
    _______________    ____________________

    0.693148381347656    3.5527136788005e-15
    0.693149381347656    3.5527136788005e-15
    0.693150381347656    3.5527136788005e-15
    0.693151381347656    3.5527136788005e-15
    0.693152381347656    3.5527136788005e-15
    0.693153381347656                      0
    0.693154381347656    3.5527136788005e-15
    0.693155381347656                      0
    0.693156381347656    3.5527136788005e-15
    0.693157381347656    1.4210854715202e-14
    0.693158381347656     7.105427357601e-15
```

```
Root Estimate 3 = 0.69315491
Computed root upto 6 decimal places = 0.693155

Table of checking errors

           x                    g
    ________________    ____________________

    0.693149907226563                     0
    0.693150907226563    3.5527136788005e-15
    0.693151907226563    3.5527136788005e-15
    0.693152907226563                     0
    0.693153907226563     7.105427357601e-15
    0.693154907226563                     0
    0.693155907226563     7.105427357601e-15
    0.693156907226563     7.105427357601e-15
    0.693157907226563    1.4210854715202e-14
    0.693158907226563    1.4210854715202e-14
    0.693159907226563    1.4210854715202e-14
```

The table of checking errors contains multiple zeros. All values evaluate to very small numbers and some evaluate to 0, which is clearly not correct, since there is only one root and many values evaluate to 0. The problem is machine precision. The values evaluate to numbers less than the machine precision which are interpreted as 0 by MATLAB.

Appendix - Complete Command Window Output

```
f =

  function_handle with value:

    @(x)exp(3*x)-6*exp(2*x)+12*exp(x)-15/2

Root Estimate 1 = 0.18755740
Root Estimate 2 = 0.18755732
Root Estimate 3 = 0.18755738
Hence, computed root upto 6 decimal places = 0.187557

Table of checking errors

            x                     f
    ________________    ______________________

    0.187552375431061    -1.14265211266229e-05
    0.187553375431061    -9.14673699714541e-06
    0.187554375431061    -6.86695751550559e-06
    0.187555375431061     -4.5871826834798e-06
    0.187556375431061    -2.30741250195621e-06
    0.187557375431061    -2.76469744875385e-08
    0.187558375431061     2.25211391047253e-06
    0.187559375431061     4.53187014226586e-06
    0.187560375431061     6.81162172000427e-06
    0.187561375431061     9.09136865345772e-06
    0.187562375431061     1.13711109328563e-05


g =

  function_handle with value:

    @(x)exp(3*x)-6*exp(2*x)+12*exp(x)-8

Root Estimate 1 = 0.69314194
Computed root upto 6 decimal places = 0.693142

Table of checking errors

            x                     g
    ________________    ______________________

    0.693136937255859      -7.105427357601e-15
    0.693137937255859     -3.5527136788005e-15
    0.693138937255859                        0
    0.693139937255859    -5.32907051820075e-15
    0.693140937255859                        0
    0.693141937255859                        0
    0.693142937255859      3.5527136788005e-15
    0.693143937255859    -1.77635683940025e-15
    0.693144937255859     1.77635683940025e-15
    0.693145937255859                        0
    0.693146937255859     1.77635683940025e-15
```

```
Root Estimate 2 = 0.69315338
Computed root upto 6 decimal places = 0.693153

Table of checking errors

           x                    g
    ________________    ____________________

    0.693148381347656    3.5527136788005e-15
    0.693149381347656    3.5527136788005e-15
    0.693150381347656    3.5527136788005e-15
    0.693151381347656    3.5527136788005e-15
    0.693152381347656    3.5527136788005e-15
    0.693153381347656                      0
    0.693154381347656    3.5527136788005e-15
    0.693155381347656                      0
    0.693156381347656    3.5527136788005e-15
    0.693157381347656    1.4210854715202e-14
    0.693158381347656     7.105427357601e-15

Root Estimate 3 = 0.69315491
Computed root upto 6 decimal places = 0.693155

Table of checking errors

           x                    g
    ________________    ____________________

    0.693149907226563                      0
    0.693150907226563    3.5527136788005e-15
    0.693151907226563    3.5527136788005e-15
    0.693152907226563                      0
    0.693153907226563     7.105427357601e-15
    0.693154907226563                      0
    0.693155907226563     7.105427357601e-15
    0.693156907226563     7.105427357601e-15
    0.693157907226563    1.4210854715202e-14
    0.693158907226563    1.4210854715202e-14
    0.693159907226563    1.4210854715202e-14
```

Appendix - MATLAB Code - bisectionMethod.m

```
function root = bisectionMethod(f, lowerBound, upperBound, tol)
% computes root using bisection method
% f - function handle
% tol - tolerance

while ((upperBound - lowerBound) > tol)
% compute root
root = (upperBound + lowerBound)/2;

% root found to machine precision
if (abs(f(root)) < 1e-16)
return
end

% find new bracket
if (f(root)*f(upperBound) < 0)
lowerBound = root;
else
if (f(root)*f(lowerBound) < 0)
upperBound = root;
end
end
end

end
```

Appendix - MATLAB Code - proj_1.m

```
clc; clear; close all

%% Q1 and Q2

% function definition
f = @(x) exp(3*x) - 6*exp(2*x) + 12*exp(x) - 15/2

% compute root correct to 6 decimal places
% apply bisection method with different starting intervals
intervals = [0, 1; 0.1, 0.9; 0.1 0.6];
tol = 1e-7;

for i = 1:3
lowerBound = intervals(i, 1);
upperBound = intervals(i, 2);

% check for valid interval
if f(lowerBound)*f(upperBound) < 0
%fprintf("Yes\n");
root = bisectionMethod(f, lowerBound, upperBound, tol);
fprintf('Root Estimate %d = %.8f\n', i, root)
end
end

fprintf('Hence, computed root upto 6 decimal places = %.6f\n\n', ...
round(root, 6))

% make a table of errors
coeffs = -5:1:5;
testValues = root + coeffs*(1e-6);
checkingErrors = f(testValues);

fprintf('Table of checking errors \n\n')
tbl = table;
tbl.x = testValues';
tbl.f = checkingErrors';
format longEng
disp(tbl)
```

```
%% Q3 and Q4

% function definition
g = @(x) exp(3*x) - 6*exp(2*x) + 12*exp(x) - 8

%compute root correct to 6 decimal places
% apply bisection method with different starting intervals
intervals = [0, 1; 0.1, 0.9; 0.4 0.8];
tol = 1e-7;

for i = 1:3
lowerBound = intervals(i, 1);
upperBound = intervals(i, 2);

% check for valid interval
if (g(lowerBound)*g(upperBound) < 0)
%fprintf("Yes\n");
root = bisectionMethod(g, lowerBound, upperBound, tol);
fprintf('Root Estimate %d = %.8f\n', i, root)

fprintf('Computed root upto 6 decimal places = %.6f\n\n', ...
round(root, 6))

% make a table of errors
coeffs = -5:1:5;
testValues = root + coeffs*(1e-6);
checkingErrors = g(testValues);

fprintf('Table of checking errors \n\n')
tbl = table;
tbl.x = testValues';
tbl.g = checkingErrors';
format longEng
disp(tbl)
end
end

format default
```