```
% Anthony Galczak - Tristin Glunt
% CS375 HW5

clc
close all
% Q1 - Find number of zeros of f(x).

% f(x) = x^3 - y looks to have 1 real root.
% This is intuitively true by just thinking about
% some sample y's.
% y = 8: 8^(1/3) has 1 real root, 2.
% y = -8: (-8)^(1/3) has 1 real root, -2.
% The other 2 roots are in the complex plane,
% shifted by 2pi/3 and 4pi/3 in e^(i*theta).

% y is a constant, f(x) = x^3 - y
sym x;
y = 8;
f = @(x) x.^3 - y;

num_vals = 1000;
f_evals = zeros(1,num_vals);
x_s = linspace(-4,4,num_vals);

for i=1:num_vals
    f_evals(i) = f(x_s(i));
end

plot(x_s,f_evals);
title("Q1.) f(x) for y=8");
xlabel("x-axis");
ylabel("y-axis");
legend("f(x) = x^3 - y", "Location", "Northwest");

% Q3 - Test Bisection, Newton, Secant method.
fprintf("Output starts here\n");
tol = 10^-10;
format long
bi_arr = my_bisection(f, 1, 4, tol);
fprintf("Bisection root: %1.15f\n\n", bi_arr(end));

df = @(x) 3*x^2;
newt_arr = my_newton(f, df, 4, tol);
fprintf("Newton root: %1.15f\n\n", newt_arr(end));

sec_arr = my_secant(f, 1, 4, tol);
fprintf("Secant root: %1.15f\n\n", sec_arr(end));

% Q4 - Compute convergence rate for each method.
% Note: First real iteration on Newton method is 2 and on Secant
% method is 3.
bi_vals = bi_arr;
```

```
convergence_table("Bisection", bi_vals, 1);

newt_vals = newt_arr(1,2:end);
convergence_table("Newton", newt_vals, 2);

sec_vals = sec_arr(1,3:end);
convergence_table("Secant", sec_vals, 1.62);

% Function for printing tables for convergence analysis
function c = convergence_table(method_name, vals, r)

    for i=1:length(vals)
        iters(i) = i;
        errors(i) = abs(vals(i) - 2);
        if i == 1
            rates(i) = 0;
        else
            rates(i) = errors(i) / (errors(i-1) ^ r);
        end
    end

    fprintf("%s Method table, r = %1.2f\n", method_name, r);
    T = table(iters',errors',rates');
    T.Properties.VariableNames = {'i' 'Error' 'Conv_Rate'};
    disp(T);

end


% Q2 - Implement Bisection, Newton's, Secant as fn's.
function[x_arr] = my_bisection(f, a, b, tol)
    for i=1:100
        x_m = a + (b-a)/2;
        x_arr(i) = x_m;

        if sign(f(x_m)) == sign(f(a))
            a = x_m;
        else
            b = x_m;
        end

        % First halting criteria
        if (i >= 2)
            last_guess = x_arr(i-1);
            if abs(x_m - last_guess) < tol
                fprintf(['Bisection method hit halting criteria',...
                    ' 1 after %d iterations.\n'],i);
                return
            end
        end

        % Second halting criteria
        if abs(f(x_m)) < eps
            fprintf(['Bisection method hit halting criteria',...
```

```
                ' 2 after %d iterations.\n'],i);
            return
        end
    end
end

function[x_arr] = my_newton(f, df, x_0, tol)
    x_arr(1) = x_0;
    for i=2:100
        last_guess = x_arr(i-1);
        x_k = last_guess - (f(last_guess)/df(last_guess));
        x_arr(i) = x_k;

        if abs(x_k - last_guess) < tol
            fprintf(['Newton method hit halting criteria',...
                ' 1 after %d iterations.\n'],(i-1));
            return
        end

        if abs(f(x_k)) < eps
            fprintf(['Newton method hit halting criteria',...
                ' 2 after %d iterations.\n'],(i-1));
        end
    end

end

function[x_arr] = my_secant(f, x_0, x_1, tol)
    x_arr(1) = x_0;
    x_arr(2) = x_1;

    for i=3:100
        x_k2 = x_arr(i-2);
        x_k1 = x_arr(i-1);
        term2 = f(x_k1)*((x_k1 - x_k2)/(f(x_k1) - f(x_k2)));
        x_k = x_k1 - term2;
        x_arr(i) = x_k;

        if abs(x_k - x_k1) < tol
            fprintf(['Secant method hit halting criteria',...
                ' 1 after %d iterations.\n'],(i-2));
            return
        end

        if abs(f(x_k)) < eps
            fprintf(['Secant method hit halting criteria',...
                ' 2 after %d iterations.\n'],(i-1));
            return
        end
    end

end
```

Output starts here

Bisection method hit halting criteria 1 after 35 iterations.
Bisection root: 2.000000000029104

Newton method hit halting criteria 1 after 7 iterations.
Newton root: 2.000000000000000

Secant method hit halting criteria 1 after 9 iterations.
Secant root: 2.000000000000000

Bisection Method table, r = 1.00

i	Error	Conv_Rate
1	0.5	0
2	0.25	0.5
3	0.125	0.5
4	0.0625	0.5
5	0.03125	0.5
6	0.015625	0.5
7	0.0078125	0.5
8	0.00390625	0.5
9	0.001953125	0.5
10	0.0009765625	0.5
11	0.00048828125	0.5
12	0.000244140625	0.5
13	0.0001220703125	0.5
14	6.103515625e-05	0.5
15	3.0517578125e-05	0.5
16	1.52587890625e-05	0.5
17	7.62939453125e-06	0.5
18	3.814697265625e-06	0.5
19	1.9073486328125e-06	0.5
20	9.5367431640625e-07	0.5
21	4.76837158203125e-07	0.5
22	2.38418579101562e-07	0.5
23	1.19209289550781e-07	0.5
24	5.96046447753906e-08	0.5
25	2.98023223876953e-08	0.5
26	1.49011611938477e-08	0.5
27	7.45058059692383e-09	0.5
28	3.72529029846191e-09	0.5
29	1.86264514923096e-09	0.5
30	9.31322574615479e-10	0.5
31	4.65661287307739e-10	0.5
32	2.3283064365387e-10	0.5
33	1.16415321826935e-10	0.5
34	5.82076609134674e-11	0.5
35	2.91038304567337e-11	0.5

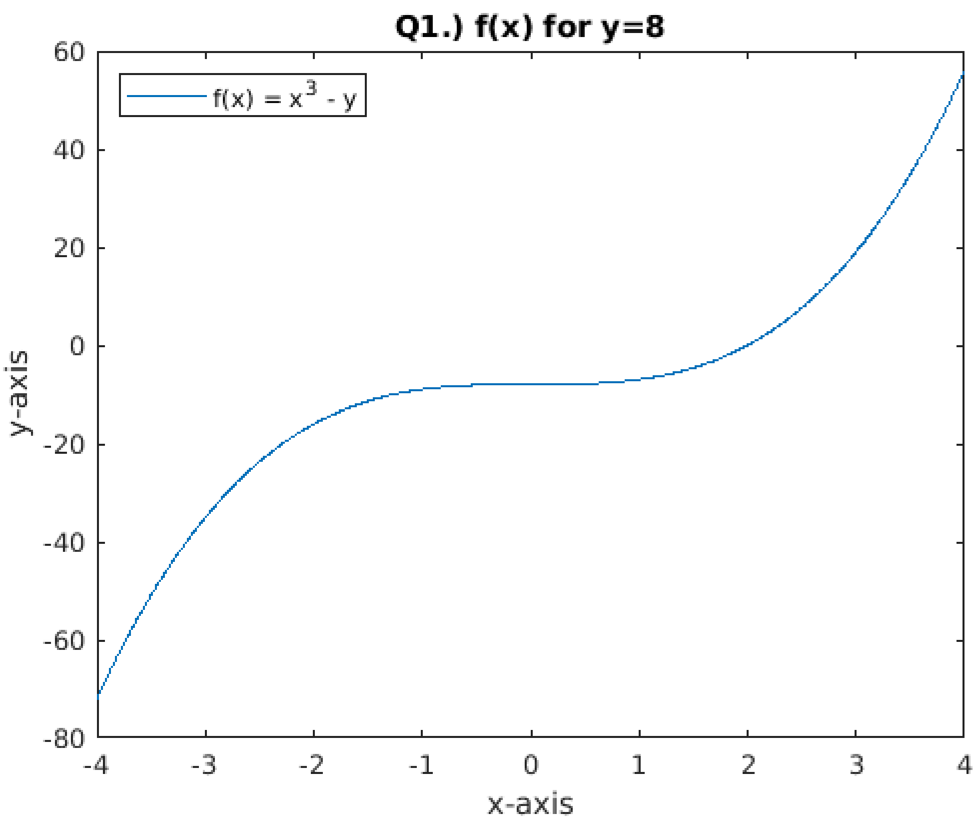
Newton Method table, r = 2.00

i	Error	Conv_Rate
1	0.833333333333333	0

```
     2       0.221068819684737    0.318339100346021
     3      0.0212735368091126    0.435296037790685
     4    0.000223114607898367    0.493001916400385
     5     2.48863623042439e-08     0.49992563649981
     6     4.44089209850063e-16    0.717046602291039
     7                        0                    0

Secant Method table, r = 1.62
     i            Error                Conv_Rate
     _    ____________________    _________________

     1       0.666666666666667                    0
     2       0.423076923076923    0.815994397306232
     3       0.217518135116297    0.876387440259626
     4      0.0489173446793951    0.579055527207449
     5     0.00502935344087141    0.667701420150649
     6     0.00012525606529934    0.662748984006864
     7     3.15494021796425e-07    0.661541563094523
     8     1.97579730354391e-11    0.672011570640884
     9                        0                    0
```



Published with MATLAB® R2018a