Problem 1

Statement: Using quadratic formula
Compute the roots of $f(x) = 4x^2 - 3x - 3$

Solution:

$$r = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

$a = 4, \quad b = -3, \quad c = -3$

$$r_1 = \frac{3 + \sqrt{(-3)^2 - 4(4)(-3)}}{2(4)}$$

$$= \frac{3 + \sqrt{9+48}}{8}$$

$$= \frac{3}{8} + \frac{\sqrt{57}}{8}$$

$$r_1 \approx 1.3187$$

$$r_2 = \frac{3 - \sqrt{(-3)^2 - 4(4)(-3)}}{2(4)}$$

$$= \frac{3}{8} - \frac{\sqrt{57}}{8}$$

$$r_2 \approx -0.5687$$

Roots: $r_1 \approx 1.3187$
$r_2 \approx -0.5687$

Problem 2

implementation of bisection included in code

Problem 3

Statement: transform function f into an appropriate function g for a fixed point problem

Solution:

$$f(x) = 4x^2 - 3x - 3$$

assume x^*: $\frac{3 - 4(x^*)^2}{3} = x^*$

$$\rightarrow 4(x^*)^2 + 3x^* - 3 = 0$$

$$g_1(x) = \frac{3 - 4x^2}{3}$$

$$g_1(x) = 1 - \frac{4}{3}x^2$$

assume x^*: $x^* = \left(\frac{3 + 3x}{4}\right)^{1/2}$

$$g_2(x) = \left(\frac{3 + 3x}{4}\right)^{1/2}$$

$$4x^2 - 3x - 3 = 0$$

$$(5x^2 - x^2) - 3x - 3 = 0$$

$$5x^2 - 3x - 3 = x^2$$

$$5x^2 - 3x = 3 + x^2$$

$$x(5x-3) = 3 + x^2$$

$$x = \frac{3 + x^2}{5x-3}$$

so that

$$g_3(x) = \frac{3 + x^2}{5x-3}$$

let $s = |g'(r)|$

$$g_1'(x) = -\frac{8}{3}x$$

$$|g_1'(x)| = \frac{8}{3}x \quad \text{will not converge}$$

$$|g_2'(x)| = \left| -\frac{3}{4\sqrt{3-3x}} \right| = \frac{3}{4\sqrt{3+3x}} < 1 \quad \text{on } (-1, \infty]$$

will converge

$$g_3'(x) = \frac{(5x-3)(2x) - (3+x^2)(5)}{(5x-3)(5x-3)}$$

$$= \frac{10x^2 - 6x - 15 - 5x^2}{(5x-3)^2}$$

$$|g_3'(x)| = \left| \frac{5x^2 - 6x - 15}{(5x-3)^2} \right|$$

@ $x = 1.3187$

$|g_3'(x)| > 1$ won't converge

use $g_2(x)$

Problem 4

implemented in Matlab Code

Problem 5

implemented in Matlab Code

Table of Contents

house keeping

```
clc;
clear all;
close all;
%{
        CSCI 3656 Homework 2
        Author: Connor O'Reilly
        Email: coor1752@colorado.edu
        Last Edited: 9/8/2021
%}
```

Question 2

implementaion of bisection method is at the bottom

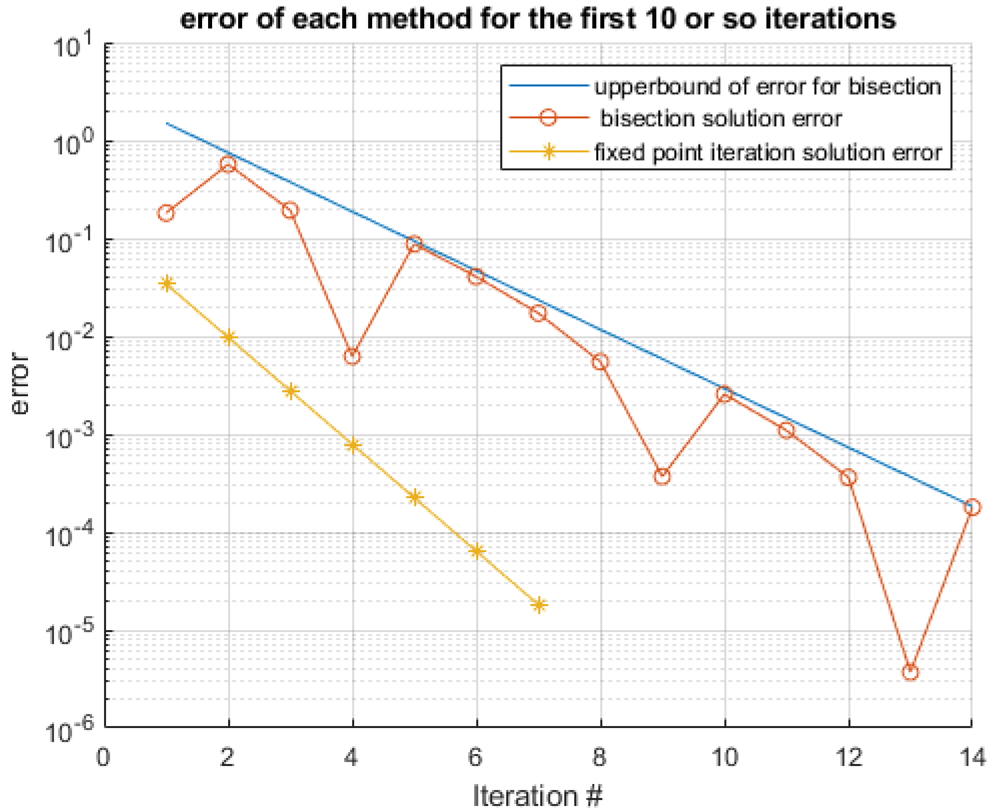
Part 5

```
%Determine interval for bisection method
f = @ ( x ) 4*x.^2 - 3*x - 3;
%f(x) < 0 at x = 0 and f(x) > 0 at x =3 , inwhich f(0) * f(3) <  0 so
 there
%is a root on interval [ 0 , 3 ]
interval = [0 , 3];
%tolerance set based off of quadratic formula values also same
 tolerance
%used in class
tolerance = 1e-4;
approx = bisection(f , interval , tolerance);
%approx of root using quadratic forumla
r = 1.318729304408844;

%display results
fprintf("actual root of f(x) : %0.8f\n",r);
fprintf("Approximation of root after %i iterations of bisection
 method: %0.8f\n", length(approx) , approx(end))

%plot error of bisection method
% error after n steps is (b-a)/2^(n+1)
%upper bound
err_bisection = (interval(2) - interval(1)) ./ 2.^(1:
(length(approx)-1) + 1);
```

```
%actual error
err_act_bi = abs( r - approx);

%fixed point method
%use g 2 from hw question
g2 = @(x) sqrt( ( 3+ 3*x) / 4 );
%get approximation of fixed point
% initial point should be around a root we computed using the
 quadratic
% formula but after guess and check g(1.2) is kinda close to 1.2
approx_fixed = fixedpoint(g2, 1.2 , tolerance );
err_fixed = abs(r - approx_fixed);
fprintf("Approximation of root after %i iterations of fixed point
 method: %0.8f\n", length(approx_fixed) , approx_fixed(end))
%plot error
figure(1)
grid on
hold on
set(gca, 'YScale', 'log')
xlabel('Iteration #');
ylabel('error')
title('error of each method for the first 10 or so iterations')
plot(1:(length(approx)), err_bisection)
plot(1:length(approx) , err_act_bi , ' -o ');
plot(1:length(approx_fixed) , err_fixed , ' -*');
legend( 'upperbound of error for bisection', ' bisection solution
 error ', 'fixed point iteration solution error')

function [approx] = bisection(f, int, tol)
%{
    matlab implementation of bisection method for root finding
    inputs:
%}
%get a and b
a = int(1); b = int(end);
%check that the f(a)f(b) < 0
it = 1; %iterator
if ( f(a) * f(b) ) < 0
    while( (b - a )/2 > tol )
        approx(it) = (a + b)/2;
        if( f(approx(it)) == 0 )
            break
        elseif (f(a)*f(approx(it)) < 0)
            b = approx(it);
        else
            a = approx(it);
        end
        it = it +1;
    end
else
    error("Interval not valid");
end

end
```

```matlab
function[approx ] = fixedpoint(g, x0 , tol)
%{
    matlab implementation of fixed point iteration algorithm applied
 to
    function f
    Inputs:
        f: function which fixed point iteration algorithm is applied
        x0: starting guess
        tol: tolerance
    Output:
        approx: approximation of solution
%}
%max iterations limit
it_lim = 1000;

x(1) = x0;


for i = 1 : it_lim
    x(i+1) = g(x(i));
    abs_err = abs(x(i+1) - x(i));
    rel_err = abs_err / abs( x(i+1) );

    %stoping criteria 1
    if abs_err < tol
        break
        %stopping criteria 2
    elseif rel_err < tol
        break
    end
    approx(i) = x(i+1);
end

if i == it_lim
    error('method failed to converge')
end
%return approximation
approx(i) = x(i+1);

end
```

actual root of f(x) : 1.31872930
Approximation of root after 14 iterations of bisection method: 1.31890869
Approximation of root after 7 iterations of fixed point method: 1.31871112



Published with MATLAB® R2021a