MATH-4800 **NUMERICAL COMPUTING** **Summer 2023**

Assignment 2, due before class, Thursday June 8, 2023.

Haowen He heh4@rpi.edu

1. Let $P(x) = x^3 - 2x^2 - 5x + 8$.

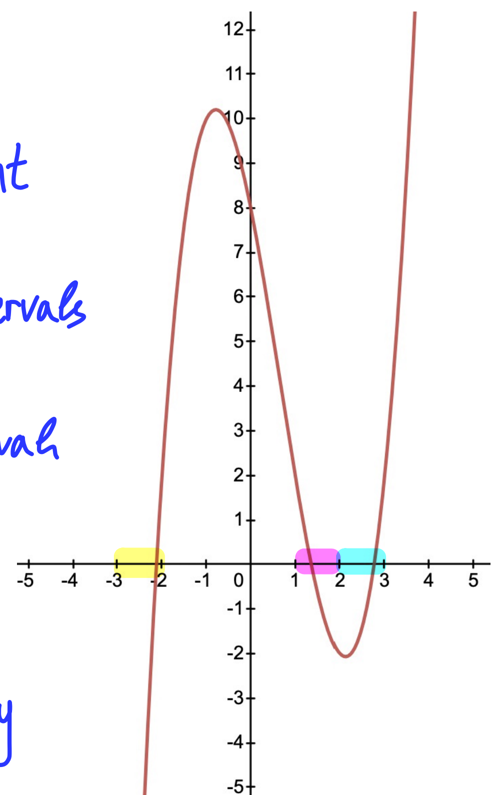
   (a) Show analytically (following the work in class) that $P(x)$ has three real roots, and find intervals of x that bracket each root individually.

   (b) Write a Matlab function that determines the root of a function using the bisection method. Use your function to determine the three roots of $P(x)$. For each root, use the stopping criterion $b - a < 10^{-6}$, where $[a, b]$ is the interval of the bracket at the n^{th} iteration. (Hint: see the example Matlab codes on LMS.)

   (c) Repeat the work in part (b), but now using the method of false position (as discussed in class). For each root, use the stopping criterion $|x_{n+1} - x_n| < 10^{-6}$, where x_n and x_{n+1} are successive approximations of the root during the iteration.

(a) Consider the cubic polynomial $P(x) = x^3 - 2x^2 - 5x + 8$.

It follows that $P(x)$ is continuous everywhere and has at most three distinct roots. We want to show that $P(x)$ changes sign on some intervals so that $P(x)$ must have a root on that interval by the intermediate value theorem. From the plot of $P(x)$ on the right, we can identify three such intervals:



check: setting $P'(x) = 3x^2 - 4x - 5 = 0$

$\Rightarrow x_1 \approx -0.8, \quad x_2 \approx 2.1$

$P''(x) = 6x - 4 \Rightarrow P''(x_1) < 0 \ \& \ P''(x_2) > 0$

- $[-3, -2]$ $\quad P(-3) = -22 < 0$, $P(-2) = 2 > 0$
- $[1, 2]$ $\quad P(1) = 2 > 0$, $P(2) = -2 < 0$
- $[2, 3]$ $\quad P(2) = -2 < 0$, $P(3) = 2 > 0$

Therefore, $P(x)$ has three distinct real roots, and there is a unique root in each of the three intervals above.

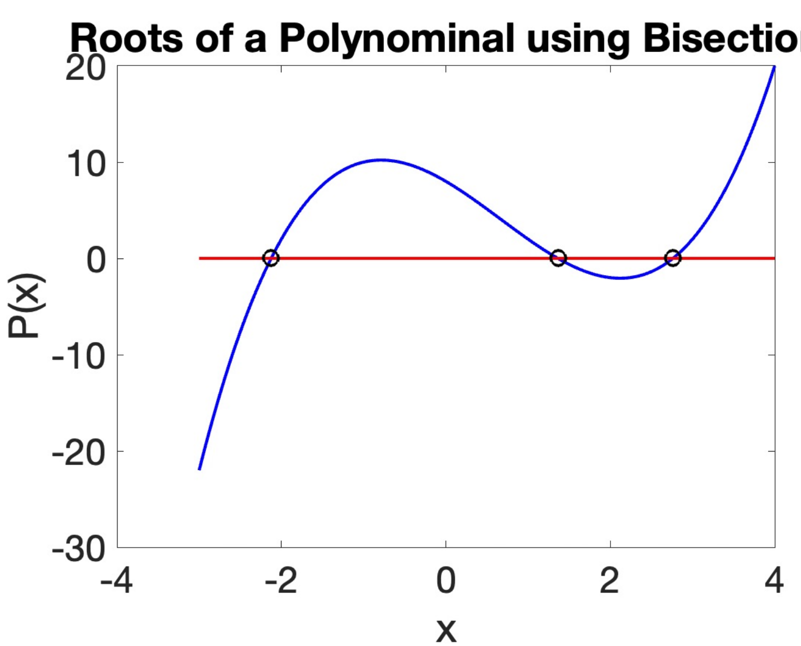
myBisection.m

from in-class example

use the midpoint of the interval to update the endpoints

```
function x=myBisection(f,xspan,tol)

% Find a root x of the equation f(x)=0 using the
% bisection method.  The initial interval is xspan=[a,b]
% and the iteration stops when b-a<tol.

a=xspan(1);
b=xspan(2);

fa=feval(f,a);
if fa==0
  x=a; return
end
fb=feval(f,b);
if fb==0
  x=b; return
end
if fa*fb>0
  error('Error : f(a)*f(b)>0')
end

n=0;
while b-a>tol
  n=n+1;
  x=(a+b)/2; fx=feval(f,x);
  fprintf('n=%d  a=%1.8f b=%1.8f f(x)=%1.2e\n',n,a,b,fx)
  if fx==0
    return
  end
  if fx*fa<0
    b=x; fb=fx;
  else
    a=x; fa=fx;
  end
end
```

```
% clear all variables and figures.
clear all
close all

% Set defaults for plotting
fontSize=24; lineWidth=2; markerSize=10;
set(0,'DefaultLineMarkerSize',markerSize);
set(0,'DefaultLineLineWidth',lineWidth);
set(0,'DefaultAxesFontSize',fontSize);
set(0,'DefaultLegendFontSize',fontSize);

% define and plot the polynomial y=x.^3-2*x.^2-5*x+8
P = @(x) x.^3-2*x.^2-5*x+8;
x=linspace(-3,4,300);
y=P(x);
plot(x,y,'b',x,zeros(size(x)),'r')

% Root on [-3,-2]
fprintf('Iteration for root on [-3,-2] :\n')
xspan=[-3,-2]; tol=10e-6;
x1=myBisection(P,xspan,tol);

% Root on [1,2]
fprintf('Iteration for root on [1,2] :\n')
xspan=[1,2]; tol=10e-6;
x2=myBisection(P,xspan,tol);

% Root on [2,3]
fprintf('Iteration for root on [2,3] :\n')
xspan=[2,3]; tol=10e-6;
x3=myBisection(P,xspan,tol);

% plot roots
hold on
roots=[x1 x2 x3];
plot(roots,zeros(3,1),'ko')
hold off
xlabel('x')
ylabel('P(x)')
title('Roots of a Polynominal using Bisection')
```




```
Iteration for root on [-3,-2] :
n=1  a=-3.00000000 b=-2.00000000 f(x)=-7.62e+00
n=2  a=-2.50000000 b=-2.00000000 f(x)=-2.27e+00
n=3  a=-2.25000000 b=-2.00000000 f(x)=-1.95e-03
n=4  a=-2.12500000 b=-2.00000000 f(x)=1.03e+00
n=5  a=-2.12500000 b=-2.06250000 f(x)=5.23e-01
n=6  a=-2.12500000 b=-2.09375000 f(x)=2.62e-01
n=7  a=-2.12500000 b=-2.10937500 f(x)=1.31e-01
n=8  a=-2.12500000 b=-2.11718750 f(x)=6.45e-02
n=9  a=-2.12500000 b=-2.12109375 f(x)=3.13e-02
n=10  a=-2.12500000 b=-2.12304688 f(x)=1.47e-02
n=11  a=-2.12500000 b=-2.12402344 f(x)=6.37e-03
n=12  a=-2.12500000 b=-2.12451172 f(x)=2.21e-03
n=13  a=-2.12500000 b=-2.12475586 f(x)=1.28e-04
n=14  a=-2.12500000 b=-2.12487793 f(x)=-9.13e-04
n=15  a=-2.12493896 b=-2.12487793 f(x)=-3.93e-04
n=16  a=-2.12490845 b=-2.12487793 f(x)=-1.32e-04
n=17  a=-2.12489319 b=-2.12487793 f(x)=-2.37e-06

Iteration for root on [1,2] :
n=1  a=1.00000000 b=2.00000000 f(x)=-6.25e-01
n=2  a=1.00000000 b=1.50000000 f(x)=5.78e-01
n=3  a=1.25000000 b=1.50000000 f(x)=-5.66e-02
n=4  a=1.25000000 b=1.37500000 f(x)=2.53e-01
n=5  a=1.31250000 b=1.37500000 f(x)=9.63e-02
n=6  a=1.34375000 b=1.37500000 f(x)=1.93e-02
n=7  a=1.35937500 b=1.37500000 f(x)=-1.88e-02
n=8  a=1.35937500 b=1.36718750 f(x)=2.29e-04
n=9  a=1.36328125 b=1.36718750 f(x)=-9.29e-03
n=10  a=1.36328125 b=1.36523438 f(x)=-4.53e-03
n=11  a=1.36328125 b=1.36425781 f(x)=-2.15e-03
n=12  a=1.36328125 b=1.36376953 f(x)=-9.61e-04
n=13  a=1.36328125 b=1.36352539 f(x)=-3.66e-04
n=14  a=1.36328125 b=1.36340332 f(x)=-6.85e-05
n=15  a=1.36328125 b=1.36334229 f(x)=8.03e-05
n=16  a=1.36331177 b=1.36334229 f(x)=5.91e-06
n=17  a=1.36332703 b=1.36334229 f(x)=-3.13e-05

Iteration for root on [2,3] :
n=1  a=2.00000000 b=3.00000000 f(x)=-1.38e+00
n=2  a=2.50000000 b=3.00000000 f(x)=-7.81e-02
n=3  a=2.75000000 b=3.00000000 f(x)=8.57e-01
n=4  a=2.75000000 b=2.87500000 f(x)=3.65e-01
n=5  a=2.75000000 b=2.81250000 f(x)=1.37e-01
n=6  a=2.75000000 b=2.78125000 f(x)=2.79e-02
n=7  a=2.75000000 b=2.76562500 f(x)=-2.55e-02
n=8  a=2.75781250 b=2.76562500 f(x)=1.10e-03
n=9  a=2.75781250 b=2.76171875 f(x)=-1.22e-02
n=10  a=2.75976562 b=2.76171875 f(x)=-5.56e-03
n=11  a=2.76074219 b=2.76171875 f(x)=-2.23e-03
n=12  a=2.76123047 b=2.76171875 f(x)=-5.64e-04
n=13  a=2.76147461 b=2.76171875 f(x)=2.70e-04
n=14  a=2.76147461 b=2.76159668 f(x)=-1.47e-04
n=15  a=2.76153564 b=2.76159668 f(x)=6.14e-05
n=16  a=2.76153564 b=2.76156616 f(x)=-4.29e-05
n=17  a=2.76155090 b=2.76156616 f(x)=9.23e-06
```

$\Rightarrow$ three roots of $P(x)$ using bisection method

$$\begin{cases} x_1 \approx -2.1249 \\ x_2 \approx 1.3633 \\ x_3 \approx 2.7616 \end{cases}$$

```
function x=myFalsePosition(f,xspan,tol,nMax)
% Find a root x of the equation f(x)=0 using the
% method of false position.  The initial interval is xspan=[a,b]
% and the iteration stops when abs(x(n+1)-x(n))<tol.
a=xspan(1);
b=xspan(2);
fa=feval(f,a);
if fa==0
  x=a; return
end
fb=feval(f,b);
if fb==0
  x=b; return
end
if fa*fb>0
  error('Error : f(a)*f(b)>0')
end
for n=1:nMax
    x1=a-fa*(b-a)/(fb-fa);
    fx=feval(f,x1);
    fprintf('n=%d x=%1.13f f(x)=%1.2e\n',n,x1,fx)
    if fx==0
        x=x1;
        return
    end
    if fx*fa<0
        b=x1; fb=fx;
    else
        a=x1; fa=fx;
    end
    if n>1
        if abs(x1-x)<tol
            x=x1;
            return
        end
    end
    x=x1;
end
```

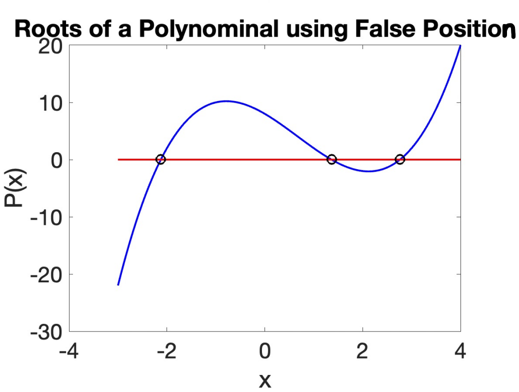
⇒ myFalsePosition.m

Instead of using the midpoint of the interval to update the interval endpoints, use the zero of the line connecting the points $(a, P(a))$ & $(b, P(b))$.

⇒ three roots of $P(x)$ using the method of false position

$$\begin{cases} x_1 \approx -2.1249 \\ x_2 \approx 1.3633 \\ x_3 \approx 2.7616 \end{cases}$$

```
Iteration for root on [-3,-2] :
n=1 x=-2.0833333333333 f(x)=6.94e-01
n=2 x=-2.1113604488079 f(x)=2.29e-01
n=3 x=-2.1205152088336 f(x)=7.43e-02
n=4 x=-2.1234766700117 f(x)=2.40e-02
n=5 x=-2.1244316547919 f(x)=7.73e-03
n=6 x=-2.1247392961840 f(x)=2.49e-03
n=7 x=-2.1248383681071 f(x)=8.02e-04
n=8 x=-2.1248702695645 f(x)=2.58e-04
n=9 x=-2.1248805415802 f(x)=8.31e-05
n=10 x=-2.1248838490514 f(x)=2.68e-05

Iteration for root on [1,2] :
n=1 x=1.5000000000000 f(x)=-6.25e-01
n=2 x=1.3809523809524 f(x)=-8.53e-02
n=3 x=1.3653686826843 f(x)=-9.94e-03
n=4 x=1.3635612027761 f(x)=-1.14e-03
n=5 x=1.3633547934626 f(x)=-1.30e-04
n=6 x=1.3633312644246 f(x)=-1.48e-05
n=7 x=1.3633285828500 f(x)=-1.68e-06
Iteration for root on [2,3] :
n=1 x=2.5000000000000 f(x)=-1.38e+00
n=2 x=2.7037037037037 f(x)=-3.74e-01
n=3 x=2.7504279356385 f(x)=-7.53e-02
n=4 x=2.7594789862117 f(x)=-1.42e-02
n=5 x=2.7611713093732 f(x)=-2.64e-03
n=6 x=2.7614856100351 f(x)=-4.89e-04
n=7 x=2.7615439092682 f(x)=-9.07e-05
n=8 x=2.7615547206036 f(x)=-1.68e-05
n=9 x=2.7615567254314 f(x)=-3.12e-06
```




```
% clear all variables and figures.
clear all
close all

% Set defaults for plotting
fontSize=24; lineWidth=2; markerSize=10;
set(0,'DefaultLineMarkerSize',markerSize);
set(0,'DefaultLineLineWidth',lineWidth);
set(0,'DefaultAxesFontSize',fontSize);
set(0,'DefaultLegendFontSize',fontSize);

% define and plot the polynomial y=x.^3-2*x.^2-5*x+8
P = @(x) x.^3-2*x.^2-5*x+8;
x=linspace(-3,4,300);
y=P(x);
plot(x,y,'b',x,zeros(size(x)),'r')

% Root on [-3,-2]
fprintf('Iteration for root on [-3,-2] :\n')
xspan=[-3,-2]; tol=10e-6; nMax=100;
x1=myFalsePosition(P,xspan,tol,nMax);

% Root on [1,2]
fprintf('Iteration for root on [1,2] :\n')
xspan=[1,2]; tol=10e-6; nMax=100;
x2=myFalsePosition(P,xspan,tol,nMax);

% Root on [2,3]
fprintf('Iteration for root on [2,3] :\n')
xspan=[2,3]; tol=10e-6; nMax=100;
x3=myFalsePosition(P,xspan,tol,nMax);

% plot roots
hold on
roots=[x1 x2 x3];
plot(roots,zeros(3,1),'ko')
hold off
xlabel('x')
ylabel('P(x)')
title('Roots of a Polynominal using False Position')
```

2. Find all fixed points of the following g(x):

$$(a)\quad \frac{x+6}{3x-2}; \qquad (b)\quad \frac{8+2x}{2+x^2}; \qquad (c)\quad x^5.$$

(a) We have $g_1(x) = \dfrac{x+6}{3x-2}$. The fixed points of $g_1(x)$ are the solutions of $g_1(x) = x \Rightarrow \dfrac{x+6}{3x-2} = x$

$$x + 6 = 3x^2 - 2x$$

$$3x^2 - 3x - 6 = 0$$

$$x^2 - x - 2 = (x-2)(x+1) = 0$$

$$\Rightarrow \boxed{x = -1, 2}$$

NOT Required

The fixed points are found to be -1, 2. Local convergence requires that $|g_1'(x)| < 1$ at the fixed point. And since $g_1'(x) = -\dfrac{20}{(3x-2)^2}$, we note that $g_1'(-1) = -\frac{4}{5}$, $g_1'(2) = -\frac{5}{4}$.

Thus, the fixed-point iteration is locally convergent at -1, but not at 2.

(b) We have $g_2(x) = \dfrac{8+2x}{2+x^2}$. The fixed points of $g_2(x)$ are the solutions of $g_2(x) = x \Rightarrow \dfrac{8+2x}{2+x^2} = x$

$$8 + 2x = 2x + x^3$$

$$x^3 - 8 = 0$$

NOT Required

therefore, $\boxed{x = 2}$ is a fixed point and the other two fixed points, $x = -1 \pm \sqrt{3}i$ are complex. Now since $g_2'(x) = \dfrac{-2x^2 - 16x + 4}{(2+x^2)^2}$, we note that $g_2'(2) = -1$, thus, the fixed-point iteration is not locally convergent at 2.

(c) We have $g_3(x) = x^5$. The fixed points of $g_3(x)$ are the solutions of $g_3(x) = x$ $\Rightarrow$

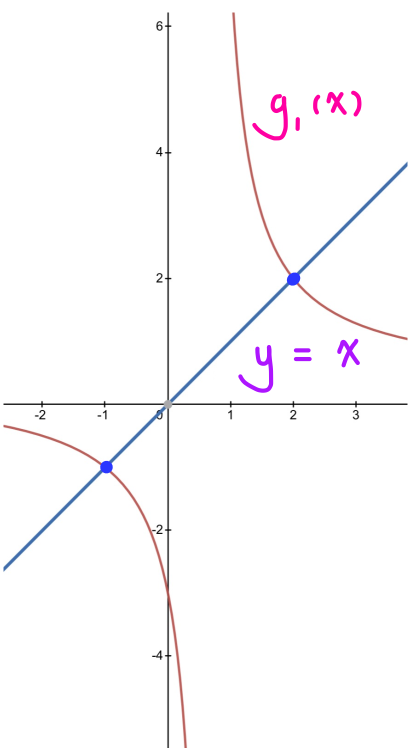
$$x^5 = x$$
$$x^5 - x = 0$$
$$x(x^4 - 1) = 0$$
$$x[(x^2+1)(x+1)(x-1)] = 0$$
$$\Rightarrow \boxed{x = 0, \ \pm 1}$$

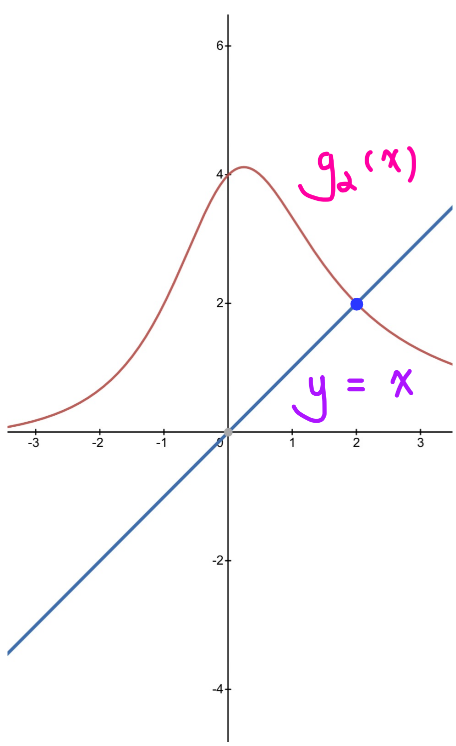
complex fixed points $\pm i$ omitted

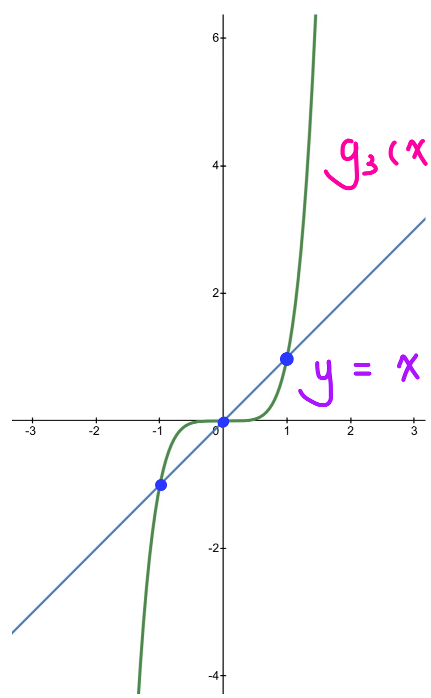
The fixed points are found to be $0, \pm 1$. Local convergence requires that $|g_3'(x)| < 1$ at the fixed point. And since $g_3'(x) = 5x^4$, we note that $g_3'(0) = 0$, $g_3'(\pm 1) = 5$.

NOT Required

Thus, the fixed-point iteration is locally convergent at 0, but not at ± 1.







3. Consider the fixed-point iteration $x_{n+1} = g(x_n)$, where $g(x) = x + \ln(2-x)$, and define the interval $I = [1/2, 4/3]$.

(a) Show that $g(x) \in I$ whenever $x \in I$, i.e., $g(x)$ maps the interval I into itself, and thus a fixed point of $g(x)$ exists for $x \in I$.

(b) Consider $g'(x)$ for $x \in I$ to show that the fixed-point iteration converges to the unique fixed point of $g(x)$ assuming x_0 is chosen in the interval I.

(a) Given $g(x) = x + \ln(2-x)$, we have $g'(x) = 1 - \frac{1}{2-x} = \frac{1-x}{2-x}$.

For $x \in I = [1/2, 4/3]$, we note that the denominator of $g'(x)$ is always positive. And the numerator is positive for $1/2 \le x < 1$, zero for $x = 1$, and negative for $1 < x < 4/3$.

It implies that $g(x)$ is increasing for $1/2 \le x < 1$ and decreasing for $1 < x < 4/3$. Note that $1/2 < g(1/2) \approx 0.9 < g(4/3) \approx 0.93 < g(1) = 1 < 4/3$, which follows that $g(x)$ remains in the desired interval for $x \in I = [1/2, 4/3]$.

Alternatively, to prove that $g(x)$ maps the interval I into itself, we can evaluate $g(x)$ at the endpoints of I and at $x = 1$ where $g'(x) = 0$. If the global extrema for I are all in I, then we can draw the conclusion that $g(x)$ maps I into I.

$$\text{At } x = \frac{1}{2}, \quad g(x) \approx 0.91 \in I.$$

$$\text{At } x = \frac{4}{3}, \quad g(x) \approx 0.93 \in I.$$

Setting $g'(x) = \frac{1-x}{2-x} = 0$, we have that $x = 1$, and that $g(1) = 1 \in I$. Therefore, we have shown that the global extrema for I are all in I, indicating that $g(x)$ remains in the desired interval for $x \in I = [1/2, 4/3]$.

(b) From above, we have $g'(x) = 1 - \frac{1}{2-x} = \frac{1-x}{2-x}$

$$g''(x) = \frac{-(2-x)+(1-x)}{(2-x)^2} = \frac{-1}{(2-x)^2}$$

indicating that $g''(x) < 0$ for all $x \in I = [\frac{1}{2}, \frac{4}{3}]$, and thus, $g'(x)$ is decreasing monotically in I, and the extrema occurs at the two endpoints. Now since $g'(\frac{1}{2}) = \frac{1}{3}$

$$g'(\tfrac{4}{3}) = -\tfrac{1}{2}$$

which follows that $|g'(x)| \le \frac{1}{2} < 1$.

4. Text exercise 16 on page 44.

$\Rightarrow$ Check if cube root of 4 is a fixed point for the following functions:

setting $g_1(x) = \frac{2}{\sqrt{x}} = x \Rightarrow 2 = x^{3/2} \Rightarrow 4 = x^3$

setting $g_2(x) = \frac{3x}{4} + \frac{1}{x^2} = x \Rightarrow \frac{1}{x^2} = \frac{x}{4} \Rightarrow 4 = x^3$

setting $g_3(x) = \frac{2}{3}x + \frac{4}{3x^2} = x \Rightarrow \frac{4}{3x^2} = \frac{1}{3}x \Rightarrow 4 = x^3$

$\Rightarrow$ Check the rate of convergence for the following functions:

$g_1'(x) = -x^{-3/2}$, at $x = \sqrt[3]{4}$, $g_1'(x) = -\frac{1}{2}$

$g_2'(x) = \frac{3}{4} - 2x^{-3}$, at $x = \sqrt[3]{4}$, $g_2'(x) = \frac{1}{4}$

$g_3'(x) = \frac{2}{3} - \frac{8}{3x^3}$, at $x = \sqrt[3]{4}$, $g_3'(x) = 0$

$\Rightarrow$ C, B, A converge from fastest to slowest.

5. Consider the functions

$$f_1(x) = x^3 - 2x - 5; \qquad f_2(x) = e^{-x} - x; \qquad f_3(x) = x\sin(x) - 1.$$

(a) Write a `Matlab` function to compute a root of the function $f(x)$ using Newton's method. Use your code to compute the smallest positive root of each of the functions $f_i(x), i = 1, 2, 3,$ above. You will need to determine a suitable starting guess, x_0 , for each case, and use the stopping criterion $|x_{n+1} - x_n| < 10^{-12}$.

(b) Repeat the calculations in part (a) using the secant method. (You may use the values for x_1 obtained in part (a) along with x_0 for the two starting values needed for the secant method.) Compare the number of iterations needed to compute each roots using the secant method versus the number needed in part (a) using Newton's method.

```
function xStar=myNewton(f,x0,tol,nMax)

% find the root of f(x) using Newton's method with starting
% guess x0. The iteration stops if abs(x(n+1)-x(x))<tol or
% n>nMax

for n=1:nMax
    x1=x0-feval(f,x0,0)/feval(f,x0,1);
    err=abs(x1-x0);
    if err>tol
        fprintf(' n=%d x(n)=%1.4f err=%1.2e', n,x1,err)
        x0=x1;
    else
        fprintf(' n=%d x(n)=%1.4f err=%1.2e (converged)', n,x1,err)
        xStar=x1;
        return
    end
end

n=nMax+1;
fprintf(' n=%d x(n)=%1.4f err=%1.2e (iteration failed)', n,x1,err)
xStar=x1;
```

```
function y=NewtonFunction1(x, derivative)

if derivative==0
    y=x^3-2*x-5;
else
    y=3*x^2-2;
end
```

```
function y=NewtonFunction2(x, derivative)

if derivative==0
    y=exp(-x)-x;
else
    y=-exp(-x)-1;
end
```

```
function y=NewtonFunction3(x, derivative)

if derivative==0
    y=x*sin(x)-1;
else
    y=sin(x)+x*cos(x);
end
```

```
% Use Newton's method to find the smallest positive root
% of the following functions
%
% f1=x^3-2*x-5, f2=exp(-x)-x, fx=x*sin(x)-1
%
% with tolerance 10e-12

x1=myNewton('NewtonFunction1',2.0,10e-12,10);
x2=myNewton('NewtonFunction2',0.5,10e-12,10);
x3=myNewton('NewtonFunction3',1.0,10e-12,10);

fprintf('Roots = %0.15e %0.15e %0.15e \n',x1,x2,x3)
```

```
 n=1 x(n)=2.1000 err=1.00e-01
 n=2 x(n)=2.0946 err=5.43e-03
 n=3 x(n)=2.0946 err=1.66e-05
 n=4 x(n)=2.0946 err=1.56e-10
 n=5 x(n)=2.0946 err=0.00e+00 (converged)
 n=1 x(n)=0.5663 err=6.63e-02
 n=2 x(n)=0.5671 err=8.32e-04
 n=3 x(n)=0.5671 err=1.25e-07
 n=4 x(n)=0.5671 err=2.89e-15 (converged)
 n=1 x(n)=1.1147 err=1.15e-01
 n=2 x(n)=1.1142 err=5.72e-04
 n=3 x(n)=1.1142 err=1.40e-08
 n=4 x(n)=1.1142 err=2.22e-16 (converged)
Roots = 2.094551481542327e+00 5.671432904097838e-01 1.114157140871930e+00
```

```
function xStar=mySecant(f,x0,x1,tol,nMax)

% find the root of f(x) using secant method with starting
% guesses x0 and x1. The iteration stops if abs(x(n+1)-x(n))<tol or
% n>nMax

n=0;
while n<nMax+1
    n=n+1;
    fx0=feval(f,x0);
    fx1=feval(f,x1);
    x=x1-fx1*(x1-x0)/(fx1-fx0);
    err=abs(x-x1);
    if err>tol
        fprintf(' n = %d x0 = %1.8f x1 = %1.8f err = %1.2d \n', n,x0,x1,err)
        x0=x1;
        x1=x;
    else
        fprintf(' n = %d x0 = %1.8f x1 = %1.8f err = %1.2d (converged) \n',
 n,x0,x1,err)
        xStar=x;
        return
    end
end
```

```
>> mySecant(@(x) x^3-2*x-5, 2.0, 2.0946, 10e-12, 10)
 n = 1 x0 = 2.00000000 x1 = 2.09460000 err = 5.12e-05
 n = 2 x0 = 2.09460000 x1 = 2.09454880 err = 2.68e-06
 n = 3 x0 = 2.09454880 x1 = 2.09455148 err = 7.33e-11
 n = 4 x0 = 2.09455148 x1 = 2.09455148 err = 00 (converged)

ans =

    2.0946

>> mySecant(@(x) exp(1).^(-x)-x, 0.5, 0.5671, 10e-12, 10)
 n = 1 x0 = 0.50000000 x1 = 0.56710000 err = 4.28e-05
 n = 2 x0 = 0.56710000 x1 = 0.56714276 err = 5.31e-07
 n = 3 x0 = 0.56714276 x1 = 0.56714329 err = 4.16e-12 (converged)

ans =

    0.5671

>> mySecant(@(x) x*sin(x)-1, 1.0, 1.1142, 10e-12, 10)
 n = 1 x0 = 1.00000000 x1 = 1.11420000 err = 4.29e-05
 n = 2 x0 = 1.11420000 x1 = 1.11415714 err = 3.57e-09
 n = 3 x0 = 1.11415714 x1 = 1.11415714 err = 6.66e-15 (converged)

ans =

    1.1142
```