110550126 曾家祐

Numerical methods Assignment 1

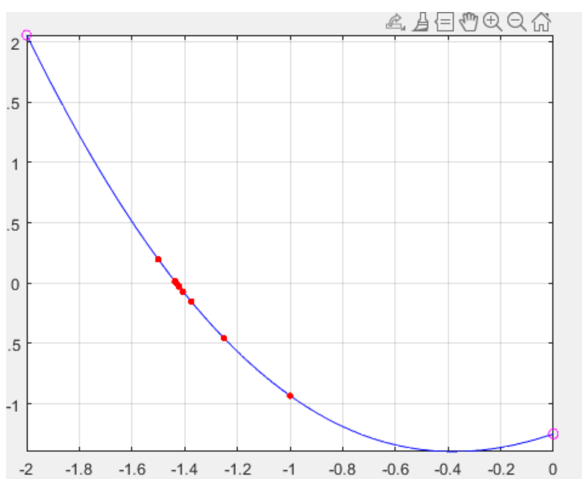
1. Find two roots for f(x)

a) Bisection method

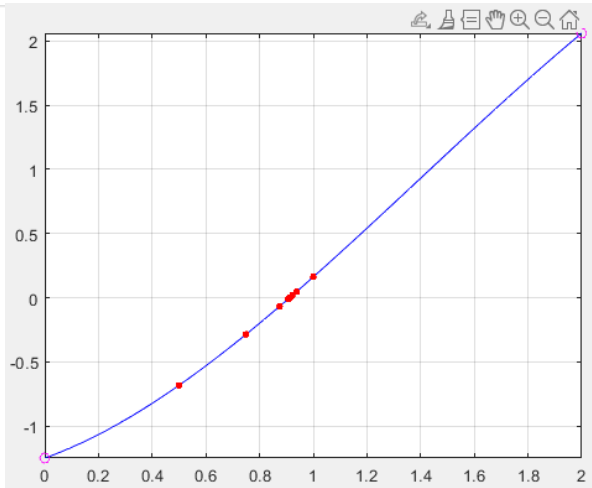
we use a while loop base on interval of L and R to decide how many iteration to run, every time we see the mid point and use new mid point to replace the old point and we can get the two roots is -1.431816 and 0.911919

```
f = @(x) x.^2 + sin(x) -exp(x)/4 -1;
tol = 1e-5;

fprintf("use bisection method to find root between (%d,%d)\n",L,R)

while abs(R-L) > tol
    n = n+1;
    mid = (R+L)/2;
    plot(mid,f(mid),'.r','MarkerSize',10)
    if f(L)*f(mid)<0
        R = mid;
    elseif f(mid)*f(R)<0
        L = mid;
    end
    fprintf('%d iteration: mid =  %.6f\n',n,mid)
end

fprintf('find root: at around %.6f',mid)
```

```
use bisection method to find root between (-2,0)
1 iteration: mid =  -1.000000
2 iteration: mid =  -1.500000
3 iteration: mid =  -1.250000
4 iteration: mid =  -1.375000
5 iteration: mid =  -1.437500
6 iteration: mid =  -1.406250
7 iteration: mid =  -1.421875
8 iteration: mid =  -1.429688
9 iteration: mid =  -1.433594
10 iteration: mid =  -1.431641
11 iteration: mid =  -1.432617
12 iteration: mid =  -1.432129
13 iteration: mid =  -1.431885
14 iteration: mid =  -1.431763
15 iteration: mid =  -1.431824
16 iteration: mid =  -1.431793
17 iteration: mid =  -1.431808
18 iteration: mid =  -1.431816
find root: at around -1.431816>>
```



```
use bisection method to find root between (0,2)
1 iteration: mid =  1.000000
2 iteration: mid =  0.500000
3 iteration: mid =  0.750000
4 iteration: mid =  0.875000
5 iteration: mid =  0.937500
6 iteration: mid =  0.906250
7 iteration: mid =  0.921875
8 iteration: mid =  0.914062
9 iteration: mid =  0.910156
10 iteration: mid =  0.912109
11 iteration: mid =  0.911133
12 iteration: mid =  0.911621
13 iteration: mid =  0.911865
14 iteration: mid =  0.911987
15 iteration: mid =  0.911926
16 iteration: mid =  0.911896
17 iteration: mid =  0.911911
18 iteration: mid =  0.911919
find root: at around 0.911919>>
```

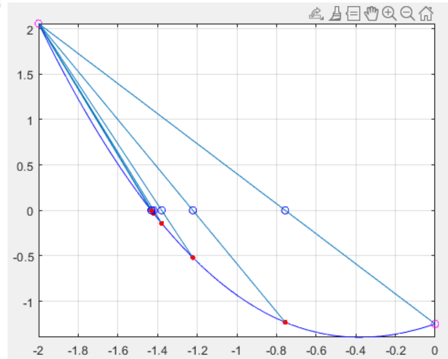


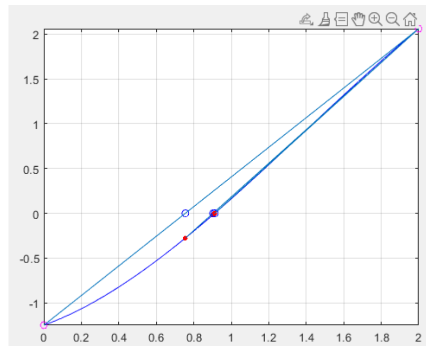
b) Secant method

we use a while loop base on f(new) to decide how many iteration to run, every time we get new point where f(new) = 0 and use new point to replace old point and we can get the two roots is -1.431807 and 0.911917

```
f = @(x) x.^2 + sin(x) -exp(x)/4 -1;
tol = 1e-5;

new = R-f(R)*(R-L)/(f(R)-f(L));
fprintf("use secant method to find root from %d,%d\n",L,R)
while abs(f(new)) > tol
    n = n+1;
    plot(new,f(new),'.r','MarkerSize',10)
    plot(new,0,'ob')
    line([L,R],[f(L),f(R)])
    if f(L)*f(new)<0
        R = new;
    else
        L = new;
    end
    fprintf('%d iteration: new =  %.6f\n',n,new)
    new = R-f(R)*(R-L)/(f(R)-f(L));
end
fprintf('find root: at around %.6f',new)
```

```
use secant method to find root from -2,0
1 iteration: new =  -0.756002
2 iteration: new =  -1.221968
3 iteration: new =  -1.379027
4 iteration: new =  -1.419369
5 iteration: new =  -1.428924
6 iteration: new =  -1.431142
7 iteration: new =  -1.431655
8 iteration: new =  -1.431773
9 iteration: new =  -1.431800
find root: at around -1.431807>>
```






```
use secant method to find root from 0,2
1 iteration: new =  0.754823
2 iteration: new =  0.902232
3 iteration: new =  0.911491
4 iteration: new =  0.911899
find root: at around 0.911917>>
```

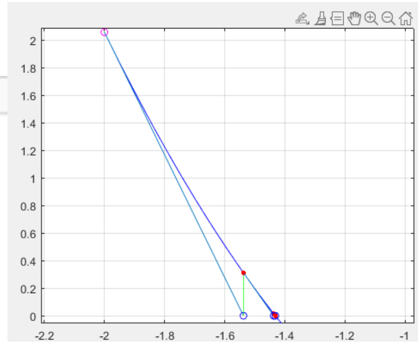
c) Newton's method

we use a while loop base on f(start) to decide how many iteration to run, every time we get new point where by the tangent line of old point and use new point to replace old point and we can get the two roots is -1.431809 and 0.911918

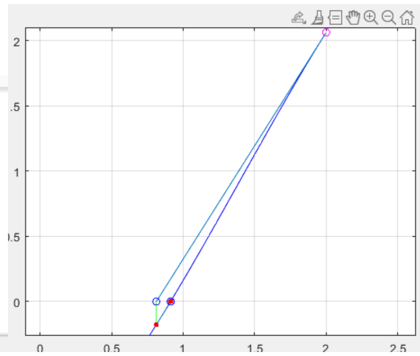
```
f = @(x) x.^2 + sin(x) -exp(x)/4 -1;
df = @(x) 2*x + cos(x) - exp(x)/4;
tol = 1e-5;
```

```
fprintf("use newton method to find root from %d\n",start);
while abs(f(start))>tol
    n = n+1;
    new = start-f(start)/df(start);
    line([new,start],[0,f(start)])
    plot([new,new],[0,f(new)],'-g')
    plot(new,f(new),'.r','MarkerSize',10)
    plot(new,0,'ob')
    start = new;
    fprintf('%d iteration: from =  %.6f\n',n,start)
end
fprintf('find root: at around %.6f',start)
```

```
Command Window
use newton method to find root from -2
1 iteration: from =  -1.537780
2 iteration: from =  -1.437143
3 iteration: from =  -1.431823
4 iteration: from =  -1.431809
fx find root: at around -1.431809>>
```




```
use newton method to find root from 2
1 iteration: from =  0.812596
2 iteration: from =  0.913804
3 iteration: from =  0.911918
fx find root: at around 0.911918>>
```

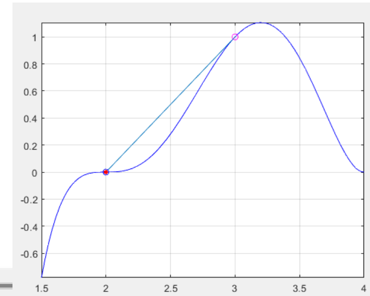


2. Use Newton's method on the polynomial P(x), with X0 =3. I
   It will coverage to root at x = 2.
   It is not convergence quadratic. If we start from x0 = 2.95, it takes 7 iterations to converge to R = 2. Since R = 2 is not simple root, it is triple root. f'(2) = 0 which cause f(R)f''(R)/f'(R)^2 can't be omitted make the convergence only linearly not quadratic.

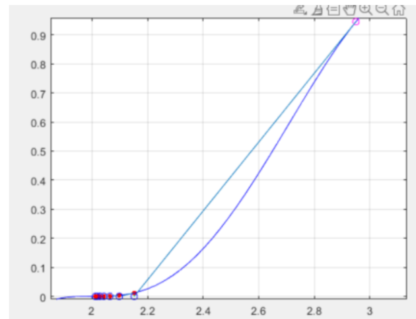
```
f = @(x) ((x-2)^3)*(x-4)^2;
df = @(x) 3*((x-2)^2)*((x-4)^2)+2*((x-2)^3)*(x-4);
tol = 1e-5;
```

```
fprintf("use newton method to find root from %d\n",start);
while abs(f(start))>tol
    n = n+1;
    new = start-f(start)/df(start);
    line([new,start],[0,f(start)])
    plot([new,new],[0,f(new)],'-g')
    plot(new,f(new),'.r','MarkerSize',10)
    plot(new,0,'ob')
    start = new;
    fprintf('%d iteration: from =  %.6f\n',n,start)
end
fprintf('find root: at around %.6f',start)
```

```
use newton method to find root from 3
1 iteration: from =  2.000000
find root: at around 2.000000>>
```




```
use newton method to find root from 2.950000e+00
1 iteration: from =  2.152000
2 iteration: from =  2.098394
3 iteration: from =  2.064424
4 iteration: from =  2.042462
5 iteration: from =  2.028100
6 iteration: from =  2.018644
7 iteration: from =  2.012390
find root: at around 2.012390>>
```



3. Below are three different g(x) functions. All are rearrangements of the same f(x). What is f(x)?

①
$f(x)=0$
$\Rightarrow g(x)=x$
$\Rightarrow \frac{4+2x^3}{x^2}-2x=x$
$\Rightarrow (4+2x^3)/x^2=3x$
$\Rightarrow 4+2x^3=3x^3$
$\Rightarrow 4-x^3=0$
$\Rightarrow x^3-4=0$
$\Rightarrow f(x)=x^3-4$

②
$f(x)=0$
$\Rightarrow g(x)=x$
$\Rightarrow \sqrt{\frac{4}{x}}=x$
$\Rightarrow 4/x=x^2$
$\Rightarrow 4=x^3$
$\Rightarrow x^3-4=0$
$\Rightarrow f(x)=x^3-4$

③
$f(x)=0$
$\Rightarrow g(x)=x$
$\Rightarrow \frac{(16+x^3)}{5x^2}=x$
$\Rightarrow 16+x^3=5x^3$
$\Rightarrow 16-4x^3=0$
$\Rightarrow x^3-4=0$
$\Rightarrow f(x)=x^3-4$

$\Rightarrow$ the $f(x)$ is x^3-4

(a) $g(x) = \frac{4+2x^3}{x^2} - 2x$

$= \frac{4}{x^2}$

$\Rightarrow g'(x) = \frac{-8}{x^3}$

$\Rightarrow$ when $x > 2$ or $x < -2$

$|g'(x)| < 1$

but $g(x) = x$

$\Rightarrow 4/x^2 = x \Rightarrow x = \sqrt[3]{4}$

not in $(-\infty, -2), (2, \infty)$

$\Rightarrow g(x) = \frac{4+2x^3}{x^2} - x$ diverge

(b) $g(x) = \sqrt{\frac{4}{x}}$

$\Rightarrow g'(x) = -\frac{1}{2} \times 2 x^{-\frac{3}{2}}$

$= -1 \times x^{-\frac{3}{2}}$

$|g'(x)| < 1 \Rightarrow |-x^{-\frac{3}{2}}| < 1$

$\Rightarrow |x^{-\frac{3}{2}}| < 1$

$\Rightarrow |x| > 1$

$\Rightarrow$ when $x > 1$ or $x < -1$, $|g'(x)| < 1$

$g(x) = x \Rightarrow \sqrt{\frac{4}{x}} = x \Rightarrow x = \sqrt[3]{4}$

in $(1, \infty)$

$\Rightarrow g(x) = \sqrt{\frac{4}{x}} = x$ converge

converge to 1.587759

(c) $g(x) = \frac{16+x^3}{5x^2} = \frac{16}{5x^2} + \frac{x}{5}$

$g'(x) = \frac{-32}{5x^3} + \frac{1}{5}$

$|g'(x)| < 1 \Rightarrow \frac{6}{5} < \frac{-32}{5x^3} < \frac{4}{5}$

$\Rightarrow \frac{6}{32} > \frac{1}{x^3} > \frac{-4}{32}$

$\Rightarrow x > \sqrt[3]{\frac{32}{6}}$ or $x < -2$.

$\Rightarrow x > \sqrt[3]{5.333}$ or $x < -2$.

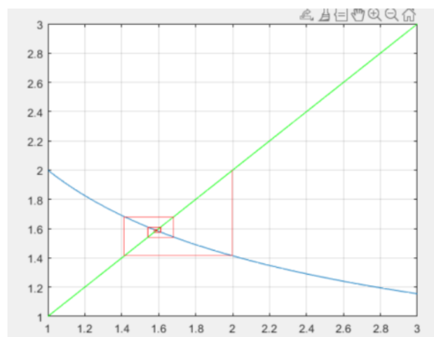
but $g(x) = x \Rightarrow x = \sqrt[3]{4}$

not in $(\sqrt[3]{5.333}, \infty), (-\infty, -2)$

$\Rightarrow g(x) = (16+x^3)/5x^2$ diverge

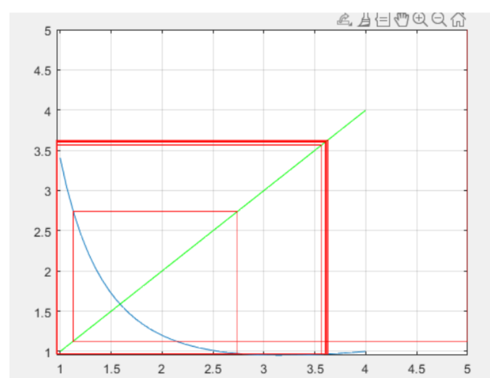
b)

```
use iterative method to find root from 2
1 iteration: start =  1.414214
2 iteration: start =  1.681793
3 iteration: start =  1.542211
4 iteration: start =  1.610490
5 iteration: start =  1.575981
6 iteration: start =  1.593142
7 iteration: start =  1.584538
8 iteration: start =  1.588834
9 iteration: start =  1.586685
10 iteration: start =  1.587759
find root: at around 1.587759>>
```



c)

```
20 iteration: start =  3.611170
21 iteration: start =  0.967622
22 iteration: start =  3.611257
23 iteration: start =  0.967628
24 iteration: start =  3.611219
25 iteration: start =  0.967626
26 iteration: start =  3.611236
27 iteration: start =  0.967627
28 iteration: start =  3.611228
29 iteration: start =  0.967626
30 iteration: start =  3.611232
find root: at around 3.611232>>
```



we can see 0.967 and 3.611 reflect to each other, so it is also diverge.

4. Solve the following system of nonlinear equations using Newton's method or fixed-point method.

I write a program to solve system of nonlinear equations using Newton's method. First, I write down the matrix of system of nonlinear equations(3x1) and derived jacobian matrix j(x)(3x3).

For every iteration we calculate the s (j*s = -f), and update the

new start point = old start point +s. Use while loop to iterate until the norm of s (the value need to update) less than tolerant error.

From 6 starting point, we get 6 solutions (4 reals,2virtual solution)

(1, 1, 1)-> (1.111408, 0.988210, 1.070878)

(1.3, 0.9, -1.2)->(1.353748, 0.925431, -1.255968)

(100, 100, 100)->(32.884631, -4434.086620, 115.490885)

(50, -50, 50)->(31.151405, -3768.157126, -106.482969)

(0+10i, 0+1i, 0+1i)->

(-1.250596+0.048995i,0.665053-0.013410i,0.088588+0.503590i)

(0-10i, 1+0i, 1+0i)->

(-1.250596-0.048995i,0.665053+0.013410i,0.088588-0.503590i)

```
f = @(x)[x(1)-3*x(2)-x(3)^2+3;
            2*x(1)^3+x(2)-5*x(3)^2+2;
            4*x(1)^2+x(2)+x(3)-7;];
j = @(x)[1,-3,-2*x(3);
            6*x(1)^2,1,-10*x(3);
            8*x(1),1,1;];
tol = 1e-5;
n = 0;
%start = [1;1;1];
%start = [1.3;0.9;-1.2];
%start = [100;100;100];
%start = [50,-50,-50];
%start = [10i;1i;1i];
start = [-10i;1i;1i];
fprintf("solving system of nonlinear equations from %.6f+%.6fi,%.6f+%.6fi,%.6f+%.6fi\n",real(start(1)),imag(s
while true
    a = f(start);
    b = j(start);
    %b*s = -a
    s = -b\a;
    %new = old +s
    start = start+s;
    n =n+1;
    fprintf("%d iteration: start =   %.6f+%.6fi,%.6f+%.6fi,%.6f+%.6fi\n",n,real(start(1)),imag(start(1)),real
    %f(start)
    if norm(s)<tol
        break
    end
end
fprintf("find solution at  %.6f+%.6fi,%.6f+%.6fi,%.6f+%.6fi \n",real(start(1)),imag(start(1)),real(start(2)),
```

```
solving system of nonlinear equations from 1.000000 1.000000 1.000000
1 iteration: start =  1.117216, 0.992674, 1.069597
2 iteration: start =  1.111423, 0.988222, 1.070869
3 iteration: start =  1.111408, 0.988210, 1.070878
4 iteration: start =  1.111408, 0.988210, 1.070878
find solution at 1.111408, 0.988210, 1.070878
```

```
solving system of nonlinear equations from 1.300000 0.900000 -1.200000
1 iteration: start =  1.354790, 0.926524, -1.256341
2 iteration: start =  1.353749, 0.925431, -1.255968
3 iteration: start =  1.353748, 0.925431, -1.255968
find solution at 1.353748, 0.925431, -1.255968
```

```
solving system of nonlinear equations from 100.000000 100.000000 100.000000
1 iteration: start =  71.297147, -17341.206862, 310.489589
2 iteration: start =  52.765133, -9959.343811, 203.448982
3 iteration: start =  41.396942, -6480.497657, 149.613376
4 iteration: start =  35.350260, -4970.078144, 124.764068
5 iteration: start =  33.183724, -4495.437060, 116.574299
6 iteration: start =  32.889867, -4435.137572, 115.509465
7 iteration: start =  32.884632, -4434.086950, 115.490891
8 iteration: start =  32.884631, -4434.086620, 115.490885
9 iteration: start =  32.884631, -4434.086620, 115.490885
find solution at 32.884631, -4434.086620, 115.490885
```

```
solving system of nonlinear equations from 50.000000 -50.000000 -50.000000
1 iteration: start =  39.192982, -5380.359891, -89.832727
2 iteration: start =  33.506129, -4238.390571, -115.890931
3 iteration: start =  31.444639, -3823.480291, -107.582160
4 iteration: start =  31.156847, -3769.162549, -106.502629
5 iteration: start =  31.151407, -3768.157481, -106.482976
6 iteration: start =  31.151405, -3768.157126, -106.482969
7 iteration: start =  31.151405, -3768.157126, -106.482969
find solution at 31.151405, -3768.157126, -106.482969
```

```
solving system of nonlinear equations from 0.000000+10.000000i,0.000000+1.000000i,0.000000+1.000000i
1 iteration: start =   0.247890+5.712766i,10.185846+-33.986035i,53.835435+14.154824i
2 iteration: start =   0.066614+3.271190i,-0.942363+-5.436531i,27.013640+7.234049i
3 iteration: start =   -0.091087+1.891249i,0.193226+-0.548231i,13.563401+3.667320i
4 iteration: start =   -0.162053+0.983249i,0.662125+-0.077287i,6.822240+1.867492i
5 iteration: start =   -0.240111+0.162709i,0.749773+-0.138741i,3.456736+0.963689i
6 iteration: start =   -1.915476+-1.082302i,0.269329+-0.375732i,1.767163+0.477523i
7 iteration: start =   -1.221019+-0.386290i,0.899767+-0.521732i,0.724907+0.615197i
8 iteration: start =   -1.158480+0.024427i,0.723526+-0.012369i,0.251452+0.444244i
9 iteration: start =   -1.255478+0.047282i,0.662560+-0.011072i,0.077024+0.468233i
10 iteration: start =   -1.250522+0.049102i,0.665017+-0.013444i,0.089495+0.504738i
11 iteration: start =   -1.250596+0.048995i,0.665053+-0.013410i,0.088590+0.503590i
12 iteration: start =   -1.250596+0.048995i,0.665053+-0.013410i,0.088588+0.503590i
find solution at  -1.250596+0.048995i,0.665053+-0.013410i,0.088588+0.503590i
```

```
solving system of nonlinear equations from -0.000000+-10.000000i,1.000000+0.000000i,1.000000+0.000000i
1 iteration: start =   -0.406157+-5.025870i,-14.545323+59.959299i,23.614906+-92.451883i
2 iteration: start =   -1.146942+-2.549584i,-6.564735+8.082335i,11.971441+-46.151297i
3 iteration: start =   -1.493679+-1.600406i,-0.853978+1.291910i,6.052021+-23.048771i
4 iteration: start =   -1.356078+-1.086084i,0.309781+0.304184i,3.070378+-11.520534i
5 iteration: start =   -1.203672+-0.643869i,0.599014+0.109928i,1.574636+-5.770811i
6 iteration: start =   -1.169820+-0.312801i,0.657100+0.070624i,0.826522+-2.908330i
7 iteration: start =   -1.211472+-0.144111i,0.657244+0.042715i,0.448287+-1.495615i
8 iteration: start =   -1.235093+-0.080301i,0.660525+0.022317i,0.249393+-0.827810i
9 iteration: start =   -1.245567+-0.054632i,0.663659+0.014956i,0.140332+-0.561489i
10 iteration: start =   -1.250087+-0.049100i,0.664913+0.013439i,0.093824+-0.504672i
11 iteration: start =   -1.250595+-0.048992i,0.665053+0.013409i,0.088599+-0.503564i
12 iteration: start =   -1.250596+-0.048995i,0.665053+0.013410i,0.088588+-0.503590i
13 iteration: start =   -1.250596+-0.048995i,0.665053+0.013410i,0.088588+-0.503590i
find solution at  -1.250596+-0.048995i,0.665053+0.013410i,0.088588+-0.503590i
```

(if +- please regard as -)