**Question1:**

**Here is my newton’s method:**

function [ xn ] = newton(f,fp,x0,Nmax,tol)

%UNTITLED Summary of this function goes here

%   Detailed explanation goes here

Results = [];

k = 0;

xn = x0;

fx = feval(f,xn);

finished = 0;

while k <= Nmax

    Results = [Results; k, xn, fx];

    if(abs(fx) < tol)

        finished = 1;

        break;

    end

    deri = feval(fp,xn);

    xn = xn - fx/deri;

    fx = feval(f,xn);

    k = k + 1;

end

if finished == 1

else

    disp('newton failed to converge after maximum iterations');

end

disp(Results);

end

**Here is my question1 script:**

format long e;

f = @(x) exp(x) - 1.5 - atan(x);

fp = @(x) exp(x) - (1 / (x^2 + 1));

%fpp = @(x) exp(x) + (2\*x / (x^2 + 1)^2);

%x = -1:0.001:1;

%plot(x,fp(x));

%f = @(x) x^2 - 4\*x + 1;

%fp = @(x) 2\*x - 4;

x0 = -7;

Nmax = 30;

tol = 10e-010;

newton(f,fp,x0,Nmax,tol);

**Here is the result:**

question1

0 -7.000000000000000e+00 -7.018884584371277e-02

1.000000000000000e+00 -1.067709617664001e+01 -2.256660810987587e-02

2.000000000000000e+00 -1.327916737563271e+01 -4.366019333912563e-03

3.000000000000000e+00 -1.405365585426924e+01 -2.390197770529845e-04

4.000000000000000e+00 -1.410110995686641e+01 -7.995848123609761e-07

5.000000000000000e+00 -1.410126977093942e+01 -9.008349621808520e-12

**as we can see, the newton method converge relatively fast for this function, in 5 steps we can a result within the error boundry.**

**Question2:**

**My modified newton’s method:**

function [ xn ] = modifiedNewton(f,fp,x0,Nmax,tol)

%UNTITLED Summary of this function goes here

%   Detailed explanation goes here

Results = [];

k = 0;

xn = x0;

fx = feval(f,xn);

finished = 0;

err = 1000000;

while k <= Nmax

    Results = [Results; k, xn, fx];

    if(err <= tol)

        finished = 1;

        break;

    end

    deri = feval(fp,xn);

    temp = xn;

    xn = xn - fx/deri;

    err = abs(xn-temp)/abs(temp);

    %disp(err);

    fx = feval(f,xn);

    k = k + 1;

end

if finished == 1

else

    disp('newton failed to converge after maximum iterations');

end

disp(Results);

end

**my question2 script:**

format long e;

x = -50:0.01:12;

f = @(x)3.^(3.\*x+1) - 7\*5.^(2.\*x);

plot(x,f(x));

fp = @(x) log(3) \* 3.^(3\*x+2) - 14 \* log(5) \* 5.^(2\*x);

disp(fzero(f, X));

x0 = 12;

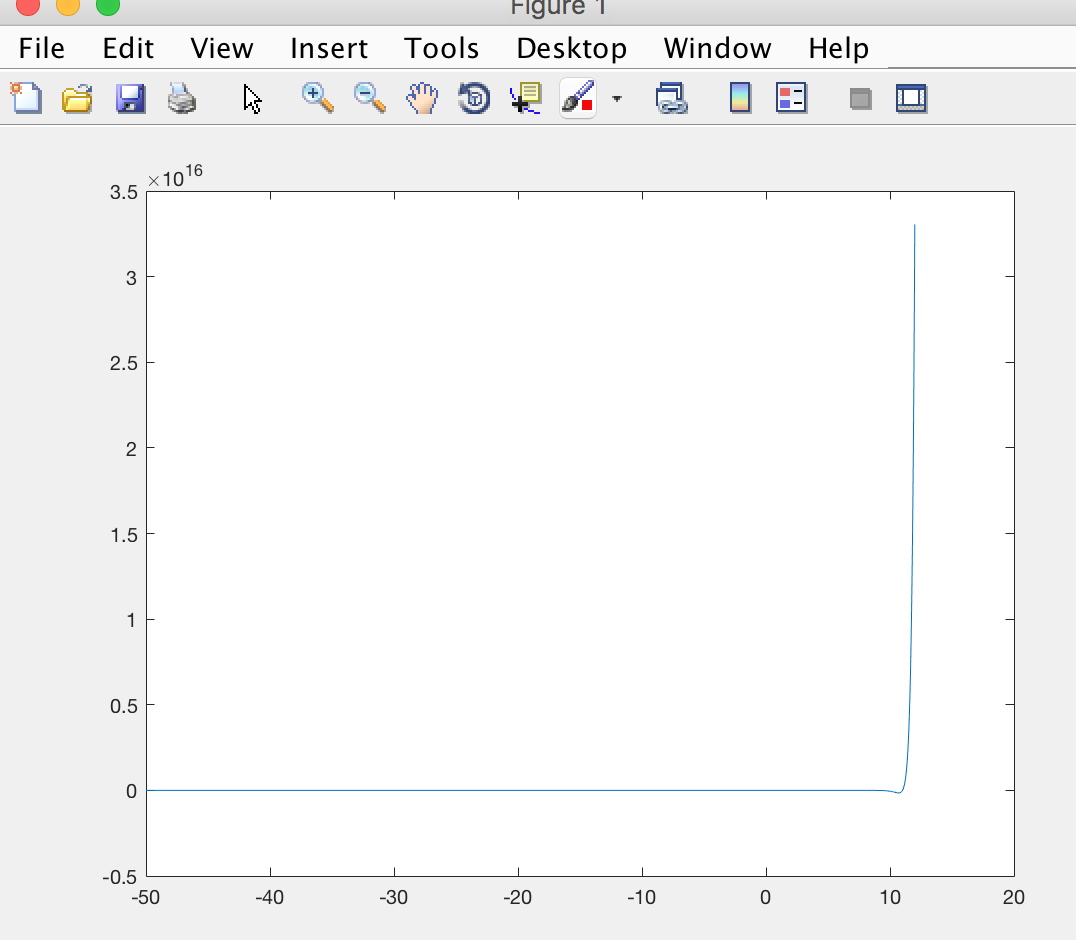
Nmax = 30;

tol = 10e-010;

modifiedNewton(f,fp,x0,Nmax,tol);

x0 = 0;

modifiedNewton(f,fp,x0,Nmax,tol);



**ok, I tried 2 initial points X0 = 0 and X0 = 11:**

**the real roots of fx I used matlab get is: 1.100943864426816e+01**

**question2**

**1.100943864426816e+01**

**for X0 = 11.**

**0 1.200000000000000e+01 3.305139246326298e+16**

**1.000000000000000e+00 1.176566383777540e+01 1.176011472600317e+16**

**2.000000000000000e+00 1.154732737434517e+01 4.107262775850992e+15**

**3.000000000000000e+00 1.135189052750941e+01 1.383458254379752e+15**

**4.000000000000000e+00 1.119001650913931e+01 4.305529348512520e+14**

**5.000000000000000e+00 1.107631842255931e+01 1.101059115599400e+14**

**6.000000000000000e+00 1.102140231031581e+01 1.646995633863600e+13**

**7.000000000000000e+00 1.100988738099651e+01 5.950180655080000e+11**

**8.000000000000000e+00 1.100943929922726e+01 8.671990540000000e+08**

**9.000000000000000e+00 1.100943864426962e+01 1.922000000000000e+03**

**1.000000000000000e+01 1.100943864426817e+01 -3.800000000000000e+01**

**for X0 = 0.**

**newton failed to converge after maximum iterations**

**0 0 -4.000000000000000e+00**

**1.000000000000000e+00 -3.163400674993609e-01 -1.470956858062820e+00**

**2.000000000000000e+00 -6.324414991104379e-01 -5.409407441538099e-01**

**3.000000000000000e+00 -9.483185437682776e-01 -1.989337437213553e-01**

**4.000000000000000e+00 -1.263984262839210e+00 -7.316030302169507e-02**

**5.000000000000000e+00 -1.579450658871791e+00 -2.690606366001003e-02**

**6.000000000000000e+00 -1.894728787278878e+00 -9.895366468727891e-03**

**7.000000000000000e+00 -2.209828853605346e+00 -3.639318983868468e-03**

**8.000000000000000e+00 -2.524760298565208e+00 -1.338487785511049e-03**

**9.000000000000000e+00 -2.839531872654379e+00 -4.922824146995941e-04**

**1.000000000000000e+01 -3.154151701840842e+00 -1.810587272213144e-04**

**1.100000000000000e+01 -3.468627345586790e+00 -6.659313810969736e-05**

**1.200000000000000e+01 -3.782965848255746e+00 -2.449311674825114e-05**

**1.300000000000000e+01 -4.097173784792322e+00 -9.008715731774333e-06**

**1.400000000000000e+01 -4.411257301426000e+00 -3.313490288322016e-06**

**1.500000000000000e+01 -4.725222152037508e+00 -1.218743357198252e-06**

**1.600000000000000e+01 -5.039073730732554e+00 -4.482727442140698e-07**

**1.700000000000000e+01 -5.352817101089277e+00 -1.648829347357871e-07**

**1.800000000000000e+01 -5.666457022480094e+00 -6.064740083809101e-08**

**1.900000000000000e+01 -5.979997973813274e+00 -2.230753626777531e-08**

**2.000000000000000e+01 -6.293444174992850e+00 -8.205287210776726e-09**

**2.100000000000000e+01 -6.606799606355814e+00 -3.018134421537410e-09**

**2.200000000000000e+01 -6.920068026311837e+00 -1.110160673622692e-09**

**2.300000000000000e+01 -7.233252987381951e+00 -4.083527043290904e-10**

**2.400000000000000e+01 -7.546357850808000e+00 -1.502059882934635e-10**

**2.500000000000000e+01 -7.859385799883469e+00 -5.525112905589737e-11**

**2.600000000000000e+01 -8.172339852138082e+00 -2.032343249908104e-11**

**2.700000000000000e+01 -8.485222870492798e+00 -7.475751446894364e-12**

**2.800000000000000e+01 -8.798037573488205e+00 -2.749884456621542e-12**

**2.900000000000000e+01 -9.110786544677447e+00 -1.011522971163932e-12**

**3.000000000000000e+01 -9.423472241264518e+00 -3.720820182951077e-13**

**as we can see, when I chose x0 = 11 or anything above 11, my newton’s method will eventually converge in approximately 10 steps, which is contributed by we changed the stop condition, for this function, after the root, fx just converge to infinity extremely fast, it will not be possible to stop for our previous stop condition. And before the root, the function is converge to 0 extremely slowly, that’s if I chose x0 < 11, I will not converge to root and another reason is newton’s method will have problem in case when derivative of fx is also close to 0 where x is near the root, since newtom’s method rely on fx/fx prime, and if derivative is zero we will have problem.**

**Question3:**

**My F.M:**

function [ Y ] = F( X0 )

%UNTITLED5 Summary of this function goes here

%   Detailed explanation goes here

x = X0(1,1);

y = X0(1,2);

Y = zeros(2,1);

Y(1,1) = 7 \* x^3 - 10 \* x - y - 1;

Y(2,1) = 8 \* y^3 - 11\*y + x - 1;

end

**MY JF.M:**

function [ A ] = JF( X )

%UNTITLED6 Summary of this function goes here

%   Detailed explanation goes here

x = X(1,1);

y = X(1,2);

A = zeros(2,2);

A(1,1) = 21 \* x^2 - 10;

A(1,2) = -1;

A(2,1) = 1;

A(2,2) = 24 \* y^2 - 11;

end

**MY Newton1.m:**

function [ X0 ] = Newton1(X0,Nmax,tol)

%UNTITLED7 Summary of this function goes here

%   Detailed explanation goes here

Results = [];

k = 0;

max = 1000;

finished = 0;

while(k <= Nmax)

    if(max < tol)

        finished = 1;

        break;

    end

    x = X0(1,1);

    y = X0(1,2);

    %disp(X0);

    Y = F(X0);

    A = JF(X0);

    Y = -Y;

    H = A\Y;

    X0 = X0 + transpose(H);

    x1 = Y(1,1);

    y1 = Y(2,1);

    if abs(x1) >= abs(y1)

        max = abs(x1);

    else

        max = abs(y1);

    end

    Results = [Results; k, x, y, max];

    k = k + 1;

end

if finished == 1

else

    disp('newton failed to converge after maximum iterations');

end

disp(Results);

end

**This is my question3 part a,b,c script:**

format long e;

x0 = [1,-2];

Nmax = 30;

tol = 10e-010;

s = Newton1(x0,Nmax,tol);

disp(s);

x0 = [2,2];

s = Newton1(x0,Nmax,tol);

disp(s);

This is the result when my x0 = (1,-2):

0 1.000000000000000e+00 -2.000000000000000e+00 4.200000000000000e+01

1.000000000000000e+00 1.226495726495727e+00 -1.508547008547009e+00 1.064366048477379e+01

2.000000000000000e+00 1.184177922607375e+00 -1.263552063534245e+00 2.055481425436459e+00

3.000000000000000e+00 1.185699651804731e+00 -1.188363729345022e+00 1.680368073844190e-01

4.000000000000000e+00 1.186071829286016e+00 -1.181039891210784e+00 1.526669420976123e-03

5.000000000000000e+00 1.186075121001791e+00 -1.180972114822810e+00 1.302040031347929e-07

6.000000000000000e+00 1.186075121283819e+00 -1.180972109041481e+00 2.664535259100376e-15

This is the result when my x0 = (2,2);

0 2.000000000000000e+00 2.000000000000000e+00 4.300000000000000e+01

1.000000000000000e+00 1.547289779049436e+00 1.499443649658242e+00 1.102337785590150e+01

2.000000000000000e+00 1.343444013089628e+00 1.247592052099088e+00 2.154807228413904e+00

3.000000000000000e+00 1.294314708173645e+00 1.167697344810652e+00 1.870461914176706e-01

4.000000000000000e+00 1.291306908316150e+00 1.159225842459507e+00 2.006369340084024e-03

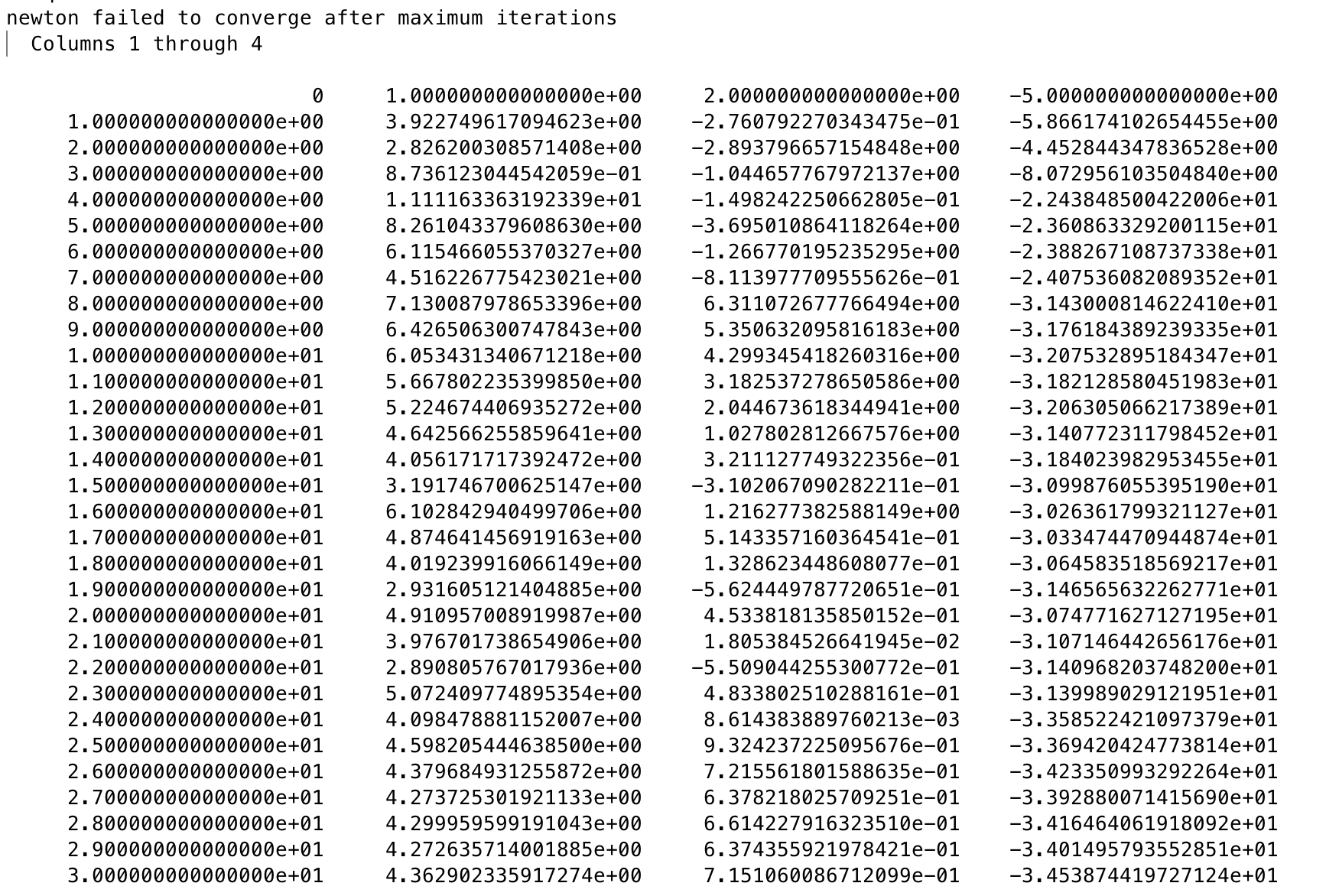
5.000000000000000e+00 1.291293338245643e+00 1.159132069448056e+00 2.446376594633648e-07

6.000000000000000e+00 1.291293337586988e+00 1.159132057964579e+00 3.108624468950438e-15

**so for both initial guess, my function will converge in 6 steps, but to different roots.**

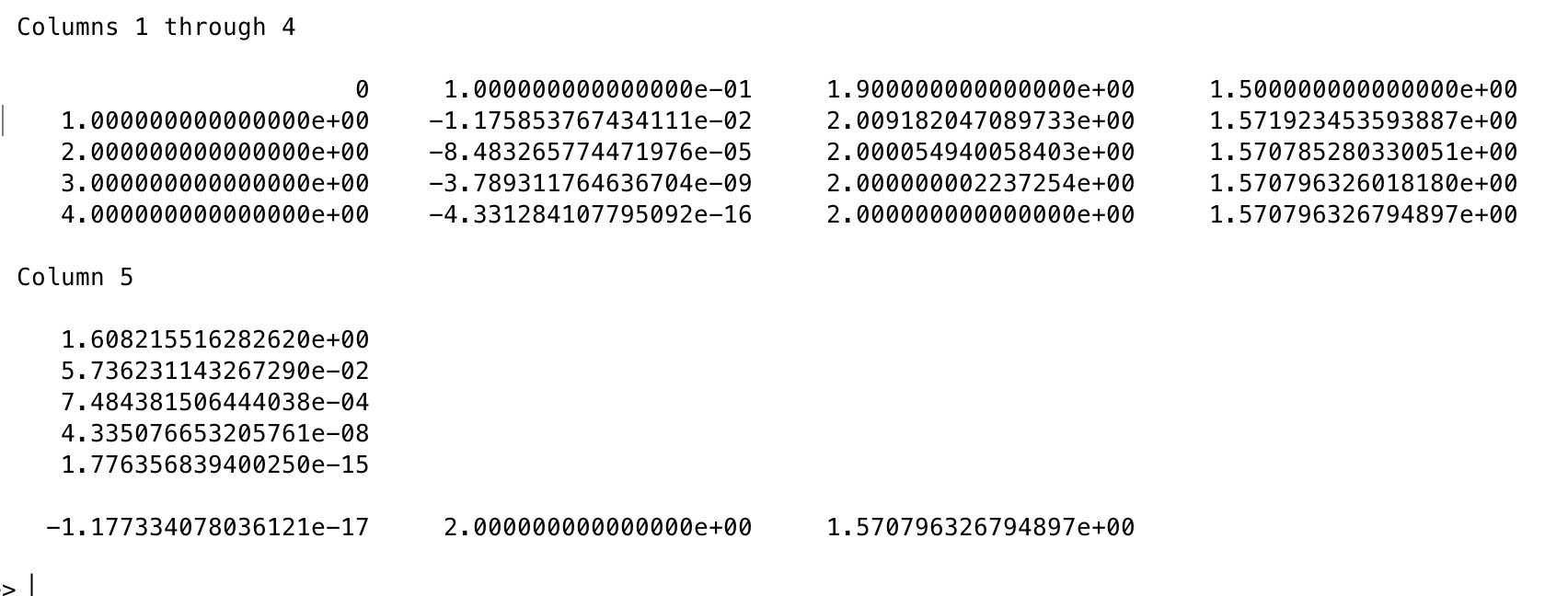
**Now for part 3, I changed my code in F.M, JF.M and Newton1.m, and here is the result:**

**For initial guess is (1,2,-5), the newton’s method does not converge in 30 steps:**

****

**the first column is iteration number, the next 3 cloumns are x, y and z.**

**then for initial guess is (0.1,1.9,1.5):**

****

**as you can see, the first column is iteration number, next three columns are x,y,z and last column is infinity norm of XN in each step, my newton’s method converge in 4 steps.**