```
%{
Jacob Leonard
MATH 467 - Fall 2015
jaleonar@usc.edu
Revision History
Date                Changes           Programmer
-------------------------------------------------
11/15/2015        Original          Jacob Leonard
       ~Seeking Counsel from Professor Wang~
12/4/2015      Reverted Function           Jacob Leonard
12/4-12/2015    Troubleshooting            Jacob Leonard
12/13/2015    Fixing Step Size             Jacob Leonard
12/15/2015   Analyzing Step Size Values   Jacob Leonard
12/16/2015        Completed               Jacob Leonard
%}
%}

%this script is for the fixed step size gradient method
%formula but for x and y given the quadratic function

%define the values of x and y from -2 to 2, increasing by 1/25, for 101
%values
for j = 1:101
    x(j) = (-2+(4*(j-1))/100);
    y(j) = (-2+(4*(j-1))/100);
end

%define an anonymous function handle for the equations that compose the gradient and the↙
hessian
f = @(x,y) ((x^4+y^4-6*x^2*y^2-1)^2+(4*x^3*y-4*x*y^3)^2);
G = {@(x,y) (8*x*(x^6+3*x^4*y^2+x^2*(3*y^4-1)+y^2*(y^4+3))),@(x,y) (8*y*(x^6+3*x^4*y^2+3*x^2*↙
(y^4+1)+y^2*(y^4-1)))};
%Gradient = [g{1}(x,y),g{2}(x,y)];
H = {@(x,y) (8*(7*x^6+15*x^4*y^2+x^2*(9*y^4-3)+y^2*(y^4+3))),@(x,y) (48*x*y*↙
(x^4+2*x^2*y^2+y^4+1));@(x,y) (48*x*y*(x^4+2*x^2*y^2+y^4+1)),@(x,y) (8*(x^6+9*x^4*y^2+3*x^2*↙
(5*y^4+1)+y^2*(7*y^4-3)))};
%Hessian = [H{1}(x,y),H{2}(x,y);H{3}(x,y),H{4}(x,y)];

%tolerance is the desired level of accuracy
tolerance = 10^(-7);

%this matrix shows the number of iterations for matlab to think the value
%is zero, or within the desired tolerance
FixedStep = zeros(101,101);
%this is the value of the function at the point the algorithm terminated
FixedStepValues = zeros(101,101);

for i = 1:101
    for j = 1:101
        Z(:,:,1) = [x(i);y(j)];
        %set the values in Z equal to 0 to track the progress
        Z(1,1,2:5000)=0;
        Z(2,1,2:5000)=0;
        g(:,:,1) = [G{1}(x(i),y(j)),G{2}(x(i),y(j))];
        %if the graident is equal to zero with the first iteration, then
        %the minimum is reached
        if (g(1,1,1) == 0) && (g(1,2,1) == 0)
            FixedStep(i,j) = 0;
            FixedStepValues(i,j) = f(Z(1,1,1),Z(2,1,1));↙
```

```
;
                continue
            end
            gT(:,:,1) = transpose(g(:,:,1));
            %define alpha
            alpha = .0005;
            %begin the iterations for steps
            for k = 2:5000
                Z(:,:,k) = Z(:,:,k-1)-alpha*(gT(:,:,k-1));
                %to speed up the code, if the Z values go to NaN or Inf, the
                %loop breaks to the next point, and sets the step number to the
                %maximum, and sets the value to 1
                if (isnan(Z(1,1,k)) == 1) || (isnan(Z(2,1,k)) == 1)
                    FixedStep(i,j) = 5000;
                    FixedStepValues(i,j) = 1;↙
;
                    break
                end
                if (isinf(Z(1,1,k)) == 1) || (isinf(Z(2,1,k)) == 1)
                    FixedStep(i,j) = 5000;
                    FixedStepValues(i,j) = 1;↙
;
                    break
                end
                %find the new graident for the updated value
                g(:,:,k) = [G{1}(Z(1,1,k),Z(2,1,k)),G{2}(Z(1,1,k),Z(2,1,k))];
                %if the gradient equals zero then the optimal value is
                %considered to have been reached
                if (g(1,1,k)) == 0 && (g(1,2,k) == 0)
                    FixedStep(i,j) = k-1;
                    FixedStepValues(i,j) = f(Z(1,1,k),Z(2,1,k));↙
;
                    break
                end
                gT(:,:,k) = transpose(g(:,:,k));
                %if the function value dips below the tolerance, then it is
                %considered to have converged to the optimal value
                if f(Z(1,1,k),Z(2,1,k))<tolerance;
                    FixedStep(i,j) = k-1;
                    FixedStepValues(i,j) = 0;
                    break
                end
                %if the algorithm reaches the end of the allowed number of
                %iterations, then the number of steps is recorded and the
                %value is sent to 1, and considered to have not converged
                if k == 5000
                    FixedStep(i,j) = 5000;
                    FixedStepValues(i,j) = 1;
                end
            end
        end
    end
xAxis = linspace(-2,2,101);
yAxis = linspace(-2,2,101);

%this plot will look at the number of steps it took for the algorithm to finish, the real↙
values of the function over the interval
%for which the algorithm finished, and the  imaginary values for which the
%algorithm finished
```

```
subplot(2,2,1:2)
%this plot will show the number of iterations it took
contourf(xAxis,yAxis,FixedStep);
xlabel('x');
ylabel('y');
title('Fixed Step Size Method # of Steps ,Alpha=.0005, Max=5000');
colorbar;
subplot(2,2,3:4)
FixedStepValuesReal = real(FixedStepValues);
contourf(xAxis,yAxis,FixedStepValuesReal);
xlabel('x');
ylabel('y');
title('Binary Convergence Plot x=[-2:2], y=[-2:2],Alhpa=.0005');
colorbar;
```