```
%{
Jacob Leonard
MATH 467 - Fall 2015
jaleonar@usc.edu
Revision History
Date                    Changes                     Programmer
------------------------------------------------------------------
11/6/2015               Original                  Jacob Leonard
11/7/2015          Developed Derivatives          Jacob Leonard
11/10/2015              Found Q                   Jacob Leonard
11/12/2015        Switched to Real Values         Jacob Leonard
11/13/2015          Troubleshooting               Jacob Leonard
11/14/2015          Developed Z Function          Jacob Leonard
%}

%this script is for the conjugate gradient method with Fletcher-Reeves formula

%determine x(0) and y(0) for the start of the methods. The script will
%eventually loop through all of these points in the 100 by 100 square of
%values, and evaluate the number of steps, as well as the rate of
%convergence

%define the values of x and y from -2 to 2, increasing by 1/25, for 100
%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)];
%when the
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)];

%analytically, the real hessian of the funciton
Q = [1,3;3,1];

%this matrix defines the size of the final graph to be plotted for
%iterations
Conjugate = zeros(101,101);

%run the algorithm for conjugate gradient and fletcher reeves
for i = 1:101
    for j = 1:101
        X(:,:,1) = [x(i);y(j)];
        g(:,:,1) = [G{1}(x(i),y(j)),G{2}(x(i),y(j))];
        d(:,:,1) = -g(:,:,1);
        dT(:,:,1) = transpose(d(:,:,1));
        gT(:,:,1) = transpose(g(:,:,1));
        %Q was computed analytically
        Q = [1,3;3,1];
        alpha = (g(:,:,1)*dT(:,:,1))/((d(:,:,1)*Q)*dT(:,:,1));
        for k = 2:101
```

```
            if g(1,1,k-1) == 0 && g(1,2,k-1) == 0
                Conjugate(i,j) = k;
                break
            end
            if isnan(g(1,1,k-1))==1 || isnan(g(1,2,k-1))==1
                Conjugate(i,j) = 0;
                break
            end
            X(:,:,k) = X(:,:,k-1)-(alpha*dT(:,:,k-1));
            g(:,:,k) = [G{1}(X(1,1,k),X(2,1,k)),G{2}(X(1,1,k),X(2,1,k))];
            gT(:,:,k) = transpose(g(:,:,k));
            beta = (g(:,:,k)*gT(:,:,k))/(g(:,:,k-1)*gT(:,:,k-1));
            d(:,:,k) = -gT(:,:,k)+(beta*dT(:,:,k-1));
            dT(:,:,k) = transpose(d(:,:,k));
            alpha = (g(:,:,k)*dT(:,:,k))/((d(:,:,k)*Q)*dT(:,:,k));
        end
    end
end
```