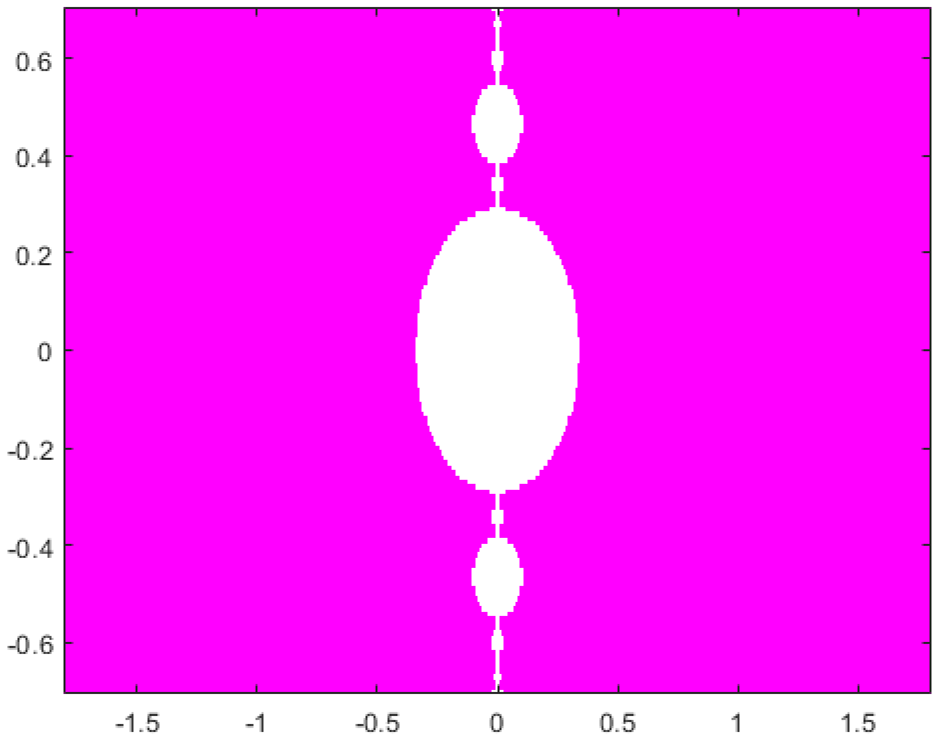
```
% part 7
% use the newton's method to iterate zk in the julia set
phi = inline('z^2 - 1.25');
fixpt1 = (1 + sqrt(6))/2;
fixpt2 = (1 - sqrt(6))/2;

colormap([1 0 1; 1 1 1 ; 0 1 1]);   % points numbered 1 (inside the
 julia set) will be colored magenta
                                    % points numbered 2 (outside the
 julia set) will be colored white
                                    % points numbered 3 (diverges in
 25 to 100 iterations) will be colored cyan
array = 2 * ones(141,361);   % initialize array of point colors to
 white

for j = 1:141   % try initial values with imaginary parts between -0.7
 and 0.7 and with real parts between -1.8 and 1.8
    y = -0.7 + (j-1)*0.01;
    for i = 1:361
        x = -1.8 + (i-1)*0.01;
        z = x + 1i*y;
        zk = z;
        iflag1 = 0;   % iflag1 and iflag2 count the number of
 iterations when a root is within 1.e-6 of a fixed point
        iflag2 = 0;
        n = 0;   % n is the total number of iterations

        while (n<100) & (abs(zk)<2) & (iflag1<5) & (iflag2<5)
            n = n + 1;
            zk = zk - (phi(zk)/(2*zk));   % this is newton's iteration
 method

            err1 = abs(zk - fixpt1);   % test for convergence to
 fixpt1
            if err1 < 1.e-6
                iflag1 = iflag1 + 1;
            else
                iflag1 = 0;
            end

            err2 = abs(zk - fixpt2);   % test for convergence to
 fixpt2
            if err2 < 1.e-6
                iflag2 = iflag2 + 1;
            else
                iflag2 = 0;
            end
        end

        if (iflag1 >= 5) | (iflag2 >= 5) | (n>=100)   % if orbit is
 bounded, set point color to magenta
            array(j,i) = 1;
```

```
        elseif (n>25) & (n<100)   % if orbit diverges in 25 to 100
 iterations, set point color to cyan
            array(j,i) = 3;
        end
    end
end

image([-1.8 1.8],[-0.7 0.7],array)    % this plots the results
axis xy
```



Published with MATLAB® R2019b