
```

%Minjun Park, CAAM 210, 09/28/15
%
%qdrive plots the basins and wasteland resulted from quadraic
function.
%Each root is assigned to different color.
%
%USAGE: These codings takes polyval, which has four roots, and it
finds the
%root using qnewt function and polyder.

function qdrive
%set four quadric vectors
xt = [.45, .0001, .55]; %vector of x grid values
yt = [-.05, 0.0001, 0.05]; %vector of y grid values
maxiter = 20; %maximum number of iterations
q1 = [1 0 -0.84 -0.1 -0.16]; %first function
q2 = [1 -0.1 -0.84 0 -0.16]; %second function
q3 = [1 -0.1i -0.84 0 -0.16]; %third function
q4 = [1 0 -0.84 0 -0.16]; %fourth function

qnewt(q1, xt, yt, maxiter)
figure
qnewt(q2, xt, yt, maxiter)
figure
qnewt(q3, xt, yt, maxiter)
figure
qnewt(q4, xt, yt, maxiter)
end
%{this function plots the 4 different graphs, taking 4 inputs}

function qnewt(q, xt, yt, maxiter)
dq = myownpolyder(q); %derivative of q
x = xt(1):xt(2):xt(3);
y = yt(1):yt(2):yt(3);
[X, Y] = meshgrid(x, y);
z = X + 1i*Y; %this saves complex numbers
r = roots(q); %this saves differnt roots of the function

%q is a vector of 5 complex coefficients of a quartic
iter = 0;
while (iter < maxiter) %runs until iter is greater than maxiter
    z = z-polyval(q,z)./polyval(dq,z); %newton's formula
    iter = 1 + iter;
end
%{this finds the value of the root}

[I1, J1] = find(abs(r(1)-z)<0.01); %this find values that are close to
one of the four roots.
[I2, J2] = find(abs(r(2)-z)<0.01); %this find values that are close to
one of the four roots.

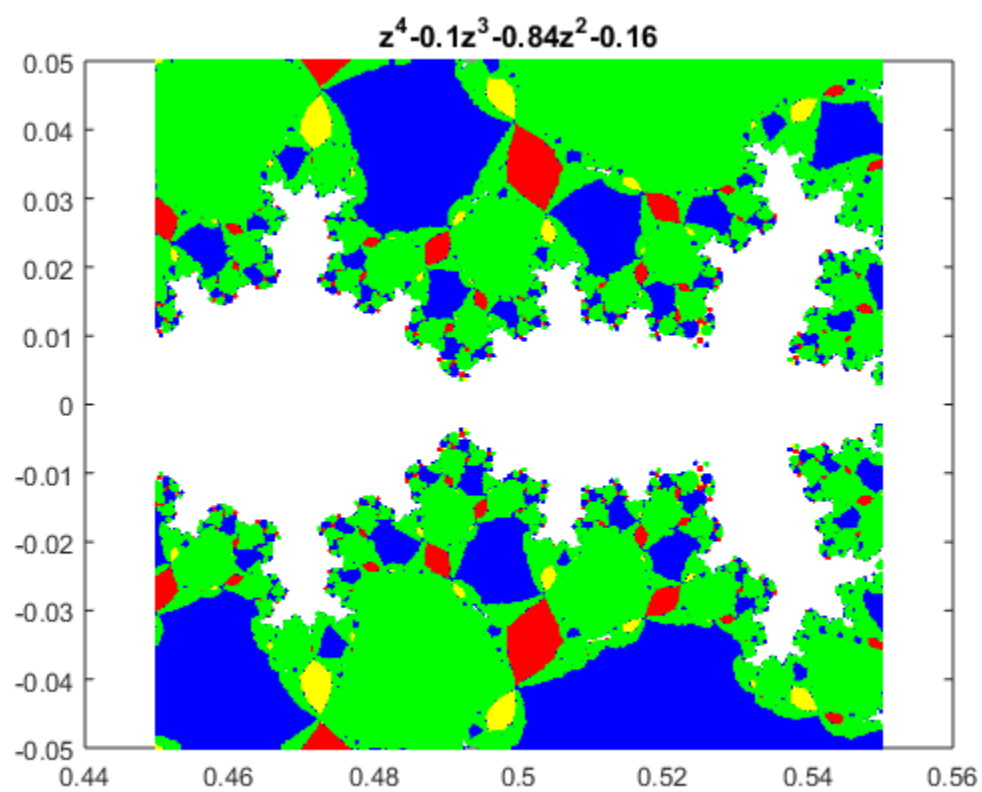
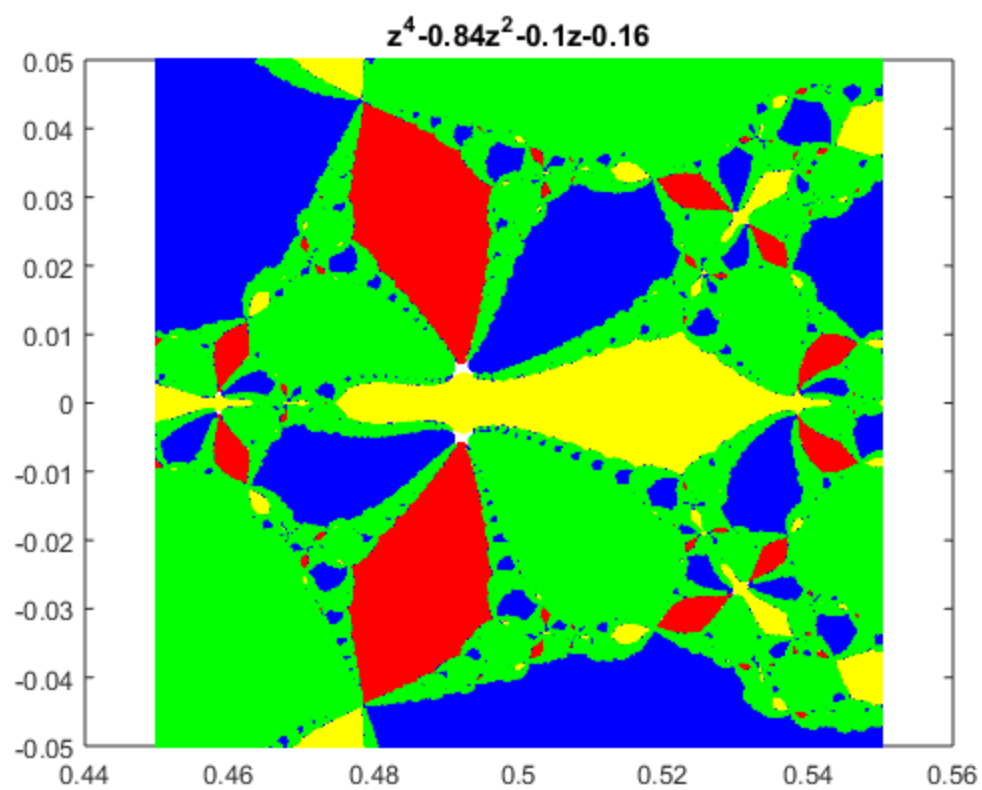
```

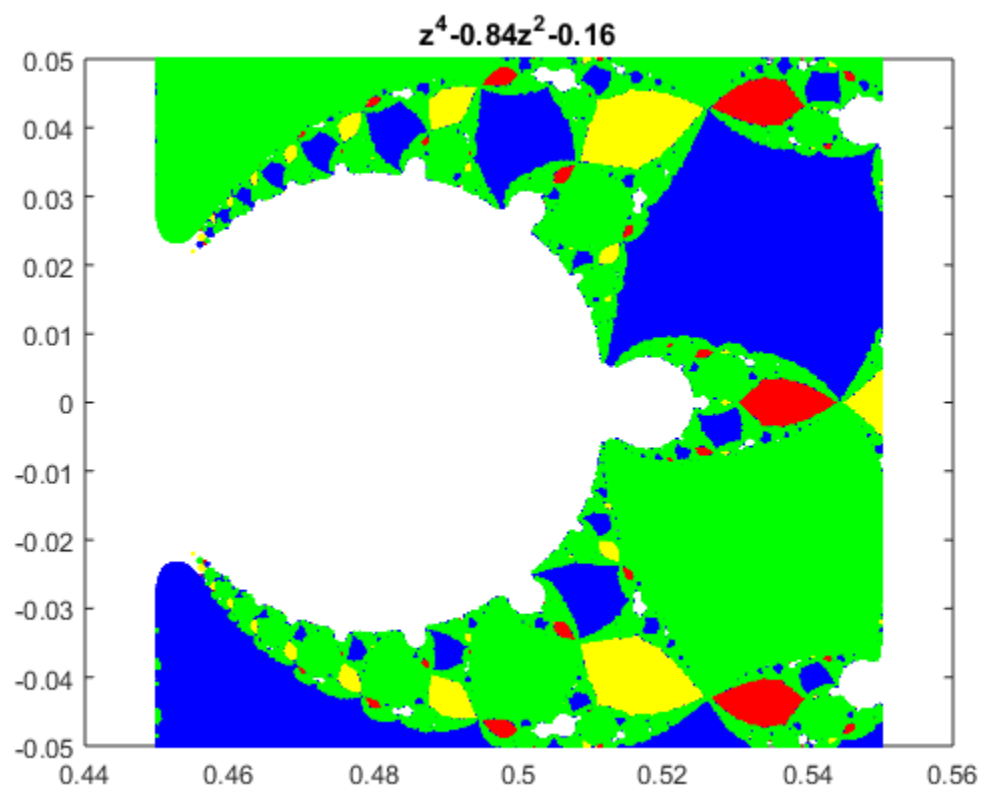
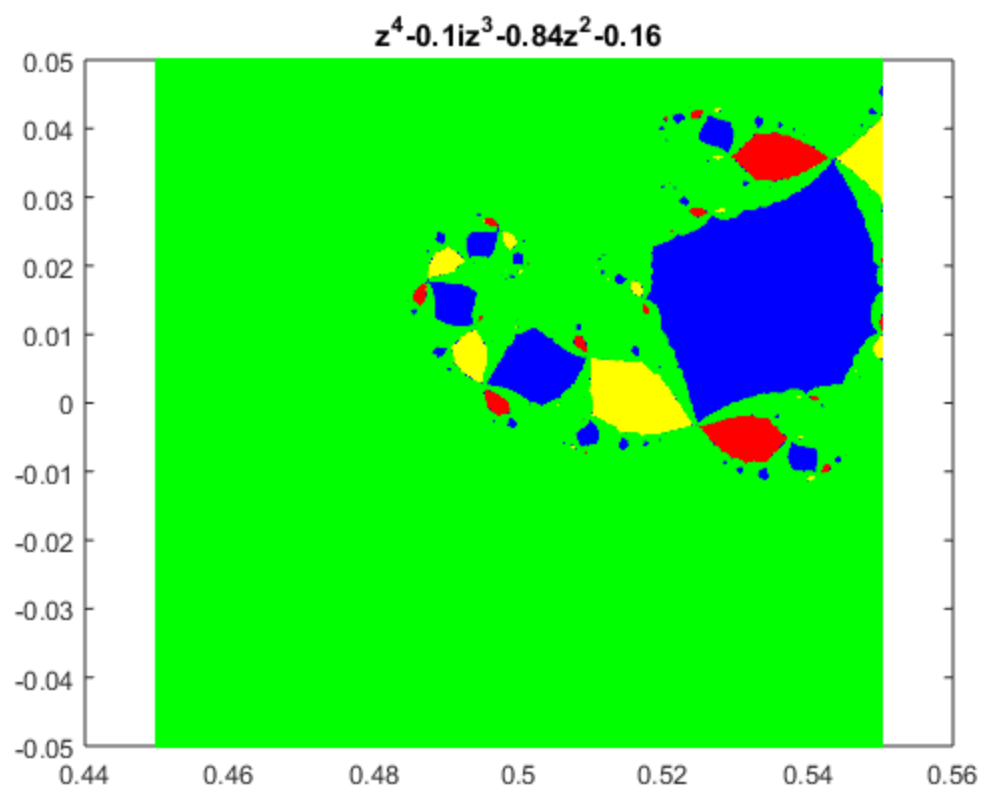
```

[I3, J3] = find(abs(r(3)-z)<0.01);%this find values that are close to
    one of the four roots.
[I4, J4] = find(abs(r(4)-z)<0.01);%this find values that are close to
    one of the four roots.
plot(x(J1), y(I1), 'r.')
hold on
plot(x(J2), y(I2), 'y.')
hold on
plot(x(J3), y(I3), 'b.')
hold on
plot(x(J4), y(I4), 'g.')
hold on
%these plots color the areas with different colors
qlab = '';
for k=1:5
    qlab = strcat(qlab, '+', num2str(q(k)), 'z^', num2str(5-k));
    %qlab gives the title of polynomial function
end
    qlab = strrep(qlab, 'z^0', '');
    qlab = strrep(qlab, '.6', '.16');
    qlab(1) = '';
    qlab = strrep(qlab, '0z^3', '');
    qlab = strrep(qlab, '-0.z^3', '');
    qlab = strrep(qlab, '0z^+', '');
    qlab = strrep(qlab, '0.z^+', '');
    qlab = strrep(qlab, '++', '+');
    qlab = strrep(qlab, '+-', '-');
    qlab = strrep(qlab, '+0-', '-');
    qlab = strrep(qlab, '--', '-');
    qlab = strrep(qlab, '0z^1', '');
    %{all these codings replaces wording(second with the third)
    if qlab(1) == '1'
        qlab(1) = '';
    end
    %if first word is equal to 1, then it erases that '1'
    qlab = strrep(qlab, '+-', '-');
    qlab = strrep(qlab, 'z^1', 'z');
title(qlab)
%this puts the designated polynomial function on top of the each graph
end
%{this function plots the graph and gives the title of the graph

function dq = myownpolyder(q)
dq = [4*q(1), 3*q(2) 2*q(3) q(4)]; %this returns the derivative
end
%this function is derivative of polynomial function

```





Published with MATLAB® R2015a