```
clear all;
colorArr = [6,3];

%RED
colorArr(1,1) = 1;colorArr(1,2) = 0;colorArr(1,3) = 0;
%GREEN
colorArr(2,1) = 0;colorArr(2,2) = 1;colorArr(2,3) = 0;
%BLUE
colorArr(3,1) = 0;colorArr(3,2) = 0;colorArr(3,3) = 1;
%YELLOW
colorArr(4,1) = 1;colorArr(4,2) = 1;colorArr(4,3) = 0;
%MAGENTA
colorArr(5,1) = 0;colorArr(5,2) = 1;colorArr(5,3) = 1;
%PURPLE
colorArr(5,1) = 1;colorArr(5,2) = 0;colorArr(5,3) = 1;
%GRAY
colorArr(6,1) = 1;colorArr(6,2) = 1;colorArr(6,3) = 1;

% f=(3+i)x^6+ 3x^5 + 4x^4 + 1x^3 + x^2 + (8-20i)x + (2-i);
% matrix representation of this function
c = [1 2-4i 0 0 0 -1 2-4i];
rootArr = roots(c);

%derivative of the function
c_der = [];
for i = 1:length(c)-1
  c_der(i)=(length(c)-i)*c(i);
end
[fdivfp, rem] = deconv(c, c_der);
while rem(1) == 0
  rem = rem(2:end);
end

%dimensions of the fractal to be generated
dim = 100;
dim = round(dim);

%variable settings for the complex guesses
min_real= -1;
max_real=  1;
min_imag= -1;
max_imag=  1;

real_step = (max_real-min_real)/dim;
imag_step = (max_imag-min_imag)/dim;

warning off MATLAB:divideByZero

min_differ = 0.01;

%maximum iteration
iter = 50;
%array for the fractal
A  =  zeros(dim, dim);
%RGB representation
B = uint8(round(A * 255));

for v=0:dim-1
    realPart = min_real + v*real_step;
    for w = 0:dim-1
        imagPart = min_imag + w*imag_step;
        xf = complex(realPart , imagPart);
        xs = xf + eps + 1;
        % Newton Raphson loop
        for k=1:iter
            z   = polyval(c, xf);
            zp  = polyval(c_der, xf);
            xs  = xf - z/zp;
            tmp = abs(repmat(xf, size(rootArr))-rootArr);
            %find the root associated with the one found in Newton Raphson
            rootIndex = find(tmp<min_differ);
            if ~isempty(rootIndex)
                 %color associated with roots and rate of convergence
                 B(v+1,w+1,1)=colorArr(rootIndex,1) * (1-(k/iter)) * 255;
                 B(v+1,w+1,2)=colorArr(rootIndex,2) * (1-(k/iter)) * 255;
                 B(v+1,w+1,3)=colorArr(rootIndex,3) * (1-(k/iter)) * 255;

                break;
            end
           xf = xs;
        end
    end
end

image(B),axis image, colormap(jet(25))
warning on MATLAB:divideByZero
```