You are to write a MATLAB function to find the cube root of a complex number.

In general, a complex number z is represented as

$$z = z_1 + \mathbf{i} z_2, \quad \mathbf{i}^2 = -1 \tag{1}$$

where z_1 and z_2 are real numbers and $\mathbf{i}$ is the imaginary unit. In MATLAB (or Octave), you can compose z out of x and y from the statement

$$z = complex(z_1, z_2)$$

and recover z_1 and z_2 from z using

$$z_1 = real(z), \quad z_2 = imag(z).$$

Given the complex number $a = a_1 + \mathbf{i} a_2$, we are looking for a root of

$$f(z; a) = z^3 - a = 0, \tag{2}$$

where z is described in (1). You are to use this function to compute the cube root of a using Newton's method for two variables.

To do that, we represent the complex numbers z and a as vectors with two real components. That is the vectors

$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}.$$

represent z in (1) and $a = a_1 + \mathbf{i} a_2$.

Then we write $f(z)$ as a function of the two real numbers z_1 and z_2 parameterized by the two real numbers a_1 and a_2. Equation (2) my be written

$$f(z; a) = g_1(z_1, z_2; \mathbf{a}) + \mathbf{i} g_2(z_1, z_2; \mathbf{a}) = 0.$$

where $g_1(\cdot)$ and $g_2(\cdot)$ are real valued functions. For $f(z)$ to be zero, both $g_1(z_1, z_2)$ and $g_2(z_1, z_2)$ must be zero since the real and imaginary parts separate. If we let

$$\mathbf{G}(\mathbf{z}; \mathbf{a}) = \mathbf{G}(z_1, z_2; \mathbf{a}) = \begin{pmatrix} g_1(z_1, z_2; \mathbf{a}) \\ g_2(z_1, z_2; \mathbf{a}) \end{pmatrix}$$

then

$$\mathbf{G}(\mathbf{z};\mathbf{a}) = \mathbf{0}$$

is equivalent to (2).

The first thing you need to do in this assignment is to identify $g_1(z_1, z_2; \mathbf{a})$ and $g_2(z_1, z_2; \mathbf{a})$ to create the function $Gcubrt(z, a)$ and implement it in MATLAB. You then need to compute the Jacobian

$$J(\mathbf{z}) = G'(\mathbf{z}) = G'(z_1, z_2).$$

and create the matrix valued function $Jcubrt(z)$ in MATLAB.

Using these two functions, create a MATLAB function with the first three lines are

function $[z, niter] = ComCubrt(a)$
avec=[real(a); imag(a)]; % Represent the complex number a as the vector avec
zvec=avec; % Just make a the initial guess for its cube root

Here a is a complex number and z is one of its cube roots. The parameter *niter* is the number of iterations required. $ComCubrt(a)$ finds z from a using Newton's method for two variables.

The last line of $ComCubrt$ should be

$$z = complex(zvec(1), zvec(2));$$

so that you can return a complex number z as your answer.

Your iteration will produce a sequence of 2-vectors $\mathbf{z}^{(0)} = \mathbf{a}, \mathbf{z}^{(1)}, \ldots,$ which terminates when

$$\|\mathbf{z}^{(k-1)} - \mathbf{z}^{(k)}\|_2 \leq tol$$

or

$$\|G(\mathbf{z}^{(k)}; \mathbf{a})\|_2 \leq tol$$

where

$$tol = 10 * \epsilon * \max\{\|\mathbf{z}^{(k)}\|_2, 10\epsilon\}$$

and ϵ is the value from the MATLAB *eps* command. *For debugging reason, do no more than 25 iterations.* Of course, *do not* keep all of the iterates $\mathbf{z}^{(k)}$, just keep the two most recent ones so that you implement the termination criteria.

Use your routine to find the cube roots of $a = 3 + 4\mathbf{i}, 10 - 5\mathbf{i}, -1 + 2\mathbf{i}$.