CECS 463 System On Chip II
FALL 2020

Kuldeep Gohil

Assignment #01 – Quadrilaterals

09/08/2020

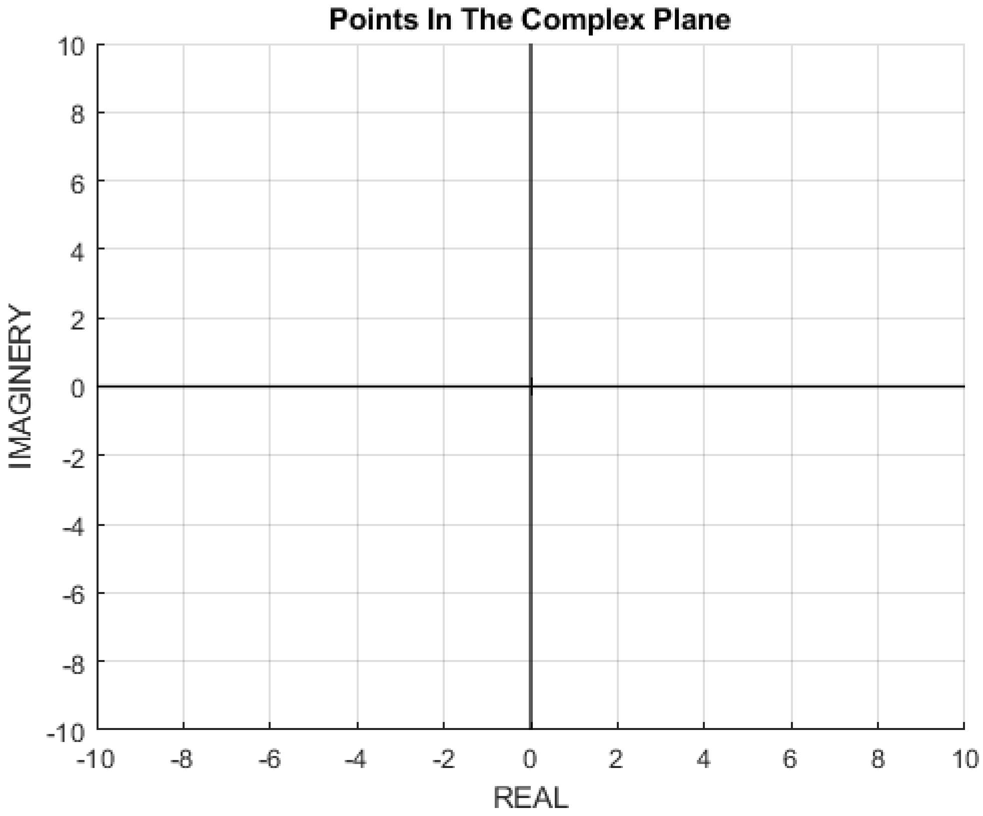
```
% Kuldeep Gohil
% CECS 463 Fall20
% Assignment #01 Due: 9/8/2020


clear all; close all; clc; format compact;
```

Displaying the graph with four quadrants

```
figure(1);


hold on; grid on;
title('Points In The Complex Plane'); xlabel('REAL'); ylabel('IMAGINERY');
plot((0),(0),'k+');                                 % Plots origin on graph
axis([-10,10,-10,10]);                              % Sets axis values
line([0,0], ylim, 'Color', 'k', 'LineWidth', 1); % Draw line for Y axis.
line(xlim, [0,0], 'Color', 'k', 'LineWidth', 1); % Draw line for X axis.
```



Generating four complex points for each quadrant

```
quad1 = +randi(10) + 1j*(randi(10));                % Quadrant 1
quad2 = -randi(10) + 1j*(randi(10));                % Quadrant 2
quad3 = -randi(10) - 1j*(randi(10));                % Quadrant 3
quad4 = +randi(10) - 1j*(randi(10));                % Quadrant 4
```

Generating four vertices from the four random complex points

```
vert1 = quad2 - quad1;                           % Vertices 1
vert2 = quad3 - quad2;                           % Vertices 2
vert3 = quad4 - quad3;                           % Vertices 3
vert4 = quad1 - quad4;                           % Vertices 4
```

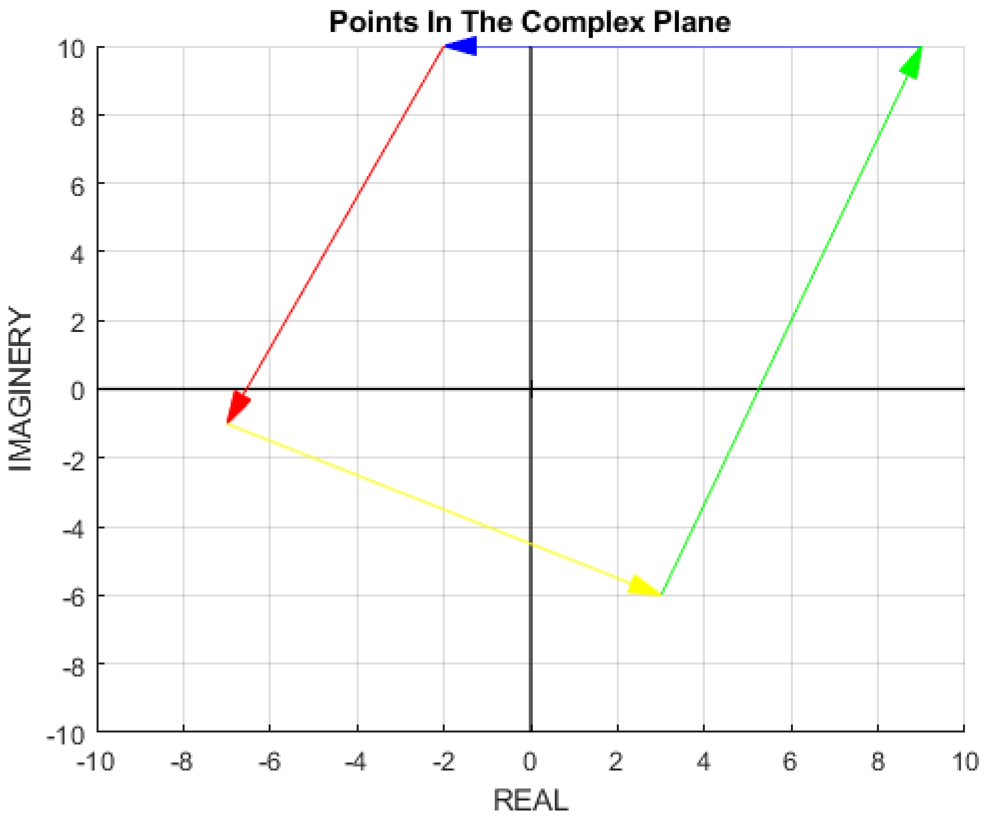
Tracing the vertices, creating quadrilateral figure

```
% blue colored vertice
arrow([real(quad1), imag(quad1)], [real(quad2), imag(quad2)], 'Color', 'b');

% red colored vertice
arrow([real(quad2), imag(quad2)], [real(quad3), imag(quad3)], 'Color', 'r');

% yellow colored vertice
arrow([real(quad3), imag(quad3)], [real(quad4), imag(quad4)], 'Color', 'y');

% green colored vertice
arrow([real(quad4), imag(quad4)], [real(quad1), imag(quad1)], 'Color', 'g');
```



Calculating the area of triangle

```
area_tri = @(quad1, quad2, quad3) 0.5*abs (real(quad1) * (imag(quad2) - imag(quad3))
...
                                         + real(quad2) * (imag(quad3) - imag(quad1))
...
                                         + real(quad3) * (imag(quad1) - imag(quad2)));
```

Calculating the area of quadrilateral

```
area_quad = @(quad1, quad2, quad3, quad4) ...
area_tri(quad1, quad2, quad3) + area_tri(quad1, quad3, quad4);

areaQ = area_quad(quad1, quad2, quad3, quad4);
```

Using cross product method to figure out the shape of quadrilateral

```
crossProduct = @(quad1, quad2) real(quad1) * imag(quad2) - imag(quad1) * real(quad2);



cp1 = crossProduct(vert1, vert2);

cp2 = crossProduct(vert2, vert3);

cp3 = crossProduct(vert3, vert4);

cp4 = crossProduct(vert4, vert1);



cp = [cp1, cp2, cp3, cp4];
```

Checking if the created figure is TRIANGLE, CONVEX, OR CONCAVE and printing it

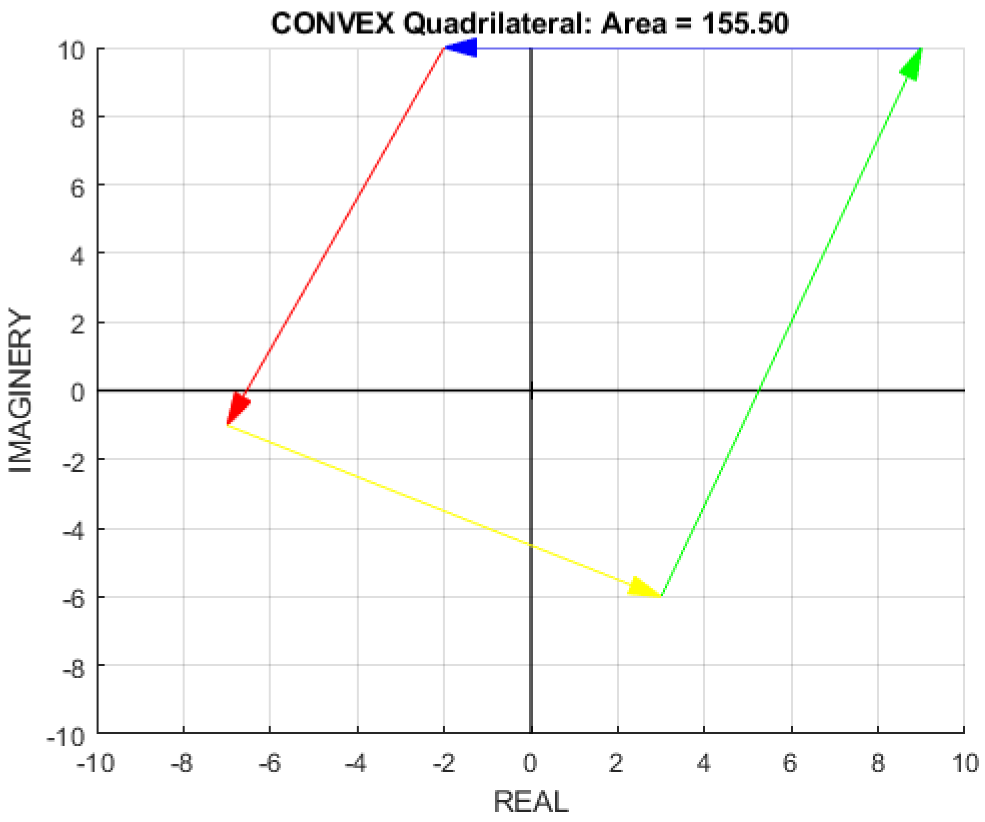
```
if (all(cp > 0) || all(cp < 0))

    % printing as title on graph
    title(sprintf('CONVEX Quadrilateral: Area = %4.2f', areaQ));
    fprintf('CONVEX Quadrilateral: Area = %4.2f', areaQ);

elseif (any (cp == 0))
    title(sprintf('TRIANGLE Quadrilateral: Area = %4.2f', areaQ));
    fprintf('TRIANGLE Quadrilateral: Area = %4.2f', areaQ);

else
    title(sprintf('CONCAVE Quadrilateral: Area = %4.2f', areaQ));
    fprintf('CONCAVE Quadrilateral: Area = %4.2f', areaQ);
end
```

```
CONVEX Quadrilateral: Area = 155.50
```



Printing all the random data points used for the figure

```
fprintf('\n Random data points happen to be at: \n');
fprintf(' z1 = %4.2f + %4.2fj\n', real(quad1),imag(quad1));
fprintf(' z2 = %4.2f + %4.2fj\n', real(quad2),imag(quad2));
fprintf(' z3 = %4.2f + %4.2fj\n', real(quad3),imag(quad3));
fprintf(' z4 = %4.2f + %4.2fj\n', real(quad4),imag(quad4));
```

```
 Random data points happen to be at: 

 z1 = 9.00 + 10.00j

 z2 = -2.00 + 10.00j

 z3 = -7.00 + -1.00j

 z4 = 3.00 + -6.00j
```