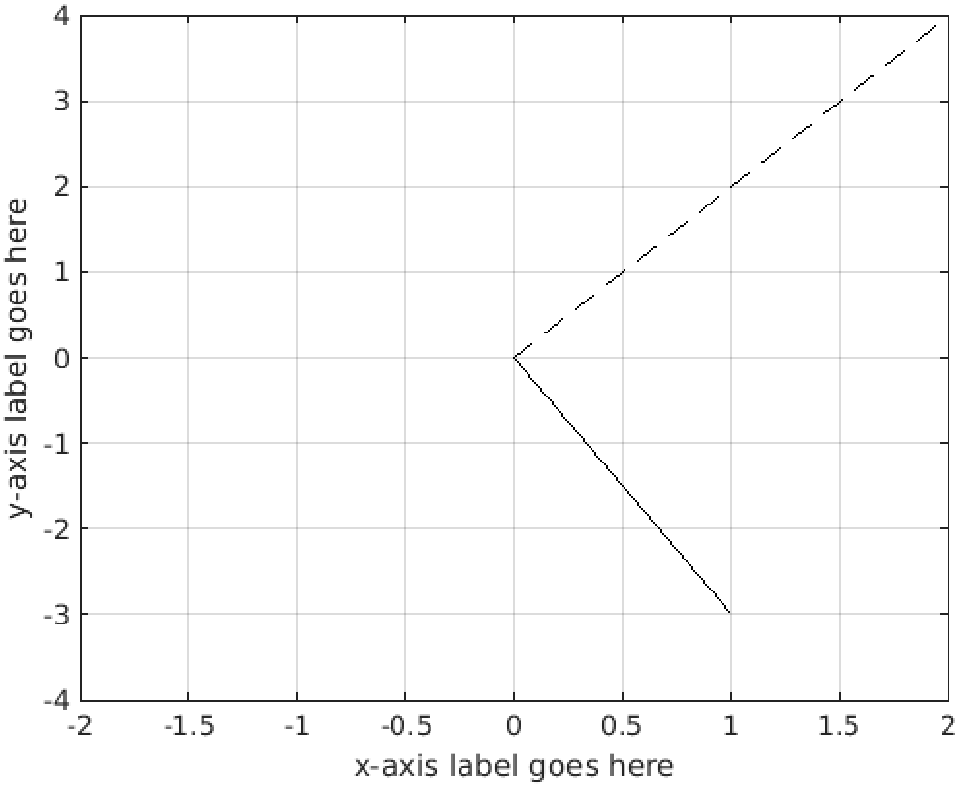
```
% BME671L
% Lab #1: clear, close, plots, vectors, complex variables, real, imag,
% atan, angle

% Your name: Ravitashaw Bathla

clear, close all
% Plotting example with u = 1 - 3j, v = 2 + 4j
figure(1), clf
plot([0 1],[0 -3],'k',[0 2],[0 4],'k--')
xlabel('x-axis label goes here')
ylabel('y-axis label goes here')
xlim([-2 2])
ylim([-4 4])
grid on
```



Q1: Define 2 complex variables u and v where u = 2 + 3j, v = -5 + j.

```
u = complex(2, 3)
v = complex(-5, 1)
```

```
u =

   2.0000 + 3.0000i
```

```
v =

  -5.0000 + 1.0000i
```

Q2: calculate the sum of u and v. Do not suppress the output.

```
u + v
```

```
ans =

  -3.0000 + 4.0000i
```

Q3: When an output is not assigned a variable it automatically gets designated as "ans". Save "ans" as a new variable z. This time suppress the output by adding a ";" to the end of the command

```
z = ans;
```

Q4: Print real and imagingary parts of z separately using the disp command. Use the matlab functions that return the components of z instead of hard coding the answer.

```
disp('real:')
disp(real(z))
disp('imag:')
disp(imag(z))
```

```
real:
    -3

imag:
     4
```

Q5: Calculate the magnitude of z using its complex conjugate either with a "'" or using the conj() command. Is there any difference between these commands? What if input was a matrix?

```
z_conj = conj(z);
z_magnitude = z*z_conj;
```

Your answer: For complex number as input, the conjugate can be calculated using either "'" or conj() function. However, the behaviour of the two operations are different for matrix. For matrix, using the "'" will calculate the conjugate along with transpose the matrix as well (this is commanly referred as CTRANSPOSE). On the other side, the conj() function will only calculate the conjugate of the matrix without transposing the matrix.

Q6: Find the angle of z using Matlab's "angle" command

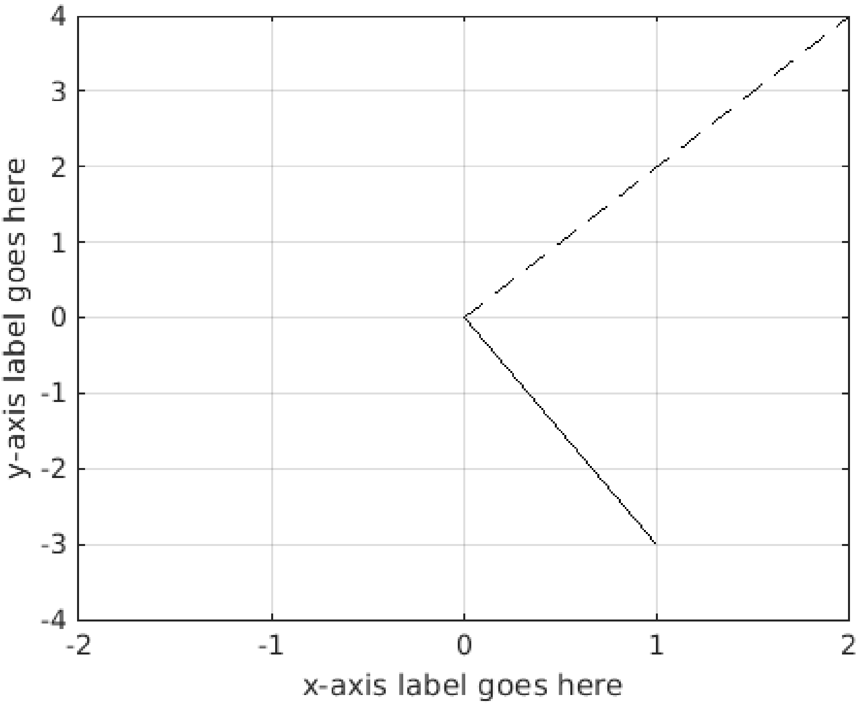
```
angle(z)
```

```
ans =
    2.2143
```

Q7: If you instead use the "atan" command to what result do you get? Why might these values disagree?

```
atan(z)

atan(imag(z)/real(z))
```

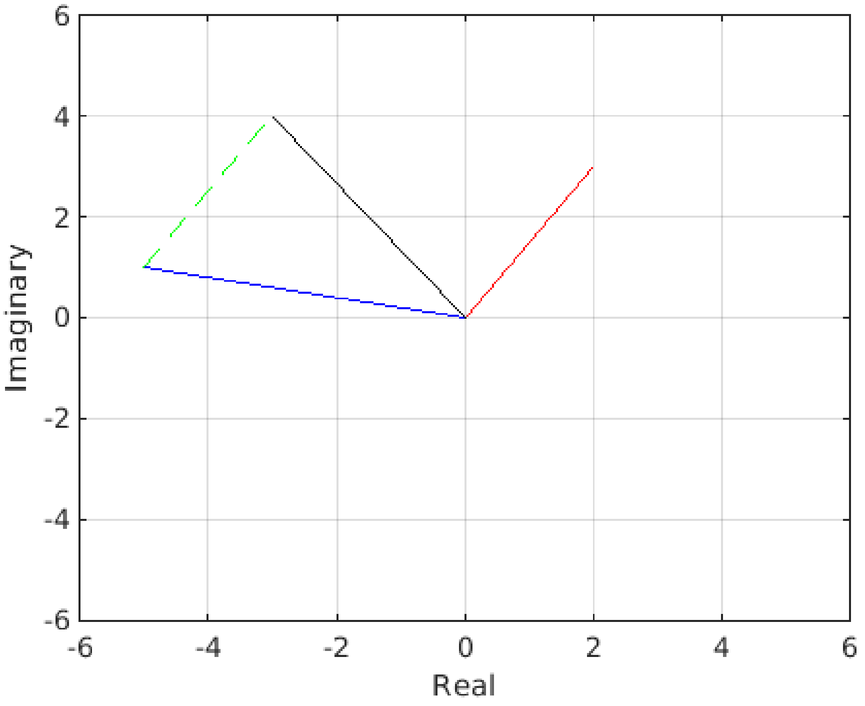
```
ans =

  -1.4483 + 0.1590i


ans =

   -0.9273
```



Your answer: The answers are different because "atan" acts on element-wise, in this case the real and imaginary part separately. However, the "angle" function works on the entire complex number, so it provides angle of the value in radians. If we want to get the same result using "atan" command, we will divide the imaginary part wrt real part and adjust the phase (as the "atan" function returns angle in range -pi/2 to pi/2). So, if we are using "atan" function, we need to make sure the angle is in correct phase by adding or subtracting pi. Otherwise, the "atan2" command should be used with two arguments for real and imaginary part separately.

Q8: Plot u in red and v in blue as solid lines on the same plot. The vectors should originate from the origin. Label the x and y axis as real and imaginary. Set both the x and y axis to display +/- 6. Add a grid.

```
figure(1), clf
plot([0 real(u)],[0 imag(u)],'r-',[0 real(v)],[0 imag(v)],'b-')
xlabel('Real')
ylabel('Imaginary')
xlim([-6 6])
ylim([-6 6])
grid on

% Add z to the plot in black using the "hold on" command.
hold on
plot([0 real(z)],[0 imag(z)],'k-')

% Add an additional vector to the plot that geometrically represents
 the
% addition of u to v. Make this vector a dashed green line. Hint:
% head-to-tail vector addition.

% calculate the tip of displaced vector u over the head v
% let the point cooridinate of head of u be (x,y)
hold on
plot([real(v) real(z)],[imag(v) imag(z)],'g--')

%
 **************************************************************************
%   SHOW FIGURE 1 TO THE TA TO RECIEVE CREDIT FOR THE LAB
%
 **************************************************************************
```



Q9: Create a function cart2polar with that has an input of a complex number in a cartesian format, z, and outputs the corresponding radius and angle for polar coordinates. The only built-in matlab commands you are allowed to use are real(), imag(), and atan(). Use if, elseif, and else. Remember that in Matlab any boolean statements require the symbol to appear twice (e.g. == or && or).

```
% ANSWER:
% in file cart2polar.m
```

Q10: Convert the following values to polar coordinates by hand and compare the answers to output of your cart2polar function. Remeber that all angles should be between -pi and pi.

```
z1 = 1+1j;
z2 = -1+j;
z3 = -3-4j;
z4 = j;

% By hand:
% z1: r = 1.414    theta = pi/4
% z2: r = 1.414    theta = 3pi/4 (or, -pi/4 + pi)
% z3: r = 5        theta = -2.2143 (or, 0.927 - pi)
% z4: r = 1        theta = pi/2

% Using cart2polar:
[r1, theta1] = cart2polar(z1)
[r2, theta2] = cart2polar(z2)
[r3, theta3] = cart2polar(z3)
[r4, theta4] = cart2polar(z4)
```

```

r1 =

    1.4142


theta1 =

    0.7854


r2 =

    1.4142


theta2 =

    2.3562


r3 =

     5


theta3 =

   -2.2143
```

```
r4 =

     1


theta4 =

    1.5708
```

When you are done:

```
% * Make sure to show the indicated result/figure to the TA during the
%     lab period to recieve credit
% * upload your script to Sakai
%   * upload your cart2polar function
%   * upload a pdf containing your script and outputs
%   * PRINT a copy of your pdf to turn in at the beginning of class on
%     Wednesday
return
```

Published with MATLAB® R2020a