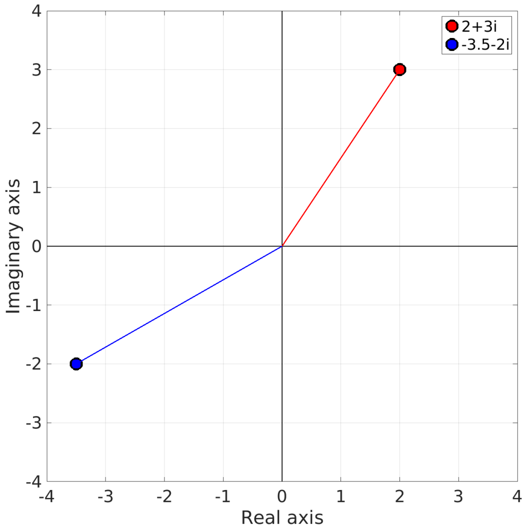
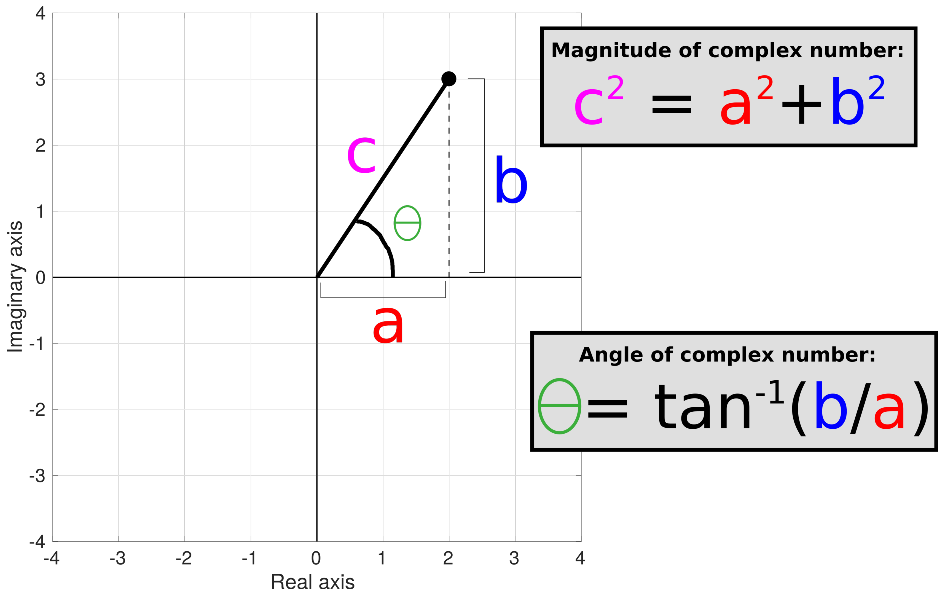
Course reader: *Foundations of the Fourier transform*

- To understand the Fourier transform, you need to know three mathematical concepts:
  1. **Sine waves**: Sine waves have three properties: frequency, power, and phase. In the Fourier transform, you specify the frequencies (more details on how this works in a subsequent video), and the power and phase are determined from the signal.
  2. **Complex numbers**: The "normal" numbers you are used to live on a 1D number line (e.g., -4 is to the left of +4). These are called *real* numbers. *Complex* numbers have both a real part and an imaginary part. Because they have two components, they can be plotted on a 2D *plane*, called the complex plane. The figure below illustrates a few complex numbers and their locations on the complex plane.



  3. **Dot product**. The dot product is a single number that reflects the relationship between two vectors (a vector is a list of numbers). The dot product is the computational backbone of many algorithms and mathematical operations, including the Fourier transform, convolution, and correlation. To compute the dot product between two vectors, simply point-wise multiply the corresponding elements and then sum over all elements. The larger the dot product, the more similar the two vectors are to each other (after accounting for the scaling of each vector, which might be arbitrary).

- The geometric interpretation of a complex numbers leads to the concept of "length" and "angle" of a complex number. Both properties are extracted by using trigonomety: length is the square root of the real part squared plus the imaginary part squared ($m = \sqrt{real(z)^2 + imag(z)^2}$), and angle is the inverse tangent of the imaginary part over the real part ($\phi = atan(\frac{imag(z)}{real(z)})$). When the complex number is a "Fourier coefficient," then the length is the signal amplitude and the angle is the signal phase. You'll learn why this is in the next video!



- When you compute the dot product between a real-valued signal and a complex sine wave, the resulting dot product is also a complex number. In the context of the Fourier transform, the dot product of the signal with a complex sine wave is called a "Fourier coefficient."

- Note that when you compute dot products between *the same signal* and *many sine waves* of different frequencies, differences in the Fourier coefficients reflect frequency-specific components of the signal.

- The next lecture will explain how the Fourier transform works, but to give you a quick preview: To compute the Fourier transform, compute the dot product between a signal and a complex sine wave. The magnitude of the dot product is the signal amplitude at the frequency of the sine wave, and the angle of the dot product with respect to the positive real axis is the phase of the signal at that frequency. This procedure (dot product between signal and complex sine wave, extract amplitude and phase) is repeated for many sine wave frequencies, and the result is a power or phase spectrum.

Exercises

1. Draw **by hand** sine waves lasting one second with the following parameters (f=frequency in Hz, p=phase in radians, a=amplitude).

a) f=2, p=0, a=1 **b)** f=4, p=$\pi/2$, a=3 **c)** f=.5, p=0, a=1 **d)** f=4, p=π, a=2

2. Write code to generate and plot two-second signals that comprise a sum of the following sine wave parameters. Test various sampling rates, ranging from 1 Hz to 1000 Hz, to determine what—if any—effect that has on the plots.

a) $\begin{array}{ccc} f=2 & p=0 & a=1 \\ f=4.2 & p=3\pi/4 & a=1.7 \end{array}$ **b)** $\begin{array}{ccc} f=200 & p=\pi & a=100 \\ f=402 & p=0 & a=10 \\ f=3.2 & p=1 & a=50 \end{array}$

3. In theory, sine waves have an equal amount of area above and below the Y=0 axis, meaning the average of a sine wave over all time points should be 0. Using a sampling rate of 1 kHz, generate a 2-second sine wave at 6 Hz and average over all values. It won't be exactly zero because of computer rounding errors, but anything less than 10^{-10} (in code: `1e−10`) you can consider to be zero. Now repeat with a sine wave of 6.1 Hz. Is the answer still zero? Try again with 6.5 Hz. To understand why this happens, plot the sine waves.

4. Adapt the previous exercise to create and average sine waves in a loop with frequencies varying from 1 to 50 Hz in 90 linearly spaced steps. Then make a plot of frequency by average amplitude value.

5. Compute the dot product **by hand** between the following pairs of vectors.

a) $\begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ **b)** $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} -3 \\ -2 \\ -1 \end{bmatrix}$ **c)** $\begin{bmatrix} 3 \\ -7 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ -7 \\ 1 \end{bmatrix}$ **d)** $\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} -3 \\ 2 \end{bmatrix}$ **e)** $\begin{bmatrix} 7 \\ -2 \\ 3 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \\ -6 \end{bmatrix}$

6. Draw the following complex numbers on a complex plane. Then extract their magnitude (distance to origin) and phase. You can use a calculator to do the inverse tangent (or brush up on your trig identities!); the important thing is to write down the equation correctly. Remember that you want to square the imaginary *component*, not the imaginary *number* itself (thus, $(2i)^2$ is treated as 2^2 and does not become -4).

a) $2 \; 3i$ **b)** $0 \; i$ **c)** $-3 \; i4$ **d)** $-1 \; -i$

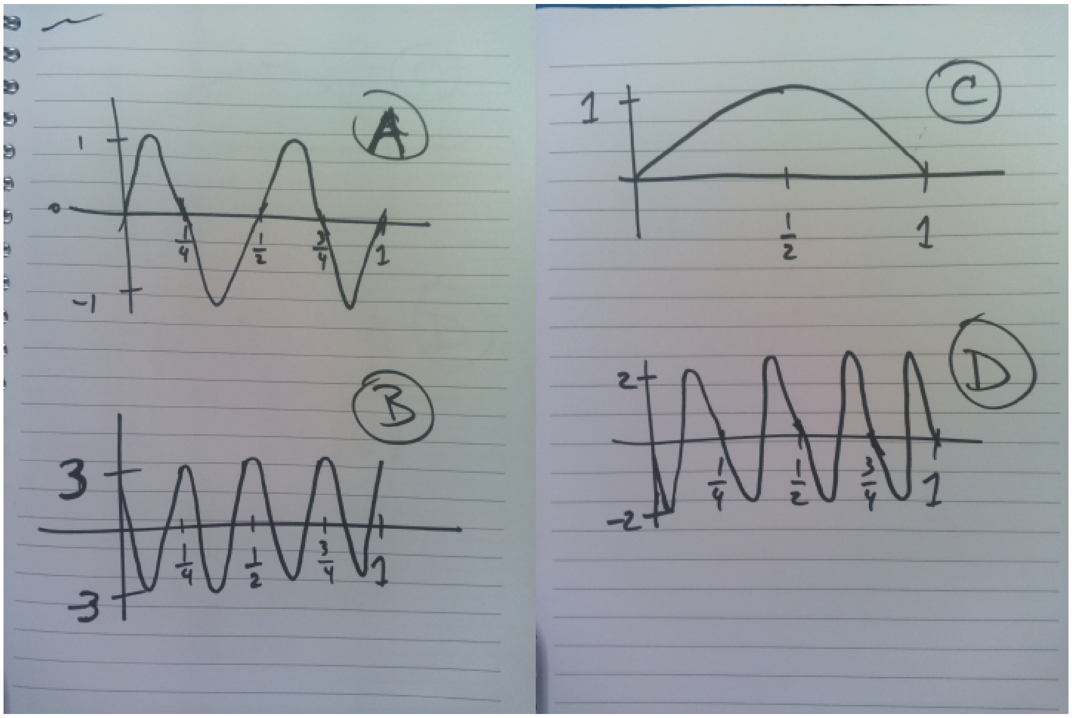
7. Compute the dot product between the following pairs of vectors. Draw the resulting complex dot product in complex plane. Finally, compute the magnitude of the dot product.

a) $\begin{bmatrix} 1 \\ -3 \end{bmatrix}, \begin{bmatrix} 1 \\ i \end{bmatrix}$ **b)** $\begin{bmatrix} 4 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ -2i \end{bmatrix}$ **c)** $\begin{bmatrix} 7i \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ **d)** $\begin{bmatrix} 4 \\ -4 \end{bmatrix}, \begin{bmatrix} i \\ i \end{bmatrix}$

MATLAB Answers

Scroll a few pages to see the solutions using Python code.

1. I'm a terrible artist. Really awful. I tried and failed to stay in the lines in coloring books up until my 20's, and finally gave up. But in the interest of completing the assignments as I wrote them, here's my best shot.



2. Partly this exercise is about practicing generating sine waves in MATLAB. The point of changing the sampling rate is that sine waves can be constructed only when the sampling rate is at least twice the sine wave frequency. This is related to the sampling theorem and the Nyquist frequency. You'll learn more about that in the lecture about frequency resolution.

```
srate = 1000;
time  = 0:1/srate:2-1/srate;

sine1 = 1.0*sin(2*pi*2.0*time + 0) + ...
1.7*sin(2*pi*4.2*time + 3*pi/4);

sine2 = 100*sin(2*pi*200*time + pi) + ...
10*sin(2*pi*402*time +  0) + ...
50*sin(2*pi*3.2*time + 1);

subplot(211)
plot(time,sine1,'linew',3)

subplot(212)
plot(time,sine2,'linew',3)
```

3. The reason why the average is above zero is because the sine wave does not go through a full cycle at the end. This is a result of the frequency, sampling rate, and duration of time. Sine waves that do not have full cycles require nonzero energy at frequencies other than what you specified, meaning that the amplitude spectra will be "less pure." This will come up in the discussion about the effects of non-stationarities on the power spectrum.

```
fs   = 1000;
time = 0:1/fs:2-1/fs;
mean( sin(2*pi*6.5*time) )
```

4. Feel free to explore the parameters. For example, try changing the range and number of frequencies, sampling rate, and time duration.

```
srate  = 1000;
time   = 0:1/srate:2-1/srate;
frex   = linspace(1,50,90);
avesin = zeros(size(frex));

for fi=1:length(frex)
    avesin(fi) = mean( sin(2*pi*frex(fi)*time) );
end

plot(frex,avesin,'linew',4)
xlabel('Frequency (Hz)'), ylabel('Average amplitude')
```

5. Don't forget: dot product is point-wise multiply and sum.

a) 3 **b)** -10 **c)** 59 **d)** 0 **e)** 0

6.
```
% list point and compute magnitude and phase
pnt = [2 3];
mag = sqrt(sum(pnt.^2)); % or norm(c)
phs = atan2(pnt(2),pnt(1));

% plot point as line and dot
plot([0 pnt(1)],[0 pnt(2)],'k','linew',3)
hold on
plot(pnt(1),pnt(2),'ko','linew',3,'markerfacecolor','w','markersize',15)
axis([-1.1 1.1 -1.1 1.1]*max(pnt))
xlabel('Real axis'), ylabel('Imag axis')
title([ '[' num2str(pnt) 'i]:  mag.=' num2str(mag) ,' phase=' num2str(phs) ])

% also draw axes and grid
plot(get(gca,'xlim'),[0 0],'k')
plot([0 0],get(gca,'ylim'),'k')
grid on
```

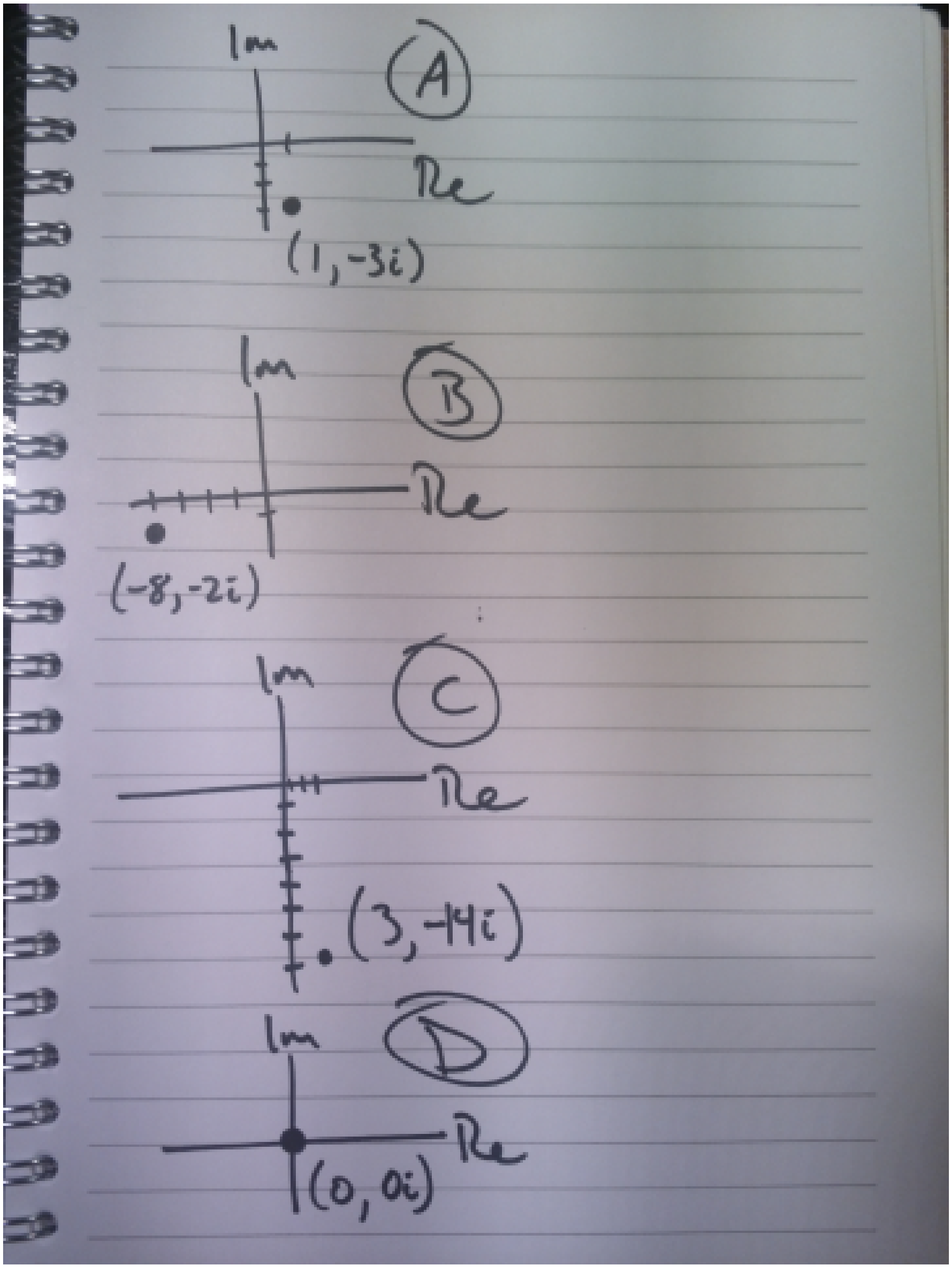
7. *dp* is dot product, *m* is magnitude. Notice that the magnitude is always a non-negative real number, even when the vector contains imaginary elements.

a) $dp = [1 - 3i]$, $m = \sqrt{10}$

b) $dp = [-8 - 2i]$, $m = \sqrt{68}$

c) $dp = [3 - 14i]$, $m = \sqrt{205}$

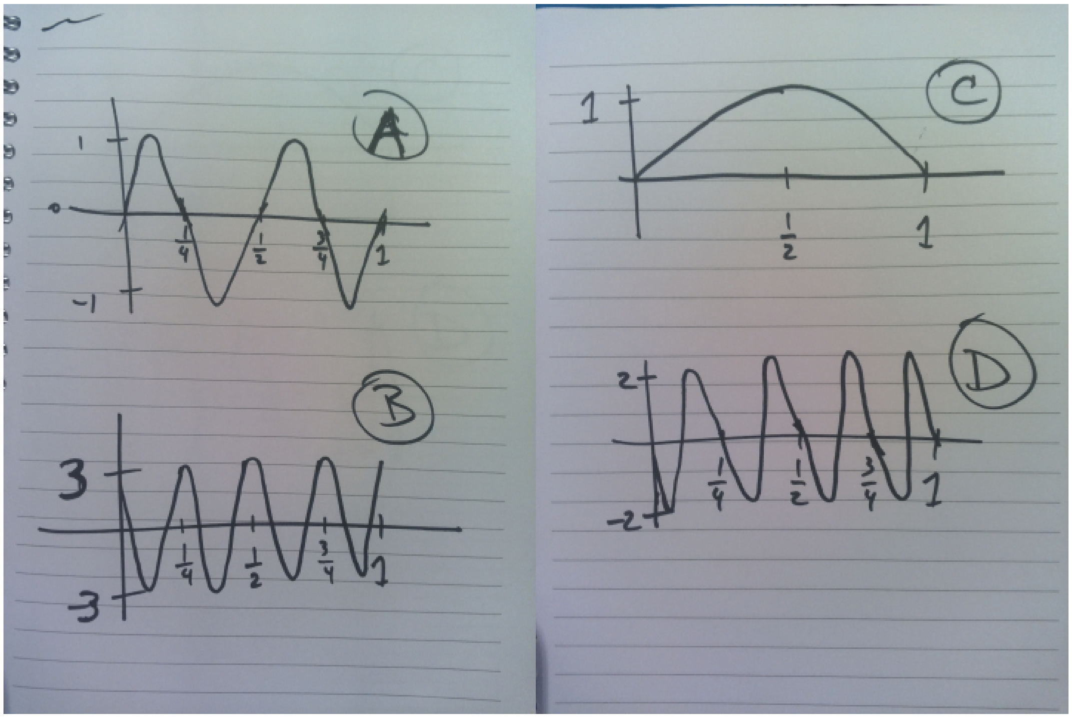
d) $dp = [00i]$, $m = 0$



PYTHON Answers

These answers are identical to the previous pages but using Python code instead of MATLAB code.

1. I'm a terrible artist. Really awful. I tried and failed to stay in the lines in coloring books up until my 20's, and finally gave up. But in the interest of completing the assignments as I wrote them, here's my best shot.



2. Partly this exercise is about practicing generating sine waves in Python. The point of changing the sampling rate is that sine waves can be constructed only when the sampling rate is at least twice the sine wave frequency. This is related to the sampling theorem and the Nyquist frequency. You'll learn more about that in the lecture about frequency resolution.

```
import numpy as np
import matplotlib.pyplot as plt

srate = 1000
time  = np.arange(0,2-1/srate,1/srate)

sine1 = 1.0*np.sin(2*np.pi*2.0*time + 0) + \
        1.7*np.sin(2*np.pi*4.2*time + 3*np.pi/4)

sine2 = 100*np.sin(2*np.pi*200*time + np.pi) + \
         10*np.sin(2*np.pi*402*time + 0) + \
         50*np.sin(2*np.pi*3.2*time + 1)

_,axs = plt.subplots(2,1)
axs[0].plot(time,sine1)
```

```
axs[0].set_title('sine1')

axs[1].plot(time,sine2)
axs[1].set_title('sine1')
axs[1].set_xlabel('Time (s)')

plt.tight_layout()
plt.show()
```

3. The reason why the average is above zero is because the sine wave does not go through a full cycle at the end. This is a result of the frequency, sampling rate, and duration of time. Sine waves that do not have full cycles require nonzero energy at frequencies other than what you specified, meaning that the amplitude spectra will be "less pure." This will come up in the discussion about the effects of non-stationarities on the power spectrum.

```
fs   = 1000
time = np.arange(0,2-1/fs,1/fs)
print(np.mean( np.sin(2*np.pi*6.5*time) ))
```

4. Feel free to explore the parameters. For example, try changing the range and number of frequencies, sampling rate, and time duration.

```
srate  = 1000
time   = np.arange(0,2-1/srate,1/srate)
frex   = np.linspace(1,50,90)
avesin = np.zeros(len(frex))

for fi in range(len(frex)):
    avesin[fi] = np.mean( np.sin(2*np.pi*frex[fi]*time) )

plt.plot(frex,avesin)
plt.xlabel('Frequency (Hz)')
plt.ylabel('Average amplitude')
plt.show()
```

5. Don't forget: dot product is point-wise multiply and sum.

a) 3 **b)** -10 **c)** 59 **d)** 0 **e)** 0

6.

```
# list point and compute magnitude and phase
pnt = np.array([2,3])
mag = np.sqrt(np.sum(pnt**2)) # or np.norm(c)
phs = np.arctan2(pnt[1],pnt[0])

# plot point as line and dot
plt.plot([0,pnt[0]],[0,pnt[1]],'k')
plt.plot(pnt[0],pnt[1],'ko',markerfacecolor='w',markersize=15)

plt.axis('square')
```

```
m = np.max(np.abs(pnt))*1.1
plt.axis([-m,m,-m,m])
plt.xlabel('Real axis')
plt.ylabel('Imag axis')
plt.title(f'{pnt}: mag={mag:.3f}, phase={phs:.3f}')

# also draw axes and grid
plt.plot([-m,m],[0,0],'k')
plt.plot([0,0],[-m,m],'k')
plt.grid('on')
plt.show()
```

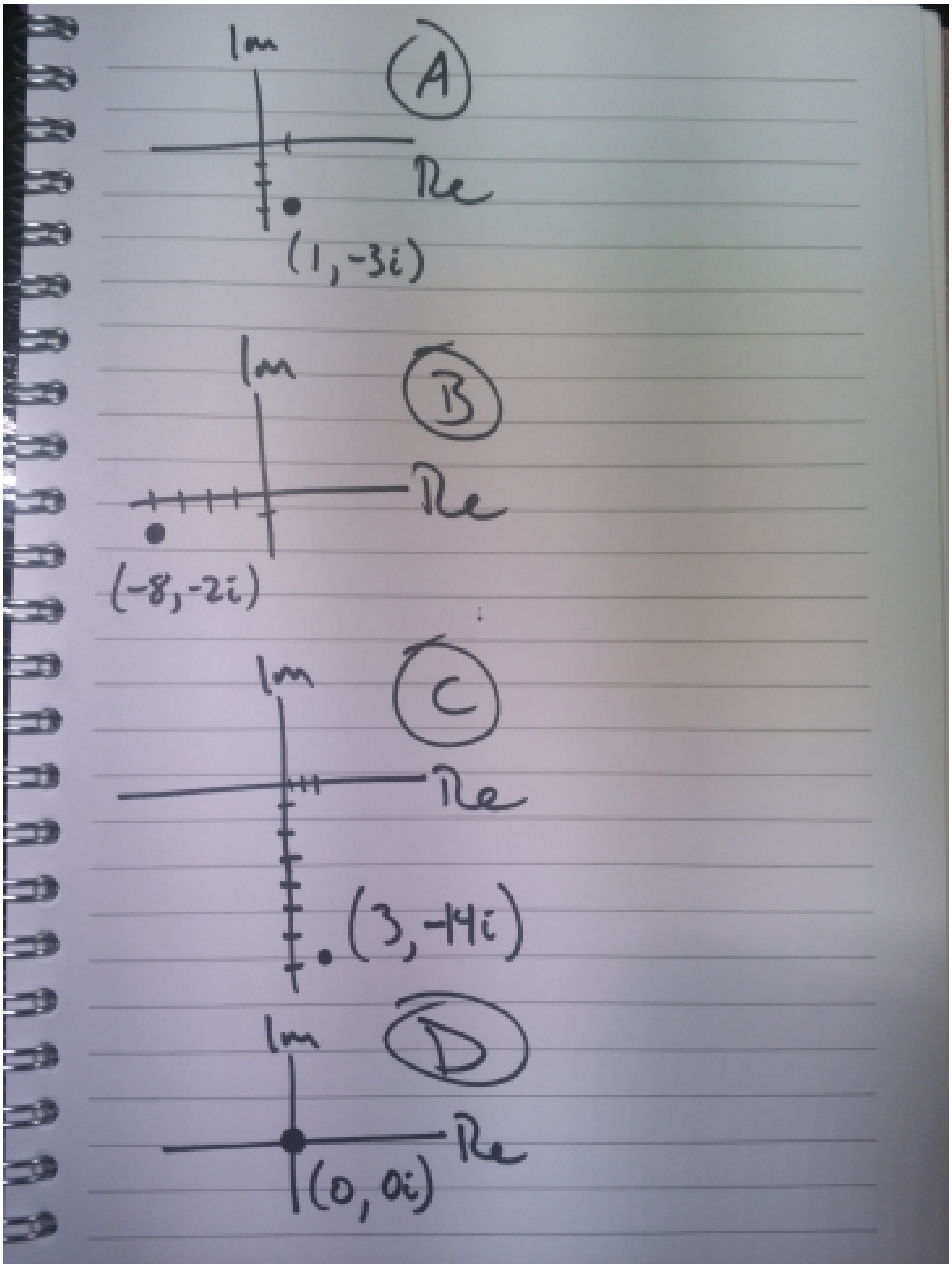
7. *dp* is dot product, *m* is magnitude. Notice that the magnitude is always a non-negative real number, even when the vector contains imaginary elements.

a) $dp = [1 - 3i]$, $m = \sqrt{10}$

b) $dp = [-8 - 2i]$, $m = \sqrt{68}$

c) $dp = [3 - 14i]$, $m = \sqrt{205}$

d) $dp = [00i]$, $m = 0$


A
Im
Re
(1, -3i)
B
Im
Re
(-8, -2i)
C
Im
Re
(3, -14i)
D
Im
Re
(0, 0i)