**Problem Set 8: Complex Numbers**

Goal: Become familiar with math operations using complex numbers; see how complex numbers can be used to show the frequency response of an RC circuit.

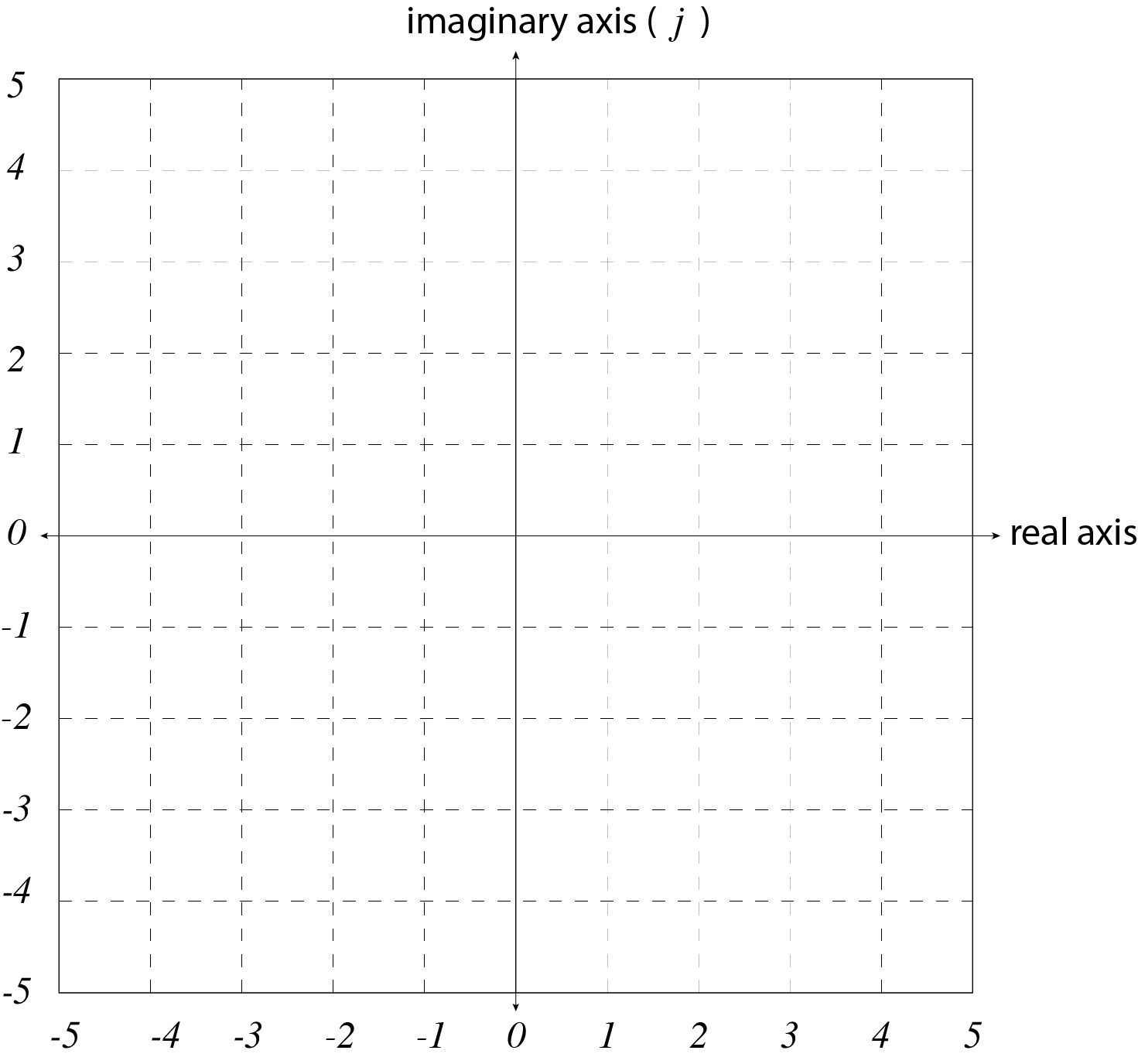
*Note: This PSet will be much easier if you have already watched the lectures on complex numbers.* 

Deliverable: This worksheet and two plots.

**Part I: Basic Operations with complex numbers**

For the following, take and .

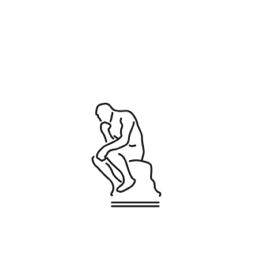
1. Convert and to polar and exponential notation (find ).
2. Plot and on the complex plane below.

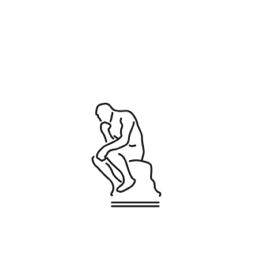
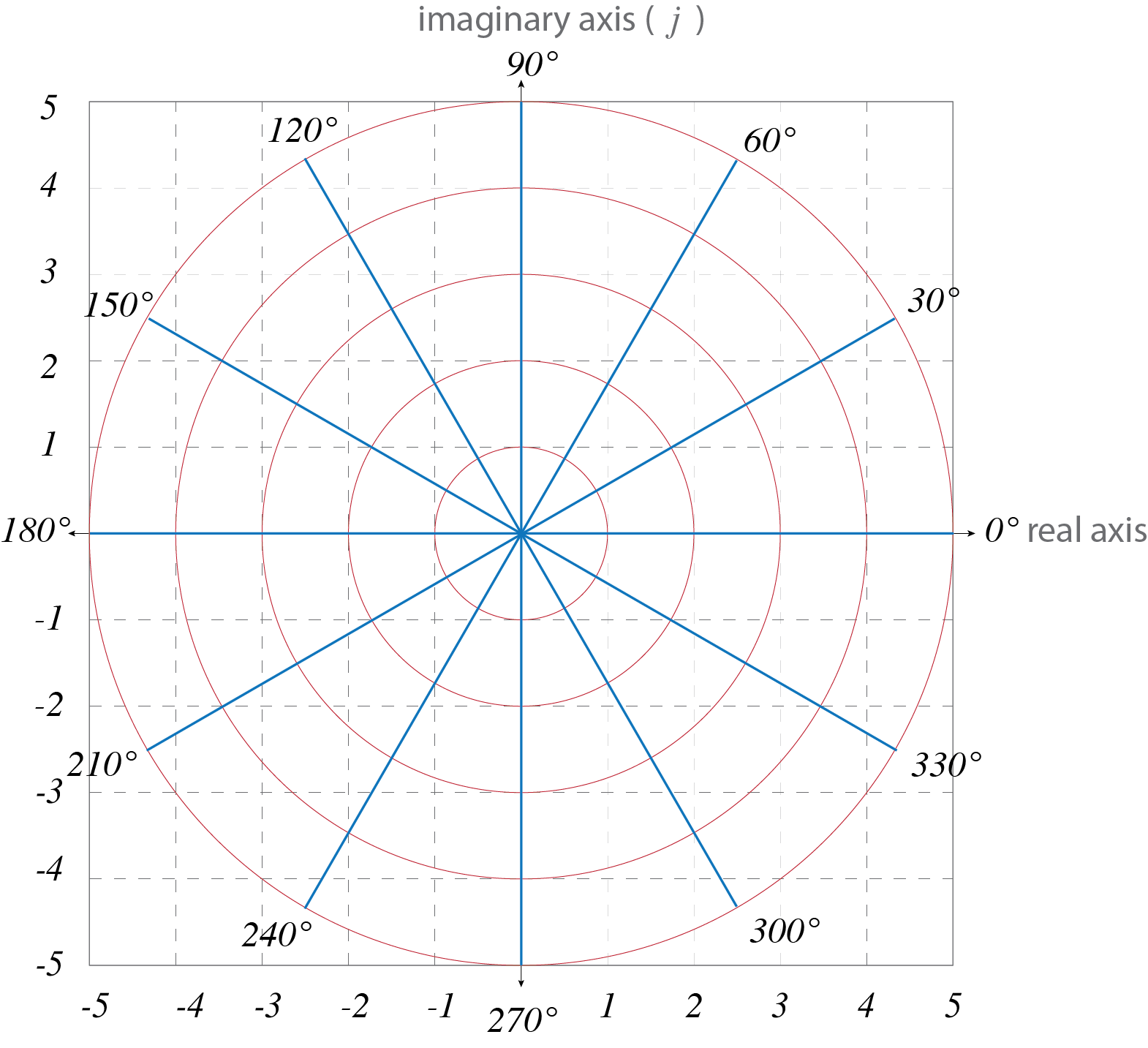


1. Compute + . Show + . graphically on a plot in the complex plane from 2.
2. Compute - . Show - . graphically on a plot in the complex plan from 2.
3. Compute . Repeat the computation using a different notation.
4. Compute using complex notation. Compute and compare.
5. Compute

**Part II: Plotting complex numbers**

Complex numbers using **polar notation** are super useful for illustrating how a circuit responds to time-varying signals.



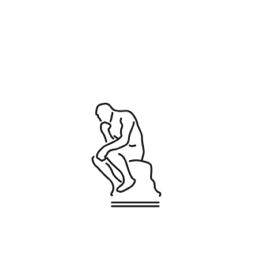


The **polar coordinates** (above grid of red & blue) make use of a special property of the ***exponential* *function*** when it operates on )***.*** You may have seen this function notated (equivalently) as:

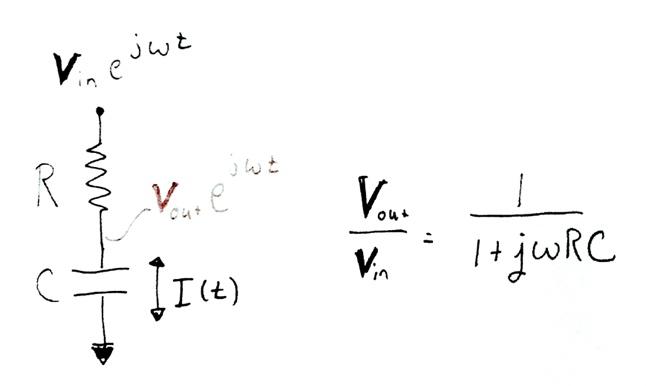
, , or

where represents an angle in radians (Recall that radians = 180°).

The amazing property of is known as Euler’s formula (section 6.3 in your book):



Recall from Figure 6.3 that if we represent our cosine voltage input to a **low-pass filter** with polar notation,



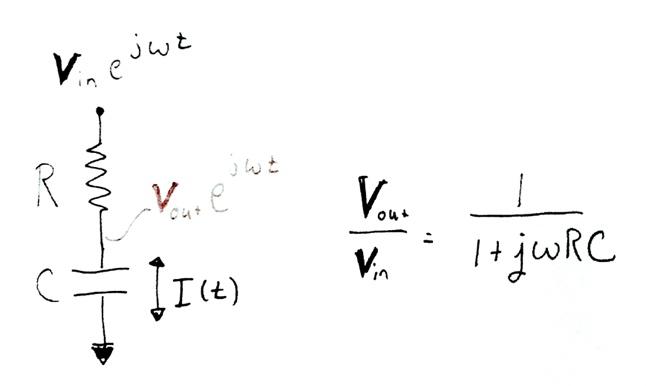
And represents a complex number.

And remember that because the R and C are in series, the time varying current passing through both will be the same, we get,

and, rearranged a bit,

.

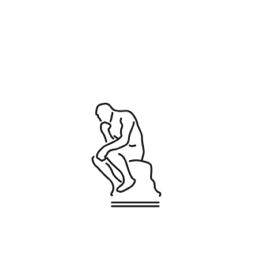
Or

solving for ,

Let’s let RC=1 second and 

And

Plot the magnitude of *r* of z3 and z4 as a function of ω on a log-log scale. Let ω\* vary from 10-3 to 103.

Plot 𝛳 in degrees for z3 and z4 as a function of ω on a semilog scale. Let ω vary from 10-3 to 103.

\*In Matlab, you can use the command:

y= logspace(-3,3)

to generate a logarithmically-spaced

vector, y, that spans 10-3 to 103.



