Complex Numbers

Background:

Up to now, we've been using real numbers for all of our calculations. Real numbers work great when dealing with constant (DC) signals. There's only so much you can do with DC signals, however.

From this point onwards, we'll start working with sinusoidal signals. Sine waves are useful. Sine waves describe a large number of signals, such as whistles, oscillations, and rotation of objects. Sine waves can be added together to create other periodic functions, such as sounds made by musical instruments. Sine waves are also eigenfuntions. That means that, if you have a differential equation with a sinusoidal input, the output will also be a sinusoid.

One problem with sine waves is, at any given frequency, there are two degrees of freedom: sine and cosine. This means that to describe a given sine wave, two numbers are needed. This is where complex numbers come into play.

With complex numbers, there are two parts: the real part and the complex part. This lets you represent both the cosine and sine term for a sine wave with a single number - greatly simplify computations. Likewise, from this point onwards, pretty much all calculations in this and future classes will be using complex numbers.

In this lecture, we'll be going over

- Complex numbers
- Computations with complex numbers, and
- Complex numbers in Matlab

note: You will need a calculator capable of doing complex operations for the tests. It's your choice, but I recommend the HP42 calculator (free app on your cell phone called *Free42*.) There is a learning curve with the Free42 calculator - so don't try to learn it on a midterm. Matlab also works well with complex numbers.

Complex Numbers:

With complex numbers, there are two parts: the real part and the complex part with the latter denoted with the letter 'i' in math, 'j' in electrical and computer engineering

$$x = a + jb$$

The symbol 'j' represents the square root of minus one:

$$j = \sqrt{-1}$$

This means that powers of j are

$$j^2 = -1$$

$$j^3 = -j$$

$$j^4 = +1$$

Other powers are possible - but more on that later.

Another definition of complex numbers is Euler's identity

$$e^{j\theta} = \cos(\theta) + j\sin(\theta)$$

This gives a second way to represent a complex number: polar form. In rectangular form, the real and complex parts are specified separately, such as

$$x = a + jb$$

In polar form, the magnitude and angle are specified

$$x = r\angle\theta$$

This is shorthand notation for

$$x = r \cdot e^{j\theta}$$

From Euler's identity, this becomes

$$x = r\cos\theta + j \cdot r\sin\theta$$

Graphically, this is a unit vector in the complex plane at an angle as measured from the real axis as in Figure 1. The real part is cos(q). The complex part is sin(q).

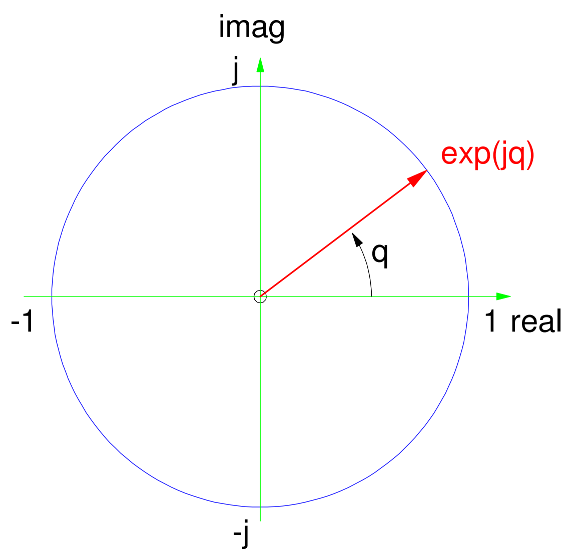


Figure 1: Euler's identity is equivalent to a unit vector in the complex plane at angle theta

Euler's identity leads to one definition of sine() and cosine()

$$\cos\theta = \frac{e^{j\theta}+e^{-j\theta}}{2}$$

$$\sin\theta = \frac{e^{j\theta}-e^{-j\theta}}{2j}$$

Inputting Complex Numbers into Matlab

With Matlab, the default value of j is $\sqrt{-1}$. If you redefine j, this no longer holds, but you can restore this as

```
>> j = sqrt(-1)
```

To input a number into Matlab in rectangular form, simply use the j variable

```
>> A = 2 + 3j

A =   2.0000 + 3.0000i
```

You can also input a variable in polar form. $B = 3\angle 1.5$ is input as

```
>> B = 3 * exp(1.5j)

B =   0.2122 + 2.9925i
```

(note: Matlab uses radians for its angle units).

The default display in Matlab is rectangular units. If you want to convert to polar units, use the following commands. Note that the angle is in radians:

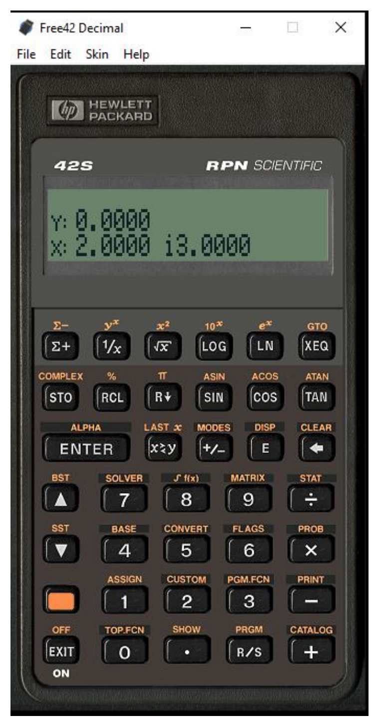
```
>> c = abs(B)

c =     3

>> angle(B)

ans =    1.5000
```

Inputting Complex Numbers on a HP42 (Free42 app)



HP42 calculators use the *COMPLEX* key to input complex numbers. In rectangular mode (yellow - MODES - RECT)

- the y register becomes the real part
- the x register becomes the complex part.

To input the number 2 + j3, press

```
MODES
RECT
2
enter
3
complex
```

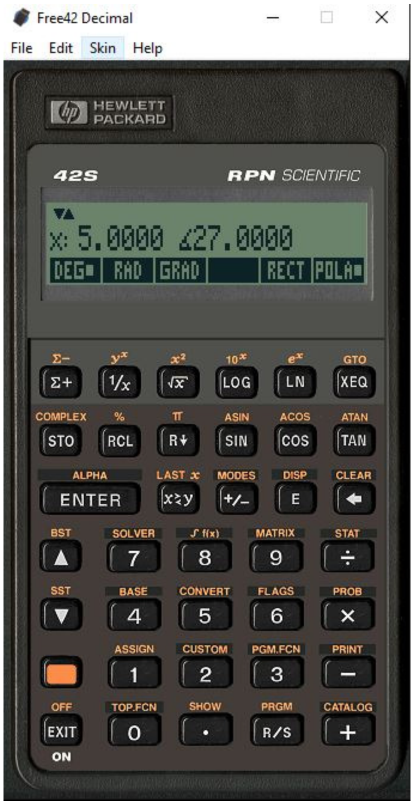
In polar mode (yellow - MODES - POLAR)

- the y register becomes the magnitude

the x register becomes the angle (in the current units degrees / rad / grad)

To input the number

```
MODES
POLAR
5
enter
27
complex
```



HP's manipulate complex numbers with ease. I personally think they're worth ten points on midterms. They're quick. They get the right answer. Both are worth points on midterms.

What calculator you use is your choice, however. Just be sure it is capable of doing operations with complex numbers before you take exams in Circuits I.

Addition, Subtraction, Multiplication, and Division

With complex numbers, you can add, subtract, multiply, and divide just like real numbers:

Addition: For addition

- The real parts add, and
- The complex parts add.

Example:

$$(a_1 + jb_1) + (a_2 + jb_2) = (a_1 + a_2) + j(b_1 + b_2)$$

Subtraction: Again, the real parts subtract and the complex parts subtract

$$(a_1 + jb_1) - (a_2 + jb_2) = (a_1 - a_2) + j(b_1 - b_2)$$

Note: You can also add and subtract complex numbers in polar form. One way to do this is to convert from rectangular form and then add or subtract. That's pretty much what Matlab does. That, or use a calculator which can add or subtract complex numbers, such as the Free42 app on you cell phone.

Multiplication: Multiplication is a little trickier

$$(a_1 + jb_1)(a_2 + jb_2) = a_1 a_2 + ja_1 b_2 + ja_2 b_1 + j^2 b_1 b_2$$

or separating the real and complex results

$$(a_1 + jb_1)(a_2 + jb_2) = (a_1 a_2 - b_1 b_2) + j(a_1 b_2 + a_2 b_1)$$

note: Multiplication is actually easier in polar form

$$r_1 \angle \theta_1 \cdot r_2 \angle \theta_2 = r_1 e^{j\theta_1} \cdot r_2 e^{j\theta_2}$$

$$= r_1 r_2 e^{j\theta_1} e^{j\theta_2}$$

$$= r_1 r_2 \angle (\theta_1 + \theta_2)$$

When you multiply complex numbers

- The magnitude multiplies and
- The angles add

It's even easier if you use a calculator which can multiply complex numbers, like Matlab or the Free42 calculator.

Division: Division with complex numbers is even a little more difficult. The trick is to use the complex conjugate of the denominator. The complex conjugate of a complex number is defined as

$$conj(a + jb) = a - jb$$

Division then becomes

$$\left(\frac{a_1+jb_1}{a_2+jb_2}\right) = \left(\frac{a_1+jb_1}{a_2+jb_2}\right)\left(\frac{a_2-jb_2}{a_2-jb_2}\right)$$

$$= \left(\frac{(a_1a_2+b_1b_2)+j(-a_1b_2+a_2b_1)}{a_2^2+b_2^2}\right)$$

$$= \left(\frac{a_1a_2+b_1b_2}{a_2^2+b_2^2}\right) + j\left(\frac{-a_1b_2+a_2b_1}{a_2^2+b_2^2}\right)$$

Or just use a calculator which can do division with complex numbers, again like Matlab of an HP42. In polar form, division is easier

$$\left(\frac{r_1\angle\theta_1}{r_2\angle\theta_2}\right) = \left(\frac{r_1}{r_2}\right)\angle(\theta_1 - \theta_2)$$

For example, find the result of this operation:

$$y = \left(\frac{(2+j3)(4+j5)+(6+j7)}{8+j9} \right)$$

Solution in Matlab: With Matlab, it's almost the way you write it

```
>> y = ( (2+3j)*(4+5j) + (6+7j) ) / (8+9j)

y =   1.7448 + 1.6621i
```

Another problem: find y

$$y = \left(\frac{\left(2\angle 30^0\right)\left(4\angle 50^0\right)+\left(6\angle 70^0\right)}{8\angle 90^0} \right)$$

In Matlab, this is a little mode clunky

```
>> a = 2 * exp(j*30*pi/180);
>> b = 3 * exp(j*50*pi/180);
>> c = 6 * exp(j*70*pi/180);
>> d = 8 * exp(j*90*pi/180);
>> y = (a*b + c)/d

y =   1.4434 - 0.3868i
```

Partial Fraction Expansion with Real Poles

A common problem in ECE is to expand a function by its roots. For example, find {a, b, c}

$$\left(\frac{2x+3}{(x+1)(x+2)(x+3)} \right) = \left(\frac{a}{x+1} \right) + \left(\frac{b}{x+2} \right) + \left(\frac{c}{x+3} \right)$$

Solution #1: (the hard way) Place the right side over a common denominator and match coefficients.

$$= \left(\frac{a}{x+1} \right)\left(\frac{(x+2)(x+3)}{(x+2)(x+3)} \right) + \left(\frac{b}{x+2} \right)\left(\frac{(x+1)(x+3)}{(x+1)(x+3)} \right) + \left(\frac{c}{x+3} \right)\left(\frac{(x+1)(x+2)}{(x+1)(x+2)} \right)$$

This places all terms over a common denominator. The numerator is then

$$2x+3 = a(x+2)(x+3) + b(x+1)(x+3) + c(x+1)(x+2)$$

$$2x+3 = a(x^2+5x+6) + b(x^2+4x+3) + c(x^2+3x+2)$$

This gives three equations for three unknowns.

Matching the x^2 terms:

$$0 = a+b+c$$

x^1 terms:

$$2 = 5a + 4b + 3c$$

x^0 terms:

$$3 = 6a + 3b + 2c$$

Place in matrix form

$$\begin{bmatrix} 1 & 1 & 1 \\ 5 & 4 & 3 \\ 6 & 3 & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix}$$

Solve in Matlab

```
>> A = [1,1,1;5,4,3;6,3,2]
     1     1     1
     5     4     3
     6     3     2

>> B = [0;2;3]
     0
     2
     3
>> inv(A)*B

a    0.5000
b    1.0000
c   -1.5000
```

so

$$\left(\frac{2x+3}{(x+1)(x+2)(x+3)}\right) = \left(\frac{0.5}{x+1}\right) + \left(\frac{1}{x+2}\right) + \left(\frac{-1.5}{x+3}\right)$$

Solution #2: (cover-up method).

Equals is a powerful symbol: it means both sides are identical everywhere.

- The right side blows up (goes to infinity) near x = {-1, -2, -3}. The left side has to match.

Near x = -1, only the first term matters since it's going to infinity while the other terms are finite. So

$$\lim_{x \to -1} \left(\frac{2x+3}{(x+1)(x+2)(x+3)}\right) = \lim_{x \to -1} \left(\frac{a}{x+1}\right)$$

Cancel (cover up) the (x+1) term and evaluate

$$a = \left(\frac{2x+3}{(x+2)(x+3)}\right)_{x=-1} = 0.5$$

Near x = -2, only the second term (b) matters on the right.

$$\lim_{x \to -2} \left(\frac{2x+3}{(x+1)(x+2)(x+3)} \right) = \lim_{x \to -2} \left(\frac{b}{x+2} \right)$$

Cancel (cover up) the (x+1) term and evaluate

$$b = \left(\frac{2x+3}{(x+1)(x+3)} \right)_{x=-2} = 1$$

Near x = -3, only the third term (c) matters:

$$\lim_{x \to -3} \left(\frac{2x+3}{(x+1)(x+2)(x+3)} \right) = \lim_{x \to -3} \left(\frac{c}{x+3} \right)$$

$$c = \left(\frac{2x+3}{(x+1)(x+2)} \right)_{x=-3} = -1.5$$

Either method works - the cover-up method is a lot easier.

The cover-up method also works in Matlab.

- Take a number close to the point you're evaluating (perturb by 1e-9)
- Solve for {a, b, c}

```
>> x = -1 + 1e-9;
>> a = (2*x + 3)/( (x+1)*(x+2)*(x+3) ) * (x+1)

a =    0.5000

>> x = -2 + 1e-9;
>> b = (2*x + 3)/( (x+1)*(x+2)*(x+3) ) * (x+2)

b =    1.0000

>> x = -3 + 1e-9;
>> c = (2*x + 3)/( (x+1)*(x+2)*(x+3) ) * (x+3)

c =   -1.5000
```

Partial Fraction with Complex Numbers

Placing all terms over a common denominator works, but is that much harder with complex numbers. The cover-up method is the same either way.

Example: Determine {a, b, c}

$$\left(\frac{5x+7}{(x+1+j3)(x+1-j3)(x+5)}\right) = \left(\frac{a}{x+1+j3}\right) + \left(\frac{b}{x+1-j3}\right) + \left(\frac{c}{x+5}\right)$$

Solving

$$a = \left(\frac{5x+7}{(x+1-j3)(x+5)}\right)_{x=-1-j3} = 0.3600 + j0.3533$$

$$b = \left(\frac{5x+7}{(x+1+j3)(x+5)}\right)_{x=-1+j3} = 0.3600 - j0.3533$$

$$c = \left(\frac{5x+7}{(x+1+j3)(x+1-j3)}\right)_{x=-5} = -0.7200$$

Solving using Matlab: A similar way is to use the lim() and pick a point that's a *small* distance from the roots - such as 1e-9. This is a solution similar to the cover-up method.

```
>> x = -1 - j*3 + 1e-9;
>> a = (5*x+7) / ( (x+1+j*3)*(x+1-j*3)*(x+5) ) * (x+1+j*3)

a =   0.3600 + 0.3533i

>> x = -1 + j*3 + 1e-9;
>> a = (5*x+7) / ( (x+1+j*3)*(x+1-j*3)*(x+5) ) * (x+1-j*3)

a =   0.3600 - 0.3533i

>> x = -5 + 1e-9;
>> a = (5*x+7) / ( (x+1+j*3)*(x+1-j*3)*(x+5) ) * (x+5)

a =   -0.7200
```

Summary:

- Complex numbers allow you to represent something with two degrees of freedom
- Addition, subtraction, multiplication, and division all work with complex numbers
- Calculations using complex numbers by hand is really painful.
- Calculations using complex numbers aren't too bad with Matlab or an HP calculator
  - Get familiar with a calculator which does complex numbers before you get to Circuits I
  - You're going to need it for midterms for pretty much all courses in ECE