Digital Signal Processing: COSC390

Jordan Hanson

April 7, 2020

Whittier College Department of Physics and Astronomy

Course Introduction

1. *What is digital signal processing?*
2. *COSC330: Computer Logic and Digital Circuit Design*
3. Read the syllabus for a roadmap
4. *This course can be fast.*
5. **Data science project and presentation**
6. Textbook: `http://dspguide.com`
7. Download and install octave:
   `https://www.gnu.org/software/octave`

Lecture format, with modifications

- Theory and examples
- Programming with Octave
- Application
- Study hall
  1. Homework help
  2. Project and presentation development
  3. Special topics lectures

Unit 1.1 Outline

This lecture will cover:

1. Complex numbers 1: Arithmetic and some calculus (continuous and discete) ... see Chapter 30 of text

Next lectures will cover:

1. Complex numbers 2: The Fourier series and Fourier transform (continuous and discrete)
2. *Time-permitting*: The Laplace transform (continuous and discrete)

Complex numbers 1: theory and examples

Complex numbers 1: Definition of a complex number

A complex number is an expression for which one term is proportional to $j = \sqrt{-1}$:

$$z = x + jy \qquad (1)$$

To call the *complex unit* j is the convention in electrical engineering, and in physics it is often called *i*.

Example of complex numbers: $(3 + 4j)$, $(x_1 + x_2 j)$. Each number has a *real* part and an *imaginary* part.

Complex numbers 1: Definition of a complex number

Operations to learn:

1. Addition
2. Subtraction
3. Real part Re and Im
4. Multiplication
5. Conjugation
6. Magnitude/Norm
7. Division

Notations to learn:

1. Cartesian
2. Polar
3. Graphical

Complex numbers 1: Operations

Addition follows the pattern of two-dimensional vectors:

$$z_1 = 3 + 4j \tag{2}$$

$$z_2 = -2 + 5j \tag{3}$$

$$z_1 + z_2 = 1 + 9j \tag{4}$$

Subtraction follows the pattern of two-dimensional vectors:

$$z_1 = 3 + 4j \tag{5}$$

$$z_2 = -2 + 5j \tag{6}$$

$$z_1 - z_2 = 5 - 1j \tag{7}$$
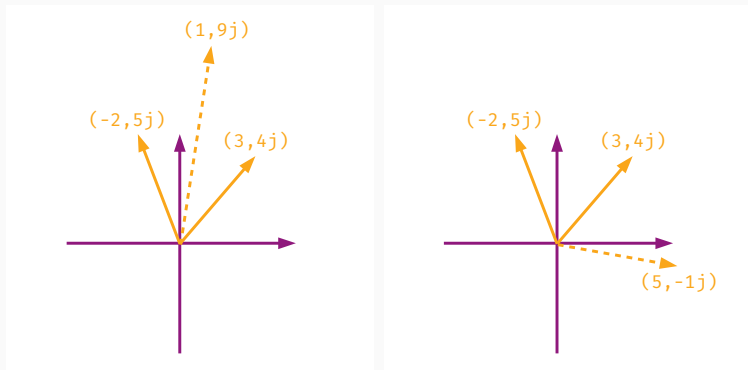
Complex numbers 1: Operations

Figure 1: Complex addition and subtraction follows the pattern of two-dimensional vectors. (Left): Addition of z_1 and z_2. (Right): Subtraction of z_1 and z_2.

Complex numbers 1: Operations

We also have the Re and Im operations:

$$z_1 = 3 + 4j \tag{8}$$

$$\mathrm{Re}\{z_1\} = 3 \tag{9}$$

$$\mathrm{Im}\{z_2\} = 4 \tag{10}$$

These are known as taking the *real*-part and the *imaginary*-part. The original complex number can be recovered by adding real and imaginary parts together:

$$z_1 = \mathrm{Re}\{z_1\} + j\,\mathrm{Im}\{z_1\} \tag{11}$$

When we add/subtract complex numbers, we combine the real parts and imaginary parts separately.

Complex numbers 1: Operations

Do these operators distribute?

$$z_1 = 3 + 4j \tag{12}$$

$$z_2 = 5 + 12j \tag{13}$$

$$\mathrm{Re}\{z_1 + z_2\} = 3 + 4j + 5 + 12j = \mathrm{Re}\{8 + 16j\} \tag{14}$$

$$\mathrm{Re}\{z_1 + z_2\} = 8 = \mathrm{Re}\{z_1\} + \mathrm{Re}\{z_2\} \tag{15}$$

$$\mathrm{Im}\{z_1 + z_2\} = 16 = \mathrm{Im}\{z_1\} + \mathrm{Im}\{z_2\} \tag{16}$$

They distribute because of the associativity of addition.

Complex numbers 1: Operations

Add or subtract, then simplify:

1. $z_1 = 7 + 7j$, $z_2 = -6 + 3j$. $z_1 + z_2 =$
2. $z_1 = 2 + 2j$, $z_2 = 3 - 3j$. $z_1 - z_2 =$
3. $z_1 = 2x + 7j$, $z_2 = 2 + 4xj$. $z_1 + z_2 =$

Let $x = -1$ and $y = 1$. *Evaluate the following expressions:*

1. $z_1 = x + yj$, $z_2 = y + xj$. $z_1 + z_2 =$
2. $z_1 = x^2 + y^2 j$, $z_2 = 2y^2 + x^2 j$. $z_1 - z_2 =$

Draw a y-axis for the imaginary part, and an x-axis for the real part, and graph the prior two exercises. Place points on the graph for z_1, z_2, and $z_1 + z_2$.

Complex numbers 1: Operations

Multiplication: Recall that $j^2 = -1$.

$$z_1 = x_1 + jy_1 \tag{17}$$

$$z_2 = x_2 + jy_2 \tag{18}$$

$$z_1 \times z_2 = x_1x_2 - y_1y_2 + j(x_1y_2 + x_2y_1) \tag{19}$$

The cross-terms are straightforward, but remember the minus sign when multiplying the imaginary parts.

Examples:

1. $z_1 = 7 + 7j$, $z_2 = -6 + 3j$.
   $z_1 \times z_2 = -42 - 21 + j(21 - 42) = -63 - 21j$.
2. $z_1 = 2 + 2j$, $z_2 = 2 - 2j$. $z_1 \times z_2 = 8$. Why no imaginary part?

Complex numbers 1: Operations

Another similarity with two-dimensional vectors?

$$z_1 = 4 - 1j \tag{20}$$

$$z_2 = 1 + 4j \tag{21}$$

$$z_1 \times z_2 = 8 + 15j \neq 0 \tag{22}$$

What would be the result if we were dealing with regular two-dimensional vectors? Plot Eqs. 20 and 21 with the imaginary part as the y-coordinate and the real part as the x-coordinate. Then draw a vector from the origin to the points. Do you recall the properties of the dot-product of two vectors?
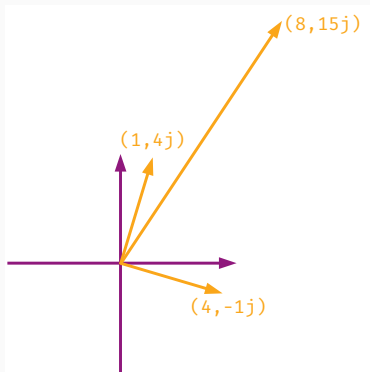
Figure 2: Complex multiplication resembles the *dot-product* for two-dimensional vectors, with key differences.

Complex numbers 1: Operations

Complex conjugation: change the sign of the imaginary part.

$$z_1 = 4 - 1j \tag{23}$$

$$z_1^* = 4 + 1j \tag{24}$$

$$z_2 = = 2x + 1j \tag{25}$$

$$z_2^* = 2x - 1j \tag{26}$$

Is there a significance to $z_1 z_2^*$? What about $z_1 z_1^*$? What about $\sqrt{z_1 z_1^*}$?

What about the taking the complex conjugate of the following expression?

$$z = \frac{x_1 + jy_1}{x_2 + jy_2} \tag{27}$$

Complex numbers 1: Operations

Multiply the denominator *by the complex conjugate of the denominator.*

$$z = \frac{x_1 + jy_1}{x_2 + jy_2} \times \frac{x_1 - jy_2}{x_2 - jy_2} \tag{28}$$

What happens? The real and imaginary parts *separate.* Now prove that

$$z^* = \frac{x_1 - jy_1}{x_2 - jy_2} \tag{29}$$

Let $z = (x_1 + jy_1)(x_2 + jy_2)$. Prove that

$$z^* = (x_1 - jy_1)(x_2 - jy_2) \tag{30}$$

So it seems that complex conjugation boils down to finding all the imaginary units and making them negative...*be careful about assuming this.*
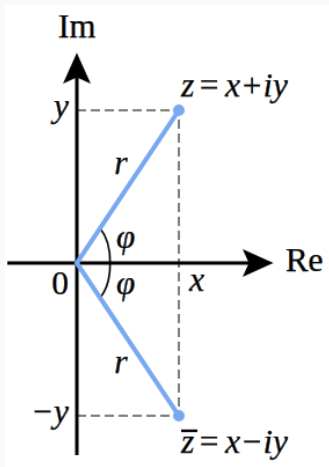
Complex numbers 1: Operations

Figure 3: Complex conjugation flips the location of the complex number if you draw the complex number as a coordinate pair. Stay tuned for polar notation.

Complex numbers 1: Operations

Let $z = x + jy$. Compute the following:

1. $zz^* =$
2. $\sqrt{zz^*} =$

The second item on this list has a special name: the *magnitude* or *norm* of the complex number, $|z|$.

Compute the norm of the following complex numbers:

1. $2 + 2j$
2. $3 + 4j$

Compute the norm. Do either of these simplify?

1. $\sin(x) + j\cos(x)$
2. $\cos(x) + j\sin(x)$

Complex numbers 1: Operations

Division of complex numbers: remember that there are multiple steps.

$$z_1 = x_1 + jy_1 \tag{31}$$

$$z_2 = x_2 + jy_2 \tag{32}$$

$$\frac{z_2}{z_1} = \frac{x_2 + jy_2}{x_1 + jy_1} \tag{33}$$

$$\frac{z_2}{z_1} = \frac{z_2 z_1^*}{z_1 z_1^*} = \frac{z_2 z_1^*}{|z_1|^2} \tag{34}$$

$$\frac{z_2}{z_1} = \frac{\text{Re}\{z_2 z_1^*\}}{|z_1|^2} + j\frac{\text{Im}\{z_2 z_1^*\}}{|z_1|^2} \tag{35}$$

$$\frac{z_2}{z_1} = \frac{x_1 x_2 + y_1 y_2}{x_1^2 + y_1^2} + j\frac{x_1 y_2 - x_2 y_1}{x_1^2 + y_1^2} \tag{36}$$

Using Eq. 36, show that if $z_1 = z_2$, that $z_2/z_1 = 1$.

Complex numbers 1: Operations

Evaluate the divisions below:

1. $z_1 = 1 + 4j$, $z_2 = 2 - 2j$. $z_2/z_1 =$
2. $z_1 = 1 + 1j$, $z_2 = -3 - 3j$. $z_2/z_1 =$
3. $z_1 = 1 + xj$, $z_2 = 1 - xj$, and $x = 0.5$. $z_2/z_1 =$

What if x had been imaginary? Can you see a way to make the division infinity?
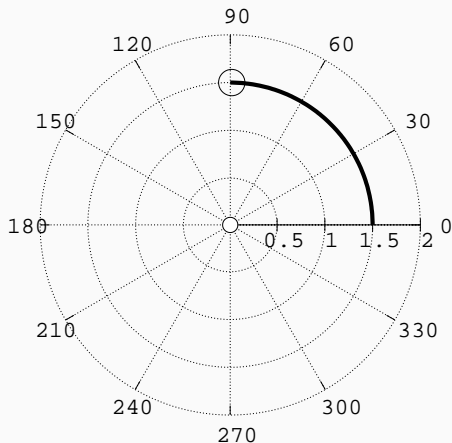
Complex numbers 1: Polar Notation

Figure 4: Polar coordinate systems rely on (ρ, ϕ) notation, rather than (x, y) notation.

Complex numbers 1: Polar Notation

Polar notation for complex numbers: let $z = x + jy$.

$$z = r\exp(j\phi) \tag{37}$$

$$r = |z| = \sqrt{x^2 + y^2} \tag{38}$$

$$\phi = \tan^{-1}(y/x) \tag{39}$$

Useful for multiplication and division:

$$z_1 = r_1 \exp(j\phi_1) \tag{40}$$

$$z_2 = r_2 \exp(j\phi_2) \tag{41}$$

$$z_1 \times z_2 = r_1 r_2 \exp(j(\phi_1 + \phi_2)) \tag{42}$$

$$\frac{r_2}{r_1} = \frac{r_2}{r_1} \exp(j(\phi_2 - \phi_1)) \tag{43}$$

Complex numbers 1: Polar Notation

Convert these complex numbers from Cartesian to polar form:

1. $z_1 = 5 + 13j$
2. $z_2 = 7 - 24j$
3. $z_3 = 20 - 21j$

Divide:

1. z_3/z_2
2. z_3/z_1

Multiply:

1. $z_1 \times z_2$
2. $z_1 \times z_1^*$

Complex numbers 1: Polar Notation

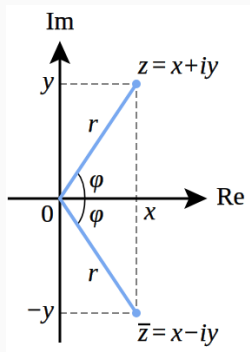How do you perform the complex conjugate in polar form? *Change the sign of the phase angle.*

Figure 5: Complex conjugation flips the sign of the angle.

Complex numbers 1: Polar Notation

Notice that the procedure for finding the modulus is evident in polar notation:

$$\sqrt{z_1 z_1^*} = \sqrt{r_1 r_1} \exp(j(\phi_1 - \phi_1)/2) = r_1 \qquad (44)$$

Shouldn't we be saying $\pm r_1$? How does the square root function work in the complex plane?[1]

---

[1]Complex fractional-power functions are outside the scope of the course, as it turns out, because it requires knowledge of the topic of *branch cuts*.
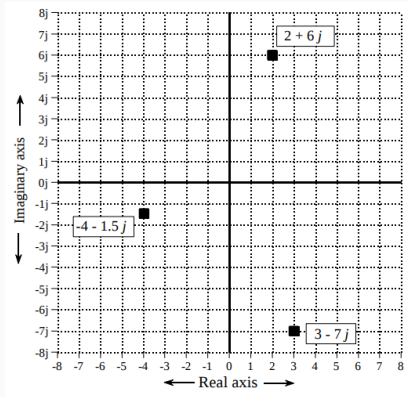
Complex numbers 1: The complex plane

Figure 6: The real and imaginary axes are an *extension* of the real number line, allowing a broader representation of physical systems than just real numbers. A prime example is AC circuits.
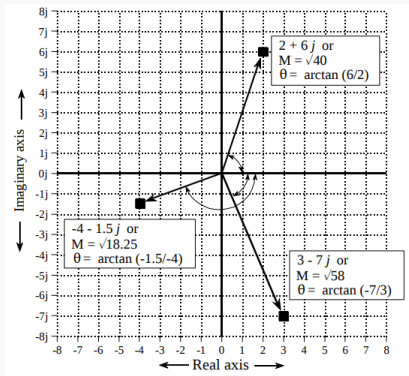
Complex numbers 1: The complex plane

Figure 7: Consider the complex plane in light of trigonometry. Think of the magnitude of a complex number as the hypoteneuse of a right triangle. Then, $\text{Re}\{z\} = |z| \cos \phi$, and $\text{Im}\{z\} = |z| \sin \phi$, and $r = |z| = \sqrt{x^2 + y^2}$.

Complex numbers 1: The complex plane

In the trigonometric picture:

$$z = |z| \cos \phi + j|z| \sin \phi \tag{45}$$

Proof of polar-notation relationship:

$$\exp(j\phi) = \sum_{i=0}^{\infty} \frac{(j\phi)^n}{n!} = \sum_{i=0}^{\infty} \frac{j^n \phi^n}{n!} \tag{46}$$

$$\exp(j\phi) = \sum_{even}^{\infty} \frac{j^n \phi^n}{n!} + \sum_{odd}^{\infty} \frac{j^n \phi^n}{n!} \tag{47}$$

$$\exp(j\phi) = \sum_{i=0}^{\infty} (-1)^n \frac{\phi^{2n}}{(2n)!} + j \sum_{i=0}^{\infty} \frac{\phi^{2n+1}}{(2n+1)!} \tag{48}$$

$$\exp(j\phi) = \cos \phi + j \sin \phi \tag{49}$$

$$|z| \exp(j\phi) = |z| \cos \phi + j|z| \sin \phi = z \tag{50}$$

27

Complex numbers 1: The complex plane

Thus we have proven:

$$\boxed{z = x + jy = r\exp(i\phi)} \tag{51}$$

Corollary (prove these):

$$\cos(x) = \frac{1}{2}\left(e^{ix} + e^{-ix}\right) \tag{52}$$

$$\sin(x) = \frac{1}{2i}\left(e^{ix} - e^{-ix}\right) \tag{53}$$

Suppose we have now have a voltage signal

$$v_i(t) = A_i\cos(2\pi f_i t + \phi_i) \tag{54}$$

We may write

$$v_i(t) = A_i\,\mathrm{Re}\{\exp(j(2\pi f_i t + \phi_i))\} \tag{55}$$

Complex numbers 1: The complex plane

What if we treat the signal as complex, but agree to take the real part at the end of our calculations?

$$v_i(t) \rightarrow A_i \exp(j(2\pi f_i t + \phi_i)) \tag{56}$$

As long as we take the real part of the right hand side, we'll have the original signal.

Now contemplate the addition of signals of the same frequency, but different amplitudes and phases. Let $x_i = 2\pi f t + \phi_i$. A signal comprised of N sinusoids can be written

$$V_i(t) \rightarrow \sum_i^N a_i \exp(jx_i) \tag{57}$$

Complex numbers 1: The complex plane

Remember that $x_i = 2\pi f t + \phi_i$. The sum of two sinusoids in the complex plane can then be written[2]

$$V(t) = a_1 \exp(jx_1) + a_2 \exp(jx_2) \tag{58}$$

1. Compute $|V|^2 = V^* V$, and $\phi_2 - \phi_1 = \pi$, $\phi_2 - \phi_1 = 0$.
2. What is $\phi_V = \tan^{-1}(\text{Im}\{V\} / \text{Re}\{V\})$ in each case?

Why do these results make sense? Thus, the complex numbers encapsulate the concepts of *constructive* and *destructive* interference.

---

[2]Notice that taking the real part distributes if the original signal is real.

Complex numbers 1: The complex plane

Treating sinusoids as rotation in the complex plane is a deep subject in physics. It actually relates to some concepts in introductory physics:
https://www.youtube.com/watch?v=jxstE6A_CYQ

What is the effect of multiplying by a complex number as follows:

$$V_i(t) \exp(j\phi') = ? \tag{59}$$

What does this look like when graphed?

Complex numbers 1: The complex plane

What if we continue to add terms, putting in carefully chosen phases and amplitudes? We can represent *any* periodic signal[3].

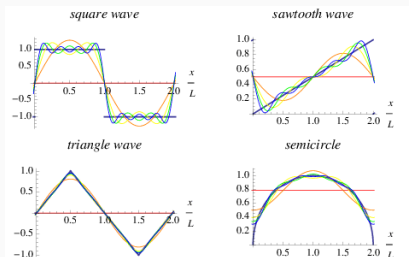This is known as the **Fourier series**:

Figure 8: The Fourier series representing four different periodic functions.

---

[3]We will return to this in Unit 1.2.

Complex numbers 1: programming with Octave

Complex numbers 1: programming with Octave

Let's take the time to get octave installed on your systems:
https://www.gnu.org/software/octave.

If we cannot get it installed on your systems, we can always run it on the local desktops.

A good tutorial can be found here:
https:
//en.wikibooks.org/wiki/Octave_Programming_Tutorial

Complex numbers 1: programming with Octave

Octave is a high-level *scripting* programming language. Although it is possible to write packages and compile code in octave, the most common application is executing a script that performs some analysis on digital data.

```
a = 1+1i;
b = conj(a);
a * b
```

The result of this code should be 2.0. Why? We are defining a complex number in the first line, computing the complex conjugate, and multiplying them.

Complex numbers 1: programming with Octave

Octave naturally handles vectors of numbers and matrices. Let's define a vector of complex numbers.

```
a = [1 2 3 5 7 11];
size(a)
ans = 1    6
a = a';
size(a)
```

The code in the fourth line *transposes* the vector. This means trading the rows for the columns of the vector. What begins as a 1×6 vector (one row by six columns) ends as a 6×1 vector (six rows by one column).

Complex numbers 1: programming with Octave

Operations are as expected, but we need special notation for vectoral calculations:

```
a = 2.0;
b = 4.0;
b/a
ans = 2
b = [4.0 4.0 4.0]
b/a
ans =
2   2   2
a./b
ans =
0.5   0.5   0.5
```

Complex numbers 1: programming with Octave

Placing a dot (.) before a standard operation indicates that the operation is to be carried out in a vectoral-sense.

```
t1 = [1 2 3 4 5 6 7 8 9 10];
t2 = t1+1;
t1.*t2
ans =
2   6   12   20   30   42   56   72   90   110
```

Complex numbers 1: programming with Octave

The colon operator (:) represents iteration in octave. Consider three cases:

```
fs = 1000.0;
t = [0.0:1/fs:10.0];
plot(t,sin(2.0 * pi * 3.0 .* t));
```

Octave should produce a plot of a sine wave with a frequency of 3.0 Hz (if the time is in seconds). How can you tell?

Complex numbers 1: programming with Octave

Count the number of complete oscillations in 2.0 seconds. Do you see 6.0 oscillations? What is the significance of f_s in the code?

```
fs = 1000.0;
t = [0.0:1/fs:10.0];
plot(t,sin(2.0 * pi * 3.0 .* t));
```

The axis command is useful for controlling the plotted region:

```
axis([0 2 -2 2]);
```

Complex numbers 1: programming with Octave

The colon operator (:) also refer to elements in a vector.

```
t(1)
ans =
0
t(2:5)
ans =
0.001   0.002   0.003   0.004
t(:)
ans =
(should be the whole vector)
t(1:end)
ans =
(should be the whole vector)
```

Complex numbers 1: programming with Octave

Now you know how to create vectors and plot them in octave. Let's solve a few programming problems! First, let's introduce the help command.

```
help <command>
```

Type the following commands

1. help plot
2. help :

What is the purpose of the *clear* and *home* commands?

The key here is that if you are confused, the help command can boost your understanding of the correct usage. Speaking of the colon command, how do we create matrices?

Complex numbers 1: programming with Octave

Create a matrix in octave in one of three ways:

1. Concatenate row vectors
2. Concatenate colomn vectors
3. Write it straight away

First, concatenation of row vectors:

```
a = [1:10];
b = [11:20];
A = [a; b];
display(A)
```

The semi-colon also tells the [] operators to concatenate vertically. What is the size of the matrix?

Complex numbers 1: programming with Octave

Create a matrix in octave in one of three ways:

1. Concatenate row vectors
2. Concatenate colomn vectors
3. Write it straight away

First, concatenation of row vectors:

```
a = [1:10]';
b = [11:20]';
A = [a b];
display(A)
```

The lack of the semi-colon tells the [] operators to concatenate
horizontally. What is the size of the matrix?

Complex numbers 1: programming with Octave

Create a matrix in octave in one of three ways:

1. Concatenate row vectors

2. Concatenate colomn vectors

3. Write it straight away

First, concatenation of row vectors:

```
A = [[1 2]; [3 4];];
display(A)
```

In this case, you can actually omit the internal concatenation operators.
Can you guess how to take the transpose of the matrix?

Complex numbers 1: programming with Octave

Create a matrix with complex numbers: First, concatenation of row vectors, and take the transpose. Do you notice something?

```
A = [[1 2i]; [3i 4];];
display(A)
```

Try using the *help* command on the transpose operator to sort out the issue. Now look up the *conj* command, and use it to take *just the transpose* of the matrix, without conjugation.

Complex numbers 1: programming with Octave

Two additional commands that are helpful for creating square
($N \times N$) matrices:

```
ones(N,M)
zeros(N,M)
```

Create a 2×2 matrix and multiply it with a 2×1 vector. *Careful with the order; the matrix has to be on the left.*

You can imagine a matrix as an **operator** that acts on a vector.
Example: two-dimensional rotation matrix (observe on board).

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{60}$$

Complex numbers 1: programming with Octave

Code the rotation matrix (Eq. 60) for $\theta = \pi/4$, and also create three 2×1 column vectors. Multiply them and display the results. Plot them on the page and observe where they have moved.

Let's take the opportunity to share code on Moodle. **See the Rotation2D example in Unit1.**

Octave *scripts* can be run within the octave environment. I prefer octave-cli (*command-line interface*), because it resembles a bash shell in linux systems. The graphical user interface (octave-gui) can display folders, scripts, and the *workspace.* The workspace contains persistent variables.

Complex numbers 1: application to AC circuits

Complex numbers 1: application to AC circuits

We touched on complex numbers in two contexts:

- Mathematical formalism
- Programming octave

Now let's apply them to AC circuits. Consider three circuit elements: the resistor, capacitor, and inductor. They come with three symbols:
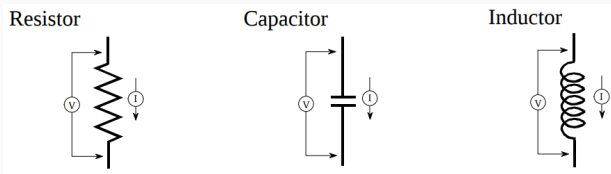

Figure 9: These three symbols can be combined in AC circuit diagrams to describe the flow of current and power.

Complex numbers 1: application to AC circuits

We also need the concept of Ohm's law in the AC context:

$$V(\omega) = i(\omega)Z(\omega) \tag{61}$$

The symbol Z represents the *complex resistance*, or *impedance* of the circuit element. Clearly, resistance can't be complex, but we think of the phase shift introduced to the current as being carried by a complex number.

$$Z_R = R + 0i \tag{62}$$

$$Z_C = 0 + \frac{1}{j\omega C} \tag{63}$$

$$Z_L = 0 + j\omega L \tag{64}$$

To *prove* these equations, we'll need the *Fourier transform*, and some circuit analysis (future lectures). For now, think of them as just AC elements with (potentially) complex resistance.

Complex numbers 1: application to AC circuits

Consider the circuit in Fig. 10. If we think of V_{in} as a sinudoidal signal, and V_{out} as another sinusoidal signal, we can use complex numbers to compute what will happen at a given frequency ω.
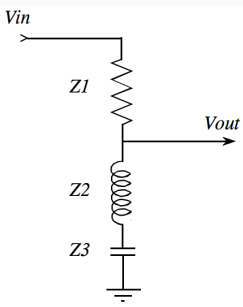

Figure 10: An example of an *RLC* circuit.

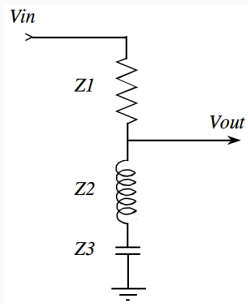## Complex numbers 1: application to AC circuits

Figure 11: Using $V(\omega) = i(\omega)Z(\omega)$, we can solve for the ratio V_{out}/V_{in}.

Pretend that each AC element is some general element, and use Ohm's law to get the ratio V_{out}/V_{in} (observe on board).

Complex numbers 1: application to AC circuits

We should find something like this:

$$h(\omega) = \frac{Z_2 + Z_3}{Z_1 + Z_2 + Z_3} \tag{65}$$

$$\omega_{LC}^{-2} = LC \tag{66}$$

$$\tau = RC \tag{67}$$

$$k^2 = 1 - \left(\frac{\omega}{\omega_{LC}}\right)^2 \tag{68}$$

$$h(\omega) = \frac{k^4}{k^4 + (\omega\tau)^2} - j\frac{k^2\omega\tau}{k^4 + (\omega\tau)^2} \tag{69}$$

Notice that for $\omega = \omega_{LC}$, $k = 0$. What happens to $h(\omega)$ in that case? What are the limits for $\omega \ll \omega_{LC}$ and $\omega \gg \omega_{LC}$?
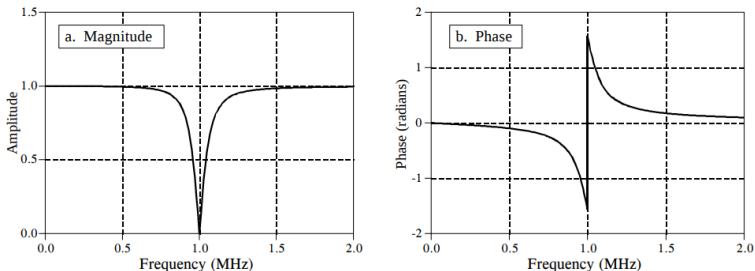
Complex numbers 1: application to AC circuits

FIGURE 30-7
Notch filter frequency response. These curves are for the component values: R = 50Ω, C = 470pF, and L = 54 μH.

Figure 12: The magnitude and phase of the quantity V_{out}/V_{in} are plotted versus frequerncy $f = \omega/(2\pi)$ for specific values of resistance, capacitance, and inductance.

Complex numbers 1: application to AC circuits

Today's octave application: write the script file *RLC.m*, that reproduces Fig. 12 for given values of R, L, and C.

- Write the script in a file entitled RLC.m
- To run it, open octave-cli in the directory in which RLC.m is stored. Type RLC, then hit enter.
- We will cover plotting techniques on-the-fly, since plotting and markup could be an entire course.
- Tweak the values of the AC elements and observe how your plot changes. Why is this circuit one example of an *RLC notch filter*?

Conclusion

Unit 1.1 Outline

Unit 1.1 covered:

1. Complex numbers 1: Arithmetic and some calculus (continuous and discete) ... see Chapter 30 of text
   - Complex number operations
   - Graphical knowledge of complex numbers
   - Octave coding
   - The RLC notch filter.

Next lectures will cover:

1. Complex numbers 2: The Fourier series and Fourier transform (continuous and discrete)

2. *Time-permitting*: The Laplace transform (continuous and discrete)