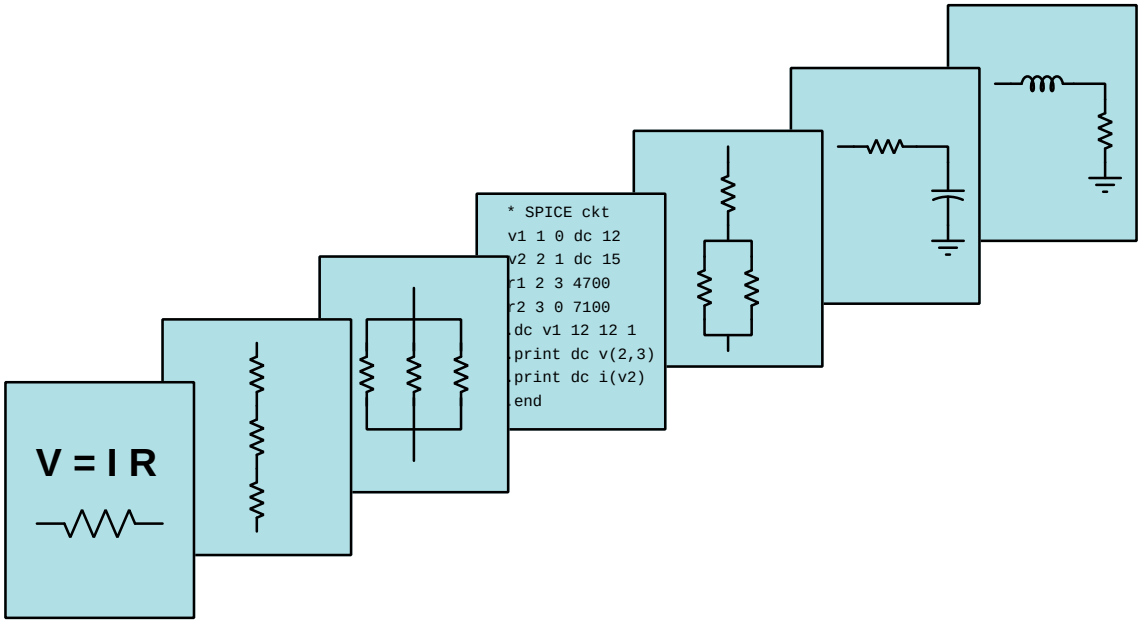
Modular Electronics Learning (ModEL) project



Series-Parallel AC Circuits

© 2018-2026 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License

Last update = 19 February 2026

This is a copyrighted work, but licensed under the Creative Commons Attribution 4.0 International Public License. A copy of this license is found in the last Appendix of this document. Alternatively, you may visit `http://creativecommons.org/licenses/by/4.0/` or send a letter to Creative Commons: 171 Second Street, Suite 300, San Francisco, California, 94105, USA. The terms and conditions of this license allow for free copying, distribution, and/or modification of all licensed works by the general public.

Contents

1 Introduction **3**
- 1.1 Recommendations for students . . . 3
- 1.2 Challenging concepts related to series-parallel AC circuits . . . 5
- 1.3 Recommendations for instructors . . . 7

2 Case Tutorial **9**
- 2.1 Example: series-parallel RLC circuit . . . 10
- 2.2 Example: sine versus non-sine AC sources . . . 12

3 Simplified Tutorial **17**

4 Full Tutorial **21**
- 4.1 Phasor analysis review . . . 22
- 4.2 Series and parallel network properties . . . 26
- 4.3 AC series-parallel analysis example . . . 29

5 Derivations and Technical References **35**
- 5.1 Equivalent series and parallel XR networks . . . 36
- 5.2 Complex-number arithmetic . . . 42
  - 5.2.1 Negating complex numbers . . . 43
  - 5.2.2 Adding complex numbers . . . 43
  - 5.2.3 Subtracting complex numbers . . . 43
  - 5.2.4 Multiplying complex numbers . . . 44
  - 5.2.5 Dividing complex numbers . . . 44
  - 5.2.6 Reciprocating complex numbers . . . 45
  - 5.2.7 Calculator tips . . . 45
- 5.3 Conductance, admittance, and susceptance . . . 46

6 Programming References **49**
- 6.1 Programming in C++ . . . 50
- 6.2 Programming in Python . . . 54
- 6.3 Modeling series-parallel RLC networks using C++ . . . 59

7 Questions **65**
7.1 Conceptual reasoning . . . 69
7.1.1 Reading outline and reflections . . . 70
7.1.2 Foundational concepts . . . 71
7.1.3 Headphone connections . . . 73
7.1.4 Explaining the meaning of calculations . . . 74
7.2 Quantitative reasoning . . . 76
7.2.1 Miscellaneous physical constants . . . 77
7.2.2 Worked example . . . 78
7.2.3 Series and parallel network impedances . . . 80
7.2.4 Series-parallel network impedance . . . 81
7.2.5 Equivalent RC networks . . . 82
7.2.6 Equivalent LR networks . . . 83
7.2.7 Current through an arbitrary AC load . . . 83
7.2.8 Output voltage of an RC network . . . 84
7.2.9 Voltage drop in an RC sub-network . . . 84
7.2.10 Current through an LR branch . . . 85
7.2.11 Series and parallel equivalent networks given impedances . . . 85
7.2.12 Series-parallel circuit tables . . . 86
7.2.13 Voltmeter range extender . . . 88
7.2.14 Induction motor model . . . 90
7.3 Diagnostic reasoning . . . 91
7.3.1 Determining possible faults in series-parallel circuits . . . 92

A Problem-Solving Strategies **95**

B Instructional philosophy **97**
B.1 First principles of learning . . . 98
B.2 Proven strategies for instructors . . . 99
B.3 Proven strategies for students . . . 101
B.4 Design of these learning modules . . . 102

C Tools used **105**

D Creative Commons License **109**

E References **117**

F Version history **119**

Index **120**

Chapter 1

Introduction

1.1 Recommendations for students

The analysis of AC series-parallel circuits is quite similar to that of DC series-parallel circuits: in both cases the analysis relies on a firm understanding of basic laws (Ohm's Law, Kirchhoff's Laws) and network properties (series versus parallel). The major novelty with AC series-parallel circuits is that all variables must be expressed and computed as *complex numbers* (phasors), each one having a magnitude and phase angle (in polar form) or a real and imaginary part (in rectangular form). Additionally, power takes on different meanings as we encounter circuits with both dissipative and reactive elements.

Important concepts related to AC series-parallel circuits include **DC** versus **AC** electricity, **phasors**, **frequency**, **phase shift**, **apparent power**, **true power**, **reactive power**, **Kirchhoff's Voltage Law**, **Kirchhoff's Current Law**, properties of **series** networks, properties of **parallel** networks, **sources** versus **loads**, complex numbers in **polar** and **rectangular** forms, **Conservation of Energy**, **Conservation of Charge**, diagram **annotation**, and **polarity**.

A very important problem-solving strategy applied throughout the Tutorials is that of *annotating diagrams* with labels showing voltages, polarities, and currents (with direction). Mapping calculated values onto these diagrams is a helpful way to avoid confusion and to maintain proper context for all the quantities. Another important point about problem-solving is the need for *patience*, and the willingness to proceed with calculations even if the final strategy for solving the problem eludes one's immediate grasp.

When reading any mathematically-based presentation, a useful habit for effective learning is to actually perform the mathematics being shown in the text. Don't just passively read what the text tells you and trust that the math works – try the math for yourself. Not only will this serve to confirm what you are reading, but it is also an excellent way to practice those mathematical techniques.

Here are some good questions to ask of yourself while studying this subject:

- How might an experiment be designed and conducted to demonstrate the properties of series

electrical networks? What hypotheses (i.e. predictions) might you pose for that experiment, and what result(s) would either support or disprove those hypotheses?

- How might an experiment be designed and conducted to demonstrate the properties of parallel electrical networks? What hypotheses (i.e. predictions) might you pose for that experiment, and what result(s) would either support or disprove those hypotheses?
- What does it mean to say that an AC quantity has a *frequency*?
- How does true power differ from apparent or reactive power in an AC circuit?
- Which principles from DC circuit analysis still apply to AC circuits?
- Which principles from DC circuit analysis do not apply to AC circuits?
- Why do we say that the phase angle of an inductor's impedance is positive 90 degrees?
- Why do we say that the phase angle of a capacitor's impedance is negative 90 degrees?
- What defines a *series* connection between two or more components?
- What defines a *parallel* connection between two or more components?
- How may we explain all the properties of series networks in terms of more fundamental principles?
- How may we explain all the properties of parallel networks in terms of more fundamental principles?
- How do we analyze a series-parallel network step-by-step?
- What is a "phasor diagram" and what does it show us?
- Why is it a good practice to store all calculated values in memory, rather than re-type those values into your calculator when needed for other calculations?
- What are some good ways to check our mathematical work when finishing a complex circuit-analysis problem?

1.2 Challenging concepts related to series-parallel AC circuits

The following list cites concepts related to this module's topic that are easily misunderstood, along with suggestions for properly understanding them:

- **Defining a series network** – series networks are defined by having a single path for current between the components. From this definition, and from the Conservation of Electric Charge, follows the conclusion that series-connected components must experience the same amount of continuous current. One challenge is that sometimes series-connected components do not physically or pictorially align to suggest a single path, even though they have but one *electrical* path for current. Another challenge is reasoning logically from this definition of "one path" to the necessary conclusion of "same current", as many students are so accustomed to memorizing facts without seeing the logical connections between those facts.

- **Defining a parallel network** – parallel networks are defined by all components spanning the same two sets of electrically-common points. From this definition, and from the Conservation of Energy, follows the conclusion that parallel-connected components must experience the same amount of continuous voltage. One challenge is that sometimes parallel-connected components do not physically or pictorially align in a parallel fashion, even though they are in fact parallel in the *electrical* sense. Another challenge is reasoning logically from this definition of "between the same two sets of electrically-common points" to the necessary conclusion of "same voltage", as many students are so accustomed to memorizing facts without seeing the logical connections between those facts. A helpful way to both identify parallel connections as well as to conceptualize the commonality of voltage is to use color-highlighting to mark all electrically-common points with the same color. This way, all points in one set have one color, all points in the other set have another color, and all components connecting to points with those two colors must be connected in parallel with each other. This coloring approach is used in the Tutorial chapters where points A-B-C-D are all colored blue and points E-F-G-H are all colored green. The *Case Tutorial* section "Example: potential in open and shorted circuits" is a helpful reference for using color-highlighting to denote different amounts of electrical potential in a circuit.

- **Carelessness with series-parallel analysis** – when analyzing a series-parallel circuit to determine component voltages and currents, one must be very careful to reduce the circuit step-by-step, section-by-section, into equivalent resistances. Students new to this process typically see all the work that is involved and try to save effort by taking shortcuts, which paradoxically causes *more* work and *more* confusion for them later on. Use lots of paper to document your work when you reduce series-parallel resistor networks to equivalent resistances, re-drawing the circuit for each reduction. This helps reduce the number of mistakes, and also makes it easier to transfer calculated values of voltage and current to the correct components (see the challenging concept shown above: using Ohm's Law in context).

- **Phasors representing AC amplitudes and phase shifts** – a powerful tool used for understanding the operation of AC circuits is the *phasor diagram*, consisting of arrows pointing in different directions: the length of each arrow representing the amplitude of some AC quantity (voltage, current, or impedance), and the angle of each arrow representing the shift

in phase relative to the other arrows. By representing each AC quantity thusly, we may more easily calculate their relationships to one another, with the phasors showing us how to apply trigonometry (Pythagorean Theorem, sine, cosine, and tangent functions) to the various calculations. An analytical parallel to the graphic tool of phasor diagrams is *complex numbers*, where we represent each phasor (arrow) by a pair of numbers: either a magnitude and angle (polar notation), or by "real" and "imaginary" magnitudes (rectangular notation). Where phasor diagrams are helpful is in applications where their respective AC quantities *add*: the resultant of two or more phasors stacked tip-to-tail being the mathematical sum of the phasors. Complex numbers, on the other hand, may be added, subtracted, multiplied, and divided; the last two operations being difficult to graphically represent with arrows.

- **Complex numbers in calculators** – while the ability of certain scientific calculators to perform complex-number arithmetic is an enormously helpful tool for students first learning to analyze AC circuits, some of these calculators prove to be finicky in their handling and entry of these quantities. Advice proven to be sound for all complex-number calculators is to save each and every complex-valued quantity into a memory location and then perform arithmetic operations on those stored variables rather than enter the complex numbers directly into the computation. For example, storing `3 - j4` into memory location `A` and `25` $\angle$ `30`^{o} into memory location `B`, then multiplying `A` $\times$ `B` rather than entering `3 - j4` $\times$ `25` $\angle$ `30`^{o}. Storing values into calculator memory and then retrieving them as needed for calculations is actually sound advice for many reasons, but many students resist taking these "extra" steps and as a result incur all the risks of hand-entering values (e.g. rounding errors due to truncating, keystroke errors when the same value must be used more than once, crowded displays where you cannot see the whole calculation, order-of-operations errors when complex numbers aren't enclosed in parentheses, etc.). Complex-number calculations reward good practices through consistently good results!

- **Complex numbers in measurement** – complex numbers may be expressed in either *rectangular* or *polar* form, either one of these being perfectly valid. However, measurement instruments such as multimeters only provide the magnitude of the polar form of the voltage or current in question. For example, if a component's voltage is 4.8 Volts RMS $\angle$ 35^o with the circuit's source voltage being the phase reference (0^o), an voltmeter reading that voltage will simply register 4.8 Volts RMS. An oscilloscope simultaneously measuring that component voltage on one channel and the source voltage on another will show the 35^o shift on the horizontal axis between the two waveforms.

- **Resistance versus Reactance versus Impedance** – these three terms represent different forms of opposition to electric current. Despite the fact that they are measured in the same unit (ohms: Ω), they are not the same concept. Resistance is best thought of as electrical *friction*, whereas reactance is best thought of as electrical *inertia*. Whereas resistance creates a voltage drop by dissipating energy, reactance creates a voltage drop by *storing* and *releasing* energy. Impedance is a term encompassing both resistance and reactance, usually a combination of both.

The *Case Tutorial* chapter contains worked problems for series-parallel AC circuits, showing solutions using complex numbers as well as solutions using scalar calculations.

1.3 Recommendations for instructors

This section lists realistic student learning outcomes supported by the content of the module as well as suggested means of assessing (measuring) student learning. The outcomes state what learners should be able to do, and the assessments are specific challenges to prove students have learned.

- **Outcome** – Demonstrate effective technical reading and writing

  Assessment – Students present their outlines of this module's instructional chapters (e.g. Case Tutorial, Tutorial, Historical References, etc.) ideally as an entry to a larger Journal document chronicling their learning. These outlines should exhibit good-faith effort at summarizing major concepts explained in the text.

  Assessment – Students show how quantitative results were obtained by the author in the Tutorial chapter's examples.

  Assessment – Students present their "Conceptual Toolbox" Theory Journal entries listing important concepts applied in the Tutorial text(s). A "Conceptual Toolbox" serves as inspiration when solving problems, each concept suggesting a potential problem-solving action. Challenge students to collaboratively generate a Conceptual Toolbox for the day's topic during the class session and then use it when solving problems together, thus fostering the habit of reasoning from principles rather than recalling procedures!

- **Outcome** – Apply the concepts of reactance and impedance to series-parallel AC circuits

  Assessment – Calculate all component voltages and currents in a series-parallel AC circuit given component values.

  Assessment – Sketch phasor diagrams of all component voltages and currents in a series-parallel AC circuit given component values.

- **Outcome** – Apply the concept of power factor to realistic circuits

  Assessment – Calculate the various powers (P, Q, S) and/or power factor for a series-parallel AC circuit given component values.

- **Outcome** – Independent research

  Assessment – Read and summarize in your own words reliable historical documents on the subject of applying complex numbers to AC circuit calculations. Recommended readings include books written by Charles Proteus Steinmetz.

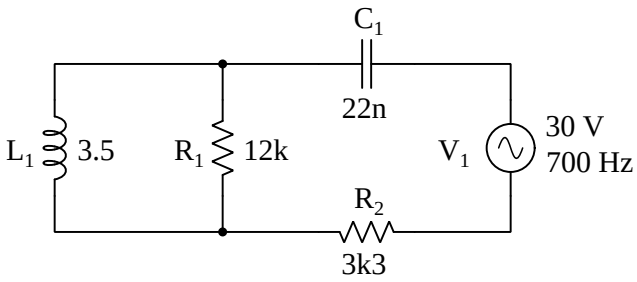
Chapter 2

Case Tutorial

The idea behind a *Case Tutorial* is to explore new concepts by way of example. In this chapter you will read less presentation of theory compared to other Tutorial chapters, but by close observation and comparison of the given examples be able to discern patterns and principles much the same way as a scientific experimenter. Hopefully you will find these cases illuminating, and a good supplement to text-based tutorials.

These examples also serve well as challenges following your reading of the other Tutorial(s) in this module – can you explain *why* the circuits behave as they do?

2.1 Example: series-parallel RLC circuit



Phasor calculations

$R_1 = 12\ \text{k}\Omega \qquad X_{L1} = 2\pi f L_1 = (2\pi)(700)(3.5) = 15.394\ \text{k}\Omega$

$R_2 = 3.3\ \text{k}\Omega \qquad X_{C1} = \frac{1}{2\pi f C_1} = \frac{1}{(2\pi)(700)(22\times10^{-9})} = 10.335\ \text{k}\Omega$

$Z_{R1} = 12\ \text{k}\Omega \angle\ 0^o \qquad Z_{L1} = 15.394\ \text{k}\Omega \angle\ 90^o$

$Z_{R2} = 3.3\ \text{k}\Omega \angle\ 0^o \qquad Z_{C1} = 10.335\ \text{k}\Omega \angle\ -90^o$

$$Z_{L1||R1} = \frac{1}{\frac{1}{Z_{R1}}+\frac{1}{Z_{L1}}} = \frac{1}{\frac{1}{12000\angle 0^o}+\frac{1}{15394\angle 90^o}} = 9.464\ \text{k}\Omega \angle\ 37.94^o$$

$$Z_{total} = Z_{L1||R1} + Z_{C1} + Z_{R2} = (9464\angle 37.94^o) + (10335\angle -90^o) + (3300\angle 0^o) = 11.67\ \text{k}\Omega \angle -22.76^o$$

$$I_{total} = \frac{V_1}{Z_{total}} = \frac{30\angle 0^o}{11673\angle -22.76^o} = 2.5670\ \text{mA} \angle\ 22.76^o$$

$$V_{C1} = I_{total} Z_{C1} = (2.5670\times10^{-3}\angle 22.76^o)(10335\angle -90^o) = 26.56\ \text{V} \angle\ -67.24^o$$

$$V_{R2} = I_{total} Z_{R2} = (2.5670\times10^{-3}\angle 22.76^o)(3300\angle 0^o) = 8.4810\ \text{V} \angle\ 22.76^o$$

$$V_{L1||R1} = I_{total} Z_{L1||R1} = (2.5670\times10^{-3}\angle 22.76^o)(9464\angle 37.94^o) = 24.32\ \text{V} \angle\ 60.70^o$$

$I_{C1} = I_{total} = 2.5670\ \text{mA} \angle\ 22.76^o$

$I_{R2} = I_{total} = 2.5670\ \text{mA} \angle\ 22.76^o$

$$I_{L1} = \frac{V_{L1||R1}}{Z_{L1}} = \frac{24.32\angle 60.70^o}{15394\angle 90^o} = 1.5800\ \text{mA} \angle\ -29.30^o$$

$$I_{R1} = \frac{V_{L1||R1}}{Z_{R1}} = \frac{24.32\angle 60.70^o}{12000\angle 0^o} = 2.0270\ \text{mA} \angle\ 60.70^o$$

SPICE netlist

```
* Series-parallel RLC circuit
v1 1 0 ac 30
c1 1 2 22n
r1 2 3 12000
l1 2 3 3.5
r2 3 0 3300
.control
set units=degrees
.endc
.ac lin 1 700 700
.print ac vm(1,2) vp(1,2)
.print ac vm(2,3) vp(2,3)
.print ac vm(3) vp(3)
.end
```

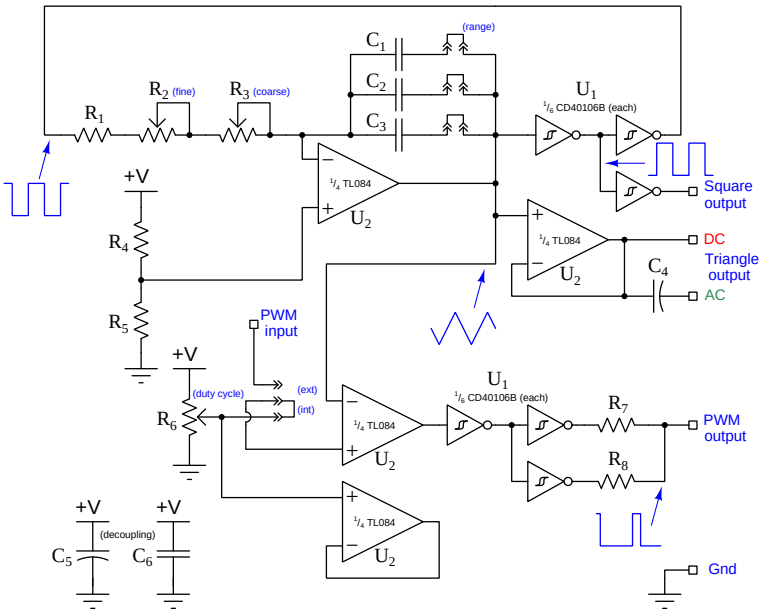
NGSPICE simulation results

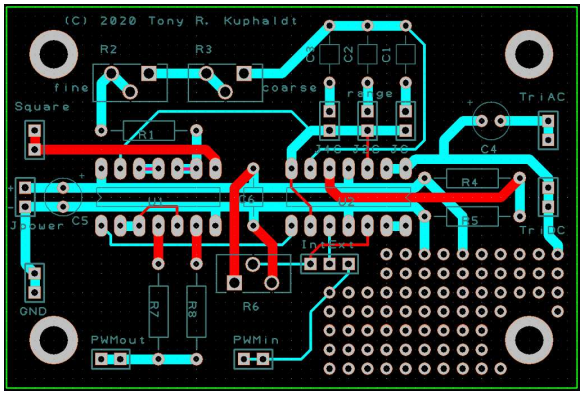
```
--------------------------------------------------------------------------------
Index   frequency       mag(v(1)-v(2))  ph(v(1)-v(2))
--------------------------------------------------------------------------------
0       7.000000e+02    2.656017e+01   -6.72394e+01

--------------------------------------------------------------------------------
Index   frequency       mag(v(2)-v(3))  ph(v(2)-v(3))
--------------------------------------------------------------------------------
0       7.000000e+02    2.432281e+01    6.069820e+01

--------------------------------------------------------------------------------
Index   frequency       vm(3)           vp(3)
--------------------------------------------------------------------------------
0       7.000000e+02    8.480968e+00    2.276055e+01
```

2.2 Example: sine versus non-sine AC sources

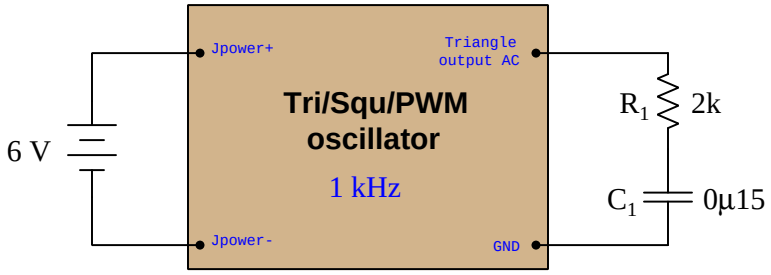
Students learning to analyze RLC networks powered by AC voltage sources typically rely on expensive signal generators to produce the pure sine-wave AC excitation voltage necessary for voltmeter measurements to closely match predictions. However, robust triangle-wave oscillator circuits are much less complicated to design and build than sine-wave oscillator circuits, so if students wish to build their own signal generators for these introductory AC experiments it is good to know that triangle-wave excitation yields results very close to sine-wave excitation.

A simple and versatile signal generator circuit appears below, outputting triangle, square, and PWM (pulse-width-modulated) signals. The first image is the schematic diagram, followed by a PCB layout:





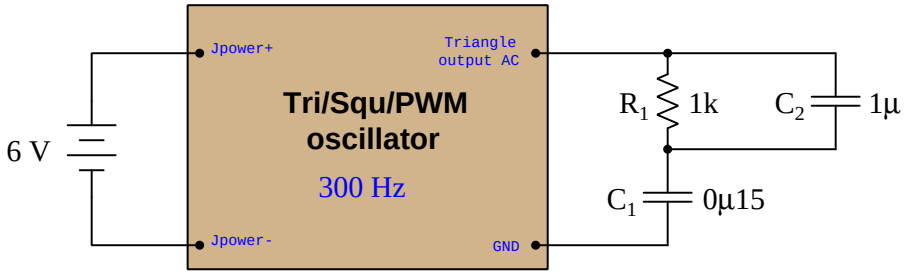
Here are some test results on simple RC networks:



Connected to the RC network, frequency was measured using a Fluke model 87-III multimeter and adjusted to 1 kHz, and then total voltage measured across the series $R_1 \leftrightarrow C_1$ combination as 233.0 mVAC.

Parameter	**Measured** (triangle-wave)	**Predicted** (sine-wave)
V_{R1}	205.7 mVAC	205.8 mVAC
V_{C1}	109.3 mVAC	109.2 mVAC

Testing a slightly more complex circuit at a frequency of 300 Hz, the loaded voltage output of the oscillator being 231.6 mVAC this time:



Parameter	**Measured** (triangle-wave)	**Predicted** (sine-wave)
V_{R1}	23.7 mVAC	27.43 mVAC
V_{C1}	207.5 mVAC	207.0 mVAC
V_{C2}	23.7 mVAC	27.43 mVAC

In both applications, the greatest error between measured voltage and predicted voltage as a percentage of total voltage was in the second circuit across $R_1||C_1$ (23.7 milliVolts rather than 27.43 milliVolts), and this is only -1.61% of the source voltage which is considerably less than the $\pm$ 5% tolerance of the resistor and capacitors!

If we compare the Fourier series for a sine wave and a triangle wave (both having unity peak values and a frequency of ω) we see that the first harmonic of the triangle wave function is identical to the sine wave, and that all the other harmonics in the triangle wave are significantly smaller-amplitude than the fundamental:

Sine wave

$$\cos \omega t$$

Triangle wave

$$\cos \omega t + \frac{1}{9} \cos 3\omega t + \frac{1}{25} \cos 5\omega t + \frac{1}{49} \cos 7\omega t + \cdots + \frac{1}{n^2} \cos n\omega t$$

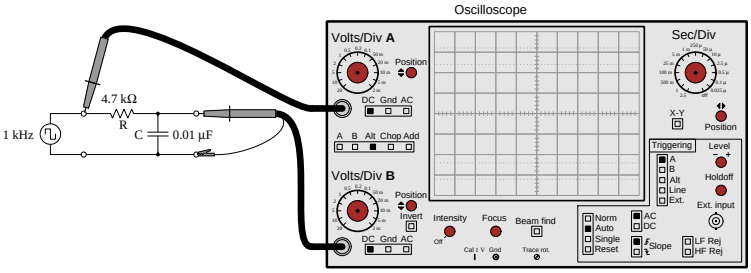
This tells us any deviations between the measured (triangle-wave) and predicted (sine-wave) voltage values are likely to be minimal, the third harmonic being only 11.1% of the fundamental's amplitude, the fifth harmonic being only 4% of the fundamental's amplitude, etc. The effects of higher-order harmonics are truly negligible due to their vastly smaller amplitudes as well as due to the fact that most digital multimeters suffer "cut off" in the audio-frequency range and therefore cannot measure signal components in the tens of thousands of Hertz.

With access to a digital oscilloscope having FFT capability[1] to show precise voltage values for each harmonic of a measured waveform, we have an even better solution for obtaining voltage measurements in agreement with predicted values when not using perfectly sinusoidal signal generators. Since the oscilloscope's FFT algorithm separates and displays each of the sinusoidal harmonics apart from one another in any non-sinusoidal waveform, if we simply pay attention to the magnitudes of a common harmonic frequency within each voltage measurement we will essentially take circuit measurements on purely sinusoidal voltages of the same frequency. For example, we could measure the fundamental (i.e. the first harmonic)[2] amplitude of source voltage, then the fundamental amplitudes of each of the other components' voltages, and check to see that these measured voltage values match well with our predictions at that frequency. This technique, in effect, lets us measure the effects of a purely sinusoidal signal even when the real signal is not sinusoidal at all, by taking measurements only on a common harmonic of the measured voltages!

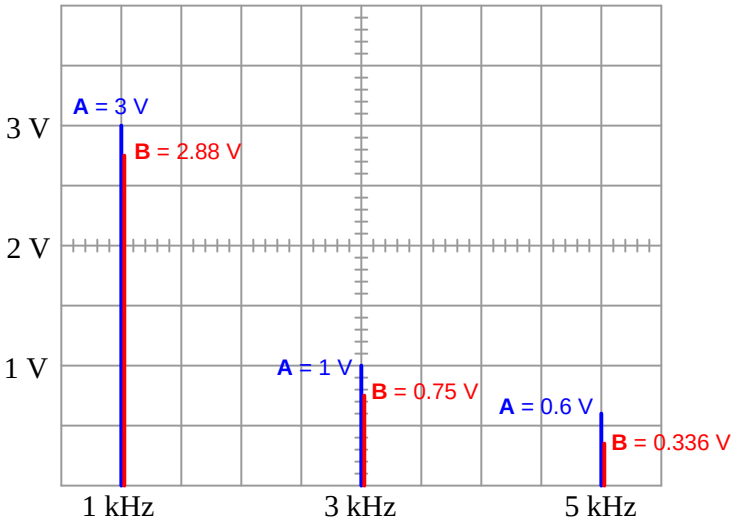
[1] At the time of this writing (2022) some inexpensive oscilloscopes may be found with rather poor FFT resolution, resulting in wide spectral peaks with uncertain height (voltage) values. You know you are working with a sufficiently precise instrument when the harmonic peaks show as thin lines rather than exaggerated bell-curves.

[2] There is no particular reason why we might choose the first harmonic over any of the others, other than the fact that with triangle and square waves this fundamental will be vastly stronger than any of the other harmonics.

This testing technique deserves some elaboration, and so we shall explore it by example. Consider the following test circuit where a signal generator configured to output a square-wave AC signal at 1000 Hz energizes a simple RC network consisting of a 4.7 kΩ resistor and a 0.01 μF capacitor:

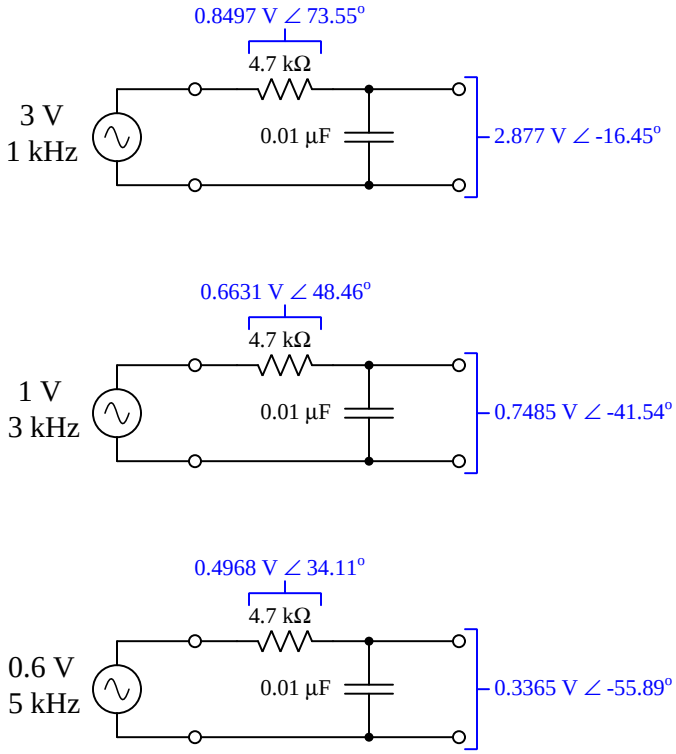


We know from Fourier analysis that a square wave is actually equivalent to a sine wave at the same fundamental frequency added to another sine wave one-third the amplitude at three times that frequency (3rd harmonic) added to another sine wave one-fifth the amplitude of the fundamental at five times that frequency (5th harmonic), and so on. If we examine the frequency-domain plots of the signal generator's output (channel A) versus the capacitor's voltage drop (channel B), we see the circuit's response to pure sine waves at each of those frequencies:



The relative peak heights of the channel A signal (3 Volts, 1 Volt, 0.6 Volts) are simply the result of the Fourier series for a square wave and has nothing to do with the RC network. The ratios between peak heights of channel A and channel B at each harmonic frequency, however, are unique to the 4.7 kΩ and 0.01 μF RC network because those voltage pairs represent the attenuation of this particular network at each of those sinusoidal frequencies.

If we mathematically analyze this same RC network for each of the square wave's harmonic amplitudes and frequencies used in the test circuit, we should obtain results verifiable by using the oscilloscope in FFT mode:

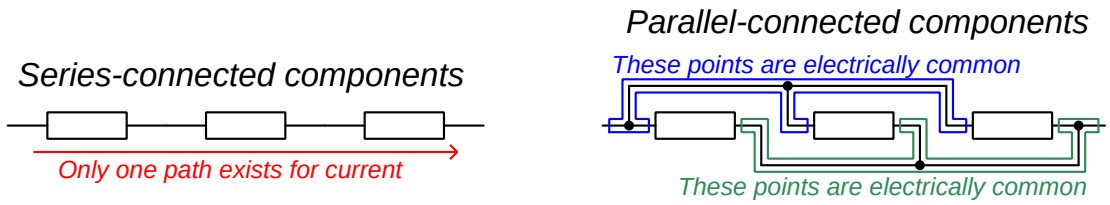


For students with access to oscilloscopes having fine-resolution FFT capability, this not only means it is unnecessary to secure a signal generator with pure sinusoidal output, but it also means the ability to energize any AC circuit with *any* waveshape and test its response at multiple sinusoidal frequencies *simultaneously*!

Chapter 3

Simplified Tutorial

A *series-parallel* circuit is one containing both *series* and *parallel* networks. A "series" network is defined by its components being connected in such a way as to form a single path for current (i.e. all components connected in a row). A "parallel" component is defined by its components sharing the same two sets of electrically common points. Both network types are shown here for comparison:



A universal property of series networks is that their components share the same amount of current. This is due to the Law of Electric Charge Conservation: electric charge carriers can neither be created nor destroyed, but must always be accounted. The only way for the amount of current entering any point to differ from the amount of current exiting that same point is if charges were to somehow either vanish or come into existence at that point. Since a series circuit is nothing more than a back-to-back collection of points, this principle extends to the series circuit in its entirety.

A universal property of parallel networks is that their components share the same amount of voltage. This is due to the Law of Energy Conservation: energy can neither be created nor destroyed, but must always be accounted. The electrical commonality between connected terminals of parallel components ensures those connected points form equipotential surfaces. By definition, charge carriers are free to move along a set of equipotential points without gaining or losing any energy, and therefore any difference in energy between a charge at *any point* along that equipotential surface and an identical charge at *any point* along a different equipotential surface must be the same.

A common application of these guarantees is in the connection of electrical meters to measure current and voltage. If we desire to measure current through some component, we must connect our ammeter in *series* with that component to ensure the meter experiences the exact same current as

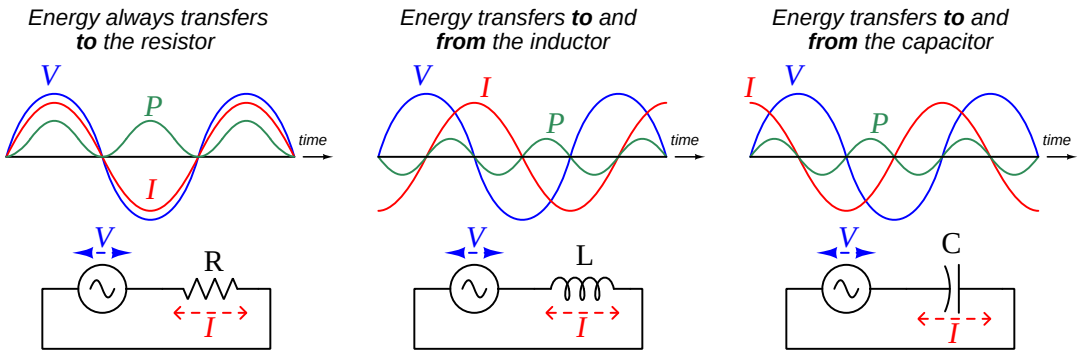
the component. Likewise, if we desire to measure voltage across some component, we must connect our voltmeter in *parallel* with that component to ensure the meter experiences the same voltage as the component.

Kirchhoff's Laws of voltage and current are important principles for analyzing series-parallel circuits as well. Kirchhoff's Voltage Law (KVL) states that the algebraic sum of all voltage rises and drops in any loop must be equal to zero. This "loop" may or may not happen to trace an actual path for current in a circuit, as Kirchhoff's Voltage Law is true regardless of path. Kirchhoff's Voltage Law is based on the Law of Energy Conservation: when any "test charge" is moved from location to location and eventually back to its starting point while experiencing a constant electric field, its net energy gain/loss must be zero. This is analogous to a hiker climbing up and down a mountain, gaining potential energy when climbing and losing potential energy when descending: when the hiker returns to their starting altitude, their net total potential energy gain/loss will be zero.

Kirchhoff's Current Law states that the algebraic sum of all currents entering and exiting a point (node) in a circuit must be equal to zero. Kirchhoff's Current Law is based on the Law of Electric Charge Conservation: all charges entering must eventually exit, and so for any continuous flow of electric charges the rate in must equal the rate out.

The astute reader will note that both of Kirchhoff's Laws depend on some form of continuity: Kirchhoff's Voltage Law assumes the moving charge inhabits a steady electric field; Kirchhoff's Current Law assumes a steady flow of current. With AC circuits we know nothing is truly continuous, and so will Kirchhoff's Laws still hold true? The answer is a qualified *yes*: so long as the propagation effects of voltage and current in a circuit (at very nearly the speed of light) are significantly faster than the period of any waveform in the circuit, we may apply these Laws to AC circuits just as confidently as we do in DC circuits. At extremely high frequencies, typically millions of Hertz and beyond, Kirchhoff's "Laws" begin to break down. We will not be considering any of these high-frequency effects in this tutorial.

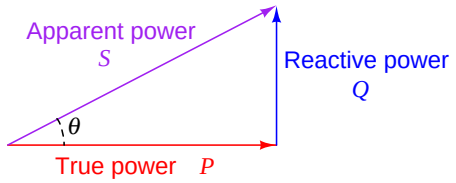
Power in AC circuits is certainly more complicated than in DC circuits, because energy does not always flow from source to load as in the case of DC circuits. Any *reactance* in an AC circuit will alternately absorb and return energy rather than dissipate it, and this results in some amount of energy exchange that does no useful work:



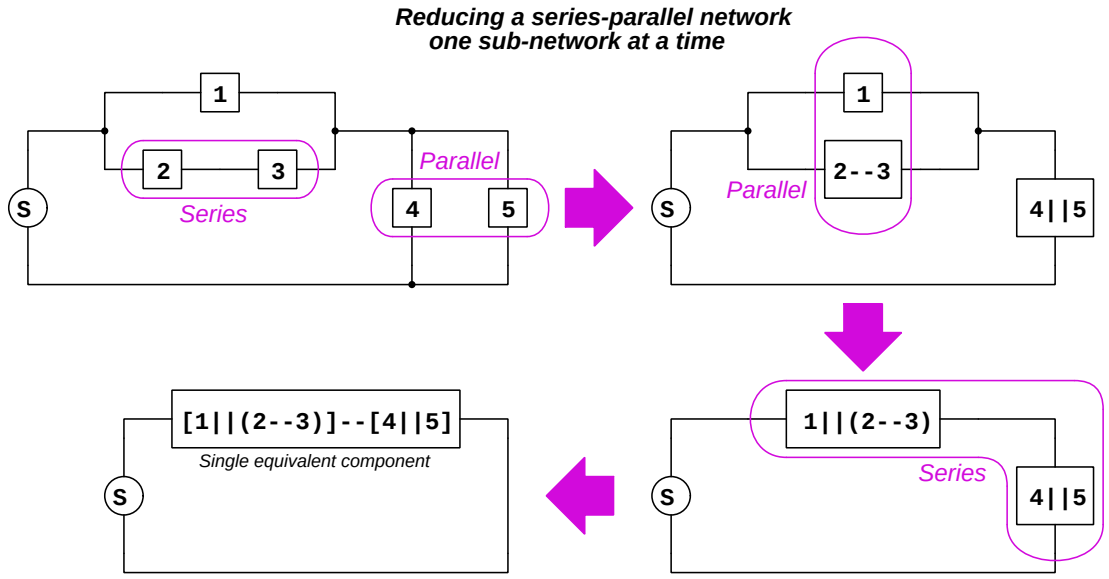
AC circuits containing reactance will always exhibit more current than what is strictly necessary to do whatever work the circuit is designed to do. Power that leaves the circuit through dissipative

elements is called *true power* (P) and is measured in Watts (W). Power that exchanges losslessly between components is called *reactive power* (Q) and is measured in Volt-Amperes Reactive (VAR). The combination of these two forms of power in an AC circuit is called *apparent power* (S) and is measured in Volt-Amperes (VA). Apparent power is what you calculate when multiplying a suitably-connected voltmeter's reading by a suitably-connected ammeter's reading in an AC circuit: it is the amount of power that "apparently" exists in the circuit. *Power factor* is the ratio of true power to apparent power $\left(\frac{P}{S}\right)$, and may alternatively be computed as the cosine of the impedance phase angle for an AC circuit.

We may represent power factor graphically in the form of a *phasor diagram*, where the phase shift between voltage and current in an AC circuit is represented as the angle of the vector and the magnitude (length) of the vector represents the circuit's apparent power (S):



Mastery of all these concepts is foundational to successful AC series-parallel circuit analysis, which consists of representing every passive component as an *impedance* in complex form, and every series and parallel sub-network in a circuit as equivalent impedances, then repeating this process until the circuit is reduced to a single equivalent impedance.



The same reduction process is applied to multiple sources, if they exist in the circuit too. At that point, Ohm's Law may be applied to that single equivalent impedance and to the single source to calculate either voltage or current. Once that is done, the equivalent circuit may be "expanded" step-by-step, transferring all voltage and current values from the simpler version as the properties of series and parallel networks dictate, re-applying various laws (Ohm's Law, Kirchhoff's Laws) as needed to calculate any unknown voltages or currents. When complete, the all voltages and currents in the circuit's original form should be known.

An excellent habit to develop when solving any problem is to *check your work*, with the various properties and laws you've learned about circuits being great tools for this purpose. For example, after "collapsing" and then "expanding" a series-parallel circuit to calculate all component voltages and currents, it is a good idea to apply Kirchhoff's Voltage and/or Current Laws to the final result as a check of your work: no matter where you apply these principles, all the final results for voltage and current should correctly check.

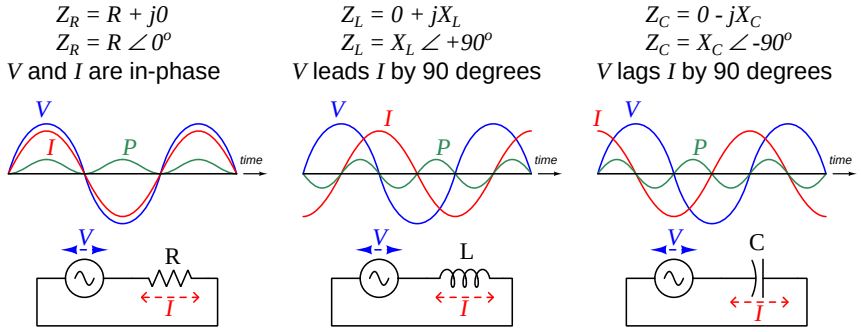
Chapter 4

Full Tutorial

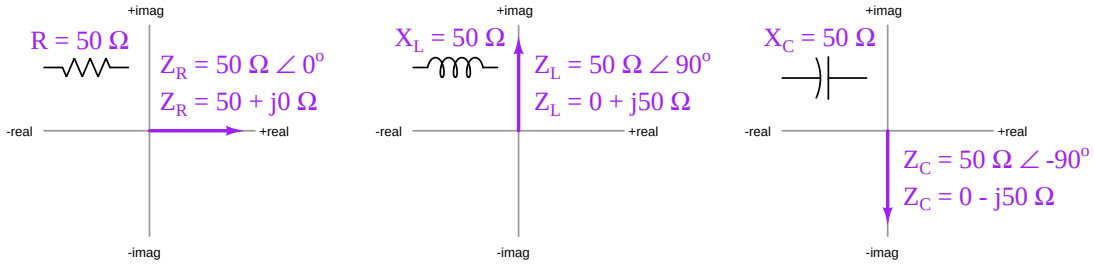
4.1 Phasor analysis review

When analyzing any AC circuit, it is not enough to simply quantify every voltage and current in terms of amplitude (e.g. how many Volts or Amperes) as we do in DC circuits. Instead, we must consider both the amplitude of each signal as well as the amount of phase shift separating them. A mathematically elegant way of accomplishing both is to use *complex numbers* which may be expressed either in rectangular form (e.g. $a + jb$) or polar form (e.g. $m\angle\theta$). When we use complex numbers to represent any AC circuit quantity, we call that value a *phasor*.

An illustrative example is how we characterize the *impedance* of passive components. Resistors function by dissipating energy in the form of heat, with voltage and current waveforms being perfectly in-phase. Inductors and capacitors, however, function by absorbing and releasing energy rather than dissipating, and as such we find voltage and current waveforms shifted by one-quarter of a cycle (i.e. 90^o) for each, voltage leading current for an inductor and voltage lagging current for a capacitor:



Graphical expressions of complex-number resistance and reactance values are called *phasor diagrams*. Three such diagrams illustrate the difference between a 50 Ω resistor versus an inductor having 50 Ω of reactance and a capacitor also having 50 Ω of reactance. Each has 50 Ohms of impedance, but each of these impedances has a different phase angle defined by the phase shift between the component's voltage and current:



The ratio of voltage to current in a DC network is resistance ($R = \frac{V}{I}$), and in an AC circuit where phase shifts exist is impedance ($Z = \frac{V}{I}$). Both are measured in unit of the Ohm (Ω). The ratio of current to voltage is the reciprocal of these quantities: for DC networks we call it *conductance* ($G = \frac{I}{V}$) and for AC it is known as *admittance* ($Y = \frac{I}{V}$). We measure both in the unit of *Siemens*[1] (S).

Some examples[2] of component impedances and admittances are shown here:

- A 570 Ω resistor at any frequency will have the following impedance and admittance values:

  $Z = 570\ \Omega \angle 0^o$ *(polar form)* $= 570 + \text{j}0\ \Omega$ *(rectangular form)*

  $Y = 0.0017544\ \text{S} \angle 0^o$ *(polar form)* $= 0.0017544 + \text{j}0\ \text{S}$ *(rectangular form)*

- A 3.5 H inductor at a frequency of 120 Hz will have the following impedance and admittance values:

  $Z = 2.639\ \text{k}\Omega \angle +90^o$ *(polar form)* $= 0 + \text{j}2.639\ \text{k}\Omega$ *(rectangular form)*

  $Y = 0.00037894\ \text{S} \angle -90^o$ *(polar form)* $= 0 - \text{j}0.00037894\ \text{S}$ *(rectangular form)*

- A 0.01 μF capacitor at a frequency of 3 kHz will have the following impedance and admittance values:

  $Z = 5.305\ \text{k}\Omega \angle -90^o$ *(polar form)* $= 0 - \text{j}5.305\ \text{k}\Omega$ *(rectangular form)*

  $Y = 0.00018850\ \text{S} \angle 90^o$ *(polar form)* $= 0 + \text{j}0.00018850\ \text{S}$ *(rectangular form)*

The utility of phasor representation in AC circuits is that with all signal and component values expressed in phasor form we find most of the foundational principles learned for DC circuit analysis still apply in AC circuits. Quantities that add in series DC networks (e.g. voltage V, resistance R) add as phasor quantities in AC networks (e.g. voltage V, impedance Z); additive quantities in parallel DC networks (e.g. current I, conductance G) add as phasor quantities in AC networks (e.g. current I, admittance Y). With phasor quantities, Ohm's Law, Kirchhoff's Voltage Law, and Kirchhoff's Current Law still hold true in AC networks just as they do for DC.

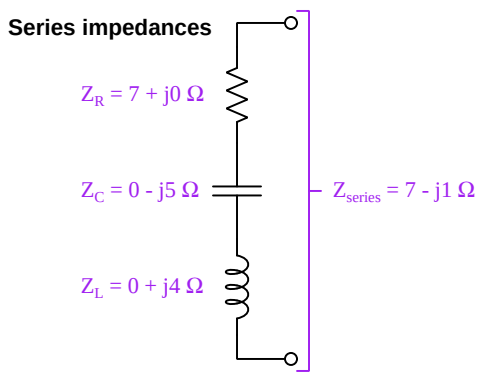
[1] Prior to the adoption of German engineer Werner von Siemens' surname as the unit of measurement for conductance and admittance, the unit of the *Mho* served quite well. This, of course, was a sort of pun on the spelling of *Ohm*, since "mho" is "ohm" spelled backwards, intended to represent the fact that the reciprocal of any Ohm value yields a value in Mhos.

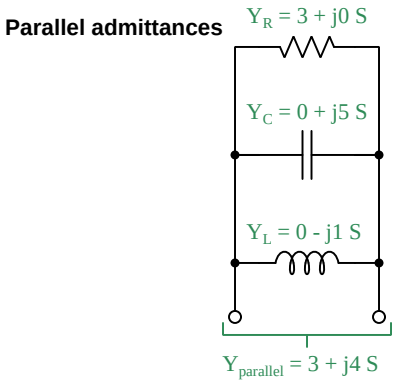
[2] Try calculating these impedance and admittance values from the given component values, to check your understanding. This is a good learning strategy to apply when reading any mathematical text: work through the presented examples on your own to see if you achieve the same results! Please note that when you apply either the $X_L = 2\pi fL$ formula or the $X_C = \frac{1}{2\pi fC}$ formula using your calculator to compute reactance, the result will *only* be a reactance value and not a (complex) impedance value. In order to attach the desired phase angle to your computed reactance value, you will have to perform the additional step of multiplying that reactance by a *unit phasor* which is nothing more than the quantity of 1 with the correct phase angle. For example, a capacitive reactance of 5.305 kΩ would be multiplied by $1 \angle -90^o$ to yield a capacitive *impedance* of 5.305 k$\Omega \angle -90^o$.

It is equally valid to express any phasor quantity in either polar or rectangular form. However, unless we have access to an electronic calculator capable of performing complex-number arithmetic, we find certain arithmetic operations much easier to perform with one notation more than the other. Specifically, addition and subtraction are simplest when phasors are in rectangular form, while multiplication and division are simplest when phasors are in polar form.

Addition of rectangular-form complex numbers consists simply of adding their real components together to find the real component of the sum, and doing the same with the imaginary components. Expressing this algebraically, $(a + jb) + (x + jy) = (a + x) + j(b + y)$. Subtraction follows much the same pattern: $(a + jb) - (x + jy) = (a - x) + j(b - y)$.

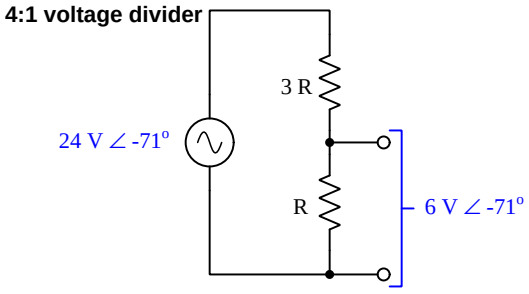
Here are some practical examples of rectangular-form phasor arithmetic where the calculations are simple enough to perform without a calculator:

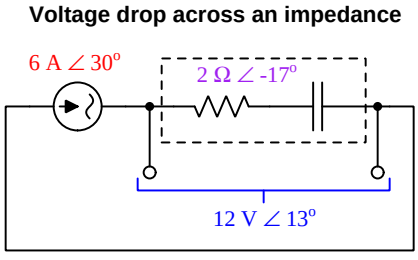




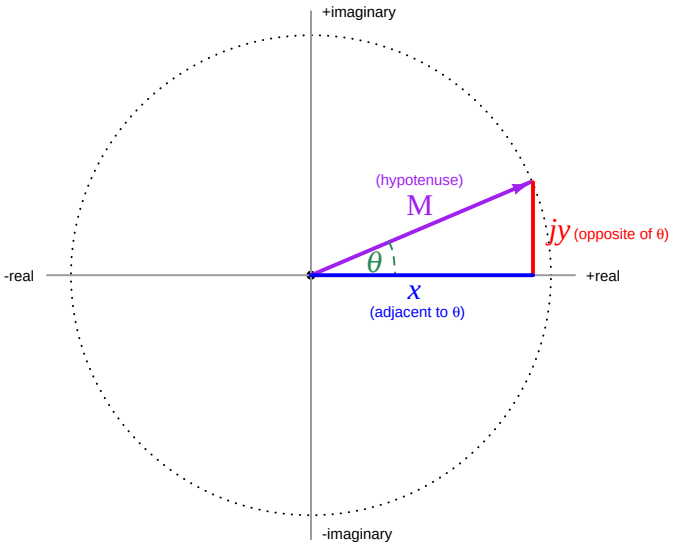
Multiplication of polar-form complex numbers consists simply of multiplying their magnitudes together to find the magnitude of the product, and adding the angles to find the angle of the product. Expressing this algebraically, $(a\angle b) \times (x\angle y) = (a \times x)\angle(b + y)$. Division follows a similar pattern: $(a\angle b) \div (x\angle y) = (a \div x)\angle(b - y)$.

Here are some practical examples of polar-form phasor arithmetic where the calculations are simple enough to perform without a calculator:





When analyzing AC circuits without the use of a complex-number calculator, we invariably must convert between rectangular and polar forms in order to prepare the phasor values for addition/subtraction or multiplication/division, respectively. Both conversions are most easily understood in terms of a right triangle, seeing the rectangular form's real and imaginary components as the adjacent and opposite sides, and the polar form's magnitude and angle as the hypotenuse:



Converting rectangular $(x + jy)$ into polar $(A\angle\theta)$:

$$A = \sqrt{x^2 + y^2} \qquad \theta = \arctan \frac{y}{x}$$

Converting polar $(A\angle\theta)$ into rectangular $(x + jy)$:

$$x = A \cos \theta \qquad y = A \sin \theta$$

Some cautionary notes are in order here. First, I highly recommend storing all computed values in your calculator's memory rather than re-entering them manually, because you will find even slight rounding errors tend to become exaggerated with trigonometric functions. Second, when computing the phase angle (θ) from real and imaginary quantities (x and jy) be careful to verify the angle against your qualitative expectations. For example, $5 + j5 = 7.071\angle 45^o$ and $-5 - j5 = 7.071\angle 225^o$, but you'll find arctan $\frac{-5}{-5}$ yields the same result (45^o) as arctan $\frac{5}{5}$ because $\frac{-5}{-5} = \frac{5}{5}$. To put it simply, the arc-tangent function does not "know" whether the phasor exists in the first or in the third quadrant of the complex plane.

Here are some rectangular and polar equivalents, useful for practice as you master these concepts:

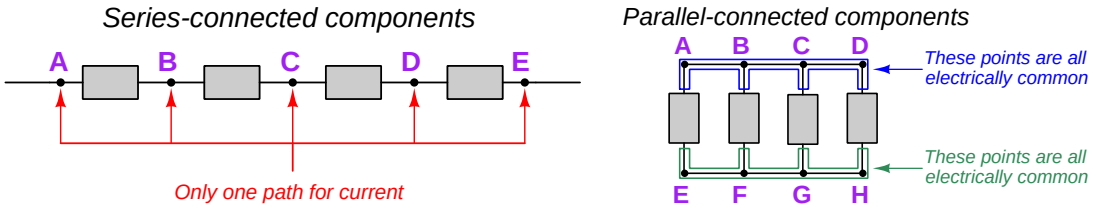
$$20 - j11 = 22.83\angle -28.81^o \qquad 11.49 + j9.642 = 15\angle 40^o$$

$$-10 + j2 = 10.20\angle 168.7^o \qquad -11.82 - j2.084 = 12\angle -170^o$$

4.2 Series and parallel network properties

Now that we have reviewed the use of phasors in AC electric circuits, let us move on to the topic at hand: *series-parallel AC circuits*. A "series-parallel" network, by definition, is one containing both series and parallel subsections.

To review, a *series* network is defined as one possessing only one path for current, while a *parallel* network is one where all components share the same two sets of electrically common points. Examples are shown below to clarify these definitions:



In any series network the amount of current at any one point in the circuit must be the same as the amount of current at any other point, because there is only one pathway for current and because electric charges are *conserved*.

In any parallel network the amount of voltage across any component must be the same as the amount of voltage across any other component, because electric charge carriers neither gain nor lose appreciable energy passing along those electrically common points, and so the amount of potential energy between any point along one common set and any point along the other set must be the same. Another way to think about the equality of voltages in a parallel network is to consider what must occur if two parallel-connected components were to somehow have different amounts of voltage: it would mean a charge carrier could lose energy passing through one component, then gain a different amount of energy passing through another component to arrive at its starting point at some different amount of energy than what it had when it began its travel, thus violating[3] the *Law of Energy Conservation*.

[3] This is an example of a logical technique called *reductio ad absurdum*, whereby a proposition is disproven by demonstrating it necessarily leads to nonsense. Parallel-connected components must share the same voltage because otherwise they would violate the Law of Energy Conservation. So long as that Law is iron-clad, this is sufficient proof that parallel-connected components must experience the same voltage.

Once defined, we may express properties unique to each network type:

Definition: *Series*-connected electrical components provide exactly one path for current.

Property #1 Series-connected components experience the *same current* at any given time, due to the Conservation of Electric Charge (i.e. electric charges cannot disappear nor come into being, and so every charge entering one portion of a series network must eventually exit that portion). $I_1 = I_2 = I_3 \cdots = I_n$

Property #2 Voltages add in series: total voltage across a string of series-connected components is equal to the *phasor sum* of the components' voltages, due to the Conservation of Energy (i.e. the sum of all energy gains and losses must equal the total gain/loss). $V_{total} = V_1 + V_2 + V_3 \cdots + V_n$, which is graphically equivalent to stacking voltage phasors tip-to-tail in a phasor diagram.

Property #3 Impedances add in series: total impedance for a string of series-connected impedances is equal to the *phasor sum* of those impedance values. $Z_{total} = Z_1 + Z_2 + Z_3 \cdots + Z_n$, which is graphically equivalent to stacking impedance phasors tip-to-tail in a phasor diagram.

Definition: *Parallel*-connected electrical components share two sets of electrically common points.

Property #1 Parallel-connected components experience the *same voltage* at any given time, due to the Conservation of Energy (i.e. if electric charge carriers along each set of electrically common points must exist at the same energy levels by virtue of those points' commonality, then the difference of potential between those two sets of points must be the same for the entire network). $V_1 = V_2 = V_3 \cdots = V_n$

Property #2 Currents add in parallel: total current for a set of parallel-connected components is equal to the *phasor sum* of the components' currents, due to the Conservation of Electric Charge (i.e. electric charges cannot be created nor destroyed, and therefore each stream of charges flowing through the components must add to form the total stream). $I_{total} = I_1 + I_2 + I_3 \cdots + I_n$, which is graphically equivalent to stacking current phasors tip-to-tail in a phasor diagram.

Property #3 Impedances diminish in parallel: total impedance for a set of parallel-connected impedances is equal to the *reciprocal* of the phasor sum of the reciprocated impedance values. $Z_{total} = 1/(1/Z_1 + 1/Z_2 + 1/Z_3 \cdots + 1/Z_n)$

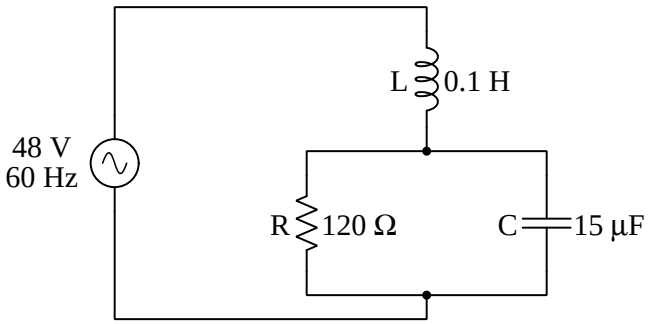
These properties hold true for any series network because they are rooted in fundamental conservation laws. They apply for AC as well as DC circuits, and they apply for any types of components thusly connected. The major difference between these properties as applied to AC versus DC is that all calculations must be done using phasors in AC rather than simple numbers as in DC.

In series-parallel circuit analysis we generally approach the task of computing component voltages and currents by reducing each series and parallel sub-section of the circuit, step-by-step, into some equivalent impedance value. When this process is complete we will end up with a single equivalent (total) impedance for the circuit. After calculating voltage or current values for this total impedance, we then "expand" the circuit back to its original form, transferring the previously-calculated voltage or current values as appropriate (e.g. transferring current to any series expansion, voltage to any parallel expansion) to the expanded circuit. This process is repeated until the circuit is in its original form again, with all voltages and currents solved.

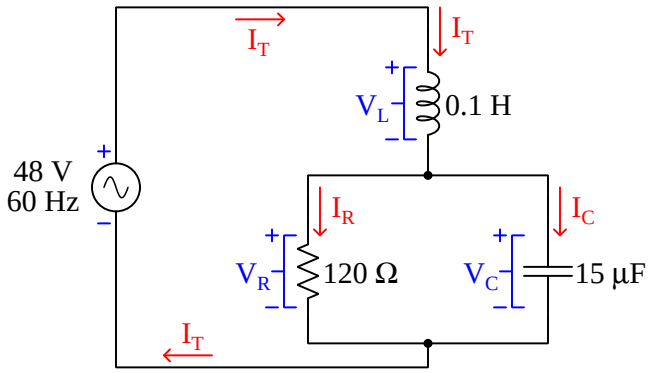
In other words, we analyze series-parallel AC circuits in the same general way as series-parallel DC circuits, the major difference being the use of complex numbers (phasors) to represent all voltage and current values, and expressing component values as complex *impedances* rather than resistances.

4.3 AC series-parallel analysis example

Let's analyze the following series-parallel circuit containing a resistor, inductor, and a capacitor all powered by an AC voltage source:



A recommended first step is to annotate the circuit diagram with voltage polarities and current arrows, as though it were DC (treating the resistor, inductor, and capacitor as loads). This annotation will give context[4] to the phase angles of all calculated voltages and currents. Since no phase reference is specified in the circuit, we will arbitrarily set the source voltage to be the zero-degree reference:



Next, we translate every given passive component value into a complex impedance, using the reactance formulae $X_L = 2\pi f L$ and $X_C = \frac{1}{2\pi f C}$ to compute phasor magnitudes for Z_L and Z_C, and remembering[5] that pure inductance has an impedance angle of $+90^o$ while pure capacitance has an impedance phase angle of -90^o:

$$Z_R = 120\,\Omega\angle 0^o \qquad Z_L = 37.699\,\Omega\angle 90^o \qquad Z_C = 176.84\,\Omega\angle -90^o$$

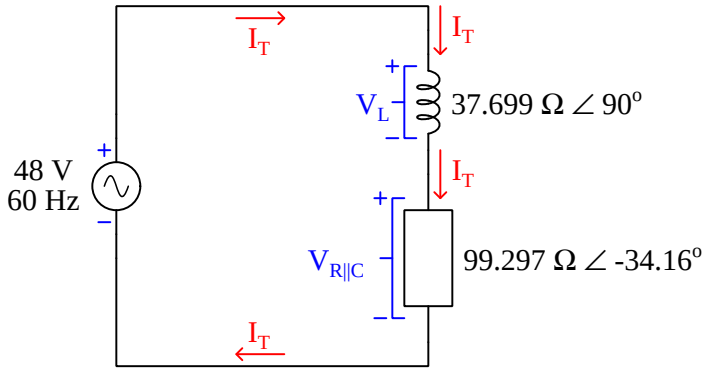
[4] These annotations define voltage polarities and current directions at each of those quantities' zero-degree points.

[5] An easy way to convert a computed reactance value into an *impedance* value complete with a phase angle is to use our calculator to multiply the reactance value by a unit phasor with the desired angle. For example, if $X_L = 37.699\ \Omega$, then we can simply multiply 37.699 by $1\ \angle\ 90^o$ to get a Z_L value of $37.699\ \Omega\ \angle\ 90^o$.

An important tip for success in analyzing AC networks is to store all computed values in your calculator's memory for future recall. This will not only eliminate the possibility of certain key-stroke errors risked by re-entering previously-calculated values, but it also eliminates rounding errors which tend to compound detrimentally in these problems. While this practice is always useful in circuit analysis – AC or DC – they tend to make an enormous difference for students new to the art. For example, the calculated values of Z_L and Z_C we just computed should be stored in memory rather than re-entered when needed later in the analysis.

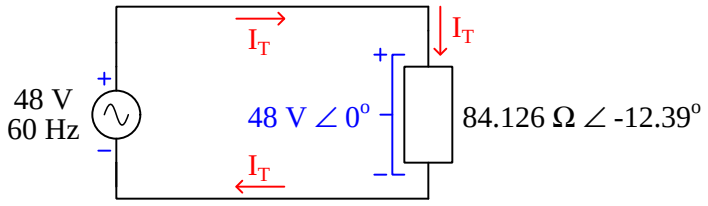
Now we are ready to begin condensing this series-parallel circuit into a single equivalent impedance. First, we recognize that R and C are in parallel with each other, and so we may calculate an equivalent parallel impedance for $Z_R \parallel Z_C$ using the parallel impedance formula:

$$Z_{R\parallel C} = \frac{1}{\frac{1}{Z_R} + \frac{1}{Z_C}} = \frac{1}{\frac{1}{120\,\Omega\angle 0^o} + \frac{1}{176.84\,\Omega\angle -90^o}} = 99.297\,\Omega\angle -34.16^o$$



Our next step will be to reduce these two series-connected impedances into a single equivalent impedance for the entire circuit, knowing that series impedances add:

$$Z_T = Z_L + Z_{R\parallel C} = (37.699\,\Omega\angle 90^o) + (99.297\,\Omega\angle -34.16) = 84.126\,\Omega\angle -12.39^o$$



With total voltage and total impedance in hand, total current is just one Ohm's Law calculation away. Again, *all* of these computed values should be stored in your calculator's memory for future recall:

$$I_T = \frac{V_T}{Z_T} = \frac{48\text{ V}\angle 0^o}{84.126\,\Omega\angle -12.39^o} = 0.5706\text{ A}\angle 12.39^o$$

Calculating total circuit powers[6]:

$$S = IV = (48 \text{ V})(0.5706 \text{ A}) = 27.39 \text{ VA}$$

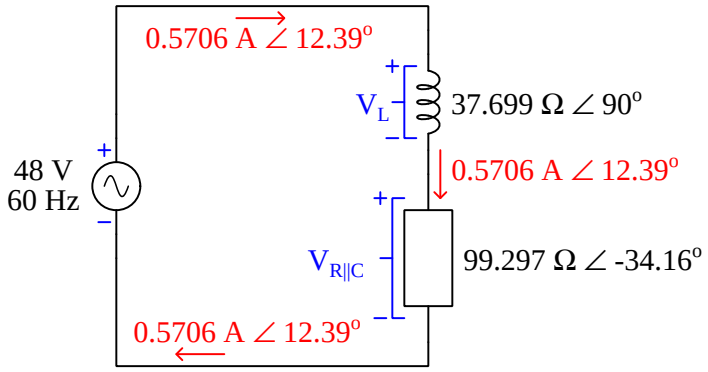
$$P = IV \cos\theta = (48 \text{ V})(0.5706 \text{ A})(\cos -12.39^o) = 26.75 \text{ W}$$

$$Q = IV \sin\theta = (48 \text{ V})(0.5706 \text{ A})(\sin -12.39^o) = -5.879 \text{ VAR} = 5.879 \text{ VAR leading}$$

The total impedance phase angle for this circuit is a negative value, which tells us its reactance is predominantly capacitive rather than inductive. This is why we obtained a negative value for reactive power rather than a positive value. Total current leads total voltage in this circuit, and so the power factor is qualified as *leading*:

$$\text{P.F.} = \cos\theta = \cos -12.39^o = 0.9767 \text{ leading}$$

Returning to our analysis of voltages and currents, we are ready to "expand" this single-impedance equivalent circuit one step closer to its original form. Since we are expanding Z_T into Z_L and $Z_{R\|C}$ connected in series, and we know it is *current* that is common throughout any series network, we may transfer the calculated value for total current to each of these two impedances:



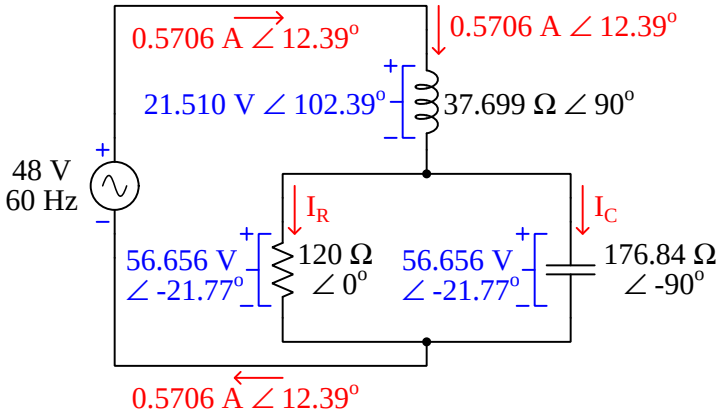
Calculating voltage across the inductor and across the resistor/capacitor subnetwork using Ohm's Law:

$$V_L = IZ_L = (0.5706 \text{ A}\angle 12.39^o)(37.699\,\Omega\angle 90^o) = 21.510 \text{ V}\angle 102.39^o$$

$$V_{R\|C} = IZ_{R\|C} = (0.5706 \text{ A}\angle 12.39^o)(99.297\,\Omega\angle -34.16^o) = 56.656 \text{ V}\angle -21.77^o$$

[6]Note how we use only the polar magnitude of current in these scalar calculations. Scientific calculators capable of complex-number operations offer a way to extract just the magnitude, or just the phase angle, of any complex number on demand. You should use that feature here when recalling the stored current value from memory.

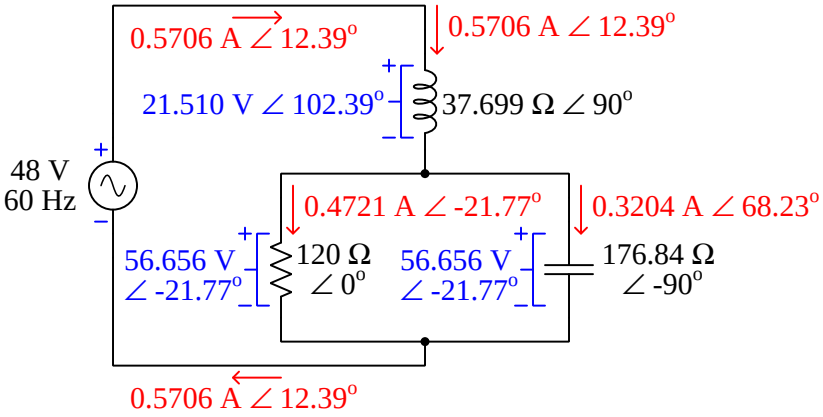
Next we may expand the parallel equivalent of R and C into individual components, and since we know it is *voltage* that is common throughout any parallel network, we may transfer this value of 56.656 Volts $\angle$ -21.77^o to both the resistor and the capacitor:



The only calculations left to complete are current through the resistor and the capacitor, again using Ohm's Law:

$$I_R = \frac{V_R}{Z_R} = \frac{56.656 \text{ V} \angle -21.77^o}{120\,\Omega \angle 0^o} = 0.4721 \text{ A} \angle -21.77^o$$

$$I_C = \frac{V_C}{Z_C} = \frac{56.656 \text{ V} \angle -21.77^o}{176.84\,\Omega \angle -90^o} = 0.3204 \text{ A} \angle 68.23^o$$



It is always a good idea to double-check our work by applying principles other than those used to arrive at the computed values to see if those computed values still agree[7]. For example, we may check our current values by applying Kirchhoff's Current Law at the node between the two circuit sub-sections, verifying that the sum of the resistor and capacitor currents indeed equals the inductor current:

$$(0.4721 \text{ A}\angle -21.77^o) + (0.3204 \text{ A}\angle 68.23^o) = 0.5706 \text{ A}\angle 12.39^o$$

Likewise, we may apply Kirchhoff's Voltage Law to the verification of voltage calculations, verifying that the sum of the resistor/capacitor and inductor voltages indeed equals source voltage:

$$(21.510 \text{ V}\angle 102.39^o) + (56.656 \text{ V}\angle -21.77^o) = 48.00 \text{ V}\angle 0.00^o$$

[7] A poor way to double-check work is to simply re-do all the previous steps to see if you get the same results. The problem with this approach is that you are likely to repeat the exact same error(s), in which case achieving identical results will give our false confidence that your answers are correct when they very well may be incorrect. By applying different principles to the verification of answers, you greatly increase your probability of detecting errors. In order for the superior method to fail, you would somehow have to make *precisely complementary mistakes* so that the application of those other principles results in validation of your previous (wrong) results. This is extremely unlikely because the double-check uses different mathematical operations than the original solution, in this particular case the double-check using addition while the original solution used multiplication and division.

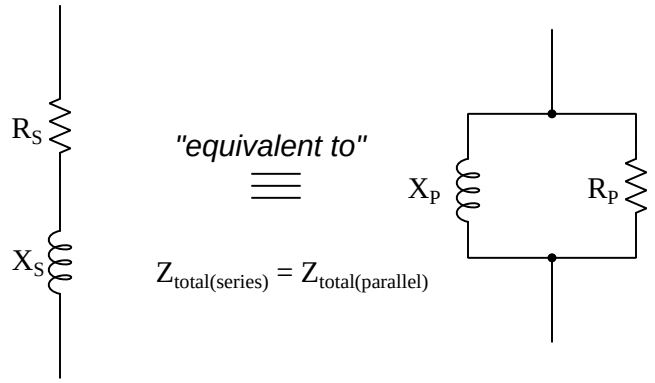
Chapter 5

Derivations and Technical References

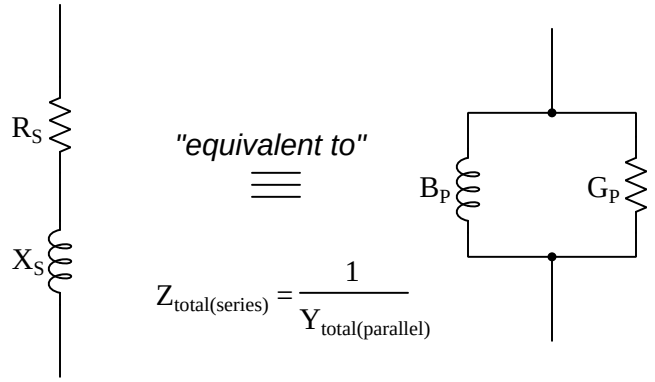
This chapter is where you will find mathematical derivations too detailed to include in the tutorial, and/or tables and other technical reference material.

5.1 Equivalent series and parallel XR networks

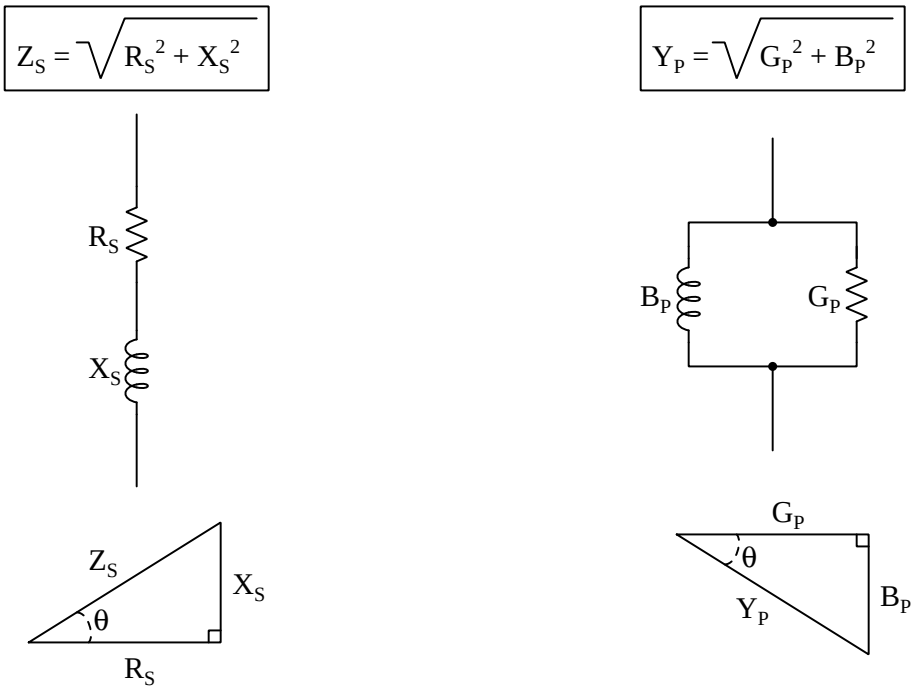
It is often useful in AC circuit analysis to be able to convert a series combination of resistance and reactance into an equivalent parallel combination of resistance and reactance, or vise-versa:



We know that while resistances and reactances add (as phasor quantities) in series, the same cannot be said in parallel. However, the reciprocal of resistance (called *conductance*, G) and the reciprocal of reactance (called *susceptance*, B) do add as phasor quantities in parallel to yield a sum known as *admittance* (Y) which is the reciprocal of impedance (Z):



These sums may be computed using scalar numbers, with series impedance (Z) being the Pythagorean sum of R and X being Z, and parallel admittance (Y) being the Pythagorean sum of G and B:



If these two circuits are truly equivalent to one another, having the same total impedance, then their representative triangles should be geometrically similar (identical angles, same proportions of side lengths). With equal proportions, $\frac{R_S}{Z_S}$ in the series circuit triangle should be the same ratio as $\frac{G_P}{Y_P}$ in the parallel circuit triangle, that is $\frac{R_S}{Z_S} = \frac{G_P}{Y_P}$.

We may derive an equation relating resistance in the series network to resistance in the parallel network (and to the total resistance of each network) by taking this proportionality and substituting $\frac{1}{R_P}$ for G_P and $\frac{1}{Z_P}$ for Y_P:

$$\frac{R_S}{Z_S} = \frac{G_P}{Y_P}$$

$$\frac{R_S}{Z_S} = \frac{\frac{1}{R_P}}{\frac{1}{Z_P}}$$

$$\frac{R_S}{Z_S} = \frac{Z_P}{R_P}$$

$$R_S R_P = Z_S Z_P$$

If these are truly equivalent circuits, then they must have the same total impedance. In other words, $Z_S = Z_P$. Simply calling this equal impedance Z and substituting for both Z_S and Z_P:

$$R_S R_P = ZZ$$

$$R_S R_P = Z^2$$

The same proportionality applies just as well to reactance in the two equivalent networks, with similar results:

$$\frac{X_S}{Z_S} = \frac{B_P}{Y_P}$$

$$\frac{X_S}{Z_S} = \frac{\frac{1}{X_P}}{\frac{1}{Z_P}}$$

$$\frac{X_S}{Z_S} = \frac{Z_P}{X_P}$$

$$X_S X_P = Z_S Z_P$$

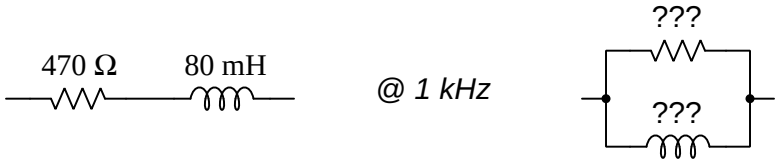
$$X_S X_P = ZZ$$

$$X_S X_P = Z^2$$

With that, we have two equations useful for finding the resistance and reactance values necessary to create an equivalent parallel network for a given series network, or vice-versa:

$$R_S R_P = Z^2 \qquad X_S X_P = Z^2$$

Let's apply this to a practical example to see how it works. Take for instance a series network comprised of a 470 Ω resistor and an 80 mH inductor operating at a frequency of 1 kHz, with the goal of determining component values necessary to build an equivalent parallel LR network.



The 80 mH inductor will have a reactance of 502.65 Ω at 1 kHz. Together, the 470 Ω resistor and the 80 mH inductor make a series impedance value of:

$$Z = \sqrt{R^2 + X^2}$$

$$Z = \sqrt{(470\,\Omega)^2 + (502.65\,\Omega)^2} = 688.16\,\Omega$$

Now that we know the total impedance of the series network, we may calculate both the parallel resistance and parallel reactance values for the equivalent parallel network using the two equations previously derived:

$$R_S R_P = Z^2$$

$$R_P = \frac{Z^2}{R_S}$$

$$R_P = \frac{(688.16\,\Omega)^2}{470\,\Omega} = 1007.58\,\Omega$$

$$X_S X_P = Z^2$$

$$X_P = \frac{Z^2}{X_S}$$

$$X_P = \frac{(688.16\,\Omega)^2}{502.65\,\Omega} = 942.12\,\Omega$$

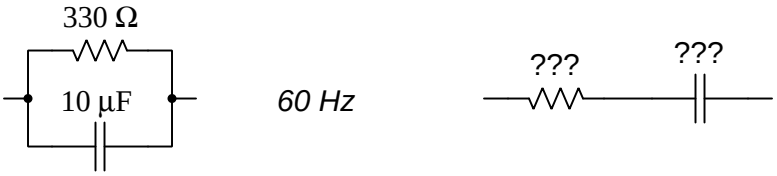
Solving for inductance at 1 kHz:

$$X_L = 2\pi f L$$

$$L = \frac{X_L}{2\pi f} = \frac{942.12\,\Omega}{(2\pi)(1000\text{ Hz})} = 149.94\text{ mH}$$

Therefore the equivalent parallel network consists of a 1007.58 Ω resistor in parallel with a 149.94 mH inductor (assuming a frequency of 1 kHz).

Let's apply this to another practical example. Take for instance a parallel network comprised of a 330 Ω resistor and a 10 μF capacitor operating at a frequency of 60 Hz, with the goal of determining component values necessary to build an equivalent series RC network.



The 10 μF capacitor will have a reactance of 265.26 Ω at 60 Hz. Avoiding the use of complex numbers, we may convert 330 Ω into a conductance (G) and the 10 μF capacitor's reactance into a susceptance (B) to compute the parallel network's total impedance:

$$X_C = \frac{1}{2\pi f C} = \frac{1}{(2\pi)(60)(10\,\mu\text{F})} = 265.26\,\Omega$$

$$B = \frac{1}{X_C} = \frac{1}{265.26\,\Omega} = 3.770\text{ mS}$$

$$G = \frac{1}{R} = \frac{1}{330\,\Omega} = 3.0303\text{ mS}$$

$$Y = \sqrt{G^2 + B^2} = \sqrt{(3.0303\text{ mS})^2 + (3.770\text{ mS})^2} = 4.837\text{ mS}$$

$$Z = \frac{1}{Y} = 206.75\,\Omega$$

Now that we know the parallel network's impedance magnitude, we may calculate resistance and reactance for the equivalent series network using our two equations:

$$R_S R_P = Z^2$$

$$R_P = \frac{Z^2}{R_S}$$

$$R_P = \frac{(206.75\,\Omega)^2}{330\,\Omega} = 129.53\,\Omega$$

$$X_S X_P = Z^2$$

$$X_P = \frac{Z^2}{X_S}$$

$$X_P = \frac{(206.75\,\Omega)^2}{265.26\,\Omega} = 161.14\,\Omega$$

Calculating the necessary capacitance to yield 161.14 Ω of reactance at 60 Hz:

$$X_C = \frac{1}{2\pi f C}$$

$$C = \frac{1}{2\pi f X_C}$$

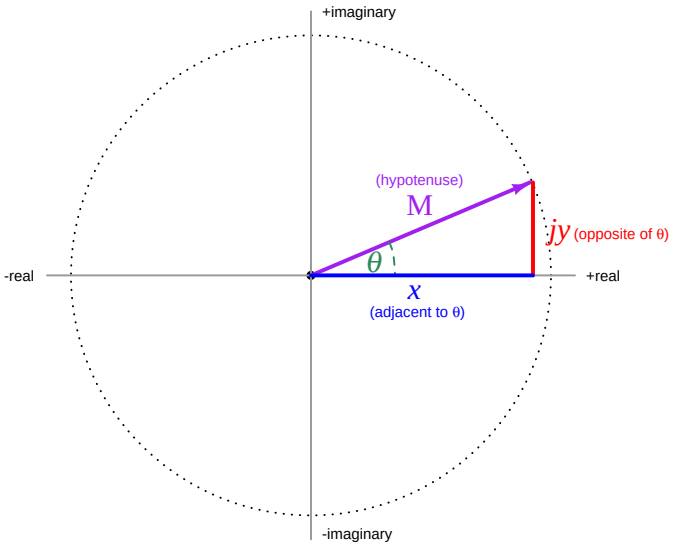
$$C = \frac{1}{(2\pi)(60\text{ Hz})(161.14\,\Omega)} = 16.46\,\mu\text{F}$$

Therefore the equivalent series network consists of a 129.53 Ω resistor in series with a 16.46 μF capacitor (assuming a frequency of 60 Hz).

5.2 Complex-number arithmetic

Complex numbers are very useful in AC circuit analysis because each one has the ability to represent both a magnitude and a phase shift between that quantity and some other reference quantity. Despite the existence of electronic calculators and computer software capable of performing arithmetic on complex-number quantities, there are still times when we must perform some calculation on these quantities "by hand". This technical reference reviews the basic arithmetic operations on complex numbers, complete with examples.

Recall that complex numbers may be represented in either *rectangular* or *polar* form, rectangular being a quantity with both a "real" and an "imaginary" component, and polar being a quantity with a magnitude and an angle. Graphically, these two forms relate to the sides of a right triangle:



Rectangular form: $x + jy$ (where $j = \sqrt{-1}$)

Polar form: $M\angle\theta$

To convert from rectangular form to polar form, $M = \sqrt{x^2 + y^2}$ and $\theta = \arctan \frac{y}{x}$

To convert from polar form to rectangular form, $x = M \cos\theta$ and $y = M \sin\theta$

As we will see, addition and subtraction is easiest to do with rectangular-form notation while multiplication and division is easiest to do with polar-form notation. Thus, circuit analysis doing "long-hand" complex-number arithmetic often involves conversions back and forth between rectangular and polar forms in order to set up the quantities before applying Ohm's Law, Kirchhoff's Laws, etc. This can be tedious, and it is also prone to rounding errors. The reader is advised to store all intermediate results in their calculator's memory and recall when needed, rather than re-type quantities and thereby incur rounding errors due to truncation.

5.2.1 Negating complex numbers

The sign of a complex number may be reversed just as easily in rectangular form as in polar form. Rectangular-form negation consists of multiplying -1 through to both the real and imaginary terms. Polar-form negation consists solely of adding 180 degrees to the angle, or alternatively, by reversing the sign of the magnitude and leaving the angle alone.

Example: reverse the sign of $5 - j4$

$$-(5 - j4)$$

$$-5 + j4$$

Example: reverse the sign of $6\angle 30^o$

$$-(6\angle 30^o)$$

$$6\angle 210^o = 6\angle -150^o = -6\angle 30^o$$

5.2.2 Adding complex numbers

Complex numbers are most easily added in *rectangular form*: simply add the real portions and then add the imaginary portions.

Example: add $5 - j4$ to $-1 - j3$

$$(5 - j4) + (-1 - j3)$$

$$(5 + (-1)) + (-j4 + (-j3))$$

$$4 - j7$$

5.2.3 Subtracting complex numbers

Complex numbers are most easily subtracted in *rectangular form*: simply subtract the real portions and then subtract the imaginary portions.

Example: subtract $5 - j4$ from $-1 - j3$

$$(-1 - j3) - (5 - j4)$$

$$(-1 - (5)) + (-j3 - (-j4))$$

$$-6 + j1$$

5.2.4 Multiplying complex numbers

Complex numbers are most easily multiplied in *polar form*: simply multiply the magnitudes and add the angles.

Example: multiply $6\angle 30^o$ by $2\angle -10^o$

$$(6\angle 30^o) \times (2\angle -10^o)$$

$$(6 \times 2)\angle(30^o + (-10^o))$$

$$12\angle 20^o$$

Multiplication of rectangular-form complex numbers less straight-forward then with polar-form numbers, and resembles multiplication of algebraic polynomials:

Example: multiply $5 - j4$ by $-1 - j3$

$$(5 - j4) \times (-1 - j3)$$

$$(5 \times (-1)) + (5 \times (-j3)) + (-j4 \times (-1)) + (-j4 \times (-j3))$$

$$(-5) + (-j15) + (j4) + (j^2 12)$$

$$(-5) + (-j15) + (j4) + ((-1)12)$$

$$(-5) + (-j15) + (j4) + (-12)$$

$$-17 - j11$$

5.2.5 Dividing complex numbers

Complex numbers are most easily divided in *polar form*: simply divide the magnitudes and subtract the angles.

Example: divide $6\angle 30^o$ by $2\angle -10^o$

$$\frac{6\angle 30^o}{2\angle -10^o}$$

$$\frac{6}{2}\angle(30^o - (-10^o))$$

$$3\angle 40^o$$

5.2.6 Reciprocating complex numbers

Reciprocation is division into one, and so complex numbers are reciprocated most easily in *polar form* just as division is best performed in polar form: simply reciprocate the magnitude and negate the angle.

Example: reciprocate $2\angle -10^o$

$$\frac{1}{2\angle -10^o}$$

$$\frac{1}{2}\angle -(-10^o)$$

$$0.5\angle 10^o$$

5.2.7 Calculator tips

Here is some advice when using calculators to do complex-number arithmetic:

- When manually entering a complex-number value, enclose that value in parentheses. Some calculators struggle to properly perform order-of-operations with complex numbers. For example, some calculators will interpret $45\angle 30^o \times 5$ as $45\angle(30^o \times 5)$ to give $45\angle 150^o$ when what was really intended was $(45\angle 30^o) \times 5 = 225\angle 30^o$. Also, note that the practice of highlighting previous results in a multi-line display and then "pasting" those results into a new calculation may suffer similar problems.
- *Never* re-enter a non-round computed result, but instead save that to a memory location and then recall from memory when needed for further calculations. You will find that rounding errors compound *aggressively* in complex-number arithmetic, and so the general good habit of using memory locations becomes a near-necessity with these calculations. Another important benefit to using memory locations is the avoidance of the order-of-operations problem mentioned previously: when recalling a complex-number value from memory and then placing that variable name (e.g. x) into subsequent calculations, the calculator treats the memory variable as a complete number rather than incorrectly operating on only one of its parts.

5.3 Conductance, admittance, and susceptance

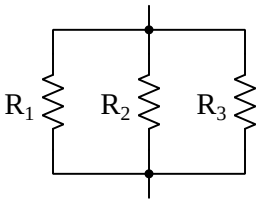
When calculating the total impedance of any parallel AC network, we know that the same reciprocal-of-reciprocals formula used for parallel resistances works if we simply use complex numbers to represent all the impedance values:

$$R_{parallel} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \cdots \frac{1}{R_n}} \qquad Z_{parallel} = \frac{1}{\frac{1}{Z_1} + \frac{1}{Z_2} + \cdots \frac{1}{Z_n}}$$

However, there is another way to approach this, using reciprocal quantities. First, we will see how this same concept applied to DC resistor networks.

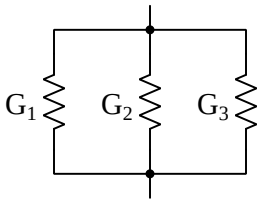
If *resistance* is the degree to which a conductive material resists the passage of electric charge carriers, then *conductance* is the degree to which that material permits their passage. We mathematically define conductance (G) as the reciprocal of resistance ($G = \frac{1}{R}$) and we measure it in the unit of *Siemens*[1] rather than Ohms. Conductance has the property of adding in parallel, much like resistance adds in series:

Resistances diminish in parallel



$$R_{parallel} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$$

Conductances add in parallel



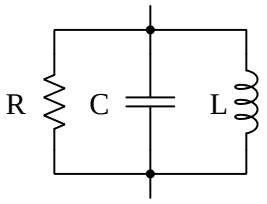
$$G_{parallel} = G_1 + G_2 + G_3$$

A close inspection of the parallel resistance formula reveals that it is really the parallel conductance formula in disguise: simply substitute G for $\frac{1}{R}$ and you will see their equivalence.

[1] Before the unit of Siemens was christened, conductance used to be expressed in *Mhos*, a "Mho" simply being "Ohm" spelled backwards.

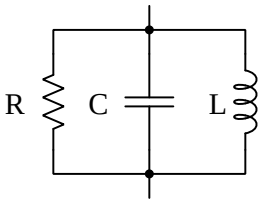
The reciprocal quantity for impedance is called *admittance* and is symbolized by the variable Y. Just as $G = \frac{1}{R}$, so $Y = \frac{1}{Z}$. Like conductance, admittance is also measured in the unit of Siemens:

Impedances diminish in parallel



$$Z_{parallel} = \frac{1}{\frac{1}{Z_R} + \frac{1}{Z_C} + \frac{1}{Z_L}}$$

Admittances add in parallel



$$Y_{parallel} = Y_R + Y_C + Y_L$$

When converting complex-number impedance values into admittance values, we see that the magnitude of each is the reciprocal of its counterpart, and that the angles are negated in sign. For example:

- 4700 Ohm resistor: $Z_R = 4700$ Ohms $\angle 0^o$ $\quad Y_R = 212.77\,\mu$S $\angle 0^o$
- 1.5 H inductor at 700 Hz: $Z_L = 6597.3$ Ohms $\angle 90^o$ $\quad Y_L = 151.58\,\mu$S $\angle -90^o$
- 0.56 μF capacitor at 60 Hz: $Z_C = 4736.8$ Ohms $\angle -90^o$ $\quad Y_C = 211.12\,\mu$S $\angle 90^o$

For the scalar quantity of reactance (X), there is even a reciprocal counterpart called *susceptance* ($B = \frac{1}{X}$) which is also measured in the unit of Siemens. Being scalar quantities rather than complex numbers, both reactance and susceptance lack phase angles. For example:

- 1.5 H inductor at 700 Hz: $X_L = 6597.3$ Ohms $\quad B_L = 151.58\,\mu$S
- 0.56 μF capacitor at 60 Hz: $X_C = 4736.8$ Ohms $\quad B_C = 211.12\,\mu$S

Chapter 6

Programming References

A powerful tool for mathematical modeling is text-based *computer programming*. This is where you type coded commands in text form which the computer is able to interpret. Many different text-based languages exist for this purpose, but we will focus here on just two of them, *C++* and *Python*.

6.1 Programming in C++

One of the more popular text-based computer programming languages is called *C++*. This is a *compiled* language, which means you must create a plain-text file containing C++ code using a program called a *text editor*, then execute a software application called a *compiler* to translate your "source code" into instructions directly understandable to the computer. Here is an example of "source code" for a very simple C++ program intended to perform some basic arithmetic operations and print the results to the computer's console:

```
#include <iostream>
using namespace std;

int main (void)
{
  float x, y;

  x = 200;
  y = -560.5;

  cout << "This simple program performs basic arithmetic on" << endl;
  cout << "the two numbers " << x << " and " << y << " and then" << endl;
  cout << "displays the results on the computer's console." << endl;

  cout << endl;

  cout << "Sum = " << x + y << endl;
  cout << "Difference = " << x - y << endl;
  cout << "Product = " << x * y << endl;
  cout << "Quotient of " << x / y << endl;

  return 0;
}
```

Computer languages such as C++ are designed to make sense when read by human programmers. The general order of execution is left-to-right, top-to-bottom just the same as reading any text document written in English. Blank lines, indentation, and other "whitespace" is largely irrelevant in C++ code, and is included only to make the code more pleasing[1] to view.

[1] Although not included in this example, *comments* preceded by double-forward slash characters (`//`) may be added to source code as well to provide explanations of what the code is supposed to do, for the benefit of anyone reading it. The compiler application will ignore all comments.

Let's examine the C++ source code to explain what it means:

- `#include <iostream>` and `using namespace std;` are set-up instructions to the compiler giving it some context in which to interpret your code. The code specific to your task is located between the brace symbols ({ and }, often referred to as "curly-braces").

- `int main (void)` labels the "Main" function for the computer: the instructions within this function (lying between the { and } symbols) it will be commanded to execute. Every complete C++ program contains a `main` function at minimum, and often additional functions as well, but the `main` function is where execution always begins. The `int` declares this function will return an *integer* number value when complete, which helps to explain the purpose of the `return 0;` statement at the end of the `main` function: providing a numerical value of zero at the program's completion as promised by `int`. This returned value is rather incidental to our purpose here, but it is fairly standard practice in C++ programming.

- Grouping symbols such as (parentheses) and {braces} abound in C, C++, and other languages (e.g. Java). Parentheses typically group data to be processed by a function, called *arguments* to that function. Braces surround lines of executable code belonging to a particular function.

- The `float` declaration reserves places in the computer's memory for two *floating-point* variables, in this case the variables' names being `x` and `y`. In most text-based programming languages, variables may be named by single letters or by combinations of letters (e.g. `xyz` would be a single variable).

- The next two lines assign numerical values to the two variables. Note how each line terminates with a semicolon character (`;`) and how this pattern holds true for most of the lines in this program. In C++ semicolons are analogous to periods at the ends of English sentences. This demarcation of each line's end is necessary because C++ ignores whitespace on the page and doesn't "know" otherwise where one line ends and another begins.

- All the other instructions take the form of a `cout` command which prints characters to the "standard output" stream of the computer, which in this case will be text displayed on the console. The double-less-than symbols (`<<`) show data being sent *toward* the `cout` command. Note how verbatim text is enclosed in quotation marks, while variables such as `x` or mathematical expressions such as `x - y` are not enclosed in quotations because we want the computer to display the numerical values represented, not the literal text.

- Standard arithmetic operations (add, subtract, multiply, divide) are represented as `+`, `-`, `*`, and `/`, respectively.

- The `endl` found at the end of every `cout` statement marks the end of a line of text printed to the computer's console display. If not for these `endl` inclusions, the displayed text would resemble a run-on sentence rather than a paragraph. Note the `cout << endl;` line, which does nothing but create a blank line on the screen, for no reason other than esthetics.

After saving this *source code* text to a file with its own name (e.g. `myprogram.cpp`), you would then *compile* the source code into an *executable* file which the computer may then run. If you are using a console-based compiler such as *GCC* (very popular within variants of the Unix operating system[2], such as Linux and Apple's OS X), you would type the following command and press the Enter key:

```
g++ -o myprogram.exe myprogram.cpp
```

This command instructs the `GCC` compiler to take your source code (`myprogram.cpp`) and create with it an executable file named `myprogram.exe`. Simply typing `./myprogram.exe` at the command-line will then execute your program:

```
./myprogram.exe
```

If you are using a graphic-based C++ development system such as Microsoft Visual Studio[3], you may simply create a new console application "project" using this software, then paste or type your code into the example template appearing in the editor window, and finally run your application to test its output.

As this program runs, it displays the following text to the console:

```
This simple program performs basic arithmetic on
the two numbers 200 and -560.5 and then
displays the results on the computer's console.

Sum = -360.5
Difference = 760.5
Product = -112100
Quotient of -0.356824
```

As crude as this example program is, it serves the purpose of showing how easy it is to write and execute simple programs in a computer using the C++ language. As you encounter C++ example programs (shown as source code) in any of these modules, feel free to directly copy-and-paste the source code text into a text editor's screen, then follow the rest of the instructions given here (i.e. save to a file, compile, and finally run your program). You will find that it is generally easier to

[2] A very functional option for users of Microsoft Windows is called *Cygwin*, which provides a Unix-like console environment complete with all the customary utility applications such as GCC!

[3] Using Microsoft Visual Studio community version 2017 at the time of this writing to test this example, here are the steps I needed to follow in order to successfully compile and run a simple program such as this: (1) Start up Visual Studio and select the option to create a New Project; (2) Select the Windows Console Application template, as this will perform necessary set-up steps to generate a console-based program which will save you time and effort as well as avoid simple errors of omission; (3) When the editing screen appears, type or paste the C++ code within the `main()` function provided in the template, deleting the "Hello World" `cout` line that came with the template; (4) Type or paste any preprocessor directives (e.g. `#include` statements, `namespace` statements) necessary for your code that did not come with the template; (5) Lastly, under the Debug drop-down menu choose either Start Debugging (F5 hot-key) or Start Without Debugging (Ctrl-F5 hotkeys) to compile ("Build") and run your new program. Upon execution a console window will appear showing the output of your program.

learn computer programming by closely examining others' example programs and modifying them than it is to write your own programs starting from a blank screen.

6.2 Programming in Python

Another text-based computer programming language called *Python* allows you to type instructions at a terminal prompt and receive immediate results without having to compile that code. This is because Python is an *interpreted* language: a software application called an *interpreter* reads your source code, translates it into computer-understandable instructions, and then executes those instructions in one step.

The following shows what happens on my personal computer when I start up the Python interpreter on my personal computer, by typing `python3`[4] and pressing the Enter key:

```
Python 3.7.2 (default, Feb 19 2019, 18:15:18)
[GCC 4.1.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The `>>>` symbols represent the prompt within the Python interpreter "shell", signifying readiness to accept Python commands entered by the user.

Shown here is an example of the same arithmetic operations performed on the same quantities, using a Python interpreter. All lines shown preceded by the `>>>` prompt are entries typed by the human programmer, and all lines shown without the `>>>` prompt are responses from the Python interpreter software:

```
>>> x = 200
>>> y = -560.5
>>> x + y
-360.5
>>> x - y
760.5
>>> x * y
-112100.0
>>> x / y
-0.35682426404995538
>>> quit()
```

[4] Using version 3 of Python, which is the latest at the time of this writing.

More advanced mathematical functions are accessible in Python by first entering the line `from math import *` which "imports" these functions from Python's math *library* (with functions identical to those available for the C programming language, and included on any computer with Python installed). Some examples show some of these functions in use, demonstrating how the Python interpreter may be used as a scientific calculator:

```
>>> from math import *
>>> sin(30.0)
-0.98803162409286183
>>> sin(radians(30.0))
0.49999999999999994
>>> pow(2.0, 5.0)
32.0
>>> log10(10000.0)
4.0
>>> e
2.7182818284590451
>>> pi
3.1415926535897931
>>> log(pow(e,6.0))
6.0
>>> asin(0.7071068)
0.78539819000368838
>>> degrees(asin(0.7071068))
45.000001524425265
>>> quit()
```

Note how trigonometric functions assume angles expressed in *radians* rather than *degrees*, and how Python provides convenient functions for translating between the two. Logarithms assume a base of e unless otherwise stated (e.g. the `log10` function for common logarithms).

The interpreted (versus compiled) nature of Python, as well as its relatively simple syntax, makes it a good choice as a person's first programming language. For complex applications, interpreted languages such as Python execute slower than compiled languages such as C++, but for the very simple examples used in these learning modules speed is not a concern.

Another Python math library is `cmath`, giving Python the ability to perform arithmetic on complex numbers. This is very useful for AC circuit analysis using *phasors*[5] as shown in the following example. Here we see Python's interpreter used as a scientific calculator to show series and parallel impedances of a resistor, capacitor, and inductor in a 60 Hz AC circuit:

```
>>> from math import *
>>> from cmath import *
>>> r = complex(400,0)
>>> f = 60.0
>>> xc = 1/(2 * pi * f * 4.7e-6)
>>> zc = complex(0,-xc)
>>> xl = 2 * pi * f * 1.0
>>> zl = complex(0,xl)
>>> r + zc + zl
(400-187.38811239154882j)
>>> 1/(1/r + 1/zc + 1/zl)
(355.837695813625+125.35793777619385j)
>>> polar(r + zc + zl)
(441.717448903332, -0.4381072059213295)
>>> abs(r + zc + zl)
441.717448903332
>>> phase(r + zc + zl)
-0.4381072059213295
>>> degrees(phase(r + zc + zl))
-25.10169387356105
```

When entering a value in rectangular form, we use the `complex()` function where the arguments are the real and imaginary quantities, respectively. If we had opted to enter the impedance values in polar form, we would have used the `rect()` function where the first argument is the magnitude and the second argument is the angle in radians. For example, we could have set the capacitor's impedance (`zc`) as $X_C \angle -90^o$ with the command `zc = rect(xc,radians(-90))` rather than with the command `zc = complex(0,-xc)` and it would have worked the same.

Note how Python defaults to rectangular form for complex quantities. Here we defined a 400 Ohm resistance as a complex value in rectangular form (400 +j0 Ω), then computed capacitive and inductive reactances at 60 Hz and defined each of those as complex (phasor) values ($0 - jX_c$ Ω and $0 + jX_l$ Ω, respectively). After that we computed total impedance in series, then total impedance in parallel. Polar-form representation was then shown for the series impedance (441.717 Ω $\angle$ -25.102^o). Note the use of different functions to show the polar-form series impedance value: `polar()` takes the complex quantity and returns its polar magnitude and phase angle in *radians*; `abs()` returns just the polar magnitude; `phase()` returns just the polar angle, once again in radians. To find the polar phase angle in degrees, we nest the `degrees()` and `phase()` functions together.

The utility of Python's interpreter environment as a scientific calculator should be clear from these examples. Not only does it offer a powerful array of mathematical functions, but also unlimited

[5] A "phasor" is a voltage, current, or impedance represented as a complex number, either in rectangular or polar form.

assignment of variables as well as a convenient text record[6] of all calculations performed which may be easily copied and pasted into a text document for archival.

It is also possible to save a set of Python commands to a text file using a text editor application, and then instruct the Python interpreter to execute it at once rather than having to type it line-by-line in the interpreter's shell. For example, consider the following Python program, saved under the filename `myprogram.py`:

```
x = 200
y = -560.5

print("Sum")
print(x + y)

print("Difference")
print(x - y)

print("Product")
print(x * y)

print("Quotient")
print(x / y)
```

As with C++, the interpreter will read this source code from left-to-right, top-to-bottom, just the same as you or I would read a document written in English. Interestingly, whitespace *is* significant in the Python language (unlike C++), but this simple example program makes no use of that.

To execute this Python program, I would need to type `python myprogram.py` and then press the Enter key at my computer console's prompt, at which point it would display the following result:

```
Sum
-360.5
Difference
760.5
Product
-112100.0
Quotient
-0.35682426405
```

As you can see, syntax within the Python programming language is simpler than C++, which is one reason why it is often a preferred language for beginning programmers.

[6] Like many command-line computing environments, Python's interpreter supports "up-arrow" recall of previous entries. This allows quick recall of previously typed commands for editing and re-evaluation.

If you are interested in learning more about computer programming in *any* language, you will find a wide variety of books and free tutorials available on those subjects. Otherwise, feel free to learn by the examples presented in these modules.

6.3 Modeling series-parallel RLC networks using C++

The following program written in C++ analyzes a series-parallel RLC circuit energized by a voltage source:

```
#include <iostream>
#include <complex>
#include <cmath>
using namespace std;

int main (void)
{
  double vsrc = 20.0;
  double f = 65.0;
  double r = 10.0;
  double l = 125e-3;
  double c = 81e-6;

  complex <double> zc, zl, zparallel, ztotal, ic, ir, vl, vrc, isrc;

  cout << "+--L--+----+ " << " V_source = " << vsrc << " Volts " << endl;
  cout << "|     |    | " << " f_source = " << f << " Hertz " << endl;
  cout << "V     R    C " << " R = " << r << " Ohms " << endl;
  cout << "|     |    | " << " L = " << l << " Henrys " << endl;
  cout << "+-----+----+ " << " C = " << c << " Farads " << endl << endl;

  zl = complex <double> (0.0, 2 * M_PI * f * l);
  zc = complex <double> (0.0, -1 / (2 * M_PI * f * c));

  zparallel = 1.0/(1.0/r + 1.0/zc);
  ztotal = zparallel + zl;
  isrc = vsrc / ztotal;
  vl = isrc * zl;
  vrc = vsrc - vl;
  ir = vrc / r;
  ic = vrc / zc;

  cout << "Z_L = " << abs (zl) << " Ohms @ " << arg (zl) * 180 /
    M_PI << " deg " << endl;
  cout << "Z_C = " << abs (zc) << " Ohms @ " << arg (zc) * 180 /
    M_PI << " deg " << endl;
  cout << "Z_total = " << abs (ztotal) << " Ohms @ " << arg (ztotal)
    * 180 / M_PI << " deg " << endl;
```

```
  cout << endl;

  cout << "Source current = " << abs (isrc) << " A @ " << arg (isrc) * 180 /
    M_PI << " deg " << endl;
  cout << "Resistor current = " << abs (ir) << " A @ " << arg (ir) * 180 /
    M_PI << " deg " << endl;
  cout << "Inductor current = " << abs (isrc) << " A @ " << arg (isrc) * 180 /
    M_PI << " deg " << endl;
  cout << "Capacitor current = " << abs (ic) << " A @ " << arg (ic) * 180 /
    M_PI << " deg " << endl;

  cout << endl;

  cout << "Apparent power = " << abs (isrc) * abs (vsrc) << " VA " << endl;
  cout << "True power = " << abs (isrc) * abs (vsrc) *
    cos (arg (ztotal)) << " W " << endl;
  cout << "Reactive power = " << abs (isrc) * abs (vsrc) *
    abs (sin (arg (ztotal))) << " VAR " << endl;
  cout << "Power factor = " << cos (arg (ztotal));

  if (arg (ztotal) < 0)
    cout << " leading";
  else
    cout << " lagging";
  cout << endl;

  return 0;
}
```

When run, the output displays as follows:

```
+--L--+----+  V_source = 20 Volts
|     |    |  f_source = 65 Hertz
V     R    C  R = 10 Ohms
|     |    |  L = 0.125 Henrys
+-----+----+  C = 8.1e-05 Farads

Z_L = 51.0509 Ohms @ 90 deg
Z_C = 30.2289 Ohms @ -90 deg
Z_total = 48.9069 Ohms @ 79.3796 deg

Source current = 0.40894 A @ -79.3796 deg
Resistor current = 0.388248 A @ -97.6843 deg
Inductor current = 0.40894 A @ -79.3796 deg
Capacitor current = 0.128436 A @ -7.68432 deg

Apparent power = 8.17881 VA
True power = 1.50736 W
Reactive power = 8.0387 VAR
Power factor = 0.184301 lagging
```

This next C++ program analyzes a four-component RLC series-parallel network energized by a voltage source:

```
#include <iostream>
#include <complex>
#include <cmath>
using namespace std;

int main (void)
{
  double vsrc = 15.0;
  double f = 50.0;
  double r1 = 470.0;
  double l1 = 250e-3;
  double l2 = 330e-3;
  double c1 = 22e-6;

  complex <double> zc1, zl1, zl2, zparallel, ztotal, il2, ir1, vl1, vc1,
    vparallel, isrc;

  cout << "+--L1--+----+  " << " V_source = " << vsrc << " Volts " << endl;
  cout << "|      |    |  " << " f_source = " << f << " Hertz " << endl;
  cout << "V      R1   L2 " << " R1 = " << r1 << " Ohms " << endl;
  cout << "|      |    |  " << " L1 = " << l1 << " Henrys " << endl;
  cout << "+--C1--+----+  " << " L2 = " << l2 << " Henrys " << endl;
  cout << "               " << " C1 = " << c1 << " Farads " << endl << endl;

  zl1 = complex <double> (0.0, 2 * M_PI * f * l1);
  zl2 = complex <double> (0.0, 2 * M_PI * f * l2);
  zc1 = complex <double> (0.0, -1 / (2 * M_PI * f * c1));

  zparallel = 1.0/(1.0/r1 + 1.0/zl2);
  ztotal = zparallel + zl1 + zc1;
  isrc = vsrc / ztotal;
  vl1 = isrc * zl1;
  vc1 = isrc * zc1;
  vparallel = isrc * zparallel;
  ir1 = vparallel / r1;
  il2 = vparallel / zl2;

  cout << "Z_L1 = " << abs (zl1) << " Ohms @ " << arg (zl1) * 180 /
    M_PI << " deg " << endl;
  cout << "Z_L2 = " << abs (zl2) << " Ohms @ " << arg (zl2) * 180 /
    M_PI << " deg " << endl;
```

```
  cout << "Z_C1 = " << abs (zc1) << " Ohms @ " << arg (zc1) * 180 /
    M_PI << " deg " << endl;
  cout << "Z_total = " << abs (ztotal) << " Ohms @ " << arg (ztotal)
    * 180 / M_PI << " deg " << endl;

  cout << endl;

  cout << "Source current = " << abs (isrc) << " A @ " << arg (isrc) * 180 /
    M_PI << " deg " << endl;
  cout << "R1 voltage = " << abs (vparallel) << " V @ " <<
    arg (vparallel) * 180 / M_PI << " deg " << endl;
  cout << "R1 current = " << abs (ir1) << " A @ " << arg (ir1) * 180 /
    M_PI << " deg " << endl;
  cout << "L1 voltage = " << abs (vl1) << " V @ " << arg (vl1) * 180 /
    M_PI << " deg " << endl;
  cout << "L1 current = " << abs (isrc) << " A @ " << arg (isrc) * 180 /
    M_PI << " deg " << endl;
  cout << "L2 voltage = " << abs (vparallel) << " V @ " <<
    arg (vparallel) * 180 / M_PI << " deg " << endl;
  cout << "L2 current = " << abs (il2) << " A @ " << arg (il2) * 180 /
    M_PI << " deg " << endl;
  cout << "C1 voltage = " << abs (vc1) << " V @ " << arg (vc1) * 180 /
    M_PI << " deg " << endl;
  cout << "C1 current = " << abs (isrc) << " A @ " << arg (isrc) * 180 /
    M_PI << " deg " << endl;

  cout << endl;

  cout << "Apparent power = " << abs (isrc) * abs (vsrc) << " VA " << endl;
  cout << "True power = " << abs (isrc) * abs (vsrc) *
    cos (arg (ztotal)) << " W " << endl;
  cout << "Reactive power = " << abs (isrc) * abs (vsrc) *
    abs (sin (arg (ztotal))) << " VAR " << endl;
  cout << "Power factor = " << cos (arg (ztotal));

  if (arg (ztotal) < 0)
    cout << " leading";
  else
    cout << " lagging";
  cout << endl;

  return 0;
}
```

When run, the output displays as follows:

```
+--L1--+----+   V_source = 15 Volts
|      |    |   f_source = 50 Hertz
V      R1   L2  R1 = 470 Ohms
|      |    |   L1 = 0.25 Henrys
+--C1--+----+   L2 = 0.33 Henrys
                C1 = 2.2e-05 Farads

Z_L1 = 78.5398 Ohms @ 90 deg
Z_L2 = 103.673 Ohms @ 90 deg
Z_C1 = 144.686 Ohms @ -90 deg
Z_total = 39.3176 Ohms @ 56.3142 deg

Source current = 0.381508 A @ -56.3142 deg
R1 voltage = 38.6235 V @ 21.2467 deg
R1 current = 0.0821776 A @ 21.2467 deg
L1 voltage = 29.9636 V @ 33.6858 deg
L1 current = 0.381508 A @ -56.3142 deg
L2 voltage = 38.6235 V @ 21.2467 deg
L2 current = 0.372553 A @ -68.7533 deg
C1 voltage = 55.199 V @ -146.314 deg
C1 current = 0.381508 A @ -56.3142 deg

Apparent power = 5.72263 VA
True power = 3.17399 W
Reactive power = 4.76175 VAR
Power factor = 0.554638 lagging
```

Chapter 7

Questions

This learning module, along with all others in the ModEL collection, is designed to be used in an inverted instructional environment where students independently read[1] the tutorials and attempt to answer questions on their own *prior* to the instructor's interaction with them. In place of lecture[2], the instructor engages with students in Socratic-style dialogue, probing and challenging their understanding of the subject matter through inquiry.

Answers are not provided for questions within this chapter, and this is by design. Solved problems may be found in the Tutorial and Derivation chapters, instead. The goal here is *independence*, and this requires students to be challenged in ways where others cannot think for them. Remember that you always have the tools of *experimentation* and *computer simulation* (e.g. SPICE) to explore concepts!

The following lists contain ideas for Socratic-style questions and challenges. Upon inspection, one will notice a strong theme of *metacognition* within these statements: they are designed to foster a regular habit of examining one's own thoughts as a means toward clearer thinking. As such these sample questions are useful both for instructor-led discussions as well as for self-study.

[1] Technical reading is an essential academic skill for any technical practitioner to possess for the simple reason that the most comprehensive, accurate, and useful information to be found for developing technical competence is in textual form. Technical careers in general are characterized by the need for continuous learning to remain current with standards and technology, and therefore any technical practitioner who cannot read well is handicapped in their professional development. An excellent resource for educators on improving students' reading prowess through intentional effort and strategy is the book textitReading For Understanding – How Reading Apprenticeship Improves Disciplinary Learning in Secondary and College Classrooms by Ruth Schoenbach, Cynthia Greenleaf, and Lynn Murphy.

[2] Lecture is popular as a teaching method because it is easy to implement: any reasonably articulate subject matter expert can talk to students, even with little preparation. However, it is also quite problematic. A good lecture always makes complicated concepts seem easier than they are, which is bad for students because it instills a false sense of confidence in their own understanding; reading and re-articulation requires more cognitive effort and serves to verify comprehension. A culture of teaching-by-lecture fosters a debilitating dependence upon direct personal instruction, whereas the challenges of modern life demand independent and critical thought made possible only by gathering information and perspectives from afar. Information presented in a lecture is ephemeral, easily lost to failures of memory and dictation; text is forever, and may be referenced at any time.

General challenges following tutorial reading

- Summarize as much of the text as you can in one paragraph of your own words. A helpful strategy is to explain ideas as you would for an intelligent child: as simple as you can without compromising too much accuracy.
- Simplify a particular section of the text, for example a paragraph or even a single sentence, so as to capture the same fundamental idea in fewer words.
- Where did the text make the most sense to you? What was it about the text's presentation that made it clear?
- Identify where it might be easy for someone to misunderstand the text, and explain why you think it could be confusing.
- Identify any new concept(s) presented in the text, and explain in your own words.
- Identify any familiar concept(s) such as physical laws or principles applied or referenced in the text.
- Devise a proof of concept experiment demonstrating an important principle, physical law, or technical innovation represented in the text.
- Devise an experiment to disprove a plausible misconception.
- Did the text reveal any misconceptions you might have harbored? If so, describe the misconception(s) and the reason(s) why you now know them to be incorrect.
- Describe any useful problem-solving strategies applied in the text.
- Devise a question of your own to challenge a reader's comprehension of the text.

General follow-up challenges for assigned problems

- Identify where any fundamental laws or principles apply to the solution of this problem, especially before applying any mathematical techniques.
- Devise a thought experiment to explore the characteristics of the problem scenario, applying known laws and principles to mentally model its behavior.
- Describe in detail your own strategy for solving this problem. How did you identify and organized the given information? Did you sketch any diagrams to help frame the problem?
- Is there more than one way to solve this problem? Which method seems best to you?
- Show the work you did in solving this problem, even if the solution is incomplete or incorrect.
- What would you say was the most challenging part of this problem, and why was it so?
- Was any important information missing from the problem which you had to research or recall?
- Was there any extraneous information presented within this problem? If so, what was it and why did it not matter?
- Examine someone else's solution to identify where they applied fundamental laws or principles.
- Simplify the problem from its given form and show how to solve this simpler version of it. Examples include eliminating certain variables or conditions, altering values to simpler (usually whole) numbers, applying a limiting case (i.e. altering a variable to some extreme or ultimate value).
- For quantitative problems, identify the real-world meaning of all intermediate calculations: their units of measurement, where they fit into the scenario at hand. Annotate any diagrams or illustrations with these calculated values.
- For quantitative problems, try approaching it qualitatively instead, thinking in terms of "increase" and "decrease" rather than definite values.
- For qualitative problems, try approaching it quantitatively instead, proposing simple numerical values for the variables.
- Were there any assumptions you made while solving this problem? Would your solution change if one of those assumptions were altered?
- Identify where it would be easy for someone to go astray in attempting to solve this problem.
- Formulate your own problem based on what you learned solving this one.

General follow-up challenges for experiments or projects

- In what way(s) was this experiment or project easy to complete?
- Identify some of the challenges you faced in completing this experiment or project.

- Show how thorough documentation assisted in the completion of this experiment or project.
- Which fundamental laws or principles are key to this system's function?
- Identify any way(s) in which one might obtain false or otherwise misleading measurements from test equipment in this system.
- What will happen if (component X) fails (open/shorted/etc.)?
- What would have to occur to make this system unsafe?

7.1 Conceptual reasoning

These questions are designed to stimulate your analytic and synthetic thinking[3]. In a Socratic discussion with your instructor, the goal is for these questions to prompt an extended dialogue where assumptions are revealed, conclusions are tested, and understanding is sharpened. Your instructor may also pose additional questions based on those assigned, in order to further probe and refine your conceptual understanding.

Questions that follow are presented to challenge and probe your understanding of various concepts presented in the tutorial. These questions are intended to serve as a guide for the Socratic dialogue between yourself and the instructor. Your instructor's task is to ensure you have a sound grasp of these concepts, and the questions contained in this document are merely a means to this end. Your instructor may, at his or her discretion, alter or substitute questions for the benefit of tailoring the discussion to each student's needs. The only absolute requirement is that each student is challenged and assessed at a level equal to or greater than that represented by the documented questions.

It is far more important that you convey your *reasoning* than it is to simply convey a correct answer. For this reason, you should refrain from researching other information sources to answer questions. What matters here is that *you* are doing the thinking. If the answer is incorrect, your instructor will work with you to correct it through proper reasoning. A correct answer without an adequate explanation of how you derived that answer is unacceptable, as it does not aid the learning or assessment process.

You will note a conspicuous lack of answers given for these conceptual questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your conceptual answers, where applicable, is to use circuit simulation software to explore the effects of changes made to circuits. For example, if one of these conceptual questions challenges you to predict the effects of altering some component parameter in a circuit, you may check the validity of your work by simulating that same parameter change within software and seeing if the results agree.

[3] *Analytical* thinking involves the "disassembly" of an idea into its constituent parts, analogous to dissection. *Synthetic* thinking involves the "assembly" of a new idea comprised of multiple concepts, analogous to construction. Both activities are high-level cognitive skills, extremely important for effective problem-solving, necessitating frequent challenge and regular practice to fully develop.

7.1.1 Reading outline and reflections

"Reading maketh a full man; conference a ready man; and writing an exact man" – Francis Bacon

Francis Bacon's advice is a blueprint for effective education: reading provides the learner with knowledge, writing focuses the learner's thoughts, and critical dialogue equips the learner to confidently communicate and apply their learning. Independent acquisition and application of knowledge is a powerful skill, well worth the effort to cultivate. To this end, students should read these educational resources closely, journal their own reflections on the reading, and discuss in detail their findings with classmates and instructor(s). You should be able to do all of the following after reading any instructional text:

☑ Briefly SUMMARIZE THE TEXT in the form of a journal entry documenting your learning as you progress through the course of study. Share this summary in dialogue with your classmates and instructor. Journaling is an excellent self-test of thorough reading because you cannot clearly express what you have not read or did not comprehend.

☑ Demonstrate ACTIVE READING STRATEGIES, including verbalizing your impressions as you read, simplifying long passages to convey the same ideas using fewer words, annotating text and illustrations with your own interpretations, working through mathematical examples shown in the text, cross-referencing passages with relevant illustrations and/or other passages, identifying problem-solving strategies applied by the author, etc. Technical reading is a special case of problem-solving, and so these strategies work precisely because they help solve any problem: paying attention to your own thoughts (metacognition), eliminating unnecessary complexities, identifying what makes sense, paying close attention to details, drawing connections between separated facts, and noting the successful strategies of others.

☑ Identify IMPORTANT THEMES, especially GENERAL LAWS and PRINCIPLES, expounded in the text and express them in the simplest of terms as though you were teaching an intelligent child. This emphasizes connections between related topics and develops your ability to communicate complex ideas to anyone.

☑ Form YOUR OWN QUESTIONS based on the reading, and then pose them to your instructor and classmates for their consideration. Anticipate both correct and incorrect answers, the incorrect answer(s) assuming one or more plausible misconceptions. This helps you view the subject from different perspectives to grasp it more fully.

☑ Devise EXPERIMENTS to test claims presented in the reading, or to disprove misconceptions. Predict possible outcomes of these experiments, and evaluate their meanings: what result(s) would confirm, and what would constitute disproof? Running mental simulations and evaluating results is essential to scientific and diagnostic reasoning.

☑ Specifically identify any points you found CONFUSING. The reason for doing this is to help diagnose misconceptions and overcome barriers to learning.

7.1.2 Foundational concepts

Correct analysis and diagnosis of electric circuits begins with a proper understanding of some basic concepts. The following is a list of some important concepts referenced in this module's tutorial. Define each of them in your own words, and be prepared to illustrate each of these concepts with a description of a practical example and/or a live demonstration.

Energy

Conservation of Energy

Voltage

Current

Source

Load

Frequency

RMS

Phasor

Resistance

Reactance

Impedance

Ohm's Law

Kirchhoff's Voltage Law

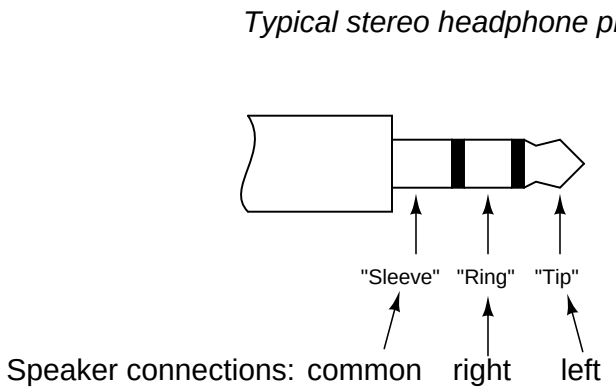
Kirchhoff's Current Law

Series network

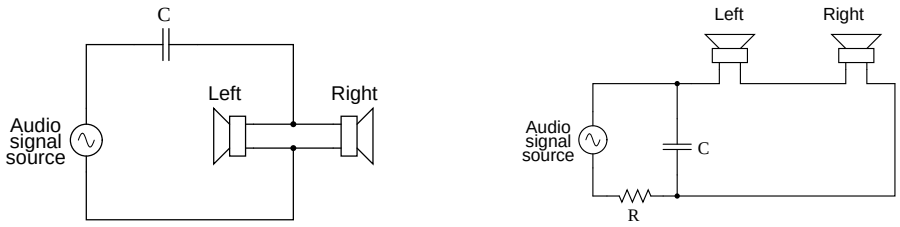
Parallel network

7.1.3 Headphone connections

Stereo (two-speaker) headphones typically use a plug with three contact points to connect the speakers to the audio amplifier. The three contact points are designated as "tip," "ring," and "sleeve" for reasons that are obvious upon inspection, and as such the plug is commonly referred to as a "TRS" plug. Both speakers in the headphone unit share a common connection (at the "sleeve" contact), with the "tip" and "ring" contacts providing connection to left and right speakers, respectively:



Sketch two pictorial diagrams, showing how connections would be made to the plug's contact points to form each of the following circuits:



Challenges

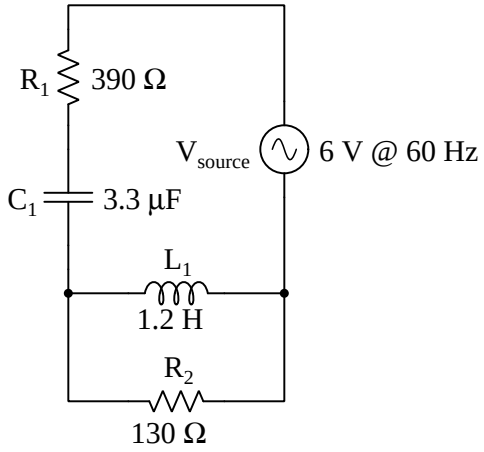
- Describe a testing procedure using an ohmmeter by which you could individually test the two speakers for "open" or "shorted" faults.

7.1.4 Explaining the meaning of calculations

An unfortunate tendency among beginning students in any quantitative discipline is to perform calculations without regard for the real-world meanings of the values, and also to follow mathematical formulae without considering the general principles embodied in each. To ignore concepts while performing calculations is a serious error for a variety of reasons, not the least of which being an increased likelihood of computing results that turn out to be nonsense.

In the spirit of honoring concepts, I present to you a quantitative problem where all the calculations have been done for you, but all variable labels, units, and other identifying data have been stripped away. Your task is to *assign proper meaning* to each of the numbers, identify the correct unit of measurement in each case, apply any appopriate metric prefixes to those values, explain the significance of each value by describing where it "fits" into the circuit being analyzed, and identify the general principle employed at each step.

Here is the schematic diagram of the circuit:



Here are all the calculations performed in order from first to last:

1. $\frac{1}{(2\pi 60)(3.3\times 10^{-6})} = 803.82$
2. $(2\pi 60)(1.2) = 452.39$
3. $\frac{1}{\frac{1}{130\ \angle 0^o}+\frac{1}{452.39\ \angle 90^o}} = 124.94\ \angle 16.03^o$
4. $124.94\ \angle 16.03^o + 390\ \angle 0^o + 803.82\ \angle -90^o = 923.05\ \angle -56.45^o$
5. $\frac{6\ \angle 0^o}{923.05\ \angle -56.45^o} = 6.500 \times 10^{-3}\ \angle 56.45^o$
6. $(6.500 \times 10^{-3}\ \angle 56.45^o)(390\ \angle 0^o) = 2.535\ \angle 56.45^o$
7. $(6.500 \times 10^{-3}\ \angle 56.45^o)(803.82\ \angle -90^o) = 5.225\ \angle -33.55^o$
8. $(6.500 \times 10^{-3}\ \angle 56.45^o)(124.94\ \angle 16.03^o) = 0.8122\ \angle 72.49^o$

9. $2.535 \angle 56.45^o + 5.225 \angle -33.55^o + 0.8122 \angle 72.49^o = 6 \angle 0^o$
10. $\frac{0.8122 \angle 72.49^o}{130 \angle 0^o} = 6.247 \times 10^{-3} \angle 72.49^o$
11. $\frac{0.8122 \angle 72.49^o}{452.39 \angle 90^o} = 1.795 \times 10^{-3} \angle -17.51^o$
12. $6.247 \times 10^{-3} \angle 72.49^o + 1.795 \times 10^{-3} \angle -17.51^o = 6.500 \times 10^{-3} \angle 56.45^o$
13. $\cos -56.45^o = 0.5526$
14. $(6)(6.500 \times 10^{-3}) = 39.00 \times 10^{-3}$
15. $(39.00 \times 10^{-3})(\cos -56.45^o) = 21.55 \times 10^{-3}$
16. $(39.00 \times 10^{-3})(\sin -56.45^o) = 32.51 \times 10^{-3}$
17. $\sqrt{(21.55 \times 10^{-3})^2 + (32.51 \times 10^{-3})^2} = 39.00 \times 10^{-3}$

Explain what each value means in the circuit, identify its unit of measurement and appropriate metric prefix, and identify the general principle used to compute it!

Challenges

- Explain how you can check your own thinking as you solve quantitative problems, to avoid the dilemma of just "crunching numbers" to get an answer.
- Do you see any alternative paths to a solution, involving specific calculations not shown above?

7.2 Quantitative reasoning

These questions are designed to stimulate your computational thinking. In a Socratic discussion with your instructor, the goal is for these questions to reveal your mathematical approach(es) to problem-solving so that good technique and sound reasoning may be reinforced. Your instructor may also pose additional questions based on those assigned, in order to observe your problem-solving firsthand.

Mental arithmetic and estimations are strongly encouraged for all calculations, because without these abilities you will be unable to readily detect errors caused by calculator misuse (e.g. keystroke errors).

You will note a conspicuous lack of answers given for these quantitative questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. My advice is to use circuit simulation software such as SPICE to check the correctness of quantitative answers. Refer to those learning modules within this collection focusing on SPICE to see worked examples which you may use directly as practice problems for your own study, and/or as templates you may modify to run your own analyses and generate your own practice problems.

Completely worked example problems found in the Tutorial may also serve as "test cases[4]" for gaining proficiency in the use of circuit simulation software, and then once that proficiency is gained you will never need to rely[5] on an answer key!

[4] In other words, set up the circuit simulation software to analyze the same circuit examples found in the Tutorial. If the simulated results match the answers shown in the Tutorial, it confirms the simulation has properly run. If the simulated results disagree with the Tutorial's answers, something has been set up incorrectly in the simulation software. Using every Tutorial as practice in this way will quickly develop proficiency in the use of circuit simulation software.

[5] This approach is perfectly in keeping with the instructional philosophy of these learning modules: *teaching students to be self-sufficient thinkers*. Answer keys can be useful, but it is even more useful to your long-term success to have a set of tools on hand for checking your own work, because once you have left school and are on your own, there will no longer be "answer keys" available for the problems you will have to solve.

7.2.1 Miscellaneous physical constants

Note: constants shown in **bold** type are *exact*, not approximations. Values inside of parentheses show one standard deviation (σ) of uncertainty in the final digits: for example, the magnetic permeability of free space value given as $1.25663706212(19) \times 10^{-6}$ H/m represents a center value (i.e. the location parameter) of $1.25663706212 \times 10^{-6}$ Henrys per meter with one standard deviation of uncertainty equal to $0.0000000000019 \times 10^{-6}$ Henrys per meter.

Avogadro's number (N_A) = **6.02214076** $\times 10^{23}$ **per mole** (mol^{-1})

Boltzmann's constant (k) = **1.380649** $\times 10^{-23}$ **Joules per Kelvin** (J/K)

Electronic charge (e) = **1.602176634** $\times 10^{-19}$ **Coulomb** (C)

Faraday constant (F) = **96,485.33212...** $\times 10^{4}$ **Coulombs per mole** (C/mol)

Magnetic permeability of free space (μ_0) = 1.25663706212(19) $\times 10^{-6}$ Henrys per meter (H/m)

Electric permittivity of free space (ϵ_0) = 8.8541878128(13) $\times 10^{-12}$ Farads per meter (F/m)

Characteristic impedance of free space (Z_0) = 376.730313668(57) Ohms (Ω)

Gravitational constant (G) = 6.67430(15) $\times 10^{-11}$ cubic meters per kilogram-seconds squared ($m^3/kg\text{-}s^2$)

Molar gas constant (R) = **8.314462618... Joules per mole-Kelvin** (J/mol-K) = 0.08205746(14) liters-atmospheres per mole-Kelvin

Planck constant (h) = **6.62607015** $\times 10^{-34}$ **joule-seconds** (J-s)

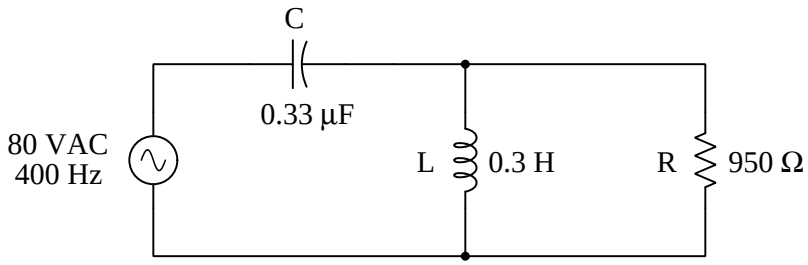
Stefan-Boltzmann constant (σ) = **5.670374419...** $\times 10^{-8}$ **Watts per square meter-Kelvin**^{4} ($W/m^2 \cdot K^4$)

Speed of light in a vacuum (c) = **299,792,458 meters per second** (m/s) = 186282.4 miles per second (mi/s)

Note: All constants taken from NIST data "Fundamental Physical Constants – Complete Listing", from `http://physics.nist.gov/constants`, National Institute of Standards and Technology (NIST), 2018 CODATA Adjustment.

7.2.2 Worked example

Let us analyze the following series-parallel RLC circuit, computing all component voltages and currents:



Our first step must be to express all these component quantities in *complex* form and store each of them in our calculator's memory[6], so that we may easily apply Ohm's Law and other principles of electric circuits.

- The source voltage of 80 Volts will become our de facto reference for phase angle, so we may express this simply as 80 Volts $\angle$ 0^o.
- Resistor impedance is simple to express, since pure resistance exhibits zero phase shift between voltage and current. Thus, Z_R is equal to 950 Ω $\angle$ 0^o.
- Inductor impedance will have a magnitude equal to inductive reactance ($X_L = 2\pi fL$) and a phase angle of 90^o since voltage always leads current by 90 degrees for a pure inductance. Thus, Z_L is equal to 753.98 Ω $\angle$ 90^o.
- Capacitor impedance will have a magnitude equal to capacitive reactance ($X_C = \frac{1}{2\pi fC}$) and a phase angle of -90^o since voltage always lags current by 90 degrees for a pure capacitance. Thus, Z_C is equal to 1205.7 Ω $\angle$ -90^o.

[6] Failing to store these intermediate calculations in memory is a recipe for error, as it practically guarantees unnecessary rounding and also invites keystroke errors when these values must be later re-entered into your calculator. Modern electronic calculators capable of complex arithmetic are able to store entire complex quantities in single memory registers, making the recall of each value as simple and as fast as specifying that memory location (usually an alphabetical letter).

Now we may begin our computations:

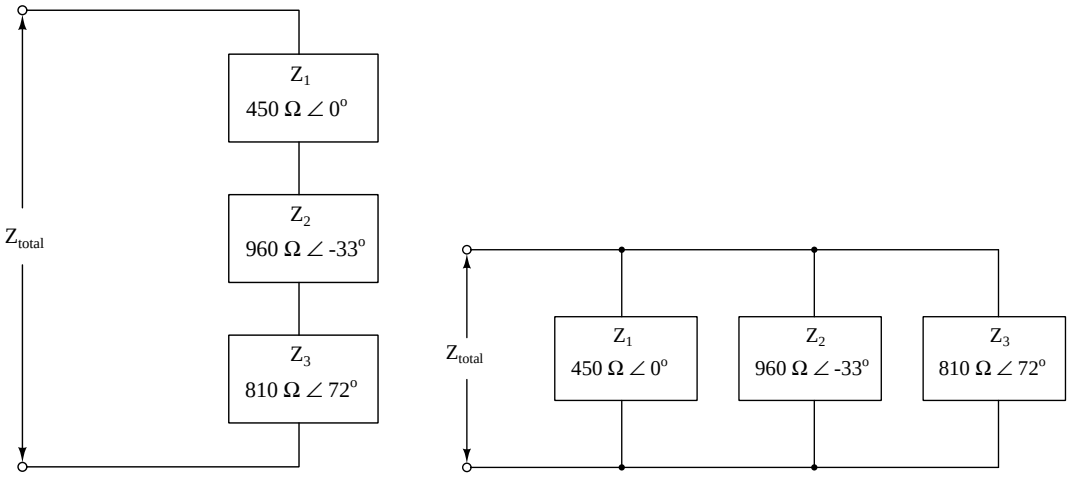
- Compute the parallel impedance of Z_L and Z_R: $Z_{L||R} = \frac{1}{\frac{1}{Z_L} + \frac{1}{Z_R}} = 590.6\ \Omega\ \angle\ 51.6^o$
- Compute total impedance: $Z_T = Z_{L||R} + Z_C = 828.9\ \Omega\ \angle\ -63.7^o$
- Apply Ohm's Law to compute total current: $I_T = \frac{V_{source}}{Z_T} = 96.52$ mA $\angle\ 63.7^o$
- Apply Ohm's Law to compute capacitor voltage: $V_C = I_T Z_C = 116.4$ V $\angle\ -26.3^o$
- Apply Ohm's Law to compute parallel inductor/resistor voltage: $V_{L||R} = I_T Z_{L||R} = 57.00$ V $\angle\ 115.3^o$
- Apply Ohm's Law to compute inductor current: $I_L = \frac{V_L}{Z_L} = 75.60$ mA $\angle\ 25.3^o$
- Apply Ohm's Law to compute resistor current: $I_R = \frac{V_R}{Z_R} = 60.00$ mA $\angle\ 115.3^o$

Finally, some applications of Kirchhoff's Laws to double-check our work:

- Apply KVL: $V_{source} = V_C + V_{L||R}$
- Apply KCL: $I_T = I_L + I_R$

7.2.3 Series and parallel network impedances

Calculate the total impedances of the following networks:

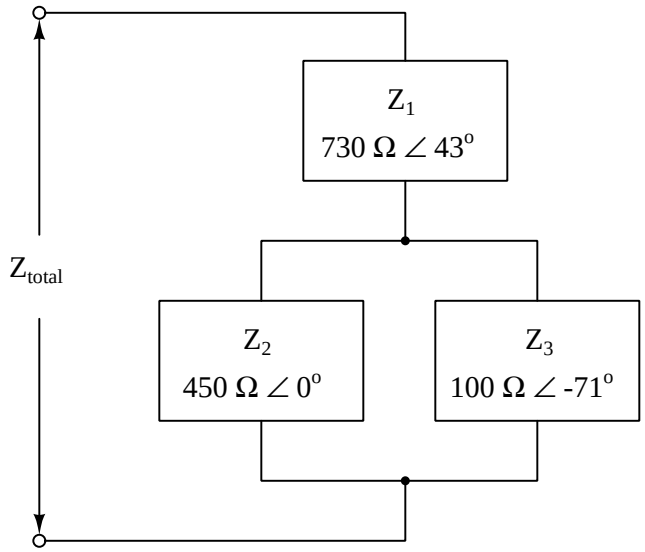


Challenges

- Is Z_1 predominantly, resistive, inductive, or capacitive?
- Is Z_2 predominantly, resistive, inductive, or capacitive?
- Is Z_3 predominantly, resistive, inductive, or capacitive?
- Which of these two networks exhibits a more ideal power factor?

7.2.4 Series-parallel network impedance

Calculate the total impedance of the following network:

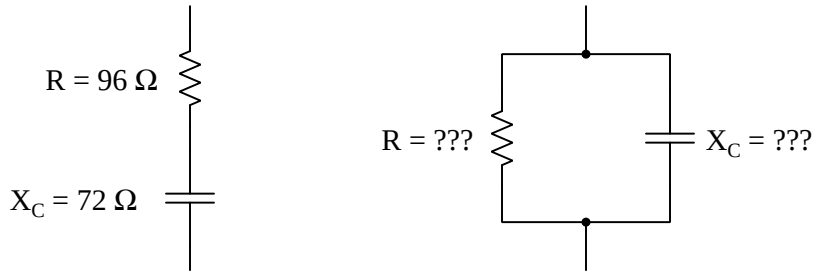


Challenges

- Is Z_1 predominantly, resistive, inductive, or capacitive?
- Is Z_2 predominantly, resistive, inductive, or capacitive?
- Is Z_3 predominantly, resistive, inductive, or capacitive?
- Identify whether each of these networks behaves more like a resistor, inductor, or capacitor.
- Calculate the power factor of this network.

7.2.5 Equivalent RC networks

Determine an equivalent *parallel* RC network for the series RC network shown on the left:



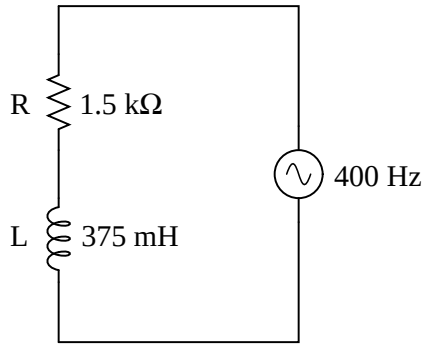
Note that I have already provided a value for the capacitor's reactance (X_C), which of course will be valid only for a particular frequency. Determine what values of resistance (R) and reactance (X_C) in the parallel network will yield the exact same total impedance (Z_T) at the same signal frequency.

Challenges

- Explain how you could check your conversion calculations, to ensure both networks are truly equivalent to each other.
- As frequency increases, what will happen to the impedance of each of these networks?

7.2.6 Equivalent LR networks

Determine the equivalent parallel-connected resistor and inductor values for this series circuit:



Also, express the total impedance of either circuit (since they are electrically equivalent to one another, they should have the same total impedance) as a complex number in both polar and rectangular forms.

Challenges

- What happens to the series network's impedance magnitude as frequency increases?
- What happens to the series network's impedance phase angle as frequency increases?

7.2.7 Current through an arbitrary AC load

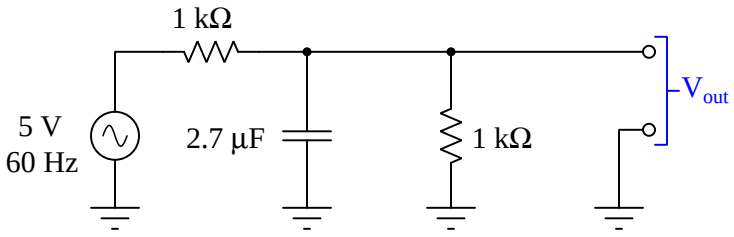
Calculate the amount of current through an AC load having an impedance of 55 kΩ $\angle$ -21^o when powered by a 30 Volt AC source.

Challenges

- Which of these two forms is more meaningful when comparing against the indication of an AC ammeter? Explain why.
- Explain why we did not need to know the *frequency* in this circuit to perform the current calculation.
- Alter this problem so that we *would* need to know circuit frequency in order to solve for current.

7.2.8 Output voltage of an RC network

Calculate V_{out} for this AC network:

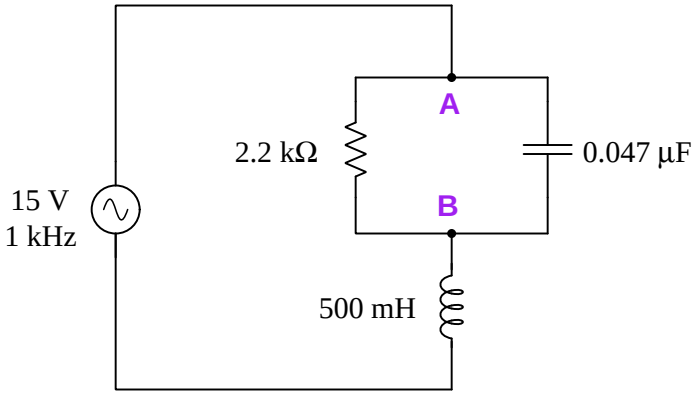


Challenges

- What will happen to V_{out} as frequency increases?

7.2.9 Voltage drop in an RC sub-network

Determine the voltage dropped between test points **A** and **B** in this circuit:



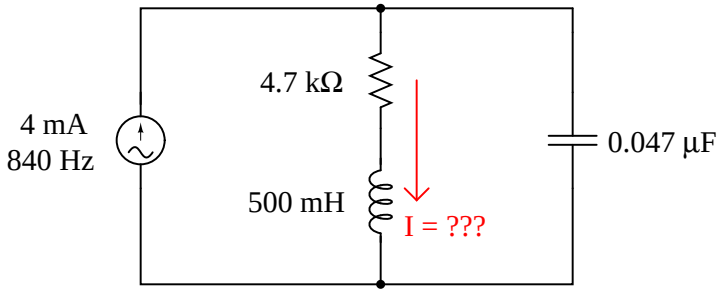
Next, calculate true power, reactive power, apparent power, and power factor for the entire circuit.

Challenges

- Calculate the same voltage drop assuming a 15 Volt DC source rather than an AC source.

7.2.10 Current through an LR branch

Determine the current through the series LR branch in this series-parallel circuit:



Challenges

- What will happen to the current through this branch as frequency increases?

7.2.11 Series and parallel equivalent networks given impedances

Scenario #1

Sketch diagrams for two networks complete with all component values, each one having the same total impedance of 22.4 kΩ $\angle$ 36^o at a frequency of 250 Hz.

Scenario #2

Sketch diagrams for two networks complete with all component values, each one having the same total impedance of 450 Ω $\angle$ -10^o at a frequency of 50 Hz.

Scenario #3

Sketch diagrams for two networks complete with all component values, each one having the same total impedance of 9.3 kΩ $\angle$ 70^o at a frequency of 2 kHz.

Scenario #4

Sketch diagrams for two networks complete with all component values, each one having the same total impedance of 8.2 kΩ − j3.5 kΩ at a frequency of 700 kHz.

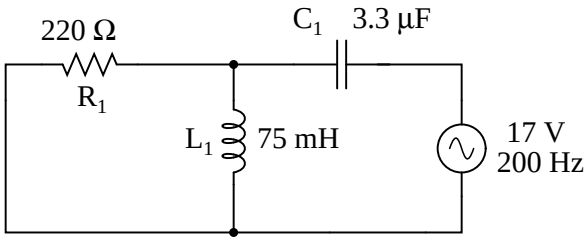
Challenges

- Which of these networks is simplest to solve from the given impedance, and why?

7.2.12 Series-parallel circuit tables

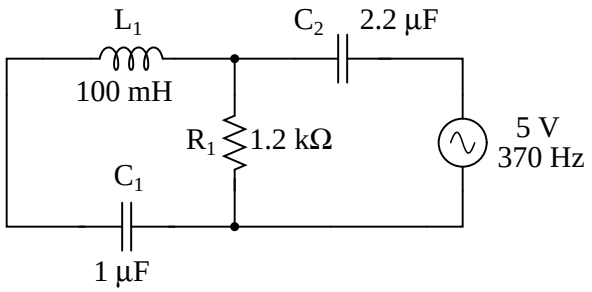
Complete the table of values for the following circuits, representing all quantities in polar form.

Circuit #1



	R_1	L_1	C_1	Total
V				
I				
Z				

Circuit #2



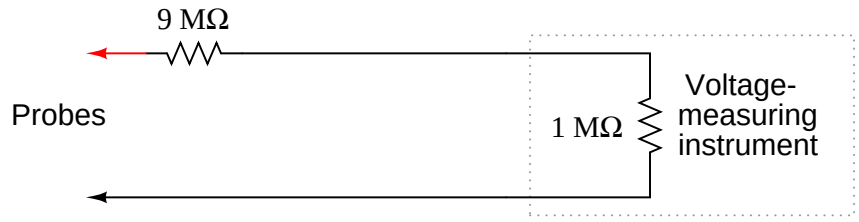
	R_1	L_1	C_1	C_2	Total
V					
I					
Z					

Challenges

- How much voltage will the resistor drop if the AC source were replaced by a DC source?
- Suppose a student calculates a Z_{total} value of 264.53 $\Omega \angle -33^o$ for circuit #1 instead of the correct value. Identify their mistake.
- Suppose a student calculates a Z_{total} value of 195.04 $\Omega \angle -80.65^o$ for circuit #2 instead of the correct value. Identify their mistake.
- Suppose a student calculates a Z_{total} value of 1262.8 $\Omega \angle -18^o$ for circuit #2 instead of the correct value. Identify their mistake.

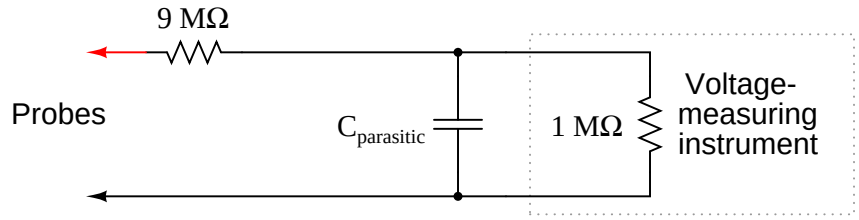
7.2.13 Voltmeter range extender

The voltage measurement range of a DC instrument may easily be "extended" by connecting an appropriately sized resistor in series with one of its test leads:



In the example shown here, the multiplication ratio with the 9 MΩ resistor in place is 10:1, meaning that an indication of 3.5 volts at the instrument corresponds to an actual measured voltage of 35 volts between the probes.

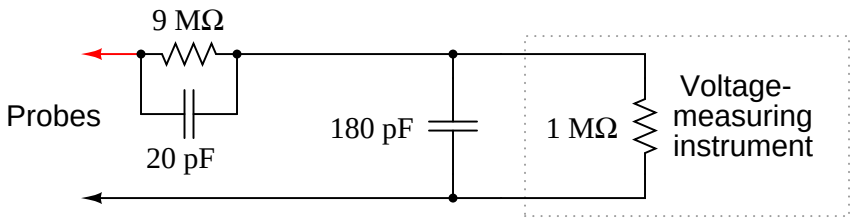
While this technique works very well when measuring DC voltage, it does not do so well when measuring AC voltage, due to the parasitic capacitance of the cable connecting the test probes to the instrument (parasitic cable *inductance* has been omitted from this diagram for simplicity):



To see the effects of this capacitance for yourself, calculate the voltage at the instrument input terminals assuming a parasitic capacitance of 180 pF and an AC voltage source of 10 volts, for the following frequencies:

- $f = 10$ Hz ; $V_{instrument} =$
- $f = 1$ kHz ; $V_{instrument} =$
- $f = 10$ kHz ; $V_{instrument} =$
- $f = 100$ kHz ; $V_{instrument} =$
- $f = 1$ MHz ; $V_{instrument} =$

The debilitating effect of cable capacitance may be compensated for with the addition of another capacitor, connected in parallel with the 9 MΩ range resistor. If we are trying to maintain a voltage division ratio of 10:1, this "compensating" capacitor must be $\frac{1}{9}$ the value of the capacitance parallel to the instrument input:



Re-calculate the voltage at the instrument input terminals with this compensating capacitor in place. You should notice quite a difference in instrument voltages across this frequency range!

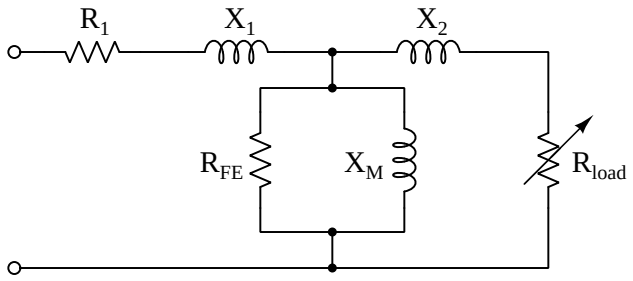
- $f = 10$ Hz ; $V_{instrument} =$
- $f = 1$ kHz ; $V_{instrument} =$
- $f = 10$ kHz ; $V_{instrument} =$
- $f = 100$ kHz ; $V_{instrument} =$
- $f = 1$ MHz ; $V_{instrument} =$

Challenges

- As you can see, the presence of a compensation capacitor is not an option for a high-frequency, 10:1 oscilloscope probe. What safety hazard(s) might arise if a probe's compensation capacitor failed in such a way that the probe behaved as if the capacitor were not there at all?

7.2.14 Induction motor model

The following diagram shows a simple model for a single-phase induction motor, which is a type of AC electric motor:



Resistor R_1 represents the resistance of the wire windings. Inductive reactances X_1 and X_2 represent *leakage inductance* caused by magnetism that does not link effectively between the stator (X_1) and rotor (X_2) to do useful work. Both R_{FE} and X_M represent magnetic characteristics of the stator's iron core, R_{FE} representing dissipative losses in the iron (e.g. hysteresis, eddy currents) and X_M representing the *magnetizing reactance* of the stator. R_{load} represents energy leaving the circuit to do actual work at the motor's shaft, and its equivalent value in Ohms varies with the mechanical load placed on the motor.

Calculate this motor's terminal current when energized by 277 Volts, assuming $R_1 = 0.075\ \Omega$, $X_1 = 0.451\ \Omega$, $X_2 = 0.509\ \Omega$, $R_{FE} = 374\ \Omega$, $X_M = 13.47\ \Omega$, and $R_{load} = 23.6\ \Omega$.

Next, calculate this motor's power factor, apparent power, true power, reactive power, and line current for the same conditions as above.

Challenges

- Would you expect R_{load} to increase or decrease with mechanical loading on the motor's shaft?
- Would you expect the motor's power factor to increase or decrease if the motor becomes mechanically unloaded (i.e. it is free to spin without having to do mechanical work)?
- Calculate the power losses (i.e. inefficiencies) within this motor under the given conditions.
- Calculate the amount of mechanical power output by this motor.
- Calculate the motor's energy efficiency under the given conditions.
- Convert this electrical model into a Thévenin equivalent so that we may plug in different values for R_{load} (simulating different mechanical load conditions) and rapidly calculate load voltages and load currents for those different resistive load values.

7.3 Diagnostic reasoning

These questions are designed to stimulate your deductive and inductive thinking, where you must apply general principles to specific scenarios (deductive) and also derive conclusions about the failed circuit from specific details (inductive). In a Socratic discussion with your instructor, the goal is for these questions to reinforce your recall and use of general circuit principles and also challenge your ability to integrate multiple symptoms into a sensible explanation of what's wrong in a circuit. Your instructor may also pose additional questions based on those assigned, in order to further challenge and sharpen your diagnostic abilities.

As always, your goal is to fully *explain* your analysis of each problem. Simply obtaining a correct answer is not good enough – you must also demonstrate sound reasoning in order to successfully complete the assignment. Your instructor's responsibility is to probe and challenge your understanding of the relevant principles and analytical processes in order to ensure you have a strong foundation upon which to build further understanding.

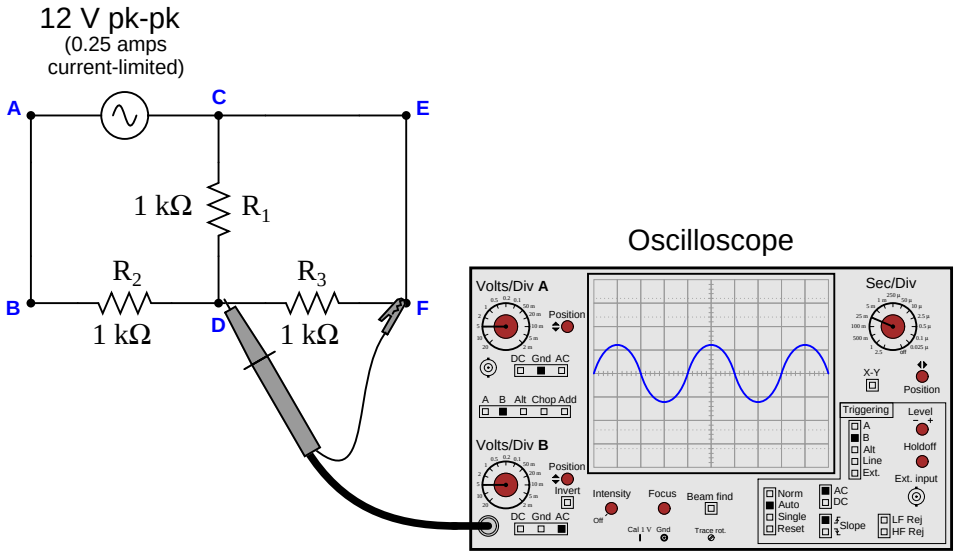
You will note a conspicuous lack of answers given for these diagnostic questions. Unlike standard textbooks where answers to every other question are given somewhere toward the back of the book, here in these learning modules students must rely on other means to check their work. The best way by far is to debate the answers with fellow students and also with the instructor during the Socratic dialogue sessions intended to be used with these learning modules. Reasoning through challenging questions with other people is an excellent tool for developing strong reasoning skills.

Another means of checking your diagnostic answers, where applicable, is to use circuit simulation software to explore the effects of faults placed in circuits. For example, if one of these diagnostic questions requires that you predict the effect of an open or a short in a circuit, you may check the validity of your work by simulating that same fault (substituting a very high resistance in place of that component for an open, and substituting a very low resistance for a short) within software and seeing if the results agree.

7.3.1 Determining possible faults in series-parallel circuits

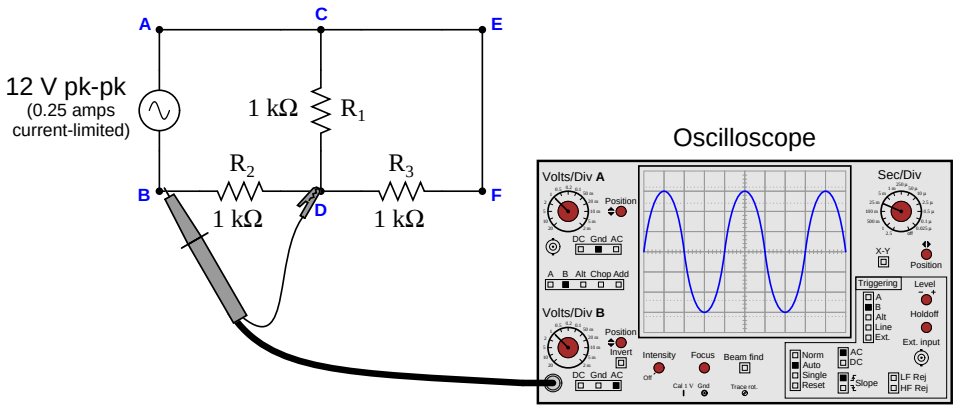
Identify the likelihood of each specified fault for these circuits. Consider each fault one at a time (i.e. no coincidental faults), determining whether or not each fault is compatible with *all* measurements and symptoms in these circuits.

Circuit #1



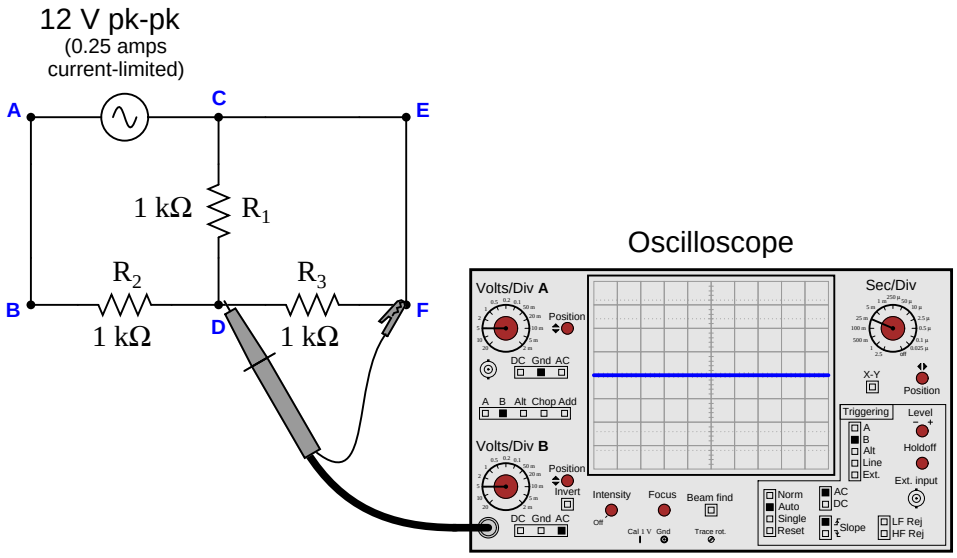
- R_1 failed open
- R_2 failed open
- R_3 failed open
- R_1 failed shorted
- R_2 failed shorted
- R_3 failed shorted
- Voltage source output too high

Circuit #2



- R_1 failed open
- R_2 failed open
- R_3 failed open
- R_1 failed shorted
- R_2 failed shorted
- R_3 failed shorted
- Voltage source dead

Circuit #3



- R_1 failed open
- R_2 failed open
- R_3 failed open
- R_1 failed shorted
- R_2 failed shorted
- R_3 failed shorted
- Voltage source dead

Challenges

- Would it matter if the circuit had a grounded point in it?

Appendix A

Problem-Solving Strategies

The ability to solve complex problems is arguably one of the most valuable skills one can possess, and this skill is particularly important in any science-based discipline.

- Study principles, not procedures. Don't be satisfied with merely knowing how to compute solutions – learn *why* those solutions work.
- Identify what it is you need to solve, identify all relevant data, identify all units of measurement, identify any general principles or formulae linking the given information to the solution, and then identify any "missing pieces" to a solution. Annotate all diagrams with this data.
- Sketch a diagram to help visualize the problem. When building a real system, always devise a plan for that system and analyze its function *before* constructing it.
- Follow the units of measurement and meaning of every calculation. If you are ever performing mathematical calculations as part of a problem-solving procedure, and you find yourself unable to apply each and every intermediate result to some aspect of the problem, it means you don't understand what you are doing. Properly done, every mathematical result should have practical meaning for the problem, and not just be an abstract number. You should be able to identify the proper units of measurement for each and every calculated result, and show where that result fits into the problem.
- Perform "thought experiments" to explore the effects of different conditions for theoretical problems. When troubleshooting real systems, perform *diagnostic tests* rather than visually inspecting for faults, the best diagnostic test being the one giving you the most information about the nature and/or location of the fault with the fewest steps.
- Simplify the problem until the solution becomes obvious, and then use that obvious case as a model to follow in solving the more complex version of the problem.
- Check for exceptions to see if your solution is incorrect or incomplete. A good solution will work for *all* known conditions and criteria. A good example of this is the process of testing scientific hypotheses: the task of a scientist is not to find support for a new idea, but rather to *challenge* that new idea to see if it holds up under a battery of tests. The philosophical

principle of *reductio ad absurdum* (i.e. disproving a general idea by finding a specific case where it fails) is useful here.

- Work "backward" from a hypothetical solution to a new set of given conditions.
- Add quantities to problems that are qualitative in nature, because sometimes a little math helps illuminate the scenario.
- Sketch graphs illustrating how variables relate to each other. These may be quantitative (i.e. with realistic number values) or qualitative (i.e. simply showing increases and decreases).
- Treat quantitative problems as qualitative in order to discern the relative magnitudes and/or directions of change of the relevant variables. For example, try determining what happens if a certain variable were to increase or decrease before attempting to precisely calculate quantities: how will each of the dependent variables respond, by increasing, decreasing, or remaining the same as before?
- Consider limiting cases. This works especially well for qualitative problems where you need to determine which direction a variable will change. Take the given condition and magnify that condition to an extreme degree as a way of simplifying the direction of the system's response.
- Check your work. This means regularly testing your conclusions to see if they make sense. This does *not* mean repeating the same steps originally used to obtain the conclusion(s), but rather to use some other means to check validity. Simply repeating procedures often leads to *repeating the same errors* if any were made, which is why alternative paths are better.

Appendix B

Instructional philosophy

B.1 First principles of learning

- **Anyone can learn anything** given appropriate time, effort, resources, challenges, encouragement, and expectations. Dedicating time and investing effort are the student's responsibility; providing resources, challenges, and encouragement are the teacher's responsibility; high expectations are a responsibility shared by both student and teacher.

- **Transfer is not automatic.** The human mind has a natural tendency to compartmentalize information, which means the process of taking knowledge learned in one context and applying it to another usually does not come easy and therefore should never be taken for granted.

- **Learning is iterative.** The human mind rarely learns anything perfectly on the first attempt. Anticipate mistakes and plan for multiple tries to achieve full understanding, using the lessons of those mistakes as feedback to guide future attempts.

- **Information is absorbed, but understanding is created.** Facts and procedures may be memorized easily enough by repeated exposure, but the ability to reliably apply principles to novel scenarios only comes through intense personal effort. This effort is fundamentally creative in nature: casting new concepts in one's own words, running mental experiments to perceive patterns and trends, etc. These acts of making knowledge "one's own" need not be perfect in order to be effective, as the value lies in the activity and not necessarily the finished product. Depth of understanding and permanence of memory are both directly related to mental effort during the learning process – i.e. *easy learning is shallow and fleeting.*

- **Education trumps training.** There is no such thing as an entirely isolated subject, as all fields of knowledge are connected. Training is narrowly-focused and task-oriented. Education is broad-based and principle-oriented. When preparing for a life-long technical career, education beats training every time.

- **Character matters.** Poor habits are more destructive than deficits of knowledge or skill. This is especially true in collective endeavors, where a team's ability to function depends on trust between its members. Simply put, no one wants an untrustworthy person on their team. An essential component of education then, is character development.

- **People learn to be responsible by bearing responsibility.** An irresponsible person is someone who has never *had* to be responsible for anything that mattered enough to them. Just as anyone can learn anything, anyone can become responsible if the personal cost of irresponsibility becomes high enough.

- **What gets measured, gets done.** Accurate and relevant assessment of learning is key to ensuring all students learn. Therefore, it is imperative to measure what matters.

- **Failure is nothing to fear.** Every human being fails, and fails in multiple ways at multiple times. Eventual success only happens when we don't stop trying.

B.2 Proven strategies for instructors

- Assume every student is capable of learning anything they desire given the proper conditions. Treat them as capable adults by granting real responsibility and avoiding artificial incentives such as merit or demerit points.
- Create a consistent culture of high expectations across the entire program of study. Demonstrate and encourage patience, persistence, and a healthy sense of self-skepticism. Anticipate and de-stigmatize error. Teach respect for the capabilities of others as well as respect for one's own fallibility.
- Replace lecture with "inverted" instruction, where students first encounter new concepts through reading and then spend class time in Socratic dialogue with the instructor exploring those concepts and solving problems individually. There is a world of difference between observing someone solve a problem versus actually solving a problem yourself, and so the point of this form of instruction is to place students in a position where they *cannot* passively observe.
- Require students to read extensively, write about what they learn, and dialogue with you and their peers to sharpen their understanding. Apply Francis Bacon's advice that "reading maketh a full man; conference a ready man; and writing an exact man". These are complementary activities helping students expand their confidence and abilities.
- Use artificial intelligence (AI) to challenge student understanding rather than merely provide information. Find productive ways for AI to critique students' clarity of thought and of expression, for example by employing AI as a Socratic-style interlocutor or as a reviewer of students' journals. Properly applied, AI has the ability to expand student access to critical review well outside the bounds of their instructor's reach.
- Build frequent and rapid feedback into the learning process so that students know at all times how well they are learning, to identify problems early and fix them before they grow. Model the intellectual habit of self-assessing and self-correcting your own understanding (i.e. a cognitive *feedback loop*), encouraging students to do the same.
- Use "mastery" as the standard for every assessment, which means the exam or experiment or project must be done with 100% competence in order to pass. Provide students with multiple opportunity for re-tries (different versions of the assessment every time).
- Require students to devise their own hypotheses and procedures on all experiments, so that the process is truly a scientific one. Have students assess their proposed experimental procedures for risk and devise mitigations for those risks. Let nothing be pre-designed about students' experiments other than a stated task (i.e. what principle the experiment shall test) at the start and a set of demonstrable knowledge and skill objectives at the end.
- Have students build as much of their lab equipment as possible: building power sources, building test assemblies[1], and building complete working systems (no kits!). In order to provide

[1] In the program I teach, every student builds their own "Development Board" consisting of a metal chassis with DIN rail, terminal blocks, and an AC-DC power supply of their own making which functions as a portable lab environment they can use at school as well as take home.

this same "ground-up" experience for every new student, this means either previous students take their creations with them, or the systems get disassembled in preparation for the new students, or the systems grow and evolve with each new student group.

- Incorporate external accountability for you and for your students, continuously improving the curriculum and your instructional methods based on proven results. Have students regularly network with active professionals through participation in advisory committee meetings, service projects, tours, jobshadows, internships, etc. Practical suggestions include requiring students to design and build projects for external clients (e.g. community groups, businesses, different departments within the institution), and also requiring students attend all technical advisory committee meetings and to dialogue with the industry representatives at those meetings.
- Repeatedly explore difficult-to-learn concepts across multiple courses, so that students have multiple opportunities to build their understanding.
- Relate all new concepts, whenever possible, to previous concepts and to relevant physical laws. Challenge each and every student, every day, to *reason* from concept to concept and to explain the logical connections between. Challenge students to verify their conclusions by multiple approaches (e.g. double-checking their work using different methods). Ask *"Why?"* often.
- Maintain detailed records on each student's performance and share these records privately with them. These records should include academic performance as well as professionally relevant behavioral tendencies.
- Hold mandatory "check-in" meetings between all program faculty and each new student during their first term. Offer these to all other students as an option, except for any students continuing to manifest unprofessional behaviors, poor academic performance, or who have some other need for a face-to-face meeting with faculty.
- Address problems while they are small, before they grow larger. This is equally true for tutoring technical concepts as it is for helping students build professional habits.
- Build rigorous quality control into the curriculum to ensure every student masters every important concept, and that the mastery is retained over time. This includes (1) review questions added to every exam to re-assess knowledge taught in previous terms, (2) cumulative exams at the end of every term to re-assess all important concepts back to the very beginning of the program, and (3) review assessments in practical (hands-on) coursework to ensure critically-important skills were indeed taught and are still retained. What you will find by doing this is that it actually boosts retention of students by ensuring that important knowledge gets taught and is retained over long spans of time. In the absence of such quality control, student learning and retention tends to be spotty and this contributes to drop-out and failure rates later in their education.
- Finally, *never rush learning.* Education is not a race. Give your students ample time to digest complex ideas, as you continually remind yourself of just how long it took you to achieve mastery! Long-term retention and the consistently correct application of concepts are always the result of *focused effort over long periods of time* which means there are no shortcuts to learning.

B.3 Proven strategies for students

The single most important piece of advice I have for any student of any subject is to take responsibility for your own development in all areas of life including mental development. Expecting others in your life to entirely guide your own development is a recipe for disappointment. This is just as true for students enrolled in formal learning institutions as it is for auto-didacts pursuing learning entirely on their own. Learning to think in new ways is key to being able to gainfully use information, to make informed decisions about your life, and to best serve those you care about. With this in mind, I offer the following advice to students:

- **Approach all learning as valuable**. No matter what course you take, no matter who you learn from, no matter the subject, there is something useful in every learning experience. If you don't see the value of every new experience, you are not looking closely enough!
- **Continually challenge yourself**. Let other people take shortcuts and find easy answers to easy problems. The purpose of education is to stretch your mind, in order to shape it into a more powerful tool. This doesn't come by taking the path of least resistance. An excellent analogy for an empowering education is productive physical exercise: becoming stronger, more flexible, and more persistent only comes through intense personal effort.
- **Master the use of language**. This includes reading extensively, writing every day, listening closely, and speaking articulately. To a great extent language channels and empowers thought, so the better you are at wielding language the better you will be at grasping abstract concepts and articulating them not only for your benefit but for others as well.
- **Do not limit yourself to the resources given to you**. Read books that are not on the reading list. Run experiments that aren't assigned to you. Form study groups outside of class. Take an entrepreneurial approach to your own education, as though it were a business you were building for your future benefit.
- **Express and share what you learn**. Take every opportunity to teach what you have learned to others, as this will not only help them but will also strengthen your own understanding[2].
- Realize that **no one can give you understanding**, just as no one can give you physical fitness. These both must be *built*.
- **Above all, recognize that learning is hard work, and that a certain level of frustration is unavoidable**. There are times when you will struggle to grasp some of these concepts, and that struggle is a natural thing. Take heart that it will yield with persistent and varied[3] effort, and never give up! That concepts don't immediately come to you is not a sign of something wrong, but rather of something right: that you have found a worthy challenge!

[2] On a personal note, I was surprised to learn just how much my own understanding of electronics and related subjects was strengthened by becoming a teacher. When you are tasked every day with helping other people grasp complex topics, it catalyzes your own learning by giving you powerful incentives to study, to articulate your thoughts, and to reflect deeply on the process of learning.

[3] As the old saying goes, "Insanity is trying the same thing over and over again, expecting different results." If you find yourself stumped by something in the text, you should attempt a different approach. Alter the thought experiment, change the mathematical parameters, do whatever you can to see the problem in a slightly different light, and then the solution will often present itself more readily.

B.4 Design of these learning modules

"The unexamined circuit is not worth energizing" – Socrates (if he had taught electricity)

These learning modules, although useful for self-study, were designed to be used in a formal learning environment where a subject-matter expert challenges students to digest the content and exercise their critical thinking abilities in the answering of questions and in the construction and testing of working circuits. Every effort has been made to embed the following instructional and assessment philosophies within:

- The first goal of education is to enhance clear and independent thought, in order that every student reach their fullest potential in a highly complex and inter-dependent world. Robust reasoning is *always* more important than particulars of any subject matter, because its application is universal.
- Literacy is fundamental to independent learning and thought because text continues to be the most efficient way to communicate complex ideas over space and time. Those who cannot read with ease are limited in their ability to acquire knowledge and perspective.
- Articulate communication is fundamental to work that is complex and interdisciplinary.
- Faulty assumptions and poor reasoning are best corrected through challenge, not presentation. The rhetorical technique of *reductio ad absurdum* (disproving an assertion by exposing an absurdity) works well to discipline student's minds, not only to correct the problem at hand but also to learn how to detect and correct future errors.
- Important principles should be repeatedly explored and widely applied throughout a course of study, not only to reinforce their importance and help ensure their mastery, but also to showcase the interconnectedness and utility of knowledge.

These learning modules were expressly designed to be used in an "inverted" teaching environment[4] where students first read the introductory and tutorial chapters on their own, then individually attempt to answer the questions and construct working circuits according to the experiment and project guidelines. The instructor never lectures, but instead meets regularly with each individual student to review their progress, answer questions, identify misconceptions, and challenge the student to new depths of understanding through further questioning. Regular meetings between instructor and student should resemble a Socratic[5] dialogue, where questions serve as scalpels to dissect topics and expose assumptions. The student passes each module only after consistently demonstrating their ability to logically analyze and correctly apply all major concepts in each question or project/experiment. The instructor must be vigilant in probing each student's understanding to ensure they are truly *reasoning* and not just *memorizing*. This is why "Challenge" points appear throughout, as prompts for students to think deeper about topics and as starting points for instructor queries. Sometimes these challenge points require additional knowledge that hasn't been covered in the series to answer in full. This is okay, as the major purpose of the Challenges is to stimulate analysis and synthesis on the part of each student.

The instructor must possess enough mastery of the subject matter and awareness of students' reasoning to generate their own follow-up questions to practically any student response. Even completely correct answers given by the student should be challenged by the instructor for the purpose of having students practice articulating their thoughts and defending their reasoning. Conceptual errors committed by the student should be exposed and corrected not by direct instruction, but rather by reducing the errors to an absurdity[6] through well-chosen questions and thought experiments posed by the instructor. Becoming proficient at this style of instruction requires time and dedication, but the positive effects on critical thinking for both student and instructor are spectacular.

An inspection of these learning modules reveals certain unique characteristics. One of these is a bias toward thorough explanations in the tutorial chapters. Without a live instructor to explain concepts and applications to students, the text itself must fulfill this role. This philosophy results in lengthier explanations than what you might typically find in a textbook, each step of the reasoning process fully explained, including footnotes addressing common questions and concerns students raise while learning these concepts. Each tutorial seeks to not only explain each major concept in sufficient detail, but also to explain the logic of each concept and how each may be developed

[4] In a traditional teaching environment, students first encounter new information via *lecture* from an expert, and then independently apply that information via *homework*. In an "inverted" course of study, students first encounter new information via *homework*, and then independently apply that information under the scrutiny of an expert. The expert's role in lecture is to simply *explain*, but the expert's role in an inverted session is to *challenge*, *critique*, and if necessary *explain* where gaps in understanding still exist.

[5] Socrates is a figure in ancient Greek philosophy famous for his unflinching style of questioning. Although he authored no texts, he appears as a character in Plato's many writings. The essence of Socratic philosophy is to leave no question unexamined and no point of view unchallenged. While purists may argue a topic such as electric circuits is too narrow for a true Socratic-style dialogue, I would argue that the essential thought processes involved with scientific reasoning on *any* topic are not far removed from the Socratic ideal, and that students of electricity and electronics would do very well to challenge assumptions, pose thought experiments, identify fallacies, and otherwise employ the arsenal of critical thinking skills modeled by Socrates.

[6] This rhetorical technique is known by the Latin phrase *reductio ad absurdum*. The concept is to expose errors by counter-example, since only one solid counter-example is necessary to disprove a universal claim. As an example of this, consider the common misconception among beginning students of electricity that voltage cannot exist without current. One way to apply *reductio ad absurdum* to this statement is to ask how much current passes through a fully-charged battery connected to nothing (i.e. a clear example of voltage existing without current).

from "first principles". Again, this reflects the goal of developing clear and independent thought in students' minds, by showing how clear and logical thought was used to forge each concept. Students benefit from witnessing a model of clear thinking in action, and these tutorials strive to be just that.

Another feature of these learning modules is that they do not treat topics in isolation. Rather, important concepts are introduced early in the series, and appear repeatedly as stepping-stones toward other concepts in subsequent modules. This helps to avoid the "compartmentalization" of knowledge, demonstrating the inter-connectedness of concepts and simultaneously reinforcing them. Each module is fairly complete in itself, reserving the beginning of its tutorial to a review of foundational concepts.

To high standards of education,

Tony R. Kuphaldt

Appendix C

Tools used

I am indebted to the developers of many open-source software applications in the creation of these learning modules. The following is a list of these applications with some commentary on each.

You will notice a theme common to many of these applications: a bias toward *code*. Although I am by no means an expert programmer in any computer language, I understand and appreciate the flexibility offered by code-based applications where the user (you) enters commands into a plain ASCII text file, which the software then reads and processes to create the final output. Code-based computer applications are by their very nature *extensible*, while WYSIWYG (What You See Is What You Get) applications are generally limited to whatever user interface the developer makes for you.

The `GNU/Linux` computer operating system

> There is so much to be said about Linus Torvalds' `Linux` and Richard Stallman's `GNU` project. First, to credit just these two individuals is to fail to do justice to the *mob* of passionate volunteers who contributed to make this amazing software a reality. I first learned of `Linux` back in 1996, and have been using this operating system on my personal computers almost exclusively since then. It is *free*, it is completely *configurable*, and it permits the continued use of highly efficient `Unix` applications and scripting languages (e.g. shell scripts, Makefiles, `sed`, `awk`) developed over many decades. `Linux` not only provided me with a powerful computing platform, but its open design served to inspire my life's work of creating open-source educational resources.

Bram Moolenaar's `Vim` text editor

> Writing code for any code-based computer application requires a *text editor*, which may be thought of as a word processor strictly limited to outputting plain-ASCII text files. Many good text editors exist, and one's choice of text editor seems to be a deeply personal matter within the programming world. I prefer `Vim` because it operates very similarly to `vi` which is ubiquitous on `Unix`/`Linux` operating systems, and because it may be entirely operated via keyboard (i.e. no mouse required) which makes it fast to use.

Donald Knuth's TEX typesetting system

Developed in the late 1970's and early 1980's by computer scientist extraordinaire Donald Knuth to typeset his multi-volume magnum opus *The Art of Computer Programming*, this software allows the production of formatted text for screen-viewing or paper printing, all by writing plain-text code to describe how the formatted text is supposed to appear. TEX is not just a markup language for documents, but it is also a Turing-complete programming language in and of itself, allowing useful algorithms to be created to control the production of documents. Simply put, TEX *is a programmer's approach to word processing*. Since TEX is controlled by code written in a plain-text file, this means anyone may read that plain-text file to see exactly how the document was created. This openness afforded by the code-based nature of TEX makes it relatively easy to learn how other people have created their own TEX documents. By contrast, examining a beautiful document created in a conventional WYSIWYG word processor such as Microsoft `Word` suggests nothing to the reader about *how* that document was created, or what the user might do to create something similar. As Mr. Knuth himself once quipped, conventional word processing applications should be called WYSIAYG (What You See Is *All* You Get).

Leslie Lamport's LATEX extensions to TEX

Like all true programming languages, TEX is inherently extensible. So, years after the release of TEX to the public, Leslie Lamport decided to create a massive extension allowing easier compilation of book-length documents. The result was LATEX, which is the markup language used to create all ModEL module documents. You could say that TEX is to LATEX as `C` is to `C++`. This means it is permissible to use any and all TEX commands within LATEX source code, and it all still works. Some of the features offered by LATEX that would be challenging to implement in TEX include automatic index and table-of-content creation.

Tim Edwards' `Xcircuit` drafting program

This wonderful program is what I use to create all the schematic diagrams and illustrations (but not photographic images or mathematical plots) throughout the ModEL project. It natively outputs PostScript format which is a true vector graphic format (this is why the images do not pixellate when you zoom in for a closer view), and it is so simple to use that I have never had to read the manual! Object libraries are easy to create for `Xcircuit`, being plain-text files using PostScript programming conventions. Over the years I have collected a large set of object libraries useful for drawing electrical and electronic schematics, pictorial diagrams, and other technical illustrations.

`Gimp` graphic image manipulation program

Essentially an open-source clone of Adobe's `PhotoShop`, I use `Gimp` to resize, crop, and convert file formats for all of the photographic images appearing in the ModEL modules. Although `Gimp` does offer its own scripting language (called `Script-Fu`), I have never had occasion to use it. Thus, my utilization of `Gimp` to merely crop, resize, and convert graphic images is akin to using a sword to slice bread.

`SPICE` circuit simulation program

`SPICE` is to circuit analysis as TEX is to document creation: it is a form of markup language designed to describe a certain object to be processed in plain-ASCII text. When the plain-text "source file" is compiled by the software, it outputs the final result. More modern circuit analysis tools certainly exist, but I prefer `SPICE` for the following reasons: it is *free*, it is *fast*, it is *reliable*, and it is a fantastic tool for *teaching* students of electricity and electronics how to write simple code. I happen to use rather old versions of `SPICE`, version 2g6 being my "go to" application when I only require text-based output. `NGSPICE` (version 26), which is based on Berkeley `SPICE` version 3f5, is used when I require graphical output for such things as time-domain waveforms and Bode plots. In all `SPICE` example netlists I strive to use coding conventions compatible with all `SPICE` versions.

Andrew D. Hwang's `ePiX` mathematical visualization programming library

This amazing project is a `C++` library you may link to any `C/C++` code for the purpose of generating PostScript graphic images of mathematical functions. As a completely free and open-source project, it does all the plotting I would otherwise use a Computer Algebra System (CAS) such as `Mathematica` or `Maple` to do. It should be said that `ePiX` is *not* a Computer Algebra System like `Mathematica` or `Maple`, but merely a mathematical *visualization* tool. In other words, it won't determine integrals for you (you'll have to implement that in your own `C/C++` code!), but it can graph the results, and it does so beautifully. What I really admire about `ePiX` is that it is a `C++` programming library, which means it builds on the existing power and toolset available with that programming language. Mr. Hwang could have probably developed his own stand-alone application for mathematical plotting, but by creating a `C++` library to do the same thing he accomplished something much greater.

`gnuplot` mathematical visualization software

Another open-source tool for mathematical visualization is `gnuplot`. Interestingly, this tool is *not* part of Richard Stallman's GNU project, its name being a coincidence. For this reason the authors prefer "gnu" *not* be capitalized at all to avoid confusion. This is a much "lighter-weight" alternative to a spreadsheet for plotting tabular data, and the fact that it easily outputs directly to an X11 console or a file in a number of different graphical formats (including PostScript) is very helpful. I typically set my `gnuplot` output format to default (X11 on my Linux PC) for quick viewing while I'm developing a visualization, then switch to PostScript file export once the visual is ready to include in the document(s) I'm writing. As with my use of `Gimp` to do rudimentary image editing, my use of `gnuplot` only scratches the surface of its capabilities, but the important points are that it's *free* and that it *works well*.

`Python` programming language

Both Python and C++ find extensive use in these modules as instructional aids and exercises, but I'm listing Python here as a *tool* for myself because I use it almost daily as a *calculator*. If you open a Python interpreter console and type `from math import *` you can type mathematical expressions and have it return results just as you would on a hand calculator. Complex-number (i.e. *phasor*) arithmetic is similarly supported if you include the complex-math library (`from cmath import *`). Examples of this are shown in the Programming References chapter (if included) in each module. Of course, being a fully-featured programming language, Python also supports conditionals, loops, and other structures useful for calculation of quantities. Also, running in a console environment where all entries and returned values show as text in a chronologically-ordered list makes it easy to copy-and-paste those calculations to document exactly how they were performed.

Appendix D

Creative Commons License

Creative Commons Attribution 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

d. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

e. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or

limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

f. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

g. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

h. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

i. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

j. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

k. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. License grant.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

A. reproduce and Share the Licensed Material, in whole or in part; and

B. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures.

For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

B. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. Other rights.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. Attribution.

1. If You Share the Licensed Material (including in modified form), You must:

A. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

4. If You Share Adapted Material You produce, the Adapter's License You apply must not prevent recipients of the Adapted Material from complying with this Public License.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material; and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors,

whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

b. To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

a. For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully

be made without permission under this Public License.

b. To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

c. No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

d. Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at `creativecommons.org/policies`, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at `creativecommons.org`.

Appendix E

References

"Product Information Packet: IDM4110T", Baldor Electric Company, 1 March 2013.

Appendix F

Version history

This is a list showing all significant additions, corrections, and other edits made to this learning module. Each entry is referenced by calendar date in reverse chronological order (newest version first), which appears on the front cover of every learning module for easy reference. Any contributors to this open-source document are listed here as well.

19 February 2026 – grammatical improvement in one of the sub-questions of the "Induction motor model" Quantitative Reasoning question courtesy of Wyatt Gleason.

9 September 2024 – divided the Introduction chapter into sections, one with recommendations for students, one with a listing of challenging concepts, and one with recommendations for instructors.

9 February 2024 – added power calculations to the "Induction motor model" Quantitative Reasoning question.

22 June 2023 – added a requirement in the "Explaining the meaning of calculations" Conceptual Question to also express each quantity using appropriate metric prefixes.

14-15 February 2023 – corrected a minor typographical error in one of the Challenge questions, and added a question to the "Voltage drop in an RC sub-network" Quantitative Reasoning problem asking to solve for power factor and the three different powers (S, P, and Q). Also added more Challenge questions.

10 February 2023 – added a Programming References chapter with examples of series-parallel circuits being analyzed using C++.

28 November 2022 – placed questions at the top of the itemized list in the Introduction chapter prompting students to devise experiments related to the tutorial content.

16 May 2022 – added a Case Tutorial section showing the effective use of a triangle-wave signal generator rather than a sine-wave signal generator to provide AC excitatin in simple RLC network experiments. It's not "perfect" like a sine-wave signal generator would (ideally) be, but the results are quite close to what they ought to be, usually within the tolerances of the components.

23 December 2021 – replaced review_05 with review_16 for the Full Tutorial's review of phasor analysis.

8 May 2021 – commented out or deleted empty chapters.

21 February 2021 – added a Technical Reference on parallel conductance and admittance.

17 February 2021 – added some more instructor notes to some of the Quantitative questions, showing intermediate results in the calculations necessary to arrive at final answers.

15 February 2021 – minor edits to some of the questions, including "Explaining the meaning of calculations" Conceptual question where I should have taken the cosine of a *negative* angle as well as some edits to Challenge questions.

5 January 2021 – added comments to the Full Tutorial about phasor addition, for series voltages and series impedances and parallel currents being equivalent to stacking phasors tip-to-tail in a phasor diagram. Also added a Case Tutorial chapter.

4 November 2020 – minor additions to the Introduction chapter, and additions to the Full Tutorial recommending students use their calculator memory storage for intermediate results in lengthy problems such as these. Also, shortened section title to "Example circuit analysis".

8 October 2020 – significantly edited the Introduction chapter to make it more suitable as a pre-study guide and to provide cues useful to instructors leading "inverted" teaching sessions.

8 April 2020 – minor edits to the Full Tutorial.

2 April 2020 – minor modification to the illustration in "Output voltage of an RC network" Quantitative Reasoning problem.

27 May 2019 – added more detail to the review section, which caused page formatting to change a bit in the final document. Also, added questions to the Conceptual, Quantitative, and Diagnostic Reasoning sections. Also, added Derivations and Technical References chapter, with a section on equivalent series and parallel XR networks.

October 2018 – minor edits to capitalization of "reductio ad absurdum".

October 2018 – document first created.

Index

Adding quantities to a qualitative problem, 96
Admittance, 23, 47
Ammeter, 18
Annotating diagrams, 3, 95

C++, 50
Checking for exceptions, 96
Checking your work, 96
Code, computer, 105
Compiler, C++, 50
Complex arithmetic, "by hand", 42
Complex number, 42
Computer programming, 49
Conductance, 23, 46
Conservation of Electric Charge, 17, 26, 27
Conservation of Energy, 17, 26, 27

Dimensional analysis, 95

Edwards, Tim, 106
Electrically common, 17
Equipotential points, 17

Fast Fourier Transform, 14
FFT, 14

Graph values to solve a problem, 96
Greenleaf, Cynthia, 65

How to teach with these modules, 103
Hwang, Andrew D., 107

Identify given data, 95
Identify relevant principles, 95
Intermediate results, 95
Interpreter, Python, 54
Inverted instruction, 103

Java, 51

KCL, 18
Kirchhoff's Current Law, 18, 23, 33
Kirchhoff's Voltage Law, 18, 23, 33
Knuth, Donald, 106

Lamport, Leslie, 106
Limiting cases, 96

Metacognition, 70
Mho, 23
Mho, unit, 46
Moolenaar, Bram, 105
Murphy, Lynn, 65

Ohm, 23
Ohm's Law, 23, 30–32
Open-source, 105

Parallel, 17, 26
Phasor, 22
Phasor diagram, 19, 22
Polar form, 42
Problem-solving: annotate diagrams, 3, 95
Problem-solving: check for exceptions, 96
Problem-solving: checking work, 96
Problem-solving: dimensional analysis, 95
Problem-solving: graph values, 96
Problem-solving: identify given data, 95
Problem-solving: identify relevant principles, 95
Problem-solving: interpret intermediate results, 95
Problem-solving: limiting cases, 96
Problem-solving: qualitative to quantitative, 96
Problem-solving: quantitative to qualitative, 96
Problem-solving: reductio ad absurdum, 96
Problem-solving: simplify the system, 95
Problem-solving: thought experiment, 95
Problem-solving: track units of measurement, 95

Problem-solving: visually represent the system, 95
Problem-solving: work in reverse, 96
Programming, computer, 49
Python, 54

Qualitatively approaching a quantitative problem, 96

Reading Apprenticeship, 65
Rectangular form, 42
Reductio ad absurdum, 26, 96, 102, 103

Schoenbach, Ruth, 65
Scientific method, 70
Series, 17, 26
Series-parallel, 17
Siemens, 23
Siemens, unit, 46, 47
Siemens, Werner von, 23
Simplifying a system, 95
Socrates, 102
Socratic dialogue, 103
Source code, 50
SPICE, 65
Stallman, Richard, 105
Susceptance, 47

Thought experiment, 95
Torvalds, Linus, 105

Unit phasor, 23
Units of measurement, 95

Visualizing a system, 95
Voltmeter, 18

Whitespace, C++, 50, 51
Whitespace, Python, 57
Work in reverse to solve a problem, 96
WYSIWYG, 105, 106