Modular Electronics Learning (ModEL) project

Capacitors and Capacitive Circuits

© 2018-2022 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License

Last update = 13 June 2022

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.
## 5.4 Modeling an energizing capacitor using C++

---

## 6 Questions

### 6.1 Conceptual reasoning
- Reading outline and reflections
- Foundational concepts
- Electric flux lines
- Leyden jar
- Water tank analogy
- Cable capacitance
- Capacitor as source versus load
- Audio power capacitors
- Capacitor charging circuit
- Analog electrical model of warming box

### 6.2 Quantitative reasoning
- Miscellaneous physical constants
- Introduction to spreadsheets
- Air capacitor gap
- Defibrillator capacitor
- Capacitor labeling
- Half-life
- Number of time constants
- Inverse exponential functions
- Current versus rate-of-voltage-change
- Capacitance of three-capacitor network
- Resistance for specified time constant
- Safely de-energizing a large capacitor
- Capacitor voltage at specified intervals
- Partial capacitor de-energization
- Rapid charge and slow discharge
- Time required to reach specified voltages
- Measuring capacitance by time delay
- Intervalometer
- SPICE analysis of an energizing capacitor
- SPICE analysis of a de-energizing capacitor

### 6.3 Diagnostic reasoning
- Dielectric breakdown
- Incorrect voltage/current curves

---

### A Problem-Solving Strategies

---

### B Instructional philosophy

---

### C Tools used

---

### D Creative Commons License
Chapter 1

Introduction

Capacitance is the ability to store energy in an electric field, caused by a difference on electric charge between two points separated by an insulating gap. Any device constructed to exhibit this property is called a capacitor. The amount of capacitance offered by any capacitor is directly proportional to the overlapping areas of the conductive surfaces (i.e. “plates”), directly proportional to the permittivity of the insulating dielectric material separating those plates, and inversely proportional to the distance of their separation. The amount of energy stored in any capacitor is directly proportional to the capacitance (measured in Farads) and directly proportional to the square of the voltage as described by the formula \( E = \frac{1}{2}CV^2 \).

Important concepts related to capacitance include Conservation of Energy, electric fields, permittivity, dielectric, sources versus loads, “Ohm’s Law” for capacitors, rates of change (e.g. “speed”), natural decay functions, time constant \( (\tau) \), capacitor ratings, properties of series networks, properties of parallel networks, and energy storage.

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

- What is a field, and how might we measure its presence?
- How may we construct a capacitor?
- What factors influence the amount of capacitance offered by a capacitor?
- Is the amount of energy stored within a capacitor a function of its voltage, its current, or both?
- Which parameter is able to vary most rapidly for a capacitor, voltage or current?
- What is the mathematical relationship between voltage and current for a capacitor?
- What are some practical applications for capacitors?
- Why do voltage and current tend to rise or “decay” in inverse-exponential fashion in a resistor-capacitor circuit?
- What does time constant mean?
• What factors influence the maximum voltage or current ratings for a capacitor?
• Why does capacitance add in parallel and diminish in series?
• What “parasitic” properties are found in real capacitors other than capacitance?
• Why are some capacitors polarized and others are not?

A useful resource for understanding inverse-exponential growth and decay as it relates to a resistor-capacitor network is found in the Programming References chapter. There you will find example code showing how these changes in voltage and current occur over time.
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: capacitor storing and releasing energy

It is easy to use a battery, capacitor\(^1\), resistor, and LED to demonstrate the principle of capacitance storing and releasing energy:

In the upper configuration the capacitor acts as a load. In the lower configuration the capacitor acts as a source.

\(^1\)An aluminum electrolytic capacitor having a value of at least 100 \(\mu\)F and a voltage rating greater than the voltage of the battery generally works well. Beware the polarity requirements of the capacitor, as electrolytic capacitors are polarized devices and will be damaged by applying a reverse-polarity DC voltage!
2.1. EXAMPLE: CAPACITOR STORING AND RELEASING ENERGY

A recommended variation of this experiment is to try using multiple capacitors of the same rating as the original, energized by the same source (battery) voltage as well. The point of these experiments is to compare how long the LED remains energized when powered by the series capacitor network, versus when powered by the parallel capacitor network, versus when powered by single capacitor, total voltage being equal in all cases:

**Series capacitors**

**Parallel capacitors**

Since capacitance is defined as the capacity to store energy in the form of an electric field, the comparative results of these experiments will prove whether capacitance is enhanced in parallel or in series. The configuration resulting in the LED remaining energized longest is the one having the most capacitance.

An important principle in scientific experimentation is to control for all variables in an experiment except the one variable we wish to explore. This is why it is important here to use identically-valued capacitors in all cases, and use the same amount of source voltage too. This way, the only variable is the total amount of capacitance resulting from the number of capacitors and the manner in which they interconnect with each other.
2.2 Example: series versus parallel capacitors

Calculating energy stored within a series network of 10 μF capacitors based on the number of capacitors \((n)\) connected in series, in each case energized by a 20 Volt source. The amount of energy stored within each capacitor is given by the following formula, and the total energy stored is the simple sum of all capacitors’ stored energy:

\[
E_C = \frac{1}{2}CV^2
\]

Where,
- \(E_C\) = capacitor energy in Joules (J)
- \(C\) = capacitance in Farads (F)
- \(V\) = voltage in Volts (V)

Connecting more capacitors in series results in the source voltage (20 Volts) being divided into smaller portions. With less voltage across each capacitor, there is dramatically less energy stored within each owing to the squared voltage term in the \(E_C = \frac{1}{2}CV^2\) equation, so that with more capacitors in the series network there is actually less total energy stored for the entire network.

Since capacitance is defined as the ability to store energy within an electric field, we must conclude that connecting capacitors in series results in less over-all capacitance.
Calculating energy stored within a parallel network of 10 $\mu$F capacitors based on the number of capacitors ($n$) connected in parallel, in each case energized by a 20 Volt source. The amount of energy stored within each capacitor is given by the following formula, and the total energy stored is the simple sum of all capacitors’ stored energy:

$$E_C = \frac{1}{2}CV^2$$

Where,
- $E_C$ = capacitor energy in Joules (J)
- $C$ = capacitance in Farads (F)
- $V$ = voltage in Volts (V)

<table>
<thead>
<tr>
<th>Number of capacitors ($n$)</th>
<th>Voltage across each capacitor</th>
<th>Energy stored in each capacitor</th>
<th>Energy stored (total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20.00 V</td>
<td>2.0000 mJ</td>
<td>2.0000 mJ</td>
</tr>
<tr>
<td>2</td>
<td>20.00 V</td>
<td>2.0000 mJ</td>
<td>4.0000 mJ</td>
</tr>
<tr>
<td>3</td>
<td>20.00 V</td>
<td>2.0000 mJ</td>
<td>6.0000 mJ</td>
</tr>
<tr>
<td>4</td>
<td>20.00 V</td>
<td>2.0000 mJ</td>
<td>8.0000 mJ</td>
</tr>
</tbody>
</table>

Connecting more capacitors in parallel results in a direct increase in stored energy, because each capacitor receives the same source voltage (20 Volts) and therefore stores the same amount of energy.

Since capacitance is defined as the ability to store energy within an electric field, we must conclude that connecting capacitors in parallel results in more over-all capacitance.
2.3 Example: RC demonstration circuit

The following circuit is useful for demonstrating the inverse-exponential behavior of simple resistor-capacitor (RC) networks:

![Diagram of RC circuit]

Pressing the upper pushbutton causes the capacitor to “charge up” to a higher voltage level; pressing the lower pushbutton causes it to “discharge” to a lower voltage level. The time constant \( \tau \) is equal to \( RC \) for which ever resistor is in effect at the time, meaning you may design the circuit to charge and discharge at different rates if you wish.

For best results using a multimeter to measure capacitor voltage and/or current, select resistors between 1 kΩ and 100 kΩ in value, and choose a capacitor with a large enough value to make the time constant reasonable for manual measurements (e.g. a \( \tau \) value of at least 1 second).

Electrolytic capacitors are the most practical choice for the amount of capacitance you will likely require in order to achieve a relatively slow \( \tau \) value, so beware that these capacitors are polarized which means any reversal of voltage polarity will likely destroy it!

An interesting phenomenon you will likely observe when using an electrolytic capacitor in this circuit is dielectric relaxation, also known as “battery action” whereby the capacitor’s voltage rebounds a small amount after being discharged.
2.4 Example: battery voltage stabilization

Electrochemical batteries are excellent sources of electrical energy for small-scale electronic projects. They are safer than utility-energized “power supply” units, portable, and readily available. However, they do suffer some limitations, one of them being internal resistance that increases over time as the battery ages. If a battery with high internal resistance is used to energize a load with a pulsing current demand (e.g. where the load is switched on and off), the battery’s terminal voltage will rise and fall accordingly:

This variation in battery terminal voltage over time may become problematic if more than one load is energized by the same battery, as a continuously-energizing load will “see” the voltage variations created by the pulsing load:
One way to help stabilize a battery’s terminal voltage is to connect a large capacitor in parallel with the battery. When terminal voltage is steady, the capacitor’s current will be zero. However, if something acts to disturb that terminal voltage the capacitor will pass current in direct proportion to the rate of voltage change \( I_C = C \frac{dV}{dt} \). In essence, the capacitor acts as a temporary voltage source to “shore up” the battery for brief periods of time while a pulsing load draws current:

\[ C = \frac{I}{\frac{dV}{dt}} = \frac{0.5}{0.25 \frac{0.002}{0.002}} = 4000 \mu F \]

Since over-sizing the capacitance will result in a more stable voltage and is therefore preferable to under-sizing, it would make sense to select for this application the common capacitor size of 4700 \( \mu F \).

Capacitors are often connected in parallel to the DC power terminals of high-power audio amplifiers used in automotive sound systems, for this very purpose: to give the bass amplifier a more stable voltage during periods of time when it draws very high current from the vehicle’s battery to deliver transient pulses of sound to the speakers. In such applications, where the current demands are quite high, “banks” of parallel-connected capacitors offering multiple Farads of total capacitance are typical.
2.4. EXAMPLE: BATTERY VOLTAGE STABILIZATION

When battery-powering electronic circuits for educational purposes, I recommend a power supply consisting of two alkaline-cell batteries, two capacitors, and two fuses. This provides a “split” power supply with voltage stabilization and overcurrent protection:

Battery-based “split” DC power supply

9-Volt “transistor” batteries are good for this purpose, as are 6-Volt “lantern” batteries. The two capacitors should be several thousand microFarads each, and may consist of multiple paralleled capacitors to achieve that amount of capacitance. Fuses should be sized to blow at a current value less than what the battery is able to deliver to a short-circuit (I recommend $\frac{1}{4}$ Ampere AGC-type glass-cartridge fuses for general-purpose use).
2.5 Example: decoupling capacitors

Many forms of electronic circuits are especially sensitive to variations in DC power supply voltage resulting from pulsations of current drawn from the supply over time, the DC terminal voltage tending to “sag” or “droop” as current surges. A highly effective safeguard against this problem is to attach a capacitor as closely as possible to the power supply terminals of the affected circuit, the capacitor serving as a reservoir of electrical energy to temper these transient voltage events.

The nature of the problem may be seen in the following schematic diagram, where we have three separate digital logic “gates” all powered by a common DC voltage source. The wires or printed circuit board traces carrying DC power to these three loads inherently possess both resistance (\(R\)) and inductance (\(L\)), both of which cause sudden voltage drops when current suddenly increases:

\[
\text{Resistance drops voltage proportional to current, which is simply Ohm’s Law in action (} V = IR). \text{ Inductance drops voltage in response to a rise in current over time (} V = L \frac{dI}{dt})\text{, the voltage drop’s magnitude being proportional to the speed at which current increases.}
\]

\[
\text{Resistance drops voltage proportional to current, which is simply Ohm’s Law in action (} V = IR). \text{ Inductance drops voltage in response to a rise in current over time (} V = L \frac{dI}{dt})\text{, the voltage drop’s magnitude being proportional to the speed at which current increases.}
\]

\[
\text{Resistance drops voltage proportional to current, which is simply Ohm’s Law in action (} V = IR). \text{ Inductance drops voltage in response to a rise in current over time (} V = L \frac{dI}{dt})\text{, the voltage drop’s magnitude being proportional to the speed at which current increases.}
\]

If we imagine the far-right logic gate suddenly drawing current because its output switches state, all the parasitic \(R\) and \(L\) elements between it and the DC voltage source will manifest additional voltage drop\(^2\), resulting in all three logic gates experiencing a momentary decrease in supply voltage if not for the stabilizing influence of the capacitors connected in parallel to each gate’s power supply terminals.

The tendency for DC supply voltage to suddenly dip in a circuit due to effects such as this is sometimes referred to as \textit{ground bounce} or \(V_{DD}\) \textit{bounce}\(^3\). The practice of connecting capacitors in parallel with the integrated circuits (ICs) is called \textit{decoupling}, because it helps prevent the switching action of one load from interfering or “coupling” to another load. Such “bounce” may severely degrade a circuit’s performance, causing noise problems in analog circuits and all manner of problems in digital circuits (e.g. erratic signal timing, false counts).

Decoupling is a good practice in any system where we anticipate rapid on/off switching of the components, and it is best to use capacitors with low equivalent series resistance (ESR) so they will be able to source high levels of transient current if necessary. For very high-frequency printed circuit boards a good practice for DC power distribution is to devote two or more of the layers in a multi-layer board as \textit{power planes}: sheets of copper sandwiched between layers of insulating board material acting as “busses” for each rail of the DC power source. Not only does this make DC power easy for each device to access (simply install a “via” which is an inter-layer connector between the desired power plane layer and the device power terminal) but the natural capacitance existing between power planes separated by board insulation contributes to the decoupling offered by discrete capacitors installed adjacent to each device.

\(^2\)Resistance drops voltage proportional to current, which is simply Ohm’s Law in action (\(V = IR\)). Inductance drops voltage in response to a rise in current over time (\(V = L \frac{dI}{dt}\)), the voltage drop’s magnitude being proportional to the speed at which current increases.

\(^3\)In CMOS digital circuits the positive pole of the DC voltage source is commonly referred to as \(V_{DD}\) in honor of its connection to the Drain terminals of many transistors within the IC.
2.5. EXAMPLE: DECOUPLING CAPACITORS

In the following photograph you can see two holes in a printed circuit board ready to accept a small capacitor ($C_4$) immediately above the 20-pin integrated circuit. When soldered into this incomplete PCB, $C_4$ will stabilize the DC power supply voltage to that IC so that it experiences less “bounce” on its power supply terminals during switching events:

Note how close to the IC the decoupling capacitor $C_4$ will be located. The rationale for positioning the decoupling capacitor immediately adjacent to the protected IC is to minimize the amount of length for any copper traces connecting it to the IC’s power supply pins, because greater conductor length means more resistance and more inductance, all other factors being equal. Recall how it is the presence of $R$ and $L$ between the load and DC power source that aggravates the “bounce” problem, and so we need our decoupling capacitor to have as little impedance as possible between it and the load whose DC voltage we intend to stabilize.

A clever way of incorporating decoupling capacitors into circuits is found in the following photograph of a 14-pin DIP socket. Here the decoupling capacitor is built-in, connected between pins 7 and 14 which are commonly the negative and positive DC power pins for a great many TTL and CMOS logic ICs:
As a general rule, the best decoupling capacitor types are ceramic, mica, and polystyrene due to their naturally low equivalent series resistance (ESR). Tantalum electrolytic capacitors, although blessed with relatively high capacitance in small packages, have fairly high ESR which limits their effectiveness as decoupling capacitors.
2.6 Example: voltage source energizing a capacitor

Switch closure time | \( V_C \) | \( V_R \) | \( I \) |
--- | --- | --- | --- |
-1 seconds | 0 V | 0 V | 0 mA |
0 seconds | 0 V | 16 V | 1.0667 mA |
1 second | 3.501 V | 12.499 V | 0.8333 mA |
2 seconds | 6.235 V | 9.765 V | 0.6510 mA |
3 seconds | 8.372 V | 7.628 V | 0.5085 mA |
4 seconds | 10.041 V | 5.959 V | 0.3973 mA |
5 seconds | 11.345 V | 4.655 V | 0.3104 mA |
6 seconds | 12.363 V | 3.637 V | 0.2425 mA |
7 seconds | 13.159 V | 2.841 V | 0.1894 mA |
8 seconds | 13.781 V | 2.219 V | 0.1480 mA |
\( \infty \) | 16 V | 0 V | 0 mA |
2.7 Example: current source energizing a capacitor

\[ I_C = \frac{5 \text{ mA}}{12 \text{k} \Omega} = 0.4166 \text{ mA} \]

\[ I_R = \frac{0 \text{ mA}}{330 \mu \text{F}} = 0 \text{ mA} \]

\[ V = 0 \text{ V} \]

<table>
<thead>
<tr>
<th>Switch open time</th>
<th>( I_C )</th>
<th>( I_R )</th>
<th>( V )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 seconds</td>
<td>0 mA</td>
<td>0 mA</td>
<td>0 V</td>
</tr>
<tr>
<td>0 seconds</td>
<td>5 mA</td>
<td>0 mA</td>
<td>0 V</td>
</tr>
<tr>
<td>1 second</td>
<td>3.8842 mA</td>
<td>1.1158 mA</td>
<td>13.39 V</td>
</tr>
<tr>
<td>2 seconds</td>
<td>3.0174 mA</td>
<td>1.9826 mA</td>
<td>23.79 V</td>
</tr>
<tr>
<td>3 seconds</td>
<td>2.3440 mA</td>
<td>2.6560 mA</td>
<td>31.88 V</td>
</tr>
<tr>
<td>4 seconds</td>
<td>1.8209 mA</td>
<td>3.1791 mA</td>
<td>38.15 V</td>
</tr>
<tr>
<td>5 seconds</td>
<td>1.4146 mA</td>
<td>3.5854 mA</td>
<td>43.03 V</td>
</tr>
<tr>
<td>6 seconds</td>
<td>1.0989 mA</td>
<td>3.9011 mA</td>
<td>46.81 V</td>
</tr>
<tr>
<td>7 seconds</td>
<td>0.8536 mA</td>
<td>4.1464 mA</td>
<td>49.76 V</td>
</tr>
<tr>
<td>8 seconds</td>
<td>0.6631 mA</td>
<td>4.3369 mA</td>
<td>52.04 V</td>
</tr>
<tr>
<td>( \infty )</td>
<td>0 mA</td>
<td>5 mA</td>
<td>60 V</td>
</tr>
</tbody>
</table>
2.8 Example: voltage divider and capacitor

This circuit is more challenging to analyze, given the presence of two resistors rather than just one:

![Circuit Diagram]

First, we close the switch and determine capacitor voltage and capacitor current at 30 millisecond intervals, then after letting the capacitor reach its full state of energization we open the switch and let it de-energize again:

**Switch closes!**

<table>
<thead>
<tr>
<th>Time (msec)</th>
<th>V_C (V)</th>
<th>I_C (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>2.5655</td>
<td>3.17604</td>
</tr>
<tr>
<td>60</td>
<td>4.19513</td>
<td>2.01745</td>
</tr>
<tr>
<td>90</td>
<td>5.23028</td>
<td>1.2815</td>
</tr>
<tr>
<td>120</td>
<td>5.88782</td>
<td>0.814022</td>
</tr>
<tr>
<td>150</td>
<td>6.30549</td>
<td>0.517074</td>
</tr>
<tr>
<td>180</td>
<td>6.5708</td>
<td>0.32845</td>
</tr>
<tr>
<td>210</td>
<td>6.73933</td>
<td>0.208634</td>
</tr>
<tr>
<td>240</td>
<td>6.84638</td>
<td>0.132526</td>
</tr>
<tr>
<td>270</td>
<td>6.91438</td>
<td>0.084182</td>
</tr>
<tr>
<td>300</td>
<td>6.95757</td>
<td>0.0534731</td>
</tr>
</tbody>
</table>

**Switch opens!**

<table>
<thead>
<tr>
<th>Time (msec)</th>
<th>V_C (V)</th>
<th>I_C (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.03279</td>
<td>1.80328</td>
</tr>
<tr>
<td>30</td>
<td>5.97101</td>
<td>1.53103</td>
</tr>
<tr>
<td>60</td>
<td>5.06954</td>
<td>1.29988</td>
</tr>
<tr>
<td>90</td>
<td>4.30417</td>
<td>1.10363</td>
</tr>
<tr>
<td>120</td>
<td>3.65435</td>
<td>0.937013</td>
</tr>
<tr>
<td>150</td>
<td>3.10264</td>
<td>0.795548</td>
</tr>
<tr>
<td>180</td>
<td>2.63422</td>
<td>0.675441</td>
</tr>
<tr>
<td>210</td>
<td>2.23652</td>
<td>0.573466</td>
</tr>
<tr>
<td>240</td>
<td>1.89886</td>
<td>0.486887</td>
</tr>
<tr>
<td>270</td>
<td>1.61218</td>
<td>0.41338</td>
</tr>
<tr>
<td>300</td>
<td>1.36878</td>
<td>0.35097</td>
</tr>
</tbody>
</table>
Chapter 3

Tutorial

3.1 Electric fields and energy storage

An electric field is an invisible web of interaction between electrically charged matter, existing whenever opposite charges are separated from each other over space. Any electrically charged matter within an electric field will experience a force acting upon it from that field, similar to how any mass within a gravitational field experiences a force we call weight:

An electric charge experiences a force when exposed to an electric field, just as a mass experiences force (called "weight") when exposed to a gravitational field.

Just as any weight suspended in a location where it could fall given the proper conditions possesses potential energy, opposing electric charge carriers separated by an insulating medium also possess electrical potential energy. This is the basis of capacitance: the ability to store energy in an electric field. The amount of energy stored by any capacitance is a function of voltage: all other factors being equal, more voltage existing between two surfaces means more energy stored in the electric field.
Electrical components called **capacitors**\(^1\) exploit this phenomenon, constructed of two conductive surfaces separated by a layer of electrical insulation called a **dielectric**. The amount of capacitance exhibited by any capacitor is directly proportional to the overlapping area of the conductive surfaces (in square meters), directly proportional to the dielectric permittivity\(^2\) (\(\varepsilon\)) of the dielectric material, and inversely proportional to the separation distance between plates (in meters):

\[
C = \frac{\varepsilon A}{d}
\]

### Table: Comparison of Capacitance and Dielectric Materials

<table>
<thead>
<tr>
<th>Area</th>
<th>Less C</th>
<th>More C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permittivity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Energy stored in any capacitance is a function of the capacitance and the amount of voltage impressed across it as described below:

\[
E_C = \frac{1}{2}CV^2
\]

Where,

- \(E_C\) = energy in Joules (J)
- \(C\) = capacitance in Farads (F)
- \(V\) = voltage in Volts (V)

Note the similarities between this equation and those describing potential energy stored in a mechanical spring (\(E_p = \frac{1}{2}kx^2\)) and kinetic energy stored in a moving mass (\(E_k = \frac{1}{2}mv^2\)). Just as the amount of potential energy stored in a spring varies with the square of the spring’s displacement \(x\) and kinetic energy varies with the square of the mass’s velocity \(v\), energy stored in a capacitor varies with the square of the voltage across it.

---

\(^1\) An obsolete term for capacitor is **condensor** or **condenser**.

\(^2\) Permittivity, simply defined, is a measure of how effective a dielectric substance is at electrostatically storing energy for any given electric field strength. It may be measured in absolute terms, expressed in units of Farads per meter, or it may be expressed as a ratio relative to the permittivity of a perfect vacuum.
One way to make a *variable* capacitor is to arrange the plates in such a way that their overlap area may be easily adjusted. In the example below we see a variable capacitor with a set of stationary plates and a set of movable plates rotated by a shaft. From left to right and top to bottom, we see a sequence beginning with maximum overlap (i.e. maximum capacitance) and progressing to less and less overlap (i.e. less capacitance):

This variable capacitor is actually *two* capacitors in one, sharing a common terminal. Its schematic diagram looks like this:

![Schematic diagram of a variable capacitor](image)

The two individual capacitor terminals appear on the left-hand side of the device shown in the photos, while the common terminal is below the rotary shaft.
Energy is always conserved, which means it must always be accounted for in any system and cannot simply appear or vanish. Another way of stating this same fact is to say that the only way for the amount of stored energy in any system to vary is if energy transfers to or from some other place\(^3\). Since energy is a function of voltage for any capacitance, the amount of voltage across a fixed capacitance’s terminals cannot vary unless energy is added to or removed from that capacitance. That energy transfer occurs when charge carriers enter or exit a capacitor. In other words, given that energy is always conserved and that stored energy within a capacitor manifests as voltage, it must be voltage that is conserved while current changes direction based on the accumulation or release of energy.

Thus, a capacitance maintains a steady store of energy when its voltage is constant and its current\(^4\) is zero. An increasing capacitor voltage means its stored energy also increases, which in turn means it must act as a load as it absorbs energy from some external source. Conversely, a decreasing capacitor voltage means its stored energy also decreases, which in turn means it must act as a source as it releases energy to some external load.

\(^3\)A close analogy based on the Conservation of Mass is that of a container holding water. Since water has mass, the only way for the amount of water in that container to vary is if water leaves it or enters it. A container’s water mass cannot simply vary without any cause, because mass is a conserved quantity.

\(^4\)At first it may seem as though current should be impossible because a capacitor is an open circuit. However, recall that open circuits do not prohibit current; they only prohibit continuous current.
3.2 Voltage, current, and capacitance

If we connect a DC current source to a capacitor, we find that the voltage across the capacitor rises at a steady rate determined by the amount of capacitance and the amount of current:

\[ I = C \frac{dV}{dt} \]

Where,
\( I \) = current in Amperes (A)
\( C \) = capacitance in Farads (F)
\( \frac{dV}{dt} \) = rate-of-change of voltage in Volts per second (V/s)

This may be thought of as “Ohm’s Law” for capacitance. Unlike a resistor where voltage and current are always in direct and immediate proportion to each other, the speed of a capacitor’s voltage is directly proportional to its current. The presence of time in the characteristic equation makes capacitance behave quite different from resistance, and any circuit containing a substantial amount of capacitance must be analyzed with time as one of its variables.
If we were to repeat this experiment on the same size capacitor, but using a current source with half as much current, we would find the voltage still rises linearly over time but at half the rate:

De-activating the current source at some point in time prior to the end of the experiment results in the capacitor holding its store of energy and maintaining a constant voltage:
3.3 Applications of capacitance

Capacitors find many applications in DC circuits. Perhaps the most obvious use of capacitance is to store energy in a manner similar to that of a secondary-cell (“rechargeable”) battery. Connecting a capacitor across the terminals of a DC voltage source will cause that capacitor to energize and absorb energy from that source. Subsequently disconnecting that capacitor from the source results in the capacitor holding that stored energy, and then when the capacitor is connected to a load it delivers that stored energy to the load. The following photograph shows an array of six so-called supercapacitors\(^5\) ganged together to serve as a replacement for the lead-acid storage battery in a passenger vehicle:

![Supercapacitors](image)

Another use of capacitors is for voltage stabilization, exploiting the principle that a capacitance’s voltage cannot vary unless energy transfer occurs. A large-valued capacitor connected in parallel with some other component receiving a DC voltage will force that voltage to be more stable than it would be otherwise, as the capacitor will act as a load when voltage rises and will act as a source when voltage falls. This is analogous to a flywheel\(^6\) acting to stabilize the rotating speed of a machine by serving as a mechanical load as speed increases and serving as a mechanical source as speed decreases. An examination of integrated circuit (IC) “chips” on a printed circuit board (PCB) typically reveals a capacitor connected in parallel with the power supply terminals on each IC for the purpose of mitigating voltage fluctuations that might make the IC misbehave. The following photograph shows two such capacitors, each soldered between the Vdd and Gnd terminals of a microcontroller PCB:

![Microcontroller PCB](image)

---

\(^5\)This term is well-deserved, and refers to the unusually high capacitance exhibited by each capacitor. One Farad is a rather large amount of capacitance by any standard, and each one of the supercapacitors shown is rated at 650 Farads.

\(^6\)A “flywheel” is typically a heavy wheel coupled to a rotating shaft, the mass of that wheel serving to store kinetic energy as it spins.
Yet another use of capacitors is creating time-delayed voltages and currents. Consider this example circuit showing a capacitor connected to a DC voltage source through a single-pole double-throw (SPDT) switch and a resistor:

![Circuit Diagram](image)

We know capacitors store energy by means of electric fields, and that the strength of a capacitor’s electric field is directly related to voltage between its plates. If we assume the capacitor begins in a fully de-energized state, its voltage therefore must be zero when the switch initially moves to the “energize” position. With zero $V_C$, Kirchhoff’s Voltage Law demands the resistor drop the full source voltage, and from Ohm’s Law ($I = \frac{V}{R}$) we can tell current must start at its maximum value. A high current value results in a high rate-of-change for capacitor voltage ($I = C\frac{dV}{dt}$), and so $V_C$ increases rapidly. As $V_C$ increases, less of the source’s voltage will be dropped across the resistor in accordance with Kirchhoff’s Voltage Law, and by Ohm’s Law this means less current in the circuit. Less current results in a slower rate-of-rise for voltage ($I = C\frac{dV}{dt}$), and so $V_C$ increases not linearly but rather it rises at a slower and slower pace until it finally levels off at the source voltage value. As $V_C$ rises, $V_R$ falls, and so current must fall as well in accordance with Ohm’s Law, eventually settling at zero but at a slower and slower speed over time.

Moving the SPDT switch to the “de-energize” position causes the capacitor to behave as a source now rather than as a load, delivering its stored energy to the resistor. As the capacitor releases energy its electric field weakens and its voltage decreases, causing current to decrease as well (Ohm’s Law, $I = \frac{V}{R}$), resulting in a slower and slower descent for both voltage and current over time.
3.4 Inverse exponential growth/decay

As with all natural decay functions, the decaying voltage is given by $V_{\text{initial}} \left( e^{-\frac{t}{\tau}} \right)$ and the decaying current by $I_{\text{initial}} \left( e^{-\frac{t}{\tau}} \right)$, where $t$ is time and $\tau$ is the time constant of the system, calculated for any simple resistor-capacitor circuit as $\tau = RC$. According to this formula, Ohms of resistance multiplied by Farads of capacitance equals the time constant of the RC network in seconds. All variables will settle to within 1% of their final values by $5\tau$ of time following the initial disturbance.

Exponential functions mathematically describe the decay and growth of quantities in any circuit characterized by a time constant ($\tau$). Any quantity decaying over time does so in proportion to $e^{-\frac{t}{\tau}}$ because the value of that exponential function decreases from 1 to 0 as time ($t$) increases from 0 to infinity. Any quantity increasing over time does so in complementary fashion, proportionate to $1 - e^{-\frac{t}{\tau}}$ because the value of that function increases from 0 to 1 as time ($t$) increases from 0 to infinity:

It is worth noting that these exponential formulae apply to more than just electric circuits. For example, the temperature of an object equalizing with its surroundings follows these same curves: an object hotter than its environment will cool down to equal ambient temperature following the $e^{-\frac{t}{\tau}}$ function; an object colder than its environment will warm up to ambient following the $1 - e^{-\frac{t}{\tau}}$ function. Other examples include the pressure of a compressed gas leaking out of an enclosed vessel (decaying over time to eventually settle at zero), and the speed of a boat following a sudden increase in engine thrust (growing over time to eventually settle at a new cruising speed). In fact, the behavior of voltage and/or current in these types of electric circuits so closely mimics these other phenomena that for many years electric circuits were used as analogues of other physical systems to electrically simulate complex behaviors – those circuits being called analog computers.

A recommended exercise for the reader is to evaluate these mathematical functions for different values of $t$ to fully understand why one of them describes a decaying function and the other a growing function. Start by calculating the value of each as time ($t$) begins at a value of zero: the decaying function $e^{-\frac{t}{\tau}}$ reduces to $e^0$ which is equal to 1, while the growing function reduces to $1 - e^0$ which is 0. Those values describe the “initial” condition of the voltage or current in relation to its minimum and maximum values. As time progresses and $t$ assumes a positive value, we see that $e^{-\frac{t}{\tau}}$ becomes smaller and smaller (approaching zero as $t$ approaches infinity) which of course means that $1 - e^{-\frac{t}{\tau}}$ must become closer and closer to a value of one.
3.5 Capacitor types and limitations

Capacitors are broadly categorized by the type of dielectric material used. Air, polyester plastic, ceramic, mica, and glass are among the different materials employed as dielectric media between capacitor plates. These dielectric materials vary in permittivity (i.e. the degree to which they enhance capacitance) and dielectric strength (i.e. the amount of electric field strength they can tolerate without “breaking down” and conducting current). As shown in the following equation, permittivity directly relates to capacitance for any given plate area and separation distance:

\[ C = \frac{A\epsilon}{d} \]

Where,
- \( C \) = Capacitance in Farads (F)
- \( A \) = Overlapping area of plates in square meters (m\(^2\))
- \( \epsilon \) = Absolute\(^7\) permittivity of dielectric material in Farads per meter (F/m)
- \( d \) = Distance of separation between plates in meters (m)

One class of dielectric deserving of special attention is electrolytic, referring to the way in which this capacitor is constructed. Recall that for a given plate area and permittivity, the amount of capacitance is inversely proportional to the distance separating the capacitor plates. This means we must make the dielectric layer extremely thin in order to achieve high capacitance with reasonable plate area. One way to create a super-thin dielectric layer is to anodize a metal surface which forms an electrically insulating (i.e. dielectric) metal-oxide film on the metal’s surface. This thin layer of metal-oxide serves as the dielectric, while an electrolyte paste or fluid serves as a conductive extension of the far plate up to the oxide’s surface. Most electrolytic capacitors are considered polarized because applying sufficient DC voltage in the wrong polarity will remove this anodized metal-oxide layer and cause the capacitor to fail in a shorted manner. Polarized capacitors are always marked as to which terminal is the positive, and may be symbolized in schematic diagrams using a curved line rather than a straight line as the negative plate:

\(^7\)Material permittivity is often specified as relative to a vacuum, in which case \( \epsilon_r \) is equal to 1 for a vacuum and greater than 1 for all other substances. The permittivity of free space (\( \epsilon_0 \)) is approximately \( 8.8541878128 \times 10^{-12} \) Farads per meter.) must be multiplied by any given relative permittivity value in order to arrive at the absolute permittivity of the substance in question.
The most elementary capacitor ratings are capacitance and maximum voltage. Capacitance is typically expressed either in Farads (F), microFarads (µF), nanoFarads (nF), or picoFarads (pF). Interestingly, one never encounters a capacitor rated in milliFarads, although there is nothing mathematically improper with the use of that metric prefix. Also, it is common to see the capital letter M used to represent micro, although that it technically improper\(^8\) as a metric prefix. Typically both capacitance and maximum voltage ratings will be found printed in text on the body of the capacitor itself. Capacitance expressed with a decimal point (e.g. 2.2) but no other markings refers to microFarads by default. Capacitance expressed without a decimal point (e.g. 33) and no other marking refers to picoFarads by default. A three-digit number with no decimal point (e.g. 473) and no other markings is read in a manner similar to the first three bands of a four-band resistor color code: the three numerals representing Digit, Digit, and Multiplier of picoFarads. For example a “473” capacitor is \(47 \times 10^3\) picoFarads, or 0.047 µF. Maximum voltage ratings are always followed by the letter “V”. Other alphabetical letters are used to designate tolerance (e.g. D = ±0.5%, F = ±1%, G = ±2%, H = ±3%, J = ±5%, K = ±10%, M = ±20%, etc.).

Other capacitor ratings include leakage resistance which is the resistance of the dielectric in Ohms (ideally infinite) and equivalent series resistance or ESR which is the effect of all dissipative losses within the capacitor modeled as a single resistance value. The wire leads of a capacitor also harbor parasitic inductance\(^9\) along their lengths.

For any given type of dielectric, the physical size of a capacitor is proportional to its capacitance as well as to its voltage rating. A capacitor having more capacitance but the same voltage rating must be made with more plate area and the same dielectric thickness, and this makes it larger. A capacitor having a higher voltage rating but the same capacitance must be made with thicker dielectric (which by itself reduces capacitance), and then also more plate area to make up for the degradation in capacitance caused by the thicker dielectric – this again makes it larger.

---

\(^8\)A capital “M” of course should mean mega \((1 \times 10^6)\), but this is never the case for capacitance!

\(^9\)Inductance is the ability to store energy in a magnetic field surrounding a conductor. A “parasitic” property is one that exists as an artifact of a component’s construction, and is not necessarily desired. For a capacitor, any stray inductance is considered parasitic, as is any resistance other than the (ideally) infinite resistance of the dielectric.
3.6 Series and parallel capacitance

Like resistors, capacitors may be connected together to form networks with different total capacitance than any single capacitor. Unlike resistors, though, connecting capacitors in series diminishes total capacitance, while connecting capacitors in parallel accumulates total capacitance:

\[ C_{\text{series}} = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}} \]

\[ C_{\text{parallel}} = C_1 + C_2 + C_3 \]

This may be understood in terms of geometry: stacking capacitors in series with each other is equivalent to making one capacitor with a wider gap between its plates (thus diminishing capacitance), while ganging capacitors in parallel with each other is equivalent to making one capacitor with larger plate area (thus increasing capacitance).

This may also be understood in terms of energy storage, and proven with a few simple thought experiments. Imagine a 3 Farad capacitor energized with 2 Volts between its plates, storing 6 Joules of energy as predicted by \( E_C = \frac{1}{2}CV^2 \). Now imagine three of these same capacitors connected in parallel so all of them share the same 2 Volts. Total stored energy must be the sum of the three individual capacitors’ energies and therefore is 18 Joules, three times that of the single capacitor. Now imagine three of these same capacitors connected in series, dividing the 2 Volt total voltage into thirds. Each 3-Farad capacitor dropping \( \frac{2}{3} \) Volt will store only \( \frac{2}{3} \) Joule of energy with the three together storing only 2 Joules of energy in total, just one third that of the single capacitor.
Chapter 4

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.
CHAPTER 4. DERIVATIONS AND TECHNICAL REFERENCES

4.1 IEC standard component values

Components such as resistors and capacitors are manufactured in several standard values, described by IEC standard 60063. Rather than having a single series of standard values, the IEC publishes lists called $E$ series based on the number of unique values spanning a single decade (i.e. a 10:1 range).

The shortest of these series, called $E_3$ contains just three values: 10, 22, and 47. The next series is called $E_6$ with six unique values: 10, 15, 22, 33, 47, and 68. These values represent significant values for components, meaning the decimal point may be freely moved to create values spanning multiple decades. For example, “33” simply means one can expect to find components manufactured in values of 33, 3.3, 0.33, and 0.033 as well as 330, 3.3 k, 33 k, etc.

Although this may seem like a strange standard for component manufacturers to follow, there is a compelling logic to it. The terms of each series are closer-spaced at the low end than at the high end, and this allows for series and/or parallel combinations of components to achieve most any desired value. For example, in the E6 series we only have values with the significant figures 10, 15, 22, 33, 47, and 68, but this doesn’t mean we are limited to total values with these significant figures. For example, if we needed 80 Ohms of resistance we could connect a 33 Ohm and 47 Ohm resistor together in series. 50 Ohms could be made from two 68 Ohm resistors in parallel (making 34 Ohms) plus a 15 Ohm and 1 Ohm resistor in series.

On the next page is a table showing the four most common E-series specified by IEC standard 60063.
4.1. **IEC STANDARD COMPONENT VALUES**

<table>
<thead>
<tr>
<th></th>
<th>E3</th>
<th>E6</th>
<th>E12</th>
<th>E24</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>18</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>27</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>39</td>
<td>39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>56</td>
<td>56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>68</td>
<td>68</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>82</td>
<td>82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*E48, E96, and E192 series are also found in the IEC 60063 standard, used for components with tighter tolerance ratings than typical.*
4.2 Capacitor labeling

The labeling of capacitor values can be confusing. Color codes used to be popular for denoting capacitance value, following a very similar pattern to resistors, but numerical labels are now the norm. However, these labels tend to be rather terse, and this is cause for confusion.

Physically large capacitors having sufficient surface area to allow many characters of printed text are the best in this regard.

The “supercapacitor” seen in the left-hand photograph is the clearest of them all: 650 Farads of capacitance with a working voltage rating of 2.7 Volts DC. The capacitor in the center photograph is also very unambiguous, having ratings of 3300 microFarads and 80 Volts DC (“WV” = working voltage). However, the capacitor seen in the right-hand photograph, despite being physically larger than either of the others, is more cryptic: 16500 microFarads of capacitance with a working voltage rating of 50 Volts DC. Here we must simply assume that the “16500” figure refers to microFarads.

Polarity is another important parameter for electrolytic-type capacitors such as these. For the supercapacitor (left photo) we see a stripe and letter “P” denoting the positive terminal of the capacitor; the 3300 microFarad capacitor (center photo) bears a black stripe and “−” symbol pointing toward its negative terminal; the 16500 microFarad capacitor (right photo) has polarity markings molded into the end-cap near its screw terminals (not visible in this photo).
4.2. CAPACITOR LABELING

Consider two more illustrative examples:

On the left we have a 2500 microFarad electrolytic capacitor rated for 50 Volts DC (positive on left), a capital letter “U” employed to symbolize micro. On the right we have a 2 microFarad plastic-film capacitor rated at 200 Volts DC, this time with the capital letter “M”\(^1\) representing micro. Since plastic-film capacitors are non-polarized, the stripe on the left-hand side must represent something other than polarity: here it represents the terminal connected to the outer-most layer of metal foil\(^2\) inside the capacitor.

Smaller capacitors become slightly more consistent, though no less confusing, in their labeling. Consider these examples:

From left to right we have (left) a 0.082 microFarad capacitor rated for 100 Volts DC\(^3\), (center) a 0.025 microFarad capacitor rated for 100 Volts DC, and (right) a 0.68 microFarad capacitor rated for 160 Volts DC. The rule here is, any numerical capacitance value with a decimal point should be read as microFarads. Note how the capacitor in the right-hand photograph uses a comma rather than a point between the “0” and the “68” because this one is of European manufacture, and it is conventional in European technical literature to use commas rather than points for decimal numbers.

---

\(^1\)In the metric system a lower-case “m” is supposed to represent the prefix milli \((\times 10^{-3})\) and an upper-case “M” is supposed to represent the prefix mega \((\times 10^6)\). However, since no technology yet invented is able to pack a megaFarad of capacitance into a single component, and for some unknown reason the prefix “milli” is never used for capacitance, we are expected to deduce that this letter “M” must represent micro. This, of course, flies in the face of standard metric notation, but it is nevertheless common to see in capacitor labeling.

\(^2\)This might be important to know for reasons of interference and signal coupling in densely-packed circuits. For a plastic-case capacitor such as this, the outer-most metal foil layer may comprise a parasitic capacitance with some adjacent component, and this in turn may degrade the circuit’s performance. Best practice is to connect the “stripe” terminal of such a capacitor to ground or a power supply “rail” or some other node having a stable electrical potential.

\(^3\)The second “0” of the number 100 happens to be partially rubbed off.
For capacitors with smaller capacitance values, the default metric prefix becomes \textit{pico} \((\times 10^{-12})\) rather than micro. We can tell the difference between a “micro” capacitor and a “pico” capacitor by the presence of lack, respectively, of a decimal point. For example, a capacitor labeled 2.2 would be 2.2 microFarads in size, but a capacitor labeled 22 would be 22 picoFarads in size.

For decimal values of picoFarads, a lower-case letter “p” may be used as a decimal point; e.g. 2p2 would be 2.2 picoFarads. Similarly, the lower-case letter “n” may be used as a decimal point as well (e.g. 2n2 would be 2.2 nanoFarads). Confusingly, some manufacturers use a capital letter “R” as a generic decimal point for the capacitance value, the default metric prefix once again being \textit{pico} (e.g. 2R2 would be 2.2 picoFarads).

A very common standard for small capacitor labeling is the use of three-digit numerical codes following the same digit-digit-multiplier format as resistor color codes, combined with the default assumption of \textit{pico}Farads. For example, 332 would represent \(33 \times 10^2\) picoFarads, which is equal to 3.3 nanoFarads. The following photograph shows a mylar film capacitor with a “103” code representing \(10 \times 10^3\) picoFarads, which is equivalent to 10 nanoFarads:
4.2. CAPACITOR LABELING

Capacitor *tolerance* is another important parameter which is often printed on the body of the capacitor. In one of the previous examples, we saw a 0.082 microFarad capacitor with a ± 10% tolerance, printed exactly as such on its body. However, for physically small capacitors where not enough surface area exists to print percentage figures, we must resort to other means for expressing tolerance.

To this end, a system of letter-codes has been developed, shown here in the following table:

<table>
<thead>
<tr>
<th>Letter code</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>± 0.05 pF</td>
</tr>
<tr>
<td>B</td>
<td>± 0.1 pF</td>
</tr>
<tr>
<td>C</td>
<td>± 0.25 pF</td>
</tr>
<tr>
<td>D</td>
<td>± 0.5 pF</td>
</tr>
<tr>
<td>E</td>
<td>± 0.5%</td>
</tr>
<tr>
<td>F</td>
<td>± 1%</td>
</tr>
<tr>
<td>G</td>
<td>± 2%</td>
</tr>
<tr>
<td>H</td>
<td>± 3%</td>
</tr>
<tr>
<td>J</td>
<td>± 5%</td>
</tr>
<tr>
<td>K</td>
<td>± 10%</td>
</tr>
<tr>
<td>L</td>
<td>± 15%</td>
</tr>
<tr>
<td>M</td>
<td>± 20%</td>
</tr>
<tr>
<td>N</td>
<td>± 30%</td>
</tr>
<tr>
<td>P</td>
<td>± −0% to +100%</td>
</tr>
<tr>
<td>S</td>
<td>± −20% to +50%</td>
</tr>
<tr>
<td>W</td>
<td>± −0% to +200%</td>
</tr>
<tr>
<td>X</td>
<td>± −20% to +40%</td>
</tr>
<tr>
<td>Z</td>
<td>± −20% to +80%</td>
</tr>
</tbody>
</table>

Note how the first four of these codes refer to absolute tolerances in picoFarads, while the rest represent tolerances in percent. When printed on a capacitor’s body, the tolerance code typically follows the numerical capacitance code. For example, the photograph on the previous page showed a green-colored capacitor with “103K” on its body, which is $10 \times 10^3$ picoFarads ± 10%.
4.3 Capacitor parameters and dielectric types

Capacitors come in many different types, the primary distinction being the type of substance used for the dielectric (i.e. the insulating layer between the capacitor plates). Rather than reproduce tables of dielectric material types and various performance ratings for each, here we will list several important parameters and for each of these cite the best and worst dielectric materials.

- **Capacitance density**: the amount of capacitance per unit volume, generally a desirable feature – “supercapacitors” exploiting such effects as double-layer and pseudocapacitance are the undisputed champions for this parameter. For conventional dielectrics, aluminum and tantalum electrolytic are best. Worst is vacuum or air.

- **Insulation resistance**: the desirable ability of the dielectric to oppose “leakage current” through the capacitor – polystyrene film is best; tantalum electrolytic is worst.

- **ESR and dissipation factor**: mechanisms of energy dissipation within the capacitor, any of which are undesirable – polystyrene, mica, and C0G/NPO ceramic is best; tantalum electrolytic is worst.

- **Dielectric absorption**: the undesirable tendency of a dielectric to maintain a DC bias after rapid discharging – polypropylene and polystyrene are best; tantalum electrolytic is worst.

- **Stability**: the degree to which a capacitor’s capacitance value undesirably drifts over long periods of time – C0G/NPO ceramic is best; several dielectric materials tie for worst (HiK ceramic, reduced titanate, multi-layer glass with “V” characteristic).

- **Temperature coefficient**: the degree to which a capacitor’s capacitance value undesirably changes with temperature – C0G/NPO ceramic is best; HiK ceramic is worst.

- **Voltage coefficient**: the degree to which a capacitor’s capacitance value undesirably changes with applied voltage – electrolytics are best; Class 2 ceramic and plastic-film are worst. Any dielectric material with paraelectric or ferroelectric properties (e.g. Class 2 ceramics such as barium titanate) exhibit this trait because their permittivity values vary significantly as the electric field increases and decreases in strength.

- **Polarization**: whether or not the capacitor requires unidirectional voltage polarity – all electrolytics are polarized; others are not.

The principal mechanism of energy dissipation within a capacitor is dielectric loss, where a varying electric field created by a varying voltage across the capacitor’s plates causes the molecules inside the dielectric to shift ever so slightly. This voltage-induced molecular movement results in the dielectric becoming warm, thus translating electrical energy into thermal energy where it cannot return to the circuit. Interestingly, this phenomenon is actually exploited in certain industrial processes such as the heating of plastics, where high-voltage and high-frequency AC applied across the insulating material causes it to heat up. In a capacitor, though, this effect is universally undesirable. Not only does dielectric loss constitute energy diverted from the circuit, but the heating will cause the capacitor’s own capacitance to drift and may even destroy the device if severe enough.
4.3. CAPACITOR PARAMETERS AND DIELECTRIC TYPES

The following table attempts to rank ten different dielectric types according to five important performance parameters. Please note that these rankings are based on some worst-case data I found, and that the performance metrics of specific capacitor models may likely vary from the values and orders shown below. As always, manufacture literature and/or empirical measurement are your best sources of information:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>NP0 = 0.3%</td>
<td>NP0 = 0.1%</td>
<td>PolyP = 0.05%</td>
<td>NP0 = 0.1%</td>
<td>PolyS = 10⁶ MΩ</td>
</tr>
<tr>
<td></td>
<td>PolyS = 1%</td>
<td>Mica = 0.1%</td>
<td>PolyS = 0.05%</td>
<td>Mica = 0.1%</td>
<td>PolyP = 10⁵ MΩ</td>
</tr>
<tr>
<td></td>
<td>Mica = 1.8%</td>
<td>PolyS = 2%</td>
<td>PolyC = 0.35%</td>
<td>PolyS = 0.1%</td>
<td>NP0 = 10⁵ MΩ</td>
</tr>
<tr>
<td></td>
<td>PolyC = 2%</td>
<td>PolyP = 3%</td>
<td>PolyE = 0.5%</td>
<td>PolyP = 0.35%</td>
<td>PolyC = 10⁴ MΩ</td>
</tr>
<tr>
<td></td>
<td>PolyP = 2.5%</td>
<td>PolyC = 5%</td>
<td>NP0 = 0.6%</td>
<td>PolyC = 1%</td>
<td>HiK = 10⁴ MΩ</td>
</tr>
<tr>
<td></td>
<td>Tant = 8%</td>
<td>PolyE = 10%</td>
<td>Mica = 0.7%</td>
<td>PolyE = 2%</td>
<td>PolyE = 10³ MΩ</td>
</tr>
<tr>
<td></td>
<td>Alum = 10%</td>
<td>Tant = 10%</td>
<td>HiK = 3%</td>
<td>RedT = 10² MΩ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PolyE = 12%</td>
<td>Alum = 10%</td>
<td>Alum = 8%</td>
<td>Mica = 10² MΩ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RedT = 30%</td>
<td>RedT = 20%</td>
<td>Alum</td>
<td>RedT = 10%</td>
<td>Alum = 10³ MΩ</td>
</tr>
<tr>
<td>Worst</td>
<td>HiK = 80%</td>
<td>HiK = 20%</td>
<td>Tant</td>
<td>Tant = 24%</td>
<td>Tant = 10¹ MΩ</td>
</tr>
</tbody>
</table>

Abbreviations for dielectric types:
- NP0 = Multi-layer NP0-type ceramic
- HiK = Multi-layer HiK-type ceramic
- RedT = Reduced titanate
- PolyE = Polyester film
- PolyC = Polycarbonate film
- PolyP = Polypropylene film
- PolyS = Polystyrene film
- Tant = Solid tantalum electrolytic
- Alum = Aluminum electrolytic

Also note that conservative design principles may help minimize undesirable capacitor traits. For example, if the capacitor is known to have a large temperature coefficient, placing it in an area of the circuit board where you would expect temperature to be most stable will mitigate this shortcoming. Similarly, if the capacitor has a large voltage coefficient a good mitigation strategy is to over-rate the capacitor in terms of voltage (i.e. select a capacitor with a much higher voltage rating than the expected voltage for your application).
4.4 Capacitor parasitics

An ideal capacitor exhibits only capacitance, with no inductance, resistance, or other characteristics to interfere. Real capacitors exhibit all these phenomena to varying degrees, and we collectively refer to these undesirable traits as *parasitic effects*. The following diagram models some of the parasitic effects observed in real capacitors:

![Capacitor model diagram]

In addition to the capacitance the capacitor is supposed to exhibit \( C_{\text{ideal}} \), the capacitor also has parasitic resistance \( R_{\text{series}} \), also known as *Equivalent Series Resistance*, or *ESR*), parasitic inductance \( L_{\text{series}} \), and additional energy storage in the form of *soakage* (also known as *dielectric absorption*) whereby the dielectric substance itself absorbs and releases energy after relatively long periods of time compared to the main (ideal) capacitance.

Some of these parasitic effects – such as leakage resistance and soakage – affect the capacitor’s performance in DC applications. Most of the other parasitic effects cause problems in AC and pulsed applications. For example, the effective series capacitor-inductor combination formed by \( C_{\text{ideal}} \) and \( L_{\text{series}} \) will cause *resonance* to occur at a particular AC frequency, resulting in much less reactance at that frequency than what would be predicted by the capacitive reactance formula \( X_C = \frac{1}{2\pi fC} \).

Next we will explore common mechanisms for each of these effects.
4.4.1 Parasitic resistance in capacitors

Wire resistance, of course, plays a part in this parasitic effect, but this is generally very small due to the short conductor lengths we typically see with capacitors. More significant is dielectric losses – energy dissipation caused by the stressing and relaxation of dipoles within the dielectric material – which act like resistance because energy ends up leaving the component (in the form of heat) and not returning to the circuit. Electrolytic capacitors have an additional source of parasitic resistance, in the form of the electrolytic gel substance used to make electrical contact from the metal-foil “plate” to the surface of the dielectric layer.

Another form of parasitic resistance within a capacitor behaves like a resistor connected in parallel with the ideal capacitance ($R_{\text{leakage}}$), resulting from the dielectric not being a perfectly insulating medium. This parasitic characteristic results in a small current passing through the capacitor even when the voltage across the capacitor is steady (i.e. $\frac{dV}{dt}$ is zero).

4.4.2 Parasitic inductance in capacitors

Any time a magnetic field forms around a current-carrying conductor, energy is stored in that magnetic field. We call this magnetic-based energy-storing capability inductance, and of course all capacitors must have some inductance due to the wire leads serving as connection points to the capacitor’s metal plates. Much of a capacitor’s parasitic inductance may be minimized by maintaining short lead lengths as it attaches to a printed-circuit board.

Parasitic inductance is a problem for capacitors in AC applications because inductive reactance ($X_L$) tends to cancel out capacitive reactance ($X_C$). If we plot the impedance of a capacitor as a function of frequency, we would expect an ideal capacitor to manifest a straight-line descent on a logarithmic plot. However, what we see is that at a certain frequency the parasitic inductance resonates with the capacitance leaving only parasitic resistance (ESR), and then past that frequency the inductive effects overshadow the capacitance:
4.4.3 Other parasitic effects in capacitors

Soakage is an interesting effect resulting from dipole relaxation within the dielectric material itself, and may be modeled (as shown) by a series of resistor-capacitor subnetworks. This effect is especially prominent in aluminum electrolytic capacitors, and may be easily demonstrated by discharging a capacitor (by briefly connecting a shorting wire across the capacitor’s terminals) and then monitoring the capacitor’s DC voltage slowly “recover” with no connection to an external source.
4.5 Electric field quantities

A useful definition of electric field \( E \) is in terms of the force \( F \) exerted on an electric charge \( Q \) influenced by that field:

\[
\vec{F} = Q \vec{E}
\]

Where,
- \( \vec{F} \) = Force exerted on the charge (Newtons)
- \( Q \) = Charge quantity (Coulombs\(^4\))
- \( \vec{E} \) = Electric field (Newtons per Coulomb)

The small “arrow” symbols above the variables for force and electric field in the equation denote those variables as *vector quantities*, having both magnitude and direction. Charge is a *scalar quantity* having only magnitude but no direction, and as a scalar quantity when multiplied by the electric field vector it simply magnifies the magnitude but does not alter the direction. Therefore, the force and electric field vectors always point in the same direction.

Alternatively electric field may be defined in terms of the voltage between the end-points and the distance separating them, in which case we may express the electric field in units of *Volts per meter* as an alternative to *Newtons per Coulomb*:

\[
\vec{E} = \frac{V}{d}
\]

This measurement of electric field strength is very important for quantifying the *breakdown* of electrical insulators: the point at which the electric field becomes so powerful that otherwise immobile charges within the insulating substance are torn free to constitute a current and that substance is no longer an insulator. For rating the dielectric strength of insulating materials, we often see electric fields expressed in units of *kiloVolts per millimeter* rather than Volts per meter just to make the numerical quantities easier to manage (1 kV/mm = 1 million V/m).

\(^4\)One Coulomb of electric charge is equal to \(6.2415 \times 10^{18}\) electrons.
The vector arrows shown in the previous illustration representing the electric field between two metal plates actually represent electric flux ($\Phi_E$). The electric field ($\vec{E}$) is related to electric flux by area ($A$), the field being a measurement of how densely-packed those flux lines are per unit area:

$$\vec{E} = \frac{\Phi_E}{A}$$

Where,
- $\vec{E}$ = Electric field, or electric flux density (Newtons per Coulomb)
- $\Phi_E$ = Electric flux (Newton-meter squared per Coulomb)
- $A$ = Area over which flux is distributed (square meters)

The mere presence of an unbalanced electric charge at any point in space is sufficient to generate lines of electric flux, the total magnitude of that flux predicted by the following equation:

$$\Phi_E = \frac{Q}{\epsilon}$$

Where,
- $\Phi_E$ = Electric flux (Newton-meter squared per Coulomb)
- $Q$ = Charge quantity (Coulombs)
- $\epsilon$ = Electric permittivity of the surrounding space (Coulombs squared per Newton-meter squared, approximately $8.85 \times 10^{-12}$ for empty space)

By convention, these flux vectors point away from positive charges and point toward negative charges, their direction indicating force exerted on any positively-charged particle influenced by that field. As the equation states, the amount of flux depends on how much charge exists at each particle as well as the permittivity of the surrounding space:

For example, identical charges suspended in a vacuum versus in a substance such as oil will have different amounts of flux associated with them as a result of oil and vacuum having different permittivity values. Perfectly empty space has the least amount of permittivity, which means anything else (gas, liquid, or solid matter) has greater $\epsilon$ which acts to diminish the amount of electric flux surrounding any charged particle. Superconducting materials have infinite permittivity, which means they forbid the existence of any electric field inside their bulk.

\[^5\]Conversely, the flux vectors point exactly opposite the direction of force applied to any negatively-charged particle within that field. This makes sense of the aphorism that like charges repel and opposite charges attract. If you consider the two charges shown in this illustration, the positive charge will be pulled in the direction of the flux vectors pointing toward the negative charge, as the negative charge will also be pulled opposite the direction of the flux vectors pointing away from the positive charge (i.e. the negative charge will be pulled toward the positive charge) – thus the positive and negative charges feel mutual attraction.
Chapter 5

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.
5.1 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:

```cpp
#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 to view.

---

1Although 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\textsuperscript{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\textsuperscript{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

---

\textsuperscript{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!

\textsuperscript{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.
5.1. PROGRAMMING IN C++

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.
5.2 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` 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()
```

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

---

4Using 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:

```python
>>> 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

>>> pow(e, 6.0)
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 \( \log_{10} \) 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:

```python
>>> 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 \(z_c\) as \(X_C \angle -90^\circ\) 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 \((400 +j0 \ \Omega)\), then computed capacitive and inductive reactances at 60 Hz and defined each of those as complex (phasor) values \((0 - jX_c \ \Omega\) and \(0 + jX_l \ \Omega\), respectively). After that we computed total impedance in series, then total impedance in parallel. Polar-form representation was then shown for the series impedance \((441.717 \ \Omega \angle -25.102^\circ)\). 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`:

```python
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.
5.3 Modeling inverse exponential growth and decay using C++

Many different types of physical systems, including capacitor and inductor circuits, often exhibit a phenomenon known as inverse exponential growth and decay, where variables asymptotically approach final values over time. Consider the following two circuits, one capacitive and the other inductive, both components receiving an input of energy from a source:

Capacitors store energy as a function of voltage (manifest as an electric field), while inductors store energy as a function of current (manifest as a magnetic field). In both circuits, the reactive component receives energy from the source over time, which explains the capacitor’s rising voltage and the inductor’s rising current. The other variable in each circuit (i.e. current for the capacitor and voltage for the inductor) is a function of the rate at the energy-related variable grows, and so these variables approach zero as the energy-related variables level off at some terminal value.

In each case, the decreasing variable is said to decay to zero over time, and its path has the same shape as an inverse exponential function, $e^{-t}$. As time progresses (i.e. $t$ grows larger), the value of $e^{-t}$ diminishes. The full formula for each circuit includes a term for the initial (peak) value as well as the time constant ($\tau$) of the circuit:

$$I_C = I_0 e^{-\frac{t}{\tau}}$$

$$V_L = V_0 e^{-\frac{t}{\tau}}$$

In each case, the increasing variable is said to experience inverse exponential growth, and its shape appears to be a mirror-image of the decaying variable. Mathematically, the inverse growth function is simply the complement of the decay function, $1 - e^{-t}$. As time progresses (i.e. larger $t$), the value of $e^{-t}$ approaches zero and the value of $1 - e^{-t}$ approaches 1. The amount of time equal to one time constant ($\tau$) is the time required for the variable in question to go 63.2% ($1 - e^{-1}$) of

---

7An asymptote is a mathematical concept, consisting of a fixed value which some other variable approaches closer and closer but never fully attains.
the way from its starting value to its final (asymptote) value. For any variable decaying to zero, this is the time required to reach 36.8% ($e^{-1}$) of its starting value.

These exponential functions are not limited to describing capacitor and inductor circuits, but are actually useful for a wide range of physical systems. The decay of a hot object’s temperature\(^8\), for example, follows the same $e^{-t}$ trajectory.

We may employ a simple computer program to repeatedly calculate percentage values for such rising and decaying variables as time ($t$) progresses from zero to ten (measured in whole-multiples of time constants). This is a very good application for a computer, as it relieves us of both the tedium and the many opportunities for error faced with repeatedly calculating $e^{-t}$ and $1 - e^{-t}$.

The “source code” for this computer program, written in the C++ language, is shown here containing all the instructions necessary to tell the computer how to perform these repeated calculations and display the results on the console:

```c++
#include <iostream>
#include <cmath>
using namespace std;

int main (void)
{
    float initial, x, t;
    initial = 100.0; // This sets the initial value to 100%
    cout.precision(4);
    cout.setf(ios::fixed, ios::floatfield);
    cout << "Time \t\t" << "Falling \t" << "Rising" << endl;
    for (t = 0.0; t <= 10 ; t = t + 1.0)
    {
        x = initial * exp(-t);
        cout << t << " tau \t";
        cout << x << "\% \t";
        cout << 100.0 - x << "\% \t";
    }
    return 0;
}
```

\(^8\)In fact, this is referred to as *Newton’s Cooling Law*. 
Let’s analyze the lines of code within this short C++ program, touching on the following programming principles:

- Reading/execution order
- Preprocessor directives (#include) and namespace
- The main function
- Delimiter characters (e.g. \{ \} ;)
- Variable declarations
- float and int data types and variable names
- Variable initialization and assignment (e.g. =)
- The cout instruction
- for loops
- Mathematical functions
- Return

As with most text-based programming languages, C and C++ alike are read by the compiler software and executed in the same order one reads an English-language document: left to right character by character, and top to bottom line by line.

The first three lines (#include and using namespace) direct the compiler software on how to interpret much of what follows. The using line tells the compiler to use the “standard” namespace\(^9\) convention for C++, while the two previous lines (called preprocessor directives) request the contents of the iostream and math header files to be included when compiling the code into an executable file. For example, the cout instruction is defined within iostream while the exp function is defined within math.

All C++ programs have a main function marking the starting point of execution. All of our circuit-modeling code appears between the main function’s curly-brace (\{ \}) symbols, those symbols denoting the boundaries of the main function. This particular main function is defined here as returning an integer number value (int) when complete, and it requires no input of information (void).

In C, C++, and some other languages such as Java, parentheses and “curly-brace” symbols serve to group certain types of information together, much the same as parentheses and brackets group terms together in mathematical formulae. Parentheses immediately following a function name enclose arguments to that function: i.e. data that function must act upon. Curly-brace symbols define the starting and ending points for lines of code to be executed within a given function.

\(^9\)Namespaces are a concept found in C++ but not in its predecessor language C. Without getting too detailed, namespaces allow advanced programmers to redefine the meaning of certain key instructions and functions within the language for their own purposes. In this simple program we are doing no such thing, and so we instruct the compiler to use the standard (std) namespace of C++.
The line beginning with `float` reserves space in the computer’s memory for three `floating-point` variables. “Floating-point” is a type of digital numeration useful for expressing non-integer values both large and small. C++ permits variable names ranging from single letters (e.g. `x`) to entire words (e.g. `initial`).

This declaration line ends with a semicolon delimiter (`;`) telling the compiler where the line ends, which is important because neither C nor C++ takes heed of “whitespace” in the source file. You might think of the C/C++ semicolon as being the programming-language equivalent of a period at the end of an English sentence. If not for these semicolons, the compiler would treat all the lines as being a single, continuous line in the same way you would read unterminated lines in a page of text as part of a long sentence. You will notice most of the lines of code end with semicolons, exceptions being `main` (because its curly-brace symbols serve the same purpose) and the `#include` directives which are technically set-up instructions for the compiler and not executed at run-time.

Immediately following the declaration line is a line of code initializing one of those variables with a numerical value, in this case setting the initial value to `100.0` which will represent 100% of source voltage or current in a capacitive or inductive circuit. In C and C++, a single “equals” symbol (`=`) signifies assignment rather than the conventional mathematical meaning of equality. Whenever you see a single equals symbol in C or C++, think “set equal to”.

Several lines of code in this short program begin with `cout` which is the C++ instruction for outputting text to the computer’s console display. When `cout` is accompanied by the “put to” operator (`<<`) it means that instruction will output either verbatim text or text-control characters (enclosed in quotation marks) or the numerical values of any variables referenced by `cout`. Text-control characters include `t` (tab) instructing a horizontal shift in text and `endl` (end-line) signifying the end of line line and the beginning of another as displayed on the computer’s console.

Those `cout` lines lacking “put to” (`<<`) operators instruct subsequent instances of `cout` how to behave. For example, `cout.precision(4)` tells `cout` to display every numerical output at four decimal places to the right of the decimal point. In the next line of code, `cout.setf(ios::fixed, ios::floatfield)` instructs `cout` to used a fixed-point notation when displaying numbers. Both of these formatting instructions will make the program’s displayed text look neater on the computer’s console, but neither is strictly necessary if all we care about is function and not appearance.

All the interesting operations occur within the `for` loop, where time (\(t\)) is incremented from 0 to 10 in steps of 1, and the computer calculates \(e^{-t}\) with the function `exp(-t)`. In C and C++, a `for` loop instructs the computer to repeat certain lines of code (enclosed in another set of curly-braces) while a particular variable changes from one value to another. In this particular case the variable in question is \(t\), starting at a value of 0.0 and incrementing by 1.0 with each new iteration of the `for` loop. The loop repeats so long as \(t\) is equal to or less than 10.0.

Three `cout` lines contained within the `for` loop work together to generate a single line of printed text. The first `cout` prints the time variable (\(t\)), the next `cout` prints the value of the “falling” variable \(x\), and the last `cout` prints the value of the “rising” variable which is simply the complement of \(x\). A single `endl` control character is printed with that last `cout` instruction to terminate the line and prepare for a new line of text with the next iteration of the `for` loop.

Finally, the last line of code in this program returns an integer value of zero as the `main` function successfully completes. This is not strictly necessary, but is a good programming practice.
5.3. MODELING INVERSE EXPONENTIAL GROWTH AND DECAY USING C++

When compiled and executed, this program prints the percentage values of the rising and falling quantities over a timespan of zero to ten time constants (i.e. 0 to 10τ). It is instructive to compare this text output to the lines of code previously shown, and to follow the execution path of the code in your mind. For example, a careful examination of the code should reveal to you why we only see one line reading **Time Falling Rising** but several lines of numbers:

<table>
<thead>
<tr>
<th>Time</th>
<th>Falling</th>
<th>Rising</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000 tau</td>
<td>100.0000%</td>
<td>0.0000%</td>
</tr>
<tr>
<td>1.0000 tau</td>
<td>36.7879%</td>
<td>63.2121%</td>
</tr>
<tr>
<td>2.0000 tau</td>
<td>13.5335%</td>
<td>86.4665%</td>
</tr>
<tr>
<td>3.0000 tau</td>
<td>4.9787%</td>
<td>95.0213%</td>
</tr>
<tr>
<td>4.0000 tau</td>
<td>1.8316%</td>
<td>98.1684%</td>
</tr>
<tr>
<td>5.0000 tau</td>
<td>0.6738%</td>
<td>99.3262%</td>
</tr>
<tr>
<td>6.0000 tau</td>
<td>0.2479%</td>
<td>99.7521%</td>
</tr>
<tr>
<td>7.0000 tau</td>
<td>0.0912%</td>
<td>99.9088%</td>
</tr>
<tr>
<td>8.0000 tau</td>
<td>0.0335%</td>
<td>99.9665%</td>
</tr>
<tr>
<td>9.0000 tau</td>
<td>0.0123%</td>
<td>99.9877%</td>
</tr>
<tr>
<td>10.0000 tau</td>
<td>0.0045%</td>
<td>99.9955%</td>
</tr>
</tbody>
</table>

At the very start (t = 0) the falling variable is at its maximum value of 100% and the rising variable is at zero. After one time constants’ worth of time (t = τ) the falling variable has fallen well over half-way to a value of approximately 36.8% of its initial value. At that same time the rising variable has risen well over half-way to just over 63.2%. After ten time constants’ worth of time (t = 10τ) we can see both variables are a small fraction of a percent away from their final values.

The first point in time where we see the variables come within one percent of their final values is at \( t = 5\tau \). This is why students are commonly taught that the variables have “settled” for all practical purposes after five time constants. It is important to understand, however, that there is nothing truly significant about \( 5\tau \). Five just happens to be the first whole-number multiple of \( \tau \) where the variables approach their final values to within the completely arbitrary threshold of 1%. Just to be clear on this point: the rising and falling variables indeed settle to within 1% of their final values after five time constants’ worth of time has passed, but their exact value at \( 5\tau \) is not 1%, nor do they reach the 1% threshold after exactly \( 5\tau \), and they certainly are not fully settled by then. The rest of this section is devoted to overturning these misconceptions by modeling the inverse exponential functions in greater detail.
We could alter our program to take smaller steps in time, incrementing $t$ by steps of 0.4 instead than by steps of 1, if we wished to take a closer look at exactly when these falling and rising variables come within 1% of their final values. We need only alter one line of code to do this, the conditions controlling execution of the for loop:

$$\text{for } (t = 0.0; t < 10; t = t + 0.4)$$

Now when we re-compile and run this program, we obtain a much more precise table of numbers:

<table>
<thead>
<tr>
<th>Time</th>
<th>Falling</th>
<th>Rising</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>tau 100.0000%</td>
<td>0.0000%</td>
</tr>
<tr>
<td>0.4000</td>
<td>tau 67.0320%</td>
<td>32.9680%</td>
</tr>
<tr>
<td>0.8000</td>
<td>tau 44.9329%</td>
<td>55.0671%</td>
</tr>
<tr>
<td>1.2000</td>
<td>tau 30.1194%</td>
<td>69.8806%</td>
</tr>
<tr>
<td>1.6000</td>
<td>tau 20.1897%</td>
<td>79.8103%</td>
</tr>
<tr>
<td>2.0000</td>
<td>tau 13.5335%</td>
<td>86.4665%</td>
</tr>
<tr>
<td>2.4000</td>
<td>tau 9.0718%</td>
<td>90.9282%</td>
</tr>
<tr>
<td>2.8000</td>
<td>tau 6.0810%</td>
<td>93.9190%</td>
</tr>
<tr>
<td>3.2000</td>
<td>tau 4.0762%</td>
<td>95.9238%</td>
</tr>
<tr>
<td>3.6000</td>
<td>tau 2.7324%</td>
<td>97.2676%</td>
</tr>
<tr>
<td>4.0000</td>
<td>tau 1.8316%</td>
<td>98.1684%</td>
</tr>
<tr>
<td>4.4000</td>
<td>tau 1.2277%</td>
<td>98.7723%</td>
</tr>
<tr>
<td>4.8000</td>
<td>tau 0.8230%</td>
<td>99.1770%</td>
</tr>
<tr>
<td>5.2000</td>
<td>tau 0.5517%</td>
<td>99.4483%</td>
</tr>
<tr>
<td>5.6000</td>
<td>tau 0.3698%</td>
<td>99.6302%</td>
</tr>
<tr>
<td>6.0000</td>
<td>tau 0.2479%</td>
<td>99.7521%</td>
</tr>
<tr>
<td>6.4000</td>
<td>tau 0.1662%</td>
<td>99.8338%</td>
</tr>
<tr>
<td>6.8000</td>
<td>tau 0.1114%</td>
<td>99.8886%</td>
</tr>
<tr>
<td>7.2000</td>
<td>tau 0.0747%</td>
<td>99.9253%</td>
</tr>
<tr>
<td>7.6000</td>
<td>tau 0.0500%</td>
<td>99.9500%</td>
</tr>
<tr>
<td>8.0000</td>
<td>tau 0.0335%</td>
<td>99.9665%</td>
</tr>
<tr>
<td>8.4000</td>
<td>tau 0.0225%</td>
<td>99.9776%</td>
</tr>
<tr>
<td>8.8000</td>
<td>tau 0.0151%</td>
<td>99.9849%</td>
</tr>
<tr>
<td>9.2000</td>
<td>tau 0.0101%</td>
<td>99.9899%</td>
</tr>
<tr>
<td>9.6000</td>
<td>tau 0.0068%</td>
<td>99.9932%</td>
</tr>
<tr>
<td>10.0000</td>
<td>tau 0.0045%</td>
<td>99.9955%</td>
</tr>
</tbody>
</table>

This simulation proves $5\tau$ is not the true point at which we get within 1% of final value, since we see the values come within 1% after only 4.8 time constants.
We may focus our computer simulation even tighter by altering the for loop conditions once again for finer resolution (and more limited range, so as to not generate an enormously long table of values):

\[
\text{for } (t = 4.55; t < 4.7 ; t = t + 0.01)
\]

The result, when compiled and run again is as follows:

<table>
<thead>
<tr>
<th>Time</th>
<th>Falling</th>
<th>Rising</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5500</td>
<td>1.0567%</td>
<td>98.9433%</td>
</tr>
<tr>
<td>4.5600</td>
<td>1.0462%</td>
<td>98.9538%</td>
</tr>
<tr>
<td>4.5700</td>
<td>1.0358%</td>
<td>98.9642%</td>
</tr>
<tr>
<td>4.5800</td>
<td>1.0255%</td>
<td>98.9745%</td>
</tr>
<tr>
<td>4.5900</td>
<td>1.0153%</td>
<td>98.9847%</td>
</tr>
<tr>
<td>4.6000</td>
<td>1.0052%</td>
<td>98.9948%</td>
</tr>
<tr>
<td>4.6100</td>
<td>0.9952%</td>
<td>99.0048%</td>
</tr>
<tr>
<td>4.6200</td>
<td>0.9853%</td>
<td>99.0147%</td>
</tr>
<tr>
<td>4.6300</td>
<td>0.9755%</td>
<td>99.0245%</td>
</tr>
<tr>
<td>4.6400</td>
<td>0.9658%</td>
<td>99.0342%</td>
</tr>
<tr>
<td>4.6500</td>
<td>0.9562%</td>
<td>99.0438%</td>
</tr>
<tr>
<td>4.6600</td>
<td>0.9466%</td>
<td>99.0534%</td>
</tr>
<tr>
<td>4.6700</td>
<td>0.9372%</td>
<td>99.0628%</td>
</tr>
<tr>
<td>4.6800</td>
<td>0.9279%</td>
<td>99.0721%</td>
</tr>
<tr>
<td>4.6900</td>
<td>0.9187%</td>
<td>99.0813%</td>
</tr>
</tbody>
</table>

With this closer view we see the values come within 1\% of final closer to 4.61 time constants. In other words, $5\tau$ is a rather coarse rounding-off of the true time value for settling within 1\%.

Of course, the best way to determine how long it takes for these variables to come within 1\% of their final values is to algebraically solve for $t$ with the decay function set equal to 0.01 (i.e. 1\%):

\[
0.01 = e^{-t}
\]

\[
\ln 0.01 = -t
\]

\[
t = -\ln 0.01
\]

\[
t = 4.605170186
\]

It should be clear by now that the frequently-taught rule of “settled within five time constants” is a rounded-off estimation based on an arbitrary threshold and should be regarded as such. In any timing applications requiring precision, the rule of $5\tau$ just isn’t good enough and you will need to actually calculate $t$. 
5.4 Modeling an energizing capacitor using C++

Here is an example C++ program intended to calculate voltage and current for an energizing capacitor:

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

int main (void)
{
    float v1, r1, c1, i, vr1, vc1, tau, t, incr;
    t = 0.0;
    incr = 0.05;
    v1 = 15.00;
    r1 = 3.3e3;
    c1 = 22e-6;

    cout << "+---V1---R1---C1---+ " << endl;
    cout << "| | " << endl;
    cout << "+------------------+ " << endl;
    cout << "V1 = " << v1 << " Volts" << endl;
    cout << "R1 = " << r1 << " Ohms" << endl;
    cout << "C1 = " << c1 << " Farads" << endl;

    tau = r1 * c1;

    // This "FOR" loop repeats until the capacitor's
    // voltage reaches 99% of the source voltage
    for (t = 0.00 ; vc1 < (0.99 * v1) ; t = t + incr)
    {
        vc1 = v1 * (1 - exp(-t / tau));
        vr1 = v1 - vc1;
        i = vr1 / r1;

        cout << endl;
        cout << "Time = " << t << " Seconds" << endl;
        cout << " I = " << i << " Amperes" << endl;
        cout << " VR1 = " << vr1 << " Volts" << endl;
        cout << " VC1 = " << vc1 << " Volts" << endl;
    }
}```
Note the use of a for “loop” in the code, commanding the computer to repeat a set of instructions so long as some condition is met. In this case, the condition is that the capacitor’s voltage is less than 99% of the source voltage, as described within the text of the code by the comment lines (those two lines of text preceded by double forward-slash characters). Note also the use of the exponential function (exp) used to compute $e^{-\frac{t}{\tau}}$, which requires an additional “include” statement at the beginning of the code (#include <cmath>) which gives the compiler access to descriptions of mathematical functions more advanced than those defined within iostream.

When compiled and executed, this program generates the following output:

```
        +---V1---R1---C1---+
        |                 |
        +------------------+

V1 = 15 Volts
R1 = 3300 Ohms
C1 = 2.2e-05 Farads

Time = 0 Seconds
    I = 0.00454545 Amperes
    VR1 = 15 Volts
    VC1 = 0 Volts

Time = 0.05 Seconds
    I = 0.00228284 Amperes
    VR1 = 7.53339 Volts
    VC1 = 7.46661 Volts

Time = 0.1 Seconds
    I = 0.0011465 Amperes
    VR1 = 3.78346 Volts
    VC1 = 11.2165 Volts

Time = 0.15 Seconds
    I = 0.000575804 Amperes
    VR1 = 1.90015 Volts
    VC1 = 13.0998 Volts

Time = 0.2 Seconds
    I = 0.000289184 Amperes
    VR1 = 0.954306 Volts
```
VC1 = 14.0457 Volts

Time = 0.25 Seconds
I = 0.000145236 Amperes
VR1 = 0.479278 Volts
VC1 = 14.5207 Volts

Time = 0.3 Seconds
I = 7.29411e-05 Amperes
VR1 = 0.240705 Volts
VC1 = 14.7593 Volts

Time = 0.35 Seconds
I = 3.66329e-05 Amperes
VR1 = 0.120889 Volts
VC1 = 14.8791 Volts
Chapter 6

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 *Reading 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.
CHAPTER 6. QUESTIONS

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.
CHAPTER 6. QUESTIONS

• 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?
6.1 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.
6.1.1 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, write their own outline and 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 OUTLINE THE TEXT, as though you were writing a detailed Table of Contents. Feel free to rearrange the order if it makes more sense that way. Prepare to articulate these points in detail and to answer questions from your classmates and instructor. Outlining is a good self-test of thorough reading because you cannot outline what you have not read or do 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.
### 6.1.2 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 full 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.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thought experiments as a problem-solving strategy</td>
<td></td>
</tr>
<tr>
<td>Electric field</td>
<td></td>
</tr>
<tr>
<td>Energy</td>
<td></td>
</tr>
<tr>
<td>Conservation of Energy</td>
<td></td>
</tr>
<tr>
<td>Capacitance</td>
<td></td>
</tr>
<tr>
<td>Dielectric</td>
<td></td>
</tr>
<tr>
<td>Permittivity</td>
<td></td>
</tr>
<tr>
<td>Capacitor</td>
<td></td>
</tr>
<tr>
<td>Electrical source</td>
<td></td>
</tr>
<tr>
<td>Electrical load</td>
<td></td>
</tr>
<tr>
<td>“Ohm’s Law” for capacitance</td>
<td></td>
</tr>
<tr>
<td>Voltage stabilization</td>
<td></td>
</tr>
</tbody>
</table>
Ohm’s Law

Kirchhoff’s Voltage Law

Time constant

Dielectric strength

Polarized capacitor

Farad

Leakage resistance

Equivalent series resistance (ESR)

Capacitor physical size

Series versus parallel capacitances
6.1. Conceptual Reasoning

6.1.3 Electric flux lines
Suppose two wires, separated by an air gap, are connected to opposite terminals on a voltage source (such as a battery). An electric field will develop in the space between the two wires: an invisible web of interaction, similar in some ways to a magnetic field. In this diagram, draw the invisible “lines of flux” for this electric field, showing their physical range:

![Diagram of electric flux lines]

Challenges
- Explain how electric flux lines differ in geometry from magnetic flux lines.

6.1.4 Leyden jar
What is a Leyden Jar, and how is its construction similar to the construction of all capacitors?

Challenges
- Identify how you could construct your own Leyden jar.
6.1.5 Water tank analogy

If we introduce a constant flow of water into a cylindrical tank with water, the height of water inside that tank will rise at a constant rate over time:

Like the water tank, electrical \textit{capacitance} also exhibits the phenomenon of integration with respect to time. Which electrical quantity (voltage or current) dictates the rate-of-change over time of which other quantity (voltage or current) in a capacitance? Or, to re-phrase the question, which quantity (voltage or current), when maintained at a constant value, results in which other quantity (current or voltage) steadily ramping either up or down over time?

<table>
<thead>
<tr>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Can you think of a way we could exploit the similarity of capacitive voltage/current integration to \textit{simulate} the behavior of a water tank’s filling, or any other physical process described by the same mathematical relationship?</td>
</tr>
</tbody>
</table>
6.1.6 Cable capacitance

Capacitance exists between any two conductors separated by an insulating medium. Given this fact, it makes sense that a length of two-conductor electrical cable will have capacitance distributed naturally along its length:

![Two-conductor cable diagram]

Devise a means of proving the existence of this “stray” capacitance in any substantial length of two-conductor cable.

- Challenges

• Devise at least two more methods to prove the existence of capacitance between the conductors of a long cable.
6.1.7 Capacitor as source versus load

Suppose a capacitor is connected directly to an adjustable-voltage source, and the voltage of that source is steadily increased over time. We know that an increasing voltage across a capacitor will produce an electric field of increasing strength. Does this increase in electric field constitute an accumulation of energy in the capacitor, or a release of energy from the capacitor? In this scenario, does the capacitor act as a load or as a source of electrical energy?

Voltage increasing

Now, suppose the adjustable voltage source is steadily decreased over time. We know this will result in an electric field of decreasing strength in the capacitor. Does this decrease in electric field constitute an accumulation of energy in the capacitor, or a release of energy from the capacitor? In this scenario, does the capacitor act as a load or as a source of electrical energy?

Voltage decreasing

For each of these scenarios, label the direction of current in the circuit.

Challenges

- Identify factors influencing the current’s magnitude in each circuit.
6.1.8 Audio power capacitors

Very large capacitors (typically in excess of 1 Farad!) are often used in the DC power wiring of high-power audio amplifier systems installed in automobiles. The capacitors are connected in parallel with the amplifier’s DC power terminals, as close to the amplifier as possible, like this:

What is the purpose of having a capacitor connected in parallel with the amplifier’s power terminals? What benefit does this give to the audio system, overall?

Challenges

- A similar application for capacitors is called decoupling in the DC power circuitry of digital electronic circuits. Research this topic and explain the similarity to the audio power application.
6.1.9 Capacitor charging circuit

Qualitatively determine the following parameters in this simple resistor-capacitor circuit at three different times: (1) just before the switch closes, (2) at the instant the switch contacts touch, and (3) after the switch has been closed for a long time. Assume that the capacitor begins in a completely discharged state:

Express your answers qualitatively: “maximum,” “minimum,” or perhaps “zero” if you know that to be the case.

**Before the switch closes:**
- $E_C$ (energy stored in capacitor) =
- $V_C =$
- $V_R =$
- $V_{\text{switch}} =$
- $I =$

**At the instant of switch closure:**
- $E_C$ (energy stored in capacitor) =
- $V_C =$
- $V_R =$
- $V_{\text{switch}} =$
- $I =$

**Long after the switch has closed:**
6.1. CONCEPTUAL REASONING

Long after the switch has closed:
\[ E_C \text{ (energy stored in capacitor)} = \]
\[ V_C = \]
\[ V_R = \]
\[ V_{\text{switch}} = \]
\[ I = \]

Hint: a graph may be a helpful tool for determining the answers!

Also, identify how the following foundational concepts apply to this circuit:

- How does the Conservation of Energy help explain the capacitor’s voltage at the instant of switch closure?
- How does Kirchhoff’s Voltage Law apply to the circuit before the switch closes?
- How does Kirchhoff’s Voltage Law apply to the circuit after the switch closes?

Challenges

- Explain how the Conservation of Energy relates to changes in voltage and current for a capacitor.
- In this experiment, is the capacitor absorbing energy or releasing energy? How can we tell?
- Explain what will happen in the circuit when the switch is re-opened after the switch has been closed for a long period of time.
6.1.10 Analog electrical model of warming box

Suppose we were measuring the interior temperature of an insulated box recently removed from a refrigerator, as it was being warmed by the ambient air around it:

(Ambient air warmer than inside of box)

Graphing the box’s temperature over time, we see a curve that looks something like this:

An engineer approaches you and says she wants you to build an electrical circuit that models this thermal system, otherwise known as an electrical analog circuit. What kind of circuit would you consider building for the engineer, to make a realistic electrical analogue of the box’s temperature? Be as specific as you can in your answer.

Challenges

- Identify how your analog circuit could be adjusted to represent a greater amount of insulation for the box.
- Identify how your analog circuit could be adjusted to represent a greater amount of mass stored inside the box.
- Identify how your analog circuit could be adjusted to represent a greater room temperature.
6.2 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” for gaining proficiency in the use of circuit simulation software, and then once that proficiency is gained you will never need to rely on an answer key!

\[\text{4In 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.}

\[\text{5This approach is perfectly in keeping with the instructional philosophy of these learning modules: teaching students to be self-sufficient thinkers. Answer keys can be useful, but it is even more useful to your long-term success to have a set of tools on hand for checking your own work, because once you have left school and are on your own, there will no longer be “answer keys” available for the problems you will have to solve.}\]
6.2.1 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, Avogadro’s number given as 6.02214179(30) × 10^{23} means the center value (6.02214179 × 10^{23}) plus or minus 0.00000030 × 10^{23}.

Avogadro’s number \( (N_A) = 6.02214179(30) \times 10^{23} \) per mole (mol\(^{-1}\))

Boltzmann’s constant \( (k) = 1.3806504(24) \times 10^{-23} \) Joules per Kelvin (J/K)

Electronic charge \( (e) = 1.602176487(40) \times 10^{-19} \) Coulomb (C)

Faraday constant \( (F) = 9.64853399(24) \times 10^{4} \) Coulombs per mole (C/mol)

Magnetic permeability of free space \( (\mu_0) = 1.25663706212(19) \times 10^{-6} \) Henrys per meter (H/m)

Electric permittivity of free space \( (\epsilon_0) = 8.8541878128(13) \times 10^{-12} \) Farads per meter (F/m)

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

Gravitational constant \( (G) = 6.67428(67) \times 10^{-11} \) cubic meters per kilogram-seconds squared (m\(^3\)/kg-s\(^2\))

Molar gas constant \( (R) = 8.314472(15) \) Joules per mole-Kelvin (J/mol-K) = 0.08205746(14) liters-atmospheres per mole-Kelvin

Planck constant \( (h) = 6.62606896(33) \times 10^{-34} \) joule-seconds (J-s)

Stefan-Boltzmann constant \( (\sigma) = 5.670400(40) \times 10^{-8} \) Watts per square meter-Kelvin\(^4\) (W/m\(^2\)-K\(^4\))

Speed of light in a vacuum \( (c) = 299792458 \text{ meters per second} \) (m/s) = 186282.4 miles per second (mi/s)

Note: All constants taken from NIST data “Fundamental Physical Constants – Extensive Listing”, from [http://physics.nist.gov/constants](http://physics.nist.gov/constants), National Institute of Standards and Technology (NIST), 2006; with the exception of the permeability of free space which was taken from NIST’s 2018 CODATA recommended values database.
6.2. QUANTITATIVE REASONING

6.2.2 Introduction to spreadsheets

A powerful computational tool you are encouraged to use in your work is a spreadsheet. Available on most personal computers (e.g. Microsoft Excel), spreadsheet software performs numerical calculations based on number values and formulae entered into cells of a grid. This grid is typically arranged as lettered columns and numbered rows, with each cell of the grid identified by its column/row coordinates (e.g. cell B3, cell A8). Each cell may contain a string of text, a number value, or a mathematical formula. The spreadsheet automatically updates the results of all mathematical formulae whenever the entered number values are changed. This means it is possible to set up a spreadsheet to perform a series of calculations on entered data, and those calculations will be re-done by the computer any time the data points are edited in any way.

For example, the following spreadsheet calculates average speed based on entered values of distance traveled and time elapsed:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Distance traveled</td>
<td>46.9</td>
<td>Kilometers</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Time elapsed</td>
<td>1.18</td>
<td>Hours</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Average speed</td>
<td>= B1 / B2</td>
<td>km/h</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Text labels contained in cells A1 through A3 and cells C1 through C3 exist solely for readability and are not involved in any calculations. Cell B1 contains a sample distance value while cell B2 contains a sample time value. The formula for computing speed is contained in cell B3. Note how this formula begins with an “equals” symbol (=), references the values for distance and speed by lettered column and numbered row coordinates (B1 and B2), and uses a forward slash symbol for division (/). The coordinates B1 and B2 function as variables\(^6\) would in an algebraic formula.

When this spreadsheet is executed, the numerical value 39.74576 will appear in cell B3 rather than the formula = B1 / B2, because 39.74576 is the computed speed value given 46.9 kilometers traveled over a period of 1.18 hours. If a different numerical value for distance is entered into cell B1 or a different value for time is entered into cell B2, cell B3’s value will automatically update. All you need to do is set up the given values and any formulae into the spreadsheet, and the computer will do all the calculations for you.

Cell B3 may be referenced by other formulae in the spreadsheet if desired, since it is a variable just like the given values contained in B1 and B2. This means it is possible to set up an entire chain of calculations, one dependent on the result of another, in order to arrive at a final value. The arrangement of the given data and formulae need not follow any pattern on the grid, which means you may place them anywhere.

---

\(^6\)Spreadsheets may also provide means to attach text labels to cells for use as variable names (Microsoft Excel simply calls these labels “names”), but for simple spreadsheets such as those shown here it’s usually easier just to use the standard coordinate naming for each cell.
Common\textsuperscript{7} arithmetic operations available for your use in a spreadsheet include the following:

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Powers (^)
- Square roots (sqrt())
- Logarithms (ln(), log10())

Parentheses may be used to ensure\textsuperscript{8} proper order of operations within a complex formula. Consider this example of a spreadsheet implementing the quadratic formula, used to solve for roots of a polynomial expression in the form of $ax^2 + bx + c$:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x_1 = (-B4 + \text{sqrt}((B4^2) - (4<em>B3</em>B5))) / (2*B3)$</td>
</tr>
<tr>
<td>2</td>
<td>$x_2 = (-B4 - \text{sqrt}((B4^2) - (4<em>B3</em>B5))) / (2*B3)$</td>
</tr>
<tr>
<td>3</td>
<td>$a = $</td>
</tr>
<tr>
<td>4</td>
<td>$b = 9$</td>
</tr>
<tr>
<td>5</td>
<td>$c = -2$</td>
</tr>
</tbody>
</table>

This example is configured to compute roots\textsuperscript{9} of the polynomial $9x^2 + 5x - 2$ because the values of 9, 5, and $-2$ have been inserted into cells B3, B4, and B5, respectively. Once this spreadsheet has been built, though, it may be used to calculate the roots of any second-degree polynomial expression simply by entering the new $a$, $b$, and $c$ coefficients into cells B3 through B5. The numerical values appearing in cells B1 and B2 will be automatically updated by the computer immediately following any changes made to the coefficients.

\textsuperscript{7}Modern spreadsheet software offers a bewildering array of mathematical functions you may use in your computations. I recommend you consult the documentation for your particular spreadsheet for information on operations other than those listed here.

\textsuperscript{8}Spreadsheet programs, like text-based programming languages, are designed to follow standard order of operations by default. However, my personal preference is to use parentheses even where strictly unnecessary just to make it clear to any other person viewing the formula what the intended order of operations is.

\textsuperscript{9}Reviewing some algebra here, a \textit{root} is a value for $x$ that yields an overall value of zero for the polynomial. For this polynomial ($9x^2 + 5x - 2$) the two roots happen to be $x = 0.269381$ and $x = -0.82494$, with these values displayed in cells B1 and B2, respectively upon execution of the spreadsheet.
6.2. QUANTITATIVE REASONING

Alternatively, one could break up the long quadratic formula into smaller pieces like this:

\[
y = \sqrt{b^2 - 4ac} \quad z = 2a
\]

\[
x = \frac{-b \pm y}{z}
\]

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x_1</td>
<td>(-B4 + C1) / C2</td>
<td>sqrt((B4^2) - (4<em>B3</em>B5))</td>
</tr>
<tr>
<td>2</td>
<td>x_2</td>
<td>(-B4 - C1) / C2</td>
<td>2*B3</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>b</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>c</td>
<td>-2</td>
<td></td>
</tr>
</tbody>
</table>

Note how the square-root term \(y\) is calculated in cell C1, and the denominator term \(z\) in cell C2. This makes the two final formulae (in cells B1 and B2) simpler to interpret. The positioning of all these cells on the grid is completely arbitrary\(^\text{10}\) – all that matters is that they properly reference each other in the formulae.

Spreadsheets are particularly useful for situations where the same set of calculations representing a circuit or other system must be repeated for different initial conditions. The power of a spreadsheet is that it automates what would otherwise be a tedious set of calculations. One specific application of this is to simulate the effects of various components within a circuit failing with abnormal values (e.g. a shorted resistor simulated by making its value nearly zero; an open resistor simulated by making its value extremely large). Another application is analyzing the behavior of a circuit design given new components that are out of specification, and/or aging components experiencing drift over time.

\(^{10}\) My personal preference is to locate all the “given” data in the upper-left cells of the spreadsheet grid (each data point flanked by a sensible name in the cell to the left and units of measurement in the cell to the right as illustrated in the first distance/time spreadsheet example), sometimes coloring them in order to clearly distinguish which cells contain entered data versus which cells contain computed results from formulae. I like to place all formulae in cells below the given data, and try to arrange them in logical order so that anyone examining my spreadsheet will be able to figure out how I constructed a solution. This is a general principle I believe all computer programmers should follow: document and arrange your code to make it easy for other people to learn from it.
6.2.3 Air capacitor gap

How far away from each other would two metal plates, 2 square meters in area each, have to be in order to create a capacitance of $1\,\mu\text{F}$? Assume that the plates are separated by air.

**Challenges**

- A common mistake is to calculate a value of 2 million meters ($2\times10^6$ meters). Explain how someone might make this mistake.

6.2.4 Defibrillator capacitor

A defibrillator is a medical device used to deliver an electric shock to a patient suffering from cardiac fibrillation, which is an ineffectual heart rhythm. The amount of electrical energy necessary to shock a person’s heart back to normal rhythm varies with body weight and a variety of other factors, but typically ranges from 70 to 360 Joules\(^{11}\).

Suppose you are part of a team designing a new model of defibrillator, using a capacitor to store the energy delivered to the patient. Assuming the capacitor will reach a voltage of 1200 Volts when fully energized, how large of a capacitor will be necessary for this design?

Suppose the design of this new defibrillator varies the amount of delivered energy by controlling the voltage value of the energized capacitor. How many Volts would the capacitor need to be energized to in order to store the 90 Joules of energy?

**Challenges**

- This illustration should be enough to convince you that energized capacitors can be very dangerous. Explain how you could test a capacitor for hazardous levels of energy before handling it with your bare hands.

---

\(^{11}\)If defibrillation is done to a patient during chest surgery, where the shock may be administered directly to the heart muscle and not through the chest, this energy figure is significantly lower!
6.2.5 Capacitor labeling

Identify the ratings of the following capacitors:

- 4700 MFD 36 V
- 25 V
- 103 16 V
- 1.0 25 V
- 4.7 16 V
- 1 kV
- 33 100 V
- 0.33 100 V
- 100 V
- 47 50 V
- 1000 MFD

Challenges

- Explain why the voltage rating of a capacitor is important. What do you suppose might happen if this rating is exceeded?
6.2.6 Half-life

When a radioactive substance decays, it does so in a way that is mathematically identical to that of a capacitor de-energizing as it powers a resistor: the quantity begins at a high level, falling at a fast rate, with the rate slowing down as the quantity approaches zero. The amount of time required for a radioactive sample to decay to a level where just one-half of its quantity remains is called the half-life of that substance. For example, 16 grams of a radioactive substance having a half-life of 3 minutes will decay to 8 grams after the first three minutes, 4 grams after the next three minutes, 2 grams after the next three minutes, and so on.

Demonstrate how the mathematical decay of voltage for a de-energizing capacitor may be measured in terms of half-life, instead of measuring that decay in terms of time constant ($\tau$).

**Challenges**

- A simple way to demonstrate half-life is to flip a large number of coins, removing from the group any coin(s) that come up “heads” rather than “tails”. Since the probability of either outcome is one-half (0.5), this means that the population of the coin group should decay by one-half with every flip. Perform an experiment to demonstrate this principle, and observe how closely the outcome matches theory.

6.2.7 Number of time constants

Determine the number of time constants ($\tau$) that 7.5 seconds is equal to in each of the following resistor-capacitor circuits:

- $R = 10 \, \text{kΩ}, \, C = 220 \, \mu\text{F} \; ; \; 7.5 \, \text{sec} =$
- $R = 33 \, \text{kΩ}, \, C = 470 \, \mu\text{F} \; ; \; 7.5 \, \text{sec} =$
- $R = 1.5 \, \text{kΩ}, \, C = 100 \, \mu\text{F} \; ; \; 7.5 \, \text{sec} =$
- $R = 790 \, \Omega, \, C = 9240 \, \text{nF} \; ; \; 7.5 \, \text{sec} =$
- $R = 100 \, \text{kΩ}, \, C = 33 \, \text{pF} \; ; \; 7.5 \, \text{sec} =$

**Challenges**

- For each of these RC networks, calculate the percentage of decay for voltage or current in a de-energizing condition following 7.5 seconds after disconnection of the source.
6.2. QUANTITATIVE REASONING

6.2.8 Inverse exponential functions

The following two expressions are frequently used to calculate values of changing variables (voltage and current) in RC and LR timing circuits:

\[ e^{-\frac{t}{\tau}} \quad \text{or} \quad 1 - e^{-\frac{t}{\tau}} \]

One of these expressions describes the percentage that a changing value in an RC or LR circuit has gone from the starting time. The other expression describes how far that same variable has left to go before it reaches its ultimate value (at \( t = \infty \)).

The question is, which expression represents which quantity? This is often a point of confusion, because students have a tendency to try to correlate these expressions to the quantities by rote memorization. Does the expression \( e^{-\frac{t}{\tau}} \) represent the amount a variable has changed, or how far it has left to go until it stabilizes? What about the other expression \( 1 - e^{-\frac{t}{\tau}} \)? More importantly, how can we figure this out so we don’t have to rely on memory?

**Increasing variable**

<table>
<thead>
<tr>
<th>Time</th>
<th>Initial</th>
<th>Voltage or Current</th>
<th>Percentage changed from initial value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Decreasing variable**

<table>
<thead>
<tr>
<th>Time</th>
<th>Initial</th>
<th>Voltage or Current</th>
<th>Percentage left to change before reaching final value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Challenges**

- A useful mathematical problem-solving technique is to set the independent variable in this case, \( t \), to some simple value such as 0 or 1 to see what happens. What would setting \( t = 0 \) simulate in either of these functions?
6.2.9 Current versus rate-of-voltage-change

A 470 $\mu$F capacitor is subjected to an applied voltage that changes at a rate of 200 Volts per second. How much current will there be “through” this capacitor?

Now suppose two 470 $\mu$F capacitors are connected to form a series network, and the whole network subjected to the same rate-of-change of voltage. How much current would you expect now?

Now suppose two 470 $\mu$F capacitors are connected to form a parallel network, and the whole network subjected to the same rate-of-change of voltage. How much current would you expect now?

Also, explain why I placed quotation marks around the word “through” in the previous sentence. Why can’t this word be used in its fullest sense when describing electric current in a capacitor circuit?

**Challenges**

- Based on the currents calculated for each capacitor network, prove that $C$ adds in parallel and diminishes in series.
6.2.10 Capacitance of three-capacitor network

Calculate the total capacitance in this collection of capacitors, as measured between the two wires:

\[
\text{C}_{\text{total}} = 33 \text{ pF} + 100 \text{ pF} + 10 \mu\text{F} = ??? 
\]

**Challenges**

- Suppose one of the terminal block’s hold-down screws were to come loose on one of the leads for the middle capacitor, making a bad (open) connection. What effect would this have on the total capacitance?

6.2.11 Resistance for specified time constant

Calculate the resistor value which when connected in series with a 33 µF capacitor will provide a time constant (\(\tau\)) of 10 seconds. Express your answer in the form of a five-band precision resistor color code (with a tolerance of +/- 0.1%).

**Challenges**

- What advantage does a five-band resistor color code enjoy over a four-band?
6.2.12 Safely de-energizing a large capacitor

An electronic service technician prepares to work on a high-voltage power supply circuit containing one large capacitor. On the side of this capacitor are the following specifications:

\[ 3000 \text{ WVDC} \quad 0.75\mu\text{F} \]

Obviously this device poses a certain amount of danger, even with the AC line power secured (lock-out/tag-out). Discharging this capacitor by directly shorting its terminals with a screwdriver or some other piece of metal might be dangerous due to the quantity of the stored charge. What needs to be done is to discharge this capacitor at a modest rate.

The technician realizes that she can discharge the capacitor at any rate desired by connecting a resistor in parallel with it (holding the resistor with electrically-insulated pliers, of course, to avoid having to make bodily contact with either capacitor terminal). What size resistor should she use, if she wants to discharge the capacitor to less than 1% charge in 15 seconds? State your answer using the standard 4-band resistor color code (tolerance = +/- 10%).

**Challenges**

- Identify some hazards that might result from accidentally choosing a resistor too small.
- Identify some hazards that might result from accidentally choosing a resistor too large.

6.2.13 Capacitor voltage at specified intervals

Suppose a capacitor is charged to a voltage of exactly 100 Volts, then connected to a resistor so it discharges slowly. Calculate the amount of voltage remaining across the capacitor terminals at the following points in time:

- 1 time constant (\( \tau \)) after connecting the resistor:
- 2 time constants (2\( \tau \)) after connecting the resistor:
- 3 time constants (3\( \tau \)) after connecting the resistor:
- 4 time constants (4\( \tau \)) after connecting the resistor:
- 5 time constants (5\( \tau \)) after connecting the resistor:

**Challenges**

- Write an equation solving for these voltages at the specified times.
6.2.14 Partial capacitor de-energization

Calculate the voltage across a 470 µF capacitor after discharging through a 10 kΩ resistor for 9 seconds, if the capacitor’s original voltage (at $t = 0$) was 24 Volts.

Also, express this amount of time (9 seconds) in terms of how many time constants have elapsed.

**Challenges**

- If the resistance was 15 kΩ instead of 10 kΩ, how would the voltage after 9 seconds be affected?

6.2.15 Rapid charge and slow discharge

The following circuit allows a capacitor to be rapidly charged and slowly discharged:

Suppose that the switch was left in the “charge” position for some substantial amount of time. Then, someone moves the switch to the “discharge” position to let the capacitor discharge. Calculate the amount of capacitor voltage and capacitor current at exactly 3 seconds after moving the switch to the “discharge” position.

\[
V_C = \text{________}_\text{ at } t = 3 \text{ seconds}
\]

\[
I_C = \text{________}_\text{ at } t = 3 \text{ seconds}
\]

Also, show the direction of discharge current in this circuit.

**Challenges**

- How would the voltage and current values be affected if this circuit had two 500 µF capacitors in parallel with each other instead of just the one in the circuit?

- How would the voltage and current values be affected if this circuit had two 500 µF capacitors in series with each other instead of just the one in the circuit?
6.2.16 Time required to reach specified voltages

Calculate the amount of time it takes for a 33 µF capacitor to charge from 0 Volts to 20 Volts, if powered by a 24 Volt battery through a 10 kΩ resistor.

Calculate the amount of time it takes for a 10 µF capacitor to discharge from 18 Volts to 7 Volts if its ultimate (final) voltage when fully discharged will be 0 Volts, and it is discharging through a 22 kΩ resistor.

A 470 µF capacitor begins in a charged state of 270 Volts, and discharges through a 100 kΩ resistor. How long will it take before the capacitor’s voltage will fall to a relatively safe value (30 Volts or less)?

Determine the amount of time needed for the capacitor voltage (V_C) to fall to the specified levels after the switch is thrown to the “discharge” position, assuming it had first been charged to full battery voltage:

- 10 Volts =
- 8 Volts =
- 6 Volts =
- 4 Volts =
- 2 Volts =

Challenges

- What will happen to all of these time values if the resistor in each circuit is increased in value?
6.2.17 Measuring capacitance by time delay

A capacitor of unknown value is connected to a precision 5 Volts DC power supply and then allowed to discharge through a 3.3 kΩ resistor. The following image shows the decay of voltage over time:

![Time-voltage graph]

Calculate the approximate size of the capacitor in microFarads based on the information you see here.

Challenges

- A common tendency among students new to electric circuit analysis is to first “think mathematically” rather than first “think conceptually”. Before plugging any numbers into any formulae, one should always identify electrical principles such as sources and loads, directions of current, and voltage polarities, usually sketching a diagram of a circuit to see where all these
concepts fit. Demonstrate how to “think conceptually” when solving this problem, rather than immediately trying to compute values.

• How would the plot of voltage over time change if the resistor were made smaller?

• What happened to the energy stored in the capacitor when it was energized to a potential of 5 Volts?
6.2.18 Intervalometer

An intervalometer is a device that measures the interval of time between two events. Such devices are commonly used to measure the speed of projectiles, given a known distance between two sensors:

A crude intervalometer may be constructed using two thin wires as sensors, which are broken by the passage of the projectile. The two wires are connected in an RC circuit as such:

In order for this circuit to function properly as an intervalometer, which wire does the projectile need to break first? Explain why. Also, the voltmeter used in this instrument must be one with as high an input resistance as possible for best accuracy. Explain why this is necessary as well.

Which will produce a greater voltage indication after the test, a fast projectile or a slow projectile? Explain your answer.

**Challenges**

- The voltmeter itself introduces a source of error in this intervalometer. Identify the error, and determine if this error makes the intervalometer register falsely high or falsely low with regard to projectile speed.
6.2.19 SPICE analysis of an energizing capacitor

The following resistor-capacitor circuit will be simulated by a computer program called SPICE. The text-based “netlist” used to describe the circuit to SPICE is shown immediately to the right of the schematic diagram. Numbered points on the diagram identify “nodes” used by SPICE to declare how the components connect to one another:

![Schematic diagram of the circuit]

In this analysis, SPICE will plot capacitor voltage (between node 2 and ground) over a timespan of 15 seconds in increments of 1 second each. The capacitor begins (i.e. “initial condition” ic=) with zero voltage across it, which means no stored energy. When run, the simulation results are as follows:

Legend:  + = v(2)

<table>
<thead>
<tr>
<th>time (s)</th>
<th>v(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000e+00</td>
<td>0.000e+00</td>
</tr>
<tr>
<td>1.000e+00</td>
<td>4.708e+00</td>
</tr>
<tr>
<td>2.000e+00</td>
<td>8.178e+00</td>
</tr>
<tr>
<td>3.000e+00</td>
<td>1.075e+01</td>
</tr>
<tr>
<td>4.000e+00</td>
<td>1.264e+01</td>
</tr>
<tr>
<td>5.000e+00</td>
<td>1.405e+01</td>
</tr>
<tr>
<td>6.000e+00</td>
<td>1.508e+01</td>
</tr>
<tr>
<td>7.000e+00</td>
<td>1.584e+01</td>
</tr>
<tr>
<td>8.000e+00</td>
<td>1.641e+01</td>
</tr>
<tr>
<td>9.000e+00</td>
<td>1.682e+01</td>
</tr>
<tr>
<td>1.000e+01</td>
<td>1.713e+01</td>
</tr>
<tr>
<td>1.100e+01</td>
<td>1.736e+01</td>
</tr>
<tr>
<td>1.200e+01</td>
<td>1.753e+01</td>
</tr>
<tr>
<td>1.300e+01</td>
<td>1.765e+01</td>
</tr>
<tr>
<td>1.400e+01</td>
<td>1.774e+01</td>
</tr>
<tr>
<td>1.500e+01</td>
<td>1.781e+01</td>
</tr>
</tbody>
</table>

Verify any of the simulated capacitor voltage values using your own manual calculations.
• Explain how you may use a SPICE simulation such as this for your own practice, instead of relying on questions and answers provided to you in a textbook.

• Explain how you could use this simulation to practice calculating capacitor or resistor values from the simulated results. In other words, imagine this was real data from a real capacitor-resistor circuit experiment, and you needed to compute either the resistor value or the capacitor value from all the other data.
6.2.20 SPICE analysis of a de-energizing capacitor

The following resistor-capacitor circuit will be simulated by a computer program called SPICE. The text-based “netlist” used to describe the circuit to SPICE is shown immediately to the right of the schematic diagram. Numbered points on the diagram identify “nodes” used by SPICE to declare how the components connect to one another:

```
* SPICE circuit
vamm1 1 0
r1 1 2 830
cl 2 0 10e-6 ic=11
.tran 1.5e-3 21e-3 uic
.width out=80
.plot tran i(vamm1)
.end
```

In this analysis, SPICE will plot circuit current (measured using the “dummy” voltage source $V_{amm1}$) over a timespan of 21 milliseconds in increments of 1.5 milliseconds each. The capacitor begins (i.e. “initial condition” $ic=11$) with 11 Volts, with no other energy source in the circuit. When run, the simulation results are as follows:

**Legend:** + = vamm1#branch

<table>
<thead>
<tr>
<th>time</th>
<th>vamm1#br</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00e+00</td>
<td>0.000e+00</td>
</tr>
<tr>
<td>1.500e-03</td>
<td>1.106e-02</td>
</tr>
<tr>
<td>3.000e-03</td>
<td>9.233e-03</td>
</tr>
<tr>
<td>4.500e-03</td>
<td>7.708e-03</td>
</tr>
<tr>
<td>6.000e-03</td>
<td>6.433e-03</td>
</tr>
<tr>
<td>7.500e-03</td>
<td>5.369e-03</td>
</tr>
<tr>
<td>9.000e-03</td>
<td>4.482e-03</td>
</tr>
<tr>
<td>1.050e-02</td>
<td>3.740e-03</td>
</tr>
<tr>
<td>1.200e-02</td>
<td>3.122e-03</td>
</tr>
<tr>
<td>1.350e-02</td>
<td>2.605e-03</td>
</tr>
<tr>
<td>1.500e-02</td>
<td>2.175e-03</td>
</tr>
<tr>
<td>1.650e-02</td>
<td>1.815e-03</td>
</tr>
<tr>
<td>1.800e-02</td>
<td>1.515e-03</td>
</tr>
<tr>
<td>1.950e-02</td>
<td>1.264e-03</td>
</tr>
<tr>
<td>2.100e-02</td>
<td>1.055e-03</td>
</tr>
</tbody>
</table>

Verify any of the simulated current values using your own manual calculations.

Challenges
6.2. QUANTITATIVE REASONING

- Explain how you may use a SPICE simulation such as this for your own practice, instead of relying on questions and answers provided to you in a textbook.

- Explain how you could use this simulation to practice calculating capacitor or resistor values from the simulated results. In other words, imagine this was real data from a real capacitor-resistor circuit experiment, and you needed to compute either the resistor value or the capacitor value from all the other data.
6.3 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.

6.3.1 Dielectric breakdown

A common mode of capacitor failure is dielectric breakdown, where the insulating medium (i.e. the "dielectric") material between a capacitor’s plates becomes compromised and is no longer an effective insulator.

Devise a simple method to test a capacitor for this fault.

Challenges

• Which do you think is more likely to cause dielectric breakdown in a capacitor, excessive voltage or excessive current?
6.3.2 Incorrect voltage/current curves

Explain what is wrong with these voltage and current plots for a capacitor energized by a constant-voltage source through a series resistance:

Furthermore, try to explain why someone might make this mistake when predicting capacitor voltage and current.

**Challenges**

- Devise a “thought experiment” whereby this error would become obviously absurd (i.e. *reductio ad absurdum*).
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

“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.

The following principles inform the instructional and assessment philosophies embodied in these learning modules:

• 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 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 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 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

---

1In 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.

2Socrates 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.

3This 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 characteristic of these learning modules is a lack of step-by-step instructions in the Project and Experiment chapters. Unlike many modern workbooks and laboratory guides where step-by-step instructions are prescribed for each experiment, these modules take the approach that students must learn to closely read the tutorials and apply their own reasoning to identify the appropriate experimental steps. Sometimes these steps are plainly declared in the text, just not as a set of enumerated points. At other times certain steps are implied, an example being assumed competence in test equipment use where the student should not need to be told *again* how to use their multimeter because that was thoroughly explained in previous lessons. In some circumstances no steps are given at all, leaving the entire procedure up to the student.

This lack of prescription is not a flaw, but rather a feature. Close reading and clear thinking are foundational principles of this learning series, and in keeping with this philosophy all activities are designed to *require* those behaviors. Some students may find the lack of prescription frustrating, because it demands more from them than what their previous educational experiences required. This frustration should be interpreted as an unfamiliarity with autonomous thinking, a problem which must be corrected if the student is ever to become a self-directed learner and effective problem-solver. Ultimately, the need for students to read closely and think clearly is more important both in the near-term and far-term than any specific facet of the subject matter at hand. If a student takes longer than expected to complete a module because they are forced to outline, digest, and reason on their own, so be it. The future gains enjoyed by developing this mental discipline will be well worth the additional effort and delay.

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.

This methodology of assigning text-based modules to students for digestion and then using Socratic dialogue to assess progress and hone students’ thinking was developed over a period of several years by the author with his Electronics and Instrumentation students at the two-year college level. While decidedly unconventional and sometimes even unsettling for students accustomed to a more passive lecture environment, this instructional philosophy has proven its ability to convey conceptual mastery, foster careful analysis, and enhance employability so much better than lecture that the author refuses to ever teach by lecture again.

Problems which often go undiagnosed in a lecture environment are laid bare in this “inverted” format where students must articulate and logically defend their reasoning. This, too, may be unsettling for students accustomed to lecture sessions where the instructor cannot tell for sure who comprehends and who does not, and this vulnerability necessitates sensitivity on the part of the “inverted” session instructor in order that students never feel discouraged by having their errors exposed. *Everyone* makes mistakes from time to time, and learning is a lifelong process! Part of the instructor’s job is to build a culture of learning among the students where errors are not seen as shameful, but rather as opportunities for progress.
To this end, instructors managing courses based on these modules should adhere to the following principles:

- Student questions are always welcome and demand thorough, honest answers. The only type of question an instructor should refuse to answer is one the student should be able to easily answer on their own. Remember, the fundamental goal of education is for each student to learn to think clearly and independently. This requires hard work on the part of the student, which no instructor should ever circumvent. Anything done to bypass the student’s responsibility to do that hard work ultimately limits that student’s potential and thereby does real harm.

- It is not only permissible, but encouraged, to answer a student’s question by asking questions in return, these follow-up questions designed to guide the student to reach a correct answer through their own reasoning.

- All student answers demand to be challenged by the instructor and/or by other students. This includes both correct and incorrect answers – the goal is to practice the articulation and defense of one’s own reasoning.

- No reading assignment is deemed complete unless and until the student demonstrates their ability to accurately summarize the major points in their own terms. Recitation of the original text is unacceptable. This is why every module contains an “Outline and reflections” question as well as a “Foundational concepts” question in the Conceptual reasoning section, to prompt reflective reading.

- No assigned question is deemed answered unless and until the student demonstrates their ability to consistently and correctly apply the concepts to variations of that question. This is why module questions typically contain multiple “Challenges” suggesting different applications of the concept(s) as well as variations on the same theme(s). Instructors are encouraged to devise as many of their own “Challenges” as they are able, in order to have a multitude of ways ready to probe students’ understanding.

- No assigned experiment or project is deemed complete unless and until the student demonstrates the task in action. If this cannot be done “live” before the instructor, video-recordings showing the demonstration are acceptable. All relevant safety precautions must be followed, all test equipment must be used correctly, and the student must be able to properly explain all results. The student must also successfully answer all Challenges presented by the instructor for that experiment or project.
Students learning from these modules would do well to abide by the following principles:

- No text should be considered fully and adequately read unless and until you can express every idea in your own words, using your own examples.

- You should always articulate your thoughts as you read the text, noting points of agreement, confusion, and epiphanies. Feel free to print the text on paper and then write your notes in the margins. Alternatively, keep a journal for your own reflections as you read. This is truly a helpful tool when digesting complicated concepts.

- Never take the easy path of highlighting or underlining important text. Instead, summarize and/or comment on the text using your own words. This actively engages your mind, allowing you to more clearly perceive points of confusion or misunderstanding on your own.

- A very helpful strategy when learning new concepts is to place yourself in the role of a teacher, if only as a mental exercise. Either explain what you have recently learned to someone else, or at least imagine yourself explaining what you have learned to someone else. The simple act of having to articulate new knowledge and skill forces you to take on a different perspective, and will help reveal weaknesses in your understanding.

- Perform each and every mathematical calculation and thought experiment shown in the text on your own, referring back to the text to see that your results agree. This may seem trivial and unnecessary, but it is critically important to ensuring you actually understand what is presented, especially when the concepts at hand are complicated and easy to misunderstand. Apply this same strategy to become proficient in the use of circuit simulation software, checking to see if your simulated results agree with the results shown in the text.

- 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\(^4\) effort, and never give up!

Students interested in using these modules for self-study will also find them beneficial, although the onus of responsibility for thoroughly reading and answering questions will of course lie with that individual alone. If a qualified instructor is not available to challenge students, a workable alternative is for students to form study groups where they challenge\(^5\) one another.

To high standards of education,

Tony R. Kuphaldt

---

\(^4\)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.

\(^5\)Avoid the temptation to simply share answers with study partners, as this is really counter-productive to learning. Always bear in mind that the answer to any question is far less important in the long run than the method(s) used to obtain that answer. The goal of education is to empower one’s life through the improvement of clear and independent thought, literacy, expression, and various practical skills.
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 pixelate 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 **T \text{E}X** 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.
APPENDIX C. TOOLS USED

**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 theLicensed 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


“How to select the proper capacitor for your design?”, document 2018-02-22 SIM, Würth Elektronik, Niedernhall, Germany, 2018.

“Impedance, Dissipation Factor and ESR”, Illinois Capacitor, Inc., Lincolnwood, IL.


“Technical Notes For Electrolytic Capacitor”, Rubycon Corporation.
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.

13 June 2022 – added a new Case Tutorial section showing energy storage calculations for series versus parallel capacitor networks.

22 April 2022 – added photographs of a variable capacitor showing different degrees of plate overlap.

31 March 2022 – added “Electrical field quantities” to the Derivations and Technical References chapter.

16 March 2022 – added some content to the Tutorial discussing the calculation of capacitance based on permittivity, plate area, and separation distance.


3 December 2021 – changed “permeability” to “permittivity”, which was a copy-and-paste error from a module written for inductors.

20-21 October 2021 – deleted the Projects and Experiments chapter. Also added some clarifying text in the Tutorial about how the Conservation of Energy relates to voltage and current for these devices.

1 August 2021 – added Case Tutorial on the use of decoupling capacitors to stabilize IC power supply voltage.

23 July 2021 – added a Case Tutorial section showing a simple charge/discharge circuit useful for exploring the inverse-exponential behavior of resistor-capacitor networks. Also added foundational concept review to the “Capacitor charging circuit” Conceptual Reasoning question.
16 July 2021 – added footnote to the Tutorial explaining what permittivity is.

9 July 2021 – improved description of electrolytic capacitor manufacturing found in the Tutorial chapter.

4 July 2021 – divided Tutorial chapter into sections and made some minor edits to the text for clarity.

2 June 2021 – added a Case Tutorial chapter with sections showing how to store and release energy from capacitors.

8 May 2021 – commented out or deleted empty chapters.

13 April 2021 – added “energy” as a parameter for students to qualitatively assess in the “Capacitor charging circuit” Conceptual Reasoning question.

6 April 2021 – added some commentary to Tutorial on units of measurement for time constant ($\tau$).

5 April 2021 – added some commentary to the “Capacitor parameters and dielectric types” Technical Reference on dissipative losses for capacitors.

18 March 2021 – corrected one instance of “volts” that should have been capitalized “Volts”.

9 February 2021 – added References for capacitor labeling source documents, to accompany techref_caplabel.


19 December 2020 – added a Case Tutorial example showing the use of a capacitor to stabilize power supply voltage for battery-sourced circuits.

26 October 2020 – added Quantitative questions.

19 October 2020 – corrected typo in the “SPICE analysis of an energizing capacitor” Quantitative question, saying that a de-energized capacitor has zero current when it should have said that an initial condition of zero meant zero voltage.

26 September 2020 – added voltage divider RC circuit to the Case Tutorial chapter.

4 September 2020 – incorporated the “Inverse exponential functions” question content into the Tutorial, explaining the formulae for decaying and growing quantities in full detail.

29 August 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. Also, added brief comment and footnote in the Tutorial on parasitic inductance.

26 August 2020 – added some Case Tutorial examples showing circuits energized by voltage and
current sources, respectively.

17 August 2020 – minor edit to the Tutorial text in order to shorten a paragraph, for typesetting purposes.

17 April 2020 – added a Technical Reference section on IEC standard component values (file techref_IEC60063.latex).

13 April 2020 – added Technical Reference section on capacitor types.

6 April 2020 – minor typographical error corrections, courtesy of Ron Felix.


4 February 2020 – added some Quantitative Reasoning problems.

2 January 2020 – removed from from C++ code execution output, to clearly distinguish it from the source code listing which is still framed.

1 January 2020 – added explanatory text for the C++ code in reference_invexp_decay.latex. Also changed main () to main (void) in another C++ programming example.

10 December 2019 – added more questions. Also, corrected error in “Cable capacitance” question. Added letter-code tolerance ratings to Tutorial.

2 November 2019 – added more questions.

26 August 2019 – added mention of difficult concepts to the Introduction.

13 April 2019 – improved the explanation of what happens in RC circuits with inverse-exponential trends of voltage and current. Also introduced the term “speed” to refer to rate-of-change for capacitor voltage (i.e. voltage “speed” = \( \frac{dV}{dt} \)).

8 April 2019 – minor edit to wording within an illustration (replaced “creates” with “means” to avoid the implication of a unidirectional causality).

24 March 2019 – added a Quantitative Reasoning questions.

13 March 2019 – added a new experiment, demonstrating capacitive time delays, and also added reference to inverse exponential functions in the Tutorial.

12 February 2019 – added a number of new entries to the Foundational Concepts subsection.

7 January 2019 – added a challenge question to a quantitative problem. Also, added illustrative examples to the Tutorial, showing what happens when a capacitor is energized by a constant current. Finally, added a Derivations and Technical References chapter, with an exposition of exponential decay functions in it.
17 December 2018 – added an example C++ program showing calculations for an energizing resistor-capacitor circuit.

October 2018 – clarified a number of details, with constructive criticism from Ron Felix.

August 2018 – wrote an Introduction where none existed before.

July 2018 – document first created.
Index

$V_{DD}$ bounce, 14
$\tau$, 57
“Ohm’s Law” for a capacitor, 25

Adding quantities to a qualitative problem, 108
Analog computer, 29
Annotating diagrams, 107
Asymptote, 57

Battery action, 10
Bounce, ground, 14
Breakdown, dielectric, 45

C++, 48, 58
Capacitance, 3, 21
Capacitor, 3
Capacitor, decoupling, 14
Checking for exceptions, 108
Checking your work, 108
Chip, 27
Code, computer, 115
Color code, resistor, 31
Compiler, C++, 48
Component values, IEC standard, 34
Computer programming, 47, 58
Computer, analog, 29
Conservation of Energy, 24
Conservation of Mass, 24
Coulomb, 45, 46

Decay function, 29
Decay, exponential, 57
Decoupling, 14
Decoupling capacitor, 14
Dielectric, 3, 22, 30
Dielectric breakdown, 45
Dielectric relaxation, 10
Dielectric strength, 30, 45

Dimensional analysis, 107
Edwards, Tim, 116
Electric field, 21, 57
Energy, kinetic, 22
Energy, potential, 21, 22
Equivalent Series Resistance, 31, 42
Equivalent series resistance, 14, 16
ESR, 14, 16, 31, 42
Exponential decay, 57

Farad, unit, 3, 22
Field, electric, 21, 57
Field, gravitational, 21
Field, magnetic, 57
Flywheel, 27

Graph values to solve a problem, 108
Gravitational field, 21
Greenleaf, Cynthia, 67
Ground bounce, 14

How to teach with these modules, 110
Hwang, Andrew D., 117

IC, 27
Identify given data, 107
Identify relevant principles, 107
IEC 60063 standard, 34
IEC standard component values, 34
Inductance, 14, 31
Instructions for projects and experiments, 111
Integrated circuit, 27
Intermediate results, 107
Interpreter, Python, 52
Inverse exponential curve, 28
Inverse exponential decay, 57
Inverted instruction, 110

133
Java, 49, 59
Joule, unit, 22
Kirchhoff’s Voltage Law, 28
Knuth, Donald, 116
Lamport, Leslie, 116
Leakage resistance, 31
Limiting cases, 108
Load, 24
Magnetic field, 57
Metacognition, 72
Moolenaar, Bram, 115
Murphy, Lynn, 67
Newton, 45, 46
Newton’s Cooling Law, 58
Ohm’s Law, 14, 28
Open-source, 115
Parallel capacitance, 32
Parasitic, 31
Parasitic effect, 42
PCB, 27
Permittivity, 3, 22, 30, 46
Power plane, PCB, 14
Printed circuit board, 27
Problem-solving: annotate diagrams, 107
Problem-solving: check for exceptions, 108
Problem-solving: checking work, 108
Problem-solving: dimensional analysis, 107
Problem-solving: graph values, 108
Problem-solving: identify given data, 107
Problem-solving: identify relevant principles, 107
Problem-solving: interpret intermediate results, 107
Problem-solving: limiting cases, 108
Problem-solving: qualitative to quantitative, 108
Problem-solving: quantitative to qualitative, 108
Problem-solving: reductio ad absurdum, 108
Problem-solving: simplify the system, 107
Problem-solving: thought experiment, 32, 107
Problem-solving: track units of measurement, 107
Problem-solving: visually represent the system, 107
Problem-solving: work in reverse, 108
Programming, computer, 47, 58
Python, 52
Qualitatively approaching a quantitative problem, 108
Reading Apprenticeship, 67
Reductio ad absurdum, 108-110
Resistance, 14
Resistor color code, 31
Schoenbach, Ruth, 67
Scientific method, 72
Series capacitance, 32
Simplifying a system, 107
Soakage, capacitor, 44
Socrates, 109
Socratic dialogue, 110
Source, 24
Source code, 48
SPDT switch, 28
SPICE, 67
Stallman, Richard, 115
Standard component values, IEC, 34
Supercapacitor, 27
Superconductor, 46
Thought experiment, 32, 107
Time constant, 29, 57
Tolerance codes, 31
Torvalds, Linus, 115
Units of measurement, 107
Via, PCB, 14
Visualizing a system, 107
Volt, unit, 22
Weight, 21
Whitespace, C++, 48, 49
Whitespace, Python, 55
Work in reverse to solve a problem, 108
WYSIWYG, 115, 116