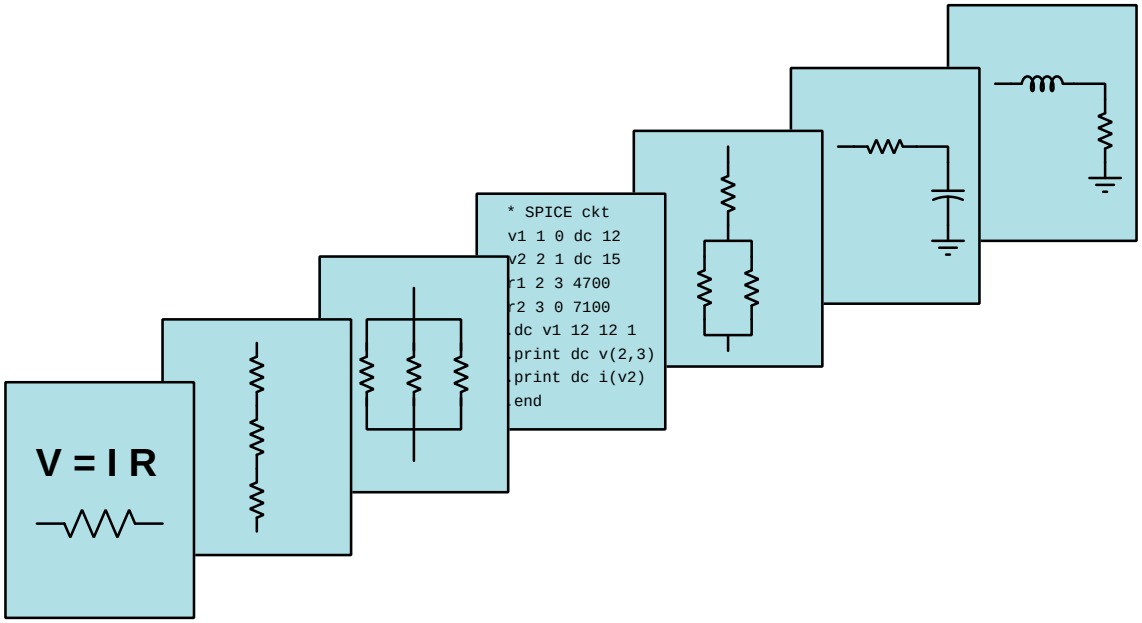
Modular Electronics Learning (ModEL) project



Manipulating Algebraic Equations

© 2020-2025 by Tony R. Kuphaldt – under the terms and conditions of the Creative Commons Attribution 4.0 International Public License

Last update = 5 January 2025

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

Contents

1 Introduction **3**
- 1.1 Recommendations for students . . . 3
- 1.2 Challenging concepts related to algebraic manipulation . . . 5
- 1.3 Recommendations for instructors . . . 6

2 Case Tutorial **7**
- 2.1 Example: manipulated equations . . . 8
  - 2.1.1 Parallel resistances . . . 8
  - 2.1.2 Voltages in a voltage divider circuit . . . 9
  - 2.1.3 Voltages in a series-parallel circuit . . . 10
  - 2.1.4 Torricelli's Theorem . . . 11
  - 2.1.5 Satellite orbital velocity . . . 11
  - 2.1.6 Voltages in an AC resistor-capacitor circuit . . . 12
  - 2.1.7 Roller bearing life . . . 13
  - 2.1.8 Pipe size . . . 13
  - 2.1.9 Transformer winding turns and impedances . . . 14
  - 2.1.10 Ship propulsion . . . 15
  - 2.1.11 Broaching a hole . . . 16
  - 2.1.12 Brinell hardness . . . 17
  - 2.1.13 Binary states . . . 17
- 2.2 Example: calculator variables . . . 18
- 2.3 Example: balanced equation checking in Python . . . 20

3 Tutorial **21**
- 3.1 What do all these letters mean? . . . 21
- 3.2 The "equals" symbol . . . 24
- 3.3 Addition and subtraction . . . 27
- 3.4 Multiplication and division . . . 29
- 3.5 Proper order of operations . . . 32
- 3.6 Factoring . . . 35
- 3.7 Reciprocation . . . 36
- 3.8 Powers and roots . . . 37
- 3.9 Exponents and logarithms . . . 40
- 3.10 Substitution . . . 44

4 Derivations and Technical References **47**
4.1 Algebraic identities . . . 48
4.2 Properties of real numbers . . . 49
4.3 Dimensional analysis . . . 50
4.4 The Ohm's Law Circle and public education in America . . . 51

5 Programming References **53**
5.1 Programming in C++ . . . 54
5.2 Programming in Python . . . 58

6 Questions **63**
6.1 Conceptual reasoning . . . 67
6.1.1 Reading outline and reflections . . . 68
6.1.2 Foundational concepts . . . 69
6.2 Quantitative reasoning . . . 70
6.2.1 Miscellaneous physical constants . . . 71
6.2.2 Introduction to spreadsheets . . . 72
6.2.3 Solving for n . . . 75
6.2.4 Manipulating the kinetic energy equation . . . 76
6.3 Diagnostic reasoning . . . 77
6.3.1 Identify the mistake, part 1 . . . 78
6.3.2 Identify the mistake, part 2 . . . 78
6.3.3 Identify the mistake, part 3 . . . 79
6.3.4 Identify the mistake, part 4 . . . 79

A Problem-Solving Strategies **81**

B Instructional philosophy **83**
B.1 First principles of learning . . . 84
B.2 Proven strategies for instructors . . . 85
B.3 Proven strategies for students . . . 87
B.4 Design of these learning modules . . . 88

C Tools used **91**

D Creative Commons License **95**

E References **103**

F Version history **105**

Index **106**

Chapter 1

Introduction

1.1 Recommendations for students

If you poll technical college faculty about the algebraic skill they find most lacking in their new students, the answer you are likely to get more than any other is *manipulating equations*. This is arguably the biggest reason why technical students *need* to learn algebra – because equations necessary to calculate important quantities aren't always found in the proper form – and paradoxically the point of greatest frustration for students and faculty alike. This module seeks to explain the logic and art of algebraic manipulation in a way that is both simple and practical. General principles are introduced section-by-section, with practical examples drawn from circuit analysis included for every principle.

Important concepts related to algebraic manipulation include the symbolic representation of numbers by non-numerical symbols called **variables**, the expression of general principles by **equations**, mathematics as a **language**, **subscripted** variables, mathematical **equality**, the **commutative** property of certain arithmetic operations, how equality remains unaffected by performing **identical operations** to both sides of an equation, **inverse operations**, **strategies** for algebraic manipulation, canceling terms when multiplying **fractions**, **order of operations**, **factoring**, **grouping symbols**, and **substitution**.

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

- How might you utilize an electronic calculator to perform experiments exploring inverse arithmetic operations? What hypothesis (i.e. prediction) might you pose for that experiment, and what result(s) would either support or disprove that hypothesis?
- What does it mean for two arithmetic operations to be the *inverse* of each other?
- How does a physical balance-beam help explain mathematical equality?
- What are some inverse-operation pairs in arithmetic?
- Why is it recommended to cultivate multiple methods for manipulating similar equations?

- What benefit is there in writing each and every step of an equation's manipulation?
- What do "grouping" symbols represent in an equation?
- Why is the order of operations significant to manipulating equations?
- Where might you apply factoring when manipulating an equation?
- What does a *root symbol* (e.g. square-root) represent in mathematics?
- How do exponents relate to logarithms?
- Why do we have multiple types of logarithm functions (e.g. natural log, common log)?
- How may we derive all the forms of Joule's Law from the most basic form ($P = IV$) and Ohm's Law?

The Tutorial provides an excellent active-reading opportunity, in that you may perform all the same manipulations shown in the text. After reading how a particular manipulation was done, you may write the original equation on a separate sheet of paper and then (without referencing the text) challenge yourself to manipulate it all on your own. Then, when you think you have arrived at a solution you may refer to the text to check your answer. This requires self-discipline, as it is much easier to simply read the original explanation and tell yourself that it all makes sense, but challenging yourself in this way is vital to building independent problem-solving skill. Technical reading is not a spectator activity, but has maximum benefit to the reader taking an active (participatory) approach.

1.2 Challenging concepts related to algebraic manipulation

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

- **Order of operations** – knowing the customary order of operations for arithmetic operations is not only important for properly evaluating any complex arithmetic expression or equation, but especially important for *manipulating* equations to solve for given variables.
- **Fractions** – of all the basic mathematical concepts, *fractions* seem to cause my students the greatest amount of trouble. Informal polls I've taken suggest a deficiency in our public-education math curriculum, where many students are encouraged to use electronic calculators to perform simple calculations, avoiding the use of fractions to perform mental arithmetic. Some students tell me fractions simply aren't emphasized much at all. This deficit causes much trouble when they must apply fractions to algebraic expressions, particularly when they must multiply or divide fractions together in order to manipulate an equation to solve for certain variables. As an instructor you should anticipate spending time in class reviewing and applying basic fractional arithmetic: how to add and subtract fractions, how to multiply and divide fractions, and how to reduce compound fractions to simpler forms. With regard to the application of fractions to equation manipulation, I recommend *multiplying by a fraction's reciprocal* in order to cancel out a fraction-multiplication as opposed to *dividing by that fraction*, since the division of one fraction by another naturally creates a compound fraction which then must be reduced in a subsequent step.

The *Case Tutorial* section "Example: manipulated equations" contains many examples of practical equations manipulated to solve for each of their variables. These serve well as practice problems for students, to see if they can perform their own manipulations to achieve the same results shown in the text.

1.3 Recommendations for instructors

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

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

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

  Assessment – Students show how the author was able to manipulation equations shown as examples in the Tutorial.

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

- **Outcome** – Manipulate algebraic equations to solve for specified variables

  Assessment – Demonstrate proper algebraic manipulations to solve for variables; e.g. pose problems in the form of the "Solving for n" Quantitative Reasoning question.

- **Outcome** – Demonstrate the use of memory (variables) in electronic calculations

  Assessment – Show how to solve quantitative problems with minimal rounding error and keystrokes by storing all intermediate results in the calculator's memory.

- **Outcome** – Diagnose a faulty algebraic manipulation

  Assessment – Identify improper algebraic manipulations and then demonstrate the correct way(s) to manipulate those same equations; e.g. pose problems in the form of the "Identify the mistake" Diagnostic Reasoning questions.

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: manipulated equations

2.1.1 Parallel resistances

The total resistance of a four-resistor parallel network is given by the following equation:

$$R_{total} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}}$$

Where,
R_{total} = Total parallel resistance (Ohms)
R_1 = Resistance of resistor #1 (Ohms)
R_2 = Resistance of resistor #2 (Ohms)
R_3 = Resistance of resistor #3 (Ohms)
R_4 = Resistance of resistor #4 (Ohms)

Equation manipulated to solve for R_1:

$$R_1 = \frac{1}{\frac{1}{R_{total}} - \left(\frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}\right)}$$

Equation manipulated to solve for R_2:

$$R_2 = \frac{1}{\frac{1}{R_{total}} - \left(\frac{1}{R_1} + \frac{1}{R_3} + \frac{1}{R_4}\right)}$$

Equation manipulated to solve for R_3:

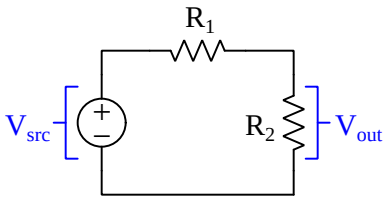
$$R_3 = \frac{1}{\frac{1}{R_{total}} - \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_4}\right)}$$

Equation manipulated to solve for R_4:

$$R_4 = \frac{1}{\frac{1}{R_{total}} - \left(\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}\right)}$$

2.1.2 Voltages in a voltage divider circuit

Voltages in the the following voltage divider circuit are related to each other by the equation shown below:



$$V_{out} = V_{src}\frac{R_2}{R_1 + R_2}$$

Where,
V_{src} = DC source voltage (Volts)
V_{out} = Voltage dropped across resistor R_2 (Volts)
R_1 = Resistance of resistor #1 (Ohms)
R_2 = Resistance of resistor #2 (Ohms)

Equation manipulated to solve for R_1:

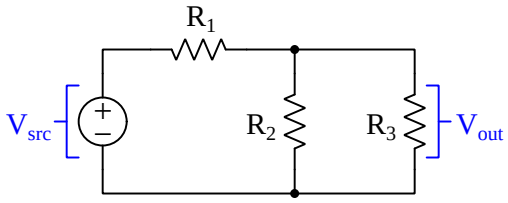
$$R_1 = \frac{V_{src}R_2}{V_{out}} - R_2$$

Equation manipulated to solve for R_2:

$$R_2 = \frac{R_1}{\frac{V_{src}}{V_{out}} - 1}$$

2.1.3 Voltages in a series-parallel circuit

Voltages in the the following series-parallel circuit are related to each other by the equation shown below:



$$V_{out} = V_{src}\frac{\frac{R_2 R_3}{R_2+R_3}}{R_1 + \frac{R_2 R_3}{R_2+R_3}}$$

Where,

V_{src} = DC source voltage (Volts)
V_{out} = Voltage dropped across both resistor R_2 and resistor R_3 (Volts)
R_1 = Resistance of resistor #1 (Ohms)
R_2 = Resistance of resistor #2 (Ohms)
R_3 = Resistance of resistor #3 (Ohms)

Equation manipulated to solve for R_1:

$$R_1 = \frac{\frac{V_{src}}{V_{out}} - 1}{\frac{1}{R_2} + \frac{1}{R_3}}$$

Equation manipulated to solve for R_2:

$$R_2 = \frac{R_1}{\frac{V_{src}}{V_{out}} - \frac{R_1}{R_3} - 1}$$

Equation manipulated to solve for R_3:

$$R_3 = \frac{R_1}{\frac{V_{src}}{V_{out}} - \frac{R_1}{R_2} - 1}$$

2.1.4 Torricelli's Theorem

An equation known as *Torricelli's Theorem* relates the velocity of water exiting a hole in a unpressurized container with the height of the liquid above the hole:

$$v = \sqrt{2gh}$$

Where,
v = Velocity of liquid exiting the hole
g = Acceleration of gravity (9.8 meters per second squared, or 32 feet per second squared)
h = Height of liquid column above the hole

Equation manipulated to solve for g:

$$g = \frac{v^2}{2h}$$

Equation manipulated to solve for h:

$$h = \frac{v^2}{2g}$$

2.1.5 Satellite orbital velocity

The velocity necessary for a satellite to maintain a circular orbit around Earth is given by this equation:

$$v = \sqrt{gh}$$

Where,
v = Satellite velocity (feet per second)
g = Acceleration of Earth gravity at sea level (32 feet per second squared)
h = Orbit altitude, (feet)

Equation manipulated to solve for g_c:

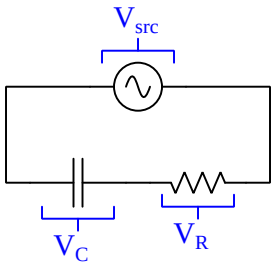
$$g = \frac{v^2}{h}$$

Equation manipulated to solve for h:

$$h = \frac{v^2}{g}$$

2.1.6 Voltages in an AC resistor-capacitor circuit

Voltages in the the following AC circuit are related to each other by the equation shown below:



$$V_{src} = \sqrt{V_C + V_R}$$

Where,
V_{src} = AC source voltage (Volts)
V_C = Voltage dropped across capacitor (Volts)
V_R = Voltage dropped across resistor (Volts)

Equation manipulated to solve for V_C:

$$V_C = V_{src}^2 - V_R$$

Equation manipulated to solve for V_R:

$$V_R = V_{src}^2 - V_C$$

2.1.7 Roller bearing life

The expected operating life of a rolling-contact bearing may be predicted by the following equation:

$$R_L = \left(\frac{C}{L}\right)^{\frac{10}{3}}$$

Where,
R_L = Operating life (millions of shaft revolutions)
C = Dynamic capacity of bearing (pounds)
L = Radial load applied to bearing (pounds)

Equation manipulated to solve for C:

$$C = L(R_L)^{\frac{3}{10}}$$

Equation manipulated to solve for L:

$$L = \frac{C}{(R_L)^{\frac{3}{10}}}$$

2.1.8 Pipe size

Mechanical, chemical, and civil engineers must often calculate the size of piping necessary to transport fluids. One equation used to relate the water-carrying capacity of multiple, small pipes to the carrying capacity of one large pipe is as follows:

$$N = \left(\frac{d_2}{d_1}\right)^{2.5}$$

Where,
N = Number of small pipes
d_1 = Diameter of each small pipe
d_2 = Diameter of large pipe

Equation manipulated to solve for d_2:

$$d_2 = d_1 N^{0.4}$$

Equation manipulated to solve for d_1:

$$d_1 = \frac{d_2}{N^{0.4}}$$

2.1.9 Transformer winding turns and impedances

The ratios of electrical transformer winding turns and impedances are related to each other as such:

$$\frac{N_p}{N_s} = \sqrt{\frac{Z_p}{Z_s}}$$

Where,
N_p = Number of primary winding turns
N_s = Number of secondary winding turns
Z_p = Impedance of primary circuit
Z_s = Impedance of secondary circuit

Equation manipulated to solve for N_p:

$$N_p = N_s \sqrt{\frac{Z_p}{Z_s}}$$

Equation manipulated to solve for N_s:

$$N_s = \frac{N_p}{\sqrt{\frac{Z_p}{Z_s}}}$$

Equation manipulated to solve for Z_p:

$$Z_p = Z_s \left(\frac{N_p}{N_s}\right)^2$$

Equation manipulated to solve for Z_s:

$$Z_s = \frac{Z_p}{\left(\frac{N_p}{N_s}\right)^2}$$

2.1.10 Ship propulsion

The amount of power required to propel a ship is given by this equation:

$$P = \frac{D^{\frac{2}{3}} V^3}{K}$$

Where,

P = Power required to turn propeller(s) (horsepower)
D = Vessel displacement (long tons)
V = Velocity (nautical miles per hour)
K = Admiralty coefficient (approximately 70 for a 30 foot long ship, load waterline)

Equation manipulated to solve for D:

$$D = \left(\frac{PK}{V^3} \right)^{\frac{3}{2}}$$

Equation manipulated to solve for V:

$$V = \left(\frac{PK}{D^{\frac{2}{3}}} \right)^{\frac{1}{3}}$$

Equation manipulated to solve for K:

$$K = \frac{D^{\frac{2}{3}} V^3}{P}$$

2.1.11 Broaching a hole

In machining production technology, the time required to "broach" a hole in a plate may be estimated by the following equation:

$$T = \frac{L}{C} + \frac{L}{R_c}$$

Where,
T = Time to broach hole (minutes)
L = Length of stroke (feet)
C = Cutting speed (feet per minute)
R_c = Return speed (feet per minute)

Equation manipulated to solve for L:

$$L = \frac{TCR_c}{R_c + C}$$

Equation manipulated to solve for C:

$$C = \frac{LR_c}{TR_c - L}$$

Equation manipulated to solve for R_c:

$$R_c = \frac{LC}{TC - L}$$

2.1.12 Brinell hardness

The *Brinell hardness* of a metal specimen is measured by pressing a ball into the specimen and measuring the size of the resulting indentation. To calculate Brinell hardness, this equation is used:

$$H = \frac{F}{(\frac{\pi d_1}{2})(d_1 - \sqrt{d_1^2 - d_s^2})}$$

Where,

H = Hardness of specimen (Brinell units)
F = Force on ball (kg)
d_1 = Diameter of ball (mm)
d_s = Diameter of indentation (mm)

Equation manipulated to solve for F:

$$F = H\Big(\frac{\pi d_1}{2}\Big)\Big(d_1 - \sqrt{d_1^2 - d_s^2}\Big)$$

Equation manipulated to solve for d_s:

$$d_s = \sqrt{d_1^2 - \Big(d_1 - \frac{2F}{H\pi d_1}\Big)^2}$$

2.1.13 Binary states

In digital electronic systems based on binary numeration, the number of possible *states* representable by the system is given by the following equation:

$$n_s = 2^{n_b}$$

Where,

n_s = Number of possible states
n_b = Number of binary "bits"

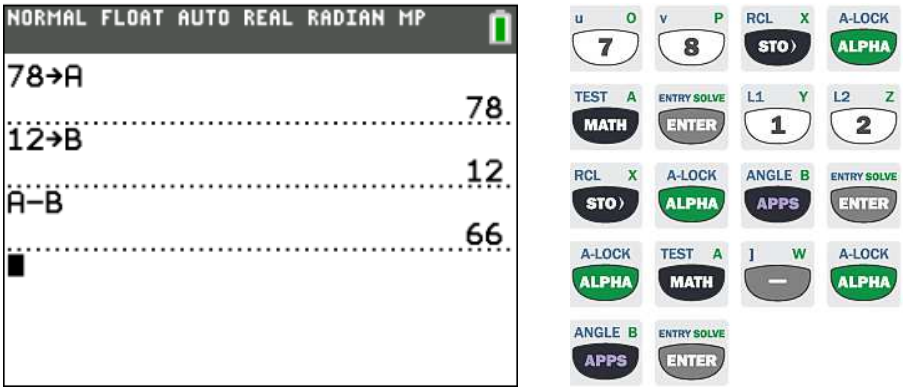
Equation manipulated to solve for F:

$$n_b = \frac{\log n_s}{\log 2}$$

2.2 Example: calculator variables

Variables are not just a useful concept for expressing mathematical relationships on paper, but they are also a practical tool for performing calculations electronically. Most hand calculators have at least *some* capacity to store numerical quantities in memory, and some offer full alphabetical naming capability for variables.

In the following example we see a Texas Instruments model TI-84 Plus C Silver Edition electronic calculator storing the value 78 in a variable named A and the value 12 in variable B, then subtracting $A - B$ to show the resulting difference of 66. The left-hand image shows the display screen of the calculator performing these operations while the right-hand image shows the sequence of keystrokes necessary to make the entries and perform the subtraction:



On this particular calculator many of the buttons have a green-colored alphabetical letter as an "overloaded" function (shown above and to the right of the button itself), these letters accessed by first pressing the "ALPHA" button. Broken down operation by operation:

- `7 8 (STO) (ALPHA) A (ENTER)` – stores the value 78 as variable A
- `1 2 (STO) (ALPHA) B (ENTER)` – stores the value 12 as variable B
- `(ALPHA) A - (ALPHA) B (ENTER)` – subtracts B from A

Variables are extremely useful for manual calculations, as they aid in multiple ways. When performing lengthy calculations I always urge my students to store all intermediate results in their calculators' memory (as variables) rather than re-enter them into the calculator manually, as it helps in the following ways:

1. When those results are stored in variables, they are stored in full precision (i.e. all decimal places) which means no rounding errors will be incurred upon recall.
2. By storing those results and recalling them from memory rather than re-typing, you avoid the possibility of keystroke errors.
3. Many fewer keystrokes are necessary to save results into memory and recall some time later than would be necessary to re-enter them every time they're needed, which saves much time in a lengthy calculation.

If one has access to a full-fledged computer to perform mathematical calculations, a favorite application of mine is the *Python* programming language. Python is an *interpreted* language which means the commands you enter are executed immediately upon pressing the "Enter" key on your keyboard. Since Python, like all programming languages, is capable of understanding basic mathematical expressions, this means it functions well as a manual calculator.

Take the following example, where the same operations shown previously on the Texas Instruments TI-84 calculator are performed in a Python interpretor window. Note that all lines beginning with the three chevrons (`>>>`) are text entered by the user of Python, while the last line lacking these chevrons is Python's response to the previous calculation:

```
>>> A = 78
>>> B = 12
>>> A - B
66
```

In Python, as in most other programming languages, the task of assigning a value to some variable consists of using the "equals" (`=`) symbol, with the variable name preceding the value. Thus, `A = 78` assigns the value 78 to the variable A. Note how this sequence is the reverse of that used in the TI-84 calculator, where we first entered the value and then used the "Store" function to copy that value to a chosen variable.

An interesting distinction between variables as offered by electronic calculators and by computer programming languages versus variables as they are typically used in mathematical notation is that calculator/computer variables may consist of multiple letters spelling words. For example, in mathematics the variable A is typically used to signify two-dimensional area, but in Python we could just as easily spell out the word *Area* and use that as our variable. Note below how we spell out variable names for a rectangle's height and width, then calculate the area, then display that area in Python:

```
>>> Length = 5.3
>>> Width = 2.9
>>> Area = Length * Width
>>> Area
15.37
```

Here, `Area`, `Length`, and `Width` are all distinct variables. Note how we do not get to view the actual calculated value of `Area` until the last statement where we simply typed `Area` and pressed Enter. The ability to *name* calculator and programming variables in this way is very helpful for making calculations self-documenting, so that anyone viewing our work should be able to tell what it is we're calculating!

2.3 Example: balanced equation checking in Python

The *Python* computer programming language is useful as an electronic calculator because each command runs as soon as we enter it into a Python interpreter window. Here we will use Python's ability to perform basic mathematics and to check for equality using the double-equals operator (==) to prove that an algebraic equation remains balanced throughout several different manipulations.

First, we will assign integer values to two variables, x and y, and then compute their sum:

```
>>> x = 7
>>> y = 3
>>> x + y
10
```

Based on the arbitrary values chosen for x and y, it is true to say that $x + y = 10$. Now we will manipulate this equation in various ways and relay on Python to confirm that both sides of each new equation form are still equal to each other:

```
>>> x + y == 10
True
>>> x == 10 - y
True
>>> 2 * x == 2 * (10 - y)
True
>>> (2 * x)/4 == (2 * (10 - y))/4
True
```

Note how we started with a basic check of equality for two terms we already knew were equal to each other: $x + y$ and 10. Next, we subtracted y from both sides of the equation and re-checked for equality. After that, we multiplied both sides of the equation by 2 and re-checked for equality. Lastly, we divided both sides by 4 and re-checked for equality. At each and every one of these steps the equality test resulted in a "True" assessment because our manipulations were done identically to both sides of the equation.

Note also the careful use of parentheses to enforce proper order of operations. When multiplying both sides of the equation $x = 10 - y$ by two we had to surround the $10 - y$ with parentheses to ensure the entirety of that right side of the equation got multiplied by two. In other words, $x = 10 - y$ now became $2x = 2(10 - y)$. If we were to have omitted these parentheses we would have disrupted the equality as proven here:

```
>>> 2 * x == 2 * 10 - y
False
```

Without parentheses surrounding $10 - y$ the multiplier of two applies only to the 10 which standard order of operations but not what we wished to do in multiplying the *entire right-hand side of the equation* by two just as we multiplied the entire left-hand side of the equation by two.

Chapter 3

Tutorial

3.1 What do all these letters mean?

Students new to algebra face something they never encountered with arithmetic: *letters* placed together with numbers. For example, any student familiar with multiplication will immediately be able to compute the following product:

$$3 \times 5$$

Three multiplied by five is equivalent to five added to itself three times: $5 + 5 + 5$, the product being fifteen (15). However, the following algebraic expression throws many a new student into confusion:

$$n \times R$$

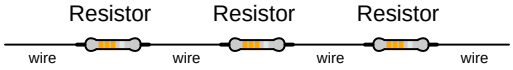
What in the world does n or R mean, any why would we be multiplying *letters* rather than *numbers*? The answer to this very good question is that these letters function as place-holders for number-values. For example, if we were told that n has a value of 3 and R has a value of 5, then $n \times R$ would be the same as saying 3×5, the result once again being 15:

If n stands for 3 and R stands for 5 ,

Then $n \times R$ means the same as 3×5 which is 15

In other words, as place-holders for numbers n and R may be *substituted* for the actual numerical values. In a sense, we may view each of these letters – called *variables* – as a kind of alias for a number, subject to the same rules we follow when using numbers. Interesting though this may be, it is still a fair question to ask *why* we might ever want to do such a thing. After all, if 3×5 is good enough to signify the multiplication of three and five, what's the point of declaring a couple of letters (variables) to be place-holders for three and five, and then multiplying those variables? This just seems like so much pointless busy-work.

A practical example will show why variables are useful. Consider a set of electrical components called *resistors* connected in-line with one another by segments of metal wire. The purpose of a resistor is to dissipate energy from electrical charge carriers passing through it, that energy being converted into heat. Resistors may be used as electrical heaters because they become warm when electricity passes through, but they have many other uses too:



Every resistor has a *resistance* value measured in the unit of the Ohm (Ω), this value describing how effective the resistor is at extracting and dissipating energy from passing electric charge carriers. When multiple resistors are connected in-line like this (i.e. called a *series* network), the total amount of resistance from end-to-end is the sum of all the series-connected resistors.

If any series resistor network has a total resistance that is the sum of every resistance value in that network, then a series resistor network comprised of identical resistors must have a total resistance equal to the number of resistors multiplied by the resistance of any one of them. For example, if we built a network of three series-connected resistors, each resistor having 5 Ohms (5 Ω) of resistance, then the entire network would have a total resistance of 15 Ω.

Here is an example of how variables may be useful. Instead of describing the electrical resistance of a series network comprised of identical resistors in verbal terms, we may simply write an algebraic *equation* or *formula* saying the same thing more efficiently:

$$\text{Total resistance} = n \times R$$

Where,

n = The number of identical series-connected resistors

R = The resistance of any one of those resistors

"Total resistance equals $n \times R$" is much simpler to write and to say than *"total resistance is equal to the number of resistors multiplied by the resistance of any one of them"*. It is especially efficient if the reader happens to already know that the variable R always means "resistance" for electrical components and that n typically means some whole-numbered quantity of things.

Thus, variables give us the ability to write *general* mathematical statements applying to wide ranges of situations. The only resistor network described by the expression 3×5 is one where three 5 Ω resistors are connected in series, but $n \times R$ describes the total resistance for *any* number of series-connected resistors of *any* common value. "3×5" merely describes one particular network of resistors, but "$n \times R$" is more powerful because it describes the *general principle*.

This is a very practical use of algebraic variables because it lets us easily say complicated things. In this way, mathematics plays the role of a *language* for describing principles. Like any language, algebra has rules of "grammar" we must master in order to use it as a tool of communication with other people, and we will soon see how these rules help us apply the described principles in ways that are not immediately apparent.

It is customary in algebra to use single letters (of any alphabet) as variables, and to interpret adjacent letters as being multiplied by one another. Thus, $n \times R = nR$. The use of single letters as variables is limiting, though, especially when we need to represent different quantities of the same *type*.

Suppose we wished to write an equation describing total resistance for any series resistor network, regardless of individual resistance values. In the previous example (nR) we were able to use a single R variable to represent individual resistance because all the resistors happened to be identical in value. But how may we represent multiple resistances of different resistance value when we only have one letter R in the English alphabet?

An answer to this practical problem is something called a *subscript*[1]. These are additional characters written immediately below and to the right of the variable letter, to distinguish similar variables from one another. For example, if we have a three-resistor series network where each resistor has a different resistance value, we could write a total resistance equation using subscripts to keep all the different resistances distinct:

$$R_T = R_1 + R_2 + R_3$$

Where,
R_T = Total series resistance
R_1 = Resistance of resistor #1
R_2 = Resistance of resistor #2
R_3 = Resistance of resistor #3

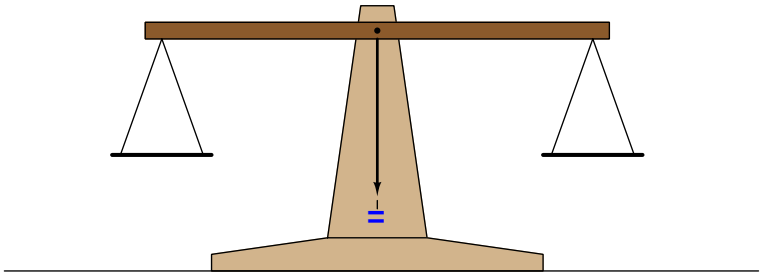
The subscript – be it a letter, number, or even a word – does not limit the numerical value of that variable in any way. It would be a mistake to think that R_3 must be larger than R_1 simply because 3 is larger than 1. These subscripts merely exist to distinguish each of the R variables in this question so we don't confuse one for another. While the letter chosen for any variable is often constrained by convention (e.g. R for electrical resistance, V for electrical voltage, I for electrical current), subscripts are entirely arbitrary.

Be aware as well that letters from non-English alphabets are fair to use as variables. This is often done in physics and engineering where the list of standardized variable letters exceeds the number of letters in the English alphabet. Greek letters are popular for standardized variables, including μ for magnetic permeability, ϵ for electric permittivity, ρ for mass density, γ for weight density, β for transistor current gain, λ for wavelength, θ for angle, ω for angular velocity, etc.

[1] Other solutions to this problem exist in different mathematics-based disciplines such as computer programming. In programming, all variables must be described by plain-text characters where there is no provision for subscripts, and so the solution there is to allow variables to have multi-character *names*. For example, we might choose to create a variable named `R1` or `R_1` in lieu of R_1 to represent the amount of resistance for resistor #1 in a computer program. Similarly, we could create a variable named `Rtotal` to represent total resistance, since the inability to represent subscripts in plain-text prevents us from making a program variable called R_{total}. The disadvantage of allowing multi-character variable names is that we must give up the convention of adjacent letters implying multiplication. The algebraic statement nR means $n \times R$, but in computer programming `nR` is simply a single variable named "nR". If we wish to instruct a computer to multiply two variables together, we must explicitly do so using a "star" character for the multiplication operator (i.e. `n * R`).

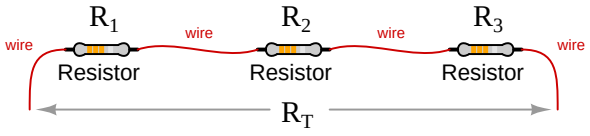
3.2 The "equals" symbol

The "equals" symbol (=) simply means two quantities share the same exact value, but many students find it helpful to visualize mathematical equality in physical terms. For this reason, I will relate the "equals" symbol to a balance-beam scale used to compare masses:

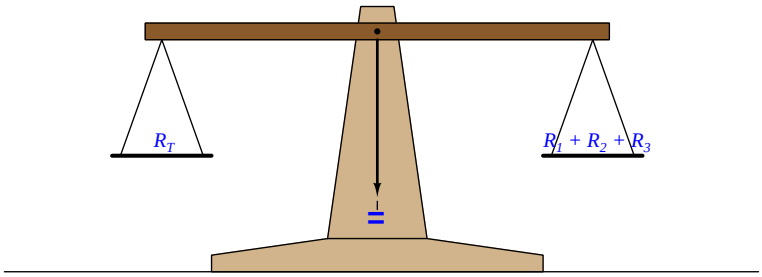


When the mass of each scale pan is equal to the other, the balance beam comes to rest in the center position, and we say that those two masses are equal to each other. As shown above with both pans empty, the scale may be said to embody the equation $0 = 0$.

Applying this concept to something previously discussed, we will show the balance beam as a metaphor for the total resistance equation of a three-resistor series network such as the three resistors shown below:

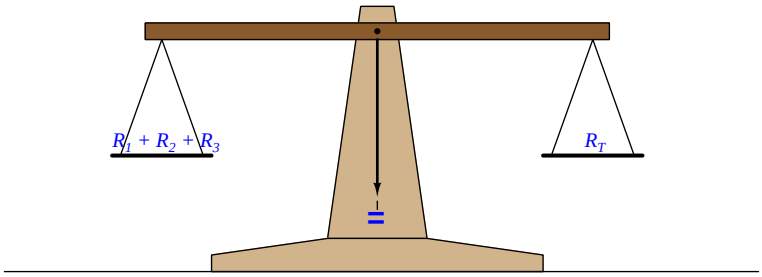


$$R_T = R_1 + R_2 + R_3$$



This equality remains just as true if we swap the written order. For example, if $R_1+R_2+R_3 = R_T$ then it must also be true to say $R_T = R_1 + R_2 + R_3$:

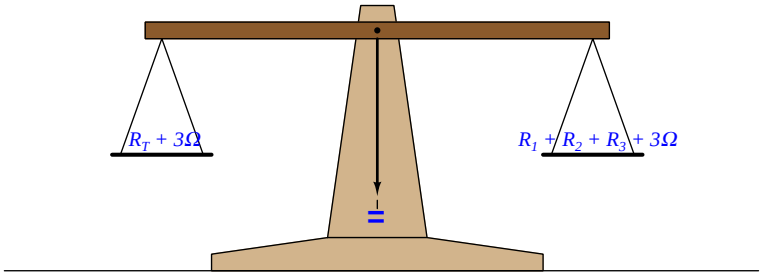
$$R_1 + R_2 + R_3 = R_T$$



A fundamental principle in algebra – which is rather obvious when related to the balance-beam analogy – is that we may arbitrarily manipulate any equation so long as we apply the exact same manipulation(s) to the terms on *both* sides of the "equals" symbol. If we imagine a balance-beam scale in a perfectly balanced condition, we ought to be able to add identical masses to both pans and still see the beam settle at a perfect balance condition.

For example, if we take our series resistor network equation and add 3 Ohms of resistance to both sides, it must remain balanced. This manipulation of the equation – so long as it is done *exactly the same* to both sides of it – will not invalidate the truth of that equation:

$$R_T + 3\Omega = R_1 + R_2 + R_3 + 3\Omega$$



It isn't just addition that works this way. It's possible for us to apply *any* mathematical operation to both sides of an equation and have that equation still remain true.

Consider the following possibilities of applying identical mathematical operations to both sides of this equation, where x represents any real-number quantity:

$$R_T - x = R_1 + R_2 + R_3 - x \qquad \textit{(Subtracting } x \textit{ from both sides)}$$

$$xR_T = x(R_1 + R_2 + R_3) \qquad \textit{(Multiplying both sides by } x\textit{)}$$

$$\frac{R_T}{x} = \frac{R_1 + R_2 + R_3}{x} \qquad \textit{(Dividing both sides by } x\textit{)}$$

$$-R_T = -(R_1 + R_2 + R_3) \qquad \textit{(Negating both sides)}$$

$$\frac{1}{R_T} = \frac{1}{R_1 + R_2 + R_3} \qquad \textit{(Reciprocating both sides)}$$

$$R_T^x = (R_1 + R_2 + R_3)^x \qquad \textit{(Raising both sides to the power of } x\textit{)}$$

$$\sqrt{R_T} = \sqrt{R_1 + R_2 + R_3} \qquad \textit{(Taking the square-root of both sides)}$$

$$x^{R_T} = x^{R_1 + R_2 + R_3} \qquad \textit{(Elevating both sides as powers of } x\textit{)}$$

$$\log R_T = \log(R_1 + R_2 + R_3) \qquad \textit{(Taking the logarithm of both sides)}$$

$$\sin R_T = \sin(R_1 + R_2 + R_3) \qquad \textit{(Applying trigonometric functions to both sides)}$$

$$\frac{d}{dx} R_T = \frac{d}{dx}(R_1 + R_2 + R_3) \qquad \textit{(Differentiating both sides with respect to } x\textit{)}$$

$$\int R_T \, dx = \int (R_1 + R_2 + R_3) \, dx \qquad \textit{(Integrating both sides with respect to } x\textit{)}$$

Appealing to the balance-beam analogy once more, if we begin with a balance-beam containing precisely equal masses, even if those masses are comprised of physically different substances or different configurations of the same substance, we ought to be able to add, subtract, multiply, divide, etc. both of those masses identically and still end up with the system being in perfect balance.

As irresistible as this logic may seem, it quite possibly strikes you as being useless. So what if we can identically manipulate both sides of an equation and have it remain equal? Who cares? Of what possible use is this in any practical circumstance? If all of these examples seem as pointless to you as lettered variables once were, read on!

3.3 Addition and subtraction

The fact that we may freely manipulate algebraic equations so long as we apply the same operations equally to both sides turns out to be an incredibly useful mathematical tool for re-writing equations in different forms. Let us return to our three-resistor series resistance equation once more:

$$R_T = R_1 + R_2 + R_3$$

This equation is written to be useful if we happen to know the resistance value of each individual resistor (R_1, R_2, and R_3) and we wish to calculate the total resistance of the network (R_T), but what if our practical problem were different? Suppose we needed to build a three-resistor network with a certain total resistance (R_T) and had to use two other resistors of pre-determined value (R_1 and R_2), and were left to calculate the necessary value of the third resistor (R_3)? What we need is an equation written with R_3 all by itself on one side and all the other resistances (R_T, R_1, and R_2) on the other side.

Knowing we may subtract any quantity we want from any equation so long as we subtract equally from both sides is the key to solving this problem. What we may do is subtract R_1 and R_2 from both sides, one at a time, and see the equation take a different form:

Step	Equation	Explanation
1	$R_T = R_1 + R_2 + R_3$	Equation in its original form
2	$R_T - R_1 = R_1 + R_2 + R_3 - R_1$	Subtracting R_1 from both sides
3	$R_T - R_1 = R_2 + R_3$	Letting $R_1 - R_1$ cancel out to zero
4	$R_T - R_1 - R_2 = R_2 + R_3 - R_2$	Subtracting R_2 from both sides
5	$R_T - R_1 - R_2 = R_3$	Letting $R_2 - R_2$ cancel out to zero

Now we have what we want: an equation solved for R_3. This tells us we need to take the total desired resistance value (R_T), subtract the known resistance values of R_1 and R_2, and what remains must be the necessary resistance value of R_3.

This example was performed in very explicit, step-by-step fashion for the purpose of avoiding assumptions. Usually when a person applies such manipulation to an equation they will do so in fewer steps:

Step	Equation	Explanation
1	$R_T = R_1 + R_2 + R_3$	Equation in its original form
2	$R_T - (R_1 + R_2) = R_1 + R_2 + R_3 - (R_1 + R_2)$	Subtracting $(R_1 + R_2)$ from both sides
3	$R_T - (R_1 + R_2) = R_3$	Letting $R_1 + R_2$ terms cancel out to zero

The result is the same, but with fewer steps than before.

We see how we may exploit the principle of identically manipulating both sides of an equation to change its form into something more useful to our immediate needs, but it is clear that this exploitation requires some strategic thinking. We must somehow examine the equation in its original form, and then decide which manipulations will re-format the equation to our liking.

Arithmetic operations are reversible, which means certain operations "un-do" other operations. Subtraction, for example, un-does addition. Division un-does multiplication. Square-rooting un-does squaring. These operation-pairs are called *inverse operations*, and we must know what they are in order to judiciously choose operations that will manipulate equations in the way we would prefer.

You will notice that we chose to *subtract* R_1 and R_2 from both sides of the series resistance equation to isolate R_3 because in its original form the equation showed R_1 and R_2 being *added* to R_3. Subtraction "un-did" that addition, and allowed us to "cancel" R_1 and R_2 from the right-hand side of the equation to leave R_3 by itself.

Now we will return to our previous problem of solving for R_3 in the series resistance equation, in order to show an alternative method of solution. You will find there is usually more than one correct method for manipulating an equation to your liking, and that it is good training for your mind to practice multiple techniques rather than settle on a favorite. Observe the following sequence of steps, where our first manipulation will be to subtract R_3 from both sides rather than subtract R_1 and R_2:

Step	**Equation**	**Explanation**
1	$R_T = R_1 + R_2 + R_3$	Equation in its original form
2	$R_T - R_3 = R_1 + R_2 + R_3 - R_3$	Subtracting R_3 from both sides
3	$R_T - R_3 = R_1 + R_2$	Letting $R_3 - R_3$ cancel out to zero
4	$R_T - R_3 - R_T = R_1 + R_2 - R_T$	Subtracting R_T from both sides
5	$-R_3 = R_1 + R_2 - R_T$	Letting $R_T - R_T$ cancel out to zero

At this point we appear to be very close to our goal of solving for R_3. However, note that we have not yet isolated R_3, but rather have isolated $-R_3$. If we add R_1 and R_2 then subtract R_T, the result will be the resistance of R_3 expressed as a *negative* quantity rather than a *positive* quantity.

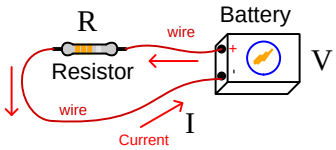
Something we may do next is *reverse the sign* of both sides of the equation, like this:

Step	**Equation**	**Explanation**
6	$-(-R_3) = -(R_1 + R_2 - R_T)$	Reversing sign of both sides
7	$R_3 = -(R_1 + R_2 - R_T)$	Letting $-(-R_3)$ cancel out to R_3

Although this result appears different from what we obtained by subtracting R_1 and R_2 in our first step, it is equally valid.

3.4 Multiplication and division

Many equations used to describe the behavior of electricity involve multiplication and division. Perhaps the most familiar is *Ohm's Law*, relating voltage (V), current (I), and resistance (R) for any component or network of components:



$$V = IR$$

Where,

V = Voltage (i.e. energy gained/lost by electric charge carriers)
I = Current (i.e. the rate at which electric charge carriers drift)
R = Resistance (i.e. voltage drop per drift rate)

The equation $V = IR$ works quite well to compute voltage if I already know the values of current and resistance. However, what if I a happen to know the amount of voltage (V) across an electrical resistance (R) and wish to calculate the amount of current (I) through that resistance? I must manipulate $V = IR$ to solve for I:

Step	Equation	Explanation
1	$V = IR$	Equation in its original form
2	$\frac{V}{R} = \frac{IR}{R}$	Dividing both sides by R
3	$\frac{V}{R} = (I)(1)$	Letting $\frac{R}{R}$ cancel out to one
4	$\frac{V}{R} = I$	Letting $(I)(1)$ simplify to I

Division was the correct operation to apply to both sides of the equation because this is what was necessary to "un-do" the *multiplication* existing between I and R in the original form. Like subtraction and addition are operationally inverse to each other, so are division and multiplication.

The same strategy works to solve for R, in case you happen to already know the values of V and I and need to calculate resistance from them:

Step	Equation	Explanation
1	$V = IR$	Equation in its original form
2	$\frac{V}{I} = \frac{IR}{I}$	Dividing both sides by I
3	$\frac{V}{I} = (1)(R)$	Letting $\frac{I}{I}$ cancel out to one
4	$\frac{V}{I} = R$	Letting $(1)(R)$ simplify to R

Of course, we should be able to apply the same principle to manipulate Ohm's Law starting from any of its forms. For example, let's try to manipulate $I = \frac{V}{R}$ to solve for R. From our previous work we already know the result should be $R = \frac{V}{I}$, but practicing algebraic manipulation is good mental exercise and so we will do it:

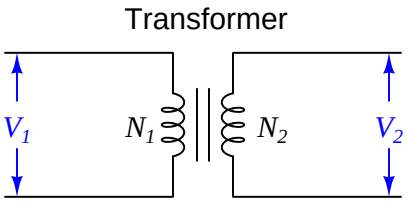
Step	**Equation**	**Explanation**
1	$I = \frac{V}{R}$	Equation in its original form
2	$IR = \frac{VR}{R}$	Multiplying both sides by R
3	$IR = (V)(1)$	Letting $\frac{R}{R}$ cancel out to one
4	$IR = V$	Letting $(V)(1)$ simplify to V
5	$\frac{IR}{I} = \frac{V}{I}$	Dividing both sides by I
6	$(1)(R) = \frac{V}{I}$	Letting $\frac{I}{I}$ cancel out to one
7	$R = \frac{V}{I}$	Letting $(1)(R)$ simplify to R

An alternative solution exists, but it requires *reciprocation* of the equation:

Step	**Equation**	**Explanation**
1	$I = \frac{V}{R}$	Equation in its original form
2	$\frac{I}{V} = \frac{V}{RV}$	Dividing both sides by V
3	$\frac{I}{V} = \frac{1}{R}$	Letting $\frac{V}{V}$ cancel out to one
4	$\frac{V}{I} = \frac{R}{1}$	Reciprocating both sides
5	$\frac{V}{I} = R$	Letting $\frac{R}{1}$ simplify to R

A *very* common mistake committed by students new to algebraic manipulation is failing to realize $\frac{V}{RV}$ simplifies to $\frac{1}{R}$ and not to R. Since R begins on the denominator of the fraction, it must remain there after the Vs cancel.

Fractions tend to confuse many students, and so it is worth taking the time to explore manipulation of equations containing them. Consider the following equation relating the ratio (fraction) of voltages in a transformer to the number of turns in each of the transformer's wire coils:



$$\frac{V_1}{V_2} = \frac{N_1}{N_2}$$

Suppose we wished to solve for N_1 in this equation. This is possible by multiplying each side by N_2, but I find it makes more sense to represent this multiplying quantity as a fraction of its own, $\frac{N_2}{1}$:

$$\left(\frac{N_2}{1}\right)\left(\frac{V_1}{V_2}\right) = \left(\frac{N_1}{N_2}\right)\left(\frac{N_2}{1}\right)$$

When we do this, N_2 cancels out of the right-hand side of the equation, leaving us with N_1 by itself on that side which is what we wanted:

$$\frac{N_2 V_1}{V_2} = N_1$$

Suppose, though, we wished to solve for N_2 instead of N_1. It's possible to do so by taking the last version of this equation (solved for N_1) and multiplying both sides of it by some quantity that will move the $\frac{V_1}{V_2}$ fraction to the right-hand side of the equation. That quantity will be the fraction $\frac{V_2}{V_1}$, designed to cancel out both V_1 and V_2 from the left-hand side:

$$\left(\frac{V_2}{V_1}\right)\left(\frac{N_2 V_1}{V_2}\right) = N_1\left(\frac{V_2}{V_1}\right)$$

$$N_2 = \frac{N_1 V_2}{V_1}$$

With these examples we see the strategy of multiplication by strategically-chosen fractions in order to cancel out any variable(s) we wish to eliminate from one side of an equation. At the beginning of this section we saw how multiplication may be "un-done" by division, but when the equation in question contains fractions, dividing by more fractions tends to confuse matters by introducing *compound fractions* (i.e. a fraction comprised of one or more other fractions). If instead we *multiply* fractions (as shown here) rather than divide, we generally avoid the creation of these cumbersome compound fractions.

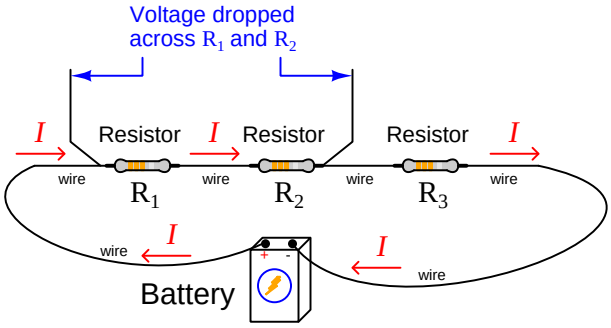
3.5 Proper order of operations

An important concept in arithmetic is *proper order of operations*, referring to the precedence some operations have over others[2]. For example, the correct equivalence for the expression $4 + 3 \times 2$ is 10 rather than 14, because multiplication takes precedence over addition. Many students learn the mnemonic *PEMDAS* to help them remember proper order of operations:

1. **P**arentheses (and other grouping symbols)
2. **E**xponents and roots
3. **M**ultiplication and **D**ivision
4. **A**ddition and **S**ubtraction

When an equation contains more than one of the operation types listed by PEMDAS, proper order of operations becomes important to achieve the correct result. The same is true when manipulating equations containing more than one PEMDAS operation type: if we manipulate the equation in the wrong order, we will not obtain a correct result.

For example, consider the following circuit consisting of three resistors connected in series to a battery, while measuring the voltage across the span of resistors R_1 and R_2. Electric current (I) of known magnitude circulates clockwise in the circuit:



Suppose we knew the desired voltage between the labeled points in this circuit, and we also knew the value of the current flowing through the resistors, and we also knew the resistance of R_1, but we did not know the resistance of R_2. The equation solving for voltage across both R_1 and R_2 with a known current is[3]:

$$V = I(R_1 + R_2)$$

[2] It is worth noting that there is no logical reason why some operations must take precedence over others. Just like the arbitrary conventions of any particular spoken language, PEMDAS is purely a mathematical convention.

[3] This equation comes from a recognition that series resistances *add*, and from Ohm's Law which tells us voltage is the product of current and resistance ($V = IR$).

Our task, then, is to manipulate this equation to solve for R_2. Before we begin, we must determine the proper order of operations for evaluating the equation in its present form. Applying PEMDAS, we must evaluate the contents of the parentheses first (because those symbols *group* the terms R_1 and R_2 together as one), then multiply that by I. Documenting this proper order of steps:

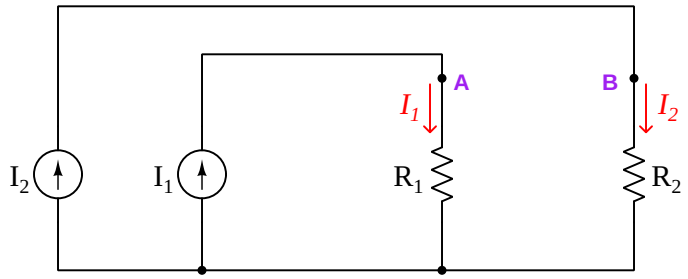
1. Addition: $R_1 + R_2$
2. Multiplication: $I(R_1 + R_2)$

If our goal is to isolate R_2 by itself in the equation, we need to "un-do" all the operations surrounding R_2 in the equation, but we need to do so in the correct order. When manipulating an equation to isolate one variable "buried" amidst other variables and arithmetic operations, the proper order of steps must be *the reverse of the standard order of operations used to evaluate the original equation.* For this particular case it means first un-doing the multiplication with I, then un-doing the addition with R_1. Here are the steps:

Step	**Equation**	**Explanation**
1	$V = I(R_1 + R_2)$	Equation in its original form
2	$\frac{V}{I} = \frac{I(R_1+R_2)}{I}$	Dividing both sides by I
3	$\frac{V}{I} = (1)(R_1 + R_2)$	Letting $\frac{I}{I}$ cancel out to one
4	$\frac{V}{I} = R_1 + R_2$	Letting $(1)(R_1 + R_2)$ simplify to $R_1 + R_2$
5	$\frac{V}{I} - R_1 = R_2$	Subtracting R_1 from both sides

A useful analogy to help understand the importance of *order of operations* is to imagine the evaluation of an equation as being the assembly of a complex machine, where parts must be put together in a particular order. Manipulating the equation to solve for one variable is akin to removing one of the parts buried within that machine: to access that part, one must disassemble the machine in the reverse order of how it was assembled.

Here is another example of manipulating an equation with a combination of operations that must be "disassembled" in a particular order. Suppose we are analyzing the following circuit, shown in standard electrical *schematic* form rather than as a pictorial illustration:



Using two important electrical principles – *Ohm's Law* and *Kirchhoff's Voltage Law*[4] – it is possible to derive an equation predicting voltage between test points A and B, that equation being $V_{AB} = (I_1R_1) - (I_2R_2)$. The proper order of operations for evaluating this equation appears below:

1. Multiplications: I_1R_1 and I_2R_2
2. Subtraction: $(I_1R_1) - (I_2R_2)$

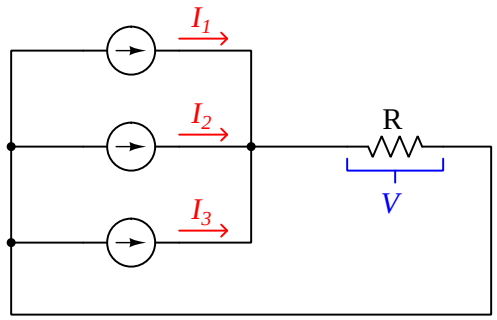
Therefore, in order to isolate any of the variables on the right-hand side of the equation, we must first un-do the subtraction, and then un-do multiplication. Suppose we needed to solve for current I_2 in this equation. Shown below is one possible solution:

Step	Equation	Explanation
1	$V_{AB} = (I_1R_1) - (I_2R_2)$	Equation in its original form
2	$V_{AB} + (I_2R_2) = (I_1R_1) - (I_2R_2) + (I_2R_2)$	Adding I_2R_2 to both sides
3	$V_{AB} + (I_2R_2) = I_1R_1$	Letting $-(I_2R_2) + (I_2R_2)$ cancel out to zero
4	$V_{AB} + (I_2R_2) - V_{AB} = I_1R_1 - V_{AB}$	Subtracting V_{AB} from both sides
5	$I_2R_2 = I_1R_1 - V_{AB}$	Letting $V_{AB} - V_{AB}$ cancel out to zero
6	$\frac{I_2R_2}{R_2} = \frac{I_1R_1 - V_{AB}}{R_2}$	Dividing both sides by R_2
7	$(I_2)(1) = \frac{I_1R_1 - V_{AB}}{R_2}$	Letting $\frac{R_2}{R_2}$ cancel out to one
8	$I_2 = \frac{I_1R_1 - V_{AB}}{R_2}$	Letting $(I_2)(1)$ simplify to I_2

[4]Ohm's Law predicts voltage across each resistance being the product of current and resistance (i.e. $V = IR$). Kirchhoff's Voltage Law predicts the algebraic sum of all voltages in a loop must equal zero, and so the voltage between points A and B must be equal to the difference in voltages across the two resistors.

3.6 Factoring

An important mathematical tool for manipulating certain equations is *factoring*, which is where we identify and isolate one or more variables common to multiple (added) terms. To illustrate its use, we will explore a circuit multiple electric currents pass through a common resistor:



The amount of voltage across the single resistor in this circuit is given by the following equation:

$$V = I_1R + I_2R + I_3R$$

Suppose we happened to know the value of all three currents in this circuit (I_1, I_2, I_3) as well as the desired voltage across the resistor (V), but did not know the necessary resistance value (R). Clearly, we must manipulate this equation to solve for R, but how can we isolate R when it appears *three times* in the equation rather than just once?

Factoring is key to the solution. By identifying the fact that R appears in all three terms on the right-hand side of the equation, we may re-write that side of the equation as a product between the common variable R and the sum of the three currents:

$$V = R(I_1 + I_2 + I_3)$$

If this equivalence is not clear to see, just *distribute*[5] R to the three current terms inside of the parentheses and you will find it reverts back to the equation's original form.

Now that we have factored R out of the original three terms, solving for R is as simple as dividing both sides of the equation by $I_1 + I_2 + I_3$ to leave R by itself:

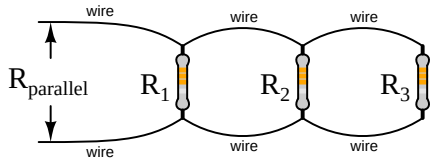
$$\frac{V}{I_1 + I_2 + I_3} = R$$

Note how the long fraction bar serves as a grouping symbol for the three current terms, just as those terms were enclosed by parentheses in the factored version of the equation. The fact that the fraction bar extends over all three current variables informs us that those three terms must be added *before* performing the division with voltage.

[5] *Distribution* is the repeated multiplication of each term inside the parentheses by the term outside the parentheses, resulting in a sum-of-products.

3.7 Reciprocation

When electrical resistors are connected together in *parallel* as shown below, the resistance of the parallel electrical network ends up being less than the lowest-valued resistance in that network. For nostalgia's sake we will return to pictorial illustrations instead of schematics:



The equation describing the network resistance of three resistors connected in parallel is ugly:

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

For anyone new to algebraic manipulation, solving for R_2 in this equation is the stuff of nightmares. However, even an ungainly equation such as this will yield to our tool-set if we remember that it is permitted to *reciprocate* both sides of an equation.

To begin, we identify proper order-of-operation for this equation's evaluation:

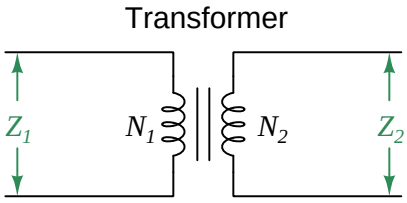
1. Reciprocation: $\frac{1}{R_1}$ and $\frac{1}{R_2}$ and $\frac{1}{R_3}$
2. Addition: $\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$
3. Reciprocation: $\frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$

As a demonstration, we will now solve for R_2:

Step	Equation	Explanation
1	$R_{parallel} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}}$	Equation in its original form
2	$\frac{1}{R_{parallel}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$	Reciprocating both sides
3	$\frac{1}{R_{parallel}} - \frac{1}{R_1} - \frac{1}{R_3} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} - \frac{1}{R_1} - \frac{1}{R_3}$	Subtracting $\frac{1}{R_1}$ and $\frac{1}{R_3}$ from both sides
4	$\frac{1}{R_{parallel}} - \frac{1}{R_1} - \frac{1}{R_3} = \frac{1}{R_2}$	Letting terms cancel out to zero
5	$\frac{1}{\frac{1}{R_{parallel}} - \frac{1}{R_1} - \frac{1}{R_3}} = R_2$	Reciprocating both sides

3.8 Powers and roots

Powers are to roots as multiplication is to division, or addition is to subtraction: they are inverse operations, with one "un-doing" the other. One example of an electrical equation using a power is the relationship between the "turns ratio" of a *transformer* versus its "impedance ratio". None of these quantities are important to grasp[6] for the sake of learning how to apply manipulation techniques:



$$\frac{Z_1}{Z_2} = \left(\frac{N_1}{N_2}\right)^2$$

Suppose we needed to solve this equation for N_2, so that we could determine the number of "turns" necessary on the right-hand side of the transformer. Assessing proper order-of-operations for evaluating the value of the impedance ratio ($\frac{Z_1}{Z_2}$) in this equation results in the following steps:

1. Division: $\frac{N_1}{N_2}$
2. Squaring: $\left(\frac{N_1}{N_2}\right)^2$

[6] If you *must* know, a "transformer" is an electrical component made up of two or more wires coiled around a common form, the number of times each wire is wrapped around that form being called a *turn* (N). "Impedance" is very similar to the concept of resistance, and one of the properties of a transformer is that it makes the impedance of whatever it's connected to appear to be different as sensed from the other coil, the ratio of apparent impedances being equal to the square of the ratio of turns in each coil.

Therefore, to solve for N_2 we must un-do these operations in reverse order:

Step	**Equation**	**Explanation**
1	$\frac{Z_1}{Z_2} = \left(\frac{N_1}{N_2}\right)^2$	Equation in its original form
2	$\sqrt{\frac{Z_1}{Z_2}} = \sqrt{\left(\frac{N_1}{N_2}\right)^2}$	Square-rooting both sides
3	$\sqrt{\frac{Z_1}{Z_2}} = \frac{N_1}{N_2}$	Letting square-root cancel out square
4	$N_2\sqrt{\frac{Z_1}{Z_2}} = \frac{N_1}{N_2}N_2$	Multiplying both sides by N_2
5	$N_2\sqrt{\frac{Z_1}{Z_2}} = (N_1)(1)$	Letting $\frac{N_2}{N_2}$ cancel out to one
6	$N_2\sqrt{\frac{Z_1}{Z_2}} = N_1$	Letting $(N_1)(1)$ simplify to N_1
7	$\frac{N_2\sqrt{\frac{Z_1}{Z_2}}}{\sqrt{\frac{Z_1}{Z_2}}} = \frac{N_1}{\sqrt{\frac{Z_1}{Z_2}}}$	Dividing both sides by $\sqrt{\frac{Z_1}{Z_2}}$
8	$(N_2)(1) = \frac{N_1}{\sqrt{\frac{Z_1}{Z_2}}}$	Letting $\frac{\sqrt{\frac{Z_1}{Z_2}}}{\sqrt{\frac{Z_1}{Z_2}}}$ cancel out to one
9	$N_2 = \frac{N_1}{\sqrt{\frac{Z_1}{Z_2}}}$	Letting $(N_2)(1)$ simplify to N_2

Equations containing roots are just as easy to manipulate: we simply raise both sides of the equation to the appropriate power to cancel the root, at the appropriate step in the reverse-order-of-operations sequence. For example, let us consider the equation describing the velocity of a communications satellite orbiting the Earth:

$$v = \sqrt{g + h}$$

Where,
v = Satellite velocity
g = Acceleration of Earth gravity at sea level
h = Orbit altitude above sea level

In this equation, g is a well-known constant because it is fixed by the mass and radius from the center of the Earth at sea level – only v and h are truly variable. Thus, it makes sense that the practical manipulated form of this equation will be one where we solve for h.

Square-root symbols act as *grouping symbols* just like parentheses[7]. Therefore, proper order of operations requires we first add g and h before computing the square-root. Thus, solving for h demands we first un-do the square-root:

Step	**Equation**	**Explanation**
1	$v = \sqrt{g + h}$	Equation in its original form
2	$v^2 = \left(\sqrt{g + h}\right)^2$	Squaring both sides
3	$v^2 = g + h$	Letting square cancel out square-root
4	$v^2 - g = g + h - g$	Subtracting g from both sides
5	$v^2 - g = h$	Letting $g - g$ cancel out to zero

[7] Another example of a mathematical operation which serves to group terms together is the *fraction bar*. For example, in the fraction $\frac{x+y+5}{z-8}$ the "bar" serves to group $x + y + 5$ together and also to group $z - 8$ together.

3.9 Exponents and logarithms

Exponents are to logarithms as powers are to roots, multiplication is to division, and addition is to subtraction. Once again, these are inverse operations. As such each one may be strategically applied to "un-do" the other in an equation to help solve for variables.

Logarithms tend to be confusing for many students, and so we will explore what logarithms mean before we try to apply them in equation manipulation. First, consider what it means to raise a number such as ten (10) to successively increasing powers:

- $10^{-3} = \frac{1}{1000} = 0.001$
- $10^{-2} = \frac{1}{100} = 0.01$
- $10^{-1} = \frac{1}{10} = 0.1$
- $10^{0} = 1$
- $10^{1} = 10$
- $10^{2} = 100$
- $10^{3} = 1000$

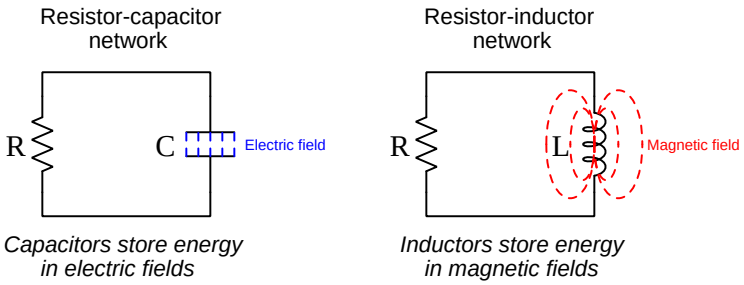
The exponent value simply declares how many times 10 is to be multiplied by itself. Negative exponents refer to reciprocated quantities, and an exponent of zero simply means 1.

A *logarithm* is the converse of an exponent: take the logarithm of a number, and it returns the exponent value you would need as a power of ten to give you that number. The following list illustrates this principle using the same numbers listed previously:

- $\log 0.001 = -3$
- $\log 0.01 = -2$
- $\log 0.1 = -1$
- $\log 1 = 0$
- $\log 10 = 1$
- $\log 100 = 2$
- $\log 1000 = 3$

The logarithms shown here are called *common logarithms* because they assume a "base" value of ten (i.e. $10^{\log x} = x$). Logarithms based on Euler's constant (e) are called *natural logarithms*, and are written as ln instead of log (i.e. $e^{\ln x} = x$).

Exponential functions based on e and their corresponding natural logarithms are widely applied in physical sciences, electricity and electronics included. A prime example of this is inverse-exponential growth and decay in resistor-capacitor and resistor-inductor circuits, examples of each shown below in schematic form:



Capacitors consist of two conductive plates separated by an insulating material, with the electric field lines extending between the two plates. When "charged" with energy, capacitors manifest a voltage between their plates. If connected to a resistor, a "charged" capacitor gradually delivers its stored energy to the resistor which in turn dissipates it in the form of heat. Inductors consist of coils of wire, with magnetic field lines looping outside and through the center of the coil. When "charged" with energy, inductors manifest current through their wire coils. If connected to a resistor, a "charged" inductor gradually delivers its stored energy to the resistor which in turn dissipates it in the form of heat.

In either case, the transfer of energy from the capacitor or inductor to the resistor starts rapidly and then slows to a trickle over time, as described by these exponential equations:

$$V_C = V_0 e^{-\frac{t}{\tau}} \qquad I_L = I_0 e^{-\frac{t}{\tau}}$$

Where,

V_C = Capacitor voltage at time t ; I_L = Inductor current at time t

V_0 = Initial capacitor voltage at time $t = 0$; I_0 = Initial inductor current at time $t = 0$

t = Time

τ = The "time constant" of the R-C or L-R network, $\tau = RC$ or $\tau = \frac{L}{R}$, respectively

Proper order-of-operations to evaluate either of these equations is as follows:

1. Divide: $\frac{t}{\tau}$
2. Negate: $-\frac{t}{\tau}$
3. Exponentiate: $e^{-\frac{t}{\tau}}$
4. Multiply: $V_0 e^{-\frac{t}{\tau}}$ or $I_0 e^{-\frac{t}{\tau}}$

As usual, if we wish to isolate any of the variables on the right-hand side of the equation, we will need to un-do each of these operations in reverse order.

Suppose we wished to manipulate the capacitor's inverse-exponential equation to solve for time (t). A practical application would be to determine how long it would take for the capacitor's voltage to decay given the initial voltage value, capacitance value, and resistance value for the network:

Step	Equation	Explanation
1	$V_C = V_0 e^{-\frac{t}{\tau}}$	Equation in its original form
2	$\frac{V_C}{V_0} = \frac{V_0 e^{-\frac{t}{\tau}}}{V_0}$	Dividing both sides by V_0
3	$\frac{V_C}{V_0} = (1)\left(e^{-\frac{t}{\tau}}\right)$	Letting $\frac{V_0}{V_0}$ cancel out to one
4	$\frac{V_C}{V_0} = e^{-\frac{t}{\tau}}$	Letting $(1)\left(e^{-\frac{t}{\tau}}\right)$ simplify to $e^{-\frac{t}{\tau}}$
5	$\ln \frac{V_C}{V_0} = \ln\left(e^{-\frac{t}{\tau}}\right)$	Taking the natural logarithm of both sides
6	$\ln \frac{V_C}{V_0} = -\frac{t}{\tau}$	Letting the logarithm cancel out the exponential
7	$-\tau \ln \frac{V_C}{V_0} = \left(-\frac{t}{\tau}\right)(-\tau)$	Multiplying both sides by $-\tau$
8	$-\tau \ln \frac{V_C}{V_0} = (t)(1)$	Letting $\left(-\frac{1}{\tau}\right)(-\tau)$ cancel out to one
9	$-\tau \ln \frac{V_C}{V_0} = t$	Letting $(t)(1)$ simplify to t

The procedure is similar when solving for τ. Consider this next example, using the resistor-inductor decay equation:

Step	Equation	Explanation
1	$I_L = I_0 e^{-\frac{t}{\tau}}$	Equation in its original form
2	$\frac{I_L}{I_0} = \frac{I_0 e^{-\frac{t}{\tau}}}{I_0}$	Dividing both sides by I_0
3	$\frac{I_L}{I_0} = (1)\left(e^{-\frac{t}{\tau}}\right)$	Letting $\frac{I_0}{I_0}$ cancel out to one
4	$\frac{I_L}{I_0} = e^{-\frac{t}{\tau}}$	Letting $(1)\left(e^{-\frac{t}{\tau}}\right)$ simplify to $e^{-\frac{t}{\tau}}$
5	$\ln \frac{I_L}{I_0} = \ln\left(e^{-\frac{t}{\tau}}\right)$	Taking the natural logarithm of both sides
6	$\ln \frac{I_L}{I_0} = -\frac{t}{\tau}$	Letting the logarithm cancel out the exponential
7	$\tau \ln \frac{I_L}{I_0} = \left(-\frac{t}{\tau}\right)(\tau)$	Multiplying both sides by τ
8	$\tau \ln \frac{I_L}{I_0} = (-t)(1)$	Letting $\frac{\tau}{\tau}$ cancel out to one
9	$\tau \ln \frac{I_L}{I_0} = -t$	Letting $(-t)(1)$ simplify to $-t$
10	$\frac{\tau \ln \frac{I_L}{I_0}}{\ln \frac{I_L}{I_0}} = \frac{-t}{\ln \frac{I_L}{I_0}}$	Dividing both sides by $\ln \frac{I_L}{I_0}$
11	$(\tau)(1) = \frac{-t}{\ln \frac{I_L}{I_0}}$	Letting $\frac{\ln \frac{I_L}{I_0}}{\ln \frac{I_L}{I_0}}$ cancel out to one
12	$\tau = \frac{-t}{\ln \frac{I_L}{I_0}}$	Letting $(\tau)(1)$ simplify to τ

A different application of logarithms and exponents in electrical circuits is the use of *decibels*. This is a fundamentally logarithmic means for expressing ratios of power values at different points in a circuit. It is frequently used in radio communication circuit calculations because it's an easy way to represent large gains and losses in signal power over a relatively narrow range of numerical values[8].

The equation relating a decibel (dB) figure to a power ratio is as follows:

$$\text{dB} = 10 \log \frac{P_1}{P_2}$$

Correct order-of-operations for evaluating this expression is shown here:

1. Divide: $\frac{P_1}{P_2}$
2. Logarithm: $\log \frac{P_1}{P_2}$
3. Multiplication: $10 \log \frac{P_1}{P_2}$

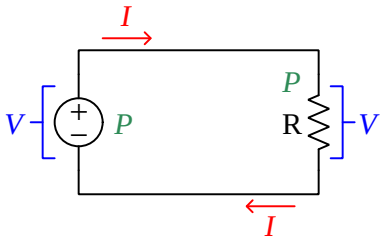
Supposing we wished to solve for P_2 given known values for decibels (dB) and the other power (P_1), we could manipulate this equation using the following steps:

Step	Equation	Explanation
1	$\text{dB} = 10 \log \frac{P_1}{P_2}$	Equation in its original form
2	$\frac{\text{dB}}{10} = \log \frac{P_1}{P_2}$	Dividing both sides by ten
3	$10^{\frac{\text{dB}}{10}} = 10^{\log \frac{P_1}{P_2}}$	Making each side a power of ten
4	$10^{\frac{\text{dB}}{10}} = \frac{P_1}{P_2}$	Letting the power and logarithm cancel
5	$P_2 10^{\frac{\text{dB}}{10}} = \frac{P_1}{P_2} P_2$	Multiplying both sides by P_2
6	$P_2 10^{\frac{\text{dB}}{10}} = (P_1)(1)$	Letting $\frac{P_2}{P_2}$ cancel out to one
7	$P_2 10^{\frac{\text{dB}}{10}} = P_1$	Letting $(P_1)(1)$ simplify to P_1
8	$\frac{P_2 10^{\frac{\text{dB}}{10}}}{10^{\frac{\text{dB}}{10}}} = \frac{P_1}{10^{\frac{\text{dB}}{10}}}$	Dividing both sides by $10^{\frac{\text{dB}}{10}}$
9	$(P_2)(1) = \frac{P_1}{10^{\frac{\text{dB}}{10}}}$	Letting $\frac{10^{\frac{\text{dB}}{10}}}{10^{\frac{\text{dB}}{10}}}$ cancel out to one
10	$P_2 = \frac{P_1}{10^{\frac{\text{dB}}{10}}}$	Letting $(P_2)(1)$ simplify to P_2

[8] This is the same advantage of the Richter scale used to describe earthquake energy, also a logarithmic system.

3.10 Substitution

So far the algebraic manipulation techniques we've seen have been applied to a single equation at a time. Another powerful technique called *substitution* allows us to combine multiple equations to create wholly new equations useful for practical purposes. Our illustrative application will be a simple source and resistor circuit, shown below:



With this application we introduce a new electrical variable: *power* (P). This is a measure of energy transfer rate, measured in Joules per second, otherwise known as *Watts*.

Ohm's Law, which we have already seen, relates voltage and current for a resistance ($V = IR$). A new principle called *Joule's Law* relates power, voltage, and current:

$$P = IV$$

Each of these formulae are useful, but we may find ourselves in a situation where we know values of some of the variables from each, but not enough to use either equation by itself. For example, if we happened to know the value of R and the value of V in this circuit and wished to calculate power (P), neither the Ohm's Law nor Joule's Law equations as given to us here will suffice on their own. Of course, we could manipulate Ohm's Law to solve for current ($I = VoverR$) and then take that computed current value and multiply by the known voltage to arrive at power ($P = IV$), but it would be ideal if we just had one equation with power as a function of voltage and resistance.

Algebraic substitution lets us create new equations out of two or more other equations. Taking the Ohm's Law and Joule's Law equations just mentioned and placing them side-by-side:

$$I = \frac{V}{R} \qquad P = IV$$

If we know from Ohm's Law that current is truly equal to voltage divided by resistance ($I = \frac{V}{R}$), then we may *substitute* $\frac{V}{R}$ in place of I wherever we see I in the Joule's Law equation. Showing this substitution below:

$$P = \left(\frac{V}{R}\right) V$$

Combining the two V variables to simplify this equation yields our desired result:

$$P = \frac{V^2}{R}$$

We may illustrate the practicality of substitution once more by creating an equation solving for power as a function of *current* and resistance. The key is to identify which variable(s) in Joule's Law we do *not* want as dependent variables for power, then substitute the variable(s) we with to have in their place. Starting with the two simple forms of Ohm's Law and Joule's Law:

$$V = IR \qquad P = IV$$

If our goal is to have a power equation in terms of I and R, we must substitute some function of I and R for V in the original Joule's Law equation. As it turns out, Ohm's Law as given to us here is perfectly suited for that role, serving as a definition of V in terms of I and R. Substituting IR for V in Joule's Law:

$$P = I(IR)$$

Re-writing this to combine the two I variables:

$$P = I^2R$$

Similar substitutions may be made to create equations solving for V, I, and R in terms of P. This is left as an exercise to the reader[9].

[9]Hint: there are multiple ways to arrive at such equations. One way is to apply substitution once more to Ohm's and Joule's Laws, this time substituting parts of Joule's Law into Ohm's Law. Another way is to manipulate the two hybrid equations ($P = \frac{V^2}{R}$ and $P = I^2R$) to solve for V, I, or R.

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.

4.1 Algebraic identities

The following equations are true for any value of x whatsoever, and are known as *identities* because the operations shown all leave the value of x unaltered:

Identity of addition:

$$0 + x = x$$

$$x + 0 = x$$

This leads to the *additive inverse* of x which is $-x$ because $x + (-x) = 0$

Identity of subtraction:

$$x - 0 = x$$

Identity of multiplication:

$$1 \times x = x$$

$$x \times 1 = x$$

This leads to the *multiplicative inverse* of x which is $\frac{1}{x}$ because $x \times \frac{1}{x} = 1$ so long as $x \neq 0$

Identity of division:

$$\frac{x}{1} = x$$

4.2 Properties of real numbers

The following equations are basic algebraic properties: rules that all real numbers adhere to.

Associative property:

$$a + (b + c) = (a + b) + c$$

$$a(bc) = (ab)c$$

Commutative property:

$$a + b = b + a$$

$$ab = ba$$

Distributive property:

$$a(b + c) = ab + bc$$

Properties of exponents:

$$a^x a^y = a^{x+y}$$

$$\frac{a^x}{a^y} = a^{x-y}$$

$$(ab)^x = a^x b^x$$

$$\left(\frac{a}{b}\right)^x = \frac{a^x}{b^x}$$

$$(a^x)^y = a^{xy}$$

Properties of roots:

$$(\sqrt[x]{a})^x = a$$

$$\sqrt[x]{a^x} = a \text{ if } a \geq 0$$

$$\sqrt[x]{ab} = \sqrt[x]{a}\sqrt[x]{b}$$

$$\sqrt[x]{\frac{a}{b}} = \frac{\sqrt[x]{a}}{\sqrt[x]{b}}$$

4.3 Dimensional analysis

A powerful technique called *dimensional analysis* helps us check for errors when evaluating formulae related to physical systems. This entails writing units of measurement in place of variables in physical formulae, to see how units cancel and combine to form the appropriate unit(s) of measurement for the result.

For example, let's take the familiar power formula used to calculate power in a simple DC electric circuit:

$$P = IV$$

Where,
P = Power (Watts)
I = Current (Amperes)
V = Voltage (Volts)

Each of the units of measurement in the above formula (Watt, Ampere, Volt) are actually comprised of more fundamental physical units. One Watt of power is one Joule of energy exchanged per second of time. One Ampere of current is one Coulomb of electric charge moving per second. One Volt of potential is one Joule of energy per Coulomb of electric charge. When we write the equation showing these units in their proper orientations, we see that the result (power in Watts, or Joules per second) actually does agree with the units for Amperes and Volts because the unit of electric charge (Coulombs) cancels out. In dimensional analysis we customarily distinguish unit symbols from variables by using non-italicized letters and surrounding each one with square brackets:

$$P = IV$$

$$[\text{Watts}] = [\text{Amperes}] \times [\text{Volts}] \qquad \text{or} \qquad [\text{W}] = [\text{A}][\text{V}]$$

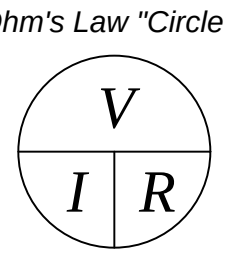
$$\left[\frac{\text{Joules}}{\text{seconds}}\right] = \left[\frac{\text{Coulombs}}{\text{seconds}}\right] \times \left[\frac{\text{Joules}}{\text{Coulombs}}\right] \qquad \text{or} \qquad \left[\frac{\text{J}}{\text{s}}\right] = \left[\frac{\text{C}}{\text{s}}\right] \left[\frac{\text{J}}{\text{C}}\right]$$

Dimensional analysis gives us a way to "check our work" when setting up new formulae for physics- and chemistry-type problems.

4.4 The Ohm's Law Circle and public education in America

In the United States of America it is widely understood that our public primary and secondary schools routinely graduate students who are barely (or not even) literate in either or both grammatical and mathematical senses of the word. Accounts of entry-level college students who cannot read effectively, cannot write a coherent paragraph, and cannot apply elementary principles of algebra are too numerous to mention, and college instructors everywhere lament this sad fact. To be fair, I am not saying *all* or even *most* public high school graduates are woefully incompetent, only enough to pose a major problem for colleges and employers of high school graduates[1].

As evidence of this fact, I present to you the "Ohm's Law Circle" commonly seen on posters in electrical and electronic theory classrooms across the nation:



This is a crutch for students who never learned elementary algebra, and it works like this:

- To solve for V, cover up that variable with your thumb and look at what remains: IR
- To solve for I, cover up that variable with your thumb and look at what remains: $\frac{V}{R}$
- To solve for R, cover up that variable with your thumb and look at what remains: $\frac{V}{I}$

Closely related to this is another variant of the "Circle" showing each and every variation of Ohm's and Joule's Laws solved for every variable in terms of all the others. All you have to do is find the manipulated version of the equation you're looking for, no thumbs required.

As you can tell from reading the Tutorial contained in this module, the task of algebraically manipulating $V = IR$ is not difficult to learn. In fact, in over twenty years of teaching I have yet to encounter a student who cannot learn the elementary algebra necessary to solve for I or R given $V = IR$. In most cases this requires no more than a single day's worth of instruction and practice. Yet, print shops continue to churn out posters of the "Circle" and instructors continue to hang them on their classroom walls, encouraging their students to hobble along with this crutch instead of learning the rudimentary algebra necessary to navigate Ohm's Law without the aid of such gimmicks.

Attention educators: do not cripple your students by absolving them of the responsibility to *reason*. Throw out your Ohm's Law Circle posters and *teach them to think through the math.*

[1] Of course, this poses problems for society extending well beyond college and entry-level employment. For instance, a nation will struggle to maintain a functioning democracy if large portions of its electorate lack the literacy necessary to read and digest complex issues for themselves.

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:

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

int main (void)
{
  float x, y;

  x = 200;
  y = -560.5;

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

  cout << endl;

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

  return 0;
}
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
./myprogram.exe
```

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

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

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

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

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

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

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

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

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`[4] and pressing the Enter key:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
x = 200
y = -560.5

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

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

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

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

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

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

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

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

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

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

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 textitReading For Understanding – How Reading Apprenticeship Improves Disciplinary Learning in Secondary and College Classrooms by Ruth Schoenbach, Cynthia Greenleaf, and Lynn Murphy.

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

General challenges following tutorial reading

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

General follow-up challenges for assigned problems

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

General follow-up challenges for experiments or projects

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

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

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, journal their own reflections on the reading, and discuss in detail their findings with classmates and instructor(s). You should be able to do all of the following after reading any instructional text:

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

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

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

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

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

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

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.

Variable

Substitution

Subscript

Equality

Commutation

Manipulating both sides of an equation

Inverse operations

Order of operations

Reverse order of operations

Factoring

Distribution

Grouping symbol

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[4]" for gaining proficiency in the use of circuit simulation software, and then once that proficiency is gained you will never need to rely[5] on an answer key!

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

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

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, the magnetic permeability of free space value given as $1.25663706212(19) \times 10^{-6}$ H/m represents a center value (i.e. the location parameter) of $1.25663706212 \times 10^{-6}$ Henrys per meter with one standard deviation of uncertainty equal to $0.0000000000019 \times 10^{-6}$ Henrys per meter.

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

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

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

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

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

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

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

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

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

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

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

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

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

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:

	A	B	C	D
1	Distance traveled	46.9	Kilometers	
2	Time elapsed	1.18	Hours	
3	Average speed	= B1 / B2	km/h	
4				
5				

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[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[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 $ax^2 + bx + c$:

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

	A	B
1	x_1	= (-B4 + sqrt((B4^2) - (4*B3*B5))) / (2*B3)
2	x_2	= (-B4 - sqrt((B4^2) - (4*B3*B5))) / (2*B3)
3	a =	9
4	b =	5
5	c =	-2

This example is configured to compute roots[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.

[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.

[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.

[9] Reviewing some algebra here, a *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.

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

$$y = \sqrt{b^2 - 4ac} \qquad z = 2a$$

$$x = \frac{-b \pm y}{z}$$

	A	B	C
1	x_1	= (-B4 + C1) / C2	= sqrt((B4^2) - (4*B3*B5))
2	x_2	= (-B4 - C1) / C2	= 2*B3
3	a =	9	
4	b =	5	
5	c =	-2	

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[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 Solving for n

Solve for n in the following equations:

Equation 1: $-56 = -14n$

Equation 2: $54 - n = 10$

Equation 3: $\frac{4}{n} = 12$

Equation 4: $28 = 2 - n$

Equation 5: $\frac{n}{3} = 12$

Equation 6: $n^3 = 50$

Equation 7: $\frac{10}{n-4} = 5$

Equation 8: $\frac{3n}{11} = 6$

Equation 9: $\sqrt{\frac{10-n}{2}} = 4$

Equation 10: $5n + 8 - 2n = 10$

Next, describe a strategy to *test* your answers to make sure they are correct.

Challenges

- For each of these problems, show more than one procedure for manipulating the equation to arrive at the same answer.

6.2.4 Manipulating the kinetic energy equation

A common equation used in physics relates the kinetic energy, velocity, and mass of a moving object:

$$E_k = \frac{1}{2}mv^2$$

Where,
E_k = Kinetic energy (Joules)
m = Mass (kilograms)
v = Velocity (meters per second)

Manipulate this equation as many times as necessary to express it in terms of all its variables.

Challenges

- Explain how kinetic energy relates to but is distinct from *potential energy*.

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 Identify the mistake, part 1

In this series of algebraic steps there is a **mistake**. Find the mistake and correct it (as well as all subsequent steps):

$$5a - 6ac = 1 - 3a^2b$$

$$\frac{5a - 6ac}{a} = \frac{1 - 3a^2b}{a}$$

$$5 - 6c = 1 - 3ab$$

$$3ab + 5 - 6c = 1$$

Challenges

- Explain why this mistake is a common one for students new to algebraic manipulation.

6.3.2 Identify the mistake, part 2

In this series of algebraic steps there is a **mistake**. Find the mistake and correct it (as well as all subsequent steps):

$$[(xy + x)y + x]y = z$$

$$[x^2y + xy + x]y = z$$

$$x^2y^2 + xy^2 + xy = z$$

Challenges

- Explain why this mistake is a common one for students new to algebraic manipulation.

6.3.3 Identify the mistake, part 3

In this series of algebraic steps there is a **mistake**. Find the mistake and correct it (as well as all subsequent steps):

$$a^2 - 3a = 0$$

$$a^2 = 3a$$

$$a = 3$$

Note: this is a difficult problem to see. Although the answer, 3, does satisfy the original equation, there is still a mistake in the solution!

Challenges

- Demonstrate how a *division-by-zero* error is being committed in the original sequence.

6.3.4 Identify the mistake, part 4

In this series of algebraic steps there is a **mistake**. Find the mistake and correct it (as well as all subsequent steps):

$$a = \frac{b}{c}$$

$$(b)(a) = \frac{b}{c}(b)$$

$$ab = \frac{1}{c}$$

$$c = \frac{1}{ab}$$

Challenges

- There is more than one sequence of steps to solve for c in this equation. Demonstrate at least two mathematically correct sequences that could be used.

Appendix A

Problem-Solving Strategies

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

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

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

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

Appendix B

Instructional philosophy

B.1 First principles of learning

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

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

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

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

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

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

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

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

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

B.2 Proven strategies for instructors

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

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

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

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

B.3 Proven strategies for students

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

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

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

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

B.4 Design of these learning modules

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

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

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

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

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

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

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

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

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

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

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

To high standards of education,

Tony R. Kuphaldt

Appendix C

Tools used

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

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

The `GNU/Linux` computer operating system

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

Bram Moolenaar's `Vim` text editor

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

Donald Knuth's TEX typesetting system

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

Leslie Lamport's LATEX extensions to TEX

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

Tim Edwards' `Xcircuit` drafting program

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

`Gimp` graphic image manipulation program

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

`SPICE` circuit simulation program

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

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

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

`gnuplot` mathematical visualization software

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

`Python` programming language

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

Appendix D

Creative Commons License

Creative Commons Attribution 4.0 International Public License

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

Section 1 – Definitions.

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

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

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

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

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

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

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

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

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

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

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

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

Section 2 – Scope.

a. License grant.

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

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

B. produce, reproduce, and Share Adapted Material.

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

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

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

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

5. Downstream recipients.

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

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

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

b. Other rights.

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

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

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

Section 3 – License Conditions.

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

a. Attribution.

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

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

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

ii. a copyright notice;

iii. a notice that refers to this Public License;

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

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

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

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

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

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

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

Section 4 – Sui Generis Database Rights.

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

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

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

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

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

Section 5 – Disclaimer of Warranties and Limitation of Liability.

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

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

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

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

Section 6 – Term and Termination.

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

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

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

2. upon express reinstatement by the Licensor.

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

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

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

Section 7 – Other Terms and Conditions.

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

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

Section 8 – Interpretation.

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

be made without permission under this Public License.

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

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

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

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

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

Appendix E

References

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.

5 January 2025 – corrected an equation error in the Case Tutorial section, showing how to calculate satellite velocity from acceleration of gravity and orbit altitude.

20 August 2024 – added another section to the Introduction chapter specifically for challenges related to this module's topic.

26-28 May 2024 – added a new Case Tutorial section on the use of variables in electronic calculators, as well as added a Programming References chapter. Also added another section to the Derivations and Technical References chapter on mathematical identities. Also re-formatted the Tutorial's step-by-step algebraic manipulation examples each in table form which is more pleasing to the eye than the free-form way it was originally formatted.

25 June 2024 – divided the Introduction chapter into two sections, one for students and one for instructors, and added content to the instructor section recommending learning outcomes and measures.

13 June 2023 – edited some of the wording where I was using "function" as a synonym for "operation". Also expanded on grouping symbols.

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

12-13 September 2022 – added more equations to the "Solve for n" Quantitative Reasoning question. Also added another page to the Tutorial section on multiplication and division discussing fractions.

22 June 2022 – added more examples of algebraic manipulation (circuit-related) to the Case

Tutorial chapter.

14 June 2022 – minor edits to the Tutorial section on multiplication and division, pointing out that these are inverse functions too. Also added index entries for inverse functions on several pages.

4 February 2022 – added some challenges in various questions to have students demonstrate strategies for checking their work.

30 August 2021 – minor edits to the Tutorial, among other details adding pictorial diagrams of circuits to accompany electrical formulae.

9 July 2021 – replaced some TeX-style italicizing markup with LaTeX-style.

27 January 2021 – added a Case Tutorial.

23 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 questions to the Quantitative and Diagnostic reasoning sections.

1-4 March 2020 – continued writing the Tutorial.

29 February 2020 – document first created (hey, it's a *leap year!*).

Index

Adding quantities to a qualitative problem, 82
Annotating diagrams, 81

C++, 54
Capacitor, 41
Carrier, electric charge, 22
Charge carrier, 22
Checking for exceptions, 82
Checking your work, 82
Code, computer, 91
Common logarithm, 40
Compiler, C++, 54
Computer programming, 53
Computer programming variables, 23
Current, 23

Decay, 41
Dimensional analysis, 50, 81

Edwards, Tim, 92
Electric charge carrier, 22
Energy, 22
Euler's constant, 40

Graph values to solve a problem, 82
Greek letters, 23
Greenleaf, Cynthia, 63
Grouping symbol, 32, 33, 35, 39
Growth, 41

How to teach with these modules, 89
Hwang, Andrew D., 93

Identify given data, 81
Identify relevant principles, 81
Inductor, 41
Intermediate results, 81
Interpreter, Python, 58
Inverse exponential function, 41
Inverse operations, 28, 29, 37, 40
Inverted instruction, 89

Java, 55
Joule's Law, 44

Kirchhoff's Voltage Law, 34
Knuth, Donald, 92

Lamport, Leslie, 92
Language, mathematics as, 22
Limiting cases, 82
Logarithm, 40

Metacognition, 68
Moolenaar, Bram, 91
Murphy, Lynn, 63

Natural logarithm, 40

Ohm, 22
Ohm's Law, 32, 34
Open-source, 91
Operations, inverse, 28, 29, 37, 40
Order of operations, 32

PEMDAS, 32
Problem-solving: annotate diagrams, 81
Problem-solving: check for exceptions, 82
Problem-solving: checking work, 82
Problem-solving: dimensional analysis, 81
Problem-solving: graph values, 82
Problem-solving: identify given data, 81
Problem-solving: identify relevant principles, 81
Problem-solving: interpret intermediate results, 81
Problem-solving: limiting cases, 82

Problem-solving: qualitative to quantitative, 82
Problem-solving: quantitative to qualitative, 82
Problem-solving: reductio ad absurdum, 82
Problem-solving: simplify the system, 81
Problem-solving: thought experiment, 81
Problem-solving: track units of measurement, 81
Problem-solving: visually represent the system, 81
Problem-solving: work in reverse, 82
Programming, computer, 53
Python, 58

Qualitatively approaching a quantitative problem, 82

Reading Apprenticeship, 63
Reductio ad absurdum, 82, 88, 89
Resistance, 22
Resistor, 22
Richter scale, 43

Schoenbach, Ruth, 63
Scientific method, 68
Series, 22
Simplifying a system, 81
Socrates, 88
Socratic dialogue, 89
Source code, 54
SPICE, 63
Stallman, Richard, 91
Subscript, 23
Substitution, 21
Symbol, grouping, 32, 33, 35, 39

Thought experiment, 81
Torvalds, Linus, 91
Transformer, 31, 37

Units of measurement, 81

Variable, 21
Variables, computer programming, 23
Visualizing a system, 81
Voltage, 23

Whitespace, C++, 54, 55
Whitespace, Python, 61
Work in reverse to solve a problem, 82
WYSIWYG, 91, 92