1 Introduction

In this chapter, we will get started with engineering computing by introducing the topic and setting up our tools to apply it throughout the rest of this book. Furthermore, the primary elements of the Python programming language are introduced.

1.1 Introduction

Engineering computing is the type of computing engineers use to design and analyze engineering systems. It is similar to **scientific computing**, and engineering and scientific computing share many tools, but the techniques and objectives of engineers differ from those of scientists. The computing tools of an engineer have become essential to the profession, and this has become increasingly true over the past few decades. Although the field can be considered inclusive of spreadsheet software and computer-aided design (CAD), we choose to focus on **computer programming**, writing text instructions called **programs** for a computer to perform calculations and store the results. We leave aside the important topic of **real-time computing** in which the computer becomes part of the engineering system; although it is increasingly important for engineers, it comes with a host of considerations that are a distraction from engineering computing.

There are many **computer languages**, but **programming languages** are those that are easiest for humans to use. Low-level programming languages give more control over the computer hardware and can be more compact; high-level languages have libraries and features that make programming easier. The most common programming languages used for engineering computing are the open-source **Python** language and the proprietary **MATLAB** language. Both are powerful languages with large userbases, but Python has been gaining in popularity in recent years. MATLAB has many built-in tools for engineering computation and "toolboxes" that extend its functionality beyond the base language. Python, on the other hand, does not have many built-in tools for engineering computation; however, it has

code libraries called **packages** that can be used for engineering computing. We will use a few key packages in this book, and there are many more available on the Internet, especially at https://engineering-computing.ricopic.one (Python Community 2024b).

There are several classes of engineering analysis performed with engineering computing. The following list captures the majority of problems, but it is far from exhaustive.

Numerical Analysis Many engineering problems can be approached by performing numerical calculations. These can be challenging or even intractable to perform manually when the problem requires many such calculations. **Numerical analysis** use systematic procedures called **algorithms** to perform the calculations with a computer. These techniques use the computer to perform, store, and organize these calculations. This class of problems, sometimes called **simulation**, comprise the majority of engineering computing problems.

Symbolic Analysis **Symbolic analysis**, sometimes called "analytic" as opposed to "numerical," is closely related to mathematics. Mathematical variables can be directly manipulated via algebraic methods (including those of calculus). Computer programs that treat these variables symbolically are called **computer algebra systems (CASs)**. Although these systems can be somewhat cumbersome, for complex problems they provide distinct advantages.

Graphical Analysis Visualization techniques are an important aspect of engineering analysis. **Graphics**—often graphs, plots, and charts—can be generated by programs much more quickly and accurately than they can be created manually. The result of an engineering computing program is often a graphic.

In this book, we will introduce all three classes of analysis.

1.2   The Development System

In general, a computer **development system** is one that is used to write, execute, debug, and deploy computer programs. Our development system is comprised of the following components:

- A personal computer (PC) (e.g., one running the Windows, macOS, or Linux operating system)
- The Anaconda distribution of the Python 3 software
- The **Spyder integrated development environment (IDE)**

An IDE is a software application in which a programmer can write, execute, and debug their programs.