1.1 The Fundamental Problems

This section states the fundamental problems of introductory applied linear algebra. These problems are critically important in many fields of Science, Technology, Engineering, and Mathematics. In our discussion, we focus on the structure of each problem, paying special attention to the given information and the unknown quantities desired. Finally, we introduce a number of problems arising in advanced applied linear algebra. As is to be expected, by learning how to solve the fundamental problems of introductory linear algebra, we set the foundations for the study of these advanced problems.

One of the most beautiful features of applied linear algebra is the deep connection between practical applications and mathematical theory. To make these connections explicit, we need to translate physical phenomena into useful mathematical concepts. In this textbook, we transform a variety of contexts arising in our observable world into one of these fundamental problems. We also present a plethora of techniques to solve each of these problems and discuss important challenges in using computers to find solutions via the algorithms we discuss.

By the end of this section, we expect the reader to be able to state the four fundamental problems, discuss the known and unknown quantities from each problem, and identify at least three major application areas that give rise to these problems.

The Fundamental Problems

We begin with the most important and often most nuanced problem.

> **Problem 0: The Applied Math Modeling Problem**
>
> Find and solve real-world problems using the applied math modeling process. In linear algebra, we focus on transforming our real-world problem into an ideal mathematical model that is stated in terms of vectors and matrices. Then, we attempt to represent the most pertinent aspects of our problem using at least one of the linear-algebraic problems stated below.

Applied mathematics focuses on solving real-world problems using mathematical models. In this book, we define a **real-world problem** to be a set of questions used to illicit information that can be stated using quantitative data observable by humans beings. Very often, real-world problems are created by scientific and entrepreneurial thinkers in order to address some specific need of a society. Many real-world problems arise in the study of the physical, biological, chemical or social spheres of our existence. Let's investigate the nature of applied mathematical modeling process. By doing so, we can learn to identify the different patterns has four distinct phases and is iterative.

The first phase of this process begins by identifying a *real-world problem*, which we define as any problem that matters in our lives and that can be studied by making observations. The practice of observing problem dynamics is more formally known as the collection of measured data though this distinction usually exists only in the mind of trained scientists. One of the fascinating features of real-world problems is that by identifying and imagining the problem, we likely develop some idea of what a meaningful solution might be. However, from the standpoint of applied mathematics, a valuable real-world problem usually includes major obstacles that block the path between the problem statement and our desired solution. In order to earn healthy paychecks, applied mathematicians pray that such obstacles are so prohibitive that spending months or years doing tedious mathematical analysis proves to be much more productive and economical than trying to solve the problem using brute force.

Assuming that we've identified a real-world problem in which mathematical modeling seems more hopeful than physical labor, we move on to the second phase of the applied mathematical modeling process. In this stage, we *mathematize* our real-world problem by transforming measurable data and nonmathematical objects into a collection of relevant mathematical ideas. This might include introducing useful mathematical notation, defining relevant variables, imposing appropriate mathematical assumptions, and focusing on a subset of important problem characteristics while ignoring other aspects of the problem.

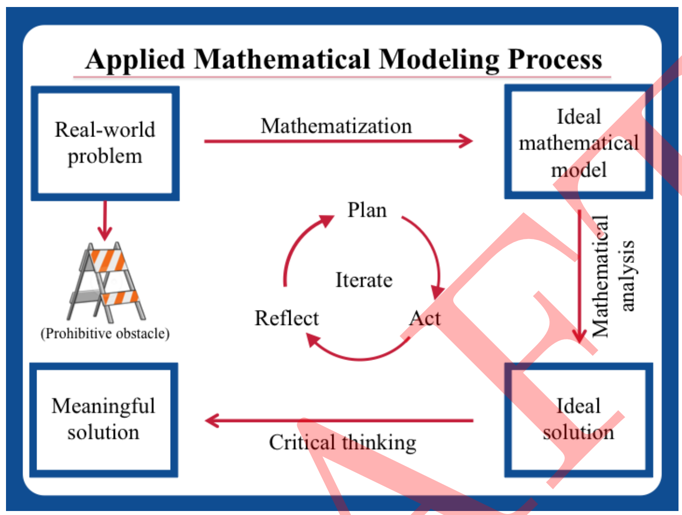

Figure 1: Diagram of the applied mathematical modeling process. Applied modeling begins with a real-world problem.

As we see in this paper, there are many subtleties and nuances involved in mathematizing a real-world problem to produce a corresponding ideal mathematical model that actually models key aspects of the problem. In good applied mathematics, this is where interdisciplinary teams prove to be invaluable. Each professional is trained in a particular field. By combining talents with a growth mindset and some luck, great learning might result in a useful modeling scheme.

Once we've decided on an ideal mathematical modeling scheme, phase three of this process involves a painstaking effort of mathematical analysis. Such analysis includes a search for the most powerful mathematical theorems and tools that can be brought to bear on our problem. For a good real-world problem, this type of applied mathematical analysis requires deep thinking about the mathematical objects we use in our model and many hours of iterated failure until we converge on the most potent mathematical ideas. A good motto for this stage is "error and error and error, but less and less and less."

The dream of applied mathematics is that we might analyze our ideal model using a suite of technical mathematical results and produce an ideal solution to this problem. The fantasy of applied mathematics is that the ideal solution we produce via mathematical analysis leads to valuable progress in discovering aspects of the meaningful real-world solution that we so desire.

One of the major reasons that we force students in STEM fields to take college coursework in mathematics is that we want our students to be able to use mathematics to accurately analyze problems that may arise in their future careers. However, a major tragedy of many college mathematics classes is that most of the applied mathematical modeling process is hidden from view. Instead, students in our college math classes focus only on mathematical analysis techniques that are required to analyze an ideal mathematical model and produce a corresponding ideal solution.

Indeed, it is very seldom that young students get a chance to participate in all four steps of the applied modeling process prior to the end of their undergraduate degree. It is even more rare that students observe this process in action while completing their lower division courses in mathematics. As college instructors, we see evidence of this missed opportunity when our students ask "when will I use this material?" or "how is this applicable in the real world?"

The modeling activity presented in this paper is designed to provide a compelling answer to these questions. This is part of the author's larger develop program to enrich our ability to teach introductory courses in applied linear algebra. The ultimate goal of these types of modeling activities is to provide students access to valuable mathematical modeling experience that illustrate all four phases of the applied modeling process. The hope is that these types of activities enriches students' understanding of how linear algebra is used in practice. The particular activity presented in this paper is especially relevant for students who want to earn degrees in electrical engineering, computer science, mechanical engineering, physics, or applied mathematics.

We continue with a discussion of the most basic problem in linear algebra: the matrix-vector multiplication problem.

Problem 1: The Matrix-Vector Multiplication Problem

Let $m, n \in \mathbb{N}$. Let $A \in \mathbb{R}^{m \times n}$ be a given matrix and $\mathbf{x} \in \mathbb{R}^n$ be a given vector. Then the matrix-vector multiplication problem is to find an unknown vector $\mathbf{b} \in \mathbb{R}^m$ such that

$$A \cdot \mathbf{x} = \mathbf{b}$$

Matrix-vector multiplication is a "forward problem." To understand this terminology, let's define the function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ given by $f(\mathbf{x}) = A \cdot \mathbf{x}$. Based on this definition, we will see that function f satisfies the following:

- the domain of f is $\mathbb{R}^n$
- the codomain f is $\mathbb{R}^m$
- the range of f is contained in $\mathbb{R}^m$ but may not be equal to $\mathbb{R}^m$ depending on the entries in the columns of matrix A

In this case, the function f is defined as a matrix-vector product. Any matrix A implicitly defines the matrix-vector product function f, as illustrated above. Matrix-vector multiplication is a forward problem because we start with a given input $\mathbf{x}$ in the domain $\mathbb{R}^n$ and move forward into the range to find the output vector $\mathbf{b}$. When solving the matrix-vector multiplication problem, we map from the domain forward into the range. Hence, we call this a forward problem.

In order to create such a problem, we need to construct a matrix A that models some component of a physical situation. Then, we need to determine an input vector $\mathbf{x}$ representing a known input state from our physical problem. The unknown output $\mathbf{b} \in \mathbb{R}^m$ is a solution to this problem if it can calculated using matrix-vector multiplication.

There are a number of applications that directly rely on matrix-vector products. For example, when defining 2D and 3D computer graphics, we can use matrix-vector multiplication to manipulate shapes and create animations. In addition, we utilize matrix-vector multiplication to discretize continuous functions and approximate derivatives using finite arithmetic. Matrix-vector products can be used to calculate forces applied to a system of coupled masses, voltages across resistors with known currents in an electric circuit, and to manipulate digital images.

The matrix-vector multiplication problem is also intimately connected with the linear-system problem. While matrix-vector multiplication represents a forward problem, linear systems of equations constitute the associated backward problem. Finally, matrix-matrix multiplication is built from matrix-vector products. Since almost all matrix computation algorithms rely on matrix-matrix multiplication, we see that matrix-vector products form the foundation for all problems in applied linear algebra. Indeed, every application we describe below is dependent on matrix-vector multiplication.

Problem 2A: The Nonsingular Linear-Systems Problem

Let $n \in \mathbb{N}$. Let $A \in \mathbb{R}^{n\times n}$ be a given nonsingular matrix and $\mathbf{b} \in \mathbb{R}^n$ be a given vector. Then the nonsingular linear-systems problem is to find an unknown vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$A \cdot \mathbf{x} = \mathbf{b}.$$

Just like matrix-vector multiplication, we can describe the nonsingular linear-systems problem using the function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ defined as $f(\mathbf{x}) = A \cdot \mathbf{x}$. Based on this definition, we have:

- the domain of f is $\mathbb{R}^n$
- the codomain f is $\mathbb{R}^n$.
- the range of f is $\mathbb{R}^n$ due to the structure of the nonsingular matrix A

The nonsingular linear-systems problem is a backward problem because we start with the function description, defined by the nonsingular matrix A, and we are given one specific output vector $\mathbf{b}$ in the range of the function f. From this information, we are asked to find all possible vectors $\mathbf{x}$ in the domain of our function such that

$$f(\mathbf{x}) = \mathbf{b}.$$

When solving the linear-systems problem, we begin in the range and work our way backwards into the domain. Hence, we call this a backward, or inverse, problem.

To craft a nonsingular linear-systems problem, we need to construct a square, nonsingular matrix A that describes a physical phenomenon. As we will see, there are many real-world applications that result in these special types of matrices. With our nonsingular matrix A in hand, we also need to determine a vector $\mathbf{b}$ that represents an output state in our system. We construct our model in such a way that output vector $\mathbf{b}$ can be written as the product of our matrix A with some unknown input vector $\mathbf{x} \in \mathbb{R}^n$. The solution to the nonsingular linear-systems problem is any input vector that results in output $\mathbf{b}$ after a matrix-vector multiplication with matrix A.

A major similarity between matrix-vector multiplication and the nonsingular linear-systems problem is that both depend on matrix-vector multiplication. The process of solving the former problem is simply to calculate a product. To solve the later problem, we "reverse engineer" our matrix-vector product in order to find input vectors that produce a given output.

One of the major differences between the matrix-vector multiplication problem and the nonsingular linear-systems problem is in the assumptions on the matrix A. For general matrix-vector products, we need only create a rectangular $m \times n$ matrix, where m may not be equal to n. No additional structure on the matrix A is needed in order to solve this problem. On the other hand, for the nonsingular linear-systems problem, we need to create a matrix with the same number of rows as columns and this matrix must have very special column structure.

Not all linear-systems problems are nonsingular. Indeed, one of the advanced problems in applied linear algebra is known as the general linear-systems problem and focuses on solving linear systems with general rectangular coefficient matrices. Because most of the techniques for solving the general linear systems depend on the solution methods we construct for nonsingular linear systems, we focus our attention most heavily on nonsingular linear systems of equations.

The nonsingular linear-systems problem arises in a wide variety of applications. Most of these depend on interconnections between many different objects. For example, we use nonsingular linear systems to analyze a collection of masses connected by springs, a so called mass-spring chain. Using similar matrix algebra but focusing on laws that govern electronics, we construct nonsingular linear-systems problems to calculate node voltage potentials in an electric circuit containing only resistors and known DC voltage and current sources. Similarly, we use nonsingular linear systems to discretize ODEs and PDEs. By doing so, we arrive at a finite approximation to our desired solutions. Very common numerical algorithms that rely on this approach are known as finite difference or finite element methods. These can be employed to analyze the static behavior of structures like buildings and bridges under known loads.

Further, we can use nonsingular linear-systems to design continuous piecewise polynomial functions that interpolate a given set of data. This technique is known as polynomial spline interpolation of which the most famous variety are cubic splines. Polynomial splines can be applied in computer graphics, the design of roller coasters and the construction of any object with a smooth surface that exits in 2D or 3D space. The preceding list are just a few of the many application areas that give rise to nonsingular linear systems.

We can make a convincing argument that the nonsingular-linear systems problem is the most fundamental problem of modern day applied mathematics. The central supporting point for this claim is that computers make it possible to solve nonsingular linear-systems problems quickly and accurately. Moreover, since the dawn of digital computation in the mid 1940's, continual improvements in computer hardware and software have enabled STEM professionals to solve larger and larger linear systems of equations. In the 1950's, state of the art computers costing millions of dollars could solve 20×20 linear-systems problems in less than a weeks time. By the 1970's, it was standard to benchmark new computers in solving linear systems with 100 equations and 100 unknowns in about a 24 hour period. Today, a cheap laptop computer can solve linear-systems problems with dimensions in the thousands in a few minutes.

These rapid improvements in computer technology have lead to a boom in linear-systems solvers. Engineers and scientist working in any application area in which linear-system problems naturally arise rely on computers in their daily work. In fact, the ease, power and availability of computer methods to solve linear systems have prompted engineers and scientists working on problems where linear system may not naturally arise to actively search for ways to reframe such problems into the linear-systems paradigm. The idea of this approach is to transform a modeling problem into a nonsingular linear-systems problem, solve the linear system of equations using a computer and use the computed solution to make advanced towards the solution of the original problem.

In this textbook, we focus not only on applications and theory but also introduce a number of tools needed to find solutions using computers. The reason that introductory applied linear algebra is a pre-requisite course in most STEM majors is due to the fact that many computational methods used in STEM fields rely on solutions to linear systems of equation. Thus, any student who begins her study of linear algebra should necessarily be introduced to some theory of computation. If and when you go on to use this theory in your future, you will most likely be solving linear algebraic problems using a computer. For that reason, it is best if you start your journey in linear algebra with a practical focus. Understand that the power of applied linear algebra is intertwined with our ability to implement computer algorithms to solve the the fundamental problems of linear algebra.

The nonsingular linear-systems problem is a special case of a much more general problem type.

Problem 2B: The General Linear-Systems Problem

Let $m, n \in \mathbb{N}$. Let $A \in \mathbb{R}^{m \times n}$ be a given rectangular matrix and $\mathbf{b} \in \mathbb{R}^m$ be a given vector. Then the general linear-systems problem is to find an unknown vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$A \cdot \mathbf{x} = \mathbf{b}.$$

We break these into two separate problems in order to specify which paradigm to use. Many of the technique used to solve general linear-systems problems are built on intuition from both matrix-vector multiplication and the nonsingular linear systems. In this text, we spend much more energy motivating and analyzing the nonsingular linear-systems problem. This is because the general linear-systems problem can be transformed into a nonsingular linear-systems problem combined with a series of matrix-vector multiplication problems. Once again, we see transform a hard problem into a nonsingular linear-systems problem.

A break down of the analogy of multiplication and division occurs when our input and output spaces are not $\mathbb{R}$. When working with multiplication and division, each unique input can get only one unique output. Similar, for any real output, we can find an input, called a quotient. However, in the matrix-vector product paradigm, there may be elements in the codomain that do not have a pre-image in the domain. Put another way, there are a number of matrices whose underlying structure results in a linear-systems type problem that has no exact solution. In this case, we state a more general problem type know as a least-squares problem.

Problem 3: The Full-Rank Least-Squares Problem

Let $m, n \in \mathbb{N}$ with $m \geq n$. Let $A \in \mathbb{R}^{m\times n}$ be a given full-rank matrix and $\mathbf{b} \in \mathbb{R}^m$ be a given vector. Then the full-rank least-squares problem is to find all $\mathbf{x} \in \mathbb{R}^n$ to minimize the 2−norm of the residual vector:

$$\|A \cdot \mathbf{x} - \mathbf{b}\|_2$$

Again, we can describe the full-rank least-squares problem using the function $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ with $f(\mathbf{x}) = A \cdot \mathbf{x}$. Based on this definition, we have:

- the domain of f is $\mathbb{R}^n$
- the codomain f is $\mathbb{R}^m$.
- the range of f is a subset of $\mathbb{R}^m$ and most likely not equal to $\mathbb{R}^m$

Like the linear-systems problem, the least-squares problem starts with a matrix A. Unlike the linear-systems problem, the given vector $\mathbf{b}$ in the least-squares problem is not necessarily in the range of our function. In fact, the only requirement on $\mathbf{b}$ is that it is in the codomain. In most meaningful least-squares problems, there will be no input vector $\mathbf{x}$ in the domain such that the equality $f(\mathbf{x}) = \mathbf{b}$ holds exactly. As we will see, to solve the least-squares problem, we produce an optimal input vector that makes $f(\mathbf{x}) \approx \mathbf{b}$. We do so by minimizing the distance between the output vector $\mathbf{b}$ and the range of function f.

In order to construct a least-squares problem, we need to create a full-rank, rectangular matrix A. The number of rows of A should be greater than or equal to the number of columns of this matrix. We also need to create an output vector $\mathbf{b}$ representing some state in our modeled phenomenon. The solution of the full-rank least-squares problem is a vector $\mathbf{x} \in \mathbb{R}^n$ such that $f(\mathbf{x})$ is as close as possible to $\mathbf{b}$.

Again, the least-squares problem depends on matrix-vector multiplication. This is a backward problem that results in an approximate answer not exactly equal to our original desired output.

The full-rank least-squares problem is used to solve a variety of applied modeling problems. Attempting to find a sum of continuous functions that can be used to interpolate measured data that contains measurement error results in a least squares problem. With mild conditions on the input data, we can guarantee that the matrix A will be full rank. The least-squares problem is also used in geographical surveying, image processing, GPS calculations, statistical regression, the Procrustes problem and speech recognition.

As we will see, one of the basic methods to solve the least-squares problem is to transform the minimization problem into a square linear-systems problem. This is yet another example of transforming a hard problem into a square linear-systems problem. We also study a host of methods based on orthogonality that set the foundations for many of today's most powerful numerical linear algebra methods for computer algorithms.

Problem 4: The Standard Eigenvalue Problem

Let $n \in \mathbb{N}$. Let $A \in \mathbb{R}^{n\times n}$ be a given square matrix. The eigenvalue problem is to find all scalars λ and all possible nonzero $n \times 1$ vectors $\mathbf{x}$ such that

$$A \cdot \mathbf{x} = \lambda \, \mathbf{x}$$

A very classic and powerful approach to constructing standard eigenvalue problems is to develop said theory in the context of analyzing coupled harmonic oscillators. In fact, the standard eigenvalue problem is central to the entire field of vibrational mechanics and sets the foundation for many applied topics in the physical sciences. From this perspective, it is useful to think about the eigenvalue problems as describing and encoding a special class of differential equations. Any useful treatment of eigenvalue theory should include a detailed introduction to differential equation modeling and the significance of solutions to differential equations.

Examples of modeling problems that involve differential equations include the analysis of the displacements of masses in a mass-spring chain, RCL circuit analysis, the solution to partial differential equations, analysis of vibrations, facial recognition software, and principal component analysis.

Advanced Problems in Linear Algebra

In addition to the four fundamental problems listed above, there are a number of advanced problems in linear algebra. We list these below for completeness. This textbook presents these problems in later chapters. Such material is designed for an upper-division course in applied linear algebra.

Problem 5: The Rank-Deficient Least-Squares Problem

Let $m, n \in \mathbb{N}$. Let $A \in \mathbb{R}^{m\times n}$ be a given matrix that does not have full column rank and $\mathbf{b} \in \mathbb{R}^m$ be a given vector. Then the rank deficient least-squares problem is to find all $\mathbf{x} \in \mathbb{R}^n$ to minimize the $2-$norm of the residual vector:

$$\|A\mathbf{x} - \mathbf{b}\|_2$$

Problem 6: The Generalize Eigenvalue Problem

Let $n \in \mathbb{N}$. Let $A, B \in \mathbb{R}^{n\times n}$ be a given square matrices. The generalized eigenvalue problem is to find scalars λ and nonzero vectors $\mathbf{x}$ such that

$$A\mathbf{x} = \lambda \, B \, \mathbf{x}$$

Problem 7: The Quadratic Eigenvalue Problem

Let $n \in \mathbb{N}$. Let $A, B \in \mathbb{R}^{n\times n}$ be a given square matrices. The generalized eigenvalue problem is to find scalars λ and nonzero vectors $\mathbf{x}$ such that

$$\lambda^2 \, M \, \mathbf{x} + \lambda \, B \, \mathbf{x} + \; K \, \mathbf{x} + = \mathbf{0}$$

- Mass-Spring Systems
- RLC Circuit analysis (using Laplace transforms)

Problem 8: The Singular Value Decomposition Problem

Let $m, n \in \mathbb{N}$. Let $A \in \mathbb{R}^{m\times n}$ be a given matrix. The singular value problem problem is to find unitary matrices U, V and a diagonal matrix Σ such that

$$A = U \, \Sigma \, V^*$$

The Four Fundamental Problems- Suggested Exercises:

1. Discuss the idea of a forward problem and backward problem. Which of the four fundamental problems can be characterized as a forward problem and which as a backward problem?

2. Compare and contrast the matrix-vector multiplication problem, the nonsingular linear-systems problem, and the least-squares problem? How are these problems similar? How are they different?

3. Identify three possible application areas that give rise to each of the four fundamental problems of applied linear algebra.