Introduction

This textbook is written for the science and engineering students of a typical university. It teaches an introductory course in linear algebra for undergraduates. It does not assume signifigant, prior proof skills on the part of the students, and has exercises designed to help teach proof-building skills. It does assume that the student has access to the MATLAB software package [4], and that the instructor is willing to implement computer-assisted instruction.

This textbook is written at a conceptual level comparable to that of Anton[**1**], a bit below Strang [**7**] and Bretscher [**2**]. It spends more time explaining the motivation for developing many of the techniques of linear algebra, and often speaks from the perspective of someone who is thinking his way through a derivation. This is to encourage private speech in the student, the "self-talk" which the psychologist Vygotsky theorized is helpful when dealing with complicated procedures, as it facilitates deductive reasoning[**6**].

Along with this, this textbook includes significantly more examples than most other linear algebra texts. It emphasizes operational knowledge of linear algebra; however, it does not forget the theory. Instead, its intent is to develop the average student into one who fully understands the principles and underlying structures of the material as he uses it, rather than just be able to do matrix computations.

To accomplish this, it begins with a discussion of linear transformations and vector spaces, well before linear systems are introduced. This contrasts many texts, such as Bretscher and Anton, which begin with discussions of linear systems. This textbook, instead, sees the notion of the linear transformation as the driving concept of the course, and ties this theme to most of the main concepts in the course. Not only does this emphasize the crucial notion of linearity, but this arrangement of the topics also allows students to learn the more basic MATLAB commands before being introduced to the more complicated procedures and decompositions.

Instructional technology is another main theme of this text. Instructional technology refers to computer technology which facilitates the student's understanding of concept. This can come in a variety of forms, from the student use of hand-held calculators, to instructor use of classroom visual displays, to numerical experiments run on a large mainframe. This text concerns itself with, among other things, the use of the MATLAB software package in exercises, to serve primarily as a tutorial, but also for discovery-based learning.

Many linear algebra texts avoid the use of software packages, favoring a classical approach. This is not an intrinsic flaw, but a reluctance to take advantage of all available tools. One thing to keep in mind is that the mere inclusion of computer exercises will not necessarily educate the student. In fact, it is quite possible for instructional technology, used incorrectly, to weaken a student's grasp of concept.

One misuse of instructional technology is to provide programming codes which independently solve problems. This refers to programs which, when run and given basic input, immediately return a solution to the student, with no outward indication of the process. Strang's text provides a myriad of MATLAB programs, often not attached to specific exercises, whose presence only serves one purpose – to allow students to skip the work and jump immediately to the solution of a problem. While these programs are beneficial to those who already understand the concept, they do nothing to help a student learn the concept.

The ATLAST linear algebra workbook [**3**] offers an abundant supply of very creative MATLAB-based projects for students, and at an appropriate level. It also provides some clever programs, ready to input into MATLAB, which allow students to do matrix manipulations such as performing elementary row operations on matrices. It is an excellent source of problems for students who already understand the material. In essence, ATLAST provides simulations which reinforce material via repetitive conditioning. What it lacks is a strong base of tutorial problems, designed to teach the concept rather than just offer drills or projects with which the students may practice.

The main difference between this textbook and other textbooks is that the computer exercises in this text are primarily tutorial in nature, as opposed to using simulations or excessive drillwork. The majority of MATLAB exercises in this text are designed to do one or more of the following:

1. Allow the student to explore and discover a particular concept, without being distracted by arithmetic and simple operations;

2. Foster a perspective or mindset, which will be useful later indeveloping another concept;

3. Provide an interesting way to practice a procedure.

It includes a few exercises which teach MATLAB commands, and of course it includes exercises in which student merely practices what he has learned, but the main theme of the MATLAB exercises is to encourage exploration, leading to discovery learning.

For example, MATLAB exercise 1.2.1 teaches the concept of spanning a vector space by having the students graphically visualize specified combinations of vectors on a plane. MATLAB gives informative and immediate feedback to the students as they work, allowing the students to discover how given vectors can be "reached" with a combination of basis vectors. They gradually realize how any vector can be similarly "reached," through a guided exploration, with the software handling any distracting arithmetic. Such exercises reinforce the theory in a student's mind, and teach them that they can often get a better grasp on abstract theory through concrete experimentation. It is important that students learn this lesson early, while the theory is still relatively simple.

There are many other minor distinctive characteristics of this textbook. Most notable of these is that it encourages students to use extra row operations, during Gaussian elimination, to produce ones in the pivot positions. While this requires a greater number of steps in performing elimination, it has been my experience that students make far fewer computational errors by this method when performing elimination by hand, which helps students who are learning elimination for the first time.

A pleasant surprise was that, by adopting this method of reduction, it also becomes simpler to form the matrix L when producing an LU-decomposition of a matrix. L is assembled quickly by using specific numbers which I have named *contributors* (Section 3.4).

Another trait of the dissertation text is the inclusion of proof-building exercises. Some exercises feature theorems with complicated proofs, broken down into more easily digestible parts. The exercises ask the student to prove each part, and then assemble them into a coherent proof. This is designed to help students spot the key ideas about which proofs are constructed, so that they will know what to look for when composing proofs in the future.

This textbook teaches the concept of the determinant by deconstructing the determinant into its elementary products, and then slowly deriving the properties of the determinant by a series of leading observations. It introduces the original notion of *sources* (Section 5.2), which provide an intermediate step between the determinant and cofactors. This extra step makes the method of cofactors easier to demonstrate, and helps in deriving Cramer's Rule.

Several applications of linear algebra techniques appear in this textbook. Rotation of coordinate axes (Section 4.3), least-squares approximation (Section 4.5), and systems of linear ordinary differential equations with constant coefficients (Section 5.4) receive special emphasis. Exercises throughout the text present other applications, such as the QR-algorithm (Section 4.4, 5.3) and singular value decomposition (Section 5.3).