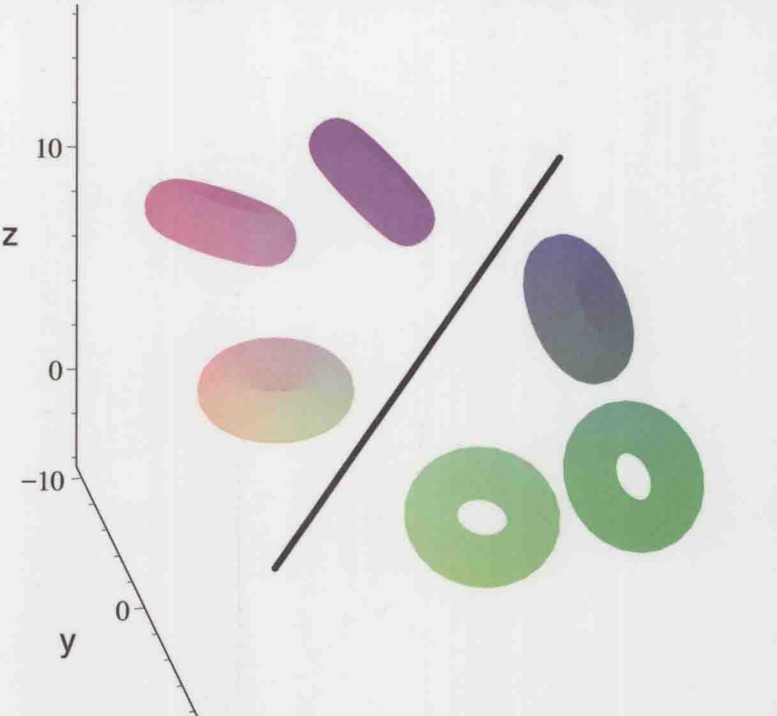
Pure and Applied Mathematics: A Wiley Series of Texts, Monographs, and Tracts

Principles of Linear Algebra with Maple™

Kenneth Shiskowski
Karl Frinkle



WILEY

WWW.
LINK AVAILABLE

Principles of Linear Algebra With *Maple*™

Kenneth Shiskowski
Eastern Michigan University

Karl Frinkle
Southeastern Oklahoma State University

WILEY

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Shiskowski, Kenneth, 1954–
Linear algebra using Maple / Kenneth Shiskowski, Karl Frinkle.
p. cm. — (Pure and applied mathematics)
On t.p. the registered trademark symbol "TM" is superscript following "Maple" in the title.
Includes index.
ISBN 978-0-470-63759-3 (cloth)
1. Algebras, Linear—Data processing. 2. Maple (Computer file) I. Frinkle, Karl, 1977– II. Title.
QA185.D37S45 2010
512'.50285--dc22 2010013923

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

PURE AND APPLIED MATHEMATICS

A Wiley Series of Texts, Monographs, and Tracts

A complete list of the titles in this series appears at the end of this volume.

Preface

This book is an attempt to cross the gap between beginning linear algebra and the computational linear algebra one encounters more frequently in applied settings. The underlying theory behind many topics in the field of linear algebra is relatively simple to grasp, however, to actually apply this knowledge to nontrivial problems becomes computationally intensive. To do these computations by hand would be tedious at best, and many times simply unrealistic. Furthermore, attempting to solve such problems by the old pencil and paper method does not give the average reader any extra insight into the problem. *Maple*™ allows the reader to overcome these obstacles, giving them the power to perform complex computations that would take hours by hand, and can help to visualize many of the geometric interpretations of linear algebra topics in two and three dimensions in a very intuitive fashion. We hope that this book will challenge the reader to become proficient in both theoretical and computational aspects of linear algebra.

Overview of the Text

Chapter 1 of this book is a brief introduction to *Maple* and will help the reader become more comfortable with the program. This chapter focuses on the commands and packages most commonly used when studying linear algebra and its applications. *Maple* commands will always be preceded by an arrow: >, while the output will be displayed in a centered fashion below the *Maple* command. The reader can enter these commands and obtain the same results, assuming that they have entered the commands correctly. Note also that all of the images in this book were produced with *Maple*. The overall intent of this book is to use *Maple* to enhance the concepts of linear algebra, and therefore *Maple* will be integrated into this book in a very casual manner. Where one normally explains how to perform some operation by hand in a standard text, many of the times, we simply use *Maple* commands to perform the same task. Thus, the reader should attempt to become as familiar with the *Maple* syntax as quickly as possible.

At the end of each section, you will find two types of problems: *Home-*

work Problems and *Maple Problems.* The former consists of strictly pen and pencil computation problems, inquiries into theory, and questions about concepts discussed in the section. The idea behind these problems is to ensure that the reader has an understanding of the concepts introduced and can put them to use in problems that can be worked out by hand. For example, *Maple* can multiply matrices together much faster than any person can and without any algebraic mistakes, so why should the reader ever perform these tasks by hand? The answer is simple: In order to fully grasp the mechanics of matrix multiplication, simple problems must be worked out by hand. This manual labor, although usually deemed tedious, is an important tool in learning reinforcement. The *Maple Problems* portion of the homework typically involve problems that would take too long, or would be too computationally complex, to solve by hand. There are many problems in the *Maple Problems* portion that simply ask you to verify your answers to questions from the preceding *Homework Problems*, implying that you can think of *Maple* as a "solutions manual" for a large percentage of this text. You will also notice that several sections are missing the *Homework Problems* section. These sections correspond to special topics that are discussed because they can only be explored in detail with *Maple.*

Website and Supplemental Material

We suggest students and instructors alike visit the book's companion website, which can be found at either of the following addresses:

http://carmine.se.edu/kfrinkle/PrinciplesOfLinearAlgebraWithMaple
http://people.emich.edu/kshiskows/PrinciplesOfLinearAlgebraWithMaple

At the above location, you can download *Maple* worksheets, corresponding to each section's *Maple* commands, along with many other resources. These files can be used with the book so that the reader does not have to retype all of the *Maple* code in order to do problems or practice the material. We highly suggest that all readers unfamiliar with *Maple* (and even those who are) read over the relevant sections of the "Introduction to *Maple*" worksheet before they get too far into the book in order to understand the book's *Maple* code better. Specifically, we suggest looking at plotting/graphing material, differences between sets, lists and strings, and expressions versus functions and how *Maple* uses each. *Homework Problems* solutions and *Maple Problems* worksheets are also available for download.

We should also mention the *Maple* Adoption Program allows an instructor to register a course with *Maple*, so that students can get the student version of *Maple* at a discount. For more information, visit the Maplesoft™ website here:

http://www.maplesoft.com/academic/adoption/

Suggested Course Outlines

It would be nice if we could always cover all of the topics that we wanted to in a given course. This rarely happens, but there are obviously core topics that should be covered. Furthermore, some of the advanced topics require knowledge beyond what students in a basic linear algebra course may have. The appropriate prerequisites for this course would be trigonometry, a precalculus course in algebra, and trig. Also, a computer programming course would be helpful because we are using *Maple*. A year-long course in calculus would also be beneficial in regards to several topics. Here is a list of sections that require advanced knowledge:

- Section 7.2 - Differentiation
- Section 7.3 - Multivariable calculus
- Section 10.1 - Green's theorem
- Section 10.3 - Divergence theorem and double integrals
- Section 11.3 - Gradients and Lagrange multipliers
- Section 12.4 - Linear differential equations

We suggest that as much of the book be covered as possible, but here is the minimum suggested course outline:

Chapter 1	Sections 1 – 2		1 lecture
Chapter 2	Sections 1 – 3		4 lectures
Chapter 3	Sections 1 – 2		3 lectures
Chapter 5	Sections 1 – 6		8 lectures
Chapter 6	Sections 1 – 4		5 lectures
Chapter 8	Sections 1 – 5		7 lectures
Chapter 9	Sections 1 – 4		5 lectures
Chapter 11	Sections 1 – 3		3 lectures
Chapter 12	Sections 1 – 3,5		3 lectures
		Total	39 lectures

Upon inspection of the above outline, you will notice that Chapters 4, 7, and 10 have been completely omitted. Chapter 4 has interesting applications of matrix multiplication to geometry, business, finance, and curve fitting, and we highly suggest covering Sections 4.1 and 4.4. Curve fitting is covered in Section 4.2, but is covered more in-depth in Chapter 11, where pseudoinverses and the method of least-squared deviation are introduced. Chapter 7 contains applications of the information learned about vectors in Chapter 6. If you wish to cover any of the topics in Chapter 10, we highly suggest that you cover Section 7.1. Chapter 10 is a fun chapter on linear maps and how they affect geometric objects. Affine maps are included in this chapter and should be given serious consideration as a topic to cover.

Final Remarks

We hope that both students and instructors will find this book to be a unique read. Our goal was to tell a story, rather than follow the standard textbook formula of: definition, theorem, example, and then repeat for 500 pages. We also hope that you really enjoy using *Maple*, both to explore the geometric and computational aspects of linear algebra, and to verify your pencil and paper work. We very much would like to hear your comments. Some of the questions we would always like answered, both from the student and the instructor, follow:

1. Were there topics that were difficult to grasp from the explanation and examples given? If so, what would you suggest we add/change to help make comprehension easier?
2. How did you enjoy the mixture of homework and *Maple* problems? Did you gain anything from verifying your answers to the homework problems with *Maple*?
3. What were some of your favorite/least favorite sections, and why?
4. Do you feel there were important topics, integral to a first semester course in linear algebra, that were missing from this text?
5. Did embedding *Maple* commands and output within the actual explanation of topics help to illustrate the topics?
6. Overall, what worked the best for you in this text, and what really did not work?

It would be wonderful if this text, in its first edition, was free of errors: both grammatical and mathematical. However, no matter how many times we read and proofread this text, it is a certainty that something will be missed. We hope you contact us with any and all mistakes that you have found, along with comments and suggestions that you may have.

Acknowledgments

First, we would like to express our thanks to Jacqueline Palmieri, Christine Punzo, Stephen Quigley, and Susanne Steitz-Filler of John Wiley & Sons, Inc. for making the entire process, from the original proposal, to project approval, to final submission, incredibly smooth. The four of you were supportive, encouraging, very enthusiastic, and quick to respond to any questions that arose over the course of this project. We appreciate this very much. We would also like to thank the following individuals who were involved in the original peer review process:

Derek Martinez, Central New Mexico University
Dror Varolin, Stony Brook University
Gian Mario Besana, DePaul University

Chris Moretti, Southeastern Oklahoma State University
Andrew Ross, Eastern Michigan University

Special thanks go to Bobbi Page, who took the time to read large portions of early drafts of this book, pored over the copious copyedits, and made many invaluable suggestions. The two successive spring semester Linear Algebra students at Southeastern Oklahoma State University deserve a warm round of applause for being guinea pigs and error hunters. Thanks also goes out to the countless students from the many courses Dr. Shiskowski has taught at Eastern Michigan University, with the help of *Maple*, specifically the Linear Algebra courses. We would also like to thank Mark Bickham, whose idea for a title to this book finally made both authors happy. Thanks again to everyone who was involved in this project, at any point, at any time. If we forgot to add your name this time around, perhaps you will make it into the second edition.

Kenneth Shiskowski
Eastern Michigan University
kenneth.shiskowski@emich.edu

Karl Frinkle
Southeastern Oklahoma State University
kfrinkle@se.edu

With all textbooks, one should attempt to be consistent with notation, not only within the text, but within the field of mathematics upon which it is based. For the most part, we have done this.

Table of Symbols and Notation

$\mathbf{B}$, $\mathbf{K_1}$, $\mathbf{Q}$	Bold capital letters designate sets of objects, usually vectors, or a field
$\mathbb{R}$, $\mathbb{C}$	Real and complex numbers, respectively
$\mathbb{R}^n$, $\mathbb{C}^n$	n-tuples of real and complex numbers, respectively
$\mathbb{R}^{m\times n}$, $\mathbb{C}^{m\times n}$	$m \times n$ matrices with real and complex entries
$\mathbb{S}$, $\mathbb{T}$, $\mathbb{R}$	Math script capital letters denote vector spaces
$\dim(\mathbb{S})$	Dimension of a vector space $\mathbb{S}$
u, x e_k	Lower-case letters are designated as scalars
$\overrightarrow{u}$, $\overrightarrow{v}$, $\overrightarrow{e_k}$	Lower-case letters with arrows over them are vectors, or column matrices
$\langle 1, 2, -1\rangle$, $\langle x_1, x_2\rangle$	Vectors expressed in component form
$\overrightarrow{u} \cdot \overrightarrow{y}$	Dot product of two vectors
$\overrightarrow{u} \times \overrightarrow{y}$	Cross product of two vectors
$\text{proj}_{\overrightarrow{v}}(\overrightarrow{w})$	Projection of $\overrightarrow{w}$ onto $\overrightarrow{v}$
$\text{comp}_{\overrightarrow{v}}(\overrightarrow{w})$	Component of $\overrightarrow{w}$ onto $\overrightarrow{v}$
A, C, X	Single capital letters represent matrices
AB, AX	Matrix multiplication has no symbol, two matrices in sequence implies multiplication
A^T, A^{-1}	Transpose, inverse of a matrix
$(A\|B)$	Augmented matrix, with A on the left, B on the right
$A_{2,3}$, $B_{j,k}$	Entries of a matrix are indexed by row,column
$\det(A)$, $\text{adj}(A)$	Determinant, adjoint of a matrix
$p(A)$	Pseudoinverse of a matrix
$T : \mathbb{R} \to \mathbb{S}$	Linear map from vector space $\mathbb{R}$ to vector space $\mathbb{S}$
$\text{Ker}(T)$, $\text{Im}(T)$	The kernel, image of a linear map T
$\int_a^b f(u)\, du$	Integral of $f(u)$ with respect to u on $[a, b]$
$\prod_{j=1}^{n} x_j$	The product $x_1 x_2 \cdots x_n$
$\sum_{j=1}^{n} x_j$	The sum $x_1 + x_2 + \cdots + x_n$
$\nabla D(\overrightarrow{d})$	Gradient of vector valued function D
$\frac{dF}{dx}$, $\frac{\partial F}{\partial x}$	Derivative, partial derivative, of F with respect to x

Contents

Chapter 1

An Introduction To Maple

This chapter presents a very brief introduction to the *Maple* computer algebra system that will be used throughout this book for the production of all of our computations and graphics. *Maple* is an amazingly powerful piece of software that is specifically designed to do all things mathematical, this also makes it extremely useful to the general users of mathematics in all the sciences. By using *Maple*, we can increase our mathematical productivity at least 10-fold since *Maple* is much faster and more accurate than we mere humans will probably ever be. You will have to be patient in the beginning when you are first learning *Maple*, since like all software (*Excel*, *Word*, etc.), it will take some time to get used to the syntax and idiosyncracies of *Maple*.

As soon as you have read this short chapter, you should immediately download the *Maple* file (worksheet) called "Introduction to *Maple*" from this books companion website, whose address can be found in the preface. Please download this *Maple* file to your home computer so that you can play with it in order to master some of the basics of *Maple* programming. Of course, you will also need to have a copy of *Maple* on your computer in order for you to actually use the worksheet. If you are using this text for a course in linear algebra, then hopefully your instructor has registered the course with the *Maple* adoption program so that you can get a discounted copy of the latest student version of *Maple* from the *Maple* website at www.maplesoft.com. The "Introduction to *Maple*" worksheet gives a much more thorough overview of *Maple* and its inner mysteries than could be given in this chapter, but this file also goes into some areas of mathematics other than those directly relevant to linear algebra, as such, you can ignore the sections on statistics, financial mathematics, and calculus, unless you have some interest and prior knowledge in these areas.

One last word about *Maple* that is also generically true about all software, the Help leaves much to be desired, although *Maple*'s Help has gotten significantly better with each new version, which is at *Maple* 14 with the writing of this book. By this, we do not mean that *Maple*'s Help is useless, merely that it does not often provide the needed examples of how to use *Maple*'s commands in more than the most basic of ways. Hopefully, our "Introduction to *Maple*" file will provide the needed assistance that *Maple*'s Help can not give you in using this text. You should also feel free to check out the Internet for other sources of information on how to use *Maple* beyond this introductory file. And a good *Maple*ing to you all!

If you are using an older version of *Maple* (e.g., *Maple* 9.5), then we suggest using the classic worksheet version. However, if you have a newer version of *Maple* (e.g., *Maple 14*), once the program has been opened, go to *File* → *New* → *Worksheet Mode*. The other type of document style under the *New* setting is *Document Mode*, and will still perform all the same commands, but it is geared more toward presentation than utility. The main reason that we focus on the worksheet versus the document mode is that we are attempting to make as much of the *Maple* code found in this book as backward compatible as possible. There have been many new and wonderful enhancements and additions to *Maple* since both of the authors started using it (long, long ago), so feel free to explore!

1.1 The Commands

Before we start, be sure you have opened up a new worksheet document. Once this is done, you will notice a bunch of commands on the left, as well as various options above the worksheet window in the toolbar. Feel free to explore these, paying particular attention to the expressions icon on the left. It has several shortcuts for popular commands you may want to use on a regular basis that allow you to simply fill in variables, limits, and other entries into standard mathematical operators. The newer versions of *Maple*, including *Maple 13*, have two different input types on the command line. For the purpose of this book, we will assume that you are using the "Text" version of the command line. The reason is that if you select the "Math" option (which is the standard) your input will be displayed in nice formula fashion, which looks good, but for the purposes of entering commands can be potentially burdensome. In "Text" mode, you can type commands character for character as they appear in this book.

No matter which version of *Maple* you use, one of the most important commands to remember is the *restart* command, which clears the values of all the currently used variables. We highly suggest that the *restart* command be placed at the very beginning of each problem you do, especially if there is more than one problem per worksheet file. The next most important thing to learn is how to load packages. Not every command is immediately accessible upon startup. This feature reduces the total amount of memory and CPU cycles that the program uses. Most of the basic commands that a user would encounter do not require a package to be loaded. However, there are many linear algebra concepts (e.g., computing the determinant of a matrix), which require the linear algebra package *linalg* to be loaded. To load the linear algebra package, the following command is used.

> with(linalg):

If you wish to know what commands are loaded with the *linalg* package, simply use a semicolon instead. Ending a command with a colon suppresses the output, while ending a command with a semicolon displays the output. In *Maple 14*, you do not even need to end the command with a semicolon; simply press "Enter" after typing in your command.

> with(linalg);

[*BlockDiagonal, GramSchmidt, JordanBlock, LUdecomp, QRdecomp, Wronskian, addcol, addrow, adj, adjoint, angle, augment, backsub, band, basis, bezout, blockmatrix, charmat, charpoly, cholesky, col, coldim, colspace, colspan, companion, concat, cond, copyinto, crossprod, curl, definite, delcols, delrows, det, diag, diverge, dotprod, eigenvals, eigenvalues, eigenvectors, eigenvects, entermatrix, equal, exponential, extend, ffgausselim, fibonacci, forwardsub, frobenius, gausselim, gaussjord, geneqns, genmatrix, grad, hadamard, hermite, hessian, hilbert, htranspose, ihermite, indexfunc, innerprod, intbasis, inverse, ismith, issimilar, iszero, jacobian, jordan, kernel, laplacian, leastsqrs, linsolve, matadd, matrix, minor, minpoly, mulcol, mulrow, multiply, norm, normalize, nullspace, orthog, permanent, pivot, potential, randmatrix, randvector, rank, ratform, row, rowdim, rowspace, rowspan, rref, scalarmul, singularvals, smith, stackmatrix, submatrix, subvector, sumbasis, swapcol, swaprow, sylvester, toeplitz, trace, transpose, vandermonde, vecpotent, vectdim, vector, wronskian*]

As you can see from the above output, the list of commands loaded with the *linalg* package is quite extensive and we will not cover all of these specialized commands in this text. Throughout the text, you will notice that a similar package, *LinearAlgebra*, is also used. This one is used most often when dealing with vectors and linear algebra plotting constructs. *Maple* considers the *LinearAlgebra* package to have superseded the earlier *linalg* package, although *linalg* is still available for use, and we primarily use the *linalg* package throughout this text. Also, the datatype of a matrix is different between the two packages, and so not always interchangeable. The reader should perform the command: with(LinearAlgebra); in order to see the difference in command names for essentially the same operations as well as the new commands not found in the *linalg* package. Next, we will move on to some simple computational commands. As a first example, we can add two numbers together.

> 5+3;

$$8$$

If we want to multiply the result of the previous example by 12, *Maple* has a shortcut that comes in handy if the previous result is quite complicated. The percent sign, %, is used to represent the previous computed output. It is very important to remember that this command refers to the previously computed output, not necessarily the output from the line previous, as *Maple* will allow you to execute command lines in any order desired.

> %*12;

$$96$$

The program also contains all of the trigonometric functions, as well as many other special functions. You can also enter several commands on the same line when separated by semicolons, but pay special attention to the syntax required in the newer version of *Maple*. You will need to separate the commands by semicolons in any version, otherwise *Maple* will read it as multiplication in the newer versions and/or an error in older versions.

> sin(Pi/4); ln(3); exp(2); 2/3;

$$\frac{1}{2}\sqrt{2}$$

$$\ln(3)$$

$$e^2$$

$$\frac{2}{3}$$

The standard operation symbols for addition, subtraction, multiplication, and division are given by +, -, *, and /, respectively. *Maple* does exact computations unless otherwise told to do approximations involving integers and fractions. You can also place a decimal point after an integer to make it approximate instead of exact. If we wish to express the expressions above in floating point form, we can use the *evalf* command.

> evalf(sin(Pi/4)); evalf(ln(3)); evalf(exp(2)); 2./3;

$$0.7071067810$$

$$1.098612289$$

$$7.389056099$$

$$0.6666666667$$