APPLIED MATRIX MODELS

A Second Course in Linear Algebra with Computer Applications

ANDY R. MAGID

APPLIED MATRIX MODELS

A Second Course in Linear Algebra with Computer Applications

ANDY R. MAGID

A WILEY-INTERSCIENCE PUBLICATION
JOHN WILEY & SONS
New York • Chichester • Brisbane • Toronto • Singapore

Library of Congress Cataloging in Publication Data

Magid, Andy R.
Applied matrix models.

"A Wiley-Interscience publication."
Includes index.
1. Matrices. 2. Matrices—Data processing. I. Title.
QA188.M34 1985 512.9'434 84-19668
ISBN 0-471-88865-6

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PREFACE

This book had its origins in a course I introduced at the University of Oklahoma in the spring semester of 1982. This was to be a second course in linear algebra with computer applications. I wanted to teach students who, in their first course in linear algebra, had learned the basics of vector and matrix algebra and how to set up and solve the primary problems of linear algebra—solving systems of linear equations and obtaining eigenvalues and eigenvectors of matrices—in the way such problems arise in actual applications. That is, I intended to teach them how to recognize the linear problem in the context of the application, how to use available computer packages to solve the linear problem, and also how to have a feeling for the linear algebra the package performs. I also had some definite ideas about what I did *not* want to teach them: I was not trying to create either a course in numerical linear algebra or a course in the programming of linear algebra algorithms—both courses are already available in the standard curriculum, and neither is really suitable for the audience I intended to reach.

Because there was no single text available which covered the contents of the course as I envisioned it, I began to assemble and integrate a number of ideas from various sources to create my lecture notes. (Foremost among these sources were *Linear Algebra and Its Applications*, by G. Strang, *Applications of Linear Algebra*, by C. Rorres and H. Anton, and *LINPACK User's Guide*, by J. Dongarra et al., and the debt the present volume owes to those works is considerable.) As with all such ventures, there was a mixture of success and failure, but with the help (and patience!) of an enthusiastic group of students the course ultimately had more of the former than the latter. So I gave some thought to turning my lecture notes into a book. A conversation with David Kaplan, ultimately my editor, and some helpful encouragement from him then helped convert these thoughts into a commitment, and this book is the result.

There are a couple of features of this book about which the reader should be warned. One such feature concerns the approach taken to using a computer to solve linear algebra problems. This is done here by using library procedures to do the linear algebra (these need to be put into simple FORTRAN calling programs to handle input and output, of course). Programming purists will object, and rightly so, that this teaches the reader nothing about converting linear algebra methods into computer code. Such coding has all sorts of pedagogic and therapeutic value for teaching programming, but, in my view, has as little place in a course in applications of linear algebra as, say, requiring students to write their own procedure to compute cosine would have in a course in calculus. Of course, calculus students need to learn about Taylor-series expansions, and they should understand that such ideas are in principle behind the cosine routines in the computers they are using, but there is no need to postpone their using the FORTRAN function COS(X) until they have studied enough numerical analysis to understand why the particular rational function approximation (which is not the Taylor series) was chosen to code this function. Similarly, students who can understand the *Q-R* factorization of a matrix, say, in terms of the Gram–Schmidt orthogonalization process, should be entitled to use the LINPACK routine that accomplishes this factorization and to solve least-squares problems with it, even if they are in no position to code such a routine themselves. Briefly put, this book is aimed at the user of linear algebra who wishes to intelligently employ available library routines to solve his or her problems, and is willing to employ mathematical technology (in the form of computer routines) that he or she may not be able to recreate.

Another feature that the reader needs to be warned about is the author's willingness to freely use determinants in theoretical arguments. This is not currently very fashionable; much of the modern trend in the theory of linear equations has led to the elimination of determinants from that theory. The reason, of course, is that a determinant is a hard thing to compute—at least by serial processing on a large general matrix—and so most modern linear algebra texts try to avoid their use as much as possible. Since computations of this type are not at issue here, determinants have been used in a number of places, in their essential algebraic sense as the basic invariants of matrices under elementary operations. Indeed, as the theory of matrix invariants in higher algebra shows, any invariants of matrices under such operations must be polynomials in determinants. So it is not surprising that determinants arise naturally in the theory of systems of linear equations, since the solutions of such systems are obtained by operations on matrices, and when they do arise, they appear in our discussions.

Finally, I am happy to be able to acknowledge the help of Rhonda Peterson, who typed the manuscript, the suggestions of the University of Oklahoma students, who have taken the course on which this book is based, and the encouragement of my family.

ANDY R. MAGID

Norman, Oklahoma
October 1984

CONTENTS

1

INTRODUCTION

1.1 THE BOOK AND HOW TO USE IT

This book is intended for readers who have learned some basic linear algebra, who have a little computing experience, and who are now ready to learn about the applications of linear algebra, perhaps with a view to using such applications in their own work. Even with small-scale applications, such as those considered in this book, the arithmetic involved with linear algebra becomes lengthy, and so the use of the computer to perform computations is stressed throughout the book. Not every, in fact not even most, users of linear algebra will have the inclination or training to do their own computer coding of linear algebra computations. Fortunately, excellent program libraries of linear algebra routines are available, and the computational sections of this book explain how to use them. There are various levels of understanding that a user of a library routine may achieve; the explanations in this book are designed to help the reader understand the theoretical ideas a routine is intended to implement. It is these three themes —applications, computations, and the background necessary to understand the computations—that form the organizing principles of this book.

Readers of this book are expected to know basic linear algebra. That is, they should know the algebra of matrices and vectors, how to solve systems of linear equations, and know about eigenvectors and eigenvalues. In fact, the actual number of prerequisites is rather small: readers who feel comfortable with the review section 1.2 on matrix and vector algebra are adequately prepared for the rest of the book; the remainder of the chapters are self-contained mathematically, although some additional theory is developed in the exercises. Readers are also expected to have some computing experience. Here again, the assumed number of prerequisites is fairly small: it is more important to have had the experience of entering programs and data and seeing the results of computation than to have had any particular programming or operating training. The language used for the programs in this book is FORTRAN. Readers without previous experience in FORTRAN will find in Section 1.4 an explanation of the language sufficient to deal with programs of subsequent chapters.

The FORTRAN language is used for two reasons: (1) because of the ease in dealing with matrices as vectors in that language; and (2) because the linear algebra program libraries used here are in that language. The most important of these libraries is LINPACK, written at Argonne National Laboratories. All of the methods in Chapters 2, 3, and 4 of this book use the LINPACK routines exclusively, and they are used in Chapter 5 as well. There is an excellent manual documenting LINPACK, complete with program listings—the *LINPACK User's Guide*† —which many readers will want to consult. The aspects of LINPACK that are needed in this book, however, are explained as they are required.

LINPACK does not contain eigenvalue/eigenvector routines. When such routines are needed, we use the procedure EISPAC,‡ and the procedure EIGRF (the latter is in the IMSL§ program library). To be able to use the programs in this book, then, the reader must have access to a computing facility on which the LINPACK and IMSL libraries, and the EISPAC procedure, have been installed.

The book is organized so that the methods discussed are introduced in the order of their logical complexity. Chapter 1 is a review chapter; Chapter 2 addresses the problem of systems of equations with a unique solution. Chapter 3 takes up the problem of systems with possibly infinitely many solutions, and Chapter 4 the problem of systems with possibly no solutions.

†J. Dongarra et al., *LINPACK User's Guide*, SIAM, Philadelphia, 1979.

‡A control program for EISPACK. See B. Smith et al., *Matrix Eigensystems Routines—EISPACK Guide* (2nd ed.), Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, 1976.

§Published by IMSL, Inc. Houston. Library Reference Manual provided to subscribers.

As a group, these last three chapters constitute a study of the linear algebra used to deal with static and steady-state applications. Chapters 5 and 6 deal with eigenvector and eigenvalue problems associated with discrete and continuous time (respectively) models. These two chapters are grouped together as the study of dynamic models. A more detailed summary now follows.

The problem dealt with in Chapter 2 is a system of n linear equations in n unknowns whose unique solution is sought. In the theoretical sections of the chapter, the solution process is explained by means of transforming the system (Gaussian elimination) and by writing the system in matrix form and factoring the matrix of coefficients (L-U decomposition). Among the applications lending to such systems, the chapter considers electrical networks, production economic models, and static distributions. The computational sections cover solution using LINPACK to perform L-U decomposition, and also solutions by iteration.

In Chapter 3 systems of m equations in n unknowns are examined. In the theoretical sections the nature of the set of solutions of such a system is covered, along with the concept of rank and row reduction. The applications considered are network flow, resource allocation, and economic exchange models. There is no row-reduction routine in the program libraries, so in the computational section one is developed, using the basic linear algebra subroutines attached to LINPACK.

Chapter 4 also deals with systems of m equations in n unknowns, but now the emphasis is on possibly inconsistent systems. With such a system, the object is to find the best approximate solution. The notion of "best" requires a discussion of vector geometry in the theoretical sections, which then leads into the various solution methods: the normal equations for least squares, the Gram–Schmidt process, and its matrix form (Q-R decomposition). The applications sections deal with fitting equations to data in various contexts. The computational sections explain how to implement the solution methods, either via the normal equations or using the Q-R decomposition.

Chapter 5 begins the study of problems for which the desired solution requires computation of eigenvectors and eigenvalues of a matrix. The theoretical sections present the theory of eigenvectors and eigenvalues, and a slight discussion of the algorithms used in their computation, as well as a more extensive discussion of their interpretation. The applications deal with models for which powers of a matrix, or powers of a matrix times a vector, need to be computed, including population growth and Markov chains. The computational sections cover using the EISPAC and IMSL eigenvalue/eigenvector procedures.

Chapter 6 applies eigenvalue/eigenvector methods to the solution of systems of differential equations. The theoretical sections discuss how such solutions are found using eigenvalues and eigenvectors to obtain the solution functions, while the applications sections focus on sources of such problems in such areas as chemical reactions and mechanical systems. (The computations necessary to solve these problems are already treated in Chapter 5.)

1.2 REVIEW OF VECTOR AND MATRIX ALGEBRA

n-vectors are column n-tuples of numbers:†

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \qquad w = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$$

The entries $v_1, \ldots, v_n$ of the vector v are its *components*. Vectors are added by adding corresponding components and we can multiply a vector by a *scalar*† (number) by multiplying each component by the scalar:

$$v + w = \begin{bmatrix} v_1 + w_1 \\ \vdots \\ v_n + w_n \end{bmatrix} \qquad av = \begin{bmatrix} av_1 \\ \vdots \\ av_n \end{bmatrix}$$

The 0 n-vector has all its components zero, and $-v$ is the vector $(-1)v$. A *linear combination* of the n-vectors $v_1, \ldots, v_m$ is a vector w of the form

$$w = a_1 v_1 + \cdots + a_m v_m$$

If all the a_i are zero, then w is zero; this is called the trivial linear combination. If $v_1, \ldots, v_m$ are *linearly dependent* if some nontrivial linear combination of them is 0; otherwise, they are *linearly independent*.

The collection of all linear combinations of a set of vectors is called their *span*. Since a sum of linear combinations, or a scalar multiple of linear combinations, is again a linear combination, the span of a set of vectors is closed under vector addition and scalar multiplication.

A set of n-vectors closed under vector addition and scalar multiplication is called a *vector subspace*. Every vector subspace is the span of a set of vectors; a linearly independent such spanning set is called a *basis* for the subspace. Every vector subspace, including the space of all n-vectors, has a basis. The number of vectors in all bases of a given subspace is the same

† These may be real or complex. The latter are required only in Chapters 5 and 6. Unless otherwise specified, we usually mean just real numbers.

and is called the *dimension* of the subspace. For the space of all n-vectors, this dimension is n. In any subspace, a minimal spanning set or maximal linearly independent set is a basis. If the subspace has dimension m, then a spanning set, or linearly independent set, of m elements is a basis.

An m by n (or $m \times n$) *matrix* is a rectangular array of mn numbers into m rows of n elements or n columns of m elements:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

The entry in row i and column j, a_{ij}, is called the (i, j) entry. A column n-vector is then just an $n \times 1$ matrix.

Pairs of m by n matrices can be added by adding corresponding entries, and a matrix can be multiplied by a scalar by multiplying all the entries by that scalar:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

$$c\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} ca_{11} & \cdots & ca_{1n} \\ \vdots & & \vdots \\ ca_{m1} & \cdots & ca_{mn} \end{bmatrix}$$

The *transpose* of the $m \times n$ matrix A is the $n \times m$ matrix A^T whose (i, j) entry is the (j, i) entry of A:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \qquad A^T = \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & & \vdots \\ a_{1n} & \cdots & a_{mn} \end{bmatrix}$$

An $m \times n$ matrix A and an $n \times p$ matrix B can be multiplied to obtain an

$m \times p$ matrix:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & \cdots & b_{1p} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{np} \end{bmatrix}$$

$$AB = \begin{bmatrix} c_{11} & \cdots & c_{1p} \\ \vdots & & \vdots \\ c_{n1} & \cdots & c_{np} \end{bmatrix}$$

where

$$c_{ij} = a_{i1}b_{ij} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

In particular, we have defined Ax where A is an $m \times n$ matrix and x an n-vector ($= n \times 1$ matrix). When the relevant products and sums can be formed matrix multiplication is associative, $A(BC) = (AB)C$, distributive over addition, $A(B + C) = (AB) + (AC)$, but not always commutative, $AB \neq BA$ for some A, B. With respect to the transpose, matrix multiplication gets reversed:

$$(AB)^T = B^T A^T$$

There are several alternative formulations of matrix multiplication. To explain these simply, we introduce some notation:

(1.2.1) Rows and Columns of A Matrix. Let A be an $m \times n$ matrix. Then A_i, $i = 1, \ldots, n$, denotes the ith column of A (an $m \times 1$ matrix) and A^i, $i = 1, \ldots, m$, denotes the ith row of A (a $1 \times n$ matrix). Symbolically, we have

$$A = \begin{bmatrix} A^1 \\ \vdots \\ A^m \end{bmatrix} = [A_1 \cdots A_n]$$

(1.2.2) Matrix Multiplication Formulas. Let A be an $m \times n$ matrix and B an $n \times p$ matrix, and let a_{ij} denote the (i, j) entry of A and b_{ij} the (i, j) entry of B. Then in terms of (1.2.1):

1. The jth column $(AB)_j$ of the product AB is given by the linea combination

$$(AB)_j = A(B_j) = b_{1j}A_1 + \cdots + b_{nj}A_n \qquad \text{for } 1 \leq j \leq p$$

2. The ith row $(AB)^i$ of the product AB is given by the matrix combination

$$(AB)^i = A^iB = a_{i1}B^1 + \cdots + a_{in}B^n \qquad \text{for } 1 \le i \le m$$

3. The (i, j) entry $(AB)_{ij}$ of the product AB is given by the matrix product

$$(AB)_{ij} = A^iB_j \qquad \text{for } 1 \le i \le m \quad \text{and} \quad 1 \le j \le p$$

Matrix products of square $(n \times n)$ matrices are always formable. The *identity* $n \times n$ matrix I_n is the matrix†

$$I_n = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

[The (i, j) entry of I_n is zero if $i \neq j$ and 1 if $i = j$.] If A is any other $(n \times n)$ matrix then $AI_n = I_nA = A$. A $(n \times n)$ matrix A is *invertible* if some $n \times n$ matrix multiplies it to the identity. If A is invertible the unique matrix multiplying it to the identity is denoted A^{-1} and we have $AA^{-1} = A^{-1}A = I_n$. A^{-1} is called the *inverse* of A.

Powers of the $n \times n$ matrix A are defined by

$$A^0 = I^n, \qquad A^1 = A, \qquad A^2 = AA, \ldots, A^{k+1} = AA^k$$

If A is invertible, the negative powers of A are defined by $A^{-k} = (A^{-1})^k$. With these definitions, the usual laws of exponents apply:

$$A^pA^q = A^{p+q} \qquad (A^p)^q = A^{pq}$$

(1.2.3) Determinants. There is a number, called the determinant, associated to any $n \times n$ matrix A, and denoted $\det(A)$. Among its properties are:

1. If the matrix A' is obtained from A by adding a multiple of a row to another row, or a multiple of a column to another column, then $\det(A') = \det(A)$.
2. If A' results from A by interchanging two rows, or by interchanging two columns, then $\det(A') = -\det(A)$.

† Zero entries of a matrix are usually denoted by blanks in this book.

3. $\det(aA) = a^n \det(A)$.
4. $\det(AB) = \det(A)\det(B)$.
5. $\det(I_n) = 1$.
6. If $(A)_{ij}$ denotes the $(n-1) \times (n-1)$ matrix obtained from A by discarding row i and column j, and a_{ij} denotes the (i, j) entry of A . then for any $k = 1, 2, \ldots, n$ we have
   (a) $\det(A) = \Sigma_i(-1)^{i+k} a_{ik} \det((A)_{ik})$ (cofactor expansion down column k);

   and

   (b) $\det(A) = \Sigma_j(-1)^{k+j} a_{kj} \det((A)_{kj})$ (cofactor expansion along row k).
7. If A is a 2×2 matrix then $\det(A)$ is calculated as follows:

$$A = \begin{bmatrix} a & b \\ c & c \end{bmatrix} \qquad \det(A) = ad - bc$$

8. If A has zeros below the diagonal (a lower triangular matrix), above the diagonal (an upper triangular matrix), or both (a diagonal matrix) then $\det(A)$ is the product of the diagonal entries of A.
9. $\det(A^T) = \det(A)$.

Determinants determine if matrices are invertible. We record this along with other criteria for invertibility:

(1.2.4) Invertibility Criteria. The $n \times n$ matrix A is invertible if any of the following hold, and conversely:

1. $\det(A) \neq 0$.
2. $Ax = 0$ implies $x = 0$ for any n-vector x.
3. The columns of A are linearly independent.
4. The columns of A span the space of n-vectors.

These criteria can be expressed in terms of the equation

$$Ax = b$$

for b a given n-vector, to be solved for x: criterion number 2 can be interpreted as saying there is a unique solution, and number 4 as saying that for any b there is at least one solution. If A is invertible, the unique solution for any given b is

$$x = A^{-1}b.$$

EXERCISES 1.2

1. Let $v = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$, $w = \begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix}$, $u = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
   (a) Show that v and w are linearly independent.
   (b) Show that u is a linear combination of v and w if and only if $2a + b - c = 0$.
   (c) Let V be the set of all 3-vectors x such that $[2 \quad 1 \quad -1]x = [0]$. Show that V is a vector subspace.
   (d) Show that v, w form a basis of V.
2. Let $v_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$, $v_2 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$, $v_3 = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$, $v_4 = \begin{bmatrix} 5 \\ 8 \\ 1 \end{bmatrix}$
   (a) Show that v_3 and v_4 are linear combinations of v_1 and v_2.
   (b) Show that v_1 and v_2 are linearly independent.
   (c) Show that v_1, v_2 form a basis for the vector subspace V spanned by v_1, v_2, v_3, v_4.
   (d) Find a 1×3 matrix A such that V is the set of all 3-vectors x with $Ax = [0]$.
3. Let A be an $n \times n$ matrix with (i, j) entry a_{ij}. Let A_{ij} be the matrix obtained from A by deleting row i and column j. Let B be the $n \times n$ matrix whose (i, j) entry is $(-1)^{i+j}\det(A_{ji})$. Prove that $AB = \det(A)I_n$ [like property 6 of (1.2.3)]. The matrix B is called the *classical adjoint* of A.
4. Assume A is an invertible $n \times n$ matrix and that $Ax = b$ for n-vectors x and b, with ith entries x_i and b_i, respectively. Let B be the classical adjoint of A from problem 3.
   (a) Show that $A^{-1} = \det(A)^{-1}B$.
   (b) From $x = A^{-1}b$ show that

   $$x_i = (\det A)^{-1}(b_{i1}b_1 + \cdots + b_{in}b_n)$$

   where b_{ij} are as in problem 3.
   (c) Let C_i, $i = 1, \ldots, n$, be the matrix obtained from A by replacing column i by b. Using this and property 6 of (1.2.3) show that

   $$x_i = (\det A)^{-1}\det(C)$$

   (This formula is known as Cramer's rule).