Matrix Computation for Engineers and Scientists

Matrix Computation for Engineers and Scientists

Alan Jennings
Queen's University, Belfast

A Wiley – Interscience Publication

JOHN WILEY & SONS
London · New York · Sydney · Toronto

Library of Congress Cataloging in Publication Data:

Jennings, Alan.
Matrix computation for engineers and scientists.

'A Wiley–Interscience publication.'
1. Matrices. I. Title.
TA347.D4J46 519.4 76–21079

ISBN 0 471 99421 9

Typeset in IBM Journal by Preface Ltd, Salisbury, Wilts
and printed in Great Britain by Unwin Brothers Ltd.,
The Gresham Press, Old Woking, Surrey.

Preface

In the past the sheer labour of numerical processes restricted their development and usefulness. Digital computers have removed this restriction, but at the same time have provided a challenge to those who wish to harness their power. Much work has been put into the development of suitable numerical methods and the computer organizational problems associated with their implementation. Also, new fields of application for numerical techniques have been established.

From the first days of computing the significance of matrix methods has been realized and exploited. The reason for their importance is that they provide a concise and simple method of describing lengthy and otherwise complicated computations. Standard routines for matrix operations are available on virtually all computers, and, where these methods are employed, duplication of programming effort is minimized. Matrices now appear on the school mathematics syllabus and there is a more widespread knowledge of matrix algebra. However, a rudimentary knowledge of matrix algebra should not be considered a sufficient background for embarking upon the construction of computer programs involving matrix techniques, particularly where large matrices are involved. Programs so developed could be unnecessarily complicated, highly inefficient or incapable of producing accurate solutions. It is even possible to obtain more than one of these deficiencies in the same program. The development of computer methods (most certainly those involving matrices) is an art which requires a working knowledge of the possible mathematical formulations of the particular problem and also a working knowledge of the effective numerical procedures and the ways in which they may be implemented on a computer. It is unwise to develop a very efficient program if it is so complicated that it requires excessive programming effort (and hence program testing time) or has such a small range of application that it is hardly ever used. The right balance of simplicity, economy and versatility should be sought which most benefits the circumstances.

Chapter 1 is intended to act as a review of relevant matrix algebra and hand computational techniques. Also included in this chapter is a discussion of the matrix properties which are most useful in numerical computation. In Chapter 2 some selected applications are briefly introduced. These are included so that the reader can see whether his particular problems are related to any of the problems mentioned. They also illustrate certain features which regularly occur in the formulation of matrix computational techniques. For instance:

(a) Alternative methods may be available for the solution of any one problem (as with the electrical resistance network, sections 2.1 and 2.2).
(b) Matrices often have special properties which can be utilized, such as symmetry and sparsity.
(c) It may be necessary to repeat the solution of a set of linear equations with modified right-hand sides and sometimes with modified coefficients (as with the non-linear cable problem, section 2.12).

Chapter 3 describes those aspects of computer programming technique which are most relevant to matrix computation, the storage allocation being particularly important for sparse matrices. Multiplication is the main matrix operation discussed in this chapter. Here it is interesting to note that some forethought is needed to program even the multiplication of two matrices if they are large and/or sparse. Numerical techniques for solving linear equations are presented in Chapters 4, 5 and 6. The importance of sparse matrices in many applications has been taken into account, including the considerable effect on the choice of procedure and the computer implementation.

Chapter 7 briefly introduces some eigenvalue problems and Chapters 8, 9 and 10 describe numerical methods for eigensolution. Although these last four chapters may be considered to be separate from the first six, there is some advantage to be gained from including procedures for solving linear equations and obtaining eigenvalues in the same book. For one reason, most of the eigensolution procedures make use of the techniques for solving linear equations. For another reason, it is necessary to be familiar with eigenvalue properties in order to obtain a reasonably comprehensive understanding of methods of solving linear equations.

Three short appendices have been included to help the reader at various stages during the preparation of application programs. They take the form of questionnaire checklists on the topics of program layout, preparation and verification.

Corresponding ALGOL and FORTRAN versions of small program segments have been included in Chapters 2, 3, 4 and 5. These segments are not intended for direct computer use, but rather as illustrations of programming technique. They have been written in such a way that the ALGOL and FORTRAN versions have similar identifiers and structure. In general this means that the ALGOL versions, while being logically correct, are not as elegant as they might be. To obtain a full appreciation of the complete text it is therefore necessary to have some acquaintance with computer programming. From the mathematical standpoint the text is meant to be as self-sufficient as possible.

I hope that the particular methods given prominence in the text, and the discussion of them, are not only justified but also stand the test of time. I will be grateful for any comments on the topics covered.

ALAN JENNINGS
Department of Civil Engineering,
Queen's University, Belfast

Acknowledgements

It was while on sabbatical leave at the Royal Military College, Kingston, Ontario, Canada, that I contemplated writing this book and started an initial draft. While I was at Kingston, and more recently at Belfast, I have had much encouragement and helpful advice from many colleagues (and tolerance from others). My particular thanks are to Professor A. A. Wells, head of the department of civil engineering at Queen's University, and Professor J. S. Ellis, head of the department of civil engineering at the R.M.C., Kingston, for helping me to provide time to prepare the script. Special thanks are due to Dr. M. Clint of the department of computer science at Queen's University who has carefully read the script and given many excellent comments. I would also like to thank D. Meegan of the department of engineering mathematics at Queen's University for carrying out checks on the program segments, Miss R. Tubman for competently typing the script and Mrs V. Kernohan for carefully preparing the figures.

Contents

It is a capital mistake to theorise before one has data

Sir Arthur Conan Doyle

Chapter 1
Basic Algebraic and Numerical Concepts

1.1 WHAT IS A MATRIX?

A matrix can be described simply as a rectangular array of elements. Thus

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 5 & 1 & 0 \\ 1 & 3 & 1 & 3 & 1 \\ 3 & 2 & 4 & 6 & 0 \end{bmatrix} \qquad (1.1)$$

is a matrix of order 3×5 as it has three rows and five columns. The elements of a matrix may take many forms. In matrix (1.1) they are all real non-negative integers; however, they could be real or complex numbers, or algebraic expressions, or, with the restrictions mentioned in section 1.15, matrices themselves or matrix expressions. The physical context of the various elements need not be the same; if one of the elements is a measure of distance, it does not follow that the other elements have also to be measures of distance. Hence matrices may come from a large variety of sources and take a variety of forms. Computation with matrices will involve matrices which have elements in numerical form. However, matrices with elements of algebraic form will be of significance in the theoretical discussion of properties and procedures.

Matrix (1.1) could represent the numbers of different coins held by three boys, the columns specifying the five coin denominations (i.e. 1 p, 2 p, 5 p, 10 p and 50 p) while the rows differentiate the boys. The interpretation of matrix A would therefore be according to Table 1.1. Whereas any table of information could be considered as a matrix by enclosing the data within square brackets, such consideration would be fruitless unless it can operate with some other matrices in

Table 1.1 Possible interpretation of matrix (1.1)

	Coins				
	1 p	2 p	5 p	10 p	50 p
Tom	2	0	5	1	0
Dick	1	3	1	3	1
Harry	3	2	4	6	0

such a way that the rules of matrix algebra are meaningful. Before describing the basic rules of matrix algebra it is necessary to be able to specify any element of a matrix algebraically. The usual method for this is to replace whatever labels the rows and columns have by numbers, say 1 to m for rows and 1 to n for columns, and then to refer to the element on row i and column j of matrix **A** as a_{ij}.

A matrix is square if $m = n$ and is rectangular if $m \neq n$.

1.2 THE MATRIX EQUATION

Probably the most fundamental aspect of matrix algebra is that matrices are equal only if they are identical, i.e. they are of the same order and have corresponding elements the same. The identity (1.1) is a valid matrix equation which implies $m \times n$ ordinary equations defining each element a_{ij}, e.g. $a_{34} = 6$. This property of being able to represent a multiplicity of ordinary equations by a single matrix equation is the main power of matrix methods. (This is in distinct contrast to determinants where the equation

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = \begin{vmatrix} 5 & 1 \\ 1 & 1 \end{vmatrix} \qquad (1.2)$$

does not define the elements a_{11}, a_{12}, a_{21} and a_{22} but only specifies a relationship between them.

If they are of the same order, two matrices may be added by adding corresponding elements. If Table 1.1 describes the state of Tom, Dick and Harry's finances at the beginning of the day and if their transactions during the day are represented by Table 1.2 which specifies a further matrix **H**, then the state of their finances at the end of the day is given by the matrix

$$\left.\begin{aligned} &\mathbf{G} = \mathbf{A} + \mathbf{H} \\ &\text{i.e.} \\ &\mathbf{G} = \begin{bmatrix} 2 & 0 & 5 & 1 & 0 \\ 1 & 3 & 1 & 3 & 1 \\ 3 & 2 & 4 & 6 & 0 \end{bmatrix} + \begin{bmatrix} -2 & 1 & 2 & -1 & 0 \\ 0 & 0 & 2 & 3 & -1 \\ -1 & 2 & 3 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 7 & 0 & 0 \\ 1 & 3 & 3 & 6 & 0 \\ 2 & 4 & 7 & 5 & 0 \end{bmatrix} \end{aligned}\right\} \qquad (1.3)$$

Matrix subtraction may be defined in a corresponding way to matrix addition. Scalar multiplication of a matrix is such that all the elements of the matrix are

Table 1.2 Transactions of Tom, Dick and Harry (negative terms imply expenditure)

	Coins				
	1 p	2 p	5 p	10 p	50 p
Tom	−2	1	2	−1	0
Dick	0	0	2	3	−1
Harry	−1	2	3	−1	0

multiplied by the scalar. From these definitions it follows that, for instance, the matrix equation

$$\mathbf{A} = \mathbf{B} + \mu\mathbf{C} - \mathbf{D} \tag{1.4}$$

where **A**, **B**, **C** and **D** are 3 × 2 matrices and μ is a scalar, is equivalent to six simple linear equations of the form

$$a_{ij} = b_{ij} + \mu c_{ij} - d_{ij} \tag{1.5}$$

1.3 MATRIX MULTIPLICATION

Two matrices may only be multiplied if the number of columns of the first equals the number of rows of the second, in which case they are said to be *conformable*. If matrix **A**, of order $m \times p$, is multiplied by matrix **B**, of order $p \times n$, the product

$$\mathbf{C} = \mathbf{AB} \tag{1.6}$$

is of order $m \times n$ with typical element

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj} \tag{1.7}$$

With **A** as in equation (1.1) and

$$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 5 \\ 1 & 10 \\ 1 & 50 \end{bmatrix} \tag{1.8}$$

the product matrix **C** = **AB** is given by

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{bmatrix} = \begin{bmatrix} 2 & 0 & 5 & 1 & 0 \\ 1 & 3 & 1 & 3 & 1 \\ \boxed{3 \;\; 2 \;\; 4 \;\; 6 \;\; 0} \end{bmatrix} \begin{bmatrix} \boxed{1} & 1 \\ \boxed{1} & 2 \\ \boxed{1} & 5 \\ \boxed{1} & 10 \\ \boxed{1} & 50 \end{bmatrix} = \begin{bmatrix} 8 & 37 \\ 9 & 92 \\ \boxed{15} & 87 \end{bmatrix} \tag{1.9}$$

The rule that the element c_{ij} is obtained by scalar multiplication of row i of **A** by column j of **B** has been illustrated for c_{31} by including the relevant elements in boxes. The choice of the matrix **B** has been such that the matrix **C** yields, in its first column, the total number of coins held by each boy and, in its second column, the total value, in pence, of the coins held by each boy.

Except in special cases the matrix product **AB** is not equal to the matrix product **BA**, and hence the order of the matrices in a product may not, in general. be reversed (the product **BA** may not even be conformable). In view of this it is not adequate to say that **A** is multiplied by **B**; instead it is said that **A** is *postmultiplied* by **B** or **B** is *premultiplied* by **A**. Unless either **A** or **B** contain zero elements the total number of multiplications necessary to evaluate **C** from equation (1.6) is $m \times p \times n$, with almost as many addition operations. Matrix multiplication can therefore involve a great deal of computation when m, p and n are all large.

If the multiplication of two large matrices is to be performed by hand it is advisable to include a check procedure to avoid errors. This can be done by including an extra row of column sums in **A** and an extra column of row sums in **B**. The resulting matrix **C** will then contain both row and column sum checks which enable any incorrect element to be pinpointed. With this checking procedure equation (1.9) would appear as

$$\begin{array}{c} \\ \\ \\ \Sigma \end{array}\begin{bmatrix} 2 & 0 & 5 & 1 & 0 \\ 1 & 3 & 1 & 3 & 1 \\ 3 & 2 & 4 & 6 & 0 \\ \hline 6 & 5 & 10 & 10 & 1 \end{bmatrix} \overset{\Sigma}{\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 5 & 6 \\ 1 & 10 & 11 \\ 1 & 50 & 51 \end{array}\right]} = \begin{array}{c} \\ \\ \\ \Sigma \end{array}\overset{\Sigma}{\left[\begin{array}{cc|c} 8 & 37 & 45 \\ 9 & 92 & 101 \\ 15 & 87 & 102 \\ \hline 32 & 216 & 248 \end{array}\right]} \qquad (1.10)$$

Multiple products

If the product matrix **C** of order $m \times n$ (equation 1.6) is further premultiplied by a matrix **D** of order $r \times m$, the final product can be written as

$$\underset{r \times n}{[\mathbf{F}]} = \underset{r \times m}{[\mathbf{D}]} \left(\underset{m \times p}{[\mathbf{A}]} \; \underset{p \times n}{[\mathbf{B}]} \right) \qquad (1.11)$$

It can be verified that the same result for **F** is obtained if the product **DA** is evaluated and the result postmultiplied by **B**. For this reason brackets are left out of multiple products so that equation (1.11) is written as

$$\mathbf{F} = \mathbf{DAB} \qquad (1.12)$$

It is important to note that whereas **D(AB)** has the same value as **DA(B)** the order of evaluation of the products may be very important in numerical computation. For example, if **D** and **A** are of order 100×100 and **B** is of order 100×1, the total number of multiplications for **(DA)B** is 1,010,000 and for **D(AB)** is 20,000. It is therefore going to be roughly fifty times faster to evaluate **R** by multiplying **AB** first. If this calculation were to be performed by hand, the operator would not get far with the multiplication of **AB** before he realized that his method was unnecessarily long-winded. However, if a computer is programmed to evaluate the multiple product the wrong way round this oversight may remain buried in the program without detection. Although such an oversight appears only to involve a penalty of extra computation time, it is likely that the accuracy of the computed results would be less due to the greater accumulation of rounding errors (see section 3.2).