Kalman Filtering: Theory and Practice Using MATLAB

Fourth Edition

Appendix B: A Matrix Refresher

Mohinder S. Grewal

California State University at Fullerton

Angus P. Andrews

Retired from Rockwell Science Center, Thousand Oaks, California

Wiley

Contents

PREFACE

This overview of matrix mathematics is to accompany the Wiley textbook

Kalman Filtering: Theory and Practice Using Matlab, Fourth Edition, 2015,
by M. S. Grewal and A. P. Andrews.

This material had been included as the second Appendix in prior editions.

This is intended for readers familiar with the general subject of linear algebra, but whose recall may be a little rusty. Matrix mathematics can get a bit slippery, so this an attempt to develop the mathematical concepts needed for Kalman filtering within the general framework of matrix theory. A more thorough treatment can be found in most college-level textbooks on linear algebra and matrix theory (e.g., [2, 3, 4, 5, 7, 9, 10, 11, 12, 15, 17, 23, 26, 27, 29, 30, 32, 33, 35]).

This overview introduces the essential notation and properties of matrices as data structures and algebras, as used in derivations of matrix formulas for more efficient and reliable Kalman filter implementations, and in methods for analyzing Kalman filter performance.

The essential mathematics of computation specific to Kalman filtering is covered in [18], and results specific to matrix operations in general are already built into MATLAB®. MATLAB is the result of decades of concerted effort funded by our National Laboratories and governmental agencies, starting in the 1960s and involving work by many university professors and graduate students. The objective of that effort was to develop algorithms which make the best use of computer arithmetic processing capabilities to solve problems involving matrices. Initial results were primarily written in Fortran or Algol. MathWorks was founded in 1984 to recode these algorithms in C and embed them in a higher-order matrix-level programming language which could be ported to most computer systems. This has been augmented by compatible graphics user interfaces and scores of applications-specific "tool boxes," including some for general Kalman filtering and some for the specific applications covered in Chapter 10.

Appendix B

A Matrix Refresher

A mathematician is a device for turning coffee into theorems.
*— **Paul Erdős** (1913–1996)*

In 1850 the English mathematician James Joseph Sylvester (1814–1897) introduced the term "matrix[1]" for a regular array of numbers, but the use of arrays of numbers for solving simultaneous equations goes back many centuries before that. Much of matrix theory as we know it would take shape in the Nineteenth Century and the first half of the Twentieth Century, and much of the associated development of better numerical methods for matrix operations would occur in the second half of the Twentieth Century. However, both the theory and the methods are still evolving today.

The following is a brief survey of some of the nomenclature, notation, and results from matrix theory which have been useful in Kalman filtering. It should provide the mathematical background for most of the material in the textbook.

B.1 Matrix Forms

B.1.1 Notation for Real Matrices

B.1.1.1 Scalars

For the purposes of this book, **scalars** are real numbers, although applications outside Kalman filtering may include complex numbers or other types of the mathematical algebras known as *fields.*

In computer implementations, real numbers must be approximated by fixed-point or floating-point numbers, which are only finite subsets of the rational numbers. (MATLAB generally uses ANSI standard 64-bit floating-point arithmetic.)

Intervals We will use parentheses to denote open intervals (intervals on the real line not including the designated endpoints), so that $(-\infty, +\infty)$ denotes the set of all real numbers and $(0, +\infty)$ denotes the positive real numbers. We will use square brackets to denote closed ends (including the designated endpoint) of intervals, so that [0, 1] denotes the unit interval with endpoints 0 and 1 included.

[1] From the Latin word for "womb," derived from the Latin word for "mother" (*mater*), but also used in the broader sense as a place where things are formed.

B.1.1.2 Real Matrices

For positive integers m and n, an m-by-n real **matrix** A is a two-dimensional rectangular array of scalars, designated by the subscript notation a_{ij} and usually displayed in the following format:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}, \tag{B.1}$$

with **rows** stacked top to bottom and **columns** stacked left to right.

The scalars a_{ij} are called the **elements** of A. We will use upper case letters to denote matrices and the corresponding lowercase letters to denote scalar elements of the associated matrices.

B.1.1.3 Indices and Subscripts

The first subscript (i) on the element a_{ij} refers to the **row** in which the element occurs, and the second subscript (j) refers to the **column** in which a_{ij} occurs in this format. The integers i and j in this notation are also called **indices** of the elements. The first index is called the **row index**, and the second index is called the **column index** of the element. The term "(ij)*th position*" in the matrix A refers to the position of a_{ij}, and a_{ij} is called the "(ij)*th element*" of A.

If juxtaposition of subscripts leads to confusion, they may be separated by commas. The element in the eleventh row and first column of the matrix A would then be denoted by $a_{11,1}$, not a_{111}.

B.1.1.4 Dimensions

The positive integers m and n are called the **dimensions** of A: m is called the **row dimension** of A and n is called the **column dimension** of A. The dimensions of A may also be represented as $m \times n$, which is to be read "m by n". The symbol "$\times$" in this notation does not indicate multiplication, although the number of elements in the matrix A equals the product mn and this is important for determining memory requirements for data structures to hold A.

B.1.2 Square Matrices

A matrix is called **square** if it has the same row and column dimensions.

Some useful forms of square matrices used in Kalman filter implementations are illustrated in Figure B.1 and described in the following sub-subsections.

Note that the matrix forms in the third row of Figure B.1 belong to both forms in the column above. That is, diagonal matrices are both upper triangular and lower triangular, identity matrices are both unit upper triangular and unit lower triangular, and square zero matrices are both strictly upper triangular and strictly lower triangular.

B.1.2.1 Sparse and Dense Matrices

A matrix with a "significant fraction" (typically, half or more) of zero elements is called **sparse**, although—to be useful—the patterns of zeros are usually required to be regular in some way. Matrices that are decidedly not sparse are called **dense**, although both sparsity and density are matters of degree. The forms shown in Figure B.1 are sparse, although sparse matrices do not have to be square. Sparsity is an important characteristic for implementation of matrix methods because it can be exploited to reduce computer memory and computational requirements for matrix operations.

Toeplitz and Hankel matrices. There are also dense matrices with a small number of independent elements, such as the Toeplitz and Hankel matrices shown in Figure B.2. Although these can be fully dense, in the sense that there are no zero elements, all elements of $n \times n$ Toeplitz and Hankel matrices can be represented in terms of only $2n - 1$ independent values.

Upper Triangular

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{21} & a_{22} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

Unit Upper Triangular

$$\begin{bmatrix} 1 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 1 & a_{22} & \cdots & a_{2n} \\ 0 & 0 & 1 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Strictly Upper Triangular

$$\begin{bmatrix} 0 & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & 0 & a_{22} & \cdots & a_{2n} \\ 0 & 0 & 0 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Lower Triangular

$$\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

Unit Lower Triangular

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_{21} & 1 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 1 \end{bmatrix}$$

Strictly Lower Triangular

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_{21} & 0 & 0 & \cdots & 0 \\ a_{31} & a_{32} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 0 \end{bmatrix}$$

Diagonal

$$\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

Identity ($\boldsymbol{I}$)

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Zero ($\mathbf{0}$)

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Figure B.1: Special forms of sparse square matrices

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{11} & a_{12} & \ddots & \vdots \\ a_{31} & a_{21} & a_{11} & \ddots & a_{13} \\ \vdots & \ddots & \ddots & \ddots & a_{12} \\ a_{n1} & \cdots & a_{31} & a_{21} & a_{11} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{13} & \iddots & \iddots & a_{2n} \\ a_{13} & \iddots & \iddots & \iddots & a_{3n} \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{nn} \end{bmatrix}$$

Figure B.2: Toeplitz (left) and Hankel (right) matrices.

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \ddots & \vdots \\ 0 & 1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

Figure B.3: Shift matrices.

Shift matrices. There are also special Toeplitz matrices called "shift matrices," which are zero except for ones on either the super-diagonal or sub-diagonal, as shown in Figure B.3. A 1979 paper by Kailath *et al.* [25] showed how these define a new matrix characterization called "displacement rank," and how displacement rank is related to the degree to which arithmetic operations in matrix algebra (including Kalman filter implementations) can be parallelized in certain multiprocessor architectures.

The shift matrix on the left may be recognized as the dynamic coefficient matrix for a vector $x(t)$ with components

$$\dot{x}_k(t) \stackrel{\text{def}}{=} x_{k+1}(t).$$

A shift matrix with an additional 1 in the opposite corner is a cyclical permutation matrix.

B.1.2.2 Triangular Matrices

These sparse matrix forms have come to play a significant role in the implementation of Kalman filtering.

The **main diagonal** of a square matrix A is the set of elements a_{ij} for which $i = j$. The other elements are called **off-diagonal**.

Square matrices in which all the off-diagonal elements on one side of the main diagonal are zero are called **triangular**.

Triangular matrices with all their non-zero elements on or above the main diagonal are called **upper triangular**..

Triangular matrices with all their non-zero elements on or below the main diagonal are called **lower triangular**..

B.1.2.3 Diagonal Matrices

If all the off-diagonal elements of a square matrix A are zero, A is called a **diagonal** matrix.

An $n \times n$ diagonal matrix D can be defined by specifying its n diagonal elements $\{d_i \,|\, 1 \leq i \leq n\}$, as

$$D = \text{diag}\,(d) \tag{B.2}$$

$$\stackrel{\text{def}}{=} \begin{bmatrix} d_1 & 0 & 0 & \ldots & 0 \\ 0 & d_2 & 0 & \ldots & 0 \\ 0 & 0 & d_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & d_n \end{bmatrix} \tag{B.3}$$

in terms of the math operator "diag" transforming the n-vector d into a square matrix D with diagonal elements equal to the respective components of d.

B.1.2.4 Zero Matrices

The ultimate sparse matrix is a matrix in which *all* elements are 0 (zero). It is called a **zero matrix**, and it is represented by the symbol "0" (zero). The equation $A = 0$ indicates that A is a zero matrix.

Zero matrices may be non-square, as well as square. Whenever it is necessary to specify the dimensions of a zero matrix, they may be indicated by subscripting: $0_{m \times n}$ will indicate an $m \times n$ zero matrix. If the matrix is square, only one subscript will be used: 0_n will mean an $n \times n$ zero matrix. This is also the convention used in MATLAB, where `zeros(m,n)` creates an $m \times n$ zero matrix and `zeros(n)` creates an $n \times n$ zero matrix.

B.1.2.5 Identity Matrices

The identity matrix will be represented by the symbol I. If it is necessary to denote the dimension of I explicitly, it will be indicated by subscripting the symbol: I_n (`eye(n)` in MATLAB) denotes the $n \times n$ identity matrix.

B.2 Vectors

A **vector** is a one-dimensional array of scalars, either horizontal (a **row vector**) or vertical (a **column vector**). Unless otherwise specified, vectors can be assumed to be column vectors.

B.2.1 n-vectors

If the number of elements in a row or column vector is n, it may be referred to as an "n-vector" —usually assumed to be a column vector unless specified otherwise.

B.2.2 Row Vector Notation

Commas can be used for separating the elements of a row vector:

$$x = [x_1, x_2, x_3, \ldots, x_n], \tag{B.4}$$

where the notation "x_i" refers to the element in the ith column of x.

B.2.3 Column Vectors Are the Default

When matrices are used in expressions to represent linear transformations on vectors, we usually put the vector on the right-hand side of the matrix. Consequently, whenever an n-vector x is not defined to be a row vector, it is implied that it is a column vector,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}. \tag{B.5}$$

B.3 Conformable Matrix Dimensions

When a matrices are used in expressions, the matrices involved are assumed to obey all the necessary constraints, or—equivalently—to be "conformable" with their usage.

B.3.1 Syntax of Mathematical Expressions

Syntax is a set of rules governing the formation of patterns of symbols in a language. In mathematics, these symbols may stand for data structures such as scalars and matrices, operators such as addition and multiplication, or delimiters such as parentheses and brackets. Patterns of these symbols satisfying the rules of syntax are called **expressions**. Within the constraints imposed by syntax, an expression of a particular type (e.g., a scalar or a matrix expression) can be substituted for a symbol of that type, and vice versa.

B.3.2 Syntax for Matrix Dimensions

For matrix expressions, there are additional rules of syntax related to the dimensions of the matrices. For example, whenever we write a matrix equation as $A = B$, we assume that the matrices (or matrix expressions) represented by the symbols A and B have the same dimensions.

B.3.3 Implied Conformability for Matrix Operations

Additional rules of syntax for matrix operations will be introduced with the operators. Matrices whose dimensions conform to these rules for a position in a particular expression are said to be **conformable** for that position in that expression. Whenever a symbol for a matrix appears in a matrix expression, it is implied that the matrix represented by that symbol is conformable for its position in that expression.

B.4 Matrix Operations

B.4.1 Extracting Elements of Matrix Expressions

B.4.1.1 Subscripted Expressions

Subscripts represent an operation on a matrix that extracts the designated matrix element. Subscripts may also be applied to matrix expressions. The element in the ij^{th} position of a matrix expression can be indicated by subscripting the expression, as in

$$\left\{A^T\right\}_{ij} = a_{ji}. \tag{B.6}$$

Here, we have used braces {} to indicate the scope of the expression to which the subscripting applies, and the post-superscript "^{T}" to indicate matrix transposition (defined in section B.4.2.1). This subscripting notation is a handy device for defining matrix operations.

A similar notation can be used to extract a row or column from a matrix, using a "dot" ($\cdot$) to indicate the unconstrained indexed dimension:

$$\{A\}_{i\cdot} = i^{\text{th}} \text{ row} \tag{B.7}$$

$$\{A\}_{\cdot j} = j^{\text{th}} \text{ column}. \tag{B.8}$$

These are analogous to the MATLAB script expressions `A(i,:)` and `A(:,j)`, respectively.

B.4.2 Transposition

B.4.2.1 Transpose of a Matrix

All matrices are conformable for transposition. The **transpose** of A is the matrix A^T (with the superscript "T" denoting the transpose operation[2]), obtained from A by interchanging rows and columns:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{m1} \\ a_{12} & a_{22} & a_{32} & \cdots & a_{m2} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{m3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & a_{3n} & \cdots & a_{mn} \end{bmatrix}. \tag{B.9}$$

The transpose of an $m \times n$ matrix is an $n \times m$ matrix.

B.4.2.2 Transpose of a Vector

The transpose of a row n-vector is a column n-vector, and vice versa. It makes more efficient use of space on the page if we express a column vector v as the transpose of a row vector:

$$v = [v_1, v_2, v_3, \ldots, v_m]^T. \tag{B.10}$$

B.4.2.3 Symmetric Matrices

A matrix A is called **symmetric** if $A^T = A$, and **skew symmetric** (or **antisymmetric**) if $A^T = -A$. Only square matrices can be symmetric or skew symmetric. Therefore, whenever a matrix is said to be symmetric or skew symmetric, it is implied that it is a square matrix.

[2] In MATLAB, transposition is indicated by a post-fixed apostrophe, so that A^T is represented in MATLAB as `A'`.

B.4.3 Multiplication by Scalars

All matrices are conformable for multiplication by scalars, either on the left or on the right. Multiplication is indicated by juxtaposition of symbols or by infix notation with the multiplication symbol ($\times$). Multiplication of a matrix A by a scalar s is equivalent to multiplying every element of A by s:

$$\{As\}_{ij} = \{sA\}_{ij} = sa_{ij}. \tag{B.11}$$

B.4.4 Addition and Multiplication of Conformable Matrices

B.4.4.1 Addition of Conformable Matrices Is Associative and Commutative

Matrices are conformable for addition if and only if they share the same dimensions. Whenever matrices appear as sums in an expression, it is implied that they are conformable. If A and B are conformable matrices, then addition is defined by adding corresponding elements:

$$\{A + B\}_{ij} = a_{ij} + b_{ij}. \tag{B.12}$$

Addition of matrices is **commutative** and **associative**. That is, $A + B = B + A$ and $A + (B + C) = (A + B) + C$.

B.4.4.2 Additive Inverse of a Matrix

The product of a matrix A by the scalar -1 yields its **additive inverse** $-A$:

$$(-1)A = -A, \quad A + (-A) = A - A = 0. \tag{B.13}$$

Here, we have followed the not uncommon practice of using the symbol "$-$" both as a unary (additive inverse) and binary (subtraction) operator. **Subtraction** of a matrix A from a matrix B is equivalent to adding the additive inverse of A to B:

$$B - A = B + (-A). \tag{B.14}$$

B.4.4.3 Multiplication of Conformable Matrices Is Associative But Not Commutative

Multiplication of an $m \times n$ matrix A by a matrix B on the right-hand side of A, as in the matrix product AB, is defined only if *the row dimension of B equals the column dimension of A.* That is, we can multiply an $m \times n$ matrix A by a $p \times q$ matrix B in this order only if $n = p$. In that case, the matrices A and B are said to be **conformable for multiplication** in that order, and the matrix product is defined element by element by

$$\{AB\}_{ij} \stackrel{\text{def}}{=} \sum_{k=1}^{n} a_{ik} b_{kj}, \tag{B.15}$$

the result of which is an $m \times q$ matrix. Whenever matrices appear as a product in an expression, it is implied that they are conformable for multiplication.

B.4.4.4 Products with Identity Matrices

Multiplication of any $m \times n$ matrix A by a conformable **identity matrix** yields the original matrix A as the product:

$$AI_n = A = I_m A. \tag{B.16}$$

B.4.5 Vector Inner and Outer Products

These are special names given to products of vectors that are otherwise adequately defined as matrix products.

Two n-vectors of the same dimension (n) have an **inner product**, and two vectors of any dimensions have an **outer product**. For column vectors,

$$\textbf{inner product: } x^T y = \sum_k x_k y_k \tag{B.17}$$

$$\textbf{outer product: } xy^T = \begin{bmatrix} x_1y_1 & x_1y_2 & x_1y_3 & \cdots & x_1y_m \\ x_2y_1 & x_2y_2 & x_2y_3 & \cdots & x_2y_m \\ x_3y_1 & x_3y_2 & x_3y_3 & \cdots & x_3y_m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_ny_1 & x_ny_2 & x_1y_3 & \cdots & x_ny_m \end{bmatrix}, \tag{B.18}$$

where the inner product requires that x and y have the same dimension, but the outer product makes no such demands.

B.4.6 Square Matrix Powers, Roots and Fractions

B.4.6.1 Integer Powers

Square matrices are conformable for multiplication by themselves, and the resulting matrix products are again conformable for multiplication. Consequently, one can define the pth **power** of a square matrix A as

$$A^p = \underbrace{A \times A \times A \times \cdots \times A}_{p \text{ elements}}. \tag{B.19}$$

B.4.6.2 Matrix Polynomials

A matrix polynomial is a finite linear combination of powers of a matrix,

$$\sum_{j=0}^{N} s_n A^j,$$

where the s_j are scalars and N is the **order** of the polynomial.

B.4.6.3 Characteristic Polynomials

The characteristic polynomial of a square matrix A is

$$\det(\lambda I - A) = 0, \tag{B.20}$$

where det is the matrix determinant operator and λ as the unknown variable.

All square matrices A satisfy their own characteristic polynomials. That is, if

$$\det(\lambda I - A) = \sum_{j=0}^{n} a_j \lambda^j, \tag{B.21}$$

then

$$\sum_{j=0}^{n} a_j A^j = 0, \text{ the zero matrix.} \tag{B.22}$$

B.4.6.4 Eigenvalues and Eigenvectors

The solutions λ_i of the characteristic polynomial of A are called the **roots** of the characteristic polynomial, the **characteristic values** of A, or the **eigenvalues** of A. For each eigenvalue λ_i there is a corresponding unit **eigenvector** e_i of A such that

$$Ae_i = \lambda_i e_i. \tag{B.23}$$

Characteristic vectors corresponding to distinct characteristic values are mutually orthogonal.

B.4.6.5 Matrix Power Series

For any square matrix A, a convergent power series

$$\sum_{j=0}^{\infty} a_j A^j$$

represents an analytic function of A. The matrix exponential, for example, can be represented by such a power series.

B.4.6.6 Matrix Square Roots

A matrix square root[3] of a square matrix P is any solution S of the matrix equation $P = S^2$. The matrix square root is not necessarily unique, however. If S is a square root of P, then so is $-S$. Even the zero matrix does not necessarily have a unique square root. For example,

$$\begin{bmatrix} 0 & \pm 1 \\ 0 & 0 \end{bmatrix} \text{ is a square root of } \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

B.4.7 Matrix Multiplicative Inverses

B.4.7.1 Inverses of Nonsingular Square Matrices

If A and B are square matrices of the same dimension and such that their product

$$AB = I, \tag{B.24}$$

then B is the **matrix inverse** of A and A is the matrix inverse of B. (It turns out that $BA = AB = I$ in this case.) The inverse of a matrix A is unique, if it exists, and is denoted by A^{-1}. Not all matrices have inverses. **Matrix inversion** is the process of finding a matrix inverse, if it exists. If the inverse of a matrix A does not exist, A is called **singular**. Otherwise, it is called **nonsingular**.

B.4.7.2 Generalized Inverses

Even non-square or singular matrices can have **generalized inverses**. The **Moore–Penrose**[4] **generalized inverse** of an $m \times n$ matrix A is the $n \times m$ matrix $A^\dagger$ such that

$$AA^\dagger A = A, \quad A^\dagger AA^\dagger = A^\dagger, \tag{B.25}$$

$$(AA^\dagger)^T = AA^\dagger, \quad (A^\dagger A)^T = A^\dagger A. \tag{B.26}$$

The MATLAB function `pinv(M)` computes the Moore–Penrose generalized inverse of the matrix M using the singular value decomposition of M. This result is sometimes called the *pseudo-inverse* of M, because it is based on the *pseudo-rank* of M, which is the effective rank[5] within machine precision tolerance. (The "`p`" in "`pinv`" stands for "pseudo.")

B.4.7.3 Matrix Fractions and the Riccati Equation

For square matrices A and B of the same dimension, the expression A/B or its equivalent AB^{-1} is called a **matrix fraction**. An interesting feature of the nonlinear differential equation known as the **Riccati equation** is that it can be transformed into a system of linear equations if the scalar variable is replaced by a scalar fraction with independent variables as its numerator and denominator. An even more interesting feature is that this result by Count Jacopo Riccati (1676–1754) was for a scalar variables, **but the same is true for square matrix variables**.

[3] There is some notational confusion in what is called "square root" Kalman filtering about how the square root of a matrix is defined.
[4] Named for Eliakim Hastings Moore (1862-1932) and Sir Roger Penrose (b. 1931).
[5] Matrix rank is defined in section B.6.2.

B.4.8 Orthogonality

B.4.8.1 Unit, Orthogonal and Orthonormal Vectors

For vectors, **orthogonality** is a pairwise property. Vectors x and y are called **orthogonal** or **normal** if their inner product is zero. If the inner product of a vector x with itself is 1, x is called a **unit vector**. Orthogonal unit vectors are called **orthonormal**.[6]

Jointly orthogonal vectors: A set $\{x_{[1]}, x_{[2]}, x_{[3]}, \ldots\}$ of vectors $x_{[i]}$ is called *jointly orthogonal* if and only if each distinct pair of vectors $\{(x_{[i]}, x_{[j]}) \mid i \neq j\}$ is orthogonal.

B.4.8.2 Orthogonal Matrices

A matrix A is called orthogonal if $A^T = A^{-1}$. These matrices have several useful properties:

- Orthogonality of a matrix A implies that the row vectors of A are jointly orthonormal, and the column vectors of A are also jointly orthonormal.
- The dot products of vectors are invariant under multiplication by a conformable orthogonal matrix. That is, if A is orthogonal, then $x^T y = (Ax)^T (Ay)$ for all conformable x and y.
- Products and inverses of orthogonal matrices are orthogonal.

As a rule, multiplications by orthogonal matrices tend to be numerically well conditioned—compared to general matrix multiplications. (The inversion of orthogonal matrices is obviously extremely well conditioned.)

B.4.9 Square Matrix Subalgebras

Certain subclasses of $n \times n$ (square) matrices have the property that their products belong to the same class. Orthogonal matrices, for example, have the property that their products are also orthogonal matrices. These are said to form a multiplicative **subalgebra** of $n \times n$ matrices. Subalgebras have the property that their set intersections are also subalgebras. Upper triangular matrices and lower triangular matrices are two subalgebras of the square matrices that are used in implementing the Kalman filter. Their intersection is the set of diagonal matrices—another subalgebra. A Venn diagram of such multiplicative subalgebras is shown in Figure B.4.

B.5 Matrix Factoring and Decompositions

Matrix decompositions are also called **factorizations** of matrices. They are formulas for representing a matrix as a product of matrix factors with useful properties—such as being diagonal, orthogonal, or triangular.

Decomposition methods are algorithms for computing the factors, given the matrix to be factored.

B.5.1 QR Decomposition and Triangularization

The QR **decomposition** of a matrix A is a representation in the form

$$A = QR, \tag{B.27}$$

where Q is an orthogonal matrix and R is a triangular matrix. Methods for the QR decomposition of a matrix are described in Chapter 7—although the matrices labeled "Q" and "R" have different definitions in Kalman filtering.

The term "**triangularization**" generally refers to algorithms for transforming A to R without explicitly computing Q, often done "in-place" by overwriting A with R.

[6] The term "normal" has many meanings in mathematics. Its use here is synonymous with "of unit length". However, it is also used to mean "orthogonal".

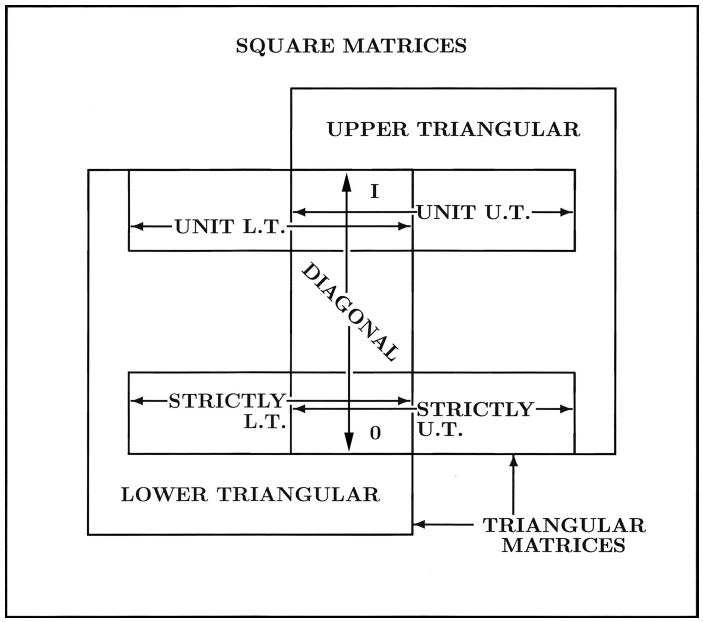
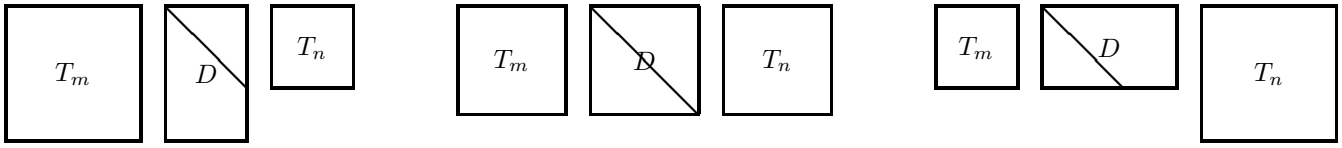


Figure B.4: Some useful multiplicative subalgebras of square matrices



(a) $m > n$ (b) $m = n$ (c) $m < n$

Figure B.5: Singular-value decompositions.

B.5.2 Singular-Value Decomposition

The **singular-value decomposition** of an $m \times n$ matrix A is a representation in the form $A = T_m D T_n$, where T_m is an $m \times m$ orthogonal matrix, T_n is an $n \times n$ orthogonal matrix, and D is an $m \times n$ matrix filled with zeros everywhere except along the main diagonal of its maximal upper-left square submatrix. This decomposition will have either of the three forms shown in Figure B.5, depending on the relative values of m and n. In all three cases, the diagonal line in D indicates the locus of the singular values, and the matrix D has the form

The diagonal matrix D in the middle has the block form

$$\begin{aligned} D &= \left\{ \begin{array}{ll} \left[\begin{array}{c} \mathrm{diag}_i\{\sigma_i\} \\ \hline 0_{(m-n)\times n} \end{array} \right] & \text{if } m > n, \\ \mathrm{diag}_i\{\sigma_i\} & \text{if } m = n, \\ \left[\begin{array}{c|c} \mathrm{diag}_i\{\sigma_i\} & 0_{m\times(n-m)} \end{array} \right] & \text{if } m < n, \\ \multicolumn{2}{c}{\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_p \geq 0,} \end{array} \right\} \\ p &= \min(m, n) \end{aligned} \tag{B.28}$$

in all three cases. That is, the diagonal nonzero elements of D are in *descending order* and non-negative. These are called the **singular values** of A. For a proof that this decomposition exists, and an algorithm for computing it, see Golub and Van Loan [16].

The singular values of a matrix characterize many useful matrix properties, such as:

$\|A\|_2 = \sigma_1(A)$.

rank $(A) = r$ such that $\sigma_r < 0$ and either $\sigma_{r+1} = 0$ or $r = p$. (The rank of a matrix is defined in Section B.6.2.)

The condition number[7] of a nonsingular square matrix A is defined relative to a matrix norm $\|\cdot\|_p$ as

$$\kappa_p(A) \stackrel{\text{def}}{=} \|A\|_p / \left\|A^{-1}\right\|_p . \tag{B.29}$$

The condition number of a matrix A with respect to the Euclidean norm will be the ratio of the largest to the smallest singular value, σ_1/σ_n.

Singular-value decomposition can also be used for generating the eigenvalue-eigenvector decomposition of symmetric matrices (next sub-section).

The condition number can be useful for the analysis of state uncertainty covariance matrices P and state transition matrices Φ of Kalman filters, which can be singular or close enough to being singular that numerical roundoff can cause the product $\Phi P \Phi^T$ to be essentially singular.

Covariance matrices used in Kalman filtering are theoretically symmetric and positive definite. Symmetry can usually be retained by replacing the matrix P in question by its symmetric part, defined as

$$\left(P + P^T\right)/2$$

(`P=0.5*(P+P')` in MATLAB script), and positive-definiteness can be tested using the singular value decomposition.

B.5.3 Eigenvalue–Eigenvector Sum Decompositions

If A is a square symmetric positive-definite (SPD) matrix, then the singular value decomposition

$$A = EDE^T$$

is equivalent to the eigenvalue-eigenvector decomposition

$$A = \sum_k e_k \lambda_k e_k^T ,$$

where the λ_k are the eigenvalues of A and the e_k are the corresponding eigenvectors. That is,

$$E = \begin{bmatrix} e_1 & e_2 & e_3 & \dots & e_m \end{bmatrix}$$

and

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_m \end{bmatrix} .$$

These relationships are useful for the analysis of covariance matrices, which are constrained to have nonnegative characteristic values, although their numerical values may stray enough in practice (due to computer roundoff errors) to develop negative characteristic values.

They can also be used in so-called "sigma-point" implementations of the Kalman filter, in which case the sigma-points are the vectors

$$\sqrt{\lambda_1}e_1,\ \sqrt{\lambda_2}e_2,\ \sqrt{\lambda_3}e_3,\ \dots,\ \sqrt{\lambda_m}e_m.$$

[7] The condition number of the matrix A in the linear equation $Ax = b$ bounds the sensitivity of the solution x to variations in b and the sensitivity of the solution to roundoff errors in determining it. The singular-value decomposition may also be used to define the "pseudo-rank" of A as the smallest singular value σ_i such that $\sigma_i < \varepsilon\sigma_1$, where ε is a processor- and precision-dependent constant such that $0 > \varepsilon \ll 1$.

B.5.4 Cholesky Decomposition

B.5.4.1 Symmetric Products of Matrices

The symmetric product P of a matrix C is the product of C and its transpose,

$$P = CC^T. \tag{B.30}$$

B.5.4.2 André Louis Cholesky

This decomposition is named after André Louis Cholesky (1875–1918), a French[8] geodesist and artillery officer, and a casualty of World War I. He discovered a method for solving linear least-squares problems using an algorithm for factoring a symmetric, positive-definite matrix P as a symmetric product of a **lower triangular** matrix L and its transpose:

$$P = LL^T. \tag{B.31}$$

Cholesky was perhaps not the first discoverer[9] of the factoring technique, but his use of the method for solving least-squares problems was unique. His results were published posthumously by a fellow officer, Commandant (Major) Benoit [6], and credited to Cholesky.

Cholesky decomposition is used in several ways for implementing Kalman filters. In the literature, the term **Cholesky factor** usually refers to the **lower triangular** solution L to Eq. B.31 with positive diagonal elements.

B.5.4.3 Square Root Filtering and Matrix Square Roots

James E. Potter (1937–2005) discovered what is now called "square root filtering," an alternative mathematical form of the Riccati equation in the Kalman filter which is much more robust against computer roundoff errors, prompting Gerald G. Bierman (1941–1987) to characterize it as

"providing the same accuracy with half as many bits."

Potter's formulation replaces the symmetric positive-definite covariance matrix P of estimation uncertainty in the matrix Riccati equation[10] by a solution C of Equation B.30.

Potter's derivation uses a symmetric "elementary matrix[11]" of the form $I \pm vv^T$, the square root of which is also a symmetric elementary matrix. However, it would not be considered a Cholesky factor of $I \pm vv^T$, because it is not triangular.

Potter's solution of the equivalent form of the Riccati equation in terms of the matrix C of Eq. B.30 does not require that C be triangular. However, it has been shown that the Potter square-root filter can be augmented to maintain triangularity of C during the temporal update of the square-root filter, in which case it remains a true Cholesky factor [1]. S. F. Schmidt and T. Kailath [24] and others have developed methods for preserving triangularity in the measurement update, as well.

The Potter matrix C in this case is "neither fish nor fowl" according to convention. It is not a Cholesky factor because it is not necessarily lower triangular, and it is not a matrix square-root unless it is symmetric.

In the literature of square-root Kalman filtering, any solution C of Eq. B.30 has come to be called a "square root" of P. Strictly speaking, however, C is not a true matrix square root of P unless C is symmetric.

Symmetric square roots. Every symmetric positive-definite P has a symmetric square root which can be obtained by using the singular value decomposition of P (MATLAB function `svd`) as $P = ADB$, where—for symmetric positive-definite P—A and $B = A^T$ are orthogonal and D is diagonal with positive diagonal entries. A symmetric square root in this case would be $S = A\sqrt{D}A^T$, where $\sqrt{D}$ is the diagonal matrix whose diagonal entries are the positive square roots of the diagonal entries of D.

[8] Because Cholesky was French, his name should perhaps be pronounced "Show-less-KEY" with the accent on the last syllable.

[9] Zurmühl [37] cites an earlier discovery by M. H. Doolittle, published in a U.S. Coast and Geodetic Report in 1878.

[10] The Riccati differential (or difference) equation in this case is the mathematical model for the evolution over time of the mean-squared uncertainty in the Kalman filter estimate.

[11] Alston S. Householder (1904–1993) used elementary matrix algebras to develop many accurate methods in computational mathematics, including some used in Kalman filter implementations.

B.5.5 Generalized Cholesky Factors

As an alternative, we will use the term **generalized Cholesky factor (GCF)** for any solution C of Eq. B.30.

Note that it does not matter whether we write Eq. B.30 with the alternative right-hand-side $P - F^T F$, because the two solutions can be related by $F = C^T$.

Also, this definition would allow the generalized Cholesky factor C to be non-square, having greater column dimension than row dimension. For example, if the matrix A is $n \times n$ and B is $n \times m$, then the symmetric product of the $n \times (n+m)$ partitioned matrix

$$C = \left[\begin{array}{c|c} A & B \end{array}\right] \tag{B.32}$$

is

$$CC^T = \left[\begin{array}{c|c} A & B \end{array}\right] \left[\frac{A}{B}\right] \tag{B.33}$$

$$= AA^T + BB^T, \tag{B.34}$$

which can be an additive decomposition of P. In fact, this type of decomposition is common in Kalman filtering implementations. These generally involve a QR decomposition of the generalized Cholesky factor as $C^T = QR$, where Q is orthogonal and R is triangular, in which case R is a true (i.e., triangular) Cholesky factor of P.

B.5.5.1 Generalized Cholesky Factors Are Not Unique

If C is a generalized Cholesky factor of P, then for any conformable *orthogonal* matrix M, the matrix

$$A \stackrel{\text{def}}{=} CM \tag{B.35}$$

satisfies the equation

$$AA^T = CM(CM)^T \tag{B.36}$$

$$= CMM^T C^T \tag{B.37}$$

$$= CC^T \tag{B.38}$$

$$= P. \tag{B.39}$$

That is, A is also a legitimate generalized Cholesky factor.

The ability to transform one generalized Cholesky factor into another using orthogonal matrices turns out to be very important in "square-root" filtering (Chapter 7).

B.5.6 Alternative Cholesky Factoring Algorithms

The factoring algorithm used by Cholesky always resulted in a lower triangular result. However, there are two possible forms of triangular factorization, corresponding to two possible forms of the defining equation:

$$P = L_1 L_1^T = U_1^T U_1 \tag{B.40}$$

$$= U_2 U_2^T = L_2^T L_2, \tag{B.41}$$

where L_1 is the form used by Cholesky, but the generalized Cholesky factors U_1, U_2 are upper triangular and the transpose $L_2 = U_2^T$ is lower triangular.

The first of these is implemented by the built-in MATLAB function `chol(P)`, with argument `P` a symmetric positive-definite matrix. The script `chol(P)` returns an lower triangular matrix L_1 satisfying Eq. B.40. The MATLAB m-file `utchol.m` in the Wiley website implements the solution to Eq. B.41. The call `utchol(P)` returns an upper triangular matrix U_2 satisfying Eq. B.41.

There are also two possible forms for each of the two factoring algorithms, depending on whether the second level of the indexing loop is by rows or columns, but this detail has no significant effect on the result.

B.5.6.1 Modified Cholesky Factorization

The algorithm for Cholesky factorization of a matrix requires taking square roots, which can be avoided by using a **modified Cholesky factorization** or "UD factorization" in the form

$$P = UDU^T, \tag{B.42}$$

where D is a diagonal matrix with positive diagonal elements and U is a **unit triangular matrix** (i.e., U has 1s along its main diagonal). This algorithm is implemented in the file `modchol.m` on the Wiley website.

B.5.6.2 Rank 1 Modifications of Generalized Cholesky factors

A "rank 1 modification" of a generalized Cholesky factor C such that $CC^T = A$ to a generalized Cholesky factor of $A \pm vv^T$, where v is a column vector and vv^T is a matrix of rank = 1 (hence the name "rank 1 modification"). The MATLAB function `cholupdate(C,v)` performs that function for upper triangular Cholesky factors. Given C and v, for factorization in the form

$$C_{\text{output}}^T C_{\text{output}} = C_{\text{input}}^T C_{\text{input}} + vv^T$$

(i.e., transposing the first factor, rather than the second factor).

Rank 1 generalized Cholesky factor modification functions `potter.m`, `carlson.m`, and `bierman.m` on the Wiley website are specialized for application to Kalman filtering, as described in Chapter 7.

B.6 Block Matrix Formulas

B.6.1 Submatrices, Partitioned Matrices, and Blocks

For any $m \times n$ matrix A and any subset $S_{\text{rows}} \subseteq \{1, 2, 3, \ldots, m\}$ of the row indices and subset $S_{\text{cols}} \subseteq \{1, 2, 3, \ldots, n\}$ of the column indices, the subset of elements

$$A' = \{a_{ij} | i \in S_{\text{rows}}, j \in S_{\text{cols}}\} \tag{B.43}$$

is called a **submatrix** of A.

A **partitioning** of an integer n is an exhaustive collection of contiguous subsets S_k of the form

$$\overbrace{1, 2, 3, \ldots, \ell_1}^{S_1}, \overbrace{(\ell_1 + 1), \ldots, \ell_2}^{S_2}, \ldots, \overbrace{(\ell_{p-1} + 1), \ldots, n}^{S_p}. \tag{B.44}$$

The collection of submatrices formed by partitionings of the row and column dimensions of a matrix is called a partitioning of the matrix, and the matrix is said to be partitioned by that partitioning. Each submatrix of a partitioned matrix A is called a **partitioned submatrix, partition, submatrix block, sub-block**, or **block** of A. Each block of a partitioned matrix A can be represented by a conformable matrix expression, and A can be displayed as a **block matrix**:

$$A = \begin{bmatrix} B & C & D & \cdots & F \\ G & H & J & \cdots & L \\ M & N & P & \cdots & R \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ V & W & X & \cdots & Z \end{bmatrix} \tag{B.45}$$

where $B, C, D, \ldots$ stand for matrix expressions. Whenever a matrix is displayed as a block matrix, it is implied that all block submatrices in the same row have the same row dimension and that all block submatrices in the same column have the same column dimension.

A block matrix of the form

$$\begin{bmatrix} A & 0 & 0 & \cdots & 0 \\ 0 & B & 0 & \cdots & 0 \\ 0 & 0 & C & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & M \end{bmatrix}, \tag{B.46}$$

in which the off-diagonal block submatrices are zero matrices, is called a **block diagonal matrix**, and a block matrix in which the block submatrices on one side of the diagonal are zero matrices is called a **block triangular matrix**.

B.6.1.1 Columns and Rows as Submatrices

There are two special partitionings of matrices in which the block submatrices are vectors. The column vectors of an $m \times n$ matrix A are the block submatrices of the partitioning of A for which all column dimensions are 1 and all row dimensions are m. The row vectors of A are the block submatrices of the partitioning for which all row dimensions are 1 and all column dimensions are n. All column vectors of an $m \times n$ matrix are m-vectors, and all row vectors are n-vectors.

For example, the eigenvalue-eigenvector sum decompositions of symmetric positive-definite matrices in section B.5.3 use the columns of the orthogonal factor E as submatrices.

B.6.2 Rank and Linear Dependence

A **linear combination** of a finite set of n-vectors $\{\nu_i\}$ is a summation of the sort $\sum_i a_i v_i$ for some set of scalars $\{a_i\}$. If some linear combination $\sum a_i v_i = 0$ and at least one coefficient $a_i \neq 0$, the set of vectors $\{v_i\}$ is called **linearly dependent**. Conversely, if the only linear combination for which $\sum a_i v_i = 0$ is the one for which all the $a_i = 0$, then the set of vectors $\{v_i\}$ is called **linearly independent**.

The **column rank** of an $n \times m$ matrix A equals the size of the *largest* collection of its column vectors that is linearly independent. Note that any such linear combination can be expressed in the form Aa, where the nonzero elements of the column m-vector a are the associated scalars of the linear combination, and the number of nonzero components of a is the size of the collection of column vectors in the linear combination.

The **row rank** of a $n \times m$ matrix A equals the size of the *largest* collection of its row vectors that is linearly independent. In this case, any linear combination or row vectors can be expressed in the form $a^T A$ for some column n-vector a.

The rank of a square $n \times n$ matrix A equals both its row rank and its column rank. In this case, A is nonsingular if and only if its rank equals its dimension n.

B.6.3 Conformable Block Operations

Block matrices with conformable partitionings may be transposed, added, subtracted, and multiplied in block format. For example,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^T = \begin{bmatrix} A^T & C^T \\ B^T & D^T \end{bmatrix}, \tag{B.47}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} + \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} A+E & B+F \\ C+G & D+H \end{bmatrix}, \tag{B.48}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \times \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}. \tag{B.49}$$

B.6.4 Frobenius–Schur Inversion Formula

The inverse of a partitioned matrix with square diagonal blocks may be represented in block form as[12]

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & F \\ G & H \end{bmatrix}. \tag{B.50}$$

[12]This formula has had many discoverers. Bodewig [8] cites nine such discoverers but gives credit to the German mathematicians Georg Ferdinand Frobenius (1849–1917) and Issai Schur (1875–1941) as the earliest discovers of record.

where

$$E = A^{-1} + A^{-1}BHCA^{-1}, \tag{B.51}$$
$$F = -A^{-1}BH, \tag{B.52}$$
$$G = -HCA^{-1}, \tag{B.53}$$
$$H = [D - CA^{-1}B]^{-1}, \tag{B.54}$$

provided that the critical matrix expressions are nonsingular. This formula can be proved by multiplying the original matrix times its alleged inverse and verifying that the result is the identity matrix.

B.6.5 Inversion Formulas for Matrix Expressions

B.6.5.1 Sherman—Morrison Formula[7]

A "rank 1" modification of a square matrix A is a sum of the form $A + bc^T$, where b and c are conformable column vectors. Its inverse is given by the formula

$$[A + bc^T]^{-1} = A^{-1} - \frac{A^{-1}bc^TA^{-1}}{1 + c^TA^{-1}b}. \tag{B.55}$$

B.6.5.2 Sherman—Morrison—Woodbury Formula

This is the generalization of the above formula for conformable matrices in place of vectors:

$$[A + BC^T]^{-1} = A^{-1} - A^{-1}B[I + C^TA^{-1}B]^{-1}C^TA^{-1}. \tag{B.56}$$

B.6.5.3 Duncan Inversion Formula

A further generalization of this formula (used in the derivation of the Kalman filter equations) includes an additional conformable square matrix factor in the modification[8]

$$[A + BC^{-1}D^T]^{-1} = A^{-1} - A^{-1}B[C + D^TA^{-1}B]^{-1}D^TA^{-1}. \tag{B.57}$$

B.7 Functions of Square Matrices

B.7.1 Determinants

B.7.1.1 Elementary Permutation Matrices

An **elementary permutation matrix** is formed by interchanging rows or columns of an identity matrix I_n:

$$P_{[ij]} = \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccccccc} & & & & i & & j & & \end{array} \\ \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 & \end{pmatrix} \end{array}. \tag{B.58}$$

Multiplication of a vector x by $P_{[ij]}$ permutes the ith and jth elements of x. Note that $P_{[ij]}$ is an *orthogonal* matrix and a square root of the identity matrix. That is, $P_{[ij]}^2 = I_n$.

[7] The naming of this and the next formula follows the convention of Golub and Van Loan [16], which is at odds with earlier publications found by Henderson and Searle [22], and by Hager [21]

[8] This is yet another formula with many discoverers. Fortmann [14] cites several of them. Bodewig ([8], page 218) credits H. Hemes for its discovery, although Henderson and Searle [22] cite earlier publications by Duncan [13] (1944) and Guttman [19] (1946).

B.7.1.2 Determinants of Elementary Permutation Matrices

The **determinant** of an elementary permutation matrix $P_{[ij]}$ is defined to be -1, unless $i = j$ (i.e., $P_{[ii]} = I_n$):

$$\det(P_{[ij]}) \stackrel{\text{def}}{=} \begin{cases} -1, & i \neq j, \\ +1, & i = j. \end{cases} \tag{B.59}$$

B.7.1.3 Permutation Matrices

A **permutation matrix** is any product of elementary permutation matrices. These are also orthogonal matrices. Let $\mathcal{P}_n$ denote the set of all distinct $n \times n$ permutation matrices. There are $n! = 1 \times 2 \times 3 \times \cdots \times n$ of them, corresponding to the $n!$ permutations of n indices.

B.7.1.4 Determinants of Permutation Matrices

The determinant of a permutation matrix can be defined by the rule that the determinant of a product of matrices is the product of the determinants:

$$\det(AB) = \det(A)\det(B). \tag{B.60}$$

Therefore, the determinant of a permutation matrix will be either $+1$ or -1.

B.7.1.5 Even and Odd Permutation Matrices

A permutation matrix is called "even" if its determinant is $+1$ and "odd" if its determinant equals -1.

B.7.1.6 Determinants of Square Matrices

The determinant of any $n \times n$ matrix A can be defined as follows:

$$\det(A) \stackrel{\text{def}}{=} \sum_{P \in \mathcal{P}_n} \det(P) \prod_{i=1}^{n} \{AP\}_{ii}. \tag{B.61}$$

This formula has $\mathcal{O}(n \times n!)$ computational complexity (for a sum over $n!$ products of n elements each).

The determinant of any square matrix A equals the product of its characteristic values, with each characteristic value occurring as many times in the product as the multiplicity of the associated root of the characteristic polynomial.

B.7.2 Definiteness of Symmetric Matrices

If A is symmetric, all its characteristic values are real numbers, which implies that they can be ordered. They are usually expressed in descending order:

$$\lambda_1(A) \geq \lambda_2(A) \geq \lambda_3(A) \geq \cdots \geq \lambda_n(A). \tag{B.62}$$

A real square symmetric matrix A is called

$$\left.\begin{array}{lllllll} \text{positive definite} & \text{if } \lambda_n(A) & > & 0, & & & \\ \text{non–negative definite} & \text{if } \lambda_n(A) & \geq & 0, & & & \\ \text{indefinite} & \text{if } \lambda_1(A) & > & 0 & \text{and} & \lambda_n(A) < 0, & \\ \text{non-positive definite} & \text{if } \lambda_1(A) & \leq & 0, & \text{and} & & \\ \text{negative definite} & \text{if } \lambda_1(A) & < & 0. & & & \end{array}\right\} \tag{B.63}$$

Non-negative-definite matrices are also called **positive semidefinite**, and non-positive-definite matrices are also called **negative semidefinite**.

B.7.3 Matrix Trace

The **trace** of a square matrix is the sum of its diagonal elements. It also equals the sum of the characteristic values and has the property that the trace of the product of conformable matrices is independent of the order of multiplication—a very useful attribute:

$$\begin{aligned} \text{trace}(AB) &= \sum_i \{AB\}_{ii} && \text{(B.64)} \\ &= \sum_i \sum_j A_{ij} B_{ji} && \text{(B.65)} \\ &= \sum_j \sum_i B_{ji} A_{ij} && \text{(B.66)} \\ &= \text{trace}(BA). && \text{(B.67)} \end{aligned}$$

Note the product AB is conformable for the trace function only if it is a square matrix, which requires that A and B^T have the same dimensions. If they are $m \times n$ (or $n \times m$), then the computation of the trace of their product requires mn multiplications, whereas the product itself would require $m^2 n$ (or mn^2) multiplications.

B.7.4 Algebraic Functions of Matrices

An algebraic function may be defined by an expression in which the independent variable (a matrix) is a free variable, such as the truncated power series

$$f(A) = \sum_{k=-n}^{n} B_k A^k, \tag{B.68}$$

where the negative power $A^{-p} = \{A^{-1}\}^p = \{A^p\}^{-1}$. In this representation, the matrix A is the independent (free) variable and the other matrix parameters (B_k) are assumed to be known and fixed.

B.7.5 Analytic Functions of Matrices

An analytic function is defined in terms of a convergent power series. It is necessary that the power series converge to a limit, and the matrix norms defined in subsection B.8.3 must be used to define and prove convergence of a power series. This level of rigor is beyond the scope of this appendix, but we do need to use one particular analytic function—the exponential function.

B.7.5.1 Exponential Function

The power series

$$e^A = \sum_{k=0}^{\infty} \frac{1}{1 \cdot 2 \cdot 3 \cdots k} A^k \tag{B.69}$$

does converge for all square matrices A, although convergence is not fast enough to make this a reasonable general-purpose formula for approximating the exponential of A. This definition does suffice to prove some elementary properties of the exponential function for matrices, such as:

- $e^{0_n} = I_n$ for 0_n the $n \times n$ zero matrix.
- $e^{I_n} = eI_n$ for I_n the $n \times n$ identity matrix.
- $e^{A^T} = \{e^A\}^T$.
- $(d/dt)e^{At} = Ae^{At} = e^{At}A$.
- The exponential of a skew-symmetric matrix is an orthogonal matrix.
- The characteristic vectors of A are also the characteristic vectors of e^A.
- If λ is a characteristic value of A, then e^λ is a characteristic value of e^A.

B.7.5.2 Computing Matrix Exponentials

In an update [31] of a 1978 journal article titled "Nineteen dubious ways to compute the exponential of a matrix" [31], Moler and Van Loan report their re-evaluations of methods for computing matrix exponentials. Many of the methods tested had serious shortcomings, and no method was declared universally superior. The one presented here was recommended as being more reliable than most. It combines several ideas due to Ward [36], including setting the algorithm parameters to meet a pre-specified error bound. It combines Padé approximation with a technique called "scaling and squaring" to maintain approximation errors within pre-specified bounds.

Padé approximation of the matrix exponential Padé approximations of functions by rational functions (ratios of polynomials) date from a 1892 publication [34] by Henri Padé.[9] They have been used in deriving solutions of differential equations, including Riccati equations[10] [28]. They can also be applied to functions of matrices, including the matrix exponential. In the matrix case, the power series is approximated as a "matrix fraction" of the form $\mathcal{D}^{-1}\mathcal{N}$, with the numerator matrix ($\mathcal{N}$) and denominator matrix ($\mathcal{D}$) represented as polynomials with matrix arguments. The "order" of the Padé approximation is two dimensional. It depends on the orders of the polynomials in the numerator and denominator of the rational function. The Taylor series is the special case in which the order of the denominator polynomial of the Padé approximation is zero. Like the Taylor series approximation, the Padé approximation tends to work best for small values of its argument. For matrix arguments, it will be some matrix norm[11] of the argument that will be required to be small.

The exponential function with argument z has the power series expansion

$$e^z = \sum_{k=0}^{\infty} \frac{1}{k!} z^k. \tag{B.70}$$

The polynomials $\mathcal{N}_p(z)$ and $\mathcal{D}_q(z)$ such that

$$\mathcal{N}_p(z) = \sum_{k=0}^{p} a_k z^k, \tag{B.71}$$

$$\mathcal{D}_q(z) = \sum_{k=0}^{q} b_k z^k, \tag{B.72}$$

$$e^z \mathcal{D}_q(z) - \mathcal{N}_p(z) = \sum_{k=p+q+1}^{\infty} c_k z^k \tag{B.73}$$

are the numerator and denominator polynomials, respectively, of the Padé approximation of e^z. The key feature of the last equation is that there are no terms of order $\leq p+q$ on the right-hand side. This constraint is sufficient to determine the coefficients a_k and b_k of the polynomial approximants, except for a common constant factor. The solution (within a common constant factor) will be [28]

$$a_k = \frac{p!(p+q-k)!}{k!(p-k)!}, \quad b_k = \frac{(-1)^k q!(p+q-k)!}{k!(q-k)!}. \tag{B.74}$$

Application to matrix exponential The above formulas may be applied to polynomials with scalar coefficients and square matrix arguments. For any $n \times n$ matrix X,

$$f_{pq}(X) = \left(q! \sum_{i=0}^{q} \frac{(p+q-i)!}{i!(q-i)!} (-X)^i \right)^{-1} \left(p! \sum_{i=0}^{p} \frac{(p+q-i)!}{i!(p-i)!} X^i \right) \tag{B.75}$$

$$\approx e^X \tag{B.76}$$

is the Padé approximation of e^X of order $(p,\ q)$.

[9] Pronounced *"pah-DAY"*.

[10] The order of the numerator and denominator of the matrix fraction are reversed here from the order used in linearizing the Riccati equation in Chapter 5.

[11] Defined in subsection B.8.3.

Bounding relative approximation error The bound given here is from Moler and Van Loan [31]. It uses the ∞-norm of a matrix, which can be computed[12] as

$$\|X\|_\infty = \max_{1\le i\le n}\left(\sum_{j=1}^{n}|x_{ij}|\right) \tag{B.77}$$

for any $n \times n$ matrix X with elements x_{ij}. The **relative** approximation error is defined as the ratio of the matrix ∞-norm of the approximation error to the matrix ∞-norm of the right answer. The relative Padé approximation error is derived as an analytical function of X in Moler and Van Loan [31]. It is shown in Golub and Van Loan [16] that it satisfies the inequality bound

$$\frac{\|f_{pq}(X) - e^X\|_\infty}{\|e^X\|_\infty} \le \varepsilon(p,q,X)e^{\varepsilon(p,q,X)}, \tag{B.78}$$

$$\varepsilon(p,q,X) = \frac{p!q!2^{3-p-q}}{(p+q)!(p+q+1)!}\|X\|_\infty. \tag{B.79}$$

Note that this bound depends only on the **sum** $p + q$. In that case, the computational complexity of the Padé approximation for a given error tolerance is minimized when $p = q$, that is, if the numerator and denominator polynomials have the same order.

The problem with the Padé approximation is that the error bound grows exponentially with the norm $\|X\|_\infty$. Ward [36] combined scaling (to reduce $\|X\|_\infty$ and the Padé approximation error) with squaring (to re-scale the answer) to obtain an approximation with a predetermined error bound. In essence, one chooses the polynomial order to achieve the given bound.

Scaling and squaring Note that, for any nonnegative integer N,

$$e^X = (e^{2^{-N}X})^{2^N} \tag{B.80}$$

$$= \underbrace{\{[(\cdots e^{2^{-N}X}\cdots)^2]^2\}^2}_{N \text{ squarings}}. \tag{B.81}$$

Consequently, X can be "down-scaled" by 2^{-N} to obtain a good Padé approximation of $e^{2^{-N}X}$, then "up-scaled" again (by N squarings) to obtain a good approximation to e^X.

MATLAB Implementations The built-in MATLAB function `expm(M)` is essentially the one recommended by Moler and Van Loan [31], as implemented by Golub and Van Loan [16, Algorithm 11.3.1, page 558]. It combines scaling and squaring with a Padé approximation for the exponential of the scaled matrix, and it is designed to achieve a specified tolerance of the approximation error. The MATLAB m-file `expm1.m` on the Wiley website is a demonstration of `expm`.

MATLAB also includes the functions `expm2` (Taylor series approximation) and `expm3` (alternative implementation using eigenvalue–eigenvector decompositions), which can be used to test accuracies and speeds relative to `expm` of these alternative implementations of the matrix exponential function.

B.7.6 Similarity Transformations and Analytic Functions

For any $n \times n$ nonsingular matrix A, the transform $X \to A^{-1}XA$ is called a **similarity transformation** of the $n \times n$ matrix X. It is a useful transformation for analytic functions of matrices

$$f(X) = \sum_{k=0}^{\infty} a_k X^k, \tag{B.82}$$

[12] This formula is not the definition of the ∞-norm of a matrix, which is defined in section B.8.3.4. However, it is a consequence of the definition, and it can be used for computing it.

because

$$f(A^{-1}XA) = \sum_{k=0}^{\infty} a_k (A^{-1}XA)^k \tag{B.83}$$

$$= A^{-1}\left(\sum_{k=0}^{\infty} a_k X^k\right) A \tag{B.84}$$

$$= A^{-1} f(X) A. \tag{B.85}$$

If the characteristic values of X are distinct, then the similarity transform performed with the characteristic vectors of X as the column vectors of A will diagonalize X with its characteristic values along the main diagonal:

$$A^{-1}XA = \operatorname*{diag}_{\ell}\{\lambda_\ell\} \tag{B.86}$$

$$f(A^{-1}XA) = \operatorname*{diag}_{\ell}\{F(\lambda_\ell)\}, \tag{B.87}$$

$$f(X) = A \operatorname*{diag}_{\ell}\{F(\lambda_\ell)\} A^{-1}. \tag{B.88}$$

(Although this is a useful analytical approach for demonstrating functional dependencies, it is not considered a robust numerical method.)

B.8 Norms

B.8.1 Normed Linear Spaces

Vectors and matrices can be considered as elements of *linear spaces,* in that they can be added and multiplied by scalars. A **norm** is *any* nonnegative real-valued function $\|\cdot\|$ defined on a linear space such that, for any scalar s and elements x and y of the linear space (vectors or matrices),

$$\|x\| = 0 \text{ iff } x = 0, \tag{B.89}$$

$$\|x\| > 0 \text{ iff } x \neq 0, \tag{B.90}$$

$$\|sx\| = |s|\|x\|, \tag{B.91}$$

$$\|x + y\| \leq \|x\| + \|y\|, \tag{B.92}$$

where "iff" stands for "if and only if". These constraints are rather loose, and many possible norms can be defined for a particular linear space. A linear space with a specified norm is called a **normed linear space**. The norm induces a **topology** on the linear space, which is used to define continuity and convergence. Norms are also used in numerical analysis for establishing error bounds, and in sensitivity analysis for bounding sensitivities. The multiplicity of norms is useful in these applications, because the user is free to pick the one that works best for her or his particular problem.

We define here many of the more popular norms, some of which are known by more than one name.

B.8.2 Hölder Norms

The inner product of a column n-vector x with itself is

$$x^T x = \text{trace } xx^T \tag{B.93}$$

$$= \sum_{i=1}^{n} x_i^2 \tag{B.94}$$

$$\stackrel{\text{def}}{=} \|x\|_E^2, \tag{B.95}$$

the square of the **Euclidean norm** of x. This is but one of a class of norms called **Hölder norms**,[13] *ℓ_p norms,*[14] or simply *p-norms*:

$$\|x\|_p \stackrel{\text{def}}{=} \left[\sum_{i=1}^{n} |x_i|^p\right]^{1/p}, \tag{B.96}$$

and in the limit (as $p \to \infty$) as the **sup**[15] norm, or ∞-norm:

$$\|x\|_\infty \stackrel{\text{def}}{=} \max_i |x_i|. \tag{B.97}$$

These norms satisfy the **Hölder inequality**:

$$|x^T y| \le \|x\|_p \|y\|_q \text{ for} \frac{1}{p} + \frac{1}{q} = 1. \tag{B.98}$$

They are also related by inequalities such as

$$\|x\|_\infty \le \|x\|_E \le \|x\|_1 \le n\|x\|_\infty. \tag{B.99}$$

The Euclidean norm (2-norm) is the default norm for vectors. When no other norm is identified, the implied norm is the Euclidean norm. The MATLAB function `norm` computes the Euclidean norm of a vector.

B.8.3 Matrix Norms

Many norms have been defined for matrices. Two general types are presented here. Both are derived from vector norms, but by different means.

B.8.3.1 Generalized Vector Norms

Vector norms can be generalized to matrices by treating the matrix like a doubly subscripted vector. For example, the Hölder norms for vectors can be generalized to matrices as

$$\|A\|_{(p)} = \left\{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{i,j}|^p\right\}^{1/p}. \tag{B.100}$$

The matrix (2)-norm defined in this way is also called the **Euclidean norm, Schur norm**, or **Frobenius norm**. We will use the notation $\|\cdot\|_F$ in place of $\|\cdot\|_{(2)}$ for the Frobenius norm.

The reason for putting the parentheses around the subscript p in the above definition is that there is another way that the vector p-norms are used to define matrix norms, and it is this alternative definition that is usually allowed to wear an unadorned p subscript. These alternative norms also have the following desirable properties.

B.8.3.2 Desirable Multiplicative Properties of Matrix Norms

Because matrices can be multiplied, one could also apply the additional constraint that

$$\|AB\|_M \le \|A\|_M \|B\|_M \tag{B.101}$$

for conformable matrices A and B and a matrix norm $\|\cdot\|_M$. This is a good property to have for some applications. One might also insist on a similar property with respect to multiplication by vector x, for which a norm $\|\cdot\|_{V_1}$ may already be defined:

$$\|Ax\|_{V_2} \le \|A\|_M \|x\|_{V_1}. \tag{B.102}$$

This property is called **compatibility** between the matrix norm $\|\cdot\|_M$ and the vector norms $\|\cdot\|_{V_1}$ and $\|\cdot\|_{V_2}$. (Note that there can be two distinct vector norms associated with a matrix norm: one in the normed linear space containing x, and one in the space containing Ax.)

[13] Named for the German mathematician Otto Ludwig Hölder (1859–1937).

[14] This "little ℓ" notation is used for infinite-dimensional normed vector spaces (sequences), which include finite-dimensional normed vector spaces as a subclass.

[15] "sup" (sounds like "soup") stands for **supremum**, a mathematical term for the **least upper bound** of a set of real numbers. The maximum (max) is the supremum over a finite set.

B.8.3.3 Matrix Norms Subordinate to Vector Hölder Norms

There is a family of alternative matrix "p-norms" [but not (p)-norms] defined by the formula

$$\|A\|_p \stackrel{\text{def}}{=} \sup_{\|x\|\neq 0} \frac{\|Ax\|_p}{\|x\|_p}, \tag{B.103}$$

where the norms on the right-hand side are the vector Hölder norms and the induced matrix norms on the left are called **subordinate** to the corresponding Hölder norms. The 2-norm defined in this way is also called the **spectral norm** of A. It has the properties

$$\|\mathrm{diag}_i\{\lambda_i\}\|_2 = \max_i |\lambda_i|, \text{ and } \|Ax\|_2 \leq \|A\|_2\|x\|_2. \tag{B.104}$$

The first of these properties implies that $\|I\|_2 = 1$. The second property is compatibility between the spectral norm and the vector Euclidean norm. (Subordinate matrix norms are guaranteed to be compatible with the vector norms used to define them.) All matrix norms subordinate to vector norms also have the property that $\|I\| = 1$.

B.8.3.4 Computation of Matrix Hölder Norms

The following formulas may be used in computing 1-norms and ∞-norms of $m \times n$ matrices A:

$$\|A\|_1 = \max_{1\leq j\leq n}\left\{\sum_{i=1}^{m}|a_{ij}|\right\}, \tag{B.105}$$

$$\|A\|_\infty = \max_{1\leq i\leq m}\left\{\sum_{j=1}^{n}|a_{ij}|\right\}. \tag{B.106}$$

The norm $\|A\|_2$ can be computed as the square root of the largest characteristic value of A^TA, which takes considerably more effort.

B.8.3.5 Default Matrix Norm

When the type of norm applied to a matrix is not specified (by an appropriate subscript), the default will be the spectral norm (Hölder matrix 2-norm). It satisfies the following bounds with respect to the Frobenius norm and the other matrix Hölder norms for $m \times n$ matrices A:

$$\left.\begin{array}{lcll} \|A\|_2 & \leq & \|A\|_F & \leq \sqrt{n}\|A\|_2, \\ \frac{1}{\sqrt{m}}\|A\|_1 & \leq & \|A\|_2 & \leq \sqrt{n}\|A\|_1, \\ \frac{1}{\sqrt{n}}\|A\|_\infty & \leq & \|A\|_2 & \leq \sqrt{m}\|A\|_\infty, \\ \max\limits_{\substack{1\leq i\leq m\\1\leq j\leq n}}|a_{ij}| & \leq & \|A\|_2 & \leq \sqrt{mn}\max\limits_{\substack{1\leq i\leq m\\1\leq j\leq n}}|a_{ij}|. \end{array}\right\} \tag{B.107}$$

B.9 Quadratic Forms

B.9.0.6 Bilinear and Quadratic Forms

For a matrix A and all conformable column vectors x and y, the functional mapping $(x, y) \to x^TAy$ is called a **bilinear form**. As a function of x and y, it is linear in both x and y and hence **bilinear**. In the case that $x = y$, the functional mapping $x \to x^TAx$ is called a **quadratic form**. The matrix A of a quadratic form is always a square matrix.

B.9.1 Symmetric Decomposition of Quadratic Forms

Any square matrix A can be represented uniquely as the sum of a symmetric matrix and a skew-symmetric matrix:

$$A = \underbrace{\tfrac{1}{2}(A + A^T)}_{\text{symm.}} + \underbrace{\tfrac{1}{2}(A - A^T)}_{\text{skew}-\text{symm.}}, \tag{B.108}$$

where $\frac{1}{2}(A + A^T)$ is called the symmetric part of A and $\frac{1}{2}(A - A^T)$ is called the skew-symmetric part of A. Because

$$x^T Ax = x^T A^T x,$$

the range of a quadratic form with a skew-symmetric matrix is always zero. Therefore, the quadratic form $x^T Ax$ depends only on the symmetric part of A:

$$x^T Ax = x^T \{\tfrac{1}{2}(A + A^T)\}x. \tag{B.109}$$

One can then assume that the matrix of a quadratic form is symmetric, and one can express the quadratic form in summation form as

$$\begin{aligned} x^T Ax &= \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j && \text{(B.110)} \\ &= \sum_{i=j} a_{ij}x_i x_j + \sum_{i\neq j} a_{ij}x_i x_j && \text{(B.111)} \\ &= \sum_{i=1}^{n} a_{ii}x_i^2 + 2\sum_{i>j} a_{ij}x_i x_j && \text{(B.112)} \end{aligned}$$

for symmetric A.

B.9.1.1 Ranges of Quadratic Forms

The set of possible inputs to a function is called its **domain**, and the set of possible outputs is called its **range**.

The domain of a quadratic form for an $n \times n$ matrix A is n-dimensional Euclidean space, and the range is in $(-\infty, +\infty)$, the real line. For $x \neq 0$,

If A is positive definite, the range of $x \to x^T Ax$ is $(0, +\infty)$,

If A is non–negative definite, the range of $x \to x^T Ax$ is $[0, +\infty)$,

If A is indefinite, the range of $x \to x^T Ax$ is $(-\infty, +\infty)$,

If A is non–positive definite, the range of $x \to x^T Ax$ is $(-\infty, 0]$,

If A is negative definite, the range of $x \to x^T Ax$ is $(-\infty, 0)$.

If $x^T x = 1$, then $\lambda_n(A) \leq x^T Ax \leq \lambda_1(A)$. That is, the quadratic form maps the unit n-sphere onto the closed interval $[\lambda_n(A), \lambda_1(A)]$.

B.10 Derivatives of Matrices

B.10.1 Derivatives of Matrix-Valued Functions

The derivative of a matrix with respect to a scalar is the matrix of derivatives of its elements:

$$F(t) = \begin{bmatrix} f_{11}(t) & f_{12}(t) & f_{13}(t) & \cdots & f_{1n}(t) \\ f_{21}(t) & f_{22}(t) & f_{23}(t) & \cdots & f_{2n}(t) \\ f_{31}(t) & f_{32}(t) & f_{33}(t) & \cdots & f_{3n}(t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{m1}(t) & f_{m2}(t) & f_{m3}(t) & \cdots & f_{mn}(t) \end{bmatrix}, \tag{B.113}$$

$$\frac{d}{dt}F(t) = \begin{bmatrix} \frac{d}{dt}f_{11}(t) & \frac{d}{dt}f_{12}(t) & \frac{d}{dt}f_{13}(t) & \cdots & \frac{d}{dt}f_{1n}(t) \\ \frac{d}{dt}f_{21}(t) & \frac{d}{dt}f_{22}(t) & \frac{d}{dt}f_{23}(t) & \cdots & \frac{d}{dt}f_{2n}(t) \\ \frac{d}{dt}f_{31}(t) & \frac{d}{dt}f_{32}(t) & \frac{d}{dt}f_{33}(t) & \cdots & \frac{d}{dt}f_{3n}(t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{d}{dt}f_{m1}(t) & \frac{d}{dt}f_{m2}(t) & \frac{d}{dt}f_{m3}(t) & \cdots & \frac{d}{dt}f_{mn}(t) \end{bmatrix}. \tag{B.114}$$

The rule for the derivative of a product applies also to matrix products:

$$\frac{d}{dt}[A(t)B(t)] = \left[\frac{d}{dt}A(t)\right] B(t) + A(t)\left[\frac{d}{dt}B(t)\right], \tag{B.115}$$

provided that the order of the factors is preserved. If $F(t)$ is square and nonsingular, then $F(t)F^{-1}(t) = I$, a constant. As a consequence, its derivative will be zero. This fact can be used to derive the formula for the derivative of a matrix inverse:

$$\begin{aligned} 0 &= \frac{d}{dt}I && \text{(B.116)}\\ &= \frac{d}{dt}[F(t)F^{-1}(t)] && \text{(B.117)}\\ &= \left[\frac{d}{dt}F(t)\right] F^{-1}(t) + F(t)\left[\frac{d}{dt}F^{-1}(t)\right], && \text{(B.118)}\\ \frac{d}{dt}F^{-1}(t) &= -F^{-1}\left[\frac{d}{dt}F(t)\right] F^{-1}. && \text{(B.119)} \end{aligned}$$

B.10.2 Gradients of Quadratic Forms

If $f(x)$ is a differentiable scalar-valued function of an n-vector x, then the vector

$$\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \cdots, \frac{\partial f}{\partial x_n}\right]^T \tag{B.120}$$

is called the **gradient** of f with respect to x. In the case that f is a quadratic form with symmetric matrix A, then the ith component of its gradient will be

$$\begin{aligned} \left[\frac{\partial}{\partial x}(x^T Ax)\right]_i &= \frac{\partial}{\partial x_i}\left(\sum_j a_{jj}x_j^2 + 2\sum_{j>k} a_{jk}x_j x_k\right) && \text{(B.121)}\\ &= \left(2a_{ii}x_i + 2\sum_{i>k} a_{ik}x_k + 2\sum_{j>i} a_{ji}x_j\right) && \text{(B.122)}\\ &= \left(2a_{ii}x_i + 2\sum_{i\neq k} a_{ik}x_k\right) && \text{(B.123)}\\ &= 2\sum_{k=1}^{n} a_{ik}x_k && \text{(B.124)}\\ &= (2Ax)_i, && \text{(B.125)} \end{aligned}$$

That is, the gradient vector can be expressed as

$$\frac{\partial}{\partial x}(x^T Ax) = 2Ax. \tag{B.126}$$

Bibliography

[1] A. Andrews, "A square root formulation of the Kalman covariance equations," *AIAA Journal*, Vol. 6, pp. 1165–1166, 1968.

[2] H. Anton and C. Rorres, *Elementary Linear Algebra: Applications Version,* 10th Edition, Wiley, 2010.

[3] M. Artin, *Algebra,* Prentice Hall, 1991.

[4] D. Bau III and L. N. Trefethen, *Numerical Linear Algebra,* SIAM, 1997.

[5] R. A. Beauregard and J. B. Fraleigh, *A First Course In Linear Algebra: with Optional Introduction to Groups, Rings, and Fields,* Houghton Mifflin, 1973.

[6] Commandant Benoit, "Sur une méthode de résolution des équations normales provenant de l'application de la méthode des moindes carrés a un système d'équations linéaires en numbre inférieur a celui des inconnues— application de la méthode a la resolution d'un système defini d'équations linéaires (Procédé du Commandant Cholesky)," *Bulletin Géodêsique et Géophysique Internationale*, Vol. 2, Toulouse, pp. 67–77, 1924.

[7] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas,* 2nd Edition, Princeton University Press, 2009.

[8] E. Bodewig, *Matrix Calculus*, North-Holland, 1959.

[9] O. Bretscher, *Linear Algebra with Applications,* 4th Edition, Pearson, 2009.

[10] R. Bronson, *Schaum's Outline of Theory and Problems of Matrix Operations,* McGrawHill, 1998.

[11] M. Brooks, *The Matrix Reference Manual,* (available online at http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html)

[12] W. A. Brown, *Matrices and Vector Spaces,* Marcel Dekker, 1991

[13] W. J. Duncan, "Some devices for the solution of large sets of simultaneous linear equations (with an appendix on the reciprocation of partitioned matrices)," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,* Ser. 7, Vol. 35, pp. 660–670, 1944.

[14] T. E. Fortmann, "A matrix inversion identity," *IEEE Transactions on Automatic Control,* Vol. 20, pp. 801–836, 1978.

[15] J. B. Fraleigh, *A First Course In Abstract Algebra,* 7th Edition, Pearson, 2002.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 4th Edition, Johns Hopkins University Press, 2013.

[17] W. H. Greub, *Graduate Texts in Mathematics Vol. 23: Linear algebra,* Springer-Verlag, 1975.

[18] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB,* Fourth Edition, Wiley, 2015.

[19] L. Guttman, "Enlargement methods for computing the inverse of a matrix," *Annals of Mathematical Statistics,* Vol. 17, pp. 336–343, 1946.

[20] R. A. Horn, and C. R. Johnson, *, Matrix Analysis,* Cambridge University Press, 1985.

[21] W. W. Hager, "Updating the Inverse of a Matrix," *SIAM Review,* Vol. 31, No. 2, pp. 221–239, 1989.

[22] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *SIAM Review,* Vol. 23, No. 1, pp. 53–60, January 1981.

[23] A. S. Householder, *The Theory of Matrices in Numerical Analysis,* Dover, 2006.

[24] T. Kailath, "State-space modelling: Square root algorithms," in *Systems and Control Encyclopedia* (M. G. Singh, Ed.), Pergamon, 1984.

[25] T. Kailath, S.-Y. Kung, and M. Morf, "Displacement ranks of a matrix," *Bulletin of the American Mathematical Society,* (New Series), Vol. 1, No. 5, pp. 703–806, 1979

[26] A. K. Kaw, *Introduction to Matrix Algebra,* 2nd Edition, 2011 (available from http://autarkaw.com/books/matrixalgebra/index.html).

[27] S. Lang, *Introduction to Linear Algebra,* Springer-Verlag, 1987.

[28] Y. L. Luke, *The Special Functions and Their Approximations*, Vol. 2, Academic, New York, 1969.

[29] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra,* SIAM, 2000.

[30] L. Mirsky, *An Introduction to Linear Algebra,* Courier Dover, 1990.

[31] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Review,* Vol. 45, No. 1, pp. 3–49, 2003.

[32] K. Namboordiri, *Matrix Algebra: An Introduction,* Sage Publications, 1984.

[33] E. D. Nering, *Linear Algebra and Matrix Theory,* 2nd Edition, Wiley, 1970.

[34] H. Padé, "Sur la réprésentation approchée d'une fonction par des fractions rationelle," Annals d'écoles Vol. 9s Suppl., 1892.

[35] G. Strang, *Introduction to Linear Algebra,* 4th Edition, SIAM, 2009.

[36] R. C. Ward, "Numerical computation of the matrix exponential with accuracy estimate," *SIAM Journal of Numerical Analysis*, Vol. 14, pp. 600–611, 1977.

[37] R. Zurmühl, *Matrizen*, Academic Press, 1961.