Contents

Appendix A

Linear Vector Spaces and Matrix Computations

Many mathematical objects, e.g., geometric vectors, matrices of some size, real functions etc., can be added together and be multiplied by real or complex numbers (scalars), so that the usual algebraic rules hold. Such objects may be called **vectors**, which is a generalization of the classical usage of this word. A set V of vectors, such that the results of addition and multiplication of real (complex) scalars does not lead outside V is called a **linear vector space**.

In this section we recall basic elements of linear vector spaces and related matrix algebra, and introduce notations to be used in the rest of the text. The exposition is brief and meant as a convenient reference.

A.1 Linear Vector Spaces

A **linear vector space** over $\mathbf{K}$ is a set of **vectors** $\mathbf{V}$, for which the operation addition and scalar multiplication are defined for all vectors in $\mathbf{V}$ and all scalars in the field of real or complex numbers, with the following properties. For all v, $w \in \mathbf{V}$ and all scalars $\alpha, \beta \in \mathbf{R}$ (or $\mathbf{C}$) it holds:

1. addition is commutative and associative and scalar multiplication is associative;

2. distributive properties $\alpha(v+w) = \alpha v + \alpha w$, $(\alpha+\beta)v = \alpha v + \beta v$, for all scalars α, β and v, $w \in \mathbf{V}$;

3. there is an element $0 \in \mathbf{V}$ called the **null vector** such that $v + 0 = v$ for all $v \in \mathbf{V}$;

4. for each vector v there exists a vector $-v$ such that $v + (-v) = 0$;

5. $0 \cdot v = 0$, $1 \cdot v = v$.

Example A.1.1. Familiar examples of a vector space are $\mathbf{V} = \mathbf{R}^n$ ($\mathbf{V} = \mathbf{C}^n$), i.e. the set of n-tuples, $1 \le n < \infty$, of real (complex) numbers. In approximation theory the vector space $\mathcal{P}_n$ of polynomials

$$p_n(x) = \sum_{k=0}^{n-1} a_k x^k,$$

of degree less than n plays an important role. Another example is $\mathbf{V} = \mathbf{C}^p([a,b])$, the set of complex-valued functions which are continuous up to their pth derivatives ($0 \le p < \infty$) on $[a,b]$.

Let $v_1, v_2, \ldots, v_k$ be vectors, and let $\alpha_1, \alpha_2, \ldots, \alpha_k$ be scalars. Then

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_k v_k$$

is called a linear combination of $v_1, v_2, \ldots, v_k$. The vectors are said to be **linearly independent** if none of them is a linear combination of the others, i.e.

$$\sum_{i=1}^{k} \alpha_i v_i = 0, \quad \Rightarrow \quad \alpha_i = 0, \quad i = 1:k.$$

Otherwise, if a nontrivial linear combination of $v_1, \ldots, v_k$ is zero, the vectors are said to be linearly dependent. Then at least one vector v_i will be a linear combination of the rest.

A **basis** in V is any set of linearly independent vectors $v_1, v_2, \ldots, v_n \in V$ such that all vectors $v \in V$ can be expressed as a linear combination

$$v = \sum_{i=1}^{n} \xi_i v_i.$$

The scalars ξ_i are called the components or coordinates of v with respect to the basis $\{v_i\}$. If the vector space $\mathbf{V}$ has a basis of n vectors, then every system of linearly independent vectors of $\mathbf{V}$ has at most k elements and any other basis of $\mathbf{V}$ has the same number k of elements. The number k is called the **dimension** of $\mathbf{V}$ and denoted by $\dim(\mathbf{V})$.

The linear space of column vectors, $x = (x_1, x_2, \ldots, x_n)^T$, where $x_i \in \mathbf{R}$ is denoted $\mathbf{R}^n$; if $x_i \in \mathbf{C}$ then it is denoted $\mathbf{C}^n$. The dimension of this space is n, and the unit vectors $e_1, e_2, ..., e_n$, where

$$e_1 = (1,0,\ldots,0)^T, \quad e_2 = (0,1,\ldots,0)^T, \ldots, e_n = (0,0,\ldots,1)^T,$$

constitute the **standard basis** Note that the coordinates $x_1, x_2, \ldots, x_n$ are the coefficients, when the vector x is expressed as a linear combination of the standard basis. We shall use the same name for a vector as for its coordinate representation by a column vector, with respect to the standard basis.

An arbitrary basis V can be characterized by the *non-singular* matrix $V = (v_1, v_2, \ldots, v_n)$, and the coordinate transformation reads, $x = V\xi$. The standard basis itself is characterized by the unit matrix I.

If $\mathbf{W} \subset \mathbf{V}$ is a vector space then $\mathbf{W}$ is called a **vector subspace** of $\mathbf{V}$. The set of all linear combinations of $v_1, \dots, v_k \in \mathbf{V}$ form a vector subspace denoted by

$$\operatorname{span}\{v_1, \dots, v_k\} = \sum_{i=1}^{k} \alpha_i v_i, \quad \alpha_i \in \mathbf{K}, \quad i = 1 : k.$$

If $\mathbf{S_1}, \dots, \mathbf{S_k}$ are vector subspaces of $\mathbf{V}$ then their sum defined by

$$S = \{v_1 + \cdots + v_k |\ v_i \in \mathbf{S_i},\ i = 1 : k\}$$

is also a vector subspace. The intersection T of a set of vector subspaces is also a subspace,

$$T = \mathbf{S_1} \cap \mathbf{S_2} \cdots \cap \mathbf{S_k}.$$

(The union of vector spaces is generally no vector space.) If the intersection of the subspaces are empty, $\mathbf{S_i} \cap \mathbf{S_j} = 0$, $i \neq j$, then the sum of the subspaces is called their **direct sum** and denoted by

$$\mathbf{S} = \mathbf{S_1} \oplus \mathbf{S_2} \cdots \oplus \mathbf{S_k}.$$

All this may look like a quick repeat of elementary linear algebra. The new thing is that it also applies to linear spaces of **infinite dimension**, i.e. **function spaces**. The elements (vectors) then are functions of one or several real variables on a compact set, i.e. a closed bounded region. The idea of a functions space is now illustrated on an example.

Example A.1.2. Consider the set of functions representable by a convergent power series on the interval $[-1, 1]$,

$$f(t) = c_0 + c_1 t + c_2 t^2 + \cdots.$$

This is an infinite-dimensional linear space. The functions $1, t, t^2, \dots$ can be considered as a standard basis of this space. The coordinates of $f(t)$ then is the vector $c_0, c_1, c_2, \dots$.

A function F from one linear space to another (or the same) linear space is said to **linear** if

$$F(\alpha u + \beta v) = \alpha F(u) + \beta F(v)$$

for all vectors $u, v \in V$ and all scalars α, β. Note that this terminology excludes nonhomogeneous functions like $\alpha u + \beta$, which are sometimes called linear in elementary mathematics. Such functions are called **affine**. A linear function is often expressed in the form Au, where A is called a **linear operator**.

A.2 Matrix and Vector Algebra

A **matrix** A is a collection of $m \times n$ numbers ordered in m rows and n columns

$$A = (a_{ij}) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}.$$

We write $A \in \mathbf{R}^{m\times n}$, where $\mathbf{R}^{m\times n}$ denotes the set of all real $m \times n$ matrices. If $m = n$, then the matrix A is said to be square and of order n. If $m \neq n$, then A is said to be rectangular.

Consider a linear function $u = F(v)$, $v \in \mathbf{C}^n$, $u \in \mathbf{C}^m$, and let x and y be the column vectors representing the vectors v and $F(v)$, respectively, using the standard basis of the two spaces. Then there is a unique matrix $A \in \mathbf{C}^{m\times n}$ representing this map such that

$$y = Ax.$$

This gives a link between linear maps and matrices.

We will follow a convention introduced by Householder[1] and use capital letters (e.g. A, B) to denote matrices. The corresponding lower case letters with subscripts ij then refer to the (i, j) component of the matrix (e.g. a_{ij}, b_{ij}). Greek letters $\alpha, \beta, \ldots$ are usually used to denote scalars. Column vectors are usually denoted by lower case letters (e.g. x, y).

Two matrices in $\mathbf{R}^{m\times n}$ are said to be **equal**, $A = B$, if

$$a_{ij} = b_{ij}, \quad i = 1 : m, \quad j = 1 : n.$$

The basic operations with matrices are defined as follows. The product of a matrix A with a scalar α is

$$B = \alpha A, \qquad b_{ij} = \alpha a_{ij}.$$

The **sum** of two matrices A and B in $\mathbf{R}^{m\times n}$ is

$$C = A + B, \qquad c_{ij} = a_{ij} + b_{ij}. \tag{A.2.1}$$

The **product** of two matrices A and B is defined if and only if the number of columns in A equals the number of rows in B. If $A \in \mathbf{R}^{m\times n}$ and $B \in \mathbf{R}^{n\times p}$ then

$$C = AB \in \mathbf{R}^{m\times p}, \qquad c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}, \tag{A.2.2}$$

and can be computed with mnp multiplications.

Matrix multiplication is *not commutative*. The product BA is defined only if $p = m$. Then the matrices $AB \in \mathbf{R}^{m\times m}$ and $BA \in \mathbf{R}^{n\times n}$ are both square, but if $m \neq n$ of different orders. In general, $AB \neq BA$ even when $m = n$. If $AB = BA$ the matrices are said to **commute**.

Matrix multiplication satisfies the rules

$$A(BC) = (AB)C, \qquad A(B + C) = AB + AC.$$

Note, however, that *the number of arithmetic operations required to compute, respectively, the left- and right-hand sides of these equations can be very different*!

[1] A. S. Householder 1904–1993, mathematician at Oak Ridge National Laboratory and University of Tennessee. He pioneered the use of matrix factorization and orthogonal transformations in numerical linear algebra.

Example A.2.1. If $C \in \mathbf{R}^{p\times q}$ then computing the product ABC as $(AB)C$ requires $mp(n+q)$ operations whereas $A(BC)$ requires $nq(m+p)$ operations. For example, if A and B are square $n \times n$ matrices and x a column vector of length n then computing the product ABx as $(AB)x$ requires $n^3 + n^2$ operations whereas $A(Bx)$ only requires $2n^2$ operations. When $n \gg 1$ this makes a great difference!

The **transpose** A^T of a matrix $A = (a_{ij})$ is the matrix whose rows are the columns of A, i.e., if $C = A^T$ then $c_{ij} = a_{ji}$. For the transpose of a product we have

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

i.e., the product of the transposed matrices in *reverse order.* For a complex matrix A^H denotes the complex conjugate transpose of A

$$A = (a_{ij}), \qquad A^H = (\bar{a}_{ji}),$$

and it holds that $(AB)^H = B^H A^H$.

A **column vector** is a matrix consisting of just one column and we write $x \in \mathbf{R}^m$ instead of $x \in \mathbf{R}^{m\times 1}$. As a special case of the multiplication rule if $A \in \mathbf{R}^{m\times n}$, $x \in \mathbf{R}^n$ then

$$y = Ax \in \mathbf{R}^m, \qquad y_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1 : m.$$

A **row vector** is a matrix consisting of just one row and is obtained by transposing a column vector (e.g. x^T).

The Euclidean **inner product** of two vectors x and y in $\mathbf{R}^n$ is given by

$$x^T y = \sum_{i=1}^{n} x_i y_i = y^T x.$$

In particular

$$x^T x = \sum_{i=1}^{n} |x_i|^2$$

is the Euclidian length of the vector x.

The **outer product** of $x \in \mathbf{R}^m$ and $y \in \mathbf{R}^n$ is the matrix

$$xy^T = \begin{pmatrix} x_1 y_1 & \dots & x_1 y_n \\ \vdots & & \vdots \\ x_m y_1 & \dots & x_m y_n \end{pmatrix} \in \mathbf{R}^{m\times n}.$$

For many problems it often is more relevant and convenient to work with complex vectors and matrices, i.e., the vector space $\mathbf{C}^{n\times m}$ of all complex $n \times m$ matrices whose components are complex numbers.[2]

[2] In MATLAB the only data type used is a matrix with either real or complex elements.

Most concepts introduced here carry over to complex matrices. Addition and multiplication of vectors and matrices follow the same rules as before. The most common inner product of two vectors x and y in $\mathbf{C}^n$ is the Hermitian. It is defined by

$$x^H y = \sum_{k=1}^{n} \bar{x}_k y_k, \tag{A.2.3}$$

where $x^H = (\bar{x}_1, \dots, \bar{x}_n)$ and $\bar{x}_k$ denotes the complex conjugate of x_k. Hence $x^H y = \overline{y^H x}$.

It is useful to define **array operations**, which are carried out element-by-element on vectors and matrices. Following the convention in MATLAB we denote array multiplication and division by $.*$ and $./$, respectively. If A and B have the same dimensions $A \,.* B$ is the matrix with elements equal to $a_{ij} \cdot b_{ij}$ and $A \,./B$ has elements a_{ij}/b_{ij}. (Note that for $+, -$ array operations coincides with matrix operations so no distinction is necessary.)

Any matrix D for which $d_{ij} = 0$ if $i \neq j$ is called a **diagonal matrix**. If $x \in \mathbf{R}^n$ is a vector then $D = \operatorname{diag}(x) \in \mathbf{R}^{n \times n}$ is the diagonal matrix formed by the elements of x. For a matrix $A \in \mathbf{R}^{n \times n}$ the elements a_{ii}, $i = 1 : n$, form the **main diagonal** of A, and we write

$$\operatorname{diag}(A) = \operatorname{diag}(a_{11}, a_{22}, \dots, a_{nn}).$$

For $k = 1 : n - 1$ the elements $a_{i,i+k}$ $(a_{i+k,i})$, $i = 1 : n - k$ form the kth **superdiagonal** (**subdiagonal**) of A. The elements $a_{i,n-i+1}$, $i = 1 : n$ form the (main) **antidiagonal** of A.

The **unit matrix** $I = I_n \in \mathbf{R}^{n \times n}$ is defined by

$$I_n = \operatorname{diag}(1, 1, \dots, 1) = (e_1, e_2, \dots, e_n),$$

and the k-th column of I_n is denoted by e_k. We have that $I_n = (\delta_{ij})$, where δ_{ij} is the **Kronecker symbol** $\delta_{ij} = 0, i \neq j$, and $\delta_{ij} = 1, i = j$. For all square matrices of order n it holds $AI = IA = A$. If desirable, we set the size of the unit matrix as a subscript of I, e.g., I_n.

A matrix A for which all nonzero elements are located in consecutive diagonals is called a **band matrix**. A is said to have **upper bandwidth** r if r is the smallest integer such that

$$a_{ij} = 0, \quad j > i + r,$$

and similarly **lower bandwidth** s if r is the smallest integer such that

$$a_{ij} = 0, \quad i > j + s.$$

The number of nonzero elements in each row of A is then at most equal to $w = r + s + 1$, which is the **bandwidth** of A. For a matrix $A \in \mathbf{R}^{m \times n}$ which is not square we define the bandwidth as

$$w = \max_{1 \le i \le m} \{ j - k + 1 \mid a_{ij} a_{ik} \neq 0 \}.$$

Several classes of band matrices that occur frequently have special names. Thus, a matrix for which $r = s = 1$ is called **tridiagonal**, if $r = 0$, $s = 1$ ($r = 1$, $s = 0$) it is called lower (upper) **bidiagonal** etc. A matrix with $s = 1$ ($r = 1$) is called an upper (lower) **Hessenberg** matrix.

An **upper triangular** matrix is a matrix R for which $r_{ij} = 0$ whenever $i > j$. A square upper triangular matrix has form

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{nn} \end{pmatrix}.$$

If also $r_{ij} = 0$ when $i = j$ then R is **strictly** upper triangular. Similarly a matrix L is **lower triangular** if $l_{ij} = 0, i < j$, and strictly lower triangular if $l_{ij} = 0, i \le j$. Sums, products and inverses of square upper (lower) triangular matrices are again triangular matrices of the same type.

A square matrix A is called **symmetric** if its elements are symmetric about its main diagonal, i.e. $a_{ij} = a_{ji}$, or equivalently $A^T = A$. The product of two symmetric matrices is symmetric if and only if A and B commute, that is, $AB = BA$. If $A^T = -A$, then A is called **skew-symmetric**.

The classical definition of the **determinant**[3] of a matrix requires some elementary facts about permutations, which we now state. Let $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ be a permutation of the integers $\{1, 2, \dots, n\}$. The pair α_r, α_s, $r < s$ is said to form an inversion in the permutation if $\alpha_r > \alpha_s$. For example, in the permutation $\{2, \dots, n, 1\}$ there are $(n-1)$ inversions $(2,1), (3,1), \dots, (n,1)$. A permutation α is said to be even and $\operatorname{sign}(\alpha) = 1$ if it contains an even number of inversions; otherwise the permutation is odd and $\operatorname{sign}(\alpha) = -1$.

A **transposition** τ is a permutation which only interchanges two elements. Any permutation can be decomposed into a sequence of transpositions, but this decomposition is not unique. We now show that a transposition will change the number of inversions by an odd number and thus $\operatorname{sign}(\tau) = -1$. If τ interchanges two adjacent elements α_r and α_{r+1} in the permutation $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$, this will not affect inversions in other elements. Hence the number of inversions increases by 1 if $\alpha_r < \alpha_{r+1}$ and decreases by 1 otherwise. Suppose now that τ interchanges α_r and α_{r+q}. This can be achieved by first successively interchanging α_r with α_{r+1}, then with α_{r+2}, and finally with α_{r+q}. This takes q steps. Next the element α_{r+q} is moved in $q-1$ steps to the position which α_r previously had. In all it takes an *odd number* $2q-1$ of transpositions of adjacent elements, in each of which the sign of the permutation changes.

The determinant of a square matrix A is denoted by $\det(A)$ and defined by

$$\det(A) = \sum_{\alpha \in S_n} \operatorname{sign}(\alpha)\, a_{1,\alpha_1} a_{2,\alpha_2} \cdots a_{n,\alpha_n}, \tag{A.2.4}$$

[3] Determinants were first introduced by Leibniz (1693) and Cayley (1841). The theory of determinants are covered in a monumental five volume work "The Theory of Determinants in the Historical Order of Development" by Thomas Muir (1844–1934).

where the sum is over all permutations of the set $\{1,\dots,n\}$ and sign α is ± 1 according to whether α is an even or odd permutation. (Note that each term in (A.2.4) contains exactly one factor from each row and each column in A.) If $\det(A) \neq 0$ then the matrix A is nonsingular and the solution of the linear system $Ax = b$ can be expressed as

$$x_i = \det(A_j)/\det(A), \quad i = 1:n, \tag{A.2.5}$$

where A_j is the matrix A where the jth column has been replaced by the right hand side b. This expression is known as **Cramer's rule**.[4] Cramer's rule is useful for numerical computation only in very special cases, e.g., if $n = 2$.

Using the definition (A.2.4) to evaluate $\det(A)$ would require $n \cdot n!$ arithmetic operations. By the following three rules $\det(A)$ can be computed much more efficiently:

(i) The value of the determinant is unchanged if a row (column) multiplied by a scalar is added to another row (column).

(ii) The determinant of a triangular matrix equals the product of the elements in the main diagonal, i.e., if R is upper triangular

$$\det(R) = r_{11}r_{22}\cdots r_{nn}.$$

(iii) If two rows (columns) are interchanged the value of the determinant is multiplied by (-1).

Obviously $\det(\alpha A) = \alpha^n \det(A)$. The following rules are also valid:

$$\det(A^T) = \det(A), \qquad \det(AB) = \det(A)\det(B).$$

A matrix is **nonsingular** if and only if $\det(A) \neq 0$. Otherwise the matrix is **singular**. Hence a triangular matrix is nonsingular if and only if all its diagonal elements are nonzero. If A is nonsingular then there exists an **inverse matrix** denoted by A^{-1} with the property that

$$A^{-1}A = AA^{-1} = I.$$

By A^{-T} we will denote the matrix $(A^{-1})^T = (A^T)^{-1}$. For the inverse of a product of two matrices we have

$$(AB)^{-1} = B^{-1}A^{-1},$$

where the product of the inverse matrices are taken in reverse order.

A.3 Partitioning and Block Matrices

A matrix formed by the elements at the intersection of a set of rows and columns of a matrix A is called a **submatrix**. For example, the matrices

$$\begin{pmatrix} a_{22} & a_{24} \\ a_{42} & a_{44} \end{pmatrix}, \qquad \begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix},$$

are submatrices of A. The second submatrix is called a contiguous submatrix since it is formed by contiguous elements of A.

[4] Named after Swiss mathematician Gabriel Cramer 1704–1752.

Definition A.3.1.

A **submatrix** *of $A = (a_{ij}) \in \mathbf{R}^{m\times n}$, is a matrix $B \in \mathbf{R}^{p\times q}$ formed by selecting p rows and q columns of A,*

$$B = \begin{pmatrix} a_{i_1j_1} & a_{i_1j_2} & \cdots & a_{i_1j_q} \\ a_{i_2j_1} & a_{i_2j_2} & \cdots & a_{i_2j_q} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i_pj_1} & a_{i_pj_2} & \cdots & a_{i_pj_q} \end{pmatrix},$$

where

$$1 \le i_1 \le i_2 \le \cdots \le i_p \le m, \quad 1 \le j_1 \le j_2 \le \cdots \le j_q \le n.$$

If $p = q$ and $i_k = j_k$, $k = 1 : p$, then B is a **principal submatrix** *of A. If in addition, $i_k = j_k = k$, $k = 1 : p$, then B is a* **leading** *principal submatrix of A.*

It is often convenient to think of a matrix (vector) as being built up of contiguous submatrices (subvectors) of lower dimensions. This can be achieved by **partitioning** the matrix or vector into blocks. We write, e.g.,

$$A = \begin{array}{r@{}c} & \begin{array}{cccc} q_1 & q_2 & \dots & q_N \end{array} \\ \begin{array}{r} p_1\{ \\ p_2\{ \\ \vdots \\ p_M\{ \end{array} & \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \dots & A_{MN} \end{pmatrix} \end{array}, \qquad x = \begin{array}{r@{}c} \begin{array}{r} p_1\{ \\ p_2\{ \\ \vdots \\ p_M\{ \end{array} & \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} \end{array} \tag{A.3.1}$$

where A_{IJ} is a matrix of dimension $p_I \times q_J$. We call such a matrix a **block matrix**. The partitioning can be carried out in many ways, and is often suggested by the structure of the underlying problem. For square matrices the most important case is when $M = N$, and $p_I = q_I$, $I = 1 : N$. Then the diagonal blocks A_{II}, $I = 1 : N$, are square matrices.

The great convenience of block matrices lies in the fact that the operations of addition and multiplication can be performed by treating the blocks A_{IJ} as *non-commuting scalars* and applying the definitions (**??**) and (**??**). Therefore many algorithms defined for matrices with scalar elements have another simple generalization to partitioned matrices. Of course the dimensions of the blocks must correspond in such a way that the operations can be performed. When this is the case, the matrices are said to be partitioned **conformally**.

The great convenience of block matrices lies in the fact that the operations of addition and multiplication can be performed by treating the blocks A_{ij} as *non-commuting scalars* and applying the definitions (A.2.1) and (A.2.2). Therefore many algorithms defined for matrices with scalar elements have another simple generalization to partitioned matrices. Of course the dimensions of the blocks must correspond in such a way that the operations can be performed. When this is the case, the matrices are said to be partitioned **conformally**. Then we have, e.g.,

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{pmatrix}.$$

Be careful to note the *order* of the factors in the products! In the special case of block upper triangular matrices this reduces to

$$\begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix} = \begin{pmatrix} R_{11}S_{11} & R_{11}S_{12} + R_{12}S_{22} \\ 0 & R_{22}S_{22} \end{pmatrix}.$$

Note that the product is again block upper triangular and its block diagonal equals the products of the diagonal blocks of the factors.

More generally, let $A = (A_{ik})$ and $B = (B_{kj})$ be block matrices of block dimensions $m \times n$ and $n \times p$ respectively, where the partitioning corresponding to the index k is the same for each matrix. Then we have $C = AB = (C_{ij})$, where

$$C_{ij} = \sum_{k=1}^{N} A_{ik}B_{kj}, \quad 1 \le i \le m, \quad 1 \le j \le p.$$

Often it is convenient to partition a matrix into rows or columns. In the special case when $M = 1$ and $A \in \mathbf{R}^{m\times n}$ we write

$$A = (a_1, a_2, \dots, a_n),$$

where $a_j \in \mathbf{R}^m$, $j = 1 : n$, is the j-th column of A. Similarly, when $N = 1$, we write

$$A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix},$$

$a_i \in \mathbf{R}^n$, $i = 1 : m$, which means that a_i^T is the i-th row of A. Let $A \in \mathbf{R}^{m\times n}$, $B \in \mathbf{R}^{n\times p}$. Then the matrix product $C = AB \in \mathbf{R}^{m\times p}$ can be written

$$C = AB = (a_1 \ a_2 \ \cdots \ a_n) \begin{pmatrix} b_1^T \\ b_2^T \\ \vdots \\ b_n^T \end{pmatrix} = \sum_{k=1}^{n} a_k b_k^T, \tag{A.3.2}$$

where $a_k \in \mathbf{R}^m$, $b_k \in \mathbf{R}^p$. Note that each term in the sum of (A.3.2) is an *outer product.*

The more common inner product formula (A.2.2) is obtained from the partitioning

$$C = AB = \begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{pmatrix} (b_1 \ b_2 \ \cdots \ b_p) = (c_{ij}), \qquad c_{ij} = a_i^T b_j.$$

with $a_i, b_j \in \mathbf{R}^n$. Note that when the matrices A and B only have relatively few nonzero elements *the outer product formula* (A.3.2) *is a more efficient way to compute* AB!

A.4 Inner Products, Orthogonality and Projections

An inner product on a vector space $\mathbf{V}$ defined over $\mathbf{K}$ is a continuous mapping $(\cdot,\cdot)$ from $\mathbf{V} \times \mathbf{V}$ onto $\mathbf{K}$ with the properties

1. $(v,v) > 0 \iff v \neq 0;$
2. $(v,w) = \overline{(w,v)};$
3. $(\alpha u + \beta v, w) = \alpha(u,w) + \beta(v,w).$

A vector space for which an inner product is defined is called an **inner product space**. We have already seen examples of an inner product space, namely $\mathbf{R}^n$ $(\mathbf{C}^n)$ with the Euclidian inner product $(x,y) = x^T y$ $((x,y) = x^H y)$.

For a nonsingular linear transformation A which maps a vector space $\mathbf{V}$ onto $\mathbf{V}$ there is a unique **adjoint** transformation A^*, such that

$$(x, A^* y) = (Ax, y).$$

A matrix $A \in \mathbf{C}^{n\times n}$ is called **self-adjoint** if $A^* = A$.

For $A \in \mathbf{R}^{n\times n}$ with the Euclidian inner product we have

$$(Ax, y) = (Ax)^T y = x^T A^T y,$$

that is $A^* = A^T$, the transpose of A. Hence A is self-adjoint if A is symmetric. A symmetric matrix A is called **positive definite** if

$$x^T Ax > 0, \quad \forall x \in \mathbf{R}^n, \quad x \neq 0. \tag{A.4.1}$$

and **positive semidefinite** if $x^T Ax \geq 0$, for all $x \in \mathbf{R}^n$. Otherwise it is called **indefinite**.

Similarly, $A \in \mathbf{C}^{n\times n}$ is self-adjoint or **Hermitian** if $A = A^H$, conjugate transpose of A. A Hermitian matrix has analogous properties to a real symmetric matrix. If A is Hermitian, then $(x^H Ax)^H = x^H Ax$ is real, and A is **positive definite** if

$$x^H Ax > 0, \quad \forall x \in \mathbf{C}^n, \quad x \neq 0,$$

For the vector space $\mathbf{R}^n$ $(\mathbf{C}^n)$ any inner product can be written as $(x,y) = y^T Gx$ $((x,y) = y^H Gx)$, where the matrix G is positive definite.

Any matrix $A \in \mathbf{C}^{n\times n}$ can be written as the sum of its Hermitian and a skew-Hermitian part, $A = H(A) + S(A)$, where

$$H(A) = \frac{1}{2}(A + A^H), \qquad S(A) = \frac{1}{2}(A - A^H).$$

A is Hermitian if and only if $S(A) = 0$. It is easily seen that A is positive definite if and only if its symmetric part $H(A)$ is positive definite.

Two vectors v and w in $\mathbf{R}^n$ are said to be **orthogonal** if $(v,w) = 0$. A set of vectors $v_1, \ldots, v_k$ in $\mathbf{R}^n$ is called orthogonal with respect to the Euclidian inner product if

$$v_i^T v_j = 0, \quad i \neq j,$$

and **orthonormal** if also $v_i^T v_i = 1, i = 1 : k$. An orthogonal set of vectors is linearly independent. More generally, a collection of subspaces $S_1, \dots, S_k$ of $\mathbf{R}^n$ are mutually orthogonal if

$$x^T y = 0, \quad x \in S_i, \quad y \in S_j, \quad i \neq j.$$

The **orthogonal complement** $S^\perp$ of a subspace $S \in \mathbf{R}^n$ is defined by

$$S^\perp = \{y \in \mathbf{R}^n |\ y^T x = 0,\ x \in S\}.$$

The vectors $q_1, \dots, q_k$ form an orthonormal basis for a subspace $S \subset \mathbf{R}^n$ if they are orthonormal and $\operatorname{span}\{q_1, \dots, q_k\} = S$. Such a basis can always be extended to a full orthonormal basis $q_1, \dots, q_n$ for $\mathbf{R}^n$, and then $S^\perp = \operatorname{span}\{q_{k+1}, \dots, q_n\}$.

Let $q_1, \dots, q_n \in \mathbf{R}^m$ be orthonormal and form the matrix $Q = (q_1, \dots, q_n) \in \mathbf{R}^{m \times n}$, $m \geq n$. Then Q is called an **orthogonal matrix** and $Q^T Q = I_n$. If Q also is square $(m = n)$ then we have $Q^{-1} = Q^T$, and hence also $QQ^T = I_n$. Further since $\det(Q^T Q) = \det(I) = 1$ and $\det(Q^T Q) = \det(Q^T)\det(Q) = (\det(Q))^2$ and hence $|\det(Q)| = 1$.

Let S_1 and S_2 be two subspaces such that $S_1 \oplus S_2 = \mathbf{R}^n$ and their intersection is the origin. Then any vector $v \in \mathbf{R}^n$ can be decomposed in a unique way as

$$v = v_1 + v_2, \quad v_1 \in S_1, \quad v_2 \in S_2.$$

The vector v is mapped into v_1 by a linear transformation P_1 called a **projector** onto S_1 along S_2. Since it holds that

$$P_1^2 v = P_1 v, \ \ \forall\ v \in \mathbf{R}^n,$$

we have $P_1^2 = P_1$ and P_1 is called **idempotent**.

A matrix P_1 is a **projector** onto the subspace S_1 if and only if it holds:

$$(i)\ \ P_1 v = v, \ \ \forall \ \ v \in S_1, \qquad (ii)\ \ P_1^2 = P_1. \tag{A.4.2}$$

The decomposition of an arbitrary vector $v \in \mathbf{R}^n$ can be written

$$v = P_1 v + (I - P_1)v = v_1 + v_2, \tag{A.4.3}$$

and $P_2 = I - P_1$ is the projector onto S_2 along S_1.

If it also holds: (iii) $P_1^T = P_1$, then

$$P_1^T P_2 v = P_1^T (I - P_1)v = (P_1 - P_1^2)v = 0, \quad \forall\ v \in \mathbf{R}^n.$$

and it follows that $P_1^T P_2 = 0$. Hence $v^T P_1^T P_2 v = v_1^T v_2 = 0$, for all $b \in \mathbf{R}^n$, that is $v_2 \perp v_1$. In this case P_1 is the **orthogonal projector** onto S_1 and $P_2 = I - P_1$ the orthogonal projector onto $S_1^\perp$. It can be shown that the orthogonal projector P_1 is unique. Orthogonal projections play a central role in the study of least squares problems (see Sec. A.5 and Chapter 8, Volume II).

In the complex case, $A = (a_{ij}) \in \mathbf{C}^{m \times n}$ the Hermitian inner product leads to modifications in the definition of symmetric and orthogonal matrices. Two vectors

x and y in $\mathbf{C}^n$ are called orthogonal if $x^H y = 0$. A square matrix U for which $U^H U = I$ is called **unitary**. From (A.2.3) we find that

$$(Ux)^H Uy = x^H U^H Uy = x^H y.$$

Unitary matrices are characterized by the property that they preserve the Hermitian inner product. In particular the Euclidian length of a vector is invariant under unitary transformations, i.e., $\|Ux\|_2^2 = \|x\|_2^2$. Note that when the vectors and matrices are real the definitions for the complex case are consistent with those made for the real case.

A.5 Linear Least Squares Problems

Four fundamental subspaces are associated with a matrix $A \in \mathbf{R}^{m\times n}$. Two of them are the **range** $\mathcal{R}(A)$ of A and the **null space** $\mathcal{N}(A^T)$ of A^T, which are subspaces of $\mathbf{R}^m$ and defined by

$$\mathcal{R}(A) = \{z \in \mathbf{R}^m |\ z = Ax,\ x \in \mathbf{R}^n\}, \tag{A.5.1}$$

$$\mathcal{N}(A^T) = \{w \in \mathbf{R}^m |\ A^T w = 0\}. \tag{A.5.2}$$

The other two fundamental subspaces are $\mathcal{R}(A^T)$ and $\mathcal{N}(A)$, which are subspaces of $\mathbf{R}^n$.

$$\mathcal{R}(A^T) = \{x \in \mathbf{R}^n |\ x = A^T y,\ y \in \mathbf{R}^m\}, \tag{A.5.3}$$

$$\mathcal{N}(A) = \{y \in \mathbf{R}^n |\ Ay = 0\}. \tag{A.5.4}$$

If $y \in \mathcal{R}(A)$ and $z \in \mathcal{N}(A^T)$ then $y^T z = x^T A^T z = 0$, i.e., y is orthogonal to z. It follows that $\mathcal{N}(A^T)$ *is the orthogonal complement to* $\mathcal{R}(A)$ *in* $\mathbf{R}^m$. Likewise $\mathcal{N}(A)$ *is the orthogonal complement to* $\mathcal{R}(A^T)$ *in* $\mathbf{R}^n$.

The rank r of a matrix A equals the maximum number of independent row or column vectors of A, and thus $r \le \min(m, n)$. If $\operatorname{rank}(A) = n$ we say that A has full **column rank**. If $\operatorname{rank}(A) = m$, then A is said to have full **row rank**. A square matrix $A \in \mathbf{R}^{n\times n}$ is nonsingular if and only if $\mathcal{N}(A) = \{0\}$.

The linear system $Ax = b$, $A \in \mathbf{R}^{m\times n}$ is said to be **consistent** iff $b \in \mathcal{R}(A)$, or equivalently iff $\operatorname{rank}(A,\ b) = \operatorname{rank}(A)$. A consistent linear system always has *at least one solution* x; If $b \notin \mathcal{R}(A)$, or, equivalently, $\operatorname{rank}(A,\ b) > \operatorname{rank}(A)$ the system is **inconsistent** and has no solution. If $m > n$ there are always right hand sides b such that $Ax = b$ is inconsistent.

For an inconsistent linear system $Ax = b$ there are many possible ways of defining a vector x, which in some sense "best" satisfies the system. A choice which can often be motivated for statistical reasons and also leads to a simple computational problem is to take x to be a vector which minimizes the Euclidian length of the **residual vector** $r = b - Ax$

$$\min_x \|b - Ax\|_2, \tag{A.5.5}$$

where we have used the notation

$$\|x\|_2 = (|x_1|^2 + \cdots + |x_n|^2)^{1/2} = (x^T x)^{1/2}.$$

for the Euclidian length of a vector x. We call (A.5.5) a **linear least squares problem** and any minimizer x a **least squares solution** of the system $Ax = b$.

The set of all solutions to problem (A.5.5) can be characterized as follows:

Theorem A.5.1.

The vector x minimizes $\|b - Ax\|_2$ if and only if the residual vector $r = b - Ax$ is orthogonal to $\mathcal{R}(A)$, or equivalently

$$A^T(b - Ax) = 0. \tag{A.5.6}$$

Proof. Let x be a vector for which $A^T(b - Ax) = 0$. Then for any $y \in \mathbf{R}^n$, it holds that $b - Ay = (b - Ax) + A(x - y)$. Squaring this and using (A.5.6) we obtain

$$\|b - Ay\|_2^2 = \|b - Ax\|_2^2 + \|A(x - y)\|_2^2 \geq \|b - Ax\|_2^2,$$

On the other hand assume that $A^T(b - Ax) = z \neq 0$. Then if $x - y = -\epsilon z$ we have for sufficiently small $\epsilon \neq 0$,

$$\begin{aligned}\|b - Ay\|_2^2 &= \|b - Ax\|_2^2 + \epsilon^2 \|Az\|_2^2 - 2\epsilon (Az)^T(b - Ax) \\ &= \|b - Ax\|_2^2 + \epsilon^2 \|Az\|_2^2 - 2\epsilon \|z\|_2^2 < \|b - Ax\|_2^2,\end{aligned}$$

so x does not minimize $\|b - Ax\|_2$. □

Theorem A.5.1 shows that any least squares solution x decomposes the right hand side b into two orthogonal components

$$b = Ax + r, \qquad r \perp Ax. \tag{A.5.7}$$

Here Ax is the orthogonal projection onto $\mathcal{R}(A)$ and $r \in \mathcal{N}(A^T)$; see Fig. 1.6.1. Note that although the least squares solution x may not be unique the decomposition (A.5.7) always is unique.

The above characterization of a least squares solution immediately leads to a classical method for solving the least squares problem (A.5.5). It follows from (A.5.6) that a least squares solution always satisfies the **normal equations**

$$A^T A x = A^T b. \tag{A.5.8}$$

Here $A^T A \in \mathbf{R}^{n \times n}$ is a symmetric, positive semidefinite matrix. The normal equations are always *consistent* since

$$A^T b \in \mathcal{R}(A^T) = \mathcal{R}(A^T A),$$

and therefore a least squares solution always exists. We now give a condition for the least squares solution to be unique.

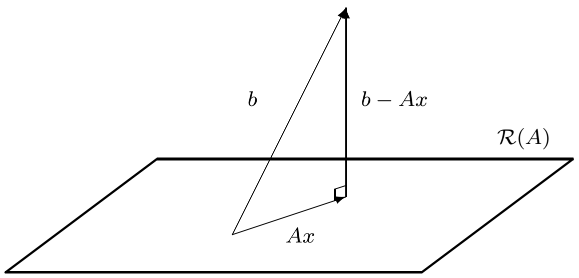

Figure A.5.1. *Geometric characterization of the least squares solution.*

Theorem A.5.2.

The matrix A^TA is positive definite if and only if the columns of A are linearly independent, i.e., when $\operatorname{rank}(A) = n$. *In this case the least squares solution x is unique and given by*

$$x = (A^TA)^{-1}A^Tb. \tag{A.5.9}$$

Proof. If the columns of A are linearly independent, then $x \neq 0 \Rightarrow Ax \neq 0$. Therefore $x \neq 0 \Rightarrow x^TA^TAx = \|Ax\|_2^2 > 0$, and hence A^TA is positive definite. On the other hand, if the columns are linearly dependent, then for some $x_0 \neq 0$ we have $Ax_0 = 0$. Then $x_0^TA^TAx_0 = 0$, and therefore A^TA is not positive definite. When A^TA is positive definite it is also nonsingular and (A.5.9) follows. □

In the full column rank case, $\operatorname{rank}(A) = n$, the residual $r = b - Ax$ can be written

$$r = b - P_{\mathcal{R}(A)}b, \qquad P_{\mathcal{R}(A)} = A(A^TA)^{-1}A^T, \tag{A.5.10}$$

which gives an expression for $P_{\mathcal{R}(A)}$, the orthogonal projector onto $\mathcal{R}(A)$, the range space of A. It follows that any solution to the consistent linear system $Ax = P_{\mathcal{R}(A)}b$ is a least squares solution.

In more general least squares problems $Ax = b$ we can have $\operatorname{rank}(A) < n$, and then A has a nontrivial nullspace. In this case if $\hat{x}$ is any vector that minimizes $\|Ax - b\|_2$, then the set of all least squares solutions is

$$\mathcal{S} = \{x = \hat{x} + y \mid y \in \mathcal{N}(A)\}. \tag{A.5.11}$$

In this set there is a unique solution of minimum norm characterized by $x \perp \mathcal{N}(A)$.

A.6 Eigenvalues of Matrices

Of central importance in the study of matrices are the special vectors whose directions are not changed when multiplied by A. A complex scalar λ such that

$$Ax = \lambda x, \quad x \neq 0, \tag{A.6.1}$$

is called an **eigenvalue** of A and x is an **eigenvector** of A. Eigenvalues and eigenvectors give information about the behavior of evolving systems governed by a matrix or operator and are a standard tools in the mathematical sciences and in scientific computing.

Consider the linear transformation $y = Ax$, where $A \in \mathbf{R}^{n\times n}$. Let V be nonsingular and suppose we change basis by setting $x = V\xi$, $y = V\eta$. Then the column vectors ξ and η represents the vectors x and y with respect to the basis $V = (v_1, \dots, v_n)$. Now $V\eta = AV\xi$, and hence $\eta = V^{-1}AV\xi$, which shows that the matrix

$$B = V^{-1}AV$$

represents the operator A in the new basis V. The mapping $A \to B = V^{-1}AV$ is called a **similarity transformation**. If $Ax = \lambda x$ then

$$V^{-1}AVy = By = \lambda y, \quad y = V^{-1}x,$$

which shows the important fact that B has the same eigenvalues as A. In other words: eigenvalues and eigenvectors are properties of the operator itself, and are independent of the basis used for its representation by a matrix.

From (A.6.1) it follows that λ is an eigenvalue if and only if the linear homogeneous system $(A-\lambda I)x = 0$ has a nontrivial solution $x \neq 0$, or equivalently if and only if $A - \lambda I$ is singular. It follows that the eigenvalues satisfy the **characteristic equation**

$$p(\lambda) = \det(A - \lambda I) = 0. \tag{A.6.2}$$

Obviously, if x is an eigenvector so is αx for any scalar $\alpha \neq 0$.

The polynomial $p(\lambda) = \det(A - \lambda I)$ is the **characteristic polynomial** of the matrix A. Expanding the determinant in (A.6.2) it follows that $p(\lambda)$ has the form

$$p(\lambda) = (a_{11} - \lambda)(a_{22} - \lambda)\cdots(a_{nn} - \lambda) + q(\lambda), \tag{A.6.3}$$

where $q(\lambda)$ has degree at most $n-2$. Hence $p(\lambda)$ is a polynomial of degree n in λ with leading term $(-1)^n\lambda^n$. By the fundamental theorem of algebra the matrix A has exactly$n$ (possibly complex) eigenvalues λ_i, $i = 1, 2, \dots n$, counting multiple roots according to their multiplicities, The set of eigenvalues of A is called the **spectrum** of A and denoted by $\lambda(A)$. The largest modulus of an eigenvalue is called the **spectral radius** and denoted by

$$\rho(A) = \max_i |\lambda_i(A)|. \tag{A.6.4}$$

Putting $\lambda = 0$ in $p(\lambda) = (\lambda_1 - \lambda)(\lambda_2 - \lambda)\cdots(\lambda_n - \lambda)$ and (A.6.2) it follows that

$$p(0) = \lambda_1\lambda_2\cdots\lambda_n = \det(A), \tag{A.6.5}$$

The **trace** of a square matrix of order n is the sum of its diagonal elements

$$\operatorname{trace}(A) = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i. \tag{A.6.6}$$

The last equality follows using the relation between the coefficients and roots of the characteristic equation. Hence the trace of the matrix is invariant under similarity transformations.

Given $A \in \mathbf{C}^{n\times n}$ there exists a unitary matrix $U \in \mathbf{C}^{n\times n}$ such that

$$U^H AU = T = \begin{pmatrix} \lambda_1 & t_{12} & \dots & t_{1n} \\ & \lambda_2 & \dots & t_{2n} \\ & & \ddots & \vdots \\ & & & \lambda_n \end{pmatrix},$$

where T is upper triangular. This is the **Schur normal form** of A. (A proof will be given in Chapter 9, Volume II.) Since

$$\det(T - \lambda I) = (\lambda_1 - \lambda)(\lambda_2 - \lambda) \cdots (\lambda_n - \lambda)$$

the diagonal elements $\lambda_1, \cdots, \lambda_n$ of T are the eigenvalues of A.

To each *distinct* eigenvalue λ_i there is at least one eigenvector w_i. Let $V = (v_1, \dots, v_k)$ be eigenvectors corresponding to the eigenvectors $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_k)$ the eigenvalues of a matrix A. Then,

$$AV = V\Lambda.$$

If there are n linearly independent eigenvectors then $V(v_1, \dots, v_n)$ is nonsingular and

$$A = V\Lambda V^{-1},$$

Then A is said to be **diagonalizable**.

A matrix $A \in \mathbf{C}^{\mathbf{n}\times\mathbf{n}}$ is said to be **normal** if $A^H A = AA^H$. It follows that for a normal matrix the upper triangular matrix T in the Schur normal form is normal, i.e.

$$T^H T = TT^H.$$

It can be shown that this implies that all nondiagonal elements in T vanishes. Hence the matrix T in the Schur normal form for a normal matrix A is diagonal, $T = \Lambda$. Then we have $AU = UT = U\Lambda$, where $\Lambda = \operatorname{diag}(\lambda_i)$, or with $U = (u_1, \dots, u_n)$,

$$Au_i = \lambda_i u_i, \quad i = i : n.$$

This shows the important result that a normal matrix always has a set of mutually unitary (orthogonal) eigenvectors.

Important classes of normal matrices are Hermitian ($A = A^H$), skew-Hermitian ($A^H = -A$), unitary ($A^{-1} = A^H$). Hermitian matrices have real eigenvalues, skew-Hermitian matrices have imaginary eigenvalues, and unitary matrices have eigenvalues on the unit circle; see Chapter 9, Volume II).

An example of a non-diagonalizable matrix is

$$J_m(\lambda) = \begin{pmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{pmatrix} \in \mathbf{C}^{m\times m}.$$

The matrix $J_m(\lambda)$ is called a **Jordan block**. It has one eigenvalue λ of multiplicity m to which corresponds only one eigenvector,

$$J_m(\lambda)e_1 = \lambda e_1, \quad e_1 = (1, 0, \dots, 0)^T.$$

A.7 The Singular Value Decomposition

Let $A \in \mathbf{R}^{m\times n}$ be a matrix of rank r. Then there is a decomposition of A into a product of three matrices

$$A = U\Sigma V^T, \qquad \Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{m\times n}, \tag{A.7.1}$$

where $U \in \mathbf{R}^{m\times m}$ and $V \in \mathbf{R}^{n\times n}$ are orthogonal, $\Sigma_1 = \operatorname{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$, and

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0.$$

(Note that if $r = n$ and/or $r = m$, some of the zero submatrices in Σ disappear.) The σ_i are called the **singular values** of A and if we write

$$U = (u_1, \dots, u_m), \qquad V = (v_1, \dots, v_n),$$

then

$$u_i, \quad i = 1 : m, \qquad v_i, \quad i = 1 : n,$$

$\|u_i\|_2 = 1$, $\|v_i\|_2 = 1$, are left and right **singular vectors**, respectively. The rank of A equals the number of nonzero singular values.

Similarly, for any complex matrix $A \in \mathbf{C}^{m\times n}$ we have the decomposition $A = U\Sigma V^H$, where U and V are unitary matrices and Σ a *real* diagonal matrix. (A proof of the singular value decomposition (SVD) will be given in Sec. 8.3, Volume II.)

The SVD is of great theoretical and practical importance.[5] The geometrical significance of the SVD can be described as follows: The rectangular matrix A represents a mapping from $\mathbf{R}^n$ to $\mathbf{R}^m$. From the SVD it follows that there is an orthogonal basis in each of these two spaces, with respect to which this mapping is represented by the generalized diagonal matrix Σ. Note that transposing (A.7.1) we obtain the SVD of A^T,

$$A^T = V\Sigma^T U^T. \tag{A.7.2}$$

The singular values of A are uniquely determined. For any distinct singular value $\sigma_j \neq \sigma_i$, $i \neq j$, the corresponding singular vector v_j is unique (up to a factor ± 1). For multiple singular values, the corresponding singular vectors can be chosen as any orthonormal basis for the unique subspace that they span. Once the singular vectors v_j, $1 \leq j \leq r$ have been chosen, the vectors u_j, $1 \leq j \leq r$ are uniquely determined, and vice versa, by

$$u_j = \frac{1}{\sigma_j} A v_j, \qquad v_j = \frac{1}{\sigma_j} A^T u_j \quad j = 1 : r. \tag{A.7.3}$$

[5] The SVD was published more than a century ago by Eugenio Beltrami in 1873 and independently by Camille Jordan in 1874. Its use in numerical computations is much more recent.

If U and V are partitioned according to

$$U = (U_1,\ U_2), \quad U_1 \in \mathbf{R}^{m\times r}, \quad V = (V_1,\ V_2), \quad V_1 \in \mathbf{R}^{n\times r}. \tag{A.7.4}$$

then the SVD can be written in the **compact form**

$$A = U_1\Sigma_1 V_1^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T. \tag{A.7.5}$$

The last expression expresses A as a sum of r matrices of rank one.

The **pseudoinverse** of A is defined as

$$A^\dagger = V\Sigma^\dagger U^T, \qquad \Sigma^\dagger = \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbf{R}^{n\times m}, \tag{A.7.6}$$

The pseudoinverse solution of the linear system $Ax = b$ is

$$x = A^\dagger b = V\Sigma^\dagger U^T b$$

and equals the least squares solution of minimum Euclidian length.

The SVD gives complete information about the four fundamental subspaces associated with A. Using (A.7.1)–(A.7.2) it is easy to verify that the range and nullspace of A and A^T are given by

$$\mathcal{R}(A) = \mathcal{R}(U_1) \qquad \mathcal{N}(A^T) = \mathcal{R}(U_2) \tag{A.7.7}$$

$$\mathcal{R}(A^T) = \mathcal{R}(V_1) \qquad \mathcal{N}(A) = \mathcal{R}(V_2). \tag{A.7.8}$$

Hence we immediately find the well-known relations

$$\mathcal{R}(A)^\perp = \mathcal{N}(A^T), \qquad \mathcal{N}(A)^\perp = \mathcal{R}(A^T).$$

In general, we have

$$\dim \mathcal{R}(A) = \dim \mathcal{R}(A^T) = r, \quad \dim \mathcal{N}(A) = n - r, \qquad \dim \mathcal{N}(A^T) = m - r,$$

where $r = \operatorname{rank}(A)$.

If $S = \operatorname{span}(U)$ and $U = (u_1, \dots, u_k)$ is orthogonal, $U^TU = I$, then it is easily seen that the orthogonal projector onto S can be written $P = UU^T$. Similarly the orthogonal projectors onto the four fundamental subspaces of A can be expressed in terms of the singular vectors of A as

$$P_{\mathcal{R}(A)} = AA^\dagger = U_1U_1^T, \qquad P_{\mathcal{N}(A^T)} = U_2U_2^T, \tag{A.7.9}$$

$$P_{\mathcal{R}(A^T)} = A^T(A^T)^\dagger = V_1V_1^T, \qquad P_{\mathcal{N}(A)} = V_2V_2^T.$$

A.8 Norms of Vectors and Matrices

In many applications it is useful to have a measure of the size of a vector or a matrix. An example is the quantitative discussion of errors in matrix computation. Such measures are provided by vector and matrix norms, which can be regarded as generalizations of the absolute value function on $\mathbf{R}$.

A **norm** on a vector space $\mathbf{V} \in \mathbf{C}^n$ is a function $\mathbf{V} \to \mathbf{R}$ denoted by $\|\cdot\|$ that satisfies the following three conditions:

1. $\|x\| > 0, \quad \forall x \in \mathbf{V}, \quad x \neq 0$ (definiteness)
2. $\|\alpha x\| = |\alpha|\,\|x\|, \quad \forall \alpha \in \mathbf{C}, \quad x \in \mathbf{C}^n$ (homogeneity)
3. $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in \mathbf{V}$ (triangle inequality)

The triangle inequality is often used in the form (see Problem 11) $\|x \pm y\| \geq \big|\,\|x\| - \|y\|\,\big|$.

The most common vector norms are special cases of the family of **Hölder** norms or p-norms

$$\|x\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}, \qquad 1 \leq p < \infty. \tag{A.8.1}$$

The p-norms have the property that $\|x\|_p = \|\,|x|\,\|_p$. Vector norms with this property are said to be **absolute**. The three most important particular cases are $p = 1, 2$ and the limit when $p \to \infty$:

$$\begin{aligned} \|x\|_1 &= |x_1| + \cdots + |x_n|, \\ \|x\|_2 &= (|x_1|^2 + \cdots + |x_n|^2)^{1/2} = (x^H x)^{1/2}, \\ \|x\|_\infty &= \max_{1 \leq i \leq n} |x_i|. \end{aligned} \tag{A.8.2}$$

The vector 2-norm is also called the Euclidean norm. It is invariant under unitary (orthogonal) transformations since

$$\|Qx\|_2^2 = x^H Q^H Q x = x^H x = \|x\|_2^2$$

if Q is orthogonal.

Another important property of the p-norms is the **Hölder inequality**

$$|x^H y| \leq \|x\|_p \|y\|_q, \quad \frac{1}{p} + \frac{1}{q} = 1, \quad p \geq 1. \tag{A.8.3}$$

For $p = q = 2$ this becomes the **Cauchy–Schwarz inequality**

$$|x^H y| \leq \|x\|_2 \|y\|_2.$$

Norms can be obtained from inner products by taking

$$\|x\|^2 = (x, x) = x^H G x,$$

where G is Hermitian and positive definite. It can be shown that the unit ball $\{x : \|x\| \leq 1\}$ corresponding to this norm is an ellipsoid, and hence they are also called elliptic norms. A special case that frequently is useful is the **scaled** p-norms defined by

$$\|x\|_{p,D} = \|Dx\|_p, \quad D = \operatorname{diag}(d_1, \ldots, d_n), \quad d_i \neq 0, \quad i = 1 : n. \tag{A.8.4}$$

All norms on $\mathbf{C}^n$ are equivalent in the following sense: For each pair of norms $\|\cdot\|$ and $\|\cdot\|'$ there are positive constants c and c' such that

$$\frac{1}{c}\|x\|' \leq \|x\| \leq c'\|x\|', \quad \forall x \in \mathbf{C}^n. \tag{A.8.5}$$

In particular it can be shown that for the p-norms we have

$$\|x\|_q \le \|x\|_p \le n^{\left(\frac{1}{p}-\frac{1}{q}\right)}\|x\|_q, \quad 1 \le p \le q \le \infty. \tag{A.8.6}$$

We now consider **matrix norms**. We can construct a matrix norm from a vector norm by defining

$$\|A\| = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\|. \tag{A.8.7}$$

This norm is called the **operator norm**, or the matrix norm **subordinate** to the vector norm. From the definition it follows directly that

$$\|Ax\| \le \|A\|\ \|x\|, \qquad x \in \mathbf{C}^n.$$

Whenever this inequality holds, we say that the matrix norm is **consistent** with the vector norm.

It is an easy exercise to show that operator norms are **submultiplicative**, i.e., whenever the product AB is defined it satisfies the condition

4. $N(AB) \le N(A)N(B)$

The matrix norms

$$\|A\|_p = \sup_{\|x\|=1} \|Ax\|_p, \quad p = 1, 2, \infty,$$

subordinate to the vector p-norms are especially important. For these it holds that $\|I_n\|_p = 1$. The 1-norm and ∞-norm are easily computable from

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}|, \qquad \|A\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}|, \tag{A.8.8}$$

respectively. Note that the 1-norm equals the maximal column sum and the ∞-norm equals the maximal row sum of the magnitude of the elements. Consequently $\|A\|_1 = \|A^H\|_\infty$.

The 2-norm is also called the **spectral norm**. Its major drawback is that it is expensive to compute. We have

$$\|A\|_2 = \sup_{\|x\|=1} (x^H A^H A x)^{1/2} = \sigma_1(A), \tag{A.8.9}$$

where $\sigma_1(A)$ is the largest singular value of A. Since the nonzero eigenvalues of $A^H A$ and AA^H are the same it follows that $\|A\|_2 = \|A^H\|_2$. A useful upper bound for the matrix 2-norm is

$$\|A\|_2 \le (\|A\|_1 \|A\|_\infty)^{1/2}. \tag{A.8.10}$$

The proof of this bound is given as an exercise in Problem 16.

Another way to proceed in defining norms for matrices is to regard $\mathbf{C}^{m\times n}$ as an mn-dimensional vector space and apply a vector norm over that space. With the exception of the **Frobenius norm** [6] derived from the vector 2-norm

$$\|A\|_F = \Big(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\Big)^{1/2} \tag{A.8.11}$$

such norms are not much used. Note that $\|A^H\|_F = \|A\|_F$. Useful alternative characterizations of the Frobenius norm are

$$\|A\|_F^2 = \operatorname{trace}(A^HA) = \sum_{i=1}^{k}\sigma_i^2(A), \quad k=\min(m,n), \tag{A.8.12}$$

where $\sigma_i(A)$ are the nonzero singular values of A. The Frobenius norm is submultiplicative. However, it is often larger than necessary; e.g., $\|I_n\|_F = n^{1/2}$. This tends to make bounds derived in terms of the Frobenius norm not as sharp as they might be. From (A.8.9) and (A.8.12) we also get lower and upper bounds for the matrix 2-norm

$$\frac{1}{\sqrt{k}}\|A\|_F \le \|A\|_2 \le \|A\|_F, \quad k=\min(m,n).$$

An important property of the Frobenius norm and the 2-norm is that both are invariant with respect to orthogonal transformations.

Lemma A.8.1. *For all orthogonal matrices Q and P ($Q^HQ=I$, and $P^HP=I$) of appropriate dimensions it holds*

$$\|QAP\| = \|A\| \tag{A.8.13}$$

where $\|\cdot\|$ is the Frobenius norm and the 2-norm.

We finally remark that the 1-,∞- and the Frobenius norm satisfy

$$\|\,|A|\,\| = \|A\|, \qquad |A| = (|a_{ij}|),$$

but for the 2-norm the best result is that $\|\,|A|\,\|_2 \le n^{1/2}\|A\|_2$. The vector and matrix norms defined in this section can immediately be extended to complex vectors and matrices.

One use of norms is the study of *limits of sequences of vectors and matrices* (see Sec. 9.2.4). Consider an infinite sequence $x_1, x_2, \dots$ of elements of a vector space $\mathbf{V}$ and let $\|\cdot\|$ be a norm on $\mathbf{V}$. The sequence is said to converge (strongly if V is infinite dimensional) to a limit $x\in\mathbf{V}$, and we write $\lim_{k\to\infty} x_k = x$ if

$$\lim_{k\to\infty}\|x_k - x\| = 0,$$

[6] Ferdinand George Frobenius (1849–1917) German mathematician, professor at ETH Zürich (1875–1892) before he succeeded Weierstrass at Berlin University.

For a finite dimensional vector space the equivalence of norms (A.8.5) shows that convergence is independent of the choice of norm. The particular choice $\|\cdot\|_\infty$ shows that convergence of vectors in $\mathbf{C}^n$ is equivalent to convergence of the n sequences of scalars formed by the components of the vectors. By considering matrices in $\mathbf{C}^{m\times n}$ as vectors in $\mathbf{C}^{mn}$ the same conclusion holds for matrices.

Review Questions

1. Define the concepts:

(i) Real symmetric matrix. (ii) Real orthogonal matrix.

(iii) Real skew-symmetric matrix. (iv) Triangular matrix.

(v) Hessenberg matrix.

2 To compute the matrix product $C = AB \in \mathbf{R}^{m\times p}$ we can either use an outer product or an inner product formulation. Discuss the merits of the two resulting algorithms when A and B have relatively few nonzero elements.

3. (a) Give conditions for a matrix P to be the orthogonal projector onto a subspace $S \in \mathbf{R}^n$.

(b) Define the orthogonal complement of S in $\mathbf{R}^n$.

4. What is the Schur normal form of a matrix $A \in \mathbf{C}^{n\times n}$?

(b)What is meant by a normal matrix? How does the Schur form simplify for a normal matrix?

5. (a) Show that $A^\dagger = A^{-1}$ when A is a nonsingular matrix.

(b) Construct an example where $G \neq A^\dagger$ despite the fact that $GA = I$.

6. (a) Construct an example where $(AB)^\dagger \neq B^\dagger A^\dagger$.

(b) Show that if A is an $m \times r$ matrix, B is an $r \times n$ matrix, and $\operatorname{rank}(A) = \operatorname{rank}(B) = r$, then $(AB)^\dagger = B^\dagger A^\dagger$.

7. Show, using the SVD, that $P_{\mathcal{R}(A)} = AA^\dagger$ and $P_{\mathcal{R}(A^T)} = A^\dagger A$.

8. Define the matrix subordinate norm to a given vector norm.

9. Define the p norm of a vector x. Give explicit expressions for the matrix p norms for $p = 1, 2, \infty$. Show that

$$\frac{1}{n}\|x\|_1 \leq \frac{1}{\sqrt{n}}\|x\|_2 \leq \|x\|_\infty.$$

which are special cases of (A.8.6).

Problems

1. (a) A square matrix A is called **persymmetric** if it is symmetric about its antidiagonal, i.e., $a_{ij} = a_{n-j+1,n-i+1}$. Show that A is persymmetric if and only if PA is symmetric, where P is the permutation matrix that reverses the rows of A.

(b) Show that if $A, B \in \mathbf{R}^{n\times n}$ are both symmetric and persymmetric, then the matrix $AB + BA$ also has this property.

2. Let $A \in \mathbf{R}^{m\times n}$ have rows a_i^T, i.e., $A^T = (a_1, \dots, a_m)$. Show that

$$A^TA = \sum_{i=1}^{m} a_i a_i^T.$$

What is the corresponding expression for A^TA if A is instead partitioned into columns?

3. (a) If A and B are square upper triangular matrices show that AB is upper triangular, and that A^{-1} is upper triangular if it exists. Is the same true for lower triangular matrices?

(b) Let $A, B \in \mathbf{R}^{n\times n}$ have lower bandwidth r and s respectively. Show that the product AB has lower bandwidth $r + s$.

(c) An upper Hessenberg matrix H is a matrix with lower bandwidth $r = 1$. Using the result in (a) deduce that the product of H and an upper triangular matrix is again an upper Hessenberg matrix.

(d) Show that if $R \in \mathbf{R}^{n\times n}$ is strictly upper triangular, then $R^n = 0$.

4. To solve a linear system $Ax = b$, where $A \in \mathbf{R}^n$, by Cramer's rule (see Equation (A.2.5)) requires the evaluation of $n + 1$ determinants of order n. Estimate the number of multiplications needed for $n = 50$ if the determinants are evaluated in the naive way. Estimate the time it will take on a computer performing 10^9 floating point operations per second!

5. Consider an upper block triangular matrix

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

and suppose that R_{11}^{-1} and R_{22}^{-1} exists. Show that R^{-1} exists.

6 (a) Show that if $w \in \mathbf{R}^n$ and $w^Tw = 1$, then the matrix $P(w) = I - 2ww^T$ is both symmetric and orthogonal.

(b) Given two vectors $x, y \in \mathbf{R}^n$, $x \neq y$, $\|x\|_2 = \|y\|_2$, then

$$P(w)x = y, \qquad w = (y - x)/\|y - x\|_2.$$

7. Let $A \in \mathbf{R}^{n\times n}$ be a given matrix. Show that if $Ax = y$ has *at least one* solution for any $y \in \mathbf{R}^n$, then it has *exactly one* solution for any $y \in \mathbf{R}^n$. (This is a useful formulation for showing uniqueness of approximation formulas.)

8. Show that for $x \in \mathbf{R}^n$,

$$\lim_{p\to\infty} \|x\|_p = \max_{1\le i\le n} |x_i|.$$

9. Prove that the following inequalities are valid and best possible:

$$\|x\|_2 \le \|x\|_1 \le n^{1/2}\|x\|_2, \qquad \|x\|_\infty \le \|x\|_1 \le n\|x\|_\infty.$$

Derive similar inequalities for the comparison of the operator norms $\|A\|_1$, $\|A\|_2$, and $\|A\|_\infty$.

10. Show that any vector norm is uniformly continuous by proving the inequality

$$|\,\|x\| - \|y\|\,| \le \|x - y\|, \qquad x, y \in \mathbf{R}^n.$$

11. Show that for any matrix norm there exists a consistent vector norm.
Hint: Take $\|x\| = \|xy^T\|$ for any vector $y \in \mathbf{R}^n$, $y \neq 0$.

12. Derive the formula for $\|A\|_\infty$ given in (A.8.8).

13. Show that for any subordinate matrix norm

$$\|A + B\| \le \|A\| + \|B\|, \qquad \|AB\| \le \|A\|\|B\|.$$

14. Show that $\|A\|_2 = \|PAQ\|_2$ if P and Q are orthogonal matrices.

15. Use the result $\|A\|_2^2 = \rho(A^TA) \le \|A^TA\|$, valid for any matrix operator norm $\|\cdot\|$, where $\rho(A^TA)$ denotes the spectral radius of A^TA, to deduce the upper bound in (A.8.10).

16. (a) Let T be a nonsingular matrix, and let $\|\cdot\|$ be a given vector norm. Show that the function $N(x) = \|Tx\|$ is a vector norm.

(b) What is the matrix norm subordinate to $N(x)$?

(c) If $N(x) = \max_i |k_i x_i|$, what is the subordinate matrix norm?

Appendix B

A Multiple Precision Package

B.1 Introduction

In the following we describe the basics of **Mulprec**, a collection of Matlab m-files for, in principle, unlimited multiple precision floating point computation. and give examples of its use.

The version of Mulprec described here was worked out by the first author during April–December 2001. It is a preliminary version and there may still exist bugs. Originally, a shorter version of this package and text was meant as a start impulse for a Master's project at the Royal Institute of Technology (KTH), Stockholm. Some new ideas about chopping strategies and error estimation and control have been applied in some of the m-files for the basic operations and elementary functions.

A **normalized mulprec number** is a row vector x with the usual Matlab notations; the value of x reads

$$\mathrm{val}(x) = P^{x(1)} \sum_{j=2:k} x(j)P^{k-j}; \quad P = 10^7, \quad k \geq 2.$$

The $x(j)$, $j > 1$, are integers called **gytes** (or *gits*), i.e., giant digits. They should all have the same sign (equal to $\mathrm{sgn}(x)$), and $|x(j)| < P$, $j = 2 : k$, and $x(2) \neq 0$. So, we have a position system with base $P = 10^7$. $x(1)$ is the exponent of $\mathrm{val}(x)$, in a floating point representation with base P. Please note that $P^{x(1)}$ denotes the unit of the *least significant* gyte, contrary to the traditional floating point convention.[1] The length k of a mulprec number x may vary during a computation.

Example B.1.1.

π equals, with an absolute error less than $P^{-10} = 10^{-70}$, the following 12 gyte number:

[1] It seems to be rather easy to change this if desirable.

Columns 1 through 6

```
-10   3   1415926   5358979   3238462   6433832
```

Columns 7 through 12

```
7950288   4197169   3993751    582097   4944592   3078164
```

The decimal point, or rather the gyte point, is located immediately after column $(12 - 10) = 2$.

$$1 = [0, 1], \qquad 0.5 = [-1, 5000000], \qquad -0.125 = [-1, -1250000].$$

We call the Matlab numbers **floats**. You rarely have to write the mulprec form of numbers that are *exactly representable as floats.* The commands for the elementary operations and most functions are namely so constructed that they *accept single floats (not expressions) as input data* and convert them to normalized mulprec numbers by means of the command *npr*, or *npc* for complex floats, see below.[2] Mulprec distinguishes between floats and mulprec numbers by the length, which is equal to 1 or larger than 1, respectively.

For a **complex**, normalized mulprec number these conditions typically hold for both the real and the imaginary part. The exponent and the length are common for both parts; an exception: $x(2)$ may thus be zero for one of the parts.

It is fundamental for Mulprec that P can be squared without overflow with some margin. In fact, $2^{53} > 90P^2$. Hence, if the shorter one of two positive normalized mulprec numbers has at most 90 gytes, we can obtain their product by the multiplication of gytes and addition of integers, so that the sums do not exceed 2^{53}. Typically, there is only one normalization in a multiplication.

The normalized representation of x is *unique* (if $x \neq 0$). For example, note that, if you subtract two positive normalized mulprec numbers, the gytes of the result may have varying signs, unless you **normalize** the result by the mulprec operation *nize* (or the simpler operation *rnize* if the number is real). Since the operation *rnize* is not fast compared to the operations *add* and *sub*, there is as a rule no normalization in *add* and *sub*.

For such reasons we now introduce a more general concept: the **legal mulprec number**; val(x) has the same value and the same form as the normalized mulprec number, but all the $x(j)$ need not have the same sign and they have a looser bound: $|x(j)| < 45\,P^2$.[3] Evidently such a representation of a number is not unique.

Allowing this more general type of mulprec number in additions and subtractions, makes it *unnecessary to transport carry digits inside these operations*; this is typically done later, if a normalization is needed.[4] A typical suboperation of the

[2] Expressions with mulprec operations are, however, allowed as input data.

[3] The addition of two legal numbers does not cause overflow, but the sum can be illegal at first and must be immediately normalized, see the next footnote.

[4] An exception: if the result of an *add* or a *sub* has become illegal, then it becomes acceptable after an automatic call of nize inside *add.m* (or *sub.m*).

normalization is to subtract a multiple cP from one of the $x(j)$; this is typically compensated for by adding c to $x(j-1)$, in order to keep val(x) constant. (Is not that how we learned to handle the carry in addition in elementary school?)

Multiplication, division, elementary functions etc., do include normalization, both of the operands and of the results. Normalized numbers only should be printed.

B.2 The Mulprec Library

The mfiles for about 60 mulprec functions are packed together in the text file mulprec.lib, which can be downloaded from the books homepage. The numbers in the beginning of the lines of the

The mfiles for the following mulprec functions are packed together in the text file mulprec.lib, which can be downloaded from the books homepage. The numbers in the beginning of the lines of the lists below are only for making references in the text more convenient. They are thus *not* to be used in the codes and your commands.

Since the condensed comments in the table below, may be unclear, you are recommended to study the codes a little before you use the system. x, y, z are typically mulprec numbers. As mentioned above, most of the commands accept also floats as input if it makes sense.

In a command like $z = mul(x, y, s)$, the parameter s means the number of gytes wanted in the result (including the exponent; hence it equals the length in the Matlab sense). It is optional; if s is omitted the *exact* product is computed and normalized (not chopped).

An asterisk means that the code is longer than 500 bytes. The absence of an asterisk usually indicates, e.g., that the code is a relatively short combination of other library codes. The number in the beginning of the lines of the following table are not used in the computations; they are just for easy reference to the table and to mulprec.lib.

B.2.1 Basic arithmetic operations

Addition, subtraction were commented above. Multiplication is performed as in elementary school—the amount of work is approximately proportional to the product of the sizes of the factors. Perhaps one of the fast algorithms presented in Knuth [**?**, Sec. 4.3.3], in the binary case, will be adapted to the gyte system in the future.

In the table below mul.m, the shorter of the operands is chosen to be the multiplier. In order to avoid overflow (in the additions inside the multiplication), the multiplier is chopped to 90 gytes (at most 623 decimal places). An operation that can handle a multiplier by partitioning it into 90-gyte pieces and calling mul.m once for each piece, is tentatively called mullong.m. It has not yet been implemented. *At present* there are bounds also for the accuracy for division, square root, elementary functions etc., since multiplication is used in their codes.

1/x and 1/sqrt(x) are implemented by Newton's iteration method, (with variable precision) that (roughly) doubles the number of gytes in each iteration. The initial approximation is obtained by the ordinary Matlab operations (giving approximately 16 correct decimals). See more details in mulprec.lib. The square root algorithm is division-free.

At present, some limitations of Mulprec are set by the restriction of the length of the shorter operand of a multiplication to at most 90 gytes). It does not seem to be very difficult to remove these bounds, or at least to widen them considerably.

1.*	add.m	$z = add(x, y)$	$z = x + y$
2a*.	sub.m	$z = sub(x, y)$	$z = x - y$
2b.	subb.m	$z = subb(x, y)$	$z = x + mi(y)$, shorter but slower than 2a
3*.	mul.m	$z = mul(x, y, s)$	$z = x \cdot y$, s optional, see above
3b.	mullong.m	$z = mul(x, y, s)$	$z = x \cdot y$. Unrestricted multiplication. Not yet implemented.
4*.	recip.m	$z = recip(x, s)$	$z = 1/x$
5.	div.m	$z = div(x, y, s)$	$z = x/y$
6a.	mi.m	$z = mi(x)$	$z = -x$; all components change sign *except* the exponent
6b.	muabs.m	$z = \|x\|$	Absolute value of a real or complex mulprec number Not yet implemented.
7*.	musqrt.m	$[y, iny] = musqrt(x, s)$	Returns sqrt(x) and optionally 1/sqrt(x)

B.2.2 Some special mulprec operations

The operation chop.m is more general than just chopping to a desired length. See the code in mulprec.lib.The normalization code *rnize* still has a bug (?) that violates the uniqueness. It can happen, e.g., that the last two gytes of a positive number read −1 9999634 (say). Such nine-sequences may also occur at other places in the vector. Sometimes such a representation is more easily interpreted than a strictly normalized normalized mulprec number. I have therefore not yet tried to eliminate this "bug".

8.	npr.m	$xx = npr(x)$	Converts real float to normalized mulprec number
9.	npc.m	$xx = npc(x)$	Converts complex float to normalized mulprec number
10.	flo.m	$y = flo(x)$	Approximates mulprec number by float
11*.	chop.m	$y = chop(x, k)$	Returns approximately equivalent mulprec number, length k
12*.	rnize.m	$y = rnize(x)$	Normalizes real mulprec number
13.	nize.m	$y = nize(x)$	Normalizes complex mulprec number
14.	elizer.m	$y = elizer(x)$	Eliminates zero gytes in mulprec number, left and right.
15.	muzero.m	$y = muzero(x)$	If $x == 0$, $y = 1$, else $y = 0$.

B.2.3 Elementary functions

In the computation of e^x, x real, we first seek $\bar{x}$ and an integer n, such that

$$e^x = e^{\bar{x}} P^n, \quad \text{and} \quad |\bar{x}| < \tfrac{1}{2} \ln P.$$

Then, for an appropriate integer m, $e^{\bar{x}/2^m}$ is computed by the $k-1$-term Maclaurin expansion, a Horner scheme with variable precision. $e^{\bar{x}}$ is then obtained by squaring the sum of the Maclaurin expansion m times. Suppose that the volume of computation is proportional to $m+ck$; the value $c = 0.4$ has been found by a combination of heuristic theory and experiment. In the code, the parameters m and k are obtained from an approximate formula for finding the minimum of $m+ck$ with the constraint that the bound for the relative error of $e^{\bar{x}/2^m}$, due to the Maclaurin truncation and the squarings of the Maclaurin sum does not exceed P^{-s}.

A similar idea is applied for e^{ix}. Now $\bar{x} \in [-8\pi, 8\pi]$, and $k-1$ terms of the Taylor expansion into powers of $x/2^m$ are used. These methods are inspired from ideas developed by Napier and Briggs, when they computed the first tables of logarithms. See Goldstine [12].

The algorithms in lnr.m and muat2.m are based on Newton's method for the equations $e^y = x$ and $\tan y = x$, respectively, with initial approximations from the Matlab operations $\ln x$ and atan2(y, x). The commands muat2, lnc and mulog do not yet allow floats as input, and the codes are not well tested.

16*.	expo.m	$y = expo(x, s)$	$y = e^x$, x real,
17*.	expi.m	$[cox, six, eix] = expi(x, s)$	$\cos x$ and optionally $\sin x$, e^{ix}, x real
18.	muexp.m	$w = muexp(z, s)$	$y = e^z$, z complex
19*.	lnr.m	$y = lnr(x, s)$	$x > 0$
20*.	muat2.m	$v = muat2(y, x, s)$	adapted from atan2(y, x); not yet with float input
21a.	lnc.m	$w = lnc(z, s)$	$w = ln\ z,\ z \neq 0$; not yet with float input
21b.	mulog.m	$w = mulog(z, s)$	A better(?) name for 21a

B.2.4 A library for mulprec vector algorithms

A **mulprec column vector** is represented by a (Matlab) rectangular matrix. A **mulprec row vector** is a row of mulprec numbers (where each mulprec number is a row of gytes). In a **rectangular mulprec matrix**, each column is a mulprec column vector, and each row is a mulprec row vector. So, we can say that a mulprec matrix is a row of rectangular (Matlab) matrices, all of the same size. The following set of operations is very preliminary. It was worked out for an application to repeated Richardson h^2 extrapolation, see the m-file rich3.m.

30*.	fixcom.m	$y = fixcom(x, a)$	x, y mulprec vectors. Returns $y \approx x$, $y(1) = a(1)$, length($y(i)$) =length(a)
31*.	musv.m	$y = musv(sca, vec)$,	sca is mulprec scalar, vec is mulprec vector $y = sca \cdot vec$
32*.	scalp.m	$y = scalp(vec1, vec2)$,	scalar prod. in n-dim Eucl. space, vec_1, vec_2 mulprec column vectors
33.	adv.m	$z = adv(x, y)$	$z = x + y$; x, y, z mulprec vectors
34.	rnizev.m	$y = rnizev(x)$	Normalizes real mulprec vector
35.	chopv.m	$y = chopv(x, s)$	Chops components of mulprec vector to length s
36.	chonizv.m	$y = chonizv(x, s)$	Normalizes and chops a mulprec vector

B.2.5 Miscellaneous

50*.	intro.m	Starting routine for Mulprec. See below.
51*.	rich3.m	Mulprec algorithm for repeated Richardson h^2 extr.
52*.	polygons.m	Compute circumference for a sequence of polygons. Calls rich3.m
53.	why.m	

There are also *edited diaries* of a few test experiments (comparisons of computations with different precision), e.g., pippi2.dia (π computed by polygons.m and rich3.m), muat2est.dia ($\pi = 4 \arctan 1$, etest.dia (e computed by expo.m).

B.2.6 How to start Mulprec

Change directory to the seat of the Mulprec files.
Run intro.m. (*If you forget this*, you are likely to obtain confusing error messages. Ignore them and run intro.m!)
Then intro.m brings down the file *const.mat* from the disk. The file const.mat contains, e.g., 50 gytes mulprec approximations to π (called pilong), and to $\ln P$ (called LP), and the default values of some other global variables. Now Matlab is ready for your Mulprec adventures.

B.3 More Subprojects

A mulprec analog to the matlab command rat for finding accurate (or exact) rational approximations to floating point results. In connection with this the basic operations of exact rational arithmetic and continued fractions, including gcd and lcm. (See Knuth [**?**, vol.II, sec. 4.5.2], in particular p. 327). Mulprec can, of course, not compete with Maple and similar systems for rational arithmetic. Minor tasks of this type may, however, appear in a context where Mulprec is used.

Interesting specific examples: difference schemes, the generalized Euler Transformation, the Euler–Maclaurin Formula, and other methods of convergence acceleration. Illconditioned power series, transformation of a moment sequence to the three-term recurrence coefficients for the orthogonal polynomials to the same weight distribution or, equivalently, transformation of a power series to a continued fraction. Gaussian elimination, Gram–Schmidt orthogonalization.

Theoretical analysis, if possible applied to built-in error estimation and control, e.g., chopping strategies for the construction of the m-files, both for the Mulprec library, and for suggestions to the Mulprec users.

Documentation, both comments in the codes, a detailed report, and (in particular) a clear, short and attractive booklet with a user's manual.

At present, some limitations of Mulprec are set by the restriction of the length of the shorter operand of a multiplication to at most 90 gytes. It does not seem to be very difficult to remove these bounds, or at least to widen them considerably.

Computer Exercises

1. As is well known $f(x) = (1+x)^{1/x}$ has the limit $e = 2.71828\ 18284\ 59045\ldots$, when $x \to \infty$. Study the sequences $f(x_n)$ for $x_n = 10^{-n}$ and $x_n = 2^{-n}$, for $n = 1, 2, 3, \ldots$. Stop when $x_n < 10^{-10}$ (or when $x_n < 10^{-20}$ if you are using double precision). Give your results as a table of n, x_n, and the relative error $g_n = (f(x_n) - e)/e$. Also plot $\log(|g_n|)$ against $\log(|x_n)|$. Comment on and explain your observations.

Hint: The Maclaurin expansion of $\ln(1+x)$ is useful. Both truncation and roundoff errors occur.

2. Make up and run some simple examples with several choices of the parameter s, such that you can easily check the accuracy of the result. For example: $1/7$, $\sqrt{0.75}$, $\sin(\pi/3)$, e, $4 \arctan 1$. (Compare also the calculations in the dia files.)

3. The ancient Greeks computed approximate values of the circumference of the unit circle, 2π, by inscribing a regular polygon and computing its perimeter. Archimedes considered the inscribed 96-sided regular polygon, whose perimeter is 6.2821. In general, a regular n-sided polygon inscribed in a circle with radius 1 has circumference $2a_n = 2n\sin(\pi/n)$. If we put $h = 1/n$, then

$$a(h) = a_{1/h} = \frac{1}{h}\sin \pi h = 2\pi - \frac{\pi^3}{3}h^2 + \frac{\pi^5}{60}h^4 - \ldots,$$

and thus $a(h)$ satisfies the assumptions for repeated Richardson extrapolation with $p_k = 2k$.

A recursion formula that leads from a_n to a_{2n} was given in Example 3.3.19. Setting $n_m = n_1 \cdot 2^{m-1}$, we have $a_{n_m} = n_m/s_m$, where $s_m = 1/\sin(\pi/n_m)$ and $t_m = 1/\tan(\pi/n_m)$ satisfy the recursion

$$t_m = s_{m-1} + t_{m-1}, \qquad s_m = \sqrt{t_m^2 + 1}, \quad m = 1, 2, \ldots. \tag{B.3.1}$$

The script file polygons.m uses this recursion after the substitutions

$$n = 6 * 2^{m-1}, \quad m = 1 : M,\ M \le 36, \quad a_n = p(m+1), \quad q = p/n.$$

The script polygons.m then calls the function rich3.m that performs Richardson extrapolations until the list of M polygons is exhausted or the sequence of estimates of the limit 2π ceases to be monotonic.
Choose a suitable M, $M \leq 36$, and call polygons.m. Compare with the diary file pippi2.dia that contains previous runs of this. Study the elapsed time.

4. Write a code for the summation of an infinite series $\sum f(n)$ by Euler–Maclaurin's Summation Formula, assuming that convenient algorithms exist for the integral and for derivatives of arbitrary order. Consider also how to handle generalized cases where a *limit* is asked for, rather than a sum, e.g., Stirling's asymptotic expansion for $\ln\Gamma(z)$ or the Euler constant γ.
The numerators and denominators of some Bernoulli numbers B_{2n}, $n = 1 : 17$, are found in the file const.mat, in the vectors $B2nN$ and $B2nD$, respectively. $B0 = 1,\ B1 = -1/2$, are given separately.

4. An interesting table of mathematical constants (40 decimal places) is given in Knuth [**?**, Appendix A, p. 659]. Compute a few of them to much higher accuracy. For some of them an estimate of the accuracy may be most easily obtained by comparing results obtained using different values of the parameter s. (Compare also the diary files given in the directory of Mulprec.[5]) Some of the constants may require some version of the Euler–Maclaurin formula, see Example 3. Incorporate them to your const.mat, if they are interesting. $\Gamma(1/3)$ and $-\zeta'(2)$ (the derivative of Riemann's ζ-function) seem to be relatively advanced tasks.

5. Write and test a code for the product of a mulprec matrix by a mulprec vector. Incorporate into your mulprec library.

6. Implement mullong.m according to the indications given above in "Basic arithmetic operations", or in some different way. Do something about the consequences of this for expo.m, if you want to treat the next exercise.

7. Poisson's Summation Formula reads, in the case $f(t) = e^{-t^2h^2}$ with the Fourier Transform $\hat{f}(\omega) = (\sqrt{\pi}/h)e^{-\omega^2/(4h^2)}$,

$$h \sum_{n=-N}^{N} e^{-n^2h^2} = \sqrt{\pi} \sum_{k=-K+1}^{K-1} e^{-\pi^2k^2/h^2} + R_{h,N,K,choppings}.$$

This particular case is also known as the Theta Transformation Formula.

8. Suppose that you want to compute $\sqrt{\pi}$ to an extreme accuracy, by letting a computer that didn't cost more than \$2000 (say) work over a weekend with the use of Mulprec (with a few amendments). For a given (appprox.) bound for $R_{h,N,K,choppings}$, determine a good choice of the parameters h, N, K, and the parameter s in the various terms. Estimate roughly the relation of computing time to error.
• Problem 6 must have been treated, at least in principle, before you can solve this.

[5] We suspect that one digit is wrong in $\sqrt{5}$. Are we right? By the way, Knuth denotes $(\sqrt{5}+1)/2$ by ϕ.

- Note that the function evaluation can be arranged as a set of recursion formulas with basic arithmetic operations only. I believe that only two or three evaluations of the exponential will be needed in the whole computation.
- Leave the door open for the use of variable precision, but I am not sure that it will reduce the computing time by a terrific amount in this exercise.
- Note that π appears in several places in the equation. Think of the computation as an iterative process (although in practice one iteration is perhaps enough).
- Before you make a full scale experiment, make sure that neither your computer—nor your office—will be a ruin, when you return after weekend.

Appendix C

Guide to Literature

C.1 Introduction

For many readers numerical analysis is studied as an important applied subject. Since the subject is still in a dynamic stage of development, it is important to keep track of recent literature. Therefore we give in the following a more complete overview of the literature than is usually given in textbooks. We restrict ourselves to books written in English. Although the selection presented is by no means complete and reflects a subjective choice, we hope it can serve as a guide for a reader who out of interest (or necessity!) wishes to deepen his knowledge. Both more recent textbooks and older classics are included. Reviews of most books of interest can be found in *Mathematical Reviews* as well as in *SIAM Review* and *Mathematics of Computation* A valuable source book to the literature before 1956 is Parke [29, 1958]. An interesting account of the history of numerical analysis from the 16th through the 19th century can be found in Goldstine [12, 1977]

More monographs specialized in linear algebra, approximation, ordinary and partial differential equations, and other various areas, will be listed and commented on in the two later volumes in this series.

Starting in the 1960s much general purpose software, often collected in large libraries or packages have been developed. Two large suppliers of commercial scientific subroutine libraries are NAG and IMSL. MATLAB is a much used interactive system for matrix computations, with "toolboxes" available for many application areas, e.g., control problems. Many programs and packages are available in the public domain and can be downloaded free. A prime example is LAPACK, which superseded LINPACK and EISPACK in the mid 1990s, and contains programs for solving linear systems and eigenvalue problems. Other packages like DASSL are available for solving ordinary systems of differential equations. For a survey of these we refer to Vol. III.

For software the National Institute of Standards and Technology (NIST) Guide to Available Mathematical Software (GAMS) is available at the Internet URL "gams.nist.gov". GAMS is an on-line cross-index of mathematical and statistical software providing abstracts, documentation, and source code of software modules and provides access to multiple repositories operated by others. Currently four repositories are indexed, three within NIST, and netlib. Both public-domain and proprietary software are indexed although source code of proprietary software is not redistributed by GAMS. Netlib is a repository of public domain mathematical software, data, address lists, and other useful items for the scientific computing community. Access to netlib is via the Internet URL ”www.netlib.bell-labs.com”

C.2 Textbooks in Numerical Analysis

Recent textbooks, which can be read as a complement to this book, include Deuflhard and Hohmann [7, 2003]. Gautschi [11, 1997] is a carefully written introductory text with a wealth of computer exercises and much valuable information in notes after each chapter. The book by Stoer and Bulirsch [39, 2002] is particularly suitable for a reader with a good mathematical background. More elementary but useful books are Van Loan [43, 2000], and Stewart [37, 38]. Conte and de Boor [6, 1980], Several books contain listings of algorithms, or even comes with a disk containing software, for example, the introductory book by Forsythe, Malcolm, and Moler [8, 1977] and its successor Kahaner, Moler and Nash [23, 1988].

Press et al. [30, 1993] gives an unsurpassed survey of contemporary numerical methods for the applied scientist together with software available on-line. The book by Gander and Hřebiček [10, 1997] contains a collection of weel chosen problems in scientific computing and their solution by modern software tools like Matlab and Maple . Another collection of solved problems which is entertaining and highly instructive is contained in the SIAM 100-digit challange [4].

Advanced classical texts include Isaacson and Keller [22, 1966], Hamming [17, 1974], Ralston and Rabinowitz [32, 1978], and Schwarz [36, 1997]. Strang [40, 1986] gives an excellent modern introduction to applied mathematics.

[1] F. S Acton. *Numerical Methods That (Usually) Work.* Math. Assoc. of America, New York, second edition, 1990.

[2] K. E. Atkinson. *An Introduction to Numerical Analysis.* Wiley, New York, second edition, 1989.

[3] E. K. Blum. *Numerical Analysis and Computation: Theory and Practice.* Addison-Wesley, Reading, MA, 1972.

[4] F. Borneman, D. Laurie, S. Wagon, and J. Waldvogel. *The SIAM 100-digit Challenge. A Study in High-Accuracy Numerical Computing.* SIAM, Philadelphia, PA, 2004.

[5] E. W. Cheney and D. Kincaid. *Numerical Mathematics and Computing.* Brooks/Cole, Pacific Grove, CA, third edition, 1994.

[6] S. D. Conte and C. de Boor. *Elementary Numerical Analysis. An Algorithmic Approach.* McGraw-Hill, New York, third edition, 1980.

[7] P. Deuflhard and A. Hohmann. *Numerical Analysis in Modern Scientific Computing.* Springer, Berlin, second edition, 2003.

[8] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer Methods for Mathematical Computations.* Prentice-Hall, Englewood Cliffs, NJ, 1977.

[9] C.-E. Fröberg. *Numerical Mathematics. Theory and Computer Applications.* Benjamin/Cummings, Menlo Park, CA, 1985.

[10] W. Gander and J. Hřebiček. *Solving Problems in Scientific Computing using MAPLE and MATLAB.* Springer-Verlag, Berlin, third edition, 1997.

[11] W. Gautschi. *Numerical Analysis, an Introduction.* Birkhäuser, Boston, MA, 1997.

[12] H. H. Goldstine. *A History of Numerical Analysis from the 16th through the 19th Century.* Springer-Verlag, New York, 1977.

[13] G. H. Golub, editor. *Studies in Numerical Analysis.* The Math. Assoc. of America, 1984.

[14] G. H. Golub and J. M. Ortega. *Scientific Computing and Differential Equations. An Introduction to Numerical Methods.* Academic Press, San Diego, CA, 1992.

[15] G. H. Golub and J. M. Ortega. *Scientific Computing. An Introduction with Parallel Computing.* Academic Press, 1993.

[16] G. Hämmerlin and K.-H. Hoffmann. *Numerical Mathematics.* Springer-Verlag, Berlin, 1991.

[17] R. W. Hamming. *Numerical Methods for Scientists and Engineers.* McGraw-Hill, New York, second edition, 1974.

[18] M. T. Heath. *Scientific Computing. An Introductory Survey.* McGraw-Hill, Boston, MA, second edition, 2002.

[19] P. Henrici. *Elements of Numerical Analysis.* John Wiley, New York, 1964.

[20] F. B. Hildebrand. *Introduction to Numerical Analysis.* McGraw-Hill, New York, 1974.

[21] A. S. Householder. *Principles of Numerical Analysis.* McGraw-Hill, New York, 1953.

[22] E. Isaacson and H. B. Keller. *Analysis of Numerical Methods.* Dover, New York, NY, 1994.

[23] D. Kahaner, C. B. Moler, and S. Nash. *Numerical Methods and Software.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[24] D. Kincaid and W. Cheney. *Numerical Analysis.* Brooks/Cole, Pacific Grove, CA, second edition, 1996.

[25] C. Lanczos. *Applied Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1956.

[26] G. I. Marchuk. *Methods in Numerical Mathematics.* Springer-Verlag, Berlin, second edition, 1982.

[27] J. C. Nash. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation.* American Institute of Physics, New York, second edition, 1990.

[28] J. Ortega. *Numerical Analysis: A Second Course.* Academic Press, New York, 1972.

[29] N. G. Parke. *Guide to the Literature of Mathematics and Physics.* Dover Publications, New York, second edition, 1958.

[30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in Fortran 77; The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, second edition, 1993.

[31] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics.* Springer-Verlag, New York, 2000.

[32] A. Ralston and P. Rabinowitz. *A First Course in Numerical Analysis.* McGraw-Hill, New York, second edition, 1978.

[33] J. R. Rice. *Mathematical Software.* Academic Press, New York, 1971.

[34] J. R. Rice. *Numerical Methods, Software, and Analysis.* Academic Press, New York, 1983.

[35] H. Rutishauser. *Lectures on Numerical mathematics.* Birkhäuser, Boston, MA, 1990.

[36] H. R. Schwarz. *Numerische Methematik.* Teubner, Stuttgart, fourth edition, 1997. English translation of 2nd ed.: Numerical Analysis: A Comprehensive Introduction, Wiley, New York.

[37] G. W. Stewart. *Afternotes on Numerical Analysis.* SIAM, Philadelphia, PA, 1996.

[38] G. W. Stewart. *Afternotes Goes to Graduate School.* SIAM, Philadelphia, PA, 1997.

[39] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, New York, third edition, 20002.

[40] G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1986.

[41] J. Todd, editor. *A Survey of Numerical Analysis.* McGraw-Hill, New York, 1962.

[42] C. W. Ueberhuber. *Numerical Computation. 1 & 2.* Springer-Verlag, Berlin, 1997.

[43] C. F. Van Loan. *Introduction to Scientidic Computing.* Prentice-Hall, Upper Saddle River, NJ, second edition, 2000.

[44] J. S. Vandergraft. *Introduction to Numerical Computations.* Academic Press, New York, 1983.

[45] D. M. Young and R. T. Gregory. *A Survey of Numerical Analysis. Vol. 1.* Addison-Wesley, Reading, MA, 1972.

[46] D. M. Young and R. T. Gregory. *A Survey of Numerical Analysis. Vol. 2.* Addison-Wesley, Reading, MA, 1973.

C.3 Handbooks, Tables and Formulas

Some principal questions in the production of software for mathematical computation are discussed in Rice [15, 1971],

Mathematical tables are no longer as important for numerical calculations as they were in the pre-computer days. However, tables can often be an important aid in checking a calculation or planning calculations on a computer. Detailed advice about the use and choice of tables is given in Todd [17, 1962, pp. 93–106], The classical six-figure tables of A most comprehensive source of information on mathematical functions and formulas is Abramowitz and Stegun [1, 1965]. An excellent overview of software for mathematical special functions is given by Lozier and Olver [13]

The three monographs edited by Jacobs [9, 1977], Iserles and Powell [8, 1987], and Duff and Watson [5, 1997] give exellent surveys of the development of "state of the art" methods in many different areas of numerical analysis during the last decades. The Handbook of Numerical Analysis [4], edited by P. G. Ciarlet and J. L. Lions, is a multivolume sequence that offers comprehensive coverage in all areas of numerical analysis as well as many actual problems of contemporary interest. Very useful surveys articles are to be found in ACTA Numerica, a Cambridge University Press Annual started in 1992.

Two general mathematics dictionaries, which are useful to have at hand are [11] and [18].

[1] M. Abramowitz and I. A. Stegun (eds.). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover Publications, New York, 1965.

[2] Yu. A. Brychkov, A. P. Prudnikov, and O. I. Marichev. *Integrals and Series. Vol. 1: Elementary Functions.* Gordon and Breach, New York, 1986.

[3] R. Churchhouse, editor. *Handbook of Applicable Mathematics*, volume III. Numerical Methods. Wiley-Interscience, New York, 1981.

[4] P. G. Ciarlet and J. L. Lions. *Handbook of Numerical Analysis*, volume I–VII. North-Holland, Amsterdam, 1990–2000.

[5] I. S. Duff and G. A. Watson, editors. *The State of the Art in Numerical Analysis.* Clarendon Press, Oxford, 1997.

[6] B. Engquist and W. Schmid, editors. *Mathematics Unlimited—2001 and Beyond.* Springer-Verlag, Berlin, 2001.

[7] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series and Products.* Academic Press, London, UK, fifth edition, 1993.

[8] A. Iserles and M. J. D. Powell, editors. *The State of the Art in Numerical Analysis.* Clarendon Press, Oxford, 1987.

[9] D. A. H. Jacobs, editor. *The State of the Art in Numerical Analysis.* Clarendon Press, Oxford, 1977.

[10] E. Jahnke, F. Emde, and F. Lösh. *Tables of Higher Functions.* McGraw-Hill, New York, sixth edition, 1960.

[11] R. C. James and E. F. Beckenbach, editors. *James & James Mathematics Dictionary.* Van Nostrand, Princeton, NJ, third edition, 1968.

[12] A. V. Lebedev and R. M. Federova. *A Guide to Mathematical Tables.* Van Nostrand, New York, 1960.

[13] D. W. Lozier and F. W. J. Olver. Numerical evaluation of special functions. In W. Gautschi, editor, *Mathematics of Computation 1943–1993: A Half-Century of Computational Mathematics*, volume 48 of *Proc. Sympos. Appl. Math.*, pages 79–125, Providence, RI, 1994. Amer. Math. Soc.

[14] A. P. Prudnikov, Yu. A. Brychkov, and O. I. Marichev. *Integrals and Series. Vol. 2: Special Functions.* Gordon and Breach, New York, 1986.

[15] J. R. Rice. *Mathematical Software.* Academic Press, New York, 1971.

[16] J. Spanler and K. B. Oldham. *An Atlas of Functions.* Springer-Verlag, Berlin, 1987.

[17] J. Todd, editor. *A Survey of Numerical Analysis.* McGraw-Hill, New York, 1962.

[18] E. W. Weisstein, editor. *CRC Concise Encyclopedia of Mathematics.* CRC Press, Boca Raton, FL, 2000.

Index