Contents

Appendix A

Calculus in Vector Spaces

We shall introduce some notions and notations from the calculus in vector spaces that will be useful in this and in later chapters. A more general and rigorous treatment can be found, e.g., in Dieudonné [2]. Our presentation is also much influenced by Butcher [1, Chapter 1], whose purpose is rather similar to ours, but his discussion is stricter. In these books the reader may find some proofs that we omit here. There are, in the literature, several different notations for these matters, e.g., **multilinear mapping** notation, **tensor** notation, or, in some cases, **vector-matrix** notation. None of them seems to be perfect or easy to handle correctly in some complex situations. This may be a reason to become familiar with several notations.

A.1 Multilinear Mappings

Consider $k+1$ vector spaces X_1, $X_2, \ldots,$ X_k, Y, and let $x_\nu \in X_\nu$. A function $A\colon X_1 \times X_2 \ldots \times X_k \to Y$ is called k-**linear**, if it is linear in each of its arguments x_i separately. For example, the expression $(Px_1)^T Q x_2 + (Rx_3)^T S x_4$ defines a 4-linear function, mapping or operator (provided that the constant matrices P, Q, R, S have appropriate size). If $k=2$ such a function is usually called **bilinear**, and more generally one uses the term **multilinear**. , , , ,

Let $X_\nu = \mathbf{R}^{n_\nu}$, $\nu = 1, 2, \ldots, k$, $Y = \mathbf{R}^m$, and let e_{j_i} be one of the basis vectors of X_i. We use *superscripts* to denote coordinates in these spaces. Let $a^i_{j_1,j_2,\ldots,j_k}$ denote the ith coordinate of $A(e_{j_1}, e_{j_2}, \ldots, e_{j_k})$. Then, because of the linearity, the ith coordinate of $A(x_1, x_2, \ldots, x_k)$ reads

$$\sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \ldots \sum_{j_k=1}^{n_k} a^i_{j_1,j_2,\ldots,j_k} x_1^{j_1} x_2^{j_2} \ldots x_k^{j_k}, \quad x_\nu \in X_\nu. \tag{A.1.1}$$

We shall sometimes use the **sum convention** of tensor analysis; if an index occurs both as a subscript and as a superscript, the product should be summed over the range of this index, i.e., the ith coordinate of $A(x_1, x_2, \ldots, x_k)$ reads shorter $a^i_{j_1,j_2,\ldots,j_k} x_1^{j_1} x_2^{j_2} \ldots x_k^{j_k}$. (Remember always that the superscripts are no exponents.)

Suppose that $X_i = X$, $i = 1, 2, \ldots, k$. Then, the set of k-linear mappings from X^k to Y is itself a linear space called $L_k(X,Y)$. For $k = 1$, we have the space of linear functions, denoted more shortly by $L(X,Y)$. Linear functions can, of course, also be described in vector-matrix notation; $L(\mathbf{R}^n, \mathbf{R}^m) = \mathbf{R}^{m\times n}$, the set of matrices defined in Section 6.2. Matrix notation can also be used for each coordinate of a bilinear function. These matrices are in general unsymmetric.

Norms of multilinear operators are defined analogously to subordinate matrix norms. For example,

$$\|A(x_1, x_2, \ldots, x_k)\|_\infty \le \|A\|_\infty \|x_1\|_\infty \|x_2\|_\infty \cdots \|x_k\|_\infty,$$

where

$$\|A\|_\infty = \max_{i=1}^{m} \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_k=1}^{n_k} |a^i_{j_1,j_2,\ldots,j_k}|. \tag{A.1.2}$$

A multilinear function A is called *symmetric*, if $A(x_1, x_2, ..., x_k)$ is symmetric with respect to its arguments. In the cases mentioned above, where matrix notation can be used, the matrix becomes symmetric, if the multilinear function is symmetric.

We next consider a function $f\colon X \to Y$, not necessarily multilinear, where X and Y are normed vector spaces. This function is *continuous*, at the point $x_0 \in X$ if $\|f(x) - f(x_0)\| \to 0$ as $x \to x_0$, (i.e. as $\|x - x_0\| \to 0$). The function f satisfies a **Lipschitz condition** in a domain $D \subset X$, if a constant α, called a *Lipschitz constant*, can be chosen so that $\|f(x') - f(x'')\| \le \alpha\|x' - x''\|$ for all points x', $x'' \in D$.

The function f is *differentiable* at x_0, in the sense of Fréchet, if there exists a *linear* mapping A such that

$$\|f(x) - f(x_0) - A(x - x_0)\| = o(\|x - x_0\|), \quad x \to x_0.$$

This linear mapping is called the **Fréchet derivative** of f at x_0, and we write $A = f'(x_0)$ or $A = f_x(x_0)$. Note that (the value of) $f'(x_0) \in L(X,Y)$. (Considered as a function of x_0, $f'(x_0)$ is, of course, usually non-linear.)

These definitions apply also to infinite dimensional spaces. In the finite dimensional case, the Fréchet derivative is represented by the **Jacobian** matrix, the elements of which are the partial derivatives $\partial f^i/\partial x^j$, also written f^i_j, in an established notation, e.g., in tensor analysis; superscripts for coordinates and subscripts for partial derivation. If vector-matrix notation is used, it is important to note that the derivative g' of a real-valued function g is a *row* vector, since

$$g(x) = g(x_0) + g'(x_0)(x - x_0) + o(\|x - x_0\|).$$

We suggest that the notation gradient, or $\operatorname{grad} g$ is used for the transpose of $g'(x)$.

A *differential* reads, in the multilinear mapping notation, $df = f'dx$ or $df = f_x dx$. In tensor notation with the sum convention, it reads $df^i = f^i_j dx^j$.

Many results from elementary calculus carry over to vector space calculus, such as the rules for the differentiation of products. The proofs are in principle the same.

If $z = f(x, y)$ where $x \in \mathbf{R}^k$, $y \in \mathbf{R}^l$, $z \in \mathbf{R}^m$ then we define *partial derivatives* f_x, f_y with respect to the *vectors* x, y by the differential formula

$$df(x, y) = f_x dx + f_y dy, \quad \forall dx \in \mathbf{R}^k, \quad dy \in \mathbf{R}^l. \tag{A.1.3}$$

If x, y are functions of $s \in \mathbf{R}^n$, then a general version of the *chain rule* reads

$$f'(x(s), y(s)) = f_x x'(s) + f_y y'(s). \tag{A.1.4}$$

The extension to longer chains is straightforward. These equations can also be used in infinite dimensional spaces.

Consider a function $f: \mathbf{R}^k \to \mathbf{R}^k$, and consider the equation $x = f(y)$. By formal differentiation, $dx = f'(y)dy$, and we obtain $dy = \big(f'(y)\big)^{-1} dx$, provided that the Jacobian $f'(y)$ is non-singular. In Section 13.2.4, we shall see sufficient conditions for the solvability of the equation $x = f(y)$, so that it defines, in some domain, a differentiable *inverse function of* f, such that $y = g(x)$, $g'(x) = \big(f'(y)\big)^{-1}$.

Another important example: if $f(x, y) = 0$ then, by (A.1.4), $f_x dx + f_y dy = 0$. If $f_y(x_0, y_0)$ is a non-singular matrix, then, by the *implicit function theorem* (see Dieudonné [2, Section 10.2]) y becomes, under certain additional conditions, a differentiable function of x in a neighborhood of (x_0, y_0), and we obtain $dy = -(f_y)^{-1} f_x dx$, hence $y'(x) = -(f_y)^{-1} f_x|_{y=y(x)}$.

One can also show that

$$\lim_{\epsilon \to +0} \frac{f(x_0 + \epsilon v) - f(x_0)}{\epsilon} = f'(x_0)v.$$

There are, however, functions f, where such a *directional derivative* exists for any v but, for some x_0, is not a linear function of v. An important example is $f(x) = \|x\|_\infty$, where $x \in \mathbf{R}^n$. (Look at the case $n = 2$.) The name *Gateaux derivative* is sometimes used in such cases, in order to distinguish it from the Fréchet derivative $f'(x_0)$ previously defined.

If $f'(x)$ is a differentiable function of x at the point x_0, its derivative is denoted by $f''(x_0)$. This is a linear function that maps X into the space $L(X, Y)$ that contains $f'(x_0)$, i.e., $f''(x_0) \in L(X, L(X, Y))$. This space may be identified in a natural way with the space $L_2(X, Y)$ of bilinear mappings $X^2 \to Y$; if $A \in L(X, L(X, Y))$ then the corresponding $\bar{A} \in L_2(X, Y)$ is defined by $(Au)v = \bar{A}(u, v)$ for all $u, v \in X$; in the future it is not necessary to distinguish between A and $\bar{A}$. So,

$$f''(x_0)(u, v) \in Y, \qquad f''(x_0)u \in L(X, Y), \qquad f''(x_0) \in L_2(X, Y).$$

It can be shown that $f''(x_0)$: $X^2 \to Y$, *is a symmetric bilinear mapping*, i.e. $f''(x_0)(u, v) = f''(x_0)(v, u)$. The second order partial derivatives are denoted f_{xx}, f_{xy}, f_{yx}, f_{yy}. One can show that

$$f_{xy} = f_{yx}.$$

If $X = \mathbf{R}^n$, $Y = \mathbf{R}^m$, $m > 1$, $f''(x_0)$ reads $f^p_{ij}(x_0) = f^p_{ji}(x_0)$ in tensor notation. It is thus characterized by a three-dimensional array, which one rarely needs to store or write. Fortunately, most of the numerical work can be done on a lower level, e.g., with directional derivatives. For each fixed value of p we obtain a symmetric $n \times n$ matrix, named the **Hessian** matrix $H(x_0)$; note that $f''(x_0)(u,v) = u^T H(x_0)v$. The Hessian can be looked upon as the derivative of the gradient. An element of this Hessian is, in the multilinear mapping notation, the pth coordinate of the vector $f''(x_0)(e_i, e_j)$.

We suggest that the vector-matrix notation is replaced by the multilinear mapping formalism when handling derivatives of vector-valued functions of order higher than one. The latter formalism has the further advantage that it can be used also in infinite-dimensional spaces (see Dieudonné [2]). In finite dimensional spaces the tensor notation with the summation convention is another alternative.

Similarly, higher derivatives are recursively defined. If $f^{(k-1)}(x)$ is differentiable at x_0, then its derivative at x_0 is denoted $f^{(k)}(x_0)$ and called the kth derivative of f at x_0. One can show that $f^{(k)}(x_0): X^k \to Y$ is a *symmetric* k-linear mapping. **Taylor's formula** then reads, when $a,\, u \in X$, $f: X \to Y$,

$$f(a+u) = f(a) + f'(a)u + \tfrac{1}{2}f''(a)u^2 + \ldots + \frac{1}{k!}f^{(k)}(a)u^k + R_{k+1}, \quad \text{(A.1.5)}$$

$$R_{k+1} = \int_0^1 \frac{(1-t)^k}{k!} f^{(k+1)}(a+ut)dtu^{k+1};$$

it follows that

$$\|R_{k+1}\| \le \max_{0\le t\le 1} \left\| f^{(k+1)}(a+ut) \right\| \frac{\|u\|^{k+1}}{(k+1)!}.$$

After some hesitation, we here use u^2, u^k, etc. as abbreviations for the lists of input vectors (u,u), $(u,u,\ldots,u)$ etc.. This exemplifies simplifications that you may allow yourself (and us) to use when you have got a good hand with the notation and its interpretation. Abbreviations that reduce the number of parentheses often increase the clarity; there may otherwise be some risk for ambiguity, since parentheses are used around the arguments for both the usually non-linear function $f^{(k)}$: $X \to L_k(X,Y)$ and the k-linear function $f^{(k)}(x_0)$: $X^k \to Y$. You may also write, e.g., $(f')^3 = f'f'f'$; beware that you do not mix up $(f')^3$ with f'''.

The mean value theorem of differential calculus and Lagrange's form for the remainder of Taylor's formula are not true, but they can in many places be replaced by the above *integral form of the remainder*. All this holds in complex vector spaces too.

In the following subsections we show some relevant applications of these notions to numerical mathematics.

A.2 Taylor Coefficients for the Solution of a System of Ordinary Differential Equations.

Let y be a function of the real variable t, that satisfies the autonomous differential system $\dot{y} = f(y)$, $f: Y \to Y$.[1] We shall derive recursion formulas for the derivatives

[1] A differential system of equations is said to be autonomous if it does not explicitely contain the independent variable.

of the solution $y(t)$ with respect to t. We use dots for differentiation with respect to t of order less than 3, and we set $\ddot{y} = z$.

By repeated application of the chain rule, the time derivatives of $y(t)$ are expressed in terms of the derivatives of f with respect to the vector y. In the tables below the results are given first in the multilinear mapping notation with primes for differentiation with respect to y (as above). In the last line of the tables, the same vectors are expressed in tensor notation.

$\dot{y}$	$z = \ddot{y}$	$\dot{z} = y^{(3)}$
$f(y)$	$f'(y)\dot{y}$	$f''(y)\dot{y}^2 + f'(y)\ddot{y}$
f	$f'f$	$f''f^2 + (f')^2 f$
f^j	$f^j_k f^k$	$f^j_{kl} f^k f^l + f^j_k f^k_l f^l$

$\ddot{z} = y^{(4)}$
$f'''(y)\dot{y}^3 + 3f''(y)(\ddot{y}, \dot{y}) + f'(y)\dot{z}$
$f'''f^3 + 3f''(f'f, f) + f'f''f^2 + (f')^3 f$
$f^j_{klm} f^k f^l f^m + 3f^j_{km} f^k_l f^l f^m + f^j_k f^k_{lm} f^l f^m + f^j_k f^k_l f^l_m f^m$

Note that, at some places, we have here omitted the *obvious* argument y. We often do so when there is no doubt about the argument.

The individual terms on the third and fourth lines of these tables are called **elementary differentials**. The qth order derivative of y is a linear combination of the qth order elementary differentials with *integer coefficients*. They are fundamental in the theory of one-step methods for ordinary differential equations; see Section 13.3.

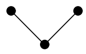
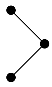
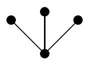
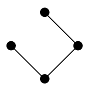
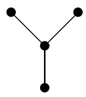
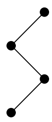
These matters can easily become rather messy. J. Butcher and others have made the analysis more transparent by employing an one-to-one correspondence between the qth order elementary differentials and a **rooted tree** with q vertices. We denote a rooted tree by $\mathbf{t}$; its order, that is the number of vertices, is denoted $\rho(\mathbf{t})$, and the corresponding elementary differential is denoted $F(\mathbf{t})$. The qth order trees are denoted $\mathbf{t}_{q1}$, $\mathbf{t}_{q2}, \ldots$.

Table 11.5.1 displays up to order 4 the elementary differentials and trees. (analogous to the tree $\mathbf{t_{32}}$). It corresponds to the elementary differential $(f')^3 f$. Study the table, and see Problem 7. Note the monotonic ordering of the labels along the branches, and see how well the tensor notation corresponds to this labeling. $F(\mathbf{t})$ denotes the elementary differential, which corresponds to the tree $\mathbf{t}$, e.g., $F(\mathbf{t_{21}}) = f'f$. A tree $\mathbf{t}$ can be labeled in several ways. A parameter named $\alpha(\mathbf{t})$ equals, in a certain sense, the number of essentially different monotonic labelings of $\mathbf{t}$; $\alpha(\mathbf{t}) = 1$ for all trees in the figure, except for $\alpha(\mathbf{t_{42}}) = 3$. (Pure permutation of the labels of leaves on the same branch is not "essential".) The precise definition of $\alpha(\mathbf{t})$ is rather subtle, and we refer to Hairer, Nørsett and Wanner [1993, Ch.2] or Butcher loc.cit. for more detailed information. We give in §13.3.1 a table with $\alpha(\mathbf{t})$ and some other data for $\rho(\mathbf{t}) \le 5$.

With these notations, the formal Taylor expansion of the solution $y(t)$ around $t = t_0$ reads

$$y(t_0 + h) = y(t_0) + y'(t_0)h + \frac{1}{2!}y''(t_0)h^2 + \frac{1}{3!}y'''(t_0)h^3 + \ldots$$

Table A.2.1. *Elementary differentials and the corresponding trees up to order* $\rho(\mathbf{t}) = 4$.

order	$\mathbf{t}$	graph	$F(\mathbf{t})$	
1	$\mathbf{t}_{11}$		f	f^j
2	$\mathbf{t}_{21}$		$f'f$	$f^j_k f^k$
3	$\mathbf{t}_{31}$		$f''f^2$	$f^j_{kl} f^k f^l$
	$\mathbf{t}_{32}$		$(f')^2 f$	$f^j_k f^k_l f^l$
4	$\mathbf{t}_{41}$		$f'''f^3$	$f^j_{klm} f^k f^l f^m$
	$\mathbf{t}_{42}$		$f''(f'f, f)$	$f^j_{km} f^k_l f^l f^m$
	$\mathbf{t}_{43}$		$f'f''f^2$	$f^j_{km} f^k_l f^l f^m$
	$\mathbf{t}_{44}$		$(f')^3 f$	$f^j_k f^k_{lm} f^l f^m$

$$
\begin{aligned}
&= y(t_0) + hfy(t_0) + \frac{h^2}{2!} h^2 f'fy(t_0) + \frac{h^3}{3!}(f''f^2 + (f')^2 f)y(t_0) + \dots \\
&= y(t_0) + \Big(hF(\mathbf{t_{11}}) + \frac{h^2}{2!} F(\mathbf{t_{21}}) + \frac{h^3}{3!}\big(F(\mathbf{t_{31}}) + F(\mathbf{t_{32}})\big) + \dots\Big) y(t_0).
\end{aligned}
$$

More generally, the Taylor expansion becomes,

$$
y(t_0 + h) = y(t_0) + \sum_{\mathbf{t}} \frac{h^{\rho(\mathbf{t})}}{\rho(\mathbf{t})!} \alpha(\mathbf{t}) F(\mathbf{t}) y(t_0), \quad \rho(\mathbf{t}) \geq 1. \tag{A.2.1}
$$

This expression is useful for the design and analysis of numerical methods. If you want to use a Taylor expansion for computing the numerical solution of a system, however, you had better use the techniques of *automatic differentiation*, see Section 3.1 and Section 13.3.

The number of elementary differentials for $q = 1 : 10$ are as follows:

$$
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
1 & 1 & 2 & 4 & 9 & 20 & 48 & 115 & 286 & 719
\end{array} \tag{A.2.2}
$$

Much more about this can be found in Butcher, loc.cit., and Hairer, Nørsett and Wanner, loc.cit..

The formulas for an autonomous system, $\dot{y} = f(y)$, include also the non-autonomous case, i.e. a system of the form $\dot{y} = f(t,y)$, for if we add the trivial equation $\dot{t} = 1$ to the latter system, then we obtain an autonomous system for the vector (t,y), (written as a column). Nevertheless, since the variable t plays a special role, it is sometimes interesting to see the formulas for the non-autonomous system more explicitly. Recall that $f_{ty} = f_{yt}$.

$$\begin{aligned}
&\dot{y} = f(t,y)\\
&z = \ddot{y} = df(t,y(t))/dt = f_t + f_y\dot{y} = f_t + f_y f\\
&\dot{z} = (f_t + f_y f)_t + (f_t + f_y f)_y f = f_{tt} + f_y f_t + 2f_{yt} f + f_{yy} f^2 + f_y f_y f,
\end{aligned}$$

PROBLEMS

4. Consider the multilinear operator A defined by (A.1.1), and suppose that $X_\nu = \mathbf{R}^n$, $\forall \nu$. What is $\|A\|$ if a weighted max-norm is used in $\mathbf{R}^n$?

5. Write a program for the approximate computation of the Hessian of a real-valued function, by central differences.

6. Consider an autonomous system, $\dot{y} = f(y)$, $f : \mathbf{R}^s \to \mathbf{R}^s$. Such a system has an infinity of solutions $y(t)$, but we shall see in Section 13.1 that, for given $\tau \in \mathbf{R}$, $\eta \in \mathbf{R}^s$, there is, under very general conditions on the function f, only one solution for which $y = \eta$ for $t = \tau$. Denote this solution by $y(t;\tau,\eta)$. Runge's 2nd order method, introduced in Section 1.3, reads $k_1 = hf(y_n)$, $k_2 = hf\Big(y_n + \frac{1}{2}k_1\Big)$, $y_{n+1} = y_n + k_2$. Show that

$$y_{n+1} - y(t_n + h; t_n, y_n) = h^3\Big(\tfrac{1}{8}f''\dot{y}^2 - \tfrac{1}{6}y'''\Big) + O(h^4).$$

(This is called the local error.) Also show that $k_2 - k_1 = \frac{1}{2}h^2\ddot{y} + O(h^3)$. (The vector $k_2 - k_1$ is used for the choice of step size in the algorithm of Section 1.3. See also Section 13.2.)

7. (a) Draw the tree $\mathbf{t}_{44}$, and write down the corresponding elementary differential in multilinear mapping notation and in tensor notation.

(b) Given all trees of order $q-1$, two ways of producing (different) trees of order q are as follows. You can *either* put one more vertex on the first level above the root (and label it with the next character in the alphabet), *or* you can create a new root (labeled j) below the old one and change the other labels. Note that for $q = 3$ and $q = 4$ these operations yield all trees. Find the rules, how the elementary differentials are modified at these tree operations.

(c) For $q = 5$, however, the operations in (b) produce together 8 trees, instead of 9, according to the table in Example A.2. What does the missing tree look like? Find the corresponding elementary differential.

Comment: There is more material about this in Section 13.3.

8. Consider a function $f : X \to X$, $\dim X > 1$. Do expressions like $f''(x_0)f''(x_0)$ and $f''(x_0)f''(x_0)f''(x_0)$ ever make sense?

[1] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods.* Wiley-Interscience, Chichester, 1987.

[2] J. Dieudonné. *Foundations of Modern Analysis.* Academic Press, New York, NY, 1961.

Appendix B

Guide to Literature in Linear Algebra

The literature on linear algbera is very extensive. For an advanced theoretical treatise a classical source is Gantmacher [18, 1959]. Several nonstandard topics are covered in Lancaster and Tismenetsky [33, 1985] and in two excellent volumes by Horn and Johnson [29, 1985] and [30, 1991]. A very complete and useful book on and perturbation theory and related topics is Stewart and Sun [48, 1990]. Analytical aspects are emphasized in Lax [35, 1997].

An interesting survey of classical numerical methods in linear algebra can be found in Faddeev and Faddeeva [15, 1963], although many of the methods treated are now dated. A compact, lucid and still modern presentation is given by Householder [31, 1964]. Bellman [6, 1970] is an original and readable complementary text.

An excellent textbook on matrix computation are Stewart [45, 1973]. The recent book [46, 1998] by the same author is the first in a new series. A book which should be within reach of anyone interested in computational linear algebra is the monumental work by Golub and Van Loan [23, 1996], which has become a standard reference. The book by Higham [28, 1995] is another indispensible source book for information about the accuracy and stability of algorithms in numerical linear algebra. A special treatise on least squares problems is Björck [7, 1996].

Two classic texts on iterative methods for linear systems are Varga [53, 1962] and Young [58, 1971]. The more recent book by Axelsson [2, 1994], also covers conjugate gradient methods. Barret et al. [5, 1994] is a compact survey of iterative methods and their implementation. Advanced methods that may be used with computers with massiv parallel processing capabilities are treated by Saad [43, 1996].

A still unsurpassed text on computational methods for the eigenvalue problem is Wilkinson [56, 1965]. Wilkinson and Reinsch [57, 1971] contain detailed discussions and programs, which are very instructive. For an exhaustive treatment of the symmetric eigenvalue problem see the classical book by Parlett [40, 1980]. Large scale eigenvalue problems are treated by Saad [42, 1992]. For an introduction to the implementation of algorithms for vector and parallel computers, see also Dongarra et al. [13, 1998]. Many important pratical details on implementation of algorithms can be found in the documentation of LINPACK and EISPACK software given in Dongarra et al. [12, 1979] and Smith et al. [44, 1976]. Direct methods for sparse symmetric positive definite systems are covered in George and Liu [20, 1981], while a more general treatise is given by Duff et al. [14, 1986].

LAPACK95 is a Fortran 95 interface to the Fortran 77 LAPACK library documented in [1, 1999]. It is relevant for anyone who writes in the Fortran 95 language and needs reliable software for basic numerical linear algebra. It improves upon the original user-interface to the LAPACK package, taking advantage of the considerable simplifications that Fortran 95 allows. LAPACK95 Users' Guide [4, 2001] provides an introduction to the design of the LAPACK95 package, a detailed description of its contents, reference manuals for the leading comments of the routines, and example programs. For more information on LAPACK95 go to http://www.netlib.org/lapack95/.

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, editors. *LAPACK Users' Guide*. SIAM, Philadelphia, PA, 1999.

[2] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, Cambridge, UK, 1994.

[3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. A. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, PA, 2000.

[4] V. A. Barker, L. S. Blackford, J. J. Dongarra, S. Hammarling J. Du Croz, M. Marinova, J. Was'niewski, and P. Yalamov, editors. *LAPACK 95 Users' Guide*. SIAM, Philadelphia, PA, 2001.

[5] R. Barret, M. W. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, editors. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA, 1993.

[6] R. Bellman. *Introduction to Matrix Analysis*. McGraw-Hill, New York, 1970.

[7] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, PA, 1996.

[8] C. Brezinski. *Projection Methods for System of Equations*. Elsevier, Amsterdam, 1997.

[9] F. Chatelin. *Eigenvalues of Matrices*. Wiley, Chichester, 1993.

[10] P. G. Ciarlet. *Introduction to Numerical Linear Algebra and Optimization*. Cambridge University Press, Cambridge, UK, 1989.

[11] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[12] J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart. *LINPACK Users' Guide*. SIAM, Philadelphia, PA, 1979.

[13] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. Van der Vorst. *Numerical Linear Algebra for High-Performance Computers*. SIAM, Philadelphia, PA, 1998.

[14] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Oxford University Press, London, 1986.

[15] D. K. Faddeev and V. N. Faddeeva. *Computational Methods of Linear Algebra*. W. H. Freeman, San Francisco, CA, 1963.

[16] L. Fox. *Introduction to Numerical Linear Algebra*. Clarendon Press, Oxford, 1964.

[17] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, 1:57–100, 1992.

[18] F. R. Gantmacher. *The Theory of Matrices. Vols. I and II.* Chelsea Publishing Co, New York, 1959.

[19] B. S. Garbow, J. M. Boyle, J. J. Dongarra, and G. W. Stewart. *Matrix Eigensystems Routines: EISPACK Guide Extension*. Springer-Verlag, New York, 1977.

[20] A. George and J. W. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[21] G. H. Golub and D. O'Leary. Some history of the conjugate gradient and Lanczos algorithms: 1948–1976. *SIAM Review*, 31:50–102, 1989.

[22] G. H. Golub and H. A. van der Vorst. Closer to the solution: iterative linear solvers. In I. S. Duff and G. A. Watson, editors, *The State of the Art in Numerical Analysis*, pages 63–92. Clarendon Press, Oxford, 1997.

[23] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[24] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, PA, 1997.

[25] R. T. Gregory and L. K. Karney. *A Collection of Matrices for Testing Computational Algorithms*. Wiley, New York, 1969.

[26] L. A. Hageman and D. M. Young. *Applied Iterative Methods*. Academic Press, New York, 1981.

[27] W. W. Hager. *Applied Numerical Linear Algebra*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[28] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, 2002.

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

[30] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1991.

[31] A. S. Householder. *The Theory of Matrices in Numerical Analysis*. Dover, New York, 1964.

[32] A. Jennings and J. J. McKeown. *Matrix Computation.* Wiley, New York, second edition, 1992.

[33] P. Lancaster and M. Tismenetsky. *The Theory of Matrices.* Academic Press, New York, 1985.

[34] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* Prentice-Hall, Englewood Cliffs, NJ, 1974. Reprinted by SIAM, Philadelphia, PA, 1995.

[35] P. D. Lax. *Linear Algebra.* Wiley, New York, 1997.

[36] M. Marcus and H. Minc. *A Survey of Matrix Theory and Matrix Inequalities.* Allyn and Bacon, Boston, MA, 1964.

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

[38] B. Noble and J. W. Daniel. *Applied Linear Algebra.* Prentice-Hall, Englewood Cliffs, second edition, 1977.

[39] J. M. Ortega. *Matrix Theory. A Second Course.* Plenum Publ., New York, 1987.

[40] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Classsics in Applied Mathematics 20. SIAM, Philadelphia, PA, 1998.

[41] J. R. Rice. *Matrix Computation and Mathematical Software.* Mc Graw-Hill, New York, 1981.

[42] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Halstead Press, New York, 1992.

[43] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Co., Boston, MA, 1996.

[44] B. T. Smith, J. M. Boyle, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler. *Matrix Eigensystems Routines—EISPACK Guide.* Springer-Verlag, New York, second edition, 1976.

[45] G. W. Stewart. *Introduction to Matrix Computations.* Academic Press, New York, 1973.

[46] G. W. Stewart. *Matrix Algorithms Volume I: Basic Decompositions.* SIAM, Philadelphia, PA, 1998.

[47] G. W. Stewart. *Matrix Algorithms Volume II: Eigensystems.* SIAM, Philadelphia, PA, 2001.

[48] G. W. Stewart and J.-G. Sun. *Matrix Perturbation Theory.* Academic Press, Boston, MA, 1990.

[49] G. Strang. *Introduction to Applied Mathematics.* Wellesley-Cambridge Press, Wellesley, MA, 1988.

[50] L. N. Trefethen. Pseudospectra of linear operators. *SIAM Review*, 39:383–406, 1997.

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

[52] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem; Computational Aspects and Analysis*. SIAM, Philadelphia, PA, 1991.

[53] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, 1962.

[54] D. S. Watkins. *Fundamentals of Matrix Computation*. Wiley-InterScience, New York, 2002.

[55] J. H. Wilkinson. Error analysis of direct methods of matrix inversion. *J. ACM*, 8:281–330, 1961.

[56] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.

[57] J. H. Wilkinson and C. Reinsch, editors. *Handbook for Automatic Computation. Vol. II, Linear Algebra*. Springer-Verlag, New York, 1971.

[58] D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.

Index