Literature

Literature Cited in the Text

[1] A. KOLMOGOROV, *Ergebn. Math.* **2** (1933) 3

[2] D.E. KNUTH, *The Art of Computer Programming*, vol. 2, Addison-Wesley, Reading MA 1981

[3] P. L'ECUYER, *Comm. ACM* **31** (1988) 742

[4] B.A. WICHMANN and I.D. HILL, *Appl. Stat.* **31** (1982) 188

[5] G.E.P. BOX and M.E. MULLER, *Ann. Math. Stat.* **29** (1958) 611

[6] L. VON BORTKIEWICZ, *Das Gesetz der kleinen Zahlen*, Teubner, Leipzig 1898

[7] D.J. DE SOLLA PRICE, *Little Science, Big Science*, Columbia University Press, New York 1965

[8] M.G. KENDALL and A. STUART, *The Advanced Theory of Statistics*, vol. 2, Charles Griffin, London 1968

[9] S.S. WILKS, *Ann. Math. Stat.*, **9** (1938) 60

[10] A.H. ROSENFELD, H.A. BARBERO-GALTIERI, W.J. PODOLSKI, L.R. PRICE, P. SÖDING, CH.G. WOHL, M. ROOS, and W.J. WILLIS, *Rev. Mod. Phys.* **33** (1967) 1

[11] PARTICLE DATA GROUP, *Physics Letters* **B204** (1988) 1

[12] W.H. PRESS, B.P. FLANNERY, S.A. TEUKOLSKY and W.T. VETTERLING, *Numerical Recipes*, Cambridge University Press, Cambridge 1986

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2,

[13] R.P. BRENT, *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs NJ 1973

[14] J.A. NELDER and R. MEAD, *Computer Journal* **7** (1965) 308

[15] M.J.D. POWELL, *Computer Journal* **7** (1965) 155

[16] D.W. MARQUARDT, *J. Soc. Ind. Appl. Math.* **11** (1963) 431

[17] C. LANCZOS, *SIAM J. Numerical Analysis* **1** (1964) 86

[18] C.L. LAWSON and R.J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs NJ 1974

[19] G.H. GOLUB and W. KAHN, *SIAM J. Numerical Analysis* **2** (1965) 205

[20] P.A. BUSINGER and G.H. GOLUB, *Comm. ACM* **12** (1969) 564

[21] G.H. GOLUB and C. REINSCH, in *Linear Algebra* (J.H. WILKINSON and C. REINSCH, eds.), p. 134, Springer, Berlin 1971

[22] J.G.F. FRANCIS, *Computer Journal* **4** (1960) 265, 332

[23] F. JAMES and M. ROOS, *Nuclear Physics* **B172** (1980) 475

[24] M.L. SWARTZ, *Nuclear Instruments and Methods* **A294** (1966) 278

[25] V.H. REGENER, *Physical Review* **84** (1951) 161

[26] G. ZECH, *Nuclear Instruments and Methods* **A277** (1989) 608

[27] H. RUTISHAUSER, *Numerische Mathematik* **5** (1963) 48

[28] W. ROMBERG, *Det. Kong. Norske Videnskapers Selskap Forhandlinger* **28** (1955) Nr. 7

[29] K.S. KOELBIG, in *CERN Computer Centre Program Library*, Program D401, CERN, Geneva 1990

Bibliography

The following short list contains several books which can serve to extend the breadth and depth of the present material. It is of course in no way complete, but instead merely indicates a number of books of varied levels of difficulty.

Probability

L. BREIMAN, *Probability*, Addison-Wesley, Reading MA 1968

H. CRAMÈR, *The Elements of Probability Theory*, Wiley, New York 1955

B.V. GNEDENKO, *The Theory of Probability, 4th ed.*, Chelsea, New York 1967

KAI LAI CHUNG, *A Course in Probability Theory*, Harcourt, Brace and World, New York 1968

K. KRICKEBERG, *Wahrscheinlichkeitsrechnung*, Teubner, Stuttgart 1963

Mathematical Statistics

P.R. BEVINGTON, *Data Reduction and Error Analysis for the Physical Sciences*, McGraw-Hill, New York 1969

B.E. COOPER, *Statistics for Experimentalists*, Pergamon, Oxford 1969

G. COWAN, *Statistical Data Analysis*, Clarendon Press, Oxford 1998

H. CRAMÈR, *Mathematical Methods of Statistics*, University Press, Princeton 1946

W.J. DIXON and F.J. MASSEY, *Introduction to Statistical Analysis*, McGraw-Hill, New York 1969

D. DUMAS DE RAULY, *L'Estimation Statistique*, Gauthier-Villars, Paris 1968

W.T. EADIE, D. DRIJARD, F.E. JAMES, M. ROOS and B.SADOULET, *Statistical Methods in Experimental Physics*, North-Holland, Amsterdam 1971

W. FELLER, *An Introduction to Probability Theory and its Allpications*, 2 Vols., Wiley, New York 1968

M. FISZ, *Probability Theory and Mathematical Statistics*, Wiley, New York 1963

D.A.S. FRASER, *Statistics; An Introduction*, Wiley, New York 1958

H. FREEMAN, *Introduction to Statistical Inference*, Addison-Wesley, Reading, MA 1963

A.G. FRODENSEN and O. SKJEGGESTAD, *Probability and Statistics in Particle Physics*, Universitetsforlaget, Bergen 1979

P.G. HOEL, *Introduction to Mathematical Statistics, 4th ed.*, Wiley, New York 1971

M.G. KENDALL and A. STUART *The Advanced Theory of Statistics, 4th ed., 3 vols.*, Charles Griffin, London 1977

L. LYONS, *Statistics for Nuclear and Particle Physicists*, Cambridge University Press, Cambridge 1986

J. MANDEL, *The Statistical Analysis of Experimental Data*, Interscience, New York 1964

S.L. MEYER, *Data Analysis for Scientists and Engineers*, Wiley, New York 1975

L. SACHS, *Statistische Auswertungsmethoden, 5. Aufl.*, Springer, Berlin 1978

E. SVERDRUP, *Laws and Chance Variations*, 2 Bde., North-Holland, Amsterdam 1967

B.L. VAN DER WAERDEN, *Mathematische Statistik, 3. Aufl.*, Springer, Berlin 1971

S.S. WILKS, *Mathematical Statistics*, Wiley, New York 1962

T. YAMANE, *Elementary Sampling Theory*, Prentice-Hall, Englewood Cliffs, NJ 1967

T. YAMANE, *Statistics, An Introductory Analysis*, Harper and Row, New York 1967

Numerical Methods: Matrix Programs

A. BJÖRK and G. DAHLQUIST**,** *Numerische Methoden*, R. Oldenbourg Verlag, München 1972

R.P. BRENT, *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ 1973

C.T. FIKE, *Computer Evaluation of Mathematical Functions*, Prentice-Hall, Englewood Cliffs, NJ 1968

G.H. GOLUB and C.F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore 1983

R.W. HAMMING, *Numerical Methods for Engineers and Scientists, 2nd ed.*, McGraw-Hill, New York 1973

D.E. KNUTH, *The Art of Computer Programming*, 3 vols., Addison-Wesley, Reading MA 1968

C.L. LAWSON and R.J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs NJ 1974

W.H. PRESS, B.P. FLANNERY, S.A. TEUKOLSKY, and W.T. VETTERLING, *Numerical Recipes*, Cambridge University Press, Cambridge 1986

J.R. RICE, *Numerical Methods, Software and Analysis*, McGraw-Hill, New York 1983

J. STOER and R. BURLISCH**,** *Einführung in die Numerische Mathematik*, 2 Bde., Springer, Berlin 1983

J.H. WILKINSON and C. REINSCH, *Linear Algebra*, Springer, Berlin 1971

Collections of Formulas: Statistical Tables

M. ABRAMOWITZ and A. STEGUN, *Handbook of Mathematical Functions*, Dover, New York, 1965

R.A. FISHER and F. YATES, *Statistical Tables for Biological, Agricultural and Medical Research*, Oliver and Boyd, London 1957

U. GRAF and H.J. HENNING, *Formeln und Tabellen zur Mathematischen Statistik*, Springer, Berlin 1953

A. HALD, *Statistical Tables and Formulas*, Wiley, New York 1960

G.A. KORN and T.M. KORN, *Mathematical Handbook for Scientists and Engineers, 2nd ed.*, McGraw-Hill, New York 1968

D.V. LINDLEY and J.C.P. MILLER, *Cambridge Elementary Statistical Tables*, University Press, Cambridge 1961

D.B. OWEN, *Handbook of Statistical Tables*, Addison-Wesley, Reading MA 1962

E.S. PEARSON and H.O. HARTLEY, *Biometrica Tables for Statisticians*, University Press, Cambridge 1958

W. WETZEL, M.D. JÖHNK, and P. NAEVE, *Statistische Tabellen*, Walter de Gruyter, Berlin 1967

A. Matrix Calculations

The solution of the over-determined linear system of equations $A\mathbf{x} \approx \mathbf{b}$ is of central significance in data analysis. This can be solved in an optimal way with the singular value decomposition, first developed in the late 1960s. In this appendix the elementary definitions and calculation rules for matrices and vectors are summarized in Sects. A.1 and A.2. In Sect. A.3 orthogonal transformations are introduced, in particular the Givens and Householder transformations, which provide the key to the singular value decomposition.

After a few remarks on determinants (Sect. A.4) there follows in Sect. A.5 a discussion of various cases of matrix equations and a theorem on the orthogonal decomposition of an arbitrary matrix, which is of central importance in this regard. The classical procedure of normal equations, which, however, is inferior to the singular value decomposition, is described here.

Sections A.6–A.8 concern the particularly simple case of exactly determined, non-singular matrix equations. In this case the inverse matrix A^{-1} exists, and the solution to the problem $A\mathbf{x} = \mathbf{b}$ is $\mathbf{x} = A^{-1}\mathbf{b}$. Methods and programs for finding the solution are given. The important special case of a positive-definite symmetric matrix is treated in Sect. A.9.

In Sect. A.10 we define the pseudo-inverse matrix A^{+} of an arbitrary matrix A. After introducing eigenvectors and eigenvalues in Sect. A.11, the singular value decomposition is presented in Sects. A.12 and A.13. Computer routines are given in Sect. A.14. Modifications of the procedure and the consideration of constraints are the subject of Sects. A.15 through A.18.

It has been attempted to make the presentation illustrative rather than mathematically rigorous. Proofs in the text are only indicated in a general way, or are omitted entirely. As mentioned, the singular value decomposition is not yet widespread. An important goal of this appendix is to make its use possible. For readers with a basic knowledge of matrix calculations, the material covered in Sects. A.3, A.12, A.13, A.14.1, and A.18 is sufficient for this task. Sections A.14.2 through A.14.5 contain technical details on carrying out the singular value decomposition and can be omitted by hurried users.

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

All procedures described in this Appendix are implemented as methods of the classes DatanVector orDatanMatrix, respectively. Only in a few cases will we refer to these methods explicitly, in oder to establish the connection with some of the more complicated algorithms in the text.

A.1 Definitions: Simple Operations

By a *vector* in m dimensions (an m-vector) $\mathbf{a}$ we mean an m-tuple of real numbers, which are the *components* of $\mathbf{a}$. The arrangement of the components in the form

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \tag{A.1.1}$$

is called a *column vector*.

A $m \times n$ *matrix* is a rectangular arrangement of $m \times n$ numbers in m rows and n columns,

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix} \quad . \tag{A.1.2}$$

It can be viewed to be composed of n column vectors. By *transposition* of the $m \times n$ matrix A one obtains an $n \times m$ matrix A^{T} with elements

$$A^{\mathrm{T}}_{ik} = A_{ki} \quad . \tag{A.1.3}$$

Under transposition a column vector becomes a *row vector*,

$$\mathbf{a}^{\mathrm{T}} = (a_1, a_2, \ldots, a_m) \quad . \tag{A.1.4}$$

A column vector is an $m \times 1$ matrix; a row vector is a $1 \times m$ matrix.

For matrices one has the following elementary rules for *addition*, *subtraction* and *multiplication by a constant*

$$A \pm B = C \quad , \qquad C_{ik} = A_{ik} \pm B_{ik} \quad , \tag{A.1.5}$$

$$\alpha A = B \quad , \qquad B_{ik} = \alpha A_{ik} \quad . \tag{A.1.6}$$

The *product AB of two matrices* is only defined if the number of columns of the first matrix is equal to the number of rows of the second, e.g., $A = A_{m\times\ell}$ and $B = B_{\ell\times m}$. One has then

$$AB = C \quad , \qquad C_{ik} = \sum_{j=1}^{\ell} A_{ij} B_{jk} \quad . \tag{A.1.7}$$

Since

$$C_{ik}^{\mathrm{T}} = C_{ki} = \sum_{j=1}^{\ell} A_{kj} B_{ji} = \sum_{j=1}^{\ell} A_{jk}^{\mathrm{T}} B_{ij}^{\mathrm{T}} = \sum_{j=1}^{\ell} B_{ij}^{\mathrm{T}} A_{jk}^{\mathrm{T}} \quad ,$$

one has

$$C^{\mathrm{T}} = (AB)^{\mathrm{T}} = B^{\mathrm{T}} A^{\mathrm{T}} \quad . \tag{A.1.8}$$

With (A.1.7) one can also define the product of a row vector $\mathbf{a}^{\mathrm{T}}$ with a column vector $\mathbf{b}$, if both have the same number of elements m,

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^{\mathrm{T}} \mathbf{b} = c \quad , \qquad c = \sum_{j=1}^{m} a_j b_j \quad . \tag{A.1.9}$$

The result is then a number, i.e., a *scalar*. The product (A.1.9) is called the *scalar product*. It is usually written without indicating the transposition simply as $\mathbf{a} \cdot \mathbf{b}$. The vectors $\mathbf{a}$ and $\mathbf{b}$ are *orthogonal* to each other if their scalar product vanishes. Starting from (A.1.9) one obtains the following useful property of the matrix product (A.1.7). The element C_{ik}, which is located at the intersection of the ith row and the kth column of the product matrix C, is equal to the scalar product of the ith row of the first matrix A with the kth column of the second matrix B.

The *diagonal elements* of a matrix (A.1.2) are the elements A_{ii}. They form the *main diagonal* of the matrix A. If all of the non-diagonal elements vanish, $A_{ij} = 0, i \neq j$, then A is a *diagonal matrix*. An $n \times n$ diagonal matrix all of whose diagonal elements are unity is the n-dimensional *unit* matrix $I_n = I$,

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & 0 \\ & 1 & & \\ & & \ddots & \\ 0 & & & 1 \end{pmatrix} = I \quad . \tag{A.1.10}$$

The *null matrix* has only zeros as elements:

$$0 = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad . \tag{A.1.11}$$

A null matrix with only one column is the *null vector* $\mathbf{0}$.

We will now mention several more special types of square matrices.

A square matrix is *symmetric* if

$$A_{ik} = A_{ki} \quad . \tag{A.1.12}$$

If

$$A_{ik} = -A_{ki} \quad , \tag{A.1.13}$$

then the matrix is *antisymmetric*.

A *bidiagonal matrix* B possesses non-vanishing elements only on the main diagonal (b_{ii}) and on the parallel diagonal directly above it ($b_{i,i+1}$).

A *tridiagonal matrix* possesses in a addition non-vanishing elements directly below the main diagonal. A *lower triangular matrix* has non-vanishing elements only on and below the main diagonal, an *upper triangular matrix* only on and above the main diagonal.

The *Euclidian norm* or the *absolute value* of a vector is

$$|\mathbf{a}| = \|\mathbf{a}\|_2 = a = \sqrt{\mathbf{a}^{\mathrm{T}}\mathbf{a}} = \sqrt{\sum_j a_j^2} \quad . \tag{A.1.14}$$

A vector with unit norm is called a *unit vector*. We write this in the form

$$\widehat{\mathbf{a}} = \mathbf{a}/a \quad .$$

More general *vector norms* are

$$\begin{aligned} \|\mathbf{a}\|_p &= (\sum_j |a_j|^p)^{1/p} \quad , \qquad 1 \le p < \infty \quad , \qquad (A.1.15) \\ \|\mathbf{a}\|_\infty &= \max_j |a_j| \quad . \end{aligned}$$

For every vector norm $\|\mathbf{x}\|$ one defines a *matrix norm* $\|A\|$ as

$$\|A\| = \max_{\mathbf{x}\neq 0} \|A\mathbf{x}\|/\|\mathbf{x}\| \quad . \tag{A.1.16}$$

Matrix norms have the following properties:

$$\|A\| > 0 \quad , \quad A \neq 0 \quad ; \quad \|A\| = 0 \quad , \quad A = 0 \quad , \tag{A.1.17}$$

$$\|\alpha A\| = |\alpha| \|A\| \quad , \qquad \alpha \quad \text{real} \, , \tag{A.1.18}$$

$$\|A + B\| \le \|A\| + \|B\| \quad , \tag{A.1.19}$$

$$\|A\, B\| \le \|A\| \, \|B\| \quad . \tag{A.1.20}$$

A.2 Vector Space, Subspace, Rank of a Matrix

An *n-dimensional vector space* is the set of all n-dimensional vectors. If $\mathbf{u}$ and $\mathbf{v}$ are vectors in this space, then $\alpha\mathbf{u}$ and $\mathbf{u}+\mathbf{v}$ are also in the space, i.e., the vector space is *closed* under vector addition and under multiplication with a scalar α. The vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ are *linearly independent* if

$$\sum_{j=1}^{k} \alpha_j \mathbf{a}_j \neq 0 \tag{A.2.1}$$

for all α_j except for $\alpha_1 = \alpha_2 = \cdots = \alpha_k = 0$. Otherwise they are *linearly dependent*. The maximum number $k_{\max}$ of vectors that can be linearly independent is equal to the *dimension* of the vector space n. An arbitrary set of n linearly independent vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ forms a *basis* of the vector space. Any vector $\mathbf{a}$ can be expressed as a *linear combination* of the basis vectors,

$$\mathbf{a} = \sum_{j=1}^{n} \alpha_j \mathbf{a}_j \quad . \tag{A.2.2}$$

A special basis is

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad , \qquad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad , \qquad \ldots \quad , \qquad \mathbf{e}_n = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \quad . \tag{A.2.3}$$

These basis vectors are *orthonormal*, i.e.,

$$\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij} = \begin{cases} 1\,, & i = j \\ 0\,, & i \neq j \end{cases} \quad . \tag{A.2.4}$$

The component a_j of the vector $\mathbf{a}$ is the scalar product of $\mathbf{a}$ with the basis vector $\mathbf{e}_j$,

$$\mathbf{a} \cdot \mathbf{e}_j = a_j \quad , \tag{A.2.5}$$

cf. (A.1.1) and (A.1.9).

For $n \leq 3$ vectors can be visualized geometrically. A vector $\mathbf{a}$ can be represented as an arrow of length a. The basis vectors (A.2.3) are perpendicular to each other and are of unit length. The perpendicular projections of $\mathbf{a}$ onto the directions of the basis vectors are the components (A.2.5), as shown in Fig. A.1.

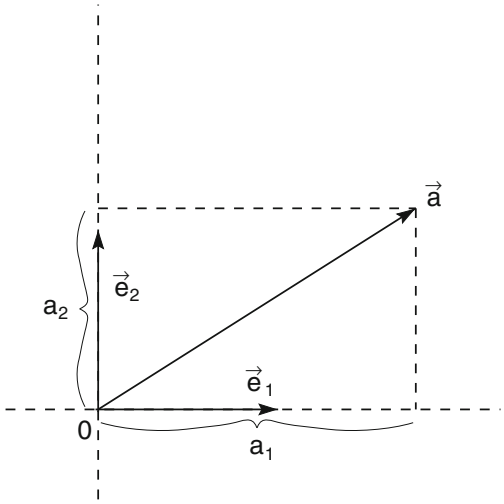

Fig. A.1: The vector **a** in the system of orthonormal basis vectors $\mathbf{e}_1, \mathbf{e}_2$.

A subset T of a vector space S is called a *subspace* if it is itself closed under vector addition and multiplication with a scalar. The greatest possible number of linearly independent vectors in T is the dimension of T. The product of an $m \times n$ matrix A with an n-vector $\mathbf{a}$ is an m-vector $\mathbf{b}$,

$$\mathbf{b} = A\mathbf{a} \quad . \tag{A.2.6}$$

The relation (A.2.6) can be regarded as a *mapping* or *transformation* of the vector $\mathbf{a}$ onto the vector $\mathbf{b}$.

The *span* of a set of vectors $\mathbf{a}_1, \ldots, \mathbf{a}_k$ is the vector space defined by the set of all linear combinations $\mathbf{u}$ of these vectors,

$$\mathbf{u} = \sum_{j=1}^{k} \alpha_j \mathbf{a}_j \quad . \tag{A.2.7}$$

It has a dimension $m \leq k$. The *column space* of an $m \times n$ matrix A is the span of the n column vectors of A; in this case the m-vectors $\mathbf{u}$ have the form $\mathbf{u} = A\mathbf{x}$ with arbitrary n-vectors $\mathbf{x}$. Clearly the dimension of the column space is $\leq \min(m, n)$. Similarly, the *row space* of A is the span of the m row vectors.

The *null space* or *kernel* of A consists of the set of vectors $\mathbf{x}$ for which

$$A\mathbf{x} = 0 \quad . \tag{A.2.8}$$

Column and row spaces of an $m \times n$ matrix have the same dimension. This is called the *rank* of the matrix. An $m \times n$ has *full rank* if

$$\mathrm{Rang}(A) = \min(m, n) \quad . \tag{A.2.9}$$

Otherwise it has *reduced rank*. An $n \times n$ matrix with $\mathrm{Rang}(A) < n$ is said to be *singular*.

A vector $\mathbf{a}$ is orthogonal to a subspace T if it is orthogonal to every vector $t \in T$. (A trivial example is $\mathbf{t} = t_1\mathbf{e}_1 + t_2\mathbf{e}_2$, $\mathbf{a} = a\mathbf{e}_3$, $\mathbf{a}\cdot\mathbf{t} = 0$.) A subspace U is orthogonal to the subspace T if for every pair of vectors $\mathbf{u} \in U$, $\mathbf{t} \in T$ one has $\mathbf{u}\cdot\mathbf{t} = 0$. The set of all vectors $\mathbf{u}+\mathbf{t}$ forms a vector space V, called the *direct sum* of T and U,

$$V = T \oplus U \quad . \tag{A.2.10}$$

Its dimension is

$$\dim(V) = \dim(T) + \dim(U) \quad . \tag{A.2.11}$$

If (A.2.10) holds, then T and U are subspaces of V. They are called *orthogonal complements*, $T = U^{\perp}$, $U = T^{\perp}$. If T is a subspace of S, then there always exists an orthogonal complement $T^{\perp}$, such that $S = T \oplus T^{\perp}$. Every vector $\mathbf{a} \in S$ can then be uniquely decomposed into the form $\mathbf{a} = \mathbf{t}+\mathbf{u}$ with $\mathbf{t} \in T$ and $\mathbf{u} \in T^{\perp}$. For the norms of the vectors the relation $a^2 = t^2 + u^2$ holds.

If T is an $(n-1)$-dimensional subspace of an n-dimensional vector space S and if $\mathbf{s}$ is a fixed vector in S, then the set of all vectors $\mathbf{h} = \mathbf{s}+\mathbf{t}$ with $\mathbf{t} \in T$ forms an $(n-1)$-dimensional *hyperplane* H in S. If H is given and $\mathbf{h}_0$ is an arbitrary fixed vector in H, then T is the set of all vectors $\mathbf{t} = \mathbf{h} - \mathbf{h}_0$, $\mathbf{h} \in H$, as shown in Fig. A.2. If $\widehat{\mathbf{u}}$ is a unit vector in the one-dimensional subspace $T = H^{\perp}$ orthogonal to H, then the scalar product

$$\widehat{\mathbf{u}}\cdot\mathbf{h} = d \tag{A.2.12}$$

has the same value for all $\mathbf{h} \in H$, where d is the distance of the hyperplane from the origin (see Fig. A.3).

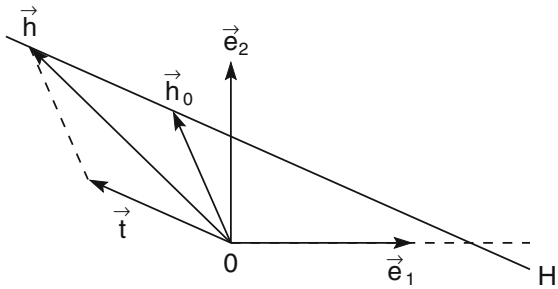

Fig. A.2: Hyperplane H in a two-dimensional vector space.

For a given $\widehat{\mathbf{u}}$ and d, Eq. (A.2.12) defines a hyperplane H. It divides the n-dimensional vector space into two *half spaces*, which consist of the set of vectors $\mathbf{x}$ for which $\widehat{\mathbf{u}}\cdot\mathbf{x} < 0$ and $\widehat{\mathbf{u}}\cdot\mathbf{x} > 0$.

A.3 Orthogonal Transformations

According to (A.2.6), the mapping of an n-vector $\mathbf{a}$ onto an n-vector $\mathbf{b}$ is performed by multiplication by a square $n \times n$ matrix, $\mathbf{b} = Q\mathbf{a}$. If the length of

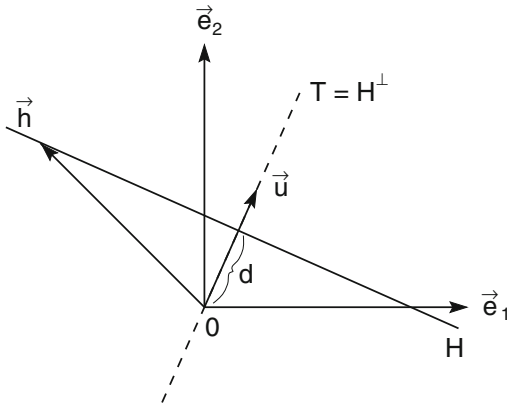

Fig. A.3: Hyperplane H and complementary one-dimensional vector space T.

the vector (A.1.14) remains unchanged, one speaks of an *orthogonal transformation*. For such a case one has $b = a$ or $b^2 = a^2$, i.e.,

$$\mathbf{b}^{\mathrm{T}}\mathbf{b} = \mathbf{a}^{\mathrm{T}} Q^{\mathrm{T}} Q\mathbf{a} = \mathbf{a}^{\mathrm{T}}\mathbf{a} \quad ,$$

and thus

$$Q^{\mathrm{T}} Q = I \quad . \tag{A.3.1}$$

A square matrix Q that fulfills (A.3.1) is said to be *orthogonal*.

It is clear that transformations are orthogonal when the transformed vector **b** is obtained from **a** by means of a spatial rotation and/or reflection. We will examine some orthogonal transformations that are important for the applications of this appendix.

A.3.1 Givens Transformation

The *Givens rotation* is a transformation that affects only components in a plane spanned by two orthogonal basis vectors. For simplicity we will first consider only two-dimensional vectors.

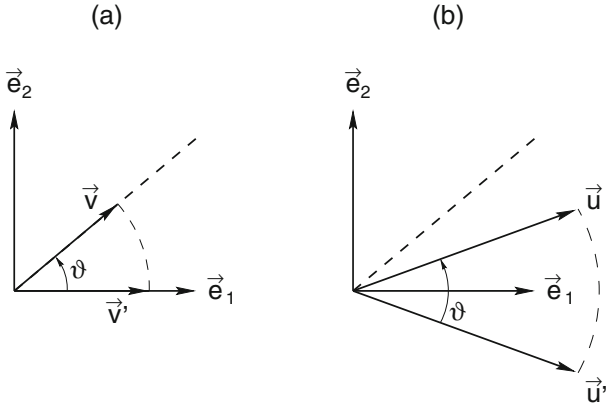

Fig. A.4: Application of the Givens transformation (**a**) to the vector **v** that defines the transformation and (**b**) to an arbitrary vector **u**.

A vector $\mathbf{v}$ can be represented as

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = v \begin{pmatrix} c \\ s \end{pmatrix} \quad , \qquad c = \cos\vartheta = v_1/v \quad , \qquad s = \sin\vartheta = v_2/v \quad . \tag{A.3.2}$$

A rotation by an angle $-\vartheta$ transforms the vector $\mathbf{v}$ into the vector $\mathbf{v}' = G\mathbf{v}$, whose second component vanishes, as in Fig. A.4a. Clearly one has

$$G = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \quad , \qquad \mathbf{v}' = G\mathbf{v} = \begin{pmatrix} v \\ 0 \end{pmatrix} \quad . \tag{A.3.3}$$

Of course the transformation thus defined with the vector $\mathbf{v}$ can also be applied to any other vector $\mathbf{u}$, as shown in Fig. A.4b. In n dimensions the Givens rotation leads to the transformation

$$\mathbf{v} \to \mathbf{v}' = G\mathbf{v} \quad ,$$

such that $v'_k = 0$. The components $v'_\ell = v_\ell$, $\ell \neq k$, $\ell \neq i$, remain unchanged and v'_i is determined such that the norm of the vector remains unchanged, $v' = v$. This is clearly given by

$$G = \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & & s & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & -s & & & c & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \leftarrow 1 \\ \\ \\ \leftarrow i \\ \\ \\ \\ \leftarrow k \\ \\ \\ \leftarrow n \end{matrix} \quad . \tag{A.3.4}$$

In practical applications, however, the full matrix is not needed. The method DatanMatrix.defineGivensTransformation defines a Givens transformation by the input of the two components v_1, v_2, DatanMatrix._ApplyGivensTransformation applies that transformation to two components of another vector. The method DatanMatrix.defineANDApplyGivensTransformation defines a transformation and directly applies it to the defining vector.

A.3.2 Householder Transformation

The Givens rotation is used to transform a vector in such a way that a given vector component vanishes. A more general transformation is the *Householder transformation*. If the original vector is

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \quad , \tag{A.3.5}$$

then for the transformed vector we want to have

$$\mathbf{v}' = H\mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_{p-1} \\ v'_p \\ v_{p+1} \\ \vdots \\ v_{\ell-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad . \tag{A.3.6}$$

That is, the components v'_ℓ, $v'_{\ell+1}$, …, v'_n should vanish. The remaining components should (with the exception of v'_p) remain unchanged. The component v'_p must be changed in such a way that one has $v = v'$. From the Pythagorean theorem in $n - \ell + 1$ dimensions one has

$$v_p'^2 = v_H^2 = v_p^2 + \sum_{i=\ell}^{n} v_i^2$$

or

$$v'_p = -\sigma\, v_H = -\sigma \sqrt{v_p^2 + \sum_{i=\ell}^{n} v_i^2} \tag{A.3.7}$$

with $\sigma = \pm 1$. We choose

$$\sigma = \mathrm{sign}(v_p) \quad . \tag{A.3.8}$$

We now construct the matrix H of (A.3.6). To do this we decompose the vector $\mathbf{v}$ into a sum,

$$\mathbf{v} = \mathbf{v}_H + \mathbf{v}_{H^\perp} \quad , \qquad \mathbf{v}_H = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ v_p \\ 0 \\ \vdots \\ 0 \\ v_\ell \\ \vdots \\ v_n \end{pmatrix} \quad , \qquad \mathbf{v}_{H^\perp} = \begin{pmatrix} v_1 \\ \vdots \\ v_{p-1} \\ 0 \\ v_{p+1} \\ \vdots \\ v_{\ell-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad . \tag{A.3.9}$$

Here the vector $\mathbf{v}_H$ is in the subspace spanned by the basis vectors $\mathbf{e}_p$, $\mathbf{e}_\ell$, $\mathbf{e}_{\ell+1}, \ldots, \mathbf{e}_n$, and $\mathbf{v}_{H^\perp}$ is in the subspace orthogonal to it. We now construct

$$\mathbf{u} = \mathbf{v}_H + \sigma\, v_H \mathbf{e}_p \tag{A.3.10}$$

and

$$H = I_n - \frac{2\mathbf{u}\mathbf{u}^{\mathrm{T}}}{u^2} \quad . \tag{A.3.11}$$

If we now decompose an arbitrary vector $\mathbf{a}$ into a sum of vectors parallel and perpendicular to $\mathbf{u}$,

$$\mathbf{a} = \mathbf{a}_\parallel + \mathbf{a}_\perp \quad ,$$

with

$$\mathbf{a}_\parallel = \frac{\mathbf{u}\mathbf{u}^{\mathrm{T}}}{u^2}\mathbf{a} = \frac{\mathbf{u}}{u^2}(\mathbf{u}\cdot\mathbf{a}) = \widehat{\mathbf{u}}(\widehat{\mathbf{u}}\cdot\mathbf{a}) \quad , \qquad \mathbf{a}_\perp = \mathbf{a} - \mathbf{a}_\parallel \quad ,$$

then one has

$$\mathbf{a}'_\parallel = H\mathbf{a}_\parallel = -\mathbf{a}_\parallel \quad , \qquad \mathbf{a}'_\perp = H\mathbf{a}_\perp = \mathbf{a}_\perp \quad .$$

Thus we see that the transformation is a reflection in the subspace that is orthogonal to the vector $\mathbf{u}$, as in Fig. A.5. One can easily verify that H in fact yields the transformation (A.3.6).

The transformation is uniquely determined by the vector $\mathbf{u}$. According to (A.3.10) one has for the components of this vector $u_p = v_p + \sigma\, v_H$, $u_\ell = v_\ell$,

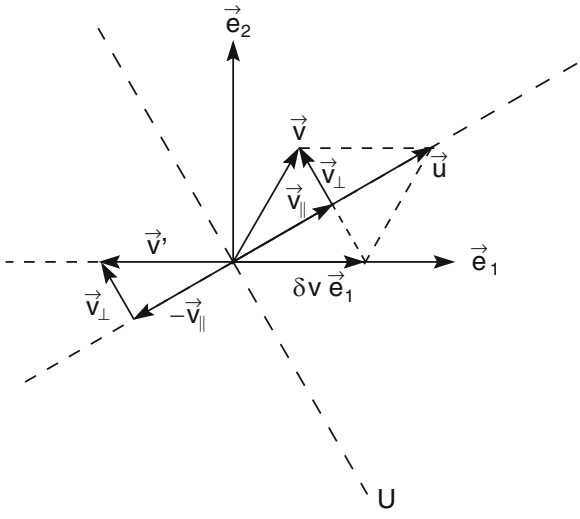

Fig. A.5: The vector $\mathbf{v}$ is mapped onto $\mathbf{v}'$ according to a Householder transformation such that $v'_2 = 0$. The mapping corresponds to a reflection in the subspace U that is orthogonal to the auxiliary vector $\mathbf{u}$.

$u_{\ell+1} = v_{\ell+1}$, ..., $u_n = v_n$, and $u_i = 0$ for all other i. If the vector $\mathbf{v}$ and the indices p and ℓ are given, then only u_p must be computed. The quantity u^2 appearing in (A.3.11) is then

$$\begin{aligned} u^2 &= u_p^2 + \sum_{i=\ell}^{n} u_i^2 = (v_p + \sigma v_H)^2 + \sum_{i=\ell}^{n} v_i^2 \\ &= v_p^2 + \sum_{i=\ell}^{n} v_i^2 + v_H^2 + 2\sigma v_H v_p \\ &= 2v_H^2 + 2\sigma v_H v_p = 2v_H(v_H + \sigma v_p) = 2v_H u_p \quad . \end{aligned}$$

We can thus write (A.3.11) in the form

$$H = I_n - b\mathbf{u}\mathbf{u}^{\mathrm{T}} \quad , \qquad b = (v_H u_p)^{-1} \quad . \tag{A.3.12}$$

The matrix H, however, is not needed explicitly to compute a transformed vector,

$$\mathbf{c}' = H\mathbf{c} \quad .$$

It is sufficient to know the vector $\mathbf{u}$ and the constant b. Since, however, $\mathbf{u}$ only differs from $\mathbf{v}$ in the element u_p (and in the vanishing elements), it is sufficient, starting from $\mathbf{v}$, to first compute the quantities u_p and b and when applying the transformation to use in addition the elements v_ℓ, $v_{\ell+1}$, ..., v_n. These are at the same time the corresponding elements of $\mathbf{u}$.

By the method DatanMatrix.defineHouseholderTranformation a transformation is defined; with DatanMatrix.applyHouseHolderTransformation it is applied to a vector.

A.3.3 Sign Inversion

If the diagonal element I_{ii} of the unit matrix is replaced by -1, then one obtains a symmetric orthogonal matrix,

$$R^{(i)} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & -1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \quad .$$

Applying this to the vector $\mathbf{a}$,

$$\mathbf{a}' = R^{(i)}\mathbf{a} \quad ,$$

changes the sign of the element a_i and leaves all the other elements unchanged. Clearly $R^{(i)}$ is a Householder matrix which produces a reflection in the subspace orthogonal to the basis vector $\mathbf{e}_i$. This can be seen immediately by substituting $\mathbf{u} = \mathbf{e}_i$ in (A.3.11).

A.3.4 Permutation Transformation

The $n \times n$ unit matrix (A.1.10) can easily be written as an arrangement of the basis vectors (A.2.3) in a square matrix,

$$I_n = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \end{pmatrix} \quad .$$

It is clearly an orthogonal matrix. The orthogonal transformation $I\mathbf{a} = \mathbf{a}$ leaves the vector $\mathbf{a}$ unchanged. If we now exchange two of the basis vectors $\mathbf{e}_i$ and $\mathbf{e}_k$, then we obtain the symmetric orthogonal matrix P^{ik}. As an example we show this for $n = 4$, $i = 2$, $k = 4$,

$$P^{(ik)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad .$$

The transformation $\mathbf{a}' = P^{(ik)}\mathbf{a}$ leads to an exchange of the elements a_i and a_k. All other elements remain unchanged:

$$\mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_k \\ \vdots \\ a_n \end{pmatrix} \quad , \qquad \mathbf{a}' = P^{(ik)}\mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_i \\ \vdots \\ a_n \end{pmatrix} \quad .$$

Multiplication of an $n \times m$ matrix A on the left with $P^{(ik)}$ permutes the lines i and k of A. Multiplication of an $m \times n$ matrix A from the right exchanges the columns i and k. If D is an $n \times n$ diagonal matrix, then the elements D_{ii} and D_{kk} are exchanged by the operation

$$D' = P^{(ik)} D \, P^{(ik)} \quad .$$

A.4 Determinants

To every $n \times n$ matrix A one can associate a number, its *determinant*, $\det A$. A determinant, like the corresponding matrix, is written as a square arrangement of the matrix elements, but is enclosed, however, by vertical lines. The determinants of orders two and three are defined by

$$\det A = \begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} = A_{11}A_{22} - A_{12}A_{21} \tag{A.4.1}$$

and

$$\begin{aligned} \det A &= \begin{vmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{vmatrix} \\ &= \begin{array}{rrrrr} & A_{11}A_{22}A_{33} & - & A_{11}A_{23}A_{32} \\ + & A_{12}A_{23}A_{31} & - & A_{12}A_{21}A_{33} \\ + & A_{13}A_{21}A_{32} & - & A_{13}A_{22}A_{31} \end{array} \end{aligned} \tag{A.4.2}$$

or, written in another way,

$$\begin{aligned} \det A &= \begin{array}{rrrrr} & A_{11}(A_{22}A_{33} & - & A_{23}A_{32}) \\ - & A_{12}(A_{21}A_{33} & - & A_{23}A_{31}) \\ + & A_{13}(A_{21}A_{32} & - & A_{22}A_{31}) \end{array} \quad . \end{aligned} \tag{A.4.3}$$

A general determinant of order n is written in the form

$$\det A = \begin{vmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \vdots & & & \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{vmatrix} \quad . \tag{A.4.4}$$

The *cofactor* $A^{\dagger}_{ij}$ of the element A_{ij} of a matrix is a determinant of order $(n-1)$, calculated from the matrix obtained by deleting the ith row and jth column of the original matrix, and multiplying by $(-1)^{i+j}$,

$$A^{\dagger}_{ij} = (-1)^{i+j} \begin{vmatrix} A_{11} & A_{12} & \dots & A_{1,j-1} & A_{1,j+1} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2,j-1} & A_{2,j+1} & \dots & A_{2n} \\ \vdots & & & & & & \\ A_{i-1,1} & A_{i-1,2} & \dots & A_{i-1,j-1} & A_{i-1,j+1} & \dots & A_{i-1,n} \\ A_{i+1,1} & A_{i+1,2} & \dots & A_{i+1,j-1} & A_{i+1,j+1} & \dots & A_{i+1,n} \\ \vdots & & & & & & \\ A_{n1} & A_{n2} & \dots & A_{n,j-1} & A_{n,j+1} & \dots & A_{nn} \end{vmatrix} \quad . \tag{A.4.5}$$

Determinants of higher order can be written as the sum of all elements of any row or column multiplied with the corresponding cofactors,

$$\det A = \sum_{k=1}^{n} A_{ik} A^{\dagger}_{ik} = \sum_{k=1}^{n} A_{kj} A^{\dagger}_{kj} \quad . \tag{A.4.6}$$

One can easily show that the result is independent of the choice of row i or column j. Equation (A.4.3) already shows that (A.4.6) is correct for $n = 3$. Determinants of arbitrary order can be computed by decomposing them according to their cofactors until one reaches, for example, the order two. A singular matrix, i.e., a square matrix whose rows or columns are not linearly independent, has *determinant zero*.

From A we can construct a further matrix by replacing each element ij by the cofactor of the element ji. In this way we obtain the *adjoint matrix* of A,

$$A^{\dagger} = \begin{pmatrix} A^{\dagger}_{11} & A^{\dagger}_{21} & \dots & A^{\dagger}_{n1} \\ A^{\dagger}_{12} & A^{\dagger}_{22} & \dots & A^{\dagger}_{n2} \\ \vdots & & & \\ A^{\dagger}_{1n} & A^{\dagger}_{2n} & \dots & A^{\dagger}_{nn} \end{pmatrix} \quad . \tag{A.4.7}$$

For determinants the following rules hold:

$$\det A = \det A^{\mathrm{T}} \quad , \tag{A.4.8}$$

$$\det AB = \det A \det B \quad . \tag{A.4.9}$$

For an orthogonal matrix Q one has $Q\,Q^{\mathrm{T}} = I$, i.e.,

$$\det I = 1 = \det Q \det Q$$

and thus

$$\det Q = \pm 1 \quad . \tag{A.4.10}$$

A.5 Matrix Equations: Least Squares

A system of m linear equations with n unknowns $x_1, x_2, \ldots, x_n$ has the general form

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n - b_1 = 0 \quad , \\ &a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n - b_2 = 0 \quad , \\ &\vdots \\ &a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n - b_m = 0 \end{aligned} \tag{A.5.1}$$

or in matrix notation

$$A\mathbf{x} - \mathbf{b} = 0 \quad . \tag{A.5.2}$$

In finding the solution to this equation $\mathbf{x}$ we must distinguish between various cases, which can be characterized by the values of m, n, and

$$k = \mathrm{Rang}(A) \quad .$$

The vector $A\mathbf{x}$ is in the column space of A, which is of dimension k. Since $\mathbf{b}$ is an m-vector, the equation can in general only be fulfilled if $k = m$, i.e., for $k = n = m$ and for $k = m < n$, since $k \leq \min(m, n)$. For $k = n = m$ one has n independent equations (A.5.1) with n unknowns, which have a *unique* solution. If $k = m < n$, then there are arbitrarily many n-vectors $\mathbf{x}$ that can be mapped on to the m-vector $A\mathbf{x} = \mathbf{b}$ such that (A.5.2) is fulfilled. The system of equations is *underdetermined*. The solution is not unique.

For $k = \mathrm{Rang}(A) \neq m$ there is, in general, no solution of (A.5.2). In this case we look for a vector $\widetilde{\mathbf{x}}$, for which the left-hand side of (A.5.2) is a vector of minimum Euclidian norm. That is, we replace Eq. (A.5.2) by

$$r^2 = (A\mathbf{x} - \mathbf{b})^2 = \min \quad , \tag{A.5.3}$$

i.e., we look for a vector $\widetilde{\mathbf{x}}$ for which the mapping $A\widetilde{\mathbf{x}}$ differs as little as possible from $\mathbf{b}$.

Given a m-vector $\mathbf{c}$, only for $k = \mathrm{Rang}(A) = n$ does there exist only one n-vector $\mathbf{x}$, such that $A\mathbf{x} = \mathbf{c}$. Therefore, only for the case $k = n$ is there a unique solution $\widetilde{\mathbf{x}}$. Thus there exists for $\mathrm{Rang}(A) = n$ and $m \geq n$ a *unique solution* $\widetilde{\mathbf{x}}$ of (A.5.3). For $n = m$ one has $r = 0$.

The relation (A.5.3) is also often written simply in the form

$$A\mathbf{x} - \mathbf{b} \approx 0 \tag{A.5.4}$$

or even, not entirely correctly, in the form (A.5.2). One calls the solution vector $\widetilde{\mathbf{x}}$ the *least-squares solution of* (A.5.4). In Table A.1 we list again the various cases that result from different relationships between m, n, and k.

Table A.1: Behavior of the solutions of (A.5.3) for various cases. m is the row number, n is the column number and k is the rank of the matrix A.

Case		Rang(A)	Residual	Solution unique
1a	$m = n$	$k = n$	$r = 0$	Yes
1b		$k < n$	$r \geq 0$	No
2a	$m > n$	$k = n$	$r \geq 0$	Yes
2b		$k < n$	$r \geq 0$	No
3a	$m < n$	$k = m$	$r = 0$	No
3b		$k < m$	$r \geq 0$	No

We now want to state more formally what we have determined in this section with respect to the solution of (A.5.4).

Theorem on the orthogonal decomposition of a matrix:
Every $m \times n$ matrix A of rank k can be written in the form

$$A = H\,R\,K^{\mathrm{T}} \quad , \tag{A.5.5}$$

where H is an $m \times m$ orthogonal matrix, K is an $n \times n$ orthogonal matrix, and R is an $m \times n$ matrix of the form

$$R = \begin{pmatrix} R_{11} & 0 \\ 0 & 0 \end{pmatrix} \tag{A.5.6}$$

and where R_{11} is a $k \times k$ matrix of rank k.

Substituting (A.5.5) into (A.5.4) and multiplying from the left by H^{T} leads to

$$R\,K^{\mathrm{T}}\mathbf{x} \approx H^{\mathrm{T}}\mathbf{b} \quad . \tag{A.5.7}$$

We define

$$H^{\mathrm{T}}\mathbf{b} = \mathbf{g} = \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{pmatrix} \begin{matrix} \}k \\ \}m-k \end{matrix} \tag{A.5.8}$$

and

$$K^{\mathrm{T}}\mathbf{x} = \mathbf{p} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix} \begin{matrix} \}k \\ \}n-k \end{matrix} \quad , \tag{A.5.9}$$

so that (A.5.7) takes on the form

$$R\mathbf{p} \approx \mathbf{g} \quad . \tag{A.5.10}$$

Because of (A.5.6), this breaks into two independent relations

$$R_{11}\mathbf{p}_1 = \mathbf{g}_1 \tag{A.5.11}$$

and

$$0 \cdot \mathbf{p}_2 \approx \mathbf{g}_2 \quad . \tag{A.5.12}$$

If $m = k$ and/or $n = k$, then the corresponding lower partial vectors are absent in (A.5.8) and/or (A.5.9) and the corresponding lower matrices are absent in (A.5.6). Since in (A.5.11) R_{11} is a $k \times k$ matrix of rank k, and $\mathbf{p}_1$ and $\mathbf{g}_1$ are k-vectors, there exists a solution vector $\widetilde{\mathbf{p}}_1$ for which equality holds. Because of the null matrix on the left-hand side of (A.5.12), we cannot derive any information about $\mathbf{p}_2$ from this relation.

Theorem on the solutions of $A\mathbf{x} \approx \mathbf{b}$: If $\widetilde{\mathbf{p}}_1$ is the unique solution vector of (A.5.11), then the following statements hold:

(i) All solutions of (A.5.3) have the form

$$\widehat{\mathbf{x}} = K \begin{pmatrix} \widetilde{\mathbf{p}}_1 \\ \widetilde{\mathbf{p}}_2 \end{pmatrix} \quad . \tag{A.5.13}$$

Here $\widetilde{\mathbf{p}}_2$ is an arbitrary $(n-k)$-vector, i.e., the solution is unique for $k = n$. There is always, however, a *unique solution of minimum absolute value*:

$$\widetilde{\mathbf{x}} = K \begin{pmatrix} \widetilde{\mathbf{p}}_1 \\ 0 \end{pmatrix} \quad . \tag{A.5.14}$$

(ii) All solutions $\widehat{\mathbf{x}}$ have the same residual vector

$$\mathbf{r} = \mathbf{b} - A\widehat{\mathbf{x}} = H \begin{pmatrix} 0 \\ \mathbf{g}_2 \end{pmatrix} \tag{A.5.15}$$

with the absolute value $r = g_2 = |\mathbf{g}_2|$. The residual vanishes for $k = m$.

The problem $A\mathbf{x} \approx \mathbf{b}$ should always be handled with orthogonal decompositions, preferably with the singular value decomposition and singular value analysis described in Sects. A.12 and A.13. Numerically the results are at least as accurate as with other methods, and are often more accurate (cf. Sect. A.13, Example A.4).

Nevertheless we will briefly present as well the method of *normal equations*. The method is very transparent compared to the orthogonal decomposition and is therefore always described in textbooks.

We consider the square (A.5.3) of the residual vector

$$\begin{aligned} r^2 &= (A\mathbf{x}-\mathbf{b})^{\mathrm{T}}(A\mathbf{x}-\mathbf{b}) = \mathbf{x}^{\mathrm{T}}A^{\mathrm{T}}A\mathbf{x} - 2\mathbf{b}^{\mathrm{T}}A\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{b} \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{\ell=1}^{n} A_{ij}A_{i\ell}x_j x_\ell - 2\sum_{i=1}^{m}\sum_{j=1}^{n} b_i A_{ij} x_j + \sum_{i=1}^{m} b_i^2 \quad . \end{aligned}$$

The requirement $r^2 = \min$ leads to

$$\frac{\partial r^2}{\partial x_k} = 2\sum_{i=1}^{m}\sum_{\ell=1}^{n} A_{ik}A_{i\ell}x_\ell - 2\sum_{i=1}^{m} b_i A_{ik} = 0 \quad , \qquad k = 1,\ldots,n \quad .$$

These n linear equations are called normal equations. They can be arranged in a matrix,

$$A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b} \quad . \tag{A.5.16}$$

This is a system of n equations with n unknowns. If the equations are linearly independent, then the $n \times n$ matrix $(A^{\mathrm{T}}A)$ is of full rank n; it is not singular. According to Sect. A.6 there then exists an inverse $(A^{\mathrm{T}}A)^{-1}$, such that $(A^{\mathrm{T}}A)^{-1}(A^{\mathrm{T}}A) = I$. Thus the desired solution to (A.5.3) is

$$\widetilde{\mathbf{x}} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}\mathbf{b} \quad . \tag{A.5.17}$$

This simple prescription is, however, useless if $A^{\mathrm{T}}A$ is singular or nearly singular (cf. Example A.4).

A.6 Inverse Matrix

For every non-singular $n \times n$ matrix A, the *inverse matrix* A^{-1} is defined by

$$A\,A^{-1} = I_n = A^{-1}\,A \quad . \tag{A.6.1}$$

It is also an $n \times n$ matrix.

If A^{-1} is known, then the solution of the matrix equation

$$A\mathbf{x} = \mathbf{b} \tag{A.6.2}$$

is given simply by

$$\mathbf{x} = A^{-1}\mathbf{b} \quad . \tag{A.6.3}$$

As we will show, A^{-1} only exists for non-singular square matrices, so that (A.6.3) only gives the solution for the case 1a of Table A.1.

In order to determine A^{-1}, we set $A^{-1} = X$ and express the column vectors of X as $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$. Equation (A.6.1) is then decomposed into n equations:

$$A\mathbf{x}_i = \mathbf{e}_i \quad , \qquad i = 1, 2, \ldots, n \quad . \tag{A.6.4}$$

The right-hand sides are the basis vectors (A.2.3). For the case $n = 2$ we can write the system (A.6.4) in various equivalent ways, e.g.,

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} X_{11} \\ X_{21} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad , \quad \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} X_{12} \\ X_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad , \tag{A.6.5}$$

or alternatively

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad ,$$

or as a system of four equations with four unknowns,

$$\begin{aligned} A_{11}X_{11} + A_{12}X_{21} &= 1 \quad , \\ A_{21}X_{11} + A_{22}X_{21} &= 0 \quad , \\ A_{11}X_{12} + A_{12}X_{22} &= 0 \quad , \\ A_{21}X_{12} + A_{22}X_{22} &= 1 \quad . \end{aligned} \tag{A.6.6}$$

By elimination and substitution one easily finds

$$\begin{aligned} X_{11} &= \frac{A_{22}}{A_{11}A_{22} - A_{12}A_{21}} \quad , \\ X_{12} &= \frac{-A_{12}}{A_{11}A_{22} - A_{12}A_{21}} \quad , \\ X_{21} &= \frac{-A_{21}}{A_{11}A_{22} - A_{12}A_{21}} \quad , \\ X_{22} &= \frac{A_{11}}{A_{11}A_{22} - A_{12}A_{21}} \end{aligned} \tag{A.6.7}$$

or in matrix notation

$$X = A^{-1} = \frac{1}{\det A} \begin{pmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{pmatrix} \quad . \tag{A.6.8}$$

The matrix on the right-hand side is the adjoint of the original matrix, i.e.,

$$A^{-1} = \frac{A^\dagger}{\det A} \quad . \tag{A.6.9}$$

One can show that this relation holds for square matrices of arbitrary order. From (A.6.9) it is clear that the inverse of a singular matrix, i.e., of a matrix with vanishing determinant, is not determined.

In practice the inverse matrix is not computed using (A.6.9) but rather as in our example of the 2×2 matrix by elimination and substitution from the system (A.6.4). This system consists of n sets of equations of the form

$$A\mathbf{x} = \mathbf{b} \quad , \tag{A.6.10}$$

each consisting of n equations with n unknowns. Here A is a non-singular $n \times n$ matrix. We will first give the solution algorithm for square non-singular matrices A, but we will later remove these restrictions.

A.7 Gaussian Elimination

We will write Eq. (A.6.10) for an $n \times n$ matrix A in components,

$$\begin{aligned} &A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n = b_1 \quad , \\ &A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n = b_2 \quad , \\ &\vdots \\ &A_{n1}x_1 + A_{n2}x_2 + \cdots + A_{nn}x_n = b_n \quad , \end{aligned} \tag{A.7.1}$$

and we will solve the system by *Gaussian elimination*. For this we define $n-1$ multipliers

$$m_{i1} = \frac{A_{i1}}{A_{11}} \quad , \qquad i = 2, 3, \ldots, n \quad , \tag{A.7.2}$$

multiply the first equation by m_{21}, and subtract it from the second. We then multiply the first equation by m_{31} and subtract it from the third, and so forth. We obtain the system

$$\begin{aligned} A_{11}^{(1)}x_1 \quad + \quad & A_{12}^{(1)}x_2 + \cdots + A_{1n}^{(1)}x_n = b_1^{(1)} \quad , \\ & A_{22}^{(2)}x_2 + \cdots + A_{2n}^{(2)}x_n = b_2^{(2)} \quad , \\ & \vdots \\ & A_{n2}^{(2)}x_2 + \cdots + A_{nn}^{(2)}x_n = b_n^{(2)} \quad , \end{aligned} \tag{A.7.3}$$

where the unknown x_1 has disappeared from all the equations except the first. The coefficients $A_{ij}^{(2)}$, $b_i^{(2)}$ are given by the equations

$$\begin{aligned} A_{ij}^{(2)} &= A_{ij}^{(1)} - m_{i1}A_{1j}^{(1)} \quad , \\ b_i^{(2)} &= b_i^{(1)} - m_{i1}b_1^{(1)} \quad . \end{aligned}$$

The procedure is then repeated with the last $n-1$ equations by defining

$$m_{i2} = \frac{A_{i2}^{(2)}}{A_{22}^{(2)}} \quad , \qquad i = 3, 4, \ldots, n \quad ,$$

multiplying the second equation of the system (A.7.3) with the corresponding m_{i2}, and then subtracting it from the third, fourth, ..., nth equation. In the kth step of the procedure, the multipliers

$$m_{ik} = \frac{A_{ik}^{(k)}}{A_{kk}^{(k)}} \quad , \qquad i = k+1, k+2, \ldots, n \quad , \tag{A.7.4}$$

are used and the new coefficients

$$\begin{aligned} A_{ij}^{(k+1)} &= A_{ij}^{(k)} - m_{ik} A_{kj}^{(k)} \quad , \\ b_i^{(k+1)} &= b_i^{(k)} - m_{ik} b_k^{(k)} \end{aligned} \tag{A.7.5}$$

are computed. After $n-1$ steps we have produced the following *triangular system of equations*:

$$\begin{aligned} A_{11}^{(1)} x_1 + \; A_{12}^{(1)} x_2 + \cdots + \; A_{1n}^{(1)} x_n &= b_1^{(1)} \quad , \\ A_{22}^{(2)} x_2 + \cdots + \; A_{2n}^{(2)} x_n &= b_2^{(2)} \quad , \\ \vdots \qquad\qquad\quad \vdots \qquad &\quad \vdots \\ A_{nn}^{(n)} x_n &= b_n^{(n)} \quad . \end{aligned} \tag{A.7.6}$$

The last equation contains only x_n. By substituting into the next higher equation one obtains x_{n-1}, i.e., in general

$$x_i = \frac{1}{A_{ii}^{(i)}} \left\{ b_i^{(i)} - \sum_{\ell=i+1}^{n} A_{i\ell}^{(i)} x_\ell \right\} \quad , \qquad i = n, n-1, \ldots, 1 \quad . \tag{A.7.7}$$

One should note that the change of the elements of the matrix A does not depend on the right-hand side $\mathbf{b}$. Therefore one can reduce several systems of equations with different right-hand sides simultaneously. That is, instead of $A\mathbf{x} = \mathbf{b}$, one can solve the more general system

$$A\,X = B \tag{A.7.8}$$

at the same time, where B and X are $n \times m$ matrices, the matrix X being unknown.

Example A.1: Inversion of a 3×3 matrix

As a numerical example let us consider the inversion of a 3×3 matrix, i.e., $B = I$. The individual computations for the example

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & -2 \\ 1 & 1 & 2 \end{pmatrix} X = I$$

are shown in Table A.2. The result is

$$X = \frac{1}{5} \begin{pmatrix} -4 & 1 & 7 \\ 6 & 1 & -8 \\ -1 & -1 & 3 \end{pmatrix} \quad . \blacksquare$$

Following the individual steps of the calculation one sees that division is carried out in two places, namely in Eqs. (A.7.4) and (A.7.7). The denominator is in both cases a coefficient

$$A_{ii}^{(i)} \quad , \qquad i = 1, 2, \ldots, n-1 \quad ,$$

i.e., the upper left-hand coefficient of the system, the so-called *pivot* for the step $i - 1$ of the reduction process. Our procedure fails if this coefficient is equal to zero. In such a case one can simply exchange the ith line of the system with some other lower line whose first coefficient is not zero. The system of equations itself is clearly not changed by exchanging two equations. The procedure still fails if all of the coefficients of a column vanish. In this case the matrix A is singular, and there is no solution. In practice (at least when using computers where the extra work is negligible) it is advantageous to always carry out a reshuffling so that the pivot is the coefficient with the largest absolute value among those in the first column of the reduced system. One then always has the largest possible denominator. In this way rounding errors are kept as small as possible. The procedure just described is called *Gaussian elimination with pivoting*.

Following it, the method DatanMatrix.matrixEquation solves Eq. (A.7.8). In the same way the method DatanMatrix.inverse determines the inverse of a square nonsingular matrix.

A.8 LR-Decomposition

For the elements $A_{ij}^{(k)}$ transformed by Gaussian elimination [cf. (A.7.6)], one has on and above the main diagonal

$$A_{ij}^{(n)} = A_{ij}^{(n-1)} = \cdots = A_{ij}^{(i)} \quad , \qquad i \leq j \quad , \tag{A.8.1}$$

Table A.2: Application of Gaussian elimination to Example A.1.

Reduction

	Matrix A			Matrix B			Multiplier
	1	2	3	1	0	0	–
Step 0	2	1	−2	0	1	0	2
	1	1	2	0	0	1	1
		−3	−8	−2	1	0	–
Step 1		−1	−1	−1	0	1	$\frac{1}{3}$
Step 2			$\frac{5}{3}$	$-\frac{1}{3}$	$-\frac{1}{3}$	1	–

Substitution

	$j=1$
x_{3j}	$-\frac{1}{5}$
x_{2j}	$-\frac{1}{3}(-2-8\times\frac{1}{5})=\frac{6}{5}$
x_{1j}	$1-2\times\frac{6}{5}+3\times\frac{1}{5}=-\frac{4}{5}$
	$j=2$
x_{3j}	$-\frac{1}{5}$
x_{2j}	$-\frac{1}{3}(1-\frac{8}{5})=\frac{1}{5}$
x_{1j}	$-2\times\frac{1}{5}+3\times\frac{1}{5}=\frac{1}{5}$
	$j=3$
x_{3j}	$\frac{3}{5}$
x_{2j}	$-\frac{1}{3}(0+8\times\frac{3}{5})=-\frac{8}{5}$
x_{1j}	$2\times\frac{8}{5}-3\times\frac{3}{5}=\frac{7}{5}$

and below the main diagonal

$$A_{ij}^{(n)} = A_{ij}^{(n-1)} = \cdots = A_{ij}^{(j+1)} = 0 \quad , \qquad i > j \quad . \tag{A.8.2}$$

The transformation (A.7.5) is thus only carried out for $k = 1, 2, \ldots, r$ with $r = \min(i-1,\ j)$,

$$A_{ij}^{(k+1)} = A_{ij}^{(k)} - m_{ik} A_{kj}^{(k)} \quad . \tag{A.8.3}$$

Summation over k gives

$$\sum_{k=1}^{r} A_{ij}^{(k+1)} - \sum_{k=1}^{r} A_{ij}^{(k)} = A_{ij}^{(r+1)} - A_{ij} = -\sum_{k=1}^{r} m_{ik} A_{kj}^{(k)} \quad .$$

Thus the elements $A_{ij}^{(1)} = A_{ij}$ of the original matrix A can be written in the form

$$\begin{aligned} A_{ij} &= A_{ij}^{(i)} + \textstyle\sum_{k=1}^{i-1} m_{ik} A_{kj}^{(k)} \quad , \qquad i \le j \quad , \\ A_{ij} &= 0 + \textstyle\sum_{k=1}^{j} m_{ik} A_{kj}^{(k)} \quad , \qquad i > j \quad . \end{aligned} \tag{A.8.4}$$

Noting that the multipliers m_{ik} have only been defined up to now for $i > k$, we can in addition set

$$m_{ii} = 1 \quad , \qquad i = 1, 2, \ldots, n \quad ,$$

and reduce the two relations (A.8.4) to one,

$$A_{ij} = \sum_{k=1}^{p} m_{ik} A_{kj}^{(k)} \quad , \qquad 1 \le i,\ j \le n \quad , \qquad p = \min(i, j) \quad .$$

The equation shows directly that the matrix A can be represented as the product of two matrices L and R. Here L is a lower and R an upper triangular matrix,

$$A = L\,R \quad , \tag{A.8.5}$$

$$L = \begin{pmatrix} m_{11} & & & \\ m_{21} & m_{22} & & \\ \vdots & & & \\ m_{n1} & m_{n2} & \ldots & m_{nn} \end{pmatrix} \quad , \qquad m_{ii} = 1 \quad ,$$

$$R = \begin{pmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ & r_{22} & \ldots & r_{2n} \\ & & & \vdots \\ & & & r_{nn} \end{pmatrix} = \begin{pmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \ldots & A_{1n}^{(1)} \\ & A_{22}^{(2)} & \ldots & A_{2n}^{(2)} \\ & & & \vdots \\ & & & A_{nn}^{(n)} \end{pmatrix} \quad .$$

The original system of equations (A.6.10)

$$A\,\mathbf{x} = L\,R\,\mathbf{x} = \mathbf{b}$$

is thus equivalent to two triangular systems of equations,

$$L\,\mathbf{y} = \mathbf{b} \quad , \qquad R\,\mathbf{x} = \mathbf{y} \quad .$$

Instead of resorting to the formulas from Sect. A.7, we can compute the elements of L and R directly from (A.8.5). We obtain

$$\left.\begin{array}{l} r_{kj} = A_{kj} - \sum_{\ell=1}^{k-1} m_{k\ell} r_{\ell j}\,, \quad j = k, k+1, \ldots, n \\ m_{ik} = \left(A_{ik} - \sum_{\ell=1}^{k-1} m_{i\ell} r_{\ell k}\right)/r_{kk}\,, \quad i = k+1, \ldots, n \end{array}\right\} \; k = 1, \ldots, n \quad , \tag{A.8.6}$$

i.e., for $k = 1$ one computes the first row of R (which is equal to the first row of A) and the first column of L; for $k = 2$ one computes the second row of R and the second column of L. In a computer program the elements of R and L can overwrite the original matrix A with the same indices, since when computing r_{kj} one only needs the element A_{kj} and other elements of R and L that have already been computed according to (A.8.6). The corresponding consideration holds for the computation of m_{ik}.

The algorithm (A.8.6) is called the Doolittle *LR-decomposition*. By including pivoting this is clearly equivalent to Gaussian elimination.

A.9 Cholesky Decomposition

If A is a real symmetric positive-definite matrix (cf. Sect. A.11), then it can be uniquely expressed as

$$A = U^{\mathrm{T}}\,U \quad . \tag{A.9.1}$$

Here U is a real upper triangular matrix with positive diagonal elements. The $n \times n$ matrix A has the required property for the validity of (A.9.1), in particular in those cases where it is equal to the product of an arbitrary real $n \times m$ matrix B (with $m \geq n$ and full rank) with its transpose B^{T}, $A = B\,B^{\mathrm{T}}$.

To determine U we first carry out the Doolittle LR-decomposition, define a diagonal matrix D whose diagonal elements are equal to those of R,

$$D = \mathrm{diag}(r_{11}, r_{22}, \ldots, r_{nn}) \quad , \tag{A.9.2}$$

and introduce the matrices

$$D^{-1} = \mathrm{diag}(r_{11}^{-1}, r_{22}^{-1}, \ldots, r_{nn}^{-1}) \tag{A.9.3}$$

and

$$D^{-1/2} = \mathrm{diag}(r_{11}^{-1/2}, r_{22}^{-1/2}, \ldots, r_{nn}^{-1/2}) \quad . \tag{A.9.4}$$

We can then write

$$A = L\,R = L\,D\,D^{-1}\,R = L\,D\,R' \quad , \qquad R' = D^{-1}\,R \quad . \tag{A.9.5}$$

Because of the assumed symmetry one has

$$A = A^{\mathrm{T}} = (R')^{\mathrm{T}} D\,L^{\mathrm{T}} \quad . \tag{A.9.6}$$

Comparing with (A.9.5) gives $L^{\mathrm{T}} = R'$, i.e.,

$$L = R^{\mathrm{T}}\,D^{-1} \quad .$$

With

$$U = D^{-1/2} R = D^{1/2} L^{\mathrm{T}} \tag{A.9.7}$$

Eq. (A.9.1) is indeed fulfilled. The relation (A.9.7) means for the elements u_{ij} of U

$$u_{ij} = r_{ii}^{-1/2} r_{ij} = r_{ii}^{1/2} m_{ji} \quad .$$

Thus the values r and m can be eliminated in favor of u from (A.8.6), and we obtain with

$$\left.\begin{aligned} u_{kk} &= \left(A_{kk} - \textstyle\sum_{\ell=1}^{k-1} u_{\ell k}^2\right)^{1/2} \\ u_{kj} &= \left(A_{kj} - \textstyle\sum_{\ell=1}^{k-1} u_{\ell k} u_{\ell j}\right)/u_{kk}\,, \quad j = k+1, \ldots, n \end{aligned}\right\} k = 1, \ldots, n$$

the algorithm for the *Cholesky decomposition* of a real positive-definite symmetric matrix A. Since for such matrices all of the diagonal elements are different from zero, one does not need in principle any pivoting. One can also show that it would bring no advantage in numerical accuracy. The method DatanMatrix.choleskyDecomposition performs the Cholesky decomposition of a symmetric positive definite square matrix.

Positive-definite symmetric matrices play an important role as weight and covariance matrices. In such cases one often requires the Cholesky decomposition $A = U^{\mathrm{T}}U$ as well as multiplication of U by a matrix. It is easier computationally if this multiplication is done by the method DatanMatrix.choleskyMultiply, which takes the triangular form of U into account.

Of particular interest is the inversion of a symmetric positive-definite $n \times n$ matrix A. For this we will solve the n matrix equations (A.6.4) with the previously described Cholesky decomposition of A,

$$A\mathbf{x}_i = U^{\mathrm{T}} U \mathbf{x}_i = U^{\mathrm{T}} \mathbf{y}_i = \mathbf{e}_i \quad , \qquad i = 1, \ldots, n \quad . \tag{A.9.8}$$

We denote the ℓth component of the three vectors $\mathbf{x}_i$, $\mathbf{y}_i$, and $\mathbf{e}_i$ by $x_{i\ell}$, $y_{i\ell}$, and $e_{i\ell}$. Clearly one has $x_{i\ell} = (A^{-1})_{i\ell}$ and $e_{i\ell} = \delta_{i\ell}$, since $e_{i\ell}$ is the element (i, ℓ) of the unit matrix.

We now determine $y_{i\ell}$ by means of *forward substitution*. From (A.9.8) it follows directly that

$$\sum_{k=1}^{\ell} U_{k\ell} y_{ik} = e_{i\ell} = \delta_{i\ell} \quad ,$$

i.e.,

$$\begin{aligned} U_{11} y_{i1} &= \delta_{i1} \quad , \\ U_{12} y_{i1} + U_{22} y_{i2} &= \delta_{i2} \quad , \\ &\vdots \end{aligned}$$

and thus

$$\begin{aligned} y_{i1} &= \delta_{i1} / U_{11} \quad , \\ &\vdots \\ y_{i\ell} &= \frac{1}{U_{\ell\ell}} \left(\delta_{i\ell} - \sum_{k=1}^{\ell-1} U_{k\ell} y_{ik} \right) \quad . \end{aligned}$$

Since, however, $\delta_{i\ell} = 0$ for $i \neq \ell$, this expression simplifies to

$$\begin{aligned} y_{i\ell} &= 0 \quad , \qquad \ell < i \quad , \\ y_{i\ell} &= \frac{1}{U_{\ell\ell}} \left(\delta_{i\ell} - \sum_{k=i}^{\ell-1} U_{k\ell} y_{ik} \right) \quad , \qquad \ell \geq i \quad . \end{aligned}$$

We can now obtain $x_{i\ell}$ by means of *backward substitution* of $y_{i\ell}$ into

$$U \mathbf{x}_i = \mathbf{y}_i$$

or, written in components,

$$\sum_{k=\ell}^{n} U_{\ell k} x_{ik} = y_{i\ell}$$

or

$$\begin{aligned} U_{11} x_{i1} + U_{12} x_{i2} + \cdots + U_{1n} x_{in} &= y_{i1} \quad , \\ U_{22} x_{i2} + \cdots + U_{2n} x_{in} &= y_{i2} \quad , \\ &\vdots \\ U_{nn} x_{in} &= y_{in} \quad . \end{aligned}$$

We obtain

$$\begin{aligned} x_{in} &= y_{in}/U_{nn} \quad , \\ &\vdots \\ x_{i\ell} &= \frac{1}{U_{\ell\ell}}\left(y_{i\ell} - \sum_{k=\ell+1}^{n} U_{\ell k} x_{ik}\right) \quad . \end{aligned}$$

If we compute $x_{i\ell}$ only for $\ell \geq i$, then by backward substitution one only encounters the elements $y_{i\ell}$ for $\ell \geq i$, i.e., only non-vanishing $y_{i\ell}$. The vanishing elements $y_{i\ell}$ thus do not need to be stored. The elements $x_{i\ell}$ for $\ell < i$ follow simply from the symmetry of the original matrix, $x_{i\ell} = x_{\ell i}$.

The method DatanMatrix.choleskyInversion first performs the Cholesky decomposition of A. Then a loop is carried out over the various right-hand sides e_i, $i = 1, \ldots, n$, of (A.9.8). As the result of running through the loop once one obtains the elements x_{in}, $x_{i,n-1}$, ..., x_{ii} of row i. They are stored as the corresponding elements of the output matrix A. The elements of the vector $\mathbf{y}$, which is only used for intermediate results, can be stored in the last row of A. Finally the elements below the main diagonal are filled by copies of their mirror images.

A.10 Pseudo-inverse Matrix

We now return to the problem of Sect. A.5, that of solving the Eq. (A.5.4)

$$A\mathbf{x} \approx \mathbf{b} \tag{A.10.1}$$

for an arbitrary $m \times n$ matrix A of rank k. According to (A.5.14), the unique solution of minimum norm is

$$\widetilde{\mathbf{x}} = K \begin{pmatrix} \mathbf{p}_1 \\ 0 \end{pmatrix} \quad .$$

The vector $\mathbf{p}_1$ is the solution of Eq. (A.5.11) and therefore

$$\widetilde{\mathbf{p}}_1 = R_{11}^{-1}\mathbf{g}_1 \quad ,$$

since R_{11} is non-singular. Because of (A.5.8) one has finally

$$\widetilde{\mathbf{x}} = K \begin{pmatrix} R_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} H^{\mathrm{T}}\mathbf{b} \quad . \tag{A.10.2}$$

In analogy to (A.6.3) we write

$$\widetilde{\mathbf{x}} = A^{+}\mathbf{b} \tag{A.10.3}$$

and call the $n \times m$ matrix

$$A^+ = K \begin{pmatrix} R_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} H^{\mathrm{T}} \tag{A.10.4}$$

the *pseudo-inverse* of the $m \times n$ matrix A.

The matrix A^+ is uniquely determined by A and does not depend on the particular orthogonal decomposition (A.5.5). This can easily be seen if one denotes the jth column vector of A^+ by $\mathbf{a}_j^+ = A^+ \mathbf{e}_j$, with $\mathbf{e}_j$ the jth column vector of the m-dimensional unit matrix. According to (A.10.3) the vector $\mathbf{a}_j^+$ is the minimum-length solution of the equation $A\mathbf{a}_j^+ = \mathbf{e}_j$, and is therefore unique.

A.11 Eigenvalues and Eigenvectors

We now consider the *eigenvalue equation*

$$G\mathbf{x} = \lambda \mathbf{x} \tag{A.11.1}$$

for the $n \times n$ matrix G. If this is fulfilled, then the scalar λ is called the *eigenvalue* and the n-vector $\mathbf{x}$ the *eigenvector* of G. Clearly the eigenvector $\mathbf{x}$ is only determined up to an arbitrary factor. One can choose this factor such that $|\mathbf{x}| = 1$.

We consider first the particularly simple case where G is a diagonal matrix with non-negative diagonal elements,

$$G = S^{\mathrm{T}} S = S^2 = \begin{pmatrix} s_1^2 & & & \\ & s_2^2 & & \\ & & \ddots & \\ & & & s_n^2 \end{pmatrix} , \tag{A.11.2}$$

which can be expressed as the product of an arbitrary diagonal matrix S with itself,

$$S = \begin{pmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_n \end{pmatrix} . \tag{A.11.3}$$

The eigenvalue equation $S^2\mathbf{x} = \lambda \mathbf{x}$ then has the eigenvalues $s_i^2 = \lambda_i$ and the normalized eigenvectors are the basis vectors $\mathbf{x}_i = \mathbf{e}_i$.

In place of S we now set

$$A = U\, S\, V^{\mathrm{T}} \tag{A.11.4}$$

with orthogonal matrices U and V and

$$G = A^{\mathrm{T}} A = V S^{\mathrm{T}} S V^{\mathrm{T}} \quad . \tag{A.11.5}$$

We can write the eigenvalue equation of G in the form

$$G\mathbf{x} = \lambda \mathbf{x} \tag{A.11.6}$$

or

$$V S^{\mathrm{T}} S V^{\mathrm{T}} \mathbf{x} = \lambda \mathbf{x} \quad .$$

Multiplying on the left with V^{T},

$$S^{\mathrm{T}} S V^{\mathrm{T}} \mathbf{x} = \lambda V^{\mathrm{T}} \mathbf{x} \quad ,$$

and comparing with (A.11.1) and (A.11.2) shows that G has the same eigenvalues $\lambda_i = s_i^2$ as S^2, but has the orthogonally transformed eigenvectors

$$\mathbf{x}_i = \mathbf{e}'_i = V \mathbf{e}_i \quad . \tag{A.11.7}$$

One can clearly find the eigenvalues and eigenvectors of G if one knows the orthogonal matrix V that transforms G into a diagonal matrix,

$$V^{\mathrm{T}} G V = S^{\mathrm{T}} S = S^2 \quad . \tag{A.11.8}$$

The transformation (A.11.8) is called a *principal axis transformation*. The name becomes clear by considering the equation

$$\mathbf{r}^{\mathrm{T}} G \mathbf{r} = 1 \quad . \tag{A.11.9}$$

We are interested in the geometrical position of all points $\mathbf{r}$ that fulfill (A.11.9).

For the vector $\mathbf{r}$ the following two representations are completely equivalent:

$$\mathbf{r} = \sum_i r_i \mathbf{e}_i \tag{A.11.10}$$

and

$$\mathbf{r} = \sum_i r'_i \mathbf{e}'_i \quad , \tag{A.11.11}$$

with the components r_i and r'_i taken with respect to the basis vectors $\mathbf{e}_i$ and $\mathbf{e}'_i$, respectively. With the representation (A.11.11) and by using (A.11.5) and (A.11.7) we obtain

$$\begin{aligned} 1 &= \mathbf{r}^{\mathrm{T}} G \mathbf{r} = \mathbf{r}^{\mathrm{T}} V S^2 V^{\mathrm{T}} \mathbf{r} \\ &= \sum_i (r'_i \mathbf{e}'_i)^{\mathrm{T}} V S^2 V^{\mathrm{T}} \sum_j (r'_j \mathbf{e}'_j) \\ &= \sum_i (r'_i \mathbf{e}_i^{\mathrm{T}} V^{\mathrm{T}}) V S^2 V^{\mathrm{T}} \sum_j (r'_j V \mathbf{e}_j) \\ &= \sum_i (r'_i \mathbf{e}_i^{\mathrm{T}}) S^2 \sum_j (\mathbf{r}'_j \mathbf{e}_j) \end{aligned}$$

and finally

$$\sum_{i=1}^{n} r_i'^2 s_i^2 = \sum_{i=1}^{n} r_i'^2 / a_i^2 = 1 \quad . \tag{A.11.12}$$

This is clearly the equation of an ellipsoid in n dimensions with half-diameters in the directions $\mathbf{e}'_i$ and having the lengths

$$a_i = \frac{1}{s_i} \quad . \tag{A.11.13}$$

The vectors

$$\mathbf{a}_i = \mathbf{e}'_i / s_i = V \mathbf{e}_i / s_i \tag{A.11.14}$$

are the *principal axes* of the ellipsoid. They have the directions of the eigenvectors of G. Their lengths $a_i = 1/\sqrt{s_i^2}$ are determined by the eigenvalues s_i^2.

The matrix

$$C = G^{-1} = (A^{\mathrm{T}} A)^{-1} = V (S^2)^{-1} V^{\mathrm{T}}$$

clearly has the same eigenvectors as G, but has the eigenvalues $1/s_i^2$. The lengths of the half-diameters of the ellipsoid described above are then directly equal to the square roots of the eigenvalues of C.

The matrix C is called the *unweighted covariance matrix* of A. The ellipsoid is called the *unweighted covariance ellipsoid*.

Please note that all considerations were done for matrices of the type (A.11.5) which, by construction, are symmetric and *non-negative definite*, i.e., they have real, non-negative eigenvalues. Ellipses with finite semiaxes are obtained only with *positive-definite* matrices. If the same eigenvalue occurs several times then the ellipsoid has several principal axes of the same length. In this case there is a certain ambiguity in the determination of the principal axes which, however, can always be chosen orthogonal to each other.

Up to now we have not given any prescription for finding the eigenvalues. The eigenvalue equation is $G\mathbf{x} = \lambda \mathbf{x}$ or

$$(G - \lambda I)\mathbf{x} = 0 \quad . \tag{A.11.15}$$

Written in this way it can be considered as a linear system of equations for determining $\mathbf{x}$, for which the right-hand side is the null vector. Corresponding to (A.6.5) and (A.6.9) there is a non-trivial solution only for

$$\det(G - \lambda I) = 0 \quad . \tag{A.11.16}$$

This is the *characteristic equation* for determining the eigenvalues of A. For $n = 2$ this is

$$\begin{vmatrix} g_{11}-\lambda & g_{12} \\ g_{21} & g_{22}-\lambda \end{vmatrix} = (g_{11}-\lambda)(g_{22}-\lambda) - g_{12}g_{21} = 0$$

and has the solutions

$$\lambda_{1,2} = \frac{g_{11}+g_{22}}{2} \pm \sqrt{g_{12}g_{21} + \frac{(g_{11}-g_{22})^2}{4}} \quad .$$

If G is symmetric, as previously assumed, i.e., $g_{12} = g_{21}$, then the eigenvalues are real. If G is positive definite, they are positive.

The characteristic equation of an $n \times n$ matrix has n solutions. In practice, however, for $n > 2$ one does not find the eigenvalues by using the characteristic equation, but rather by means of an iterative procedure such as the singular value decomposition (see Sect. A.12).

A.12 Singular Value Decomposition

We now consider a particular orthogonal decomposition of the $m \times n$ matrix A,

$$A = U\,S\,V^{\mathrm{T}} \quad . \tag{A.12.1}$$

Here and in Sects. A.13 and A.14 we assume that $m \geq n$. If this is not the case, then one can simply extend the matrix A with further rows whose elements are all zero until it has n rows, so that $m = n$. The decomposition (A.12.1) is a special case of the decomposition (A.5.5). The $m \times n$ matrix S, which takes the place of R, has the special form

$$S = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \quad , \tag{A.12.2}$$

and D is a $k \times k$ diagonal matrix with $k = \mathrm{Rang}(A)$. The diagonal elements of S are called the *singular values* of A. If A is of full rank, then $k = n$ and all $s_i \neq 0$. For reduced rank $k < n$ one has $s_i = 0$ for $i > k$. We will see below that U and V can be determined such that all s_i are positive and ordered to be non-increasing,

$$s_1 \geq s_2 \geq \ldots \geq s_k \quad . \tag{A.12.3}$$

The singular values of A have a very simple meaning. From Sect. A.11 one has directly that the s_i are the square roots of the eigenvalues of $G = A^{\mathrm{T}}A$. Thus the half-diameters of the covariance ellipsoid of G are $a_i = 1/s_i$. If G is singular, then A has the reduced rank $k < n$, and the $n-k$ singular values $s_{k+1}, \ldots, s_n$ vanish. In this case the determinant

$$\det G = \det U \, \det S^2 \, \det V = \det S^2 = s_1^2 s_2^2 \cdots s_n^2 \tag{A.12.4}$$

also vanishes.

With the substitutions $H \to U$, $K \to V$, $R \to S$, $R_{11} \to D$ we obtain from Sect. A.5

$$A\mathbf{x} = U\,S\,V^{\mathrm{T}}\mathbf{x} \approx \mathbf{b} \quad , \tag{A.12.5}$$

$$S\,V^{\mathrm{T}}\mathbf{x} \approx U^{\mathrm{T}}\mathbf{b} \quad . \tag{A.12.6}$$

With

$$V^{\mathrm{T}}\mathbf{x} = \mathbf{p} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix} \begin{matrix} \} & k \\ \} & n-k \end{matrix} \quad , \quad U^{\mathrm{T}}\mathbf{b} = \mathbf{g} = \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{pmatrix} \begin{matrix} \} & k \\ \} & m-k \end{matrix} \tag{A.12.7}$$

one has

$$S\mathbf{p} = \mathbf{g} \quad , \tag{A.12.8}$$

i.e.,

$$D\mathbf{p}_1 = \mathbf{g}_1 \tag{A.12.9}$$

and

$$0 \cdot \mathbf{p}_2 = \mathbf{g}_2 \quad . \tag{A.12.10}$$

These have the solutions

$$\widetilde{\mathbf{p}}_1 = D^{-1}\mathbf{g}_1 \quad , \tag{A.12.11}$$

i.e.,

$$p_\ell = g_\ell / s_\ell \quad , \qquad \ell = 1, 2, \ldots, k \quad ,$$

for arbitrary $\mathbf{p}_2$. The solution with minimum absolute value is

$$\widetilde{\mathbf{x}} = V \begin{pmatrix} \widetilde{\mathbf{p}}_1 \\ 0 \end{pmatrix} \quad . \tag{A.12.12}$$

The residual vector has the form

$$\mathbf{r} = \mathbf{b} - A\widetilde{\mathbf{x}} = U \begin{pmatrix} 0 \\ \mathbf{g}_2 \end{pmatrix} \tag{A.12.13}$$

and the absolute value

$$r = |\mathbf{g}_2| = \left(\sum_{i=k+1}^{m} g_i^2 \right)^{1/2} \quad . \tag{A.12.14}$$

A.13 Singular Value Analysis

The rank k of the matrix A plays a decisive role in finding the solution of $A\mathbf{x} \approx \mathbf{b}$. Here k characterizes the transition from non-zero to vanishing singular values, $s_k > 0$, $s_{k+1} = 0$. How should one judge the case of very small values of s_k, i.e., $s_k < \varepsilon$ for a given small ε?

Example A.2: Almost vanishing singular values

As a simple example we consider the case $m = n = 2$, $U = V = I$. One then has

$$A\mathbf{x} = U\,S\,V^{\mathrm{T}}\mathbf{x} = S\mathbf{x} = \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \mathbf{b} \quad , \qquad \text{(A.13.1)}$$

that is,

$$x_1 = b_1/s_1 \quad , \qquad x_2 = b_2/s_2 \quad . \qquad \text{(A.13.2)}$$

If we now take $s_2 \to 0$ in (A.13.2), then $|x_2| \to \infty$. At first glance one obtains a completely different picture if one sets $s_2 = 0$ in (A.13.1) directly. This gives

$$s_1 x_1 = b_1 \quad , \qquad 0 \cdot x_2 = b_2 \quad . \qquad \text{(A.13.3)}$$

Thus $x_1 = b_1/s_1$ as in (A.13.2), but x_2 is completely undetermined. The solution $\widetilde{\mathbf{x}}$ of minimum absolute value is obtained by setting $x_2 = 0$. The question is now, "What is right? $x_2 = \infty$ or $x_2 = 0$?" The answer is that one should set

$$x_2 = \begin{cases} b_2/s_2\,, & s_2 \geq \varepsilon \\ 0\,, & s_2 < \varepsilon \end{cases}$$

and choose the parameter ε such that the expression b_2/ε is still numerically well-defined. This means that one must have $\varepsilon/|b_2| \gg 2^{-m}$ if m binary digits are available for the representation of a floating point number (cf. Sect. 4.2). Thus a finite value of b_2 is computed as long as the numerical determination of this value is reasonable. If this is not the case then one approaches the situation $s_2 = 0$, where b_2 is completely undetermined, and one sets $b_2 = 0$. ■

Example A.3: Point of intersection of two almost parallel lines

We consider the two lines shown in Fig. A.6, which are described by

$$\begin{aligned} -\alpha x_1 + x_2 &= 1 - \alpha \quad , \\ \alpha x_1 + x_2 &= 1 + \alpha \quad . \end{aligned}$$

For the vector $\mathbf{x}$ of the intersection point one has $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} -\alpha & 1 \\ \alpha & 1 \end{pmatrix} \quad , \qquad \mathbf{b} = \begin{pmatrix} 1-\alpha \\ 1+\alpha \end{pmatrix} \quad .$$

One can easily check that $A = U\,S\,V^{\mathrm{T}}$ holds with

$$U = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix} \quad , \qquad S = \sqrt{2}\begin{pmatrix} 1 & 0 \\ 0 & \alpha \end{pmatrix} \quad , \qquad V = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad ,$$

i.e., $s_1 = \sqrt{2}$, $s_2 = \alpha\sqrt{2}$. Using

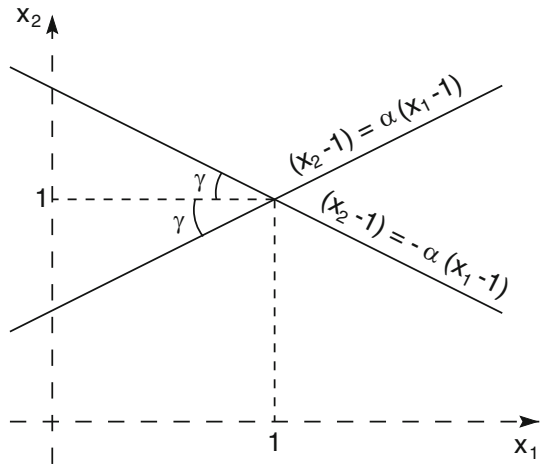

Fig. A.6: Two lines intersect at the point (1,1) with an angle $2\gamma = 2\arctan\alpha$.

$$\mathbf{g} = U^{\mathrm{T}}\mathbf{b} = \sqrt{2}\begin{pmatrix} -1 \\ \alpha \end{pmatrix} = S\mathbf{p} = \sqrt{2}\begin{pmatrix} p_1 \\ \alpha p_2 \end{pmatrix} \quad , \qquad \mathbf{p} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

one obtains

$$\mathbf{x} = V\mathbf{p} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

independent of α.

If, however, one has $s_2 = \alpha\sqrt{2} < \varepsilon$ and then sets $s_2 = 0$, then one obtains

$$\mathbf{p} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad , \qquad \mathbf{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad .$$

From Fig. A.6 one can see that for $\alpha \to 0$ the two lines come together to a single line described by $x_2 = 1$. The x_1-coordinate of the "intersection point" is completely undetermined. It is set equal to zero, since the solution vector $\mathbf{x}$ has minimum length [cf. (A.5.14)].

As in the case of "indirect measurements" in Chap. 9 we now assume that the vector $\mathbf{b}$ is equal to the vector of measurements $\mathbf{y}$, which characterize the two lines, and that their measurement errors are given by the covariance matrix $C_y = G_y^{-1}$. In the simplest case of equal uncorrelated errors one has

$$C_y = G_y^{-1} = \sigma^2 I \quad .$$

The covariance matrix for the unknowns $\widetilde{\mathbf{x}}$ is according to (9.2.27)

$$C_x = (A^{\mathrm{T}} G_y A)^{-1} \quad ,$$

and thus in the case of uncorrelated measurement errors,

$$C_x = \sigma^2 (A^{\mathrm{T}} A)^{-1} = \sigma^2 C \quad .$$

Up to the factor σ^2 it is thus equal to the unweighted covariance matrix $C = (A^{\mathrm{T}} A)^{-1}$ for the matrix A. For our matrix A one has

$$C = (A^{\mathrm{T}} A)^{-1} = \begin{pmatrix} 1/2\alpha^2 & 0 \\ 0 & 1/2 \end{pmatrix} \quad .$$

The corresponding ellipse has the half-diameters $\mathbf{e}_1/\alpha\sqrt{2}$ and $\mathbf{e}_2/\sqrt{2}$. The covariance ellipse of $\widetilde{\mathbf{x}}$ then has for the case of equal uncorrelated measurements equal half-diameters, multiplied, however, by the factor σ. They have the lengths $\sigma_{x_1} = \sigma/\alpha\sqrt{2}$, $\sigma_{x_2} = \sigma/\sqrt{2}$. Clearly one then sets $x_1 = 0$ if the inequality $x_1 \ll \sigma_{x_1}$ would hold for a finite fixed x_1, i.e., $\alpha\sqrt{2} \ll \sigma$. ■

The decision as to whether a small singular value should be set equal to zero thus depends on numerical considerations and on a consideration of the measurement errors. The following fairly general procedure of *singular value analysis* has proven to be useful in practice.

1. With a computer program one carries out the singular value decomposition, which yields, among other things, the ordered singular values

$$s_1 \geq s_2 \geq \cdots \geq s_k \quad .$$

2. Depending on the problem at hand one chooses a positive factor $f \ll 1$.
3. All singular values for which $s_i < f s_1$ are set equal to zero. In place of k one has thus $\ell \leq k$ such that $s_i = 0$ for $i > \ell$.
4. With the replacements described above ($k \to \ell$, $s_{\ell+1} = \cdots = s_k = 0$) the formulas of Sect. A.12 retain their validity.

In place of the residual (A.12.14) one obtains a somewhat larger value, since in the sum in the expression

$$r = \left(\sum_{i=\ell+1}^{m} g_i^2 \right)^{1/2} \tag{A.13.4}$$

one has more terms than in (A.12.14).

The procedure implies that in some cases one has an effective reduction of the rank k of the matrix A to a value $\ell < k$. If A has its full rank, then this can be reduced. This has the great advantage that numerical difficulties with small singular values are avoided. In contrast to, for example, the Gaussian or Cholesky procedures, the user does not have to worry about whether $G = A^{\mathrm{T}} A$ is singular or nearly singular.

Although the singular value analysis always gives a solution of minimum absolute value $\widetilde{\mathbf{x}}$ for the problem $A\mathbf{x} \approx \mathbf{b}$, caution is recommended for the case $\ell < n$ (regardless of whether $k < n$ or $\ell < k = n$). In such a case one has an (almost) linear dependence of the unknowns $x_1, \ldots, x_n$. The solution $\widetilde{\mathbf{x}}$ is not the only solution of the problem, but rather is simply the solution with the smallest absolute value out of the many possible solutions.

We have already remarked in Sect. A.5 that the singular value decomposition is advantageous also with respect to numerical precision compared to other methods, especially to the method of normal equations. A detailed discussion (see, e.g., [18]) is beyond the scope of this book. Here we limit ourselves to giving an example.

Example A.4: Numerical superiority of the singular value decomposition compared to the solution of normal equations

Consider the problem $A\mathbf{x} \approx \mathbf{b}$ for

$$A = \begin{pmatrix} 1 & 1 \\ \beta & 0 \\ 0 & \beta \end{pmatrix} \quad .$$

The singular values of A are the square roots of the eigenvalues of

$$G = A^{\mathrm{T}}A = \begin{pmatrix} 1+\beta^2 & 1 \\ 1 & 1+\beta^2 \end{pmatrix}$$

and are determined by (A.11.16) to be

$$s_1 = \sqrt{2+\beta^2} \quad , \qquad s_2 = |\beta| \quad .$$

This was done with singular value decomposition without using the matrix G. If the computing precision is ε and if $\beta^2 < \varepsilon$ but $\beta > \varepsilon$, then one obtains

$$s_1 = \sqrt{2} \quad , \qquad s_2 = |\beta| \quad ,$$

i.e., both singular values remain non-zero. If instead of the singular value decomposition one uses the normal equations

$$A^{\mathrm{T}}A\mathbf{x} = G\mathbf{x} = A^{\mathrm{T}}\mathbf{b} \quad ,$$

then the matrix G appears explicitly. With $\beta^2 < \varepsilon$ it is numerically represented as

$$G = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad .$$

This matrix is singular, $\det G = 0$, and cannot be inverted, as foreseen in (A.5.17). This is also reflected in its singular values

$$s_1 = \sqrt{2} \quad , \qquad s_2 = 0 \quad . \blacksquare$$

A.14 Algorithm for Singular Value Decomposition

A.14.1 Strategy

We will now give an algorithm for the singular value decomposition and follow for the most part the presentation of LAWSON and HANSON [18], which is based on the work of GOLUB and KAHN [19], BUSINGER and GOLUB [20], and GOLUB and REINSCH [21]. The strategy of the algorithm is based on the successive application of orthogonal transformations.

In the *first step*, the matrix A is transformed into a bidiagonal matrix C,

$$A = Q\,C\,H^{\mathrm{T}} \quad . \tag{A.14.1}$$

The orthogonal matrices Q and H are themselves products of Householder"=transformation matrices.

In *step 2* the matrix C is brought into diagonal form by means of an iterative procedure:

$$C = U'\,S'\,V'^{\mathrm{T}} \quad . \tag{A.14.2}$$

Here the matrices U' and V' are given by products of Givens-transformation matrices and if necessary reflection matrices. The latter ensure that all diagonal elements are non-negative.

In *step 3*, further orthogonal matrices U'' and V'' are determined. They are products of permutation matrices and ensure that the diagonal elements of

$$S = U''^{\mathrm{T}}\,S'\,V'' \tag{A.14.3}$$

are ordered so as to be non-increasing as in (A.12.3). As the result of the first three steps one obtains the matrices

$$U = Q\,U'\,U'' \quad , \qquad V = H\,V'\,V'' \quad , \tag{A.14.4}$$

which produce the singular value decomposition (A.12.1) as well as the diagonal elements $s_1, s_2, \ldots, s_k$.

In *step 4* the singular value analysis is finally carried out. Stated simply, all singular values are set to zero for which

$$s_i < s_{\min} \quad . \tag{A.14.5}$$

Thus one can finally give the solution vector $\mathbf{x}$ to the equation

$$A\mathbf{x} \approx \mathbf{b} \quad . \tag{A.14.6}$$

In practice, Eq. (A.14.6) is often solved simultaneously for various right-hand sides, which can then be arranged in an $m \times \ell$ matrix,

$$B = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_\ell) \quad .$$

Instead of the solution vector $\widetilde{\mathbf{x}}$ we thus obtain the $n \times \ell$ solution matrix

$$\widetilde{X} = (\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \ldots, \widetilde{\mathbf{x}}_\ell)$$

of the equation

$$A\,X \approx B \quad . \tag{A.14.7}$$

The method DatanMatrix singularValuedecomposition computes this solution. It merely consists of four calls to other methods that carry out the four steps described above. These are explained in more detail in the following sections.

Usually one is only interested in the solution matrix $\widetilde{X}$ of the problem $A\,X \approx B$ and possibly in the number of singular values not set to zero. Sometimes, however, one would like explicit access to the matrices U and V and to the singular values. This is made possible by the method DatanMatrix.pseudoInverse.

A.14.2 Bidiagonalization

The method DatanMatrix.sv1 performs the procedure described below. (It is declared as privatewithin the class DatanMatrix as are the further methods referred to in Sect. A.14; of course their source code can be studied.) The goal is to find the $m \times n$ matrix C in (A.14.1). Here C is of the form

$$C = \begin{pmatrix} C' \\ 0 \end{pmatrix} \quad , \tag{A.14.8}$$

and C' is an $n \times n$ bidiagonal matrix

$$C' = \begin{pmatrix} d_1 & e_2 & & & & \\ & d_2 & e_3 & & & \\ & & \cdot & \cdot & & \\ & & & \cdot & \cdot & \\ & & & & \cdot\ \cdot & e_n \\ & & & & & d_n \end{pmatrix} \quad . \tag{A.14.9}$$

The goal is achieved by multiplication of the matrix A with appropriate Householder matrices (alternating on the right and left):

$$C = Q_n(\cdots((Q_1 A)H_2)\cdots H_n) = Q^{\mathrm{T}} A\,H \quad . \tag{A.14.10}$$

The matrix Q_1 is computed from the elements of the first column of A and is applied to all columns of A (and B). It results in only the element

$(1,1)$ of A remaining non-zero. The matrix H_2 is computed from the first row of the matrix $Q_1 A$. It results in the element $(1,1)$ remaining unchanged, the element $(1,2)$ being recomputed and the elements $(1,3), \ldots, (1,n)$ being set equal to zero. It is applied to all rows of the matrix $(Q_1 A)$. One thus obtains

$$Q_1 A = \begin{pmatrix} \bullet & . & . & . \\ 0 & . & . & . \\ 0 & . & . & . \\ 0 & . & . & . \\ 0 & . & . & . \end{pmatrix} \quad , \qquad Q_1 A H_2 = \begin{pmatrix} \bullet & \bullet & 0 & 0 \\ 0 & . & . & . \\ 0 & . & . & . \\ 0 & . & . & . \\ 0 & . & . & . \end{pmatrix} \quad .$$

Here "•" denotes an element of the final bidiagonal matrix C', and "." an element that will be changed further. Now Q_2 is determined from the second column of $Q_1 A H_2$ such that upon application to this column the element 1 remains unchanged and the element 2 and all others are changed such that only element 2 remains non-zero, and the elements 3 through m become zero.

The procedure can be summarized in the following way. The matrix Q_i is applied to the column vectors, it leaves the elements 1 through $i-1$ unchanged and changes the elements i through m. It produces an orthogonal transformation in the subspace of the components i through m. At the time of applying Q_i, however, these elements in the columns 1 through $i-1$ are already zero, so that Q_i must only be applied explicitly to columns i through n. The corresponding considerations hold for the matrix H_i. It acts on the row vectors, leaves elements 1 through $i-1$ unchanged, and produces an orthogonal transformation in the subspace of the components i through n. When it is applied, these components in the rows 1 through $i-2$ are all zero. The matrix H_i must only be applied explicitly to the rows $i-1, \ldots, m-1$. Since only $n-1$ rows of A must be processed, the matrix H_n in (A.14.10) is the unit matrix; Q_n is the unit matrix only if $m = n$.

In addition to the transformed matrix $Q^{\mathrm{T}} A H$, the matrix $H = H_2 H_3 \cdots H_{n-1}$ is also stored. For this the information about each matrix H_i is saved. As we determined at the end of Sect. A.3, it is sufficient to store the quantities defined there, u_p and b, and the elements $i+1$ through n of the $(i-1)$th row vectors defining the matrix H_i. This can be done, however, in the array elements of these row vectors themselves, since it is these elements that will be transformed to zero and which therefore do not enter further into the calculation. One needs to declare additional variables only for the quantities u_p and b of each of the matrices H_i. If all of the transformations have been carried out, then the diagonal elements d_i and the next-to-diagonal elements e_i are transferred to the arrays e and d. Finally, the product matrix $H = H_2 H_3 \cdots H_{n-1} I$ is constructed in the first n rows of the array of the original matrix A, in the order $H_{n-1} I$, $H_{n-2}(H_{n-1} I)$, ... Here as well, the procedure is as economical as possible, i.e., the Householder matrix is only applied to those columns of

the matrix to the right for which it would actually produce a change. The unit matrix I is constructed to the extent that it is needed, row-by-row in the array of the original matrix A starting from below. For this there is exactly the right amount of space available, which up to this point was necessary for storing information about the Householder transformations that were just applied.

A.14.3 Diagonalization

This step is implemented in DatanMatrix.sv2. The bidiagonal matrix C, whose non-vanishing elements are stored in the arrays d and e, is now brought into diagonal form by appropriately chosen Givens transformations. The strategy is chosen such that the lowest non-diagonal element vanishes first and the non-diagonal elements always move up and to the left, until C is finally diagonal. All of the transformations applied to C from the left are also applied to the matrix stored in the array b, and all transformations that act from the right are also applied to the matrix in a. (We denote the matrix to be diagonalized during each step by C.)

Only the upper-left submatrix C_k with k rows and columns is not yet diagonal and must still be considered. The index k is determined such that $e_k \neq 0$ and $e_j = 0$, $j > k$. This means that the program runs through the loop $k = n, n-1, \ldots, 2$. Before the lower non-diagonal element e_k is systematically made zero by means of an iterative procedure, one checks for two special cases, which allow a shortening of the computation by means of special treatment.

Special case 1, $d_k = 0$ (handled in DatanMatrix.s21): A Givens matrix W is applied from the right, which also causes e_k to vanish. The matrix w is the product $W_{k-1}W_{k-2}\cdots W_1$. Here W_i acts on the columns i and k of C_k, but of course only on those rows where at least one of these columns has a non-vanishing element. W_{k-1} acts on the row $k-1$, annihilates the element $e_k = C_{k-1,k}$, and changes $C_{k-1,k-1}$. In addition, W_{k-1} acts on the row $k-2$, changes the element $C_{k-2,k-1}$, and produces a non-vanishing element $H = C_{k-2,k}$ in column k. Now the matrix W_{k-2} is applied, which annihilates exactly this element, but produces a new element in row $k-3$ and column k. When the additional element finally makes it to row 1, it can then be annihilated by the transformation W_1. As a result of this treatment of special case 1, C_k decomposes into a $(k-1)\times(k-1)$ submatrix C_{k-1} and a 1×1 null matrix.

Special case 2, C_k decomposes into submatrices (handled in DatanMatrix.s22): If $e_\ell = 0$ for any value ℓ, $2 \leq \ell \leq k$, then the matrix

$$C_k = \begin{pmatrix} d_1 & e_2 & & & & & \\ & \ddots & & & & & \\ & & d_{\ell-1} & 0 & & & \\ & & & d_\ell & e_{\ell+1} & & \\ & & & & \ddots & & \\ & & & & & d_{k-1} & e_k \\ & & & & & & d_k \end{pmatrix} = \begin{pmatrix} C_{\ell-1} & 0 \\ 0 & \bar{C} \end{pmatrix}$$

can be decomposed into two matrices $C_{\ell-1}$ and $\bar{C}$. One can first diagonalize $\bar{C}$ and then $C_{\ell-1}$. In particular, if $\ell = k$, then one obtains

$$C_k = \begin{pmatrix} C_{k-1} & 0 \\ 0 & d_k \end{pmatrix} \quad .$$

Here d_k is the singular value, and the loop index can only be decreased by one, $k \to k-1$. First, however, the case $d_k < 0$ must be treated; see "*change of sign*" below. If $d_{\ell-1} = 0$, but $e_\ell \neq 0$, then C can still be decomposed. For this one uses the transformation matrix $T = T_k T_{k-1}, \cdots, T_{\ell+1}$ acting on the left, where T_i acts on the rows ℓ and i. In particular, $T_{\ell+1}$ annihilates the element $e_\ell = C_{\ell,\ell+1}$ and creates an element $H = C_{\ell,\ell+2}$. $T_{\ell+2}$ annihilates this element and creates in its place $H = C_{\ell,\ell+3}$. Finally T_k annihilates the last created element $H = C_{\ell k}$ by a transformation to $C_{kk} = d_k$.

After it has been checked whether one or both of the special cases were present and such an occurrence has been treated accordingly, it remains only to diagonalize the matrix $\bar{C}$. This consists of the rows and columns ℓ through k of C. If special case 2 was not present, then one has $\ell = 1$. The problem is solved iteratively with the QR-algorithm.

QR-algorithm (carried out in DatanMatrix.s23): First we denote the square output matrix $\bar{C}$ by C_1. We then determine orthogonal matrices U_i, V_i and carry out transformations

$$C_{i+1} = U_i^{\mathrm{T}} C_i V_i \quad , \qquad i = 1, 2, \ldots \quad ,$$

which lead to a diagonal matrix S,

$$\lim_{i \to \infty} C_i = S \quad .$$

The following prescription is used for determining U_i and V_i:

(A) One determines the eigenvalues λ_1, λ_2 of the lower-right 2×2 submatrix of $C_i^{\mathrm{T}} C_i$. Here σ_i is the eigenvalue closest to the lower-right element of$C_i^{\mathrm{T}} C_i$.

(B) The matrix V_i is determined such that $V_i^{\mathrm{T}}(C_i^{\mathrm{T}}C_i - \sigma_i I)$ has upper triangular form.

(C) The matrix U_i is determined such that $C_{i+1} = U_i^{\mathrm{T}} C_i V_i$ is again bidiagonal.

The matrix V_i from step (B) exists, according to a theorem by FRANCIS [22], if

(a) $C_i^{\mathrm{T}}C_i$ is tridiagonal with non-vanishing subdiagonal elements,

(b) V_i is orthogonal,

(c) σ_i is an arbitrary scalar,

(d) $V_i^{\mathrm{T}}(C_i^{\mathrm{T}}C_i)V_i$ is tridiagonal, and

(e) The first column of $V_i^{\mathrm{T}}(C_i^{\mathrm{T}}C_i - \sigma_i I)$ except the first element vanishes.

The requirement (a) is fulfilled, since C_i is bidiagonal and the special case 2 has been treated if necessary; (b) is also fulfilled by constructing V_i as the product of Givens matrices. This is done in such a way that simultaneously (d) and (e) are fulfilled. In particular one has

$$V_i = R_1 R_2 \cdots R_{n-1} \quad , \qquad U_i^{\mathrm{T}} = L_{n-1} L_n \cdots L_1 \quad ,$$

where

R_j acts on the columns j and $j+1$ of C,
L_j acts on the rows j and $j+1$ of C,
R_1 is determined such that requirement (e) is fulfilled,
L_1, R_2, L_2, ... are determined such that (e) is fulfilled without violating (d).

For σ_i one obtains

$$\sigma_i = d_n^2 + e_n \left(e_n - \frac{d_{n-1}}{t} \right)$$

with

$$t = f + \sqrt{1+f^2} \quad , \qquad f \geq 0 \quad ,$$
$$t = f - \sqrt{1+f^2} \quad , \qquad f < 0 \quad ,$$

and

$$f = \frac{d_n^2 - d_{n-1}^2 + e_n^2 - e_{n-1}^2}{2 e_n d_{n-1}} \quad .$$

The first column of the matrix $(C_i^{\mathrm{T}}C_i - \sigma_i I)$ is

$$\begin{pmatrix} d_1^2 - \sigma_i \\ d_1 e_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} .$$

One determines the matrix R_1, which defines a Givens transformation, such that all elements of the first column of $R_1^{\mathrm{T}}(C_i^{\mathrm{T}} C_i - \sigma_i I)$ except the first vanish. Application of R_1 on C_i produces, however, an additional element $H = C_{21}$, so that C_i is no longer bidiagonal,

$$C_i = \begin{pmatrix} \cdot & \cdot & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & \cdot \end{pmatrix} , \qquad C_i R_1 = \begin{pmatrix} \cdot & \cdot & & & \\ H & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & \cdot \end{pmatrix} .$$

By application of L_1, this element is projected onto the diagonal. In its place a new element $H = C_{13}$ is created,

$$L_1 C_i R_1 = \begin{pmatrix} \cdot & \cdot & H & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & \cdot \end{pmatrix} .$$

By continuing the procedure, the additional element is moved further down and to the right, and can be completely eliminated in the last step:

$$T_{n-1} \begin{pmatrix} \cdot & \cdot & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & H & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & \cdot \end{pmatrix} = C_{i+1} .$$

If the lower non-diagonal element of C_{i+1} is already zero, then the lower diagonal element is already a singular value. Otherwise the procedure is repeated, whereby it is first checked whether now one of the two special cases is present. The procedure typically converges in about $2k$ steps; (k is the rank of the original matrix A). If convergence has still not been reached after $10k$ steps, then the algorithm is terminated without success.

Change of sign. If a singular value d_k has been found, i.e., if $e_k = 0$, then it is checked whether it is negative. If this is the case, then a simple orthogonal transformation is applied that multiplies the element $d_k = C_{kk}$ of

C and all elements in the kth column of the matrix contained in a by -1. The index k can then be reduced by one. Corresponding to (A.12.6) the matrix B is multiplied on the left by U^{T}. This is done successively with the individual factors making up U^{T}.

A.14.4 Ordering of the Singular Values and Permutation

By the method DatanMatrix.sv3 the singular values are put into non-increasing order. This is done by a sequence of permutations of neighboring singular values, carried out if the singular value that follows is larger then the preceding. The matrices stored in a and b are multiplied by a corresponding sequence of permutation matrices; cf. (A.14.4).

A.14.5 Singular Value Analysis

In the last step the singular value analysis is carried out as described in Sect. A.13 by the method DatanMatrix.sv4. For a given factor $f \ll 1$ a value $\ell \leq k$ is determined such that $s_i < f s_1$ for $i > \ell$. The columns of the array B, which now contains the vectors $\mathbf{g}$, are transformed in their first ℓ elements into $\widetilde{\mathbf{p}}_1$ according to (A.12.11), and the elements $\ell+1, \ldots, n$ are set equal to zero. Then the solution vectors $\widetilde{\mathbf{x}}$, which make up the columns of the solution matrix $\widetilde{X}$, are computed according to (A.12.12).

A.15 Least Squares with Weights

Instead of the problem (A.5.3),

$$r^2 = (A\mathbf{x} - \mathbf{b})^{\mathrm{T}}(A\mathbf{x} - \mathbf{b}) = \min \quad , \tag{A.15.1}$$

one often encounters a similar problem that in addition contains a positive-definite symmetric weight-matrix $G_{m\times m}$,

$$r^2 = (A\mathbf{x} - \mathbf{b})^{\mathrm{T}} G (A\mathbf{x} - \mathbf{b}) = \min \quad . \tag{A.15.2}$$

In (A.15.1) one simply has $G = I$. Using the Cholesky decomposition (A.9.1) of G, i.e., $G = U^{\mathrm{T}}U$, one has

$$r^2 = (A\mathbf{x} - \mathbf{b})^{\mathrm{T}} U^{\mathrm{T}} U (A\mathbf{x} - \mathbf{b}) = \min \quad . \tag{A.15.3}$$

With the definitions

$$A' = U\,A \quad , \qquad \mathbf{b}' = U\,\mathbf{b} \tag{A.15.4}$$

Eq. (A.15.3) takes on the form

$$r^2 = (A'\mathbf{x} - \mathbf{b}')^{\mathrm{T}}(A'\mathbf{x} - \mathbf{b}') = \min \quad . \tag{A.15.5}$$

After the replacement (A.15.4), the problem (A.15.2) is thus equivalent to the original one (A.15.1).

In Sect. A.11 we called the $n \times n$ matrix

$$C = (A^{\mathrm{T}}A)^{-1} \tag{A.15.6}$$

the unweighted covariance matrix of the unknowns $\mathbf{x}$ in the Problem (A.15.1). In Problem (A.15.2), the *weighted covariance matrix*

$$C_x = (A'^{\mathrm{T}}A')^{-1} = (A^{\mathrm{T}}G\,A)^{-1} \tag{A.15.7}$$

appears in its place.

A.16 Least Squares with Change of Scale

The goal in solving a problem of the type

$$(A\mathbf{x} - \mathbf{b})^{\mathrm{T}}(A\mathbf{x} - \mathbf{b}) = \min \tag{A.16.1}$$

is the most accurate numerical determination of the solution vector $\widetilde{\mathbf{x}}$ and the covariance matrix C. A change of scale in the elements of A, $\mathbf{b}$, and $\mathbf{x}$ can lead to an improvement in the numerical precision.

Let us assume that the person performing the problem already has an approximate idea of $\widetilde{\mathbf{x}}$ and C, which we call $\mathbf{z}$ and K. The matrix K has the Cholesky decomposition $K = L^{\mathrm{T}}L$. By defining

$$A' = A\,L \quad , \qquad \mathbf{b}' = \mathbf{b} - A\,\mathbf{z} \quad , \qquad \mathbf{x}' = L^{-1}(\mathbf{x} - \mathbf{z}) \quad , \tag{A.16.2}$$

Eq. (A.16.1) becomes

$$(A'\mathbf{x}' - \mathbf{b}')^{\mathrm{T}}(A'\mathbf{x}' - \mathbf{b}) = \min \quad .$$

The meaning of the new vector of unknowns $\mathbf{x}'$ is easily recognizable for the case where K is a diagonal matrix. We write

$$K = \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{pmatrix} \quad , \qquad L = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix} \quad ,$$

where the quantities σ_i^2 are the estimated values of the variances of the unknowns x_i. Thus the ith component of the vector $\mathbf{x}'$ becomes

$$x_i' = \frac{x_i - z_i}{\sigma_i} \quad .$$

If in fact one has $\widetilde{x}_i = z_i$ and the corresponding variance σ_i^2, then $x_i' = 0$ and has a variance of one. If the estimates are at least of the correct order of magnitude, the x_i' are close to zero and their variances are of order unity. In addition, in case the full matrix is, in fact, estimated with sufficient accuracy, the components of $\mathbf{x}'$ are not strongly correlated with each other.

In practice, one carries out the transformation (A.16.2) only in exceptional cases. One must take care, however, that "reasonable" variables are chosen for (A.16.2). This technique is applied in the graphical representation of data. If it is known, for example, that a voltage U varies in the region $\sigma = 10\,\mathrm{mV}$ about the value $U_0 = 1\,\mathrm{V}$, then instead of U one would plot the quantity $U' = (U - U_0)/\sigma$, or some similar quantity.

A.17 Modification of Least Squares According to Marquardt

Instead of the problem

$$(A\mathbf{x} - \mathbf{b})^{\mathrm{T}}(A\mathbf{x} - \mathbf{b}) = \min \quad , \tag{A.17.1}$$

which we have also written in the shorter form

$$A\mathbf{x} - \mathbf{b} \approx 0 \quad , \tag{A.17.2}$$

let us now consider the modified problem

$$\begin{matrix} m \\ n \end{matrix} \left\{ \underbrace{\begin{pmatrix} A \\ \lambda I \end{pmatrix}}_{n} \mathbf{x} \approx \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix} \right\} \begin{matrix} m \\ n \end{matrix} \quad . \tag{A.17.3}$$

Here I is the $n \times n$ unit matrix and λ is a non-negative number. The modified problem is of considerable importance for fitting nonlinear functions with the method of least squares (Sect. 9.5) or for minimization (Sect. 10.15). For $\lambda = 0$ Eq. (A.17.3) clearly becomes (A.17.2). If, on the other hand, λ is very large, or more precisely, if it is large compared to the absolute values of the elements of A and $\mathbf{b}$, then the last "row" of (A.17.3) determines the solution $\widetilde{\mathbf{x}}$, which is the null vector for $\lambda \to \infty$.

We first ask which direction the vector $\widetilde{\mathbf{x}}$ has for large values of λ. The normal equations corresponding to (A.17.3), cf. (A.5.16), are

$$(A^{\mathrm{T}},\, \lambda\, I)\begin{pmatrix} A \\ \lambda I \end{pmatrix}\mathbf{x} = (A^{\mathrm{T}}A + \lambda^2 I)\mathbf{x} = (A^{\mathrm{T}},\, \lambda\, I)\begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix} = A^{\mathrm{T}}\mathbf{b}$$

with the solution

$$\widetilde{\mathbf{x}} = (A^{\mathrm{T}}A + \lambda^2 I)^{-1}A^{\mathrm{T}}\mathbf{b} \quad .$$

For large λ the second term in parentheses dominates, and one obtains simply

$$\widetilde{\mathbf{x}} = \lambda^{-2}A^{\mathrm{T}}\mathbf{b} \quad .$$

That is, for large values of λ, the solution vector tends toward the direction of the vector $A^{\mathrm{T}}\mathbf{b}$.

We will now show that for a given λ the solution $\mathbf{x}^{(\lambda)}$ to (A.17.3) can easily be found with the singular value decomposition simultaneously with the determination of the solution $\widetilde{\mathbf{x}}$ of (A.17.2). The singular value decomposition (A.12.1) of A is

$$A = U\begin{pmatrix} S \\ 0 \end{pmatrix}V^{\mathrm{T}} \quad .$$

By substituting into (A.17.3) and multiplying on the left we obtain

$$\begin{pmatrix} U^{\mathrm{T}} & 0 \\ 0 & V^{\mathrm{T}} \end{pmatrix}\begin{pmatrix} U\binom{S}{0}V^{\mathrm{T}} \\ \lambda I \end{pmatrix}\mathbf{x} = \begin{pmatrix} U^{\mathrm{T}} & 0 \\ 0 & V^{\mathrm{T}} \end{pmatrix}\begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}$$

or

$$\begin{pmatrix} S \\ 0 \\ \lambda I \end{pmatrix}\mathbf{p} = \begin{pmatrix} \mathbf{g} \\ 0 \end{pmatrix} \quad , \tag{A.17.4}$$

where, using the notation as in Sect. A.12,

$$\mathbf{p} = V^{\mathrm{T}}\mathbf{x} \quad , \qquad \mathbf{g} = U^{\mathrm{T}}\mathbf{b} \quad .$$

By means of Givens transformations, the matrix on the left-hand side of (A.17.4) can be brought into diagonal form. One obtains

$$\begin{pmatrix} S^{(\lambda)} \\ 0 \\ 0 \end{pmatrix}\mathbf{p} = \begin{pmatrix} \mathbf{g}^{(\lambda)} \\ \mathbf{h}^{(\lambda)} \end{pmatrix}$$

with

$$\begin{aligned} g_i^{(\lambda)} &= \frac{g_i s_i}{s_i^{(\lambda)}} \quad , \qquad i = 1, \ldots, n \quad , \\ g_i^{(\lambda)} &= g_i \quad , \qquad i = n+1, \ldots, m \quad , \\ h_i^{(\lambda)} &= -\frac{g_i^{\lambda}}{s_i^{(\lambda)}} \quad , \qquad i = 1, \ldots, n \quad , \\ s_i^{(\lambda)} &= \sqrt{s_i^2 + \lambda^2} \end{aligned}$$

and thus

$$\begin{aligned} p_i^{(\lambda)} &= \frac{g_i s_i}{s_i^2+\lambda^2} = p_i^{(0)} \frac{s_i^2}{s_i^2+\lambda^2} \quad , \qquad i = 1, \ldots, k \quad , \\ p_i^{(\lambda)} &= 0 \quad , \qquad i = k+1, \ldots, n \quad . \end{aligned}$$

The solution $\widetilde{\mathbf{x}}^{(\lambda)}$ of (A.17.3) is then

$$\widetilde{\mathbf{x}}^{(\lambda)} = V \mathbf{p}^{(\lambda)} \quad .$$

The method DatanMatrix.marquardt computes these solution vectors for two values of λ. It proceeds mostly as DatanMatrix.sigularValueDecomposition; only in step 4 instead of Datanmatrix.sv4 the method DatanMatrix.svm is used which is adapted to the Marquardt problem.

A.18 Least Squares with Constraints

One often encounters the problem (A.5.3)

$$r^2 = (A\mathbf{x} - \mathbf{b})^2 = \min \tag{A.18.1}$$

with the *constraint*

$$E\mathbf{x} = \mathbf{d} \quad . \tag{A.18.2}$$

Here A is as before an $m \times n$ matrix and E is an $\ell \times n$ matrix. We will restrict ourselves to the only case that occurs in practice,

$$\operatorname{Rang} E = \ell < n \quad . \tag{A.18.3}$$

The determination of an extreme value with constraints is usually treated in analysis textbooks with the method of Lagrange multipliers. Here as well we rely on orthogonal transformations. The following method is due to LAWSON and HANSON [18]. It uses a basis of the null space of E. First we carry out an orthogonal decomposition of E as in (A.5.5),

$$E = H\, R\, K^{\mathrm{T}} \quad . \tag{A.18.4}$$

Here we regard the orthogonal $n \times n$ matrix K as being constructed out of an $n \times \ell$ matrix K_1 and an $n \times (n-\ell)$ matrix K_2,

$$K = (K_1,\, K_2) \quad . \tag{A.18.5}$$

According to (A.5.6) and (A.5.7), all solutions of (A.18.2) have the form

$$\widehat{\mathbf{x}} = K_1\widetilde{\mathbf{p}}_1 + K_2\mathbf{p}_2 = \widetilde{\mathbf{x}} + K_2\mathbf{p}_2 \quad . \tag{A.18.6}$$

Here $\widetilde{\mathbf{x}}$ is the unique solution of minimum absolute value of (A.18.2). For brevity we will write this in the form $\widetilde{\mathbf{x}} = E^+\mathbf{d}$; cf. (A.10.3). $\mathbf{p}_2$ is an arbitrary $(n-\ell)$-vector, since the vectors $K_2\mathbf{p}_2$ form the null space of E,

$$E\,K_2\mathbf{p}_2 = 0 \quad . \tag{A.18.7}$$

The constraint (A.18.2) thus says that the vector $\mathbf{x}$ for which (A.18.1) is a minimum must come from the set of all vectors $\widehat{\mathbf{x}}$, i.e.,

$$\begin{aligned} (A\widehat{\mathbf{x}} - \mathbf{b})^2 &= (A(\widetilde{\mathbf{x}} + K_2\mathbf{p}_2) - \mathbf{b})^2 \\ &= (A\,K_2\mathbf{p}_2 - (\mathbf{b} - A\widetilde{\mathbf{x}}))^2 = \min \quad . \end{aligned} \tag{A.18.8}$$

This relation is a least-squares problem without constraints, from which the $(n-\ell)$-vector $\mathbf{p}_2$ can be determined. We write its solution using (A.10.3) in the form

$$\widetilde{\mathbf{p}}_2 = (A\,K_2)^+(\mathbf{b} - A\widetilde{\mathbf{x}}) \quad . \tag{A.18.9}$$

By substitution into (A.18.6) we finally obtain

$$\mathbf{x} = \widetilde{\mathbf{x}} + A^+(\mathbf{b} - A\widetilde{\mathbf{x}}) = E^+\mathbf{d} + K_2(A\,K_2)^+(\mathbf{b} - A\,E^+\mathbf{d}) \tag{A.18.10}$$

as the solution of (A.18.1) with the constraint (A.18.2).

The following prescription leads to the solution (A.18.10). Its starting point is the fact that one can set $H = I$ because of (A.18.3).

Step 1: One determines an orthogonal matrix $K = (K_1, K_2)$ as in (A.18.5) such that

$$E\,K = (E\,K_1,\, E\,K_2) = (\widetilde{E}_1,\, 0)$$

and such that $\widetilde{E}_1$ is a lower triangular matrix. In addition one computes

$$A\,K = (A\,K_1,\, A\,K_2) = (\widetilde{A}_1,\, \widetilde{A}_2) \quad .$$

Step 2: One determines the solution $\widetilde{\mathbf{p}}_1$ of

$$\widetilde{E}_1\mathbf{p}_1 = \mathbf{d} \quad .$$

This is easy, since $\widetilde{E}_1$ is a lower triangular matrix of rank ℓ. Clearly one has $\widetilde{\mathbf{x}} = K_1\widetilde{\mathbf{p}}_1$.

Step 3: One determines the vector

$$\bar{\mathbf{b}} = \mathbf{b} - \widetilde{A}_1\widetilde{\mathbf{p}}_1 = \mathbf{b} - A\,K_1K_1^{\mathrm{T}}\widetilde{\mathbf{x}} = \mathbf{b} - A\widetilde{\mathbf{x}} \quad .$$

Step 4: One determines the solution $\widetilde{\mathbf{p}}_2$ to the least-squares problem (A.18.8) (without constraints)

$$(\widetilde{A}_2\mathbf{p}_2 - \bar{\mathbf{b}})^2 = \min \quad .$$

Step 5: From the results of steps 2 and 4 one finds the solution to (A.18.1) with the constraint (A.18.2)

$$\mathbf{x} = K\begin{pmatrix}\widetilde{\mathbf{p}}_1\\ \widetilde{\mathbf{p}}_2\end{pmatrix} = \begin{pmatrix}K_1\widetilde{\mathbf{p}}_1\\ K_2\widetilde{\mathbf{p}}_2\end{pmatrix} \quad .$$

We will now consider a simple example that illustrates both the least-squares problem with constraints as well as the method of solution given above.

Example A.5: Least squares with constraints
Suppose the relation (A.18.1) has the simple form

$$r^2 = \mathbf{x}^2 = \min$$

for $n = 2$. One then has $m = n = 2$, $A = I$, and $\mathbf{b} = 0$. Suppose the constraint is

$$x_1 + x_2 = 1 \quad ,$$

i.e., $\ell = 1$, $E = (1, 1)$, $\mathbf{d} = 1$.

The problem has been chosen such that it can be solved by inspection without mathematical complications. The function $z = \mathbf{x}^2 = x_1^2 + x_2^2$ corresponds to a paraboloid in the (x_1, x_2, z) space, whose minimum is at $x_1 = x_2 = 0$. We want, however, to find not the minimum in the entire (x_1, x_2) plane, but rather only on the line $x_1 + x_2 = 1$, as shown in Fig. A.7. It clearly lies at the point where the line has its smallest distance from the origin, i.e., at $x_1 = x_2 = 1/2$.

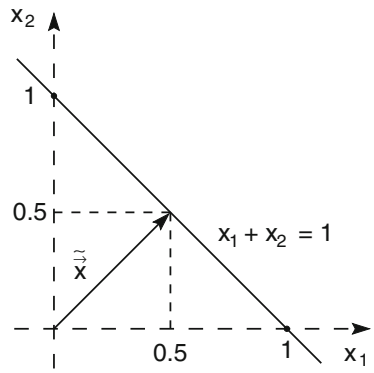

Fig. A.7: The solution $\widetilde{\mathbf{x}}$ to Example A.5 lies on the line given by the constraint $x_1 + x_2 = 1$.

Of course one obtains the same result with the algorithm. With

$$K = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

we obtain $\widetilde{E}_1 = \sqrt{2}$, $\widetilde{\mathbf{p}}_1 = 1/\sqrt{2}$,

$$\widetilde{A}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad , \qquad (\widetilde{A}_2) = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad , \qquad \bar{\mathbf{b}} = -\frac{1}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad .$$

We solve the problem $(\widetilde{A}_2 \mathbf{p}_2 - \bar{\mathbf{b}})^2 = \min$ with the normal equations

$$\widetilde{\mathbf{p}}_2 = (\widetilde{A}_2^{\mathrm{T}} \widetilde{A}_2)^{-1} \widetilde{A}_2^{\mathrm{T}} \bar{\mathbf{b}} = (\sqrt{2})^{-1} \cdot 0 = 0 \quad .$$

The full solution is then

$$\mathbf{x} = K \begin{pmatrix} \widetilde{\mathbf{p}}_1 \\ \widetilde{\mathbf{p}}_2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} \\ 0 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \quad . \blacksquare$$

The method DatanMatrix.leastSquaresWithConstraints solves the problem of least squares $(A\mathbf{x} - \mathbf{b})^2 = \min$ with the linear constraint $E\mathbf{x} = \mathbf{d}$.

A.19 Java Classes and Example Programs

Java Classes for Vector and Matrix Operations

`DatanVector` contains methods for vector operations.

`DatanMatrix` contains methods for matrix operations.

Example Program A.1: The class E1Mtx demonstrates simple operations of matrix and vector algebra

At first the matrices

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \quad , \qquad B = \begin{pmatrix} 2 & 3 & 1 \\ 1 & 5 & 4 \end{pmatrix} \quad , \qquad C = \begin{pmatrix} 1 & 5 \\ 3 & 4 \\ 2 & 3 \end{pmatrix}$$

and the vectors

$$\mathbf{u} = \begin{pmatrix} 0 \\ 3 \\ 4 \end{pmatrix} \quad , \qquad \mathbf{v} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} \quad , \qquad \mathbf{w} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

are defined. Then, with the appropriate methods, simple operations are performed with these quantities. Finally, the resulting matrices and vectors are displayed. The operations are

$R = A$, $R = A + B$, $R = A - B$, $R = AC$, $S = AB^{\mathrm{T}}$, $T = A^{\mathrm{T}} B$, $R = I$, $R = 0.5\,A$,

$R = A^{\mathrm{T}}$, $\mathbf{z} = \mathbf{w}$, $\mathbf{x} = \mathbf{u} + \mathbf{v}$, $\mathbf{x} = \mathbf{u} - \mathbf{v}$, $d = \mathbf{u} \cdot \mathbf{v}$, $s = |\mathbf{u}|$, $\mathbf{x} = 0.5\,\mathbf{u}$, $\mathbf{x} = 0$.

Example Program A.2: The class E2Mtx demonstrates the handling of submatrices and subvectors.

The matrices

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} \quad , \qquad D = \begin{pmatrix} 0 & 2 \\ 1 & 3 \end{pmatrix}$$

and the vectors

$$\mathbf{u} = \begin{pmatrix} 0 \\ 3 \\ 4 \end{pmatrix} \quad , \qquad \mathbf{w} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

are defined.

Next a submatrix S is taken from A, and a submatrix of A is overwritten by D. A column vector and a row vector are taken from A and inserted into A. Finally some elements are taken according to a list from the vector **u** and assembled in a vector **z**, and elements of **w** are put into positions (defined by list) of the vector **u**.

Example Program A.3: The class E3Mtx demonstrates the performance of Givens transformations

Fist the two vectors

$$\mathbf{u} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \quad , \qquad \mathbf{w} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

are defined. Next, by the use of DatanMatrix.defineGivensTransformation transformation parameters c and s for the vector **u** are computed and displayed. The Givens transformation of **u** with these parameters is performed with DatanMatrix.applyGivensTransformation yielding

$$\mathbf{u}' = \begin{pmatrix} 5 \\ 0 \end{pmatrix} \quad .$$

Finally, by calling DatanMatrix.applyGivensTransformation parameters c and s are computed for the vector **w** and the transformation is applied to this vector.

Example Program A.4: The class E4Mtx demonstrates the performance of Householder transformations

First the two vectors

$$\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 4 \\ 3 \\ 4 \end{pmatrix} \quad , \qquad \mathbf{c} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 4 \\ 3 \\ 4 \end{pmatrix}$$

are defined. Moreover, the indices n, p are set to ℓ $n = 6$, $p = 3$, $\ell = 5$. By calling DatanMatrix.defineHouseholderTransformation the Householder transformation defined by these indices and the vector **v** is initialized. The application of this transformation to the vectors **v** and **c** is performed by two calls of DatanMatrix.defineHouseholderTransformation. The results are displayed alphanumerically.

Example Program A.5: The class E5Mtx demonstrates the Gaussian algorithm for the solution of matrix equations

The matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & -2 \\ 1 & 1 & 2 \end{pmatrix}$$

and the 3×3 unit matrix B are defined. A call of Datanmatrix.matrix Equation solves the matrix equation $AX = B$, i.e., X is the inverse of A. The matrix A is identical to the one chosen in Example A.1. In that example the algorithm is shown step by step.

Example Program A.6: The class E6Mtx demonstrates Cholesky decomposition and Cholesky inversion

First, the matrices

$$B = \begin{pmatrix} 1 & 7 & 13 \\ 3 & 9 & 17 \\ 5 & 11 & 19 \end{pmatrix} \quad , \qquad C = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$$

are defined and the symmetric, positive-definite matrix $A = B^{\mathrm{T}}B$ is constructed. By calling DatanMatrix.choleskyDecompostion the Cholesky decomposition $A = U^{\mathrm{T}}U$ is performed and the triangular matrix U is displayed. Multiplication of U^{T} by U yields in fact $U^{\mathrm{T}}U = S$. The method DatanMatrix.cholesky-Multiply is then used to compute $R = UC$. Finally, by Cholesky inversion with DatanMatrix.choleskyInversion , the inverse S^{-1} of S is computed. Multiplication with the original matrix yields $SS^{-1} = I$.

Example Program A.7: The class E7Mtx demonstrates the singular value decomposition

The program first operates on the same matrix as E5Mtx. However, the matrix inversion is now performed with DatanMatrix.pseudoInverse

Next, the matrix is replaced by

$$A = \begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad .$$

This matrix, having two identical columns, is singular. With another call of Datan Matrix.pseudo Inversenot only the psudoinverse matrix but also the residuals, the diagonal matrix D, and the two orthogonal matrices U and V are determined.

Example Program A.8: The class E8Mtx demonstrates the solution of matrix equations by singular value decomposition for 9 different cases

In the framework of our programs, in particular that of least-squares and minimization problems, matrix equations are solved nearly exclusively by singular value decomposition. In Sect. A.5 we have listed the different cases of the matrix equation $A\mathbf{x} \approx \mathbf{b}$. If A is an $m \times n$ matrix then first we must distinguish between the cases $m = n$ (case 1), $m > n$ (case 2), and $m < n$ (case 3). A further subdivision is brought about by the rank of A. If the rank k of A is $k = \min(m, n)$ then we have the case 1a (or 2a or 3a). If, however, $k < \min(m, n)$ then we are dealing with case 1b (or 2b or 3b). The rank of a matrix is equal to its number of non-vanishing singular values. In numerical calculations, which are always performed with finite accuracy, one obviously has to define more precisely the meaning of "non-vanishing". For this definition we use the method of singular value analysis (Sect. A.13) and set a singular value equal to zero if it is smaller than a fraction f of the largest singular value. The number of finite singular values remaining in this analysis is called the *pseudorank*. In addition to the cases mentioned above we consider as well the cases 1c, 2c, and 3c, in which the matrix A has full rank but not full pseudorank.

The program consists of two nested loops. The outer loop runs through the cases 1, 2, 3, the inner loop through the subcases a, b, c. For each case the matrix A is composed of individual vectors. In the subcase b two of these vectors are chosen to be identical. In the subcase c they are identical except for one element, which in one vector differs by $\varepsilon = 10^{-12}$ compared to the value in the other vector. In case 3 the system of linear equations symbolized by the matrix equation $Ax = b$ has less equations (m) than unknowns (n). This case does not appear in practice and, therefore, is not included in the programs. It is simulated here in the following way. In case 3 with $m = 2$ and $n = 3$, the matrix A is extended to become a 3×3 matrix by addition of another row the elements, of which are all set to zero, and correspondingly m is set to 3.

If the singular value analysis shows that one or several singular values are smaller than the fraction f of the largest singular value, then they are set to zero. In our example program for each of the 9 cases the analysis is performed twice, first for $f = 10^{-15}$ and then for $f = 10^{-10}$. For $f = 10^{-15}$ in our example cases 1c, 2c, 3c the matrix A has full rank, in spite of the small value of $\varepsilon = 10^{-12}$. the singular value analysis with $f = 10^{-10}$ reduces the number of singular values. Note that the elements of the solution matrix differ as the choice of f changes for cases 1c, 2c, 3c. The unwieldly numerical values in the case of $f = 10^{-15}$ show that we are near the limits of numerical stability.

Example Program A.9: The class E9Mtx demonstrates the use of the method DatanMatrix.marquardt

The method DatanMatrix.marquardt will be rarely called directly. It was written to be used in LsqMar and MinMar. For completeness we demonstrate it with a short class. It solves the problem $A\mathbf{x} \approx \mathbf{b}$ – modified according to (A.17.3) – for given A, $\mathbf{b}$, and λ and displays the results $\mathbf{x}_1$ and $\mathbf{x}_2$.

Example Program A.10: The class E10Mtx demonstrates solution of the least squares problem with constraints by the method DatanGraphics leastSquaresWithConstraints

The problem of Examples 9.9 and 9.10 is solved, i.e., the measurement $x_1 = 89$, $x_2 = 31$, $x_3 = 62$ of the three angles of a triangle and the evaluation of these measurements using the constraint $x_1 + x_2 + x_3 = 180$. The evaluation requires the solution of $(A\mathbf{x} - \mathbf{b})^2 = \min$ with the constraint $E\mathbf{x} = \mathbf{d}$ with

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad , \qquad \mathbf{b} = \begin{pmatrix} 89 \\ 31 \\ 62 \end{pmatrix} \quad , \qquad E = (1, 1, 1) \quad , \qquad \mathbf{d} = 180 \quad .$$

In the program the matrices and vectors A, $\mathbf{b}$, E, $\mathbf{d}$ are provided. The solution is computed by calling DatanGraphics leastSquaresWithConstraints . It is, of course, identical to the results found in the previously mentioned examples.

B. Combinatorics

Consider n distinguishable objects a_1, a_2, …, a_n. We ask for the number of possible ways P_n^k, in which one can place k of them in a given order. Such orderings are called *permutations*. For the example $n = 4$, $k = 2$ these permutations are

$$\begin{array}{lll} a_1a_2 \quad , & a_1a_3 \quad , & a_1a_4 \quad , \\ a_2a_1 \quad , & a_2a_3 \quad , & a_2a_4 \quad , \\ a_3a_1 \quad , & a_3a_2 \quad , & a_3a_4 \quad , \\ a_4a_1 \quad , & a_4a_2 \quad , & a_4a_3 \quad , \end{array}$$

i.e., $P_k^n = 12$. The answer for the general problem can be derived from the following scheme. There are n different possible ways to occupy the first place in a sequence. When one of these ways has been chosen, however, there are only $n-1$ objects left, i.e., there remain $n-1$ ways to occupy the second place, and so forth. One therefore has

$$P_k^n = n(n-1)(n-2)\cdots(n-k+1) \quad . \tag{B.1}$$

The result can also be written in the form

$$P_k^n = \frac{n!}{(n-k)!} \quad , \tag{B.2}$$

where

$$n! = 1 \cdot 2 \cdots n \quad ; \qquad 0! = 1 \quad , \qquad 1! = 1 \quad . \tag{B.3}$$

Often one is not interested in the order of the k objects within a permutation (the same k objects can be arranged in $k!$ different ways within the sequence), but rather one only considers the number of different ways of choosing of k objects out of a total of n. Such a choice is called a *combination*. The number of possible combinations of k elements out of n is then

$$C_k^n = \frac{P_k^n}{k!} = \frac{n!}{k!(n-k)!} = \binom{n}{k} \quad . \tag{B.4}$$

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

n	$\binom{n}{k}$									
0	1									
1	1		1							
2	1		2		1					
3	1		3		3		1			
4	1		4		6		4		1	
5	1		5		10		10		5		1
6	1		6		15		20		15		6		1
7	1		7		21		35		35		21		7		1
8	1		8		28		56		70		56		28		8		1
$\vdots$	$\vdots$									
	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	
	0	1	2	3	4	5	6	7	8	k

Fig. B.1: Pascal's triangle.

For the *binomial coefficients* $\binom{n}{k}$ one has the simple recursion relation

$$\binom{n-1}{k} + \binom{n-1}{k-1} = \binom{n}{k} \quad , \tag{B.5}$$

which can easily be proven by computation:

$$\begin{aligned} &\frac{(n-1)!}{k!(n-k-1)!} + \frac{(n-1)!}{(k-1)!(n-k)!} \\ &= \frac{(n-k)(n-1)! + k(n-1)!}{k!(n-k)!} \\ &= \frac{n!}{k!(n-k)!} \quad . \end{aligned}$$

The recursion formula is the basis for the famous *Pascal's triangle*, which, because of its beauty, is shown in Fig. B.1. The name "binomial coefficient" comes from the well-known *binomial theorem*,

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k} \quad , \tag{B.6}$$

the proof of which (by induction) is left to the reader. We use the theorem in order to derive a very important property of the coefficient $\binom{n}{k}$. For this we write it in the simple form for $b = 1$, i.e.,

$$(a+1)^n = \sum_{k=0}^{n} \binom{n}{k} a^k$$

and we then apply it a second time,

$$(a+1)^{n+m} = (a+1)^n (a+1)^m \quad ,$$

$$\sum_{\ell=0}^{n+m} \binom{n+m}{\ell} a^\ell = \sum_{j=0}^{n} \binom{n}{j} a^j \sum_{k=0}^{m} \binom{m}{k} a^k \quad .$$

If we consider only the term with a^ℓ, then by comparing coefficients we find

$$\binom{n+m}{\ell} = \sum_{j=0}^{\ell} \binom{n}{j} \binom{m}{\ell - j} \quad . \tag{B.7}$$

C. Formulas and Methods for the Computation of Statistical Functions

C.1 Binomial Distribution

We present here two function subprograms for computing the binomial distribution (5.1.3)

$$W_k^n = \binom{n}{k} p^k (1-p)^{n-k} \tag{C.1.1}$$

and the distribution function

$$P(k < K) = \sum_{k=0}^{K-1} W_k^n \tag{C.1.2}$$

are computed by the methods StatFunct.binomial and StatFunct.-cumulativeBinomial, respectively. For reasons of numerical stability the logarithm of Euler's gamma function is used in the computation.

C.2 Hypergeometric Distribution

The hypergeometric distribution (5.3.1)

$$W_k = \binom{K}{k}\binom{N-K}{n-k} \Big/ \binom{N}{n} \quad , \qquad n \le N, k \le K \quad , \tag{C.2.1}$$

and the corresponding distribution function

$$P(k < k') = \sum_{k=0}^{k'-1} W_k \tag{C.2.2}$$

are computed by StatFunct.hypergeometric and StatFunct.-cumulativeHypergeometric, respectively.

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

C.3 Poisson Distribution

The Poisson distribution (5.4.1)

$$f(k;\lambda) = \frac{\lambda^k}{k!}\mathrm{e}^{-\lambda} \tag{C.3.1}$$

and the corresponding distribution function

$$P(k < K) = \sum_{k=0}^{K-1} f(k;\lambda) \tag{C.3.2}$$

are computed with the methods StatFunct.poisson and StatFunct.-cumulative.Poisson respectively.

The quantities $f(k;\lambda)$ and $F(K;\lambda)$ depend not only on the values of the discrete variables k and K, but also on the continuous parameter λ. For a given P there is a certain parameter value λ_P that fulfills Eq. (C.3.2). We can denote this as the quantile

$$\lambda = \lambda_P(K) \tag{C.3.3}$$

of the Poisson distribution. It is computed by the method StatFunct.-quantilePoisson.

C.4 Normal Distribution

The probability density of the *standard normal distribution* is

$$\phi_0(x) = \frac{1}{\sqrt{2\pi}}\exp(-x^2/2) \quad . \tag{C.4.1}$$

It is computed by the methodStatFunc.standardNormal.

The *normal distribution* with mean x_0 and variance σ^2,

$$\phi(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{(x-x_0)^2}{2\sigma^2}\right) \quad , \tag{C.4.2}$$

can easily be expressed in terms of the standardized variable

$$u = \frac{x - x_0}{\sigma} \tag{C.4.3}$$

using (C.4.1) to be

$$\phi(x) = \frac{1}{\sigma}\phi_0(u) \quad . \tag{C.4.4}$$

It is computed by the method StatFunct.normal.

The distribution function of the standard normal distribution

$$\psi_0(x) = \int_{-\infty}^{x} \phi_0(x)\,\mathrm{d}x = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{x^2}{2}\right) \mathrm{d}x \tag{C.4.5}$$

is an integral that cannot be computed in closed form. We can relate it, however, to the incomplete gamma function described in Sect. D.5.

The distribution function of a normal distribution with mean x_0 and variance σ^2 is found from (C.4.5) to be

$$\psi(x) = \psi_0(u) \quad , \qquad u = \frac{x - x_0}{\sigma} \quad . \tag{C.4.6}$$

We now introduce the *error function*

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2}\,\mathrm{d}t \quad , \qquad x \geq 0 \quad . \tag{C.4.7}$$

Comparing with the definition of the incomplete gamma function (D.5.1) gives

$$\mathrm{erf}(x) = \frac{2}{\Gamma(\frac{1}{2})} \int_{t=0}^{t=x} \mathrm{e}^{-t^2}\,\mathrm{d}t = \frac{1}{\Gamma(\frac{1}{2})} \int_{u=0}^{u=x^2} \mathrm{e}^{-u} u^{-1/2}\,\mathrm{d}u \quad ,$$

$$\mathrm{erf}(x) = P\left(\frac{1}{2}, x^2\right) \quad . \tag{C.4.8}$$

On the other hand there is a more direct connection between (C.4.6) and (C.4.7),

$$\psi_{x_0=0,\sigma=1/\sqrt{2}}(x) = \psi_0(u = \sqrt{2}x) = \frac{1}{2}[1 + \mathrm{sign}(x)\mathrm{erf}(|x|)]$$

or

$$\psi_0(u) = \frac{1}{2}\left[1 + \mathrm{sign}(u)\mathrm{erf}\left(\frac{|u|}{\sqrt{2}}\right)\right] \quad ,$$

i.e.,

$$\psi_0(u) = \frac{1}{2}\left[1 + \mathrm{sign}(u)\, P\left(\frac{1}{2}, \frac{u^2}{2}\right)\right] \quad . \tag{C.4.9}$$

The methods StatFunct.cumulativeStandardNormal and StatFunct.cumulativeNormal yield the distribution functions (C.4.4) and (C.4.5), respectively.

Finally we compute the quantiles of the standard normal distribution (using the method StatFunct.quantileStandardnormal). For a given probability P, the quantile x_p is defined by the relation

$$P = \psi_0(x_p) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_p} \mathrm{e}^{-x^2/2}\,\mathrm{d}x \quad . \tag{C.4.10}$$

We determine this by finding the zero of the function

$$k(x, P) = P - \psi_0(x) \tag{C.4.11}$$

using the procedure of Sect. E.2.

For the quantile x_P for a probability P of the normal distribution with mean x_0 and standard deviation σ (computed with StatFunct.quantileNormal) one has

$$P = \psi_0(u_P) \quad , \qquad x_P = x_0 + \sigma u_P \quad . \tag{C.4.12}$$

C.5 χ^2-Distribution

The probability density (6.6.10) of the χ^2-distribution for n degrees of freedom,

$$f(\chi^2) = \frac{1}{2^\lambda \Gamma(\lambda)} (\chi^2)^{\lambda-1} \mathrm{e}^{-\frac{1}{2}\chi^2} \quad , \qquad \lambda = \frac{n}{2} \quad , \tag{C.5.1}$$

is computed by the method StatFunct.chiSquared.

The distribution function

$$\begin{aligned} F(\chi^2) &= \frac{1}{\Gamma(\lambda)} \int_{u=0}^{u=\chi^2} \frac{1}{2} \left(\frac{u}{2}\right)^{\lambda-1} \mathrm{e}^{-\frac{1}{2}u}\,\mathrm{d}u \\ &= \frac{1}{\Gamma(\lambda)} \int_{t=0}^{t=\chi^2/2} \mathrm{e}^{-t} t^{\lambda-1}\,\mathrm{d}t \end{aligned} \tag{C.5.2}$$

is seen from (D.5.1) to be an incomplete gamma function

$$F(\chi^2) = P\left(\lambda, \frac{\chi^2}{2}\right) = P\left(\frac{n}{2}, \frac{\chi^2}{2}\right) \tag{C.5.3}$$

and computed by StatFunct.cumulativeChiSquared.

The quantile χ^2_P of the χ^2-distribution for a given probability P, which is given by

$$h(\chi^2_P) = P - F(\chi^2_P) = 0 \quad , \tag{C.5.4}$$

is computed as the zero of the function $h(\chi\ )$ with StatFunct.quantileChiSquared

C.6 F-Distribution

The probability density (8.2.3) of the F-distribution with f_1 and f_2 degrees of freedom,

$$f(F) = \left(\frac{f_1}{f_2}\right)^{\frac{1}{2}f_1} \frac{\Gamma(\frac{1}{2}(f_1+f_2))}{\Gamma(\frac{1}{2}f_1)\Gamma(\frac{1}{2}f_2)} F^{\frac{1}{2}f_1-1} \left(1+\frac{f_1}{f_2}F\right)^{-\frac{1}{2}(f_1+f_2)} \quad , \tag{C.6.1}$$

is computed with StatFunct.fDistribution. .

The distribution function

$$F(F) = \frac{\Gamma(\frac{1}{2}(f_1+f_2))}{\Gamma(\frac{1}{2}f_1)\Gamma(\frac{1}{2}f_2)} \left(\frac{f_1}{f_2}\right)^{\frac{1}{2}f_1} \int_0^F F^{\frac{1}{2}f_1-1} \left(1+\frac{f_1}{f_2}F\right)^{-\frac{1}{2}(f_1+f_2)} \mathrm{d}F \tag{C.6.2}$$

can be rearranged using

$$t = \frac{f_2}{f_2+f_1F} \quad , \qquad |\mathrm{d}t| = \frac{f_1f_2}{(f_2+f_1F)^2}|\mathrm{d}F|$$

to be

$$\begin{aligned} F(F) &= \frac{1}{B(\frac{1}{2}f_1,\frac{1}{2}f_2)} \int_{t=\frac{f_2}{f_2+f_1F}}^{t=1} (1-t)^{\frac{1}{2}f_1-1} t^{\frac{1}{2}f_2-1}\,\mathrm{d}t \qquad \text{(C.6.3)} \\ &= 1 - I_{f_2/(f_2+f_1F)}\left(\tfrac{1}{2}f_2,\tfrac{1}{2}f_1\right) \quad , \end{aligned}$$

i.e., it is related to the incomplete beta function; cf. (D.6.1). We compute it with the method StatFunct.cumulativeFDistribution.

The quantile F_P of the F-distribution for a given probability P is given by the zero of the function

$$h(F) = P - F(F) \quad . \tag{C.6.4}$$

It is computed by StatFunct.quantileFDistribution.

C.7 t-Distribution

The probability density (8.3.7) of Student's t-distribution with f degrees of freedom,

$$\begin{aligned} f(t) &= \frac{\Gamma(\frac{1}{2}(f+1))}{\Gamma(\frac{1}{2}f)\Gamma(\frac{1}{2})\sqrt{f}} \left(1+\frac{t^2}{f}\right)^{-\frac{1}{2}(f+1)} \qquad \text{(C.7.1)} \\ &= \frac{1}{B(\frac{1}{2},\frac{f}{2})\sqrt{f}} \left(1+\frac{t^2}{f}\right)^{-\frac{1}{2}(f+1)} \quad , \end{aligned}$$

is computed with by StatFunct.Student.

By using the substitution

$$u = \frac{f}{f+t^2} \quad ,$$

the distribution function of the t-distribution can be expressed in terms of the incomplete beta function (D.6.1),

$$\begin{aligned} F(t) &= \frac{1}{B(\frac{1}{2},\frac{f}{2})\sqrt{f}} \int_{-\infty}^{t} \left(1+\frac{t^2}{f}\right)^{-\frac{1}{2}(f+1)} \mathrm{d}t \qquad \text{(C.7.2)} \\ &= \frac{1}{2} + \frac{\mathrm{sign}(t)}{B(\frac{1}{2},\frac{f}{2})\sqrt{f}} \int_{0}^{|t|} \left(1+\frac{t^2}{f}\right)^{-\frac{1}{2}(f+1)} \mathrm{d}t \\ &= \frac{1}{2} + \frac{\mathrm{sign}(t)}{B(\frac{1}{2},\frac{f}{2})\sqrt{f}} \frac{1}{2} \int_{u=f/(f+t^2)}^{u=1} u^{\frac{f}{2}-1}(1-u)^{\frac{1}{2}} \mathrm{d}u \quad , \\ F(t) &= \frac{1}{2}\left\{1+\mathrm{sign}(t)\left[1-I_{f/(f+t^2)}\left(\frac{f}{2},\frac{1}{2}\right)\right]\right\} \quad . \qquad \text{(C.7.3)} \end{aligned}$$

It is computed by the method StatFunct.cumulativeStudent.

The quantile t_P of the t-distribution for a given probability P is computed by finding the zero of the function

$$h(t) = P - F(t) \qquad \text{(C.7.4)}$$

with the method StatFunct.quantileStudent.

C.8 Java Class and Example Program

Java Class for the Computation of Statistical Functions

StatFunct contains all methods mentioned in this Appendix.

Example Program C.1: The class FunctionsDemo demonstrates all methods mentioned in this Appendix

The user first selects a family of functions and then a function from that family. Next the parameters, needed in the chosen case, are entered. After the Go button is clicked, the function value is computed and displayed.

D. The Gamma Function and Related Functions: Methods and Programs for Their Computation

D.1 The Euler Gamma Function

Consider a real number x with $x+1>0$. We define the Euler *gamma function* by

$$\Gamma(x+1)=\int_0^\infty t^x \mathrm{e}^{-t}\,\mathrm{d}t \quad . \tag{D.1.1}$$

Integrating by parts gives

$$\int_0^\infty t^x \mathrm{e}^{-t}\,\mathrm{d}t = [-t^x \mathrm{e}^{-t}]_0^\infty + x\int_0^\infty t^{x-1}\mathrm{e}^{-t}\,\mathrm{d}t = x\int_0^\infty t^{x-1}\mathrm{e}^{-t}\,\mathrm{d}t \quad .$$

Thus one has the relation

$$\Gamma(x+1)=x\Gamma(x) \quad . \tag{D.1.2}$$

This is the so-called *recurrence relation* of the gamma function. From (D.1.1) it follows immediately that

$$\Gamma(1)=1 \quad .$$

With (D.1.2) one then has generally that

$$\Gamma(n+1)=n! \quad , \qquad n=1,2,\ldots \quad . \tag{D.1.3}$$

We now substitute t by $\frac{1}{2}u^2$ (and $\mathrm{d}t$ by $u\,\mathrm{d}u$) and get

$$\Gamma(x+1)=\left(\tfrac{1}{2}\right)^x \int_0^\infty u^{2x+1}\mathrm{e}^{-\frac{1}{2}u^2}\,\mathrm{d}u \quad .$$

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

If we now choose in particular $x = -\frac{1}{2}$, we obtain

$$\Gamma(\tfrac{1}{2}) = \sqrt{2}\int_0^\infty \mathrm{e}^{-\frac{1}{2}u^2}\,\mathrm{d}u = \frac{1}{\sqrt{2}}\int_{-\infty}^\infty \mathrm{e}^{-\frac{1}{2}u^2}\,\mathrm{d}u \quad . \tag{D.1.4}$$

The integral can be evaluated in the following way. We consider

$$A = \int_{-\infty}^\infty\int_{-\infty}^\infty \mathrm{e}^{-\frac{1}{2}(x^2+y^2)}\,\mathrm{d}x\,\mathrm{d}y = \int_{-\infty}^\infty \mathrm{e}^{-\frac{1}{2}x^2}\,\mathrm{d}x\int_{-\infty}^\infty \mathrm{e}^{-\frac{1}{2}y^2}\,\mathrm{d}y = 2\{\Gamma(\tfrac{1}{2})\}^2 \quad .$$

The integral A can transformed into polar coordinates:

$$A = \int_0^{2\pi}\int_0^\infty \mathrm{e}^{-\frac{1}{2}r^2} r\,\mathrm{d}r\,\mathrm{d}\phi = \int_0^{2\pi}\mathrm{d}\phi\int_0^\infty \mathrm{e}^{-\frac{1}{2}r^2} r\,\mathrm{d}r = 2\pi\,\Gamma(1) = 2\pi \quad .$$

Setting the two expressions for A equal gives

$$\Gamma(\tfrac{1}{2}) = \sqrt{\pi} \quad . \tag{D.1.5}$$

Using (D.1.2) we can thus determine the value of the gamma function for half-integral arguments.

For arguments that are not positive integers or half-integers, the integral (D.1.1) cannot be evaluated in closed form. In such cases one must rely on approximations. We discuss here the approximation of LANCZOS [17], which is based on the analytic properties of the gamma function. We first extend the definition of the gamma function to negative arguments by means of the reflection formula

$$\Gamma(1-x) = \frac{\pi}{\Gamma(x)\sin(\pi x)} = \frac{\pi x}{\Gamma(1+x)\sin(\pi x)} \quad . \tag{D.1.6}$$

(By relations (D.1.1) and (D.1.6) the gamma function is also defined for an arbitrary complex argument if x is complex.) One sees immediately that the gamma function has poles at zero and at all negative integer values. The approximation of LANCZOS [17],

$$\Gamma(x+1) = \sqrt{2\pi}\left(x+\gamma+\frac{1}{2}\right)^{x+\frac{1}{2}}\exp\left(-x-\gamma-\frac{1}{2}\right)(A_\gamma(x)+\varepsilon) \quad , \tag{D.1.7}$$

takes into account the first few of these poles by the form of the function A_γ,

$$A_\gamma(x) = c_0 + \frac{c_1}{x+1} + \frac{c_2}{x+2} + \cdots + \frac{c_{\gamma+1}}{x+\gamma+1} \quad . \tag{D.1.8}$$

For $\gamma = 5$ one has for the error ε the approximation $|\varepsilon| < 2 \cdot 10^{-10}$ for all points x in the right half of the complex plane. The method Gamma.gamma yields Euler's gamma function.

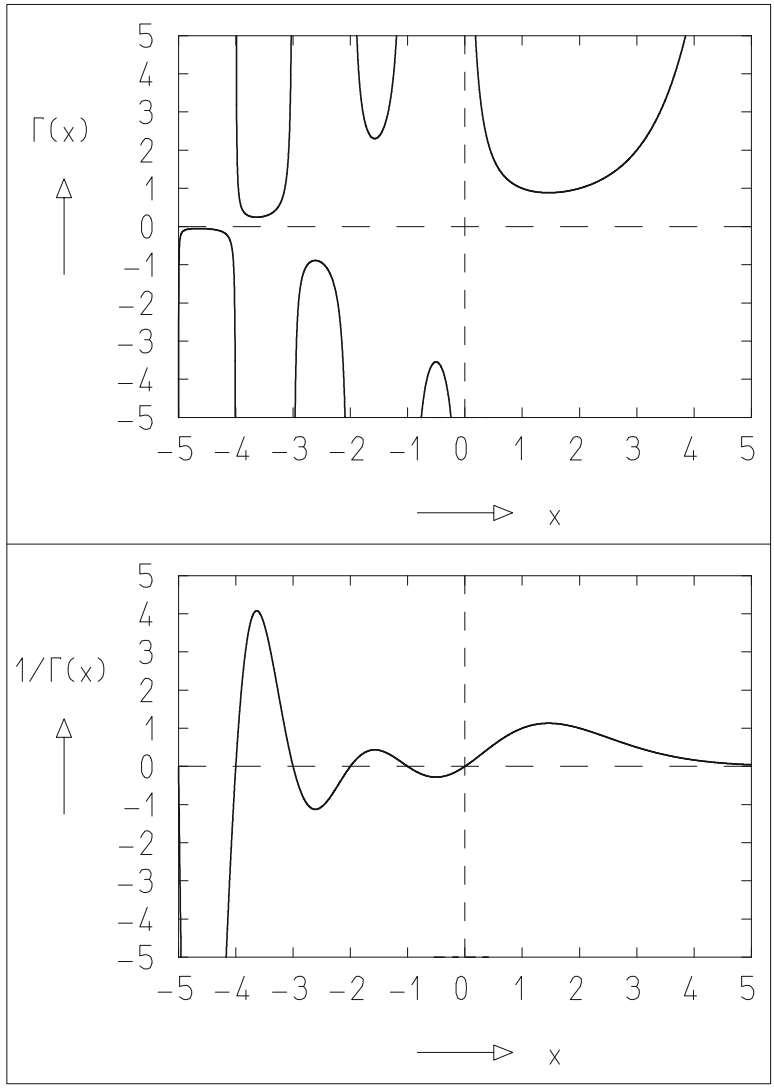

Fig. D.1: The functions $\Gamma(x)$ and $1/\Gamma(x)$.

The gamma function is plotted in Fig. D.1. For large positive arguments the gamma function grows so quickly that it is difficult to represent its value in a computer. In many expressions, however, there appear ratios of gamma functions which have values in a an unproblematic region. In such cases it is better to use the logarithm of the gamma function which is computed by the method Gamma.logGamma.

D.2 Factorial and Binomial Coefficients

The expression

$$n! = 1 \cdot 2 \cdots n \tag{D.2.1}$$

can either be directly computed as a product or as a gamma function using (D.1.3).

When computing binomial coefficients,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n}{k} \cdot \frac{n-1}{k-1} \cdots \frac{n-k+1}{1} \quad , \tag{D.2.2}$$

the expression on the right-hand side is preferable for numerical reasons to the expression in the middle and used in the method Gamma.Binomial. .

D.3 Beta Function

The *beta function* has two arguments and is defined by

$$B(z,w) = \int_0^1 t^{z-1}(1-t)^{w-1}\,\mathrm{d}t \quad . \tag{D.3.1}$$

The integral can be written in a simple way in terms of gamma functions,

$$B(z,w) = B(w,z) = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)} \quad . \tag{D.3.2}$$

In this way the method Gamma.beta computes the beta function. Figure D.2 shows it as a function of w for several fixed values of z.

D.4 Computing Continued Fractions

In the next two sections there appear *continued fractions*, i.e., expressions of the type

$$f = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \frac{a_3}{b_3 + \cdots}}} \quad , \tag{D.4.1}$$

that can also be written in the typographically simpler form

$$f = b_0 + \frac{a_1}{b_1+}\ \frac{a_2}{b_2+}\ \frac{a_3}{b_3+}\cdots \quad . \tag{D.4.2}$$

If we denote by f_n the value of the fraction (D.4.1) truncated after a finite number of terms up to the coefficients a_n and b_n, then one has

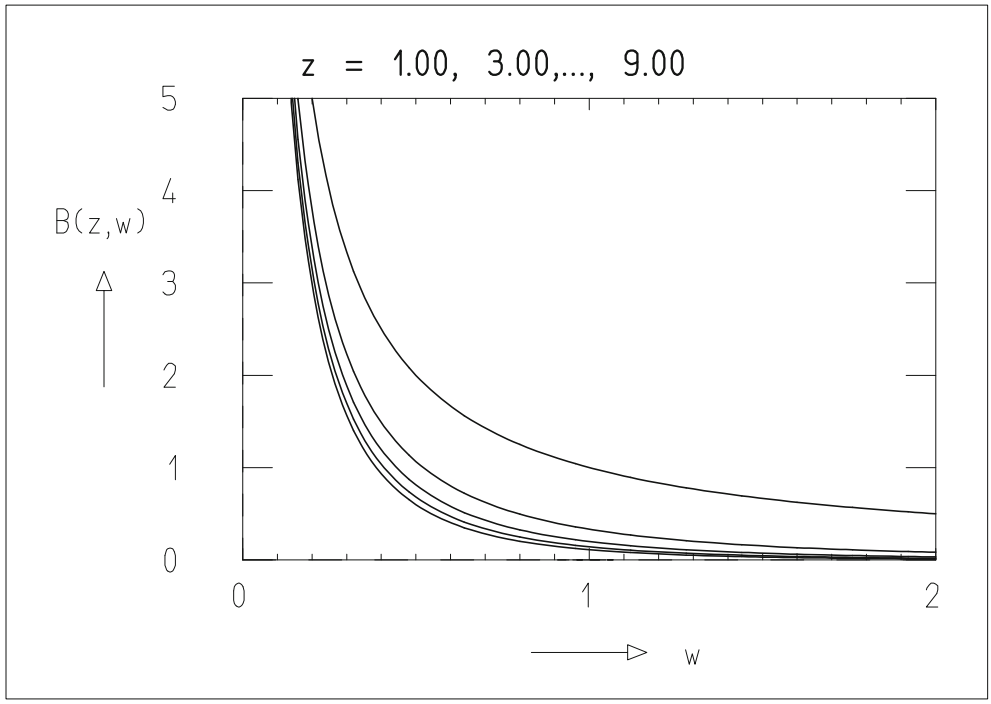

Fig. D.2: The beta function. For increasing z the curves $B(z, w)$ are shifted to the left.

$$f_n = \frac{A_n}{B_n} \quad . \tag{D.4.3}$$

The quantities A_n and B_n can be obtained from the following recursion relation,

$$A_{-1} = 1 \quad , \qquad B_{-1} = 0 \quad , \qquad A_0 = b_0 \quad , \qquad B_0 = 1 \quad , \tag{D.4.4}$$

$$A_j = b_j\, A_{j-1} + a_j\, A_{j-2} \quad , \tag{D.4.5}$$

$$B_j = b_j\, B_{j-1} + a_j\, B_{j-2} \quad . \tag{D.4.6}$$

Since the relations (D.4.5) and (D.4.6) are linear in A_{j-1}, A_{j-2} and B_{j-1}, B_{j-2}, respectively, and since in (D.4.3) only the ratio A_n/B_n appears, one can always multiply the coefficients A_j, A_{j-1}, A_{j-2} and B_j, B_{j-1}, B_{j-2} by an arbitrary normalization factor. One usually chooses for this factor $1/B_j$ and avoids in this way numerical difficulties from very large or very small numbers, which would otherwise occur in the course of the recursion. For steps in which $B_j = 0$, the normalization is not done.

Continued fractions, in a way similar to series expansions, appear in approximations of certain functions. In a region where the approximation for the continued fraction converges, the values of f_{n-1} and f_n for sufficiently

large n do not differ much. One can therefore use the following *truncation criterion*. If for a given $\varepsilon \ll 1$ the inequality

$$\left| \frac{f_n - f_{n-1}}{f_n} \right| < \varepsilon$$

holds, then f_n is a sufficiently good approximation of f.

D.5 Incomplete Gamma Function

The *incomplete gamma function* is defined for $a > 0$ by the expression

$$P(a,x) = \frac{1}{\Gamma(a)} \int_0^x \mathrm{e}^{-t} t^{a-1} \,\mathrm{d}t \quad . \tag{D.5.1}$$

It can be expressed as a series expansion

$$P(a,x) = x^a \mathrm{e}^{-x} \sum_{n=0}^{\infty} \frac{x^n}{\Gamma(a+n+1)} = \frac{1}{\Gamma(a)} x^a \mathrm{e}^{-x} \sum_{n=0}^{\infty} \frac{\Gamma(a)}{\Gamma(a+n+1)} x^n \quad . \tag{D.5.2}$$

The sum converges quickly for $x < a+1$. One uses the right-hand, not the middle form of (D.5.2), since the ratio of the two gamma functions reduces to

$$\frac{\Gamma(a)}{\Gamma(a+n+1)} = \frac{1}{a} \frac{1}{a+1} \cdots \frac{1}{a+n+1} \quad .$$

In the region $x > a+1$ we use the continued fraction

$$1 - P(a,x) = \frac{1}{\Gamma(a)} \mathrm{e}^{-x} x^a \left(\frac{1}{x+} \frac{1-a}{1+} \frac{1}{x+} \frac{2-a}{1+} \frac{2}{x+} \cdots \right) \quad . \tag{D.5.3}$$

The method Gamma Incompletegamma yields the incomplete gamma function. It is shown in Fig. D.3 for several values of a. From the figure one sees immediately that

$$P(a,0) = 0 \quad , \tag{D.5.4}$$

$$\lim_{x \to \infty} P(a,x) = 1 \quad . \tag{D.5.5}$$

D.6 Incomplete Beta Function

The *incomplete beta function* is defined for $a > 0$, $b > 0$ by the relation

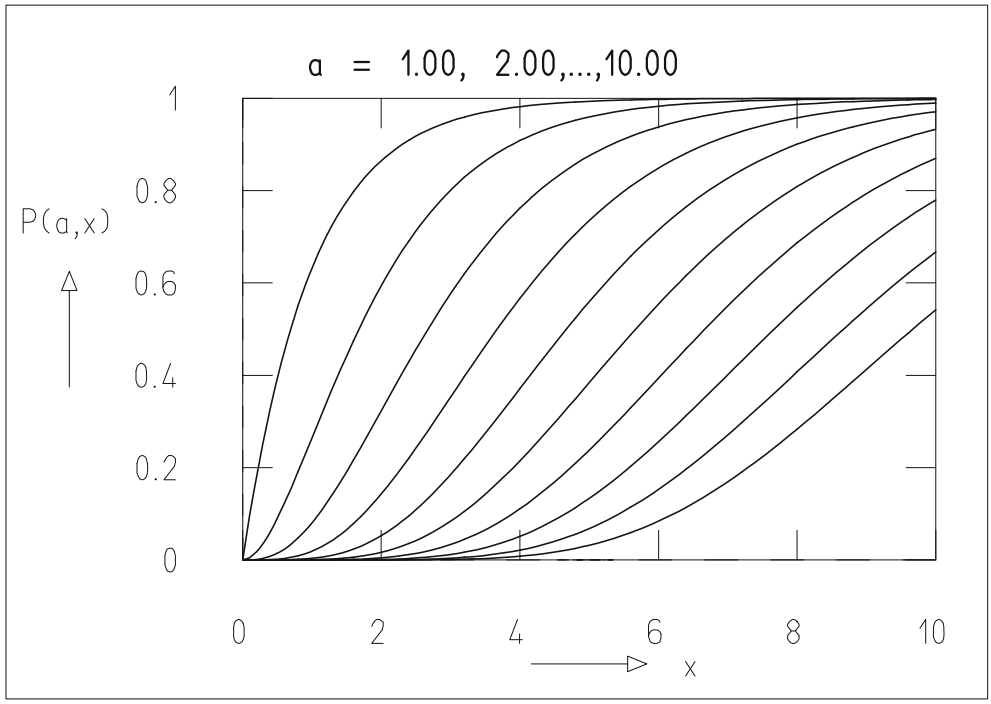

Fig. D.3: The incomplete gamma function. With increasing a the graphs $P(a,x)$ move to the right.

$$I_x(a,b) = \frac{1}{B(a,b)} \int_0^x t^{a-1}(1-t)^{b-1}\,\mathrm{d}t \quad , \qquad x \le 0 \le 1 \quad . \tag{D.6.1}$$

The function obeys the symmetry relation

$$I_x(a,b) = 1 - I_{1-x}(b,a) \quad . \tag{D.6.2}$$

The expression (D.6.1) can be approximated by the following continued fraction:

$$I_x(a,b) = \frac{x^a(1-x)^b}{aB(a,b)} \left\{ \frac{1}{1+} \frac{d_1}{1+} \frac{d_2}{1+} \cdots \right\} \tag{D.6.3}$$

with

$$\begin{aligned} d_{2m+1} &= -\frac{(a+m)(a+b+m)}{(a+2m)(a+2m-1)}\,x \quad , \\ d_{2m} &= \frac{m(b-m)}{(a+2m-1)(a+2m)}\,x \quad . \end{aligned}$$

The approximation converges quickly for

$$x > \frac{a+1}{a+b+1} \quad . \tag{D.6.4}$$

If this requirement is not fulfilled, then $1-x$ is greater than the right-hand side of (D.6.4). In this case one computes I_{1-x} as a continued fraction and then uses (D.6.2).

The method Gamma.incompleteBezta computes the incomplete beta function. In Fig. D.4 it is s displayed for various values of the parameters a and b. Regardless of these parameters one has

$$I_0(a,b) = 0 \quad , \qquad I_1(a,b) = 1 \quad . \tag{D.6.5}$$

D.7 Java Class and Example Program

Java Class for for the Computation of the Gamma Function and Related Functions

Gamma contains all methods mentioned in this Appendix.

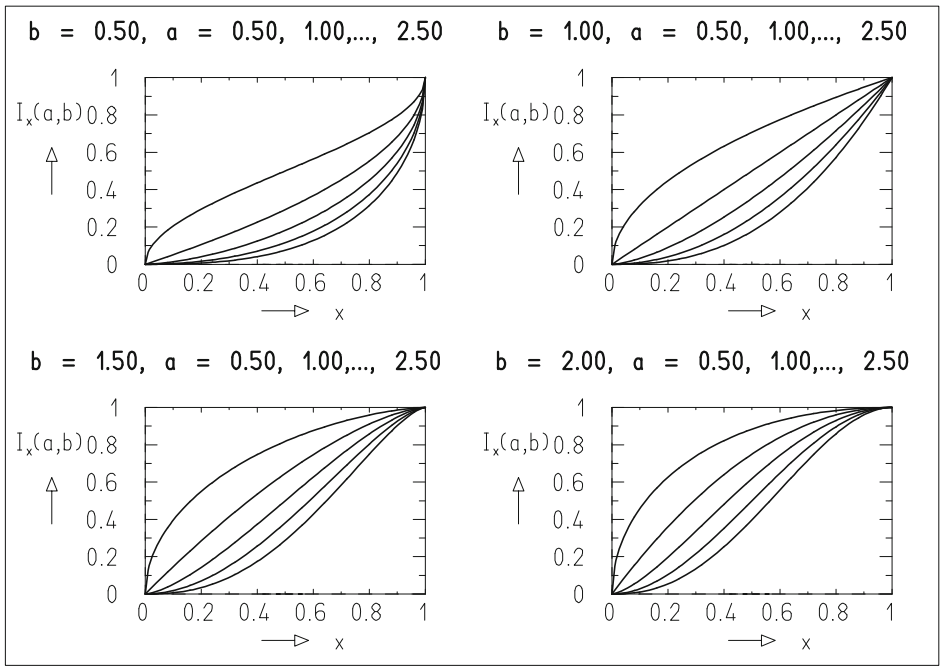

Fig. D.4: The incomplete beta function. With increasing a the curves $I_x(a,b)$ move further to the right.

Example Program D.1: The class FunctionsDemo demonstrates not only the methods of Appendix C but also all methods mentioned in the present Appendix

The user first selects a family of functions and the a function from that family. Next the parameters, needed in the chosen case, are entered. After the Go button is clicked, the function value is computed and displayed.

E. Utility Programs

E.1 Numerical Differentiation

The derivative $\mathrm{d}f(x)/\mathrm{d}x$ of a function $f(x)$ at the point x is given by the limit

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \quad .$$

Obviously one can approximate $f'(x)$ by

$$\frac{f(x+h) - f(x)}{h}$$

for a small finite value of h. In fact, for this the symmetrical difference ratio

$$\delta(h) = \frac{f(x+h) - f(x-h)}{2h} \tag{E.1.1}$$

is more appropriate. This can be seen from the Taylor expansions at the point x for $f(x+h)$ and $f(x-h)$, which give

$$\delta(h) = f'(x) + \frac{h^2}{3!} f'''(x) + \frac{h^4}{5!} f^{(5)}(x) + \cdots \quad ,$$

in which the leading additional term is already quadratic in h. Nevertheless, the choice of h is still critical, since for very small values of h there occur large rounding errors, and for larger values the approximation may not be valid.

One can compute $\delta(h)$ for a monotonic sequence of values $h = h_0, h_1, h_2, \ldots$. If the sequence $\delta(h_0)$, $\delta(h_1)$, ... is monotonic (rising or falling) then this is a sign of convergence of the series to $f'(x)$. According to RUTISHAUSER [27], from the series $\delta(h_0)$, $\delta(h_1)$ one can also obtain others that converge more quickly. The method was modeled after the ROMBERG procedure [28] for numerical integration. Starting from $h_0 = a$ one first chooses the sequence

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

$$h_0, h_1, \ldots = a,\ 3a/4,\ a/2,\ 3a/8,\ a/4, \ldots \quad ,$$

sets

$$T_0^{(k)} = \delta(h_k) \quad ,$$

and computes the additional quantities $T_m^{(k)}$

$$T_m^{(k)} = \frac{2^m T_{m-1}^{(k+1)} - 1.125\, T_{m-1}^{(k)}}{2^m - 1.125} \quad ; \qquad m \text{ odd}, k \text{ even} \quad ,$$

$$T_m^{(k)} = \frac{2^m \cdot 1.125\, T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{2^m \cdot 1.125 - 1} \quad ; \qquad m \text{ odd}, k \text{ odd} \quad ,$$

$$T_m^{(k)} = \frac{2^m T_{m-1}^{(k+1)} - T_{m-1}(k)}{2^m - 1} \quad ; \qquad m \text{ even} \quad .$$

Arranging the quantities $T_m^{(k)}$ in the form of a triangle,

$$\begin{array}{llll} T_0^{(0)} & & & \\ & T_1^{(0)} & & \\ T_0^{(1)} & & T_2^{(0)} & \\ & T_1^{(1)} & & T_3^{(0)} \\ T_0^{(2)} & & T_2^{(1)} & \\ & T_1^{(2)} & & \\ T_0^{(3)} & & & \\ \vdots & & & \end{array}$$

the first column contains the sequence of our original difference ratios. Not only does $T_0^{(k)}$ converge to $f'(x)$, but one has in general

$$\lim_{k\to\infty} T_m^{(k)} = f'(x) \quad , \qquad \lim_{m\to\infty} T_m^{(k)} = f'(x) \quad .$$

The practical significance of the procedure is based on the fact that the columns on the right converge particularly quickly.

In the class AuxDer, the sequence $T_0^{(0)}, \ldots, T_0^{(9)}$ is computed starting from $a = 1$. If it is not monotonic, then a is replaced by $a/10$ and a new sequence is computed. After 10 tries without success the procedure is terminated. If, however, a monotonic sequence is found, the triangle scheme is computed and $T_9^{(0)}$ is given as the best approximation for $f'(x)$. The class `AuxDer` is similar to the program of KOELBIG [29] with the exception of minor changes in the termination criteria.

This program requires considerable computing time. For well behaved functions it is often sufficient to replace the differential ratio by the difference ratio (E.1.1). To compute second derivatives the procedure of difference ratios is extended correspondingly. The classes AuxDri, AuxGrad and AuxHesse therefore operate on the basis of difference ratios.

E.2 Numerical Determination of Zeros

Computing the *quantile* x_P of a distribution function $F(x)$ for a given probability P is equivalent to determining the zero of the function

$$k(x) = P - F(x) \quad . \tag{E.2.1}$$

We treat the problem in two steps. In the first step we determine an interval (x_0, x_1) that contains the zero. In the second step we systematically reduce the interval such that its size becomes smaller than a given value ε.

In the first step we make use of the fact that $k(x)$ is monotonic, since $f(x)$ is monotonic. We begin with initial values for x_0 and x_1. If $f(x_0) \cdot f(x_1) < 0$, i.e., if the function values have different signs, then the zero is contained within the interval. If this is not the case, then we enlarge the interval in the direction where the function has the smallest absolute value, and repeat the procedure with the new values of (x_0, x_1).

For localizing the zero within the initial interval (x_0, x_1) we use a comparatively slow but absolutely reliable procedure. The original interval is divided in half and replaced by the half for which the end points are of the opposite sign. The procedure is repeated until the interval width decreases below a given value.

This technique is implemented in the class AuxZero. It is also employed in several methods for the computation of quantiles in StatFunct in a direct way, i.e., without a call to AuxZero. An example for the application of AuxZero is the class E1MaxLike, see also Example Program 7.1.

E.3 Interactive Input and Output Under Java

Java programs usually are *event driven*, i.e., while running they react to actions by the user. Thus an interaction between user and program is enabled. Its detailed design depends on the problem at hand and also on the user's taste. For our Example Programs four utility programs may suffice to establish simple interactions. They are explained in Fig. E.1.

It shows a screen window produced by the class DatanFrame. In its simplest form it consists only of a frame, a title line (here 'Example for the creation of random numbers') and an output region, into which the user's output can be written. The method DatanFra.add allows to add additional elements below the title line which will be arranged horizontally starting from the left. In Fig. E.1 these are an *input group*, a *radio-button group* and a *go button*. The input group is created by the class AuxJInputGroup and the radio-button group by AuxJRButtonGroup. Both (as well as DatanFra) make use of the standard Java Swing classes. The go button is

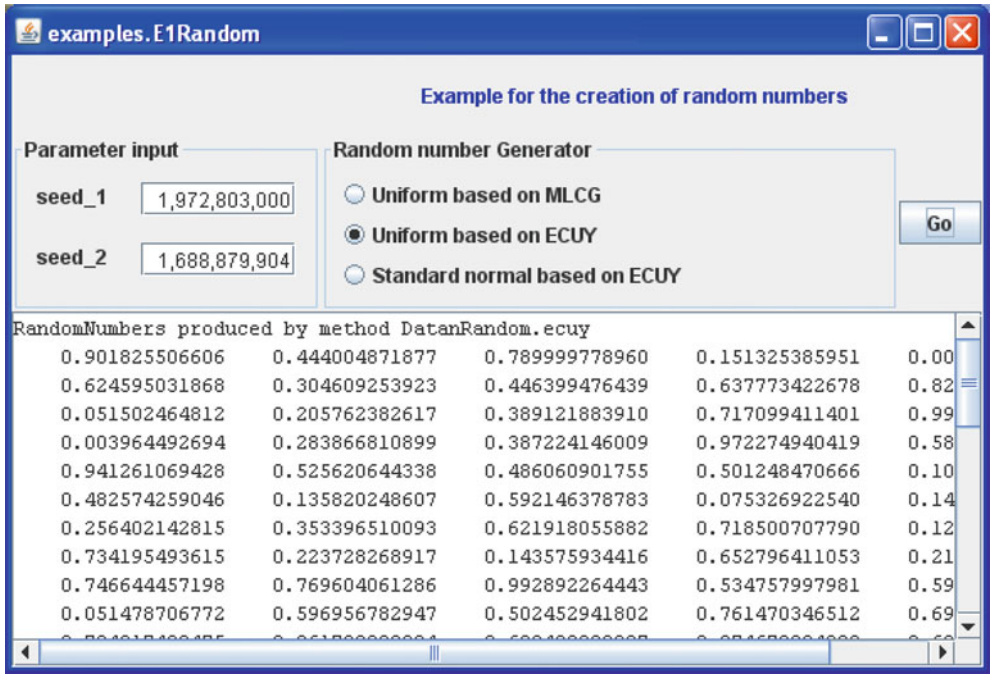

Fig. E.1: A window of the type `DatanFrame` with elements for interactive input, for starting the program, and for alphanumeric output of results.

directly created by a standard class. The input group itself is composed of an arbitrary number of *number-input regions*, arranged vertically one below the other, which are created by the class AuxJNumberInput. The detailed usage of these classes is summarized in the online documentation. we also recommend to study the source code of some of our Example Programs.

E.4 Java Classes

AuxDer computes the derivative of a function using the Rutishauser method.

AuxDri computes the matrix A of derivatives required by LsqNon and LsqMar.

AuxGrad computes the gradient of a function at a given point.

AuxHesse computes the Hessian matrix of a function at a given point.

AuxZero finds the zero of a monotonic function.

DatanFrame creates a screen window with possibilities for interactive input and output.

AuxJInputGroup creates an input group within a screen window.

AuxJInputNumber creates a number-input region within an input group.

AuxJRButtonGroup creates a radio-button group within a screen window.

F. The Graphics Class DatanGraphics

F.1 Introductory Remarks

The graphical display of data and of curves of fitted functions has always been an important aid in data analysis. Here we present the class DatanGraphics comprising methods which produce graphics in screen windows and/or postscript files. We distinguish control, transformation, drawing, and auxilliary methods. All will be described in detail in this Appendix and there usage will be explained in a number of Example Programs. For many purposes, however, it is sufficient to use one of only five classes which, in turn, resort to DatanGraphics. These classes, by a single call, produce complete graphical structures; they are listed at the beginning of Sect. F.8.

F.2 Graphical Workstations: Control Routines

As mentioned, a plot can be output either as in the form of a screen window or as a file in postscript format. The latter is easily embedded in digital documents or directly printed on paper, if necessary after conversion to another format such as pdf, by the use of a freely available program. For historical reasons we call both the screen window and the postscript file a *graphics workstation*.

The method DatanGraphics.openWorkstation"opens" a screen window or a file or both, i.e., it initializes buffers into which information is written by methods mentioned later. Only after the method DatanGraphics.closeWorstation has been called, is the window presented on the screen and/or is the postscript file made available. In this way several graphics can be produced one after another. There windows can coexist on

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

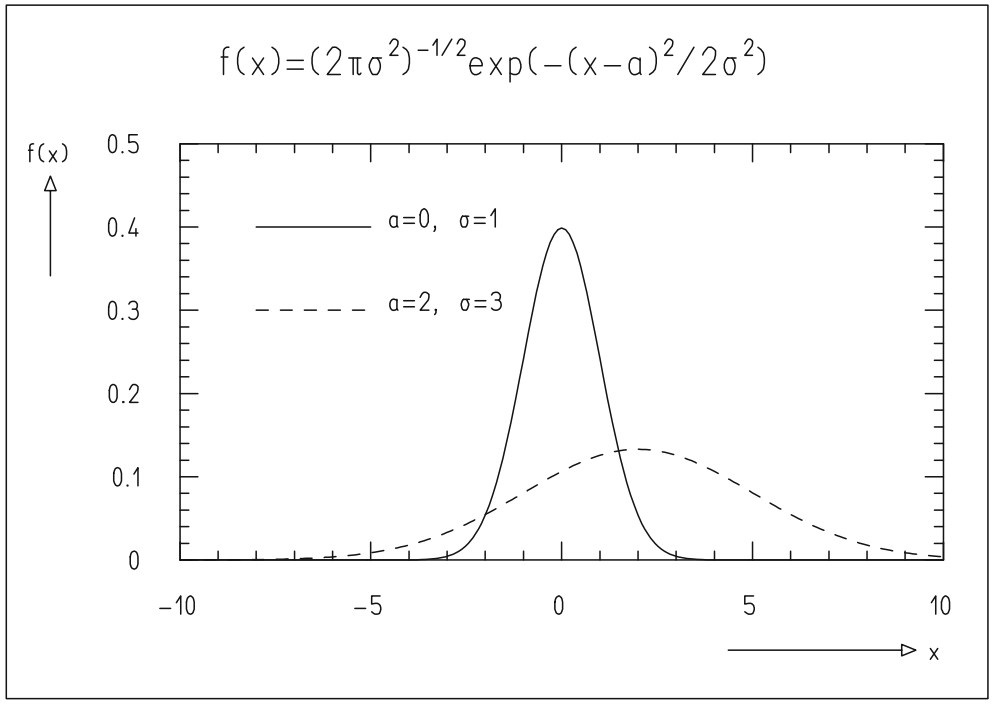

Fig. F.1: Simple example of a plot produced with DatanGraphics.

the screen. They be changed in size using the computer mouse; but their contents is not alterable.

F.3 Coordinate Systems, Transformations and Transformation Methods

F.3.1 Coordinate Systems

World Coordinates (WC)

Figure F.1 shows a plot made by DatanGraphics. Let us imagine for a moment that all of the graphical structures, including text, physically exist, e.g., that they are made out of bent wire. The coordinate system in which this wire structure is described is called the world coordinate system (WC). The coordinates of a point in the WC are denoted by (X, Y).

Computing Coordinates (CC)

If we consider the axes in Fig. F.1, we note that the axes designated with x and y have about the same length in world coordinates, but have very different lengths in terms of the numbers shown. The figure shows a plot of a function

$$y = f(x) \quad .$$

Each point (x, y) appears at the point (X, Y). We call the coordinate system of the (x, y) points the computing coordinate system (CC). The transformation between WC and CC is given by Eq. (F.3.1).

Device Coordinates (DC)

From the (fictitious) world coordinate system, the plot must be brought onto the working surface of a graphics device (terminal screen or paper). We call the coordinates (u, v) on this surface the device coordinates (DC).

F.3.2 Linear Transformations: Window – Viewport

The concepts defined in this section and the individual transformations are illustrated in Fig. F.2.

Let us assume that the computing coordinates in x cover the range

$$x_a \leq x \leq x_b \quad .$$

The corresponding range in world coordinates is

$$X_a \leq X \leq X_b \quad .$$

A linear transformation $x \rightarrow X$ is therefore defined by

$$X = X_a + (x - x_a)\frac{X_b - X_a}{x_b - x_a} \quad . \tag{F.3.1}$$

The transformation for $y \rightarrow Y$ is defined in a corresponding way. One speaks of the mapping of the *window* in computing coordinates CC,

$$x_a \leq x \leq x_b \quad , \qquad y_a \leq y \leq y_b \quad , \tag{F.3.2}$$

onto the *viewport* in world coordinates WC,

$$X_a \leq X \leq X_b \quad , \qquad Y_a \leq Y \leq Y_b \quad . \tag{F.3.3}$$

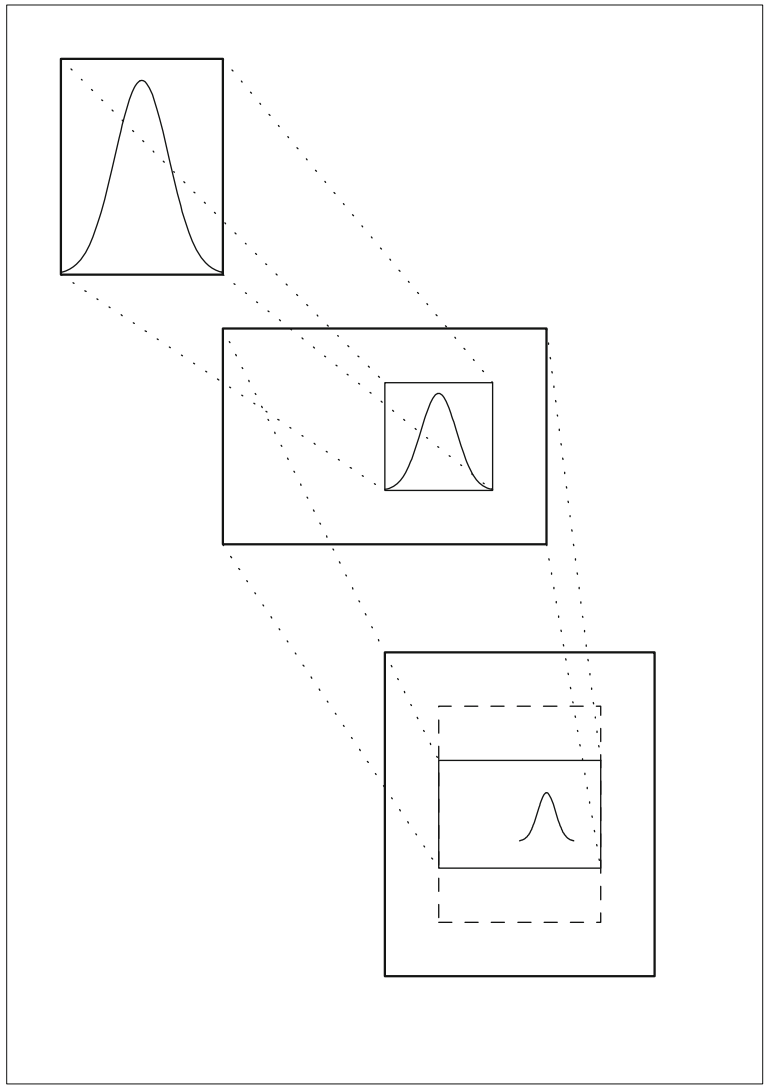

Fig. F.2: The various coordinate systems. *Above*: window in computing coordinates. *Middle*: viewport (*small rectangle*) and window (*large rectangle*) in world coordinates. *Below*: preliminary viewport (*dashed rectangle*), final adjusted viewport (*small rectangle*), and border of the display surface (*large rectangle*) in device coordinates. The mappings from computing coordinates to world coordinates and from world coordinates to device coordinates are indicated by *dotted lines*.

The mapping is in general *distorted*, i.e., a square in CC becomes a rectangle in WC. It is undistorted only if the aspect ratios of the window and viewport are equal,

$$\frac{x_b - x_a}{y_b - y_a} = \frac{X_b - X_a}{Y_b - Y_a} \quad . \tag{F.3.4}$$

The user of DatanGraphics computes values first in CC that are to be plotted. By specifying the window and viewport he or she defines the mapping into WC. A possible linear distortion is perfectly acceptable in this step, since it provides a simple means of changing the scale.

Next a mapping onto the physically implemented device coordinates DC must be done. It can of course be defined by again providing a window (in WC) and a viewport (in DC). One wants, however, to avoid an additional distortion. We define, therefore, a viewport

$$u_a \leq u \leq u_b \quad , \qquad v_a \leq v \leq v_b \tag{F.3.5}$$

and a window

$$X'_a \leq X \leq X'_b \quad , \qquad Y'_a \leq Y \leq Y'_b \quad . \tag{F.3.6}$$

The mapping from the window (F.3.6) onto the viewport (F.3.5) is only carried out if both have the same aspect ratio. Otherwise the viewport (F.3.6) is reduced symmetrically in width to the right and left or in height symmetrically above and below such that the viewport

$$u'_a \leq u \leq u'_b \quad , \qquad v'_a \leq v \leq v'_b \tag{F.3.7}$$

has the same aspect ratio as the window (F.3.6). In this way a distortion free mapping is defined between the two.

F.4 Transformation Methods

The user of DatanGraphics must define the transformations between the various coordinate systems by calling the appropriate routines. The transformations are then applied when drawing a graphical structure without any further intervention.

Transformation CC → WC

This transformation is defined by calling the following two methods. Datan Graphics.setWindowInComputingCoordinates sets the window in computing coodinates, Datan Coordinates sets the viewport in world coordinates.

Transformation WC → DC

This transformation is defined by calling the following two methods. Datan-Graphics.setWindowInWorldCoordinates sets the window

in world coordinates. DatanGraphics.setFormat defines the temporary viewport in device coordinates. Into that the final view port is fitted so that the width-to-height ratio is the same as for the window in world coordinates. If DatanGraphics.setFormat is not called at all, the format is taken to be A5 landscape. If the workstation is a screen window, then only the width-to-height ratio is taken into account. In the case of a postscript file the absolute size in centimeters is valid only if a plot of that size will fit on the paper in the printer. Otherwise the plot is demagnified until it just fits. In both cases the plot is centered on the paper. A call to the method DatanGraphics.setStandardPaperSizeAndBorders informs the program about the paper size. If it is not called, then A4 is assumed with a margin of 5 mm on all 4 sides.

In most cases, having defined the transformations in this way, the user will be interested only in computing coordinates.

Clipping

The graphical structures are not completely drawn under certain circumstances. They are truncated if they extend past the boundary of the so-called *clipping region*. The structures are said to be *clipped*. For polylines, markers, data points, and contour lines (Sect. F.5) the clipping region is the window in computing coordinates; for text and graphics utility structures the clipping region is the window in world coordinates. These regions can can be set explicitely using DatanGraphics.setSmallClippingWindow and DatanGraphics.setBigClippingWindow, respectively.

F.5 Drawing Methods

Colors and Line Widths

The methods mentioned up to this point carry out organizational tasks, but do not, however, produce any graphical structures on the workstation. All of the graphical structures created by DatanGraphics consist of lines. The lines possess a given color and width. The selection of these two attributes is done as follows. A pair of properties (color, linewidth) is assigned to each of set of 8 *color indices*. The set of 8 is different for screen window and postscript file. With the method DatanGraphics.chooseColor the user selects one particular color index. That then is valid until another color index is chosen. The user may assign his own choice of color and line width to a color index by the methods DatanGraphics.setScreenColor and/or DatanGraphics.setPSColor For the background of the screen

window (the standard is blue) another color can be chosen with DatanGraphics.setScreenBackground. For the postscript file the background is always transparent.

Polylines

The concepts of a polyline and polymarker have been introduced for particularly simple graphics structures. A *polyline* defines a sequence of line segments from the point (x_1, y_1) through the points (x_2, y_2), (x_3, y_3), ... to the point (x_n, y_n). A polyline is drwan with the method DatanGraphics.Polyline.

A *polymarker* marks a plotting point with a graphical symbol. The polymarkers available in DatanGraphics are shown in Fig. F.3.

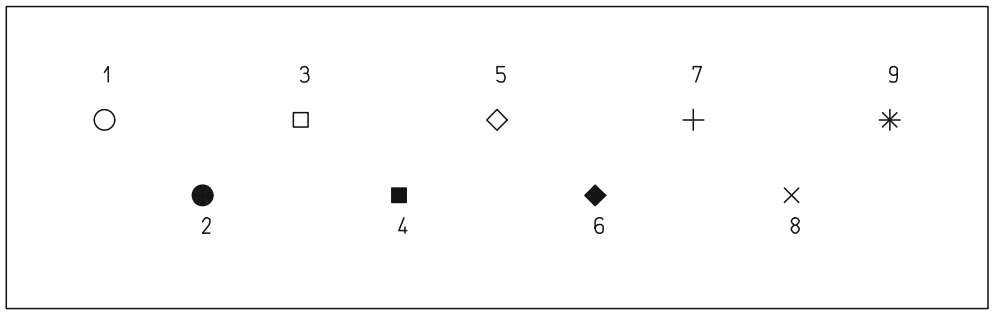

Fig. F.3: Polymarkers.

One can clearly achieve an arbitrarily good approximation of any graphical structure by means of polylines. For example, *graphs of functions* can be displayed with polylines as long as the individual points are sufficiently close to each other.

Sometimes one wants to draw a polyline not as a solid line but rather *dashed*, *dotted*, or *dot-dashed*. That is achieved by the method DatanGraphics.drawBrokenPolyline.

Polymarkers are especially suitable for marking data points. If a data point has error bars, then one would like to indicate these errors in one or both coordinates by means of *error bars*. In certain circumstances one would even like to show the complete covariance ellipse. This task is performed by the method DatanGraphics.drawDatapoint. Examples are shown in Fig. F.4.

An error bar in the x direction is only drawn if $\sigma_x > 0$, and in the y direction only if $\sigma_y > 0$. The covariance ellipse is only draw if $\sigma_x > 0, \sigma_y > 0$, and $\mathrm{cov}(x, y) \neq 0$. Error bars are not drawn if they would lie completely

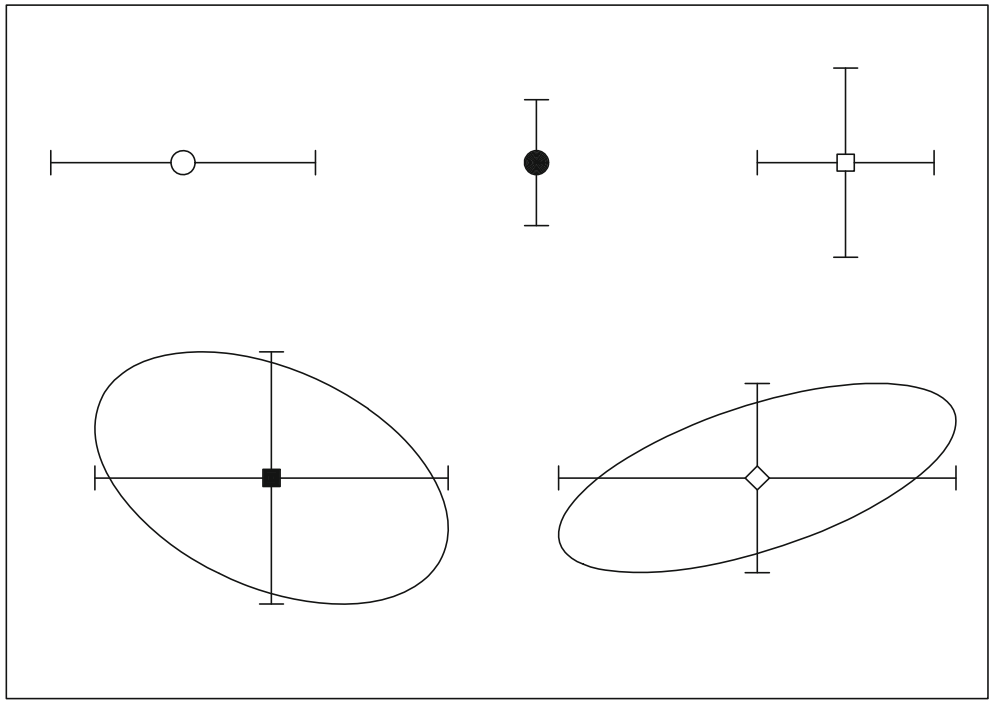

Fig. F.4: Example for plotting data points.

within the polymarker itself. If part of the structure falls outside of the CC window (F.3.2), then this part is not drawn.

Histogram

The method DatanGraphics.drawHistogram displays data in the form of a histogram.

Contour Lines

A function $f = f(x, y)$ defines a surface in a three dimensional (x, y, f) space. One can also get an idea of the function in the (x, y) plane by marking points for which $f(x, y)$ is equal to a given constant c. The set of all such points forms the contour line $f(x, y) = c$. By drawing a set of such contour lines $f(x, y) = c_1, c_2, \ldots$ one obtains (as with a good topographical map) a rather good impression of the function.

Naturally it is impossible to compute the function for all points in the (x, y) plane. We restrict ourselves to a rectangular region in the (x, y) plane, usually the window in computing coordinates, and we break it into a total of

$N = n_x n_y$ smaller rectangles. The corner points of these smaller rectangles have x coordinates that are neighboring values in the sequence

$$x_0, x_0 + \Delta x, x_0 + 2\Delta x, \ldots, x_0 + n_x \Delta x \quad .$$

The y coordinates of the corner points are adjacent points of the sequence

$$y_0, y_0 + \Delta y, y_0 + 2\Delta y, \ldots, y_0 + n_y \Delta y \quad .$$

In each rectangle the contour line is approximated linearly. To do this one considers the function $f(x, y) - c$ at the four corner points. If the function is of a different sign at two corner points that are the end points of an edge of the rectangle, then it is assumed that the contour lines intersect the edge. The intersection point is computed with linear interpolation. If the intersection points are on two edges of the rectangle, then they are joined by a line segment. If there are intersection points on more than two edges, then all pairs of such points are joined by line segments.

Clearly the approximation of the contour lines by line segments becomes better for a finer division into small rectangles. With a finer division, of course, the required computing time also becomes longer.

The method DatanGraphics.drawContour computes and draws a contour line. An example of a function represented by contour lines is shown in Fig. F.5.

F.6 Utility Methods

With the few methods described up to this point, a great variety of complicated plots can be produced. By using the methods of this section and the next, the tasks are made easier for the user, since they help to create graphical structures typically used in conjunction with the plots of data analysis, such as axes, coordinate crosses, and explanatory text.

Frames

The methode DataGrpahics.drawFrame draws a frame around the plotted part of world coordinate system, i.e., the outer frame of the plots reproduced here. The metod DatanGraphics.drawBoundary, on the other hand, draws a frame around the window of the cumputing coordinate system.

Scales

The method DatanGraphics.drawScaleX draws a scale in x direction. Ticks appear at the upper and lower edge of the CC window pointing to the

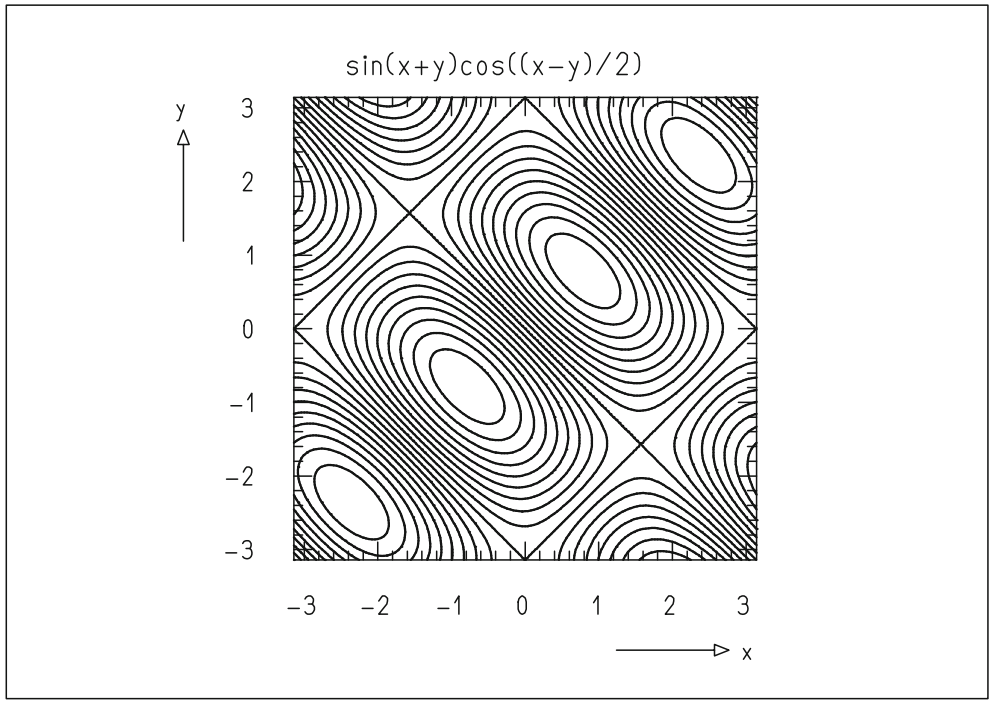

Fig. F.5: Contour lines $f(x, y) = -0.9, -0.8, \ldots, 0.8, 0.9$ of the function $f(x) = \sin(x + y)\cos([x - y]/2)$ in the (x, y) plane.

inside of the window. Below the lower edge numbers appear, marking some of these ticks. It is recommended to first call the method DatanGraphics.drawBoundary, to mark the edges themselves by lines. In addition an arrow with text can be drawn, showing in the direction of increasing x values. The method DatanGraphics.drawScaleY performs analogous tasks for an axis in y direction.

The creation of axis divisions and labels with these methods is usually done automatically without intervention of the user. Sometimes, however, the user will want to influence these operations. This can be done by using the method DatanGraphics.setParametersForScale. A call to this method influences only that scale which is generated by the very next call of DatanGraphics.drawScaleX or DatanGraphics.drawScaleY, respectively.

Coordinate Cross

The method DatanGraphics.drawCoordinateCross draws a coordinate cross in the computing coordinate system. The axes of that system appear as broken lines inside the CC window.

F.7 Text Within the Plot

Explanatory text makes plots easier to understand. The methods in this section create text superimposed on a plot which can be placed at any location.

The text must be supplied by the user as a character string. Before this text is translated into graphics characters, however, it must first be encoded. The simple encoding system used here allows the user to display simple mathematical formulas. For this there are *three character sets*: Roman, Greek, and mathematics, as shown in Table F.1. The character set is selected by *control characters*. These are the special characters

@ for Roman,
& for Greek,
% for mathematics.

A control character in the text string causes all of the following characters to be produced with the corresponding character set, until another control symbol appears. The default character set is Roman.

In addition there exist the following *positioning symbols:*

^ for superscript (exponent),
_ for subscript (index),
for normal height,
" for backspace.

All characters appear at normal height as long as no positioning symbol has appeared. One can move a maximum of two steps from normal height, e.g., A_{α_β}, A_{α^β}. The positioning symbols ^ and _ remain in effect until the appearance of a #. The symbol " acts only on the character following it. This then appears over the previous character instead of after it. In this way one obtains, e.g., A_α^β instead of $A_\alpha{}^\beta$.

The method DatanGraphics.drawCaption draws a caption, centered slightly below the upper edge of the plotted section of the world coordinate system.

Sometimes the user wants to write text at a certain place in the plot, e.g., next to an individual curve or data point, and also to choose the text size. This is made possible by the method DatanGraphics.drawText .

F.8 Java Classes and Example Programs

Java Classes Poducing Graphics

DatanGraphics contains the methods mentioned in this Appendix.

Table F.1: The various character sets for producing text.

Control characters				Control characters			
	Roman	Greek	Math		Roman	Greek	Math
Input	@	&	%	Input	@	&	%
A	A	A(ALPHA)	Ä	a	a	α(alpha)	ä
B	B	B(BETA)	B	b	b	β(beta)	b
C	C	X(CHI)	⌐	c	c	χ(chi)	c
D	D	Δ(DELTA)	Δ	d	d	δ(delta)	d
E	E	E(EPSILON)	E	e	e	ϵ(epsilon)	e
F	F	Φ(PHI)	F	f	f	φ(phi)	f
G	G	Γ(GAMMA)	≠	g	g	γ(gamma)	g
H	H	H(ETA)	H	h	h	η(eta)	h
I	I	I(IOTA)	∫	i	i	ι(iota)	i
J	J	I(IOTA)	J	j	j	ι(iota)	j
K	K	K(KAPPA)	K	k	k	κ(kappa)	k
L	L	Λ(LAMBDA)	|	l	l	λ(lambda)	l
M	M	M(MU)	±	m	m	μ(mu)	m
N	N	N(NU)	N	n	n	ν(nu)	n
O	O	Ω(OMEGA)	Ö	o	o	ω(omega)	ö
P	P	Π(PI)	Ö	p	p	π(pi)	p
Q	Q	Θ(THETA)	Q	q	q	ϑ(theta)	q
R	R	R(RHO)	∘	r	r	ρ(rho)	r
S	S	Σ(SIGMA)	ß	s	s	σ(sigma)	s
T	T	T(TAU)	¦	t	t	τ(tau)	t
U	U	O(OMICRON)	Ü	u	u	o(omicron)	ü
V	V		Ü	v	v		v
W	W	Ψ(PSI)	√	w	w	ψ(psi)	w
X	X	Ξ(XI)	X	x	x	ξ(xi)	x
Y	Y	Υ(UPSILON)	Å	y	y	υ(upsilon)	y
Z	Z	Z(ZETA)	Z	z	z	ζ(zeta)	z
~	~	~	~	-	-	-	-
!	!	!	!	=	=	=	≡
$	$	$	$	{	{	{	{
*	*	#	×	}	}	}	}
(	(	↑	←	|	|	|	|
)	)	↓	→	[	[	&	[
+	+	+	+	]	]	@	]
‘	‘	‘	‘	\			
1	1	1	1	:	:	:	:
2	2	2	2	;	;	;	;
3	3	3	3	’	’	‘	’
4	4	4	4	<	<	⊂	≤
5	5	5	5	>	>	⊃	≥
6	6	6	6	?	?	§	~
7	7	7	7	,	,	,	,
8	8	8	8	.	.	.	.
9	9	9	9	/	/	\	%
0	0	0	0				

GraphicsWithHistogram produces a complete plot with a histogram (an Example Program is E2Sample).

GraphicsWith2DScatterDiagram produces a complete plot with a two-dimensional scatter diagram (an Example Program is E3Sample).

GraphicsWithHistogramAndPolylin eproduces a complete graphics with a histogram and a polyline (an Example Program isE6Gr).

GraphicsWithDataPointsAndPolyline produces a complete plot with data points and one polyline (an Example Program is E7Gr).

GraphicsWithDataPointAndMultiplePolylines produces a complete plot with data points and several polylines (an Example Program is E8Gr).

Example Program F.1: The class E1Gr demonstrates the use of the following methods of the class DatanGraphics:
openWorkstation, closeWorkstation,
setWindowInComputingCoordinates,
setViewportInWorldCoordinates,
setWindowInWorldCoordinates, setFormat,
drawFrame, drawBoundary, chooseColor,
drawPolyline, drawBrokenPolyline, drawScaleX,
drawScaleY, drawCaption, drawText

The program generates the simple plot of Fig. F.1. It opens the workstation and defines the different coordinate systems. The outer frame is drawn (enclosing the section of the world coordinate system to be displayed) and the inner frame (the boudary of the computing coordinate system). Next, the lettered scales for abswcissa and ordinate are produced as is a caption for the plot. Now, the color index is changed. In a short loop a total of 201 coordinate pairs (x_i, y_i) are computed with $x_i = -10, -9.9, -9.8, \ldots, 10$ and $y_i = f(x_i)$. The function $f(x)$ is the probability density of the standardized normal distribution. A polyline, defined by these pairs is drawn. In a second loop the points for a polyline are computed which correspond to a normal distribution with with mean $a = 2$ and standard deviation $\sigma = 3$. That polyline is represented as a broken line. Finally, two short straight lines are displayed in the upper left corner of the plot (one as a solid line and one as a dashed line). To the right each of these polylines a short text is displayed, indicating the parameters of the Gaussians displayed as solid and dashed curves, respectively. Before termination of the program the workstation is closed.

Example Program F.2: The class E2Gr demonstrates the use of the method DatanGraphics.drawMark.

The short program generates the plot of Fig. F.3, showing the different polymarkers, which can be drawn with DatanGraphics.drawMark.

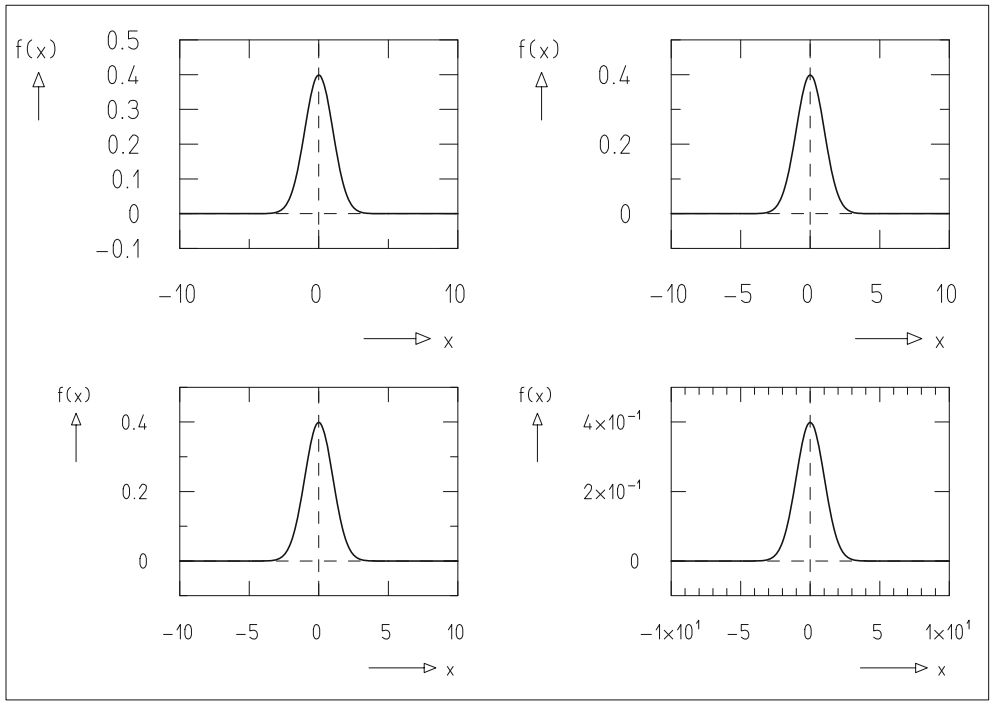

Fig. F.6: Four versions of the same plot with different types of scales.

Example Program F.3: The class E3Gr demonstrates the use of the method DatanGraphics.drawDataPoint.

The program produces the plot of Fig. F.4, which contains examples for the different ways to present data points with errors.

Example Program F.4: The class E4Gr demonstrates the use of the method DatanGrpahics.drawContour

A window of computing coordinates $-\pi \leq x \leq \pi$, $-\pi \leq y \leq \pi$ and a square viewport in world coordinates are selected. After creating scales and the caption, input parameters for DatanGraphics.drawContour are prepared. Next by successive calls of of this method in a loop, contours of the function $f(x, y) = \sin(x + y)\cos((x - y)/2)$ are drawn. The result is a plot corresponding to Fig. F.5.

Suggestions: Extend the program such that the parameters ncont and nstep defining the number of contours and the number of intervals in x and y can be set interactively by the user. Study the changes in the plot resulting from very small values of nstep.

Example Program F.5: The class E5Gr demonstrates the methods DatanGraphics.setParametersForSale and DatanGraphics.drawCoordinateCross

The program generates the plots shown in Fig. F.6. It contains four plots which differ only by the design of their scales. The plots are generated in a loop where different

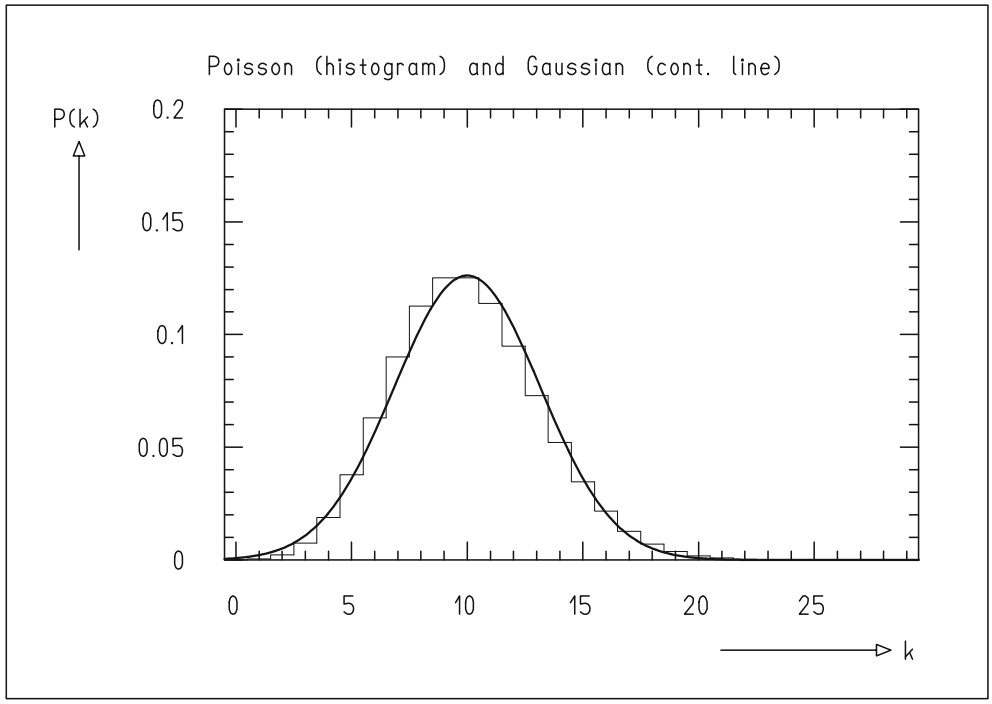

Fig. F.7: A plot generated with GraphicsWithHistogramAndPolyline containing a histogram and a polyline.

viewports in world coordinates are chosen in each step, so that the plots correspond to the upper-left, upper-right, lower-left, and lower-right quadrant of the window in world coordinates. For the upper-left plot the default values for the scale design are used. In the upper-right plot the number of ticks and the lettering of the scale is predefined. In the lower-left plot the size of the symbols used in the lettering of the scales is changed. In the lower-right plot the numbers are written in exponential notation. All plots contain a coordinate cross, which is generated by calling DatanGraphics.drawCoordinateCross and a curve corresponding to a Gaussian.

Example Program F.6: The class E6Gr demonstrates the use of the class GraphicsWithHistogramAndPolyline

The program first sets up a histogram which for each bin k contains the Poisson probability $f(k;\lambda)$ for the parameter $\lambda = 10$. Next, points on a polyline are computed corresponding to the probability density of a normal distribution with mean λ and variance λ. Finally the text strings for the plot are defined and the complete plot is displayed by a call of GraphicsWithHistogramAndPolyline (Fig. F.7).

Example Program F.7: The class E7Gr demonstrates the use of the class GraphicsWithDataPointsAndPolyline

First, by calling DatanRandom.line, data points are generated which lie on a straight line $y = at + b$ within the simulated errors. Next, the errors to be presented

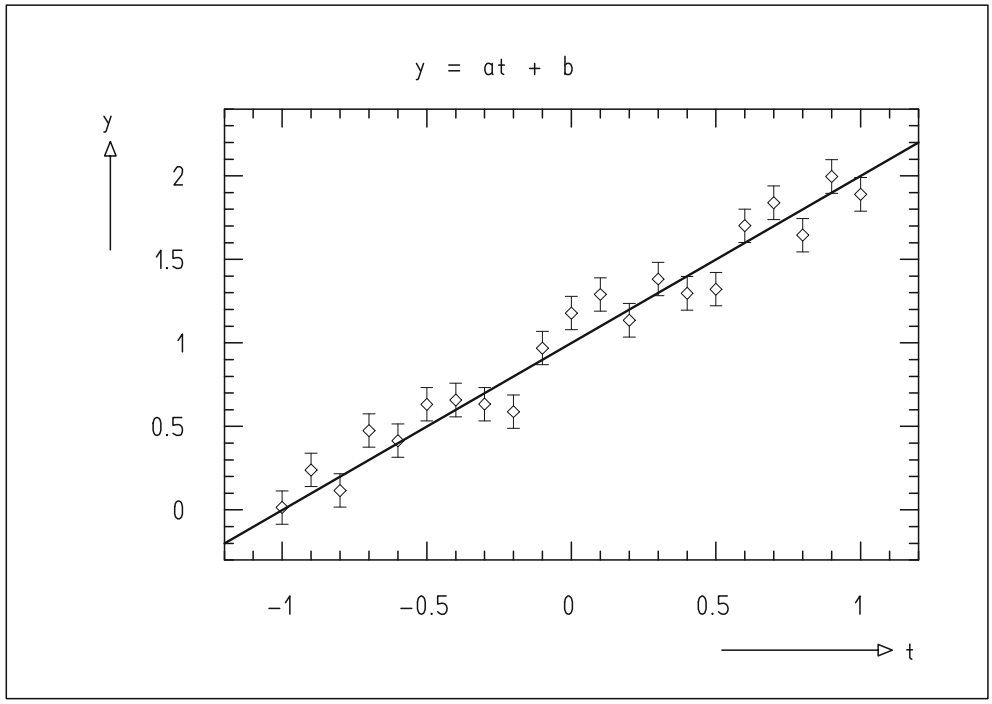

Fig. F.8: A plot with data points and a polyline generated by calling GraphicsWith DataPointsAndPolyline .

in the directions of the horizontal and vertical axes and their covariance are defined. The latter two quantities in our example are equal to zero. The polyline defining the straight line consists of only two points. Their computation is trivial. After the definition of the axis labels and the caption, the plot is displayed by calling GraphicsWithDataPointsAndPolyline (Fig. F.8).

Example Program F.8: The class E8Gr demonstrates the use of the class GraphicsWithDataPointsAndMultiplePolylines

The program generates 21 data points which lie within the simulated errors on a Gaussian curve with zero mean and standard deviation $\sigma = 1$, and which span the abscissa region $-3 \leq x \leq 3$. Next, points on three polylines are computed corresponding to Gaussian curves with means of zero and standard deviations $\sigma = 0.5$, $\sigma = 1$, and $\sigma = 1.5$. The polylines span the abscissa region $-10 \leq x \leq 10$. They are displayed in different colors. One polyline is shown as a continuous line, the other two as dashed lines. Three plots are produced: The first displays only the data points, the second only the polylines, and the third shows the data points together with the polylines. In this way the automatic choice of the scales in the different cases is demonstrated.

G. Problems, Hints and Solutions, and Programming Problems

G.1 Problems

Problem 2.1: Determination of Probabilities through Symmetry Considerations

There are n students in a classroom. What is the probability for the fact that at least two of them have their birthday on the same day? Solve the problem by working through the following questions:

(a) What is the number N of possibilities to distribute the n birthdays over the year (365 days)?

(b) How large is the number N' of possibilities for which all n birthdays are different?

(c) How large then is the probability P_{diff} that the birthdays are different?

(d) How large finally is the probability P that at least two birthdays are not different?

Problem 2.2: Probability for Non-exclusive Events

The probabilities $P(A)$, $P(B)$, and $P(AB) \neq 0$ for non-exclusive events A and B are given. How large is the probability $P(A+B)$ for the observation of A or B? As an example compute the probability that a playing card which was drawn at random out of a deck of 52 cards is either an ace or a diamond.

Problem 2.3: Dependent and Independent Events

Are the events A and B that a playing card out of a deck is an ace or a diamond independent

(a) If an ordinary deck of 52 cards is used,

(b) If a joker is added to the deck?

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

Problem 2.4: Complementary Events

Show that $\bar{A}$ and $\bar{B}$ are independent if A and B are independent. Use the result of Problem 2.2 to express $P(\overline{AB})$ by $P(A)$, $P(B)$, and $P(AB)$.

Problem 2.5: Probabilities Drawn from Large and Small Populations

A container holds a large number (>1000) of coins. They are divided into three types A, B, and C, which make up 20, 30, and 50 % of the total.

(a) What are the probabilities $P(A)$, $P(B)$, $P(C)$ of picking a coin of type A, B, or C if one coin is taken at random? What are the probabilities $P(AB)$, $P(AC)$, $P(BC)$, $P(AA)$, $P(BB)$, $P(CC)$, P(2 identical coins), P(2 different coins) for picking 2 coins?

(b) What are the probabilities if 10 coins (2 of type A, 3 of type B, and 5 of type C) are in the container?

Problem 3.1: Mean, Variance, and Skewness of a Discrete Distribution

The throwing of a die yields as possible results $x_i = 1, 2, \ldots, 6$. For an ideally symmetric die one has $p_i = P(x_i) = 1/6$, $i = 1, 2, \ldots, 6$. Determine the expectation value $\widehat{x}$, the variance $\sigma^2(x) = \mu_2$, and the skewness γ of the distribution,

(a) For an ideally symmetric die,

(b) For a die with

$$p_1 = \frac{1}{6} \quad , \qquad p_2 = \frac{1}{12} \quad , \qquad p_3 = \frac{1}{12} \quad ,$$
$$p_4 = \frac{1}{6} \quad , \qquad p_5 = \frac{3}{12} \quad , \qquad p_6 = \frac{3}{12} \quad .$$

Problem 3.2: Mean, Mode, Median, and Variance of a Continuous Distribution

Consider the probability density $f(x)$ of a *triangular distribution* of the form shown in Fig. G.1, given by

$$\begin{aligned} f(x) &= 0 \quad , \qquad x < a \quad , \qquad x \geq b \quad , \\ f(x) &= \frac{2}{(b-a)(c-a)}(x-a) \quad , \qquad a \leq x < c \quad , \\ f(x) &= \frac{2}{(b-a)(b-c)}(b-x) \quad , \qquad c \leq x < b \quad . \end{aligned}$$

Determine the mean $\widehat{x}$, the mode x_m, the median $x_{0.5}$, and the variance σ^2 of the distribution. For simplicity choose $c = 0$ (which corresponds to the substitution of x by $x' = x - c$). Give explicit results for the symmetric case $a = -b$ and for the case $a = -2b$.

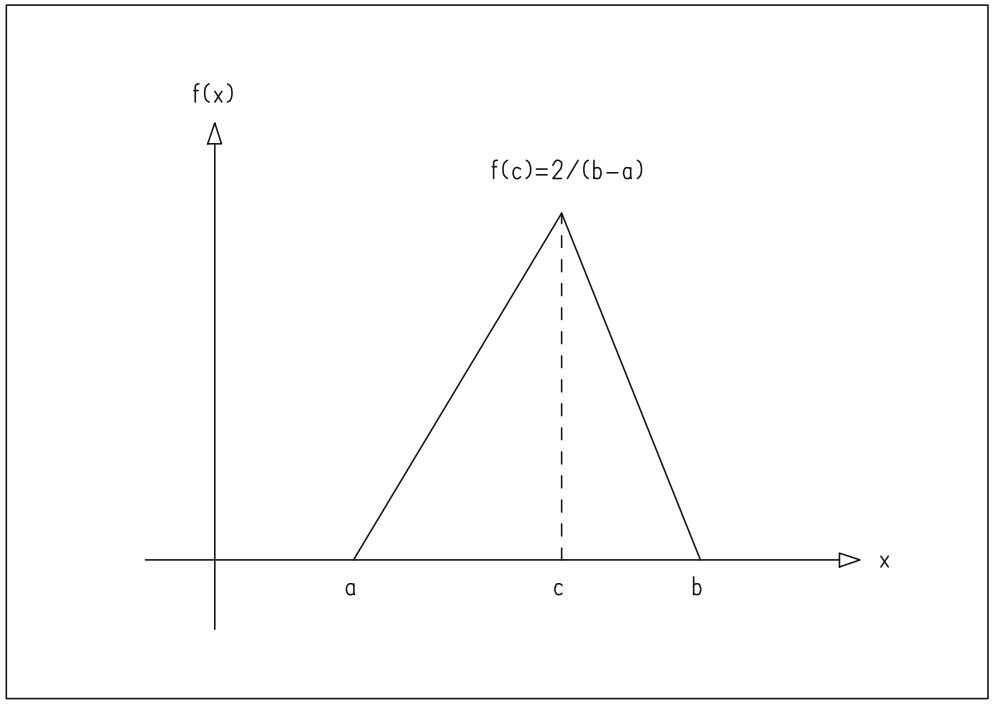

Fig. G.1: Triangular probability density.

Problem 3.3: Transformation of a Single Variable

In Appendix D it is shown that

$$\int_{-\infty}^{\infty} \exp(-x^2/2)\,\mathrm{d}x = \sqrt{2\pi} \quad .$$

Use the transformation $y = x/\sigma$ to show that

$$\int_{-\infty}^{\infty} \exp(-x^2/2\sigma^2)\,\mathrm{d}x = \sigma\sqrt{2\pi} \quad .$$

Problem 3.4: Transformation of Several Variables

A "normal distribution of two variables" (see Sect. 5.10) can take the form

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma_x^2} - \frac{1}{2}\frac{y^2}{\sigma_y^2}\right) \quad .$$

(a) Determine the marginal probability densities $f(x)$, $f(y)$ by using the results of Problem 3.3.

(b) Are x and y independent?

(c) Transform the distribution $f(x, y)$ to the variables

$$u = x\cos\phi + y\sin\phi \quad , \qquad v = y\cos\phi - x\sin\phi \quad .$$

(The u, v coordinate system has the same origin as the x, y coordinate system, but it is rotated with respect to the latter by an angle ϕ.)
Hint: Show that the transformation is orthogonal and use (3.8.12).

(d) Show that u and v are independent variables only if $\phi = 0°, 90°, 180°, 270°$ or if $\sigma_x = \sigma_y = \sigma$.

(e) Consider the case $\sigma_x = \sigma_y = \sigma$, i.e.,

$$f(x) = \frac{1}{2\pi\sigma^2}\exp\left[-\frac{1}{2\sigma^2}(x^2 + y^2)\right] \quad .$$

Transform the distribution to polar coordinates r, ϕ, determine the marginal probability densities $g(r)$ and $g(\phi)$ and show that r and ϕ are independent.

Problem 3.5: Error Propagation

The period T of a pendulum is given by $T = 2\pi\sqrt{\ell/g}$. Here ℓ is the length of the pendulum and g is the gravitational acceleration. Compute g and Δg using the measured values $\ell = 99.8\,\text{cm}$, $\Delta\ell = 0.3\,\text{cm}$, $T = 2.03\,\text{s}$, $\Delta T = 0.05\,\text{s}$ and assuming that the measurements of ℓ and T are uncorrelated.

Problem 3.6: Covariance and Correlation

We denote the mass and the velocity of an object by m and v and their measurement errors by $\Delta m = \sqrt{\sigma^2(m)}$ and $\Delta v = \sqrt{\sigma^2(v)}$. The measurements are assumed to be independent, i.e., $\text{cov}(m, v) = 0$. Furthermore, the relative errors of measurement are known, i.e.,

$$\Delta m/m = a \quad , \qquad \Delta v/v = b \quad .$$

(a) Consider the momentum $p = mv$ and the kinetic energy $E = \frac{1}{2}mv^2$ of the object and compute $\sigma^2(p)$, $\sigma^2(E)$, $\text{cov}(p, E)$, and the correlation $\rho(p, E)$. Discuss $\rho(p, E)$ for the special cases $a = 0$ and $b = 0$. Hint: Form the vectors $\mathbf{x} = (m, v)$ and $\mathbf{y} = (p, E)$. Then approximate $\mathbf{y} = \mathbf{y}(\mathbf{x})$ by a linear transformation and finally compute the covariance matrix.

(b) For the case where the measured values of E, p and the covariance matrix are known, compute the mass m and its error by error propagation. Use the results from (a) to verify your result. Note that you will obtain the correct result only if $\text{cov}(p, E)$ is taken into account in the error propagation.

Problem 5.1: Binomial Distribution

(a) Prove the recursion formula

$$W_{k+1}^n = \frac{n-k}{k+1}\frac{p}{q}W_k^n \quad .$$

(b) It may be known for a certain production process that a fraction $q = 0.2$ of all pieces produced are defective. This means that in 5 pieces produced the expected number of non-defective pieces is $np = n(1-q) = 5 \cdot 0.8 = 4$. What is the probability P_2 and the probability P_3 that at most 2 or at most 3 pieces are free from defects? Use relation (a) to simplify the calculation.

(c) Determine the value k_m for which the binomial distribution is maximum, i.e., k_m is the most probable value of the distribution. Hint: Since W_k^n is not a function of a continuous variable k, the maximum cannot be found by looking for a zero in the derivative. Therefore, one has to study finite differences $W_k^n - W_{k-1}^n$.

(d) In Sect. 5.1 the binomial distribution was constructed by considering the random variable $\mathsf{x} = \sum_{i=1}^n \mathsf{x}_i$. Here x_i was a random variable that took only the values 0 and 1 with the probabilities $P(\mathsf{x}_i = 1) = p$ and $P(\mathsf{x}_i = 0) = q$.

The binomial distribution

$$f(x) = f(k) = W_k^n = \binom{n}{k} p^k q^{n-k}$$

was then obtained by considering in detail the probability to have k cases of $\mathsf{x}_i = 1$ in a total of n observations of the variable x_i. Obtain the binomial distribution in a more formal way by constructing the characteristic function φ_{x_i} of the variable x_i. From the nth power of φ_{x_i} you obtain the characteristic function of x. Hint: Use the binomial theorem (B.6).

Problem 5.2: Poisson Distribution

In a certain hospital the doctor on duty is called on the average three times per night. The number of calls may be considered to be Poisson distributed. What is the probability for the doctor to have a completely quiet night?

Problem 5.3: Normal Distribution

The resistance R of electrical resistors produced by a particular machine may be described by a normal distribution with mean R_m and standard deviation σ.

The production cost for one resistor is C, and the price is $5C$ if $R = R_0 \pm \Delta_1$, and $2C$ if $R_0 - \Delta_2 < R < R_0 - \Delta_1$ or $R_0 + \Delta_1 < R < R_0 + \Delta_2$. Resistors outside these limits cannot be sold.

(a) Determine the profit P per resistor produced for $R_m = R_0$, $\Delta_1 = a_1 R_0$, $\Delta_2 = a_2 R_0$, $\sigma = b R_0$. Use the distribution function ψ_0.

(b) Use Table I.2 to compute the numerical values of P for $a_1 = 0.01$, $a_2 = 0.05$, $b = 0.05$.

(c) Show that the probability density (5.7.1) has points of inflection (i.e., second derivative equal to zero) at $x = a \pm b$.

Problem 5.4: Multivariate Normal Distribution

A planar xy coordinate system is used as target. The probability to observe a hit in the plane may be given by the normal distribution of Problem 3.4 (e). Use the result of that problem to determine

(a) The probability $P(R)$, to observe a hit within a given radius R around the origin,

(b) The radius R, within which a hit is observed with a given probability. Compute as a numerical example the value of R for $P = 90\,\%$ and $\sigma = 1$.

Problem 5.5: Convolution

(a) Prove the relation (5.11.11). Begin with (5.11.9) and use the expression (5.11.10) for $f_y(y)$. In the intervals $0 \le u < 1$ and $2 \le u < 3$ relations (5.11.10a) and (5.11.10b) hold, since in these intervals one always has $y < 1$ and $y > 1$, respectively. In the interval $1 \le u < 2$ the resulting distribution $f(u)$ must be constructed as sum of two integrals of the type (5.11.9) of which each contains one of the possible expression for $f_y(y)$. In this case particular care is necessary in the determination of the limits of integration. They are given by the limits of the intervals in which u and $f_y(y)$ are defined.

(b) Prove the relation (5.11.15) by performing the integration (5.11.5) for the case that f_x and f_y are normal distributions with means 0 and standard deviations σ_x and σ_y.

Problem 6.1: Efficiency of Estimators

Let $\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3$ be the elements of a sample from a continuous population with unknown mean $\widehat{x}$, but known variance σ^2.

(a) Show that the following quantities are unbiased estimators of $\widehat{x}$,

$$\begin{aligned} \mathsf{S}_1 &= \tfrac{1}{4}\mathsf{x}_1 + \tfrac{1}{4}\mathsf{x}_2 + \tfrac{1}{2}\mathsf{x}_3 \quad , \\ \mathsf{S}_2 &= \tfrac{1}{5}\mathsf{x}_1 + \tfrac{2}{5}\mathsf{x}_2 + \tfrac{2}{5}\mathsf{x}_3 \quad , \\ \mathsf{S}_3 &= \tfrac{1}{6}\mathsf{x}_1 + \tfrac{1}{3}\mathsf{x}_2 + \tfrac{1}{2}\mathsf{x}_3 \quad . \end{aligned}$$

Hint: It is simple to show that $\mathsf{S} = \sum_{i=1}^{n} a_i \mathsf{x}_i$ is unbiased if $\sum_{i=1}^{n} a_i = 1$ holds.

(b) Determine the variances $\sigma^2(\mathsf{S}_1)$, $\sigma^2(\mathsf{S}_2)$, $\sigma^2(\mathsf{S}_3)$ using (3.8.7) and the assumption that the elements $\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3$ are independent.

(c) Show that the arithmetic mean $\bar{\mathsf{x}} = \frac{1}{3}\mathsf{x}_1 + \frac{1}{3}\mathsf{x}_2 + \frac{1}{3}\mathsf{x}_3$ has the smallest variance of all estimators of the type $\mathsf{S} = \sum_{i=1}^{3} a_i \mathsf{x}_i$ fulfilling the requirement $\sum_{i=1}^{3} a_i = 1$.
Hint: Minimize the variance of $\mathsf{S} = a_1\mathsf{x}_1 + a_2\mathsf{x}_2 + (1 - a_1 - a_2)\mathsf{x}_3$ with respect to a_1 and a_2. Compute this variance and compare it with the variances which you found (b).

Problem 6.2: Sample Mean and Sample Variance

Compute the sample mean $\bar{\mathsf{x}}$, the sample variance s^2, and an estimate for the variance of the sample mean $\mathsf{s}^2_{\bar{\mathsf{x}}} = (1/n)\mathsf{s}^2$ for the following sample:

$$18,\ 21,\ 23,\ 19,\ 20,\ 21,\ 20,\ 19,\ 20,\ 17.$$

Use the method of Example 6.1.

Problem 6.3: Samples from a Partitioned Population

An opinion poll is performed on an upcoming election. In our (artificially constructed) example the population is partitioned into three subpopulations, and from each subpopulation a preliminary sample of size 10 is drawn. Each element of the sample can have the values 0 (vote for party A) or 1 (vote for party B). The samples are

$$\begin{aligned} i = 1 \quad & (p_i = 0.1) : x_{ij} = 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, \\ i = 2 \quad & (p_i = 0.7) : x_{ij} = 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, \\ i = 3 \quad & (p_i = 0.2) : x_{ij} = 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, \end{aligned}$$

(a) Use these samples to form an estimator for the result of the election and its variance $\mathsf{s}^2_{\tilde{\mathsf{x}}}$. Does this result show a clear advantage for one party?

(b) Use these samples to determine for a much larger sample of size n the sizes n_i of the subsamples in such a way that $\tilde{x}$ has the smallest variance (cf. Example 6.6).

Problem 6.4: χ^2-distribution

(a) Determine the skewness $\gamma = \mu_3/\sigma^3$ of the χ^2-distribution using (5.5.7). Begin by expressing μ_3 by λ_3, $\widehat{x}$, and $E(\mathsf{x}^2)$.

(b) Show that $\gamma \to 0$ for $n \to \infty$.

Problem 6.5: Histogram

Construct a histogram from the following measured values:

26.02, 27.13, 24.78, 26.19, 22.57, 25.16, 24.39, 22.73, 25.35, 26.13,
23.15, 26.29, 24.89, 23.76, 26.04, 22.03, 27.48, 25.42, 24.27, 26.58,
27.17, 24.22, 21.86, 26.53, 27.09, 24.73, 26.12, 28.04, 22.78, 25.36.

Use a bin size of $\Delta x = 1$.
Hint: For each value draw a cross of width Δx and height 1. In this way you do not need to order the measured values, since each cross is drawn within a bin on top of a preceding cross. In this way the bars of the histogram grow while drawing.

Problem 7.1: Maximum-Likelihood Estimates

(a) Suppose it is known that a random variable x follows the uniform distribution $f(x) = 1/b$ for $0 < x < b$. The parameter b is to be estimated from a sample. Show that $\mathsf{S} = \widetilde{b} = \mathsf{x}_{\max}$ is the maximum-likelihood estimator of b. (Hint: This result cannot be obtained by differentiation, but from a simple consideration about the likelihood function).

(b) Write down the likelihood equations for the two parameters a and Γ of the Lorentz distribution (see Example 3.5). Show that these do not necessarily have unique solutions. You can, however, easily convince yourself that for $|\mathsf{x}^{(j)} - a| \ll \Gamma$ the arithmetic mean $\bar{\mathsf{x}}$ is an estimator of a.

Problem 7.2: Information

(a) Determine the information $I(\lambda)$ of a sample of size N that was obtained from a normal distribution of known variance σ^2 but unknown mean $\lambda = a$.

(b) Determine the information $I(\lambda)$ of a sample of size N which was drawn from a normal distribution of known mean a but unknown variance $\lambda = \sigma^2$. Show that the maximum-likelihood estimator of σ^2 is given by

$$\mathsf{S} = \frac{1}{N}\sum_{j=1}^{N}(\mathsf{x}^{(j)} - a)^2$$

and that the estimator is unbiased, i.e., $B(\mathsf{S}) = E(\mathsf{S}) - \lambda = 0$.

Problem 7.3: Variance of an Estimator

(a) Use the information inequality to obtain a lower limit on the variance of the estimator of the mean in Problem 7.2 (a). Show that this limit is equal to the minimum variance that was determined in Problem 6.1 (c).

(b) Use the information inequality to obtain a lower limit on the variance of S in Problem 7.2 (b).

(c) Show using Eq. (7.3.12) that S is a minimum variance estimator with the same variance the lower limit found in (b).

Problem 8.1: F-Test

Two samples are given:

$$\begin{array}{lll} (1) & 21, 19, 14, 27, 25, 23, 22, 18, 21, & (N_1 = 9)\,, \\ (2) & 16, 24, 22, 21, 25, 21, 18, & (N_2 = 7)\,. \end{array}$$

Does sample (2) have a smaller variance than sample (1) at a significance level of $\alpha = 5\,\%$?

Problem 8.2: Student's Test

Test the hypothesis that the 30 measurements of Problem 6.5 were drawn from a population with mean 25.5. Use a level of significance of $\alpha = 10\,\%$. Assume that the population is normally distributed.

Problem 8.3: χ^2-Test for Variance

Use the likelihood-ratio method to construct a test of the hypothesis $H_0(\sigma^2 = \sigma_0^2)$ that a sample stems from a normal distribution with unknown mean a and unknown variance σ_0^2. The parameters are $\boldsymbol{\lambda} = (a, \sigma)$. In ω one has $\tilde{\boldsymbol{\lambda}}^{(\omega)} = (\bar{\mathsf{x}}, \sigma_0)$, in Ω: $\bar{\boldsymbol{\lambda}}^{(\Omega)} = (\bar{\mathsf{x}}, \mathsf{s})$.

(a) Form the likelihood ratio T.

(b) Show that instead of T, the test statistic $T' = N\mathsf{s}'^2/\sigma_0^2$ can be used as well.

(c) Show that T' follows a χ^2-distribution with $N-1$ degrees of freedom so that the test can be performed with the help of Table I.7.

Problem 8.4: χ^2-Test of Goodness-of-Fit

(a) Determine the mean $\tilde{a}$ and the variance $\tilde{\sigma}^2$ for the histogram of Fig. 6.1b, i.e.,

x_k	193	195	197	199	201	203	205	207	209	211
n_k	1	2	9	12	23	25	11	9	6	2

Use the result of Example 7.8 to construct the estimates. Give the estimators explicitly as functions of n_k and x_k.

(b) Perform (at a significance level of $\alpha = 10\,\%$) a χ^2-test on the goodness-of-fit of a normal distribution with mean $\tilde{a}$ and variance $\tilde{\sigma}^2$ to the histogram. Use only those bins of the histogram for which $np_k \geq 4$. Determine p_k from the difference of two entries in Table I.2. Give a formula for p_k as function of x_k, Δx, $\tilde{a}$, $\tilde{\sigma}^2$, and ψ_0. Construct a table for the computation of χ^2 containing columns for x_k, n_k, np_k, and $(n_k - np_k)^2/np_k$.

Problem 8.5: Contingency Table

(a) In an immunology experiment [taken from SOKAL and ROHLF, Biometry (Freeman, San Francisco, 1969)] the effect of an antiserum on a particular type of bacteria is studied. 57 mice received a certain dose of bacteria and antiserum, whereas 54 mice received bacteria only. After some time the mice of both groups were counted and the following contingency table constructed.

	Dead	Alive		Sum
Bacteria and antiserum	13	44		57
Only bacteria	25	29		54
Sum	38	73	Total	111

Test (at a significance level of $\alpha = 10\,\%$) the hypothesis that the antiserum has no influence on the survival probability.

(b) In the computation of χ^2 in (a) you will have noticed that the numerators in (8.8.1) all have the same value. Show that generally for 2×2 contingency tables the following relation holds,

$$n_{ij} - n\widetilde{p}_i\widetilde{q}_j = (-1)^{i+j}\frac{1}{n}(n_{11}n_{22} - n_{12}n_{21}) \quad .$$

G.2 Hints and Solutions

Problem 2.1

(a) $N = 365^n$.

(b) $N' = 365 \cdot 364 \cdot \cdots \cdot (365 - n + 1)$.

(c)
$$\begin{aligned} P_{\text{diff}} = N'/N &= \frac{364}{365} \cdot \cdots \cdot \frac{365-n+1}{365} \\ &= \left(1 - \frac{1}{365}\right)\left(1 - \frac{2}{365}\right)\cdots\left(1 - \frac{n-1}{365}\right) . \end{aligned}$$

(d) $P = 1 - P_{\text{diff}}$.

Putting in numbers one obtains $P \approx 0.5$ for $n = 23$ and $P \approx 0.99$ for $n = 57$.

Problem 2.2

$$P(A+B) = P(A) + P(B) - P(AB) \quad ,$$

$$\begin{aligned} P(\text{ace or diamond}) &= P(\text{ace}) + P(\text{diamond}) - P(\text{ace} + \text{diamond}) \\ &= \frac{4}{52} + \frac{13}{52} - \frac{1}{52} = \frac{4}{13} \quad . \end{aligned}$$

Problem 2.3

(a) $P(A)=\frac{4}{52}$, $P(B)=\frac{13}{52}$, $P(AB)=\frac{1}{52}$, i.e., $P(AB)=P(A)P(B)$.

(b) $P(A)=\frac{4}{53}$, $P(B)=\frac{13}{53}$, $P(AB)=\frac{1}{53}$, i.e., $P(AB)\neq P(A)P(B)$.

Problem 2.4

$$P(\overline{AB})=1-P(A+B)=1-P(A)-P(B)+P(AB) \quad .$$

For A and B independent one has $P(AB)=P(A)P(B)$. Therefore,

$$\begin{aligned} P(\overline{AB}) &= 1-P(A)-P(B)+P(A)P(B)=(1-P(A))(1-P(B)) \\ &= P(\bar{A})P(\bar{B}) \quad . \end{aligned}$$

Problem 2.5

(a) $P(A)=0.2$, $P(B)=0.3$, $P(C)=0.5$,
$P(AB)=2\cdot 0.2\cdot 0.3$, $P(AC)=2\cdot 0.2\cdot 0.5$,
$P(BC)=2\cdot 0.3\cdot 0.5$, $P(AA)=0.2^2$, $P(BB)=0.3^2$,
$P(CC)=0.5^2$.

(b) $P(A)=2/10=0.2$, $P(B)=3/10=0.3$, $P(C)=5/10=0.5$,

$$P(AB)=\frac{2}{10}\cdot\frac{3}{9}+\frac{3}{10}\cdot\frac{2}{9},\ P(AC)=\frac{2}{10}\cdot\frac{5}{9}+\frac{5}{10}\cdot\frac{2}{9},$$

$$P(BC)=\frac{3}{10}\cdot\frac{5}{9}+\frac{5}{10}\cdot\frac{3}{9},$$

$$P(AA)=\frac{2}{10}\cdot\frac{1}{9},\ P(BB)=\frac{3}{10}\cdot\frac{2}{9},\ P(CC)=\frac{5}{10}\cdot\frac{4}{9}.$$

For (a) and (b) it holds that

$$\begin{aligned} P(\text{2 identical coins}) &= P(AA)+P(BB)+P(CC), \\ P(\text{2 different coins}) &= P(AB)+P(AC)+P(BC) \\ &= 1-P(\text{2 identical coins}). \end{aligned}$$

Problem 3.1

(a)
$$\begin{aligned}
\widehat{x} &= \sum_{i=1}^{6} x_i p_i = \frac{1}{6}\sum_{i=1}^{6} i = \frac{21}{6} = 3.5 \quad , \\
\sigma^2(x) &= \sum_{i=1}^{6} (x_i - \widehat{x})^2 p_i \\
&= \frac{1}{6}(2.5^2 + 1.5^2 + 0.5^2 + 0.5^2 + 1.5^2 + 2.5^2) \\
&= \frac{2}{6}(6.25 + 2.25 + 0.25) = 2.92 \quad , \\
\mu_3 &= \sum_{i=1}^{6} (x_i - \widehat{x})^3 p_i = \frac{1}{6}\sum_{i=1}^{6} (i - 3.5)^3 = 0 \quad , \\
\gamma &= \mu_3/\sigma^3 = 0 \quad .
\end{aligned}$$

(b)
$$\begin{aligned}
\widehat{x} &= \frac{1}{12}(2+2+3+8+15+18) = 4 \quad , \\
\sigma^2(x) &= \frac{1}{12}(2\cdot 3^2 + 1\cdot 2^2 + 1\cdot 1^2 + 3\cdot 1^2 + 3\cdot 2^2) = \frac{38}{12} = 3.167 \quad , \\
\mu_3 &= \frac{1}{12}(-2\cdot 3^3 - 1\cdot 2^3 - 1\cdot 1^3 + 3\cdot 1^3 + 3\cdot 2^3) = -3 \\
\gamma &= \mu_3/\sigma^3 = -3/3.167^{3/2} = -0.533 \quad .
\end{aligned}$$

Problem 3.2

For $a = -b$: $\widehat{x} = 0$, $x_{0.5} = 0$, $\sigma^2(x) = \dfrac{b^2}{6}$.

For $a = -2b$: $\widehat{x} = -\dfrac{b}{3} = -0.33b$, $x_{0.5} = b(\sqrt{3} - 2) = -0.27b$, $\sigma^2(x) = \dfrac{7}{18}b^2$.

Problem 3.3

$$g(y) = \exp\left(-\frac{y^2}{2}\right) \quad , \quad y(x) = \frac{x}{\sigma} \quad ; \quad f(x) = \frac{\mathrm{d}y(x)}{\mathrm{d}x} g(y(x)) = \frac{1}{\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right) \quad .$$

Problem 3.4

(a) $f_x(x) = \dfrac{1}{\sqrt{2\pi}\sigma_x}\exp\left(-\dfrac{1}{2}\dfrac{x^2}{\sigma_x^2}\right)$, $f_y(y) = \dfrac{1}{\sqrt{2\pi}\sigma_y}\exp\left(-\dfrac{1}{2}\dfrac{y^2}{\sigma_y^2}\right)$.

(b) Yes, since (3.4.6) is fulfilled.

(c) The transformation can be written in the form

$$\begin{pmatrix} u \\ v \end{pmatrix} = R \begin{pmatrix} x \\ y \end{pmatrix} \quad , \qquad R = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \quad .$$

It is orthogonal, since $R^{\mathrm{T}}R = I$. Therefore,

$$\begin{aligned} g(u,v) &= f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma_y^2}\right) \\ &= \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{(u\cos\phi - v\sin\phi)^2}{2\sigma_x^2}-\frac{(u\sin\phi + v\cos\phi)^2}{2\sigma_y^2}\right) \\ &= \frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{u^2\cos^2\phi + v^2\sin^2\phi - 2uv\cos\phi\sin\phi}{2\sigma_x^2}\right. \\ &\quad \left.-\frac{u^2\sin^2\phi + v^2\cos^2\phi + 2uv\cos\phi\sin\phi}{2\sigma_y^2}\right) \quad . \end{aligned}$$

(d) For $\phi = 90°$: $\cos\phi = 0$, $\sin\phi = 1$ etc. the expression $g(u,v)$ factorizes, i.e., $g(u,v) = g_u(u)g_v(v)$.

For $\sigma_x = \sigma_y = \sigma$:

$$g(u,v) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{u^2+v^2}{2\sigma^2}\right) = g(u)g(v) \quad .$$

(e)

$$\begin{aligned} J &= \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial \phi} & \frac{\partial y}{\partial \varphi} \end{vmatrix} = \begin{vmatrix} \cos\phi & \sin\phi \\ -r\sin\phi & r\cos\phi \end{vmatrix} = r \quad , \\ g(r,\phi) &= rf(x,y) = \frac{r}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \\ &= \frac{r}{2\pi\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) \quad ; \\ g_r(r) &= \int_0^{2x} g(r,\phi)\,\mathrm{d}\phi = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) \quad , \\ g_\phi(\phi) &= \frac{1}{2\pi\sigma^2}\int_0^\infty r\exp\left(-\frac{r^2}{2\sigma^2}\right)\mathrm{d}r \\ &= \frac{1}{4\pi\sigma^2}\int_0^\infty \exp\left(-\frac{u}{2\sigma^2}\right)\mathrm{d}u \\ &= -\frac{1}{2\pi}\left[\exp\left(-\frac{u}{2\sigma^2}\right)\right]_0^\infty = \frac{1}{2\pi} \quad ; \\ g(r,\phi) &= g_r(r)g_\phi(\phi) \quad , \end{aligned}$$

therefore, r and ϕ are independent.

Problem 3.5

$$
\begin{aligned}
g &= 4\pi^2\frac{\ell}{T^2} = 4\pi^2\frac{99.8}{2.03^2}\,\mathrm{cm\,s^{-2}} = 956.09\,\mathrm{cm\,s^{-2}} \quad , \\
\frac{\partial g}{\partial \ell} &= \frac{4\pi^2}{T^2} = 9.58\,\mathrm{s^{-2}} \quad , \qquad \frac{\partial g}{\partial T} = -\frac{8\pi^2\ell}{T^3} = -942\,\mathrm{cm\,s^{-3}} \quad , \\
(\Delta g)^2 &= \left(\frac{\partial g}{\partial \ell}\right)^2(\Delta\ell)^2 + \left(\frac{\partial g}{\partial T}\right)^2\Delta T^2 = 2226\,\mathrm{cm^2\,s^{-4}} \quad , \\
\Delta g &= 47.19\,\mathrm{cm\,s^{-2}} \quad .
\end{aligned}
$$

Problem 3.6

(a) $y = Tx$,

$$
\begin{aligned}
T &= \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial p}{\partial m} & \frac{\partial p}{\partial v} \\ \frac{\partial E}{\partial m} & \frac{\partial E}{\partial v} \end{pmatrix} = \begin{pmatrix} v & m \\ \frac{1}{2}v^2 & mv \end{pmatrix} \quad , \\
C_y &= TC_xT^{\mathrm{T}} = \\
&= \begin{pmatrix} v & m \\ \frac{1}{2}v^2 & mv \end{pmatrix}\begin{pmatrix} a^2m^2 & 0 \\ 0 & b^2v^2 \end{pmatrix}\begin{pmatrix} v & \frac{1}{2}v^2 \\ m & mv \end{pmatrix} \\
&= \begin{pmatrix} (a^2+b^2)m^2v^2 & (\frac{1}{2}a^2+b^2)m^2v^3 \\ (\frac{1}{2}a^2+b^2)m^2v^3 & (\frac{1}{4}a^2+b^2)m^2v^4 \end{pmatrix} \quad , \\
\rho(p,E) &= \frac{\mathrm{cov}(p,E)}{\sigma(p)\sigma(E)} = \frac{(\frac{1}{2}a^2+b^2)m^2v^3}{\sqrt{(a^2+b^2)m^2v^2}\sqrt{(\frac{1}{4}a^2+b^2)m^2v^4}} \\
&= \frac{\frac{1}{2}a^2+b^2}{\sqrt{(a^2+b^2)(\frac{1}{4}a^2+b^2)}} \quad .
\end{aligned}
$$

For $a = 0$ or $b = 0$: $\rho = 1$. (In this case either m or v are completely determined. Therefore, there is a strict relation between E and p.) If, however, a, $b \neq 0$, one has $\rho \neq 1$, e.g., for $a = b$ one obtains $\rho = 3/\sqrt{10}$.

(b) $m = \frac{1}{2}p^2/E$, $v = E/2p$,

$$
\begin{aligned}
C_y &= \begin{pmatrix} (a^2+b^2)p^2 & (a^2+2b^2)Ep \\ (a^2+2b^2)Ep & (a^2+4b^2)E^2 \end{pmatrix} \quad , \\
m &= Ty \quad , \\
T &= \left(\frac{\partial m}{\partial y_1}, \frac{\partial m}{\partial y_2}\right) = \left(\frac{\partial m}{\partial p}, \frac{\partial m}{\partial E}\right) = \left(\frac{p}{E}, -\frac{p^2}{2E^2}\right) \quad , \\
C_m &= \sigma^2(m) = TC_yT^{\mathrm{T}} \\
&= \left(\frac{p}{E}, -\frac{p^2}{2E^2}\right)\begin{pmatrix} (a^2+b^2)p^2 & (a^2+2b^2)Ep \\ (a^2+2b^2)Ep & (a^2+4b^2)E^2 \end{pmatrix}\begin{pmatrix} \frac{p}{E} \\ -\frac{p^2}{2E^2} \end{pmatrix} \\
&= a^2\frac{p^4}{4E^2} = a^2m^2 \quad .
\end{aligned}
$$

Problem 5.1

$$\text{(a)}\quad W_{k+1}^n = \binom{n}{k+1} p^{k+1} q^{n-k-1} = \frac{n!}{(k+1)!(n-k-1)!} p^k q^{n-k} \frac{p}{q}$$
$$= \frac{n!}{k!(n-k)!} \frac{n-k}{k+1} p^k q^{n-k} \frac{p}{q} = W_k^n \frac{n-k}{k+1} \frac{p}{q} .$$

$$\text{(b)}\quad W_3^5 = \binom{5}{3} 0.8^3 \cdot 0.2^2 = 10 \cdot 0.512 \cdot 0.04 = 0.2048 ,$$
$$W_4^5 = W_3^3 \cdot \frac{2}{4} \cdot \frac{0.8}{0.2} = 0.2048 \cdot 2 = 0.4096 ,$$
$$W_5^5 = W_4^5 \cdot \frac{1}{5} \cdot \frac{0.8}{0.2} = 0.4096 \cdot 0.8 = 0.32768 ,$$
$$P_3 = 1 - W_4^5 - W_5^5 = 0.26272 ,$$
$$P_2 = P_3 - W_3^5 = 0.05792 .$$

(c) Using the result of (a) we obtain

$$W_k^n - W_{k-1}^n = W_{k-1}^n \left(\frac{n-k+1}{k} \frac{p}{q} - 1 \right) \quad .$$

Thus the probability W_k^n increases as long as the expression in brackets is positive, i.e.,

$$\frac{(n-k+1)p}{kq} - 1 > 0 \quad .$$

Since k and q are positive we have

$$(n-k+1)p > kq = k(1-p) \quad , \qquad k < (n+1)p \quad .$$

The most probable value k_m is the largest value of k for which this inequality holds.

(d) $\varphi_{\mathsf{x}_i}(t) = E\{\mathrm{e}^{\mathrm{i}t\mathsf{x}_i}\} = q\mathrm{e}^{\mathrm{i}t\cdot 0} + p\mathrm{e}^{\mathrm{i}t} = q + p\mathrm{e}^{\mathrm{i}t}$;

$$\varphi_{\mathsf{x}} = (q + p\mathrm{e}^{\mathrm{i}t})^n = \sum_{k=0}^{m} \binom{n}{k} q^{n-k} p^k \mathrm{e}^{\mathrm{i}tk} = \sum_{k=0}^{m} f(k)\mathrm{e}^{\mathrm{i}tk} = E\{\mathrm{e}^{\mathrm{i}tk}\} \quad ,$$
$$f(k) = W_k^n \quad .$$

Problem 5.2

For $\lambda = 3$ one has $f(0) = \dfrac{\lambda^0}{0!}\mathrm{e}^{-\lambda} = \mathrm{e}^{-\lambda} = 0.0498 \approx 0.05 = 5\,\%$.

Problem 5.3

(a) The fraction

$$f_r = 2\psi_0\left(\frac{(R_0 - a_2 R_0) - R_0}{b R_0}\right) = 2\psi_0\left(-\frac{a_2}{b}\right)$$

is rejected, since $R < R_0 - \Delta_2$ or $R > R_0 + \Delta_2$. Correspondingly the fractions

$$f_2 = 2\psi_0\left(-\frac{a_1}{b}\right) - f_r \qquad \text{and} \qquad f_5 = 1 - f_2 - f_r$$

give prices $2C$ and $5C$, respectively. Therefore,

$$\begin{aligned} P &= 2f_2C + 5f_5C - C = C\{2f_2 + 5 - 5f_2 - 5f_r - 1\} \\ &= C\{4 - 3f_2 - 5f_r\} \\ &= C\left\{4 - 6\psi_0\left(-\frac{a_1}{b}\right) + 3f_r - 5f_r\right\} \\ &= C\left\{4 - 6\psi_0\left(-\frac{a_1}{b}\right) - 4\psi_0\left(-\frac{a_2}{b}\right)\right\} \quad . \end{aligned}$$

(b) $\psi_0(-0.2) = 0.421 \quad ; \qquad \psi_0(-1) = 0.159 \quad ,$

i.e., $P = C\{4 - 2.526 - 0.636\} = 0.838C \quad .$

(c) $\dfrac{\mathrm{d}^2 f}{\mathrm{d}x^2} = \dfrac{1}{\sqrt{2\pi}b^3}\exp(-(x-a)^2/2b^2)\{(x-a)^2/b^2 - 1\} = 0$

is fulfilled if the expression in the last set of brackets vanishes.

Problem 5.4

(a) $$\begin{aligned} P(R) &= \int_0^R g(r)\,\mathrm{d}r = \frac{1}{\sigma^2}\int_0^R r\mathrm{e}^{-r^2/2\sigma^2}\,\mathrm{d}r \\ &= \frac{1}{2\sigma^2}\int_0^{R^2} \mathrm{e}^{-u/2\sigma^2}\,\mathrm{d}u \\ &= \left[\mathrm{e}^{-u/2\sigma^2}\right]_{R^2}^0 = 1 - \mathrm{e}^{-R^2/2\sigma^2} \quad . \end{aligned}$$

(b) $1 - P = \exp(-R^2/2\sigma^2) \quad , \qquad R^2/2\sigma^2 = -\ln(1-P) \quad .$

For $\sigma = 1$, $P = 0.9$ one obtains

$$R = \sqrt{-2\ln 0.1} = \sqrt{4.61} = 2.15 \quad .$$

Problem 5.5

(a) $0 \le \mathsf{u} < 1$:

$$f(u) = \int_{u-1}^{u} f_1(y)\,\mathrm{d}y = \int_0^u y\,\mathrm{d}y = \frac{1}{2}u^2 \quad .$$

$1 \le \mathsf{u} < 2$:

$$\begin{aligned} f(u) &= \int_{u-1}^{u} f_1(y)\,\mathrm{d}y + \int_{u-1}^{u} f_2(y)\,\mathrm{d}y \\ &= \int_{u-1}^{1} y\,\mathrm{d}y + \int_{1}^{u} (2-y)\,\mathrm{d}y \\ &= \frac{1}{2}(1-(u-1)^2) + \frac{1}{2}(1-(2-u)^2) \\ &= \frac{1}{2}(-3+6u-2u^2) \quad . \end{aligned}$$

$2 \le \mathsf{u} < 3$:

$$f(u) = \int_{u-1}^{u} f_2(y)\,\mathrm{d}y = \int_{u-1}^{2} (2-y)\,\mathrm{d}y = -\int_{3-u}^{0} z\,\mathrm{d}z = \frac{1}{2}(3-u)^2 \quad .$$

(b) $f(u) = \dfrac{1}{2\pi\sigma_x\sigma_y} \displaystyle\int_{-\infty}^{\infty} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{(u-x)^2}{2\sigma_y^2}\right) \mathrm{d}x \quad .$

Completing the square yields for the exponent

$$-\frac{\sigma^2}{2\sigma_x^2\sigma_y^2}\left\{\left(x - \frac{\sigma_x^2}{\sigma^2}u\right)^2 - \frac{\sigma_x^4}{\sigma^4}u^2 + \frac{\sigma_x^2}{\sigma^2}u^2\right\} \quad .$$

With the change of variables $v = (\sigma/\sigma_x\sigma_y)(x - \sigma_x^2 u/\sigma^2)$ we obtain

$$\begin{aligned} f(u) &= \frac{1}{2\pi\sigma_x\sigma_y} \exp\left(\frac{\sigma_x^4 - \sigma^2\sigma_x^2}{2\sigma^2\sigma_x^2\sigma_y^2}u^2\right) \frac{\sigma_x\sigma_y}{\sigma} \\ &\quad \times \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}v^2\right) \mathrm{d}v \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \quad , \end{aligned}$$

since $\sigma_x^4 = \sigma_x^2(\sigma^2 - \sigma_y^2)$.

Problem 6.1

(a) $E(\mathsf{S}) = E\{\sum a_i \mathsf{x}_i\} = \sum a_i E(\mathsf{x}_i) = \widehat{x}\sum a_i = \widehat{x}$
if and only if
$\sum a_i = 1$.

(b) $$\begin{aligned} \sigma^2 &= \sum\left(\frac{\partial \mathsf{S}}{\partial x_i}\right)^2 \sigma^2 = \sigma^2 \sum a_i^2 \quad , \\ \sigma^2(\mathsf{S}_1) &= \sigma^2\left(\frac{1}{16} + \frac{1}{16} + \frac{1}{4}\right) = \sigma^2 \cdot \frac{3}{8} = 0.375\sigma^2 \quad , \\ \sigma^2(\mathsf{S}_2) &= \sigma^2\left(\frac{1}{25} + \frac{4}{25} + \frac{4}{25}\right) = \sigma^2 \cdot \frac{9}{25} = 0.360\sigma^2 \quad , \\ \sigma^2(\mathsf{S}_3) &= \sigma^2\left(\frac{1}{36} + \frac{1}{9} + \frac{1}{4}\right) = \sigma^2 \cdot \frac{7}{18} = 0.389\sigma^2 \quad . \end{aligned}$$

$$
\begin{aligned}
\text{(c)}\quad \sigma^2(\mathsf{S}) &= [a_1^2 + a_2^2 + (1-(a_1+a_2))^2]\sigma^2 \quad , \\
\frac{\partial \sigma^2(\mathsf{S})}{\partial a_1} &= [2a_1 - 2(1-(a_1+a_2))]\sigma^2 = 0 \quad , \\
\frac{\partial \sigma^2(\mathsf{S})}{\partial a_2} &= [2a_2 - 2(1-(a_1+a_2))]\sigma^2 = 0 \quad , \\
a_1 &= 1-(a_1+a_2) \quad , \\
a_2 &= 1-(a_1+a_2) \quad , \qquad a_1 = a_2 = \tfrac{1}{3} \quad ; \\
\sigma^2(\mathsf{S}) &= \tfrac{1}{3}\sigma^2 = 0.333\sigma^2 \quad .
\end{aligned}
$$

Problem 6.2

$$
\begin{aligned}
a &= 20 \quad , \\
\Delta &= \frac{1}{n}\sum \delta_i \\
&= \frac{1}{10}(-2+1+3-1+0+1+0-1+0-3) = -0.2 \quad , \\
\bar{\mathsf{x}} &= 19.8 \quad , \\
\mathsf{s} &= \frac{1}{9}(1.8^2 + 1.2^2 + 3.2^2 + 0.8^2 + 0.2^2 + \\
&\quad + 1.2^2 + 0.2^2 + 0.8^2 + 0.2^2 + 2.8^2) \\
&= \frac{1}{9}\cdot 25.60 = 2.84 \quad , \\
\mathsf{s}_{\bar{\mathsf{x}}}^2 &= 0.284 \quad . \qquad \text{Therefore } \bar{\mathsf{x}} = 19.8 \pm 0.53 \quad .
\end{aligned}
$$

Problem 6.3

$$
\begin{aligned}
\text{(a)}\quad \bar{\mathsf{x}}_1 &= 0.2,\ \bar{\mathsf{x}}_2 = 0.5,\ \bar{\mathsf{x}}_3 = 0.7 \quad , \\
\tilde{x} &= 0.02 + 0.35 + 0.14 = 0.51 \quad , \\
\mathsf{s}_1^2 &= \frac{1}{9}(2 - 10\cdot 0.2^2) = 0.178 \quad , \\
\mathsf{s}_2^2 &= \frac{1}{9}(5 - 10\cdot 0.5^2) = 0.278 \quad , \\
\mathsf{s}_3^2 &= \frac{1}{9}(7 - 10\cdot 0.7^2) = 0.233 \quad , \\
\mathsf{s}_{\tilde{\mathsf{x}}}^2 &= \frac{0.1^2}{10}0.178 + \frac{0.7^2}{10}0.278 + \frac{0.2^2}{10}0.233 \\
&= 0.001\cdot 0.178 + 0.049\cdot 0.278 + 0.004\cdot 0.233 \\
&= 0.0147 \quad ,
\end{aligned}
$$

$$\mathsf{s}_{\tilde{\mathsf{x}}} = \sqrt{\mathsf{s}_{\tilde{\mathsf{x}}}^2} = 0.12 \quad .$$

The result $\widetilde{x} = 0.51 \pm 0.12$ does not favor any one party significantly.

(b) $$\begin{aligned} \mathsf{s}_1 &= 0.422,\ \mathsf{s}_2 = 0.527,\ \mathsf{s}_3 = 0.483 \quad , \\ p_1\mathsf{s}_1 &= 0.0422,\ p_2\mathsf{s}_2 = 0.369,\ p_3\mathsf{s}_3 = 0.0966 \quad , \\ \sum p_i\mathsf{s}_i &= 0.508 \quad , \\ \frac{n_1}{n} &= 0.083,\ \frac{n_2}{n} = 0.726,\ \frac{n_3}{n} = 0.190 \quad . \end{aligned}$$

Problem 6.4

(a) For simplicity we write $\chi^2 = u$.

$$\begin{aligned} \mu_3 &= E\{(\mathsf{u} - \widehat{u})^3\} = E\{\mathsf{u}^3 - 3\mathsf{u}^2\widehat{u} + 3\mathsf{u}\widehat{u}^2 - \widehat{u}^3\} \\ &= E(\mathsf{u}^3) - 3\widehat{u}E(\mathsf{u}^2) + 3\widehat{u}^2E(\mathsf{u}) - \widehat{u}^3 \\ &= \lambda_3 - 3\widehat{u}E(\mathsf{u}^2) + 2\widehat{u}^3 \quad , \\ E(\mathsf{u}^3) &= \lambda_3 = \frac{1}{\mathrm{i}^3}\varphi'''(0) \\ &= \mathrm{i}\varphi'''(0) = \mathrm{i}(-\lambda)(-\lambda-1)(-\lambda-2)(-2\mathrm{i})^3 \\ &= 8\lambda(\lambda+1)(\lambda+2) = 8\lambda^3 + 24\lambda^2 + 16\lambda \quad , \\ 3\widehat{u}E(\mathsf{u}^2) &= 6\lambda(4\lambda^2 + 4\lambda) = 24\lambda^3 + 24\lambda^2 \quad , \\ 2\widehat{u}^3 &= 2\cdot(2\lambda)^3 = 16\lambda^3 \quad , \\ \mu_3 &= 16\lambda \quad , \\ \gamma &= \mu_3/\sigma^3 = 16\lambda/8\lambda^{\frac{3}{2}} = 2\lambda^{-\frac{1}{2}} \quad . \end{aligned}$$

(b) Obviously $\gamma = 2/\sqrt{\lambda} \to 0$ holds for $\lambda = \frac{1}{2}n \to \infty$.

Problem 6.5

See Fig. G.2.

Problem 7.1

(a) $L = \prod_{j=1}^{N} \frac{1}{b} = \frac{1}{b^N}$; obviously one has $L = \max$ for $b = \mathsf{x}_{\max}$.

(b) $\lambda_1 = a$, $\lambda_2 = \Gamma$,

$$\begin{aligned} L &= \prod_{j=1}^{N} \frac{2}{\pi\lambda_2} \frac{\lambda_2^2}{4(\mathsf{x}^{(j)} - \lambda_1)^2 + \lambda_2^2} \\ &= \left(\frac{2\lambda_2}{\pi}\right)^N \prod_{j=1}^{N} \frac{1}{4(\mathsf{x}^{(j)} - \lambda_1)^2 + \lambda_2^2} \quad , \end{aligned}$$

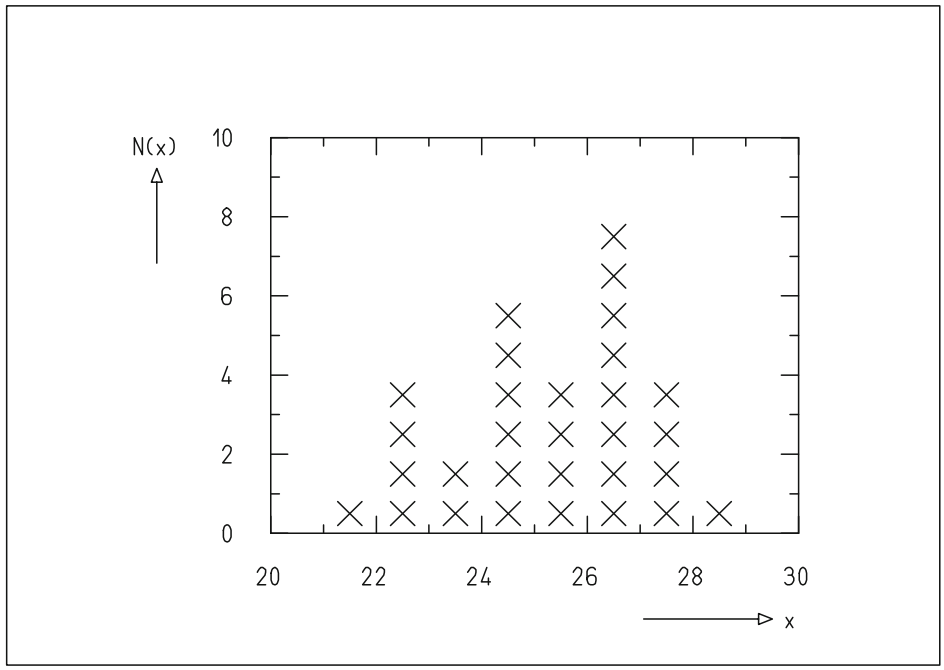

Fig. G.2: Histogram of the data of Problem 6.5.

$$\begin{aligned}
\ell &= N(\ln 2 - \ln\pi + \ln\lambda_2) - \sum_{j=1}^{N} \ln[4(\mathbf{x}^{(j)} - \lambda_1)^2 + \lambda_2^2] \quad , \\
\frac{\partial \ell}{\partial \lambda_1} &= \sum_{j=1}^{N} \frac{8(\mathbf{x}^{(i)} - \lambda_1)}{4(\mathbf{x}^{(j)} - \lambda_1)^2 + \lambda_2^2} = 0 \quad , \\
\frac{\partial \ell}{\partial \lambda_2} &= \frac{N}{\lambda_2} - 2\lambda_2 \sum_{j=1}^{N} \frac{1}{4(\mathbf{x}^{(j)} - \lambda_1)^2 + \lambda_2^2} = 0 \quad .
\end{aligned}$$

There is not necessarily a unique solution since the equations are not linear in λ_1 and λ_2. For $|\mathbf{x}^{(j)} - \lambda_1| \ll \lambda_2$, however, we may write

$$\frac{8}{\lambda_2^2} \sum_{j=1}^{N} (\mathbf{x}^{(j)} - \lambda_1) = 0 \quad , \qquad N\lambda_1 = \sum_{j=1}^{N} \mathbf{x}^{(j)} \quad , \qquad \lambda_1 = a = \bar{\mathbf{x}} \quad .$$

Problem 7.2

(a)

$$\begin{aligned}
L &= \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp[-(\mathbf{x}^{(j)} - \lambda)^2 / 2\sigma^2] \\
&= (\sqrt{2\pi}\sigma)^{-N} \prod_{j=1}^{N} \exp[-(\mathbf{x}^{(j)} - \lambda)^2 / 2\sigma^2] \quad ,
\end{aligned}$$

$$
\begin{aligned}
\ell &= -N\ln(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{j=1}^{N}(\mathsf{x}^{(j)}-\lambda)^2 \quad , \\
\ell' &= \frac{1}{\sigma^2}\sum_{j=1}^{N}(\mathsf{x}^{(j)}-\lambda) \quad , \\
\ell'' &= -N/\sigma^2 \quad , \\
I(\lambda) &= -E(\ell'') = N/\sigma^2 \quad .
\end{aligned}
$$

(b)

$$
\begin{aligned}
L &= \prod_{j=1}^{N}\frac{1}{\sqrt{2\pi}\sqrt{\lambda}}\exp\left(-\frac{(\mathsf{x}^{(j)}-a)^2}{2\lambda}\right) \\
&= (2\pi)^{-N/2}\lambda^{-N/2}\prod_{j=1}^{N}\exp\left(-\frac{(\mathsf{x}^{(j)}-a)^2}{2\lambda}\right) \quad , \\
\ell &= -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\lambda - \frac{1}{2\lambda}\sum_{j=1}^{N}(\mathsf{x}^{(j)}-a)^2 \quad , \\
\ell' &= -\frac{N}{2\lambda} + \frac{1}{2\lambda^2}\sum_{j=1}^{N}(\mathsf{x}^{(j)}-a)^2 \quad , \\
\ell'' &= \frac{N}{2\lambda^2} - \frac{1}{\lambda^3}\sum_{j=1}^{N}(\mathsf{x}^{(j)}-a)^2 \quad , \\
I(\lambda) &= -E(\ell'') = -\frac{N}{2\lambda^2} + \frac{1}{\lambda^3}E\left\{\sum_{j=1}^{N}(\mathsf{x}^{(j)}-a)^2\right\} \\
&= -\frac{N}{2\lambda^2} + \frac{1}{\lambda^3}N\lambda \quad , \\
I(\lambda) &= \frac{1}{2\lambda^2}(-N+2N) = \frac{N}{2\lambda^2} \quad , \\
E(\mathsf{S}) &= \frac{1}{N}E\left\{\sum_{j=1}^{N}(\mathsf{x}^{(j)}-a)^2\right\} = \sigma^2 = \lambda \quad .
\end{aligned}
$$

Problem 7.3

(a) $\sigma^2(\mathsf{S}) \geq \dfrac{1}{I(\lambda)} = \dfrac{\sigma^2}{N}$,

In Problem 6.1 (c) we had $\sigma^2(\bar{\mathsf{x}}) = 1/3\sigma^2$ for $N = 3$.

(b) $\sigma^2(\mathsf{S}) \geq \dfrac{1}{I(\lambda)} = \dfrac{2\lambda^2}{N}$.

(c) From Problem 7.2 we take

$$
\ell' = -\frac{N}{2\lambda} + \frac{1}{2\lambda^2}N\mathsf{S} = \frac{N}{2\lambda^2}(\mathsf{S}-\lambda) = \frac{N}{2\lambda^2}(\mathsf{S}-E(\mathsf{S})) \quad .
$$

Thus, $\sigma^2(\mathsf{S}) = \dfrac{2\lambda^2}{N}$.

Problem 8.1

$$\begin{aligned} \bar{\mathsf{x}}_1 &= 21.11 \quad , \qquad \bar{\mathsf{x}}_2 = 21.00 \quad , \\ \mathsf{s}_1^2 &= 14.86 \quad , \qquad \mathsf{s}_2^2 = 10.00 \quad , \\ T &= \mathsf{s}_1^2/\mathsf{s}_2^2 = 1.486 \quad , \qquad F_{0.95}(8,6) = 4.15 \quad . \end{aligned}$$

The variance of sample (2) is not significantly smaller.

Problem 8.2

$$\begin{aligned} \bar{\mathsf{x}} &= 25.142 \quad , \qquad \mathsf{s}^2 = 82.69/29 = 2.85 \quad , \qquad \mathsf{s} = 1.69 \quad , \\ T &= \frac{\bar{\mathsf{x}} - 25.5}{\mathsf{s}/\sqrt{30}} = \frac{25.142 - 25.5}{1.69/5.48} = -\frac{0.358}{0.309} = -1.16 \quad , \\ |T| &< t_{0.95}(f = 29) = 1.70 \quad . \end{aligned}$$

Therefore the hypothesis cannot be rejected.

Problem 8.3

(a)
$$\begin{aligned} f(\mathsf{x}^{(1)}, \mathsf{x}^{(2)}, \ldots, \mathsf{x}^{(N)}, \tilde{\boldsymbol{\lambda}}^{(\Omega)}) &= \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi}\mathsf{s}'} \exp\left(-\frac{(\mathsf{x}^{(j)} - \bar{\mathsf{x}})^2}{2\mathsf{s}'^2}\right) \\ &= (\sqrt{2\pi}\mathsf{s}')^{-N} \exp\left(-\frac{1}{2\mathsf{s}'^2}\sum_{j=1}^{N}(\mathsf{x}^{(j)} - \bar{\mathsf{x}})^2\right) \quad , \\ f(\mathsf{x}^{(1)}, \mathsf{x}^{(2)}, \ldots, \mathsf{x}^{(N)}, \tilde{\boldsymbol{\lambda}}^{(\omega)}) & \\ &= (\sqrt{2\pi}\sigma_0)^{-N} \exp\left(-\frac{1}{2\sigma_0^2}\sum_{j=1}^{N}(\mathsf{x}^{(j)} - \bar{\mathsf{x}})^2\right) \quad , \end{aligned}$$

$$\begin{aligned} T &= \left(\frac{\sigma_0}{\mathsf{s}'}\right)^N \exp\left[\tfrac{1}{2}N\mathsf{s}'^2\left(\frac{1}{\sigma_0^2} - \frac{1}{\mathsf{s}'^2}\right)\right] \quad , \\ \ln T &= N(\ln\sigma_0 - \ln\mathsf{s}') + \tfrac{1}{2}N\mathsf{s}'^2\left(\frac{1}{\sigma_0^2} - \frac{1}{\mathsf{s}'^2}\right) \\ &= N(\ln\sigma_0 - \ln\mathsf{s}') + \tfrac{1}{2}N\frac{\mathsf{s}'^2}{\sigma_0^2} - \tfrac{1}{2}N \quad . \end{aligned}$$

(b) $T' = N\mathsf{s}'^2/\sigma_0^2$ is a monotonically increasing function of s' if $\mathsf{s}' > \sigma_0$, otherwise it is monotonically decreasing. Therefore the test takes on the following form:

$$T' > T'_{1-\frac{1}{2}\alpha} \quad , \qquad T' < T'_{\frac{1}{2}\alpha} \quad \text{for} \quad H_0(\mathsf{s}' = \sigma_0) \quad ,$$
$$T' > T'_{1-\alpha} \quad \text{for} \quad H_0(\mathsf{s}' < \sigma_0) \quad ,$$
$$T' < T'_{\alpha} \quad \text{for} \quad H_0(\mathsf{s}' > \sigma_0) \quad .$$

(c) The answer follows from (6.7.2) and $\mathsf{s}'^2 = (N-1)\mathsf{s}^2/N$.

Problem 8.4

(a) $$\widetilde{a} = \frac{1}{n}\sum_k n_k \mathsf{x}_k = 202.36\,.$$
$$\widetilde{\sigma}^2 = \frac{1}{n-1}\sum_k n_k(\mathsf{x}_k - \widetilde{a})^2 = 13.40\,, \quad \widetilde{\sigma} = 3.66\,.$$

(b) $$p_k(\mathsf{x}_k) = \psi_0\left(\frac{\mathsf{x}_k + \frac{1}{2}\Delta x - \widetilde{a}}{\widetilde{\sigma}}\right) - \psi_0\left(\frac{\mathsf{x}_k - \frac{1}{2}\Delta x - \widetilde{a}}{\widetilde{\sigma}}\right) = \psi_{0+} - \psi_{0-}\,.$$

x_k	n_k	ψ_{0+}	ψ_{0-}	np_k	$\frac{(n_k - np_k)^2}{np_k}$
193	1	0.011	0.002	(0.9)	–
195	2	0.041	0.011	(3.0)	–
197	9	0.117	0.041	7.6	0.271
199	12	0.260	0.117	14.3	0.362
201	23	0.461	0.260	20.1	0.411
203	25	0.673	0.463	21.2	0.679
205	11	0.840	0.673	16.7	1.948
207	9	0.938	0.840	9.8	0.071
209	6	0.982	0.938	4.3	0.647
211	2	0.996	0.982	(1.4)	–
					$\mathsf{X}^2 = 4.389$

The number of degrees of freedom is $7-2=5$. Since $\chi^2_{0.90}(5) = 9.24$, the test does not reject the hypothesis.

Problem 8.5

(a) $$\widetilde{p}_1 = \frac{1}{111}(13+44) = \frac{57}{111} \quad , \qquad \widetilde{p}_2 = \frac{1}{111}(25+29) = \frac{54}{111} \quad ,$$
$$\widetilde{q}_1 = \frac{1}{111}(13+25) = \frac{38}{111} \quad , \qquad \widetilde{q}_2 = \frac{1}{111}(44+29) = \frac{73}{111} \quad ,$$

$$\mathsf{X}^2 = \frac{\left(13 - \frac{57\cdot 38}{111}\right)^2}{\frac{57\cdot 38}{111}} + \frac{\left(44 - \frac{57\cdot 73}{111}\right)^2}{\frac{57\cdot 73}{111}}$$

$$+\frac{\left(25-\frac{54\cdot 38}{111}\right)^2}{\frac{54\cdot 38}{111}}+\frac{\left(29-\frac{54\cdot 73}{111}\right)^2}{\frac{54\cdot 73}{111}}$$

$$= \frac{42.43}{19.51}+\frac{42.43}{37.49}+\frac{42.43}{18.49}+\frac{42.43}{35.51}=6.78 \quad .$$

Since $\chi^2_{0.90}=2.71$ for $f=1$, the hypothesis of independence is rejected.

(b) $$n_{ij}-\frac{1}{n}(n_{i1}+n_{i2})(n_{1j}+n_{2j})$$

$$=\frac{1}{n}[n_{ij}(n_{11}+n_{12}+n_{21}+n_{22})-(n_{i1}+n_{i2})(n_{1j}+n_{2j})] \quad .$$

One can easily show that the expression in square brackets takes the form $(-1)^{i+j}(n_{11}n_{22}-n_{12}n_{21})$ for all i, j.

G.3 Programming Problems

Programming Problem 4.1: Program to Generate Breit–Wigner-Distributed Random Numbers

Write a method with the following declaration

double [] breitWignerNumbers(double a, double gamma, int n).

It is to yield n random numbers, which follow a Breit–Wigner distribution having a mean of a and a FWHM of Γ. Make the method part of a class which allows for interactive input of n, a, Γ and numerical as well as graphical output of the random numbers in the form a histogram, Fig. G.3. (Example solution: S1Random)

Programming Problem 4.2: Program to Generate Random Numbers from a Triangular Distribution

Write a method with the declaration

double [] triangularNumbersTrans (double a, double b, double c, int n).

It is to yield n random numbers, following a triangular distribution with the parameters a, b, c generated by the transformation procedure of Example 4.3.

Write a second method with the declaration

double [] triangularNumbersRej (double a, double b, double c, int n),

which solves the same problem, but uses von Neumann's acceptance–rejection method. Which of the two programs is faster?

Write a class which allows to interactively choose either method. It should also allow for numerical and graphical output (as histogram) of the generated numbers, Fig. G.4. (Example solution: S2Random)

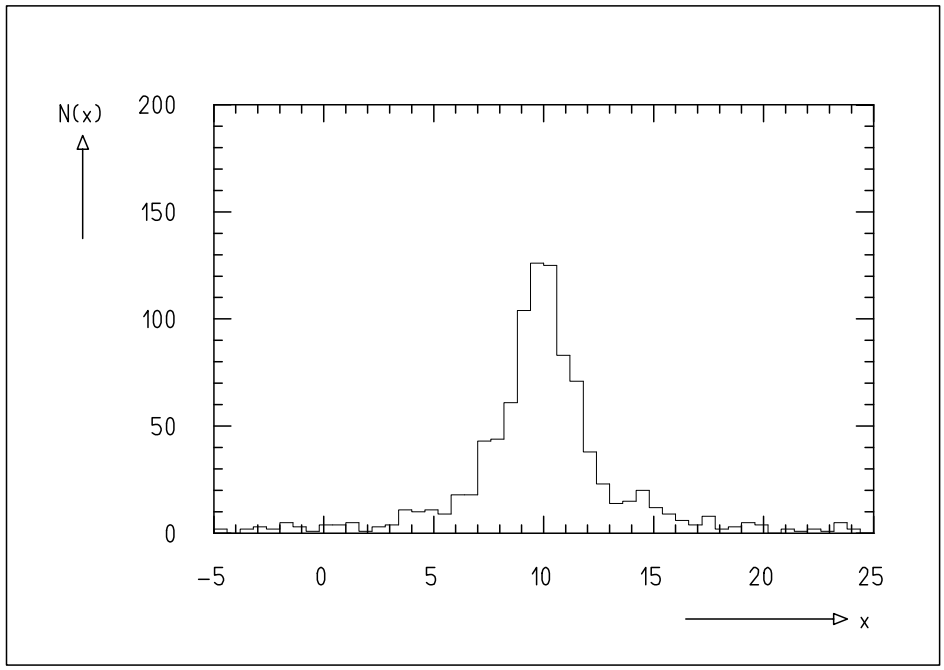

Fig. G.3: Histogram of 1000 random numbers following a Breit–Wigner distribution with $a = 10$ and $\Gamma = 3$.

Programming Problem 4.3: Program to Generate Data Points with Errors of Different Size

Write a method similar to DatanRandom.line which generates data points $y = at + b + \Delta y$. The errors Δy, however, are not to be taken for all data points y_i from the same uniform distribution with the width σ, but Δy_i is to be sampled from a normal distribution with the width σ_i. The widths σ_i are to be taken from a uniform distribution within the region $\sigma_{\min} < \sigma_i < \sigma_{\max}$.

Write a class which calls this method and which displays graphically the straight line $y = at + b$ as well as the simulated data points with error bars $y_i \pm \Delta y_i$, Fig. G.5. (Example solution: S3Random)

Programming Problem 5.1: Convolution of Uniform Distributions

Because of the Central Limit Theorem the quantity $\mathsf{x} = \sum_{i=1}^{N} \mathsf{x}_i$ follows in the limit $N \to \infty$ the standard normal distribution if the x_i come from an arbitrary distribution with mean zero and standard deviation $1/\sqrt{N}$. Choose for the x_i the uniform distribution with the limits

$$a = -\sqrt{3/N} \quad , \qquad b = -a \quad .$$

Perform a large number n_{exp} of Monte Carlo experiments, each giving a random value x. Produce a histogram of the quantity x and show in addition the distribution of x as a continuous curve which you would expect from the standard normal distribution

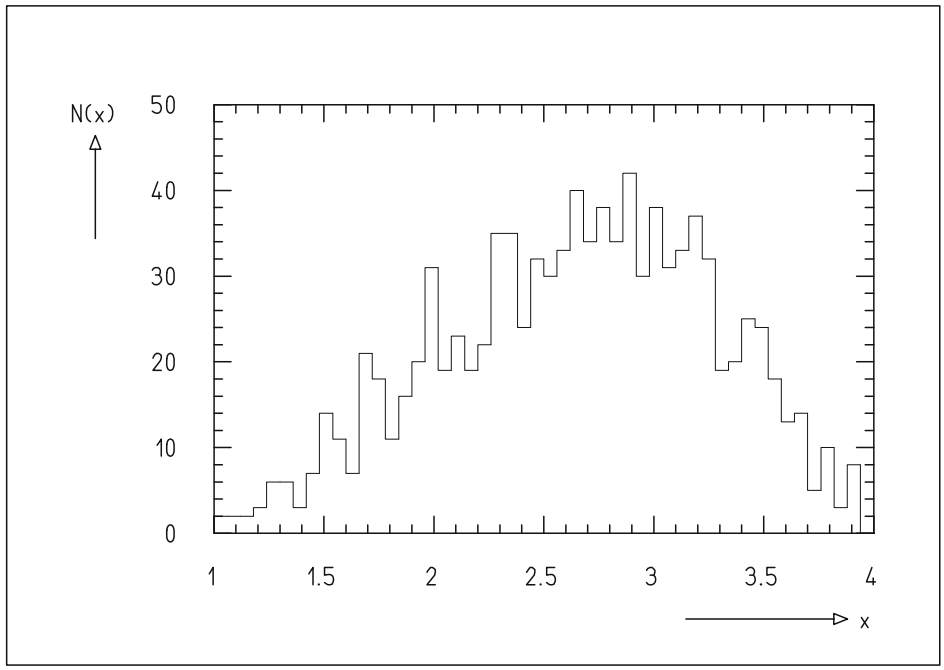

Fig. G.4: Histogram of 1000 random numbers following a triangular distribution with $a = 1$, $b = 4$, $c = 3$.

(Fig. G.6). (Use the class GraphicsWithHistogramAndPolyline for the simultaneous representation of histograms and curves.) Allow for interactive input of the quantities n_{exp} and N. (Example solution: S1Distrib)

Programming Problem 5.2: Convolution of Uniform Distribution and Normal Distribution

If x is taken from a uniform distribution between a and b and if y is taken from a normal distribution with mean zero and width σ, then the quantity $u = x + y$ follows the distribution (5.11.14). Perform a large number n_{exp} of Monte Carlo experiments, each resulting in a random number u. Display a histogram of the quantity u and show in addition a curve of the distribution you would expect from (5.11.14), Fig. G.7. Allow for the interactive input of the quantities n_{exp}, a, b, and σ.

Programming Problem 7.1: Distribution of Lifetimes Determined from a Small Number of Radioactive Decays

In Example Program 7.1, an estimate $\bar{t}$ of the mean lifetime τ and its asymmetric errors Δ_- and Δ_+ are found from a single small sample. In all cases the program yields $\Delta_- < \Delta_+$. Write a program that simulates a large number n_{exp} of experiments, in each of which N radioactive decays of mean lifetime $\tau = 1$ are measured. Compute for each experiment the estimate $\bar{t}$ and construct a histogram of the quantity $\bar{t}$ for all experiments. Present this histogram $N_i(\bar{t}_i \le \bar{t} < \bar{t}_i + \Delta\bar{t})$ and also the cumulative frequency distribution $h_i = (1/n_{\text{exp}}) \sum_{\bar{t}<\bar{t}_i} N_i$. Allow for interactive input of n_{exp}

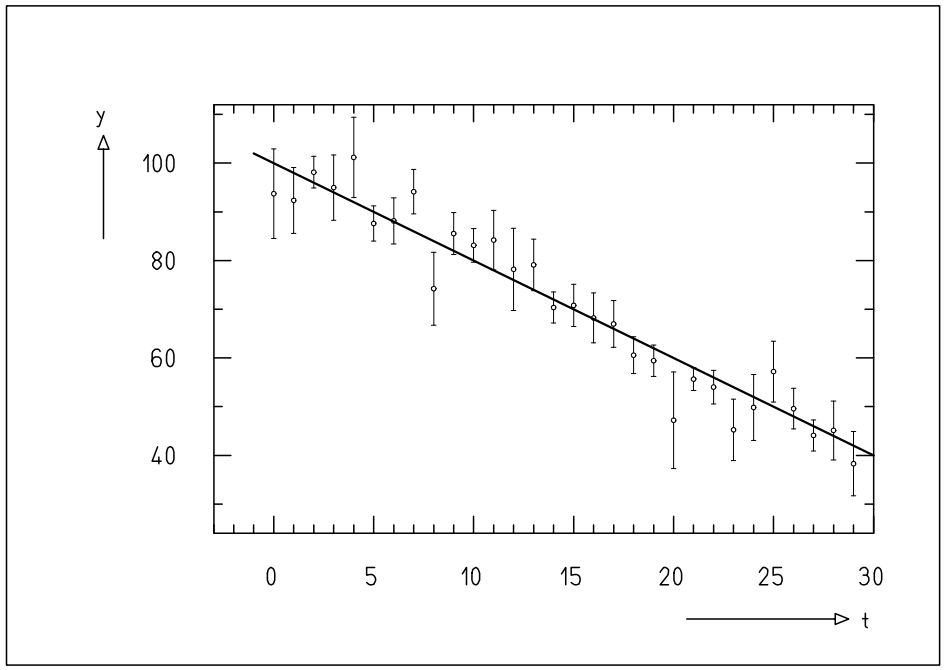

Fig. G.5: Data points with errors of different size.

and N. Demonstrate that the distributions are asymmetric for small N and that they become symmetric for large N. Show that for small N the value $\bar{t} = 1$ is not the most probable value, but that it is the expectation value of $\bar{t}$. Determine for a fixed value of N, e.g., $N = 4$, limits Δ_- and Δ_+ in such a way that $\bar{t} < 1 - \Delta_-$ holds with the probability $0.683/2$ and that with the same probability one has $\bar{t} > 1 + \Delta_+$. Compare the results found with a series of simulated experiments from the program E1MaxLike, Example Program 7.1. (Example solution: S1MaxLike)

Programming Problem 7.2: Distribution of the Sample Correlation Coefficient

Modify the class E2MaxLike so that instead of numerical output, a histogram of the correlation coefficient r is presented (Fig. G.8). Produce histograms for $\rho = 0$ and $\rho = 0.95$, each for $n_{\text{pt}} = 5, 50, 500$. Under what circumstances is the distribution asymmetric and why? Is this asymmetry in contradiction to the Central Limit theorem? (Example solution: S2MaxLike)

Programming Problem 9.1: Fit of a First-Degree Polynomial to Data that Correspond to a Second-Degree Polynomial

In experimental or empirical studies one is often confronted with a large number of measurements or objects of the same kind (animals, elementary particle collisions, industrial products from a given production process, ...). The outcomes of the measurements performed on each object are described by some law. Certain assumptions are made about that law, which are to be checked by experiment.

Consider the following example. A series of measurements may contain n_{exp} experiments. Each experiment yields the measurements $y_i = x_1 + x_2 t_i + x_3 t_i^2 + \varepsilon_i$ for 10 values $t_i = 1, 2 \ldots, 10$ of the controlled variable t. The ε_i are taken from a normal distribution with mean zero and width σ. In the analysis of the experiments it is assumed, however, that the true values η_i underlying the measurements y_i can be described by a first-degree polynomial $\eta_i = x_1 + x_2 t$. As result of the fit we obtain a minimum function M from which we can compute the "χ^2-probability" $P = 1 - F(M, n-r)$. Here $F(M, f)$ is the distribution function of a χ^2 distribution with f degrees of freedom, n is the number of data points, and r the number of parameters determined in the fit. If $P < \alpha$, then the fit of a first-degree polynomial to the data is rejected at a confidence level of $\beta = 1 - \alpha$.

Write a class performing the following steps:

(i) Interactive input of n_{exp}, x_1, x_2, x_3, σ, Δy.

(ii) Generation of n_{exp} sets of data $(t_i, y_i, \Delta y)$, fit of a first-degree polynomial to each set of data and computation of P. Entry of P into a histogram.

(iii) Graphical representation of the histogram.

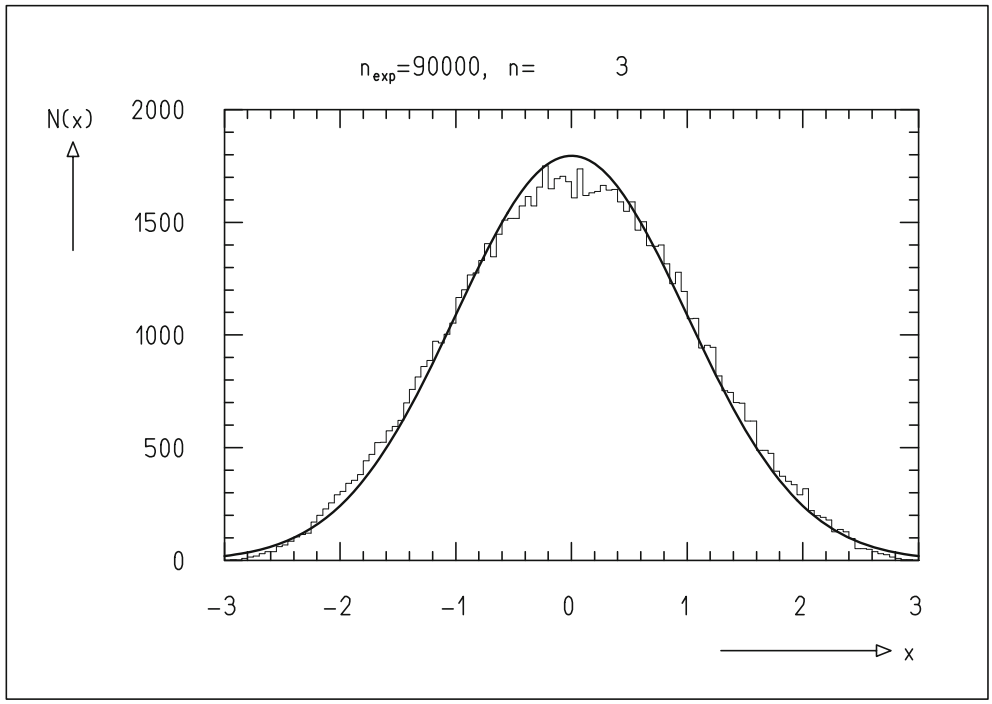

Fig. G.6: Histogram of 90 000 random numbers x, each of which is a sum of three uniformly distributed random numbers. The curve corresponds to the standard normal distribution. Significant differences between curve and histogram are visible only because of the very large number of random numbers used.

Suggestions: (a) Choose $n_{\text{exp}} = 1000$, $x_1 = x_2 = 1$, $x_3 = 0$, $\sigma = \Delta y = 1$. As expected you will obtain a flat distribution for P. (b) Choose (keeping the other input quantities as above) different values $x_3 \neq 0$. You will observe a shift of the distribution towards small P values, cf. Fig. G.9. Determine approximately the smallest positive value of x_3 such that the hypothesis of a first-degree polynomial is rejected at 90 % confidence level in 95 % of all experiments. (c) Choose $x_3 = 0$, but $\sigma \neq \Delta y$. You will again observe a shift in the distribution, e.g., towards larger P values for $\Delta y > \sigma$. (d) From the experience gained in (a), (b), and (c), one might conclude that if erroneously too large measurement errors are assumed ($\Delta y > \sigma$) then a flat P distribution would result. In this way one would get the impression that a first-degree polynomial could describe the data. Begin with $n_{\text{exp}} = 1000$, $x_1 = x_2 = 1$, $x_3 = 0.2$, $\sigma = 1$, $\Delta y = 1$ and increase Δy in steps of 0.1 up to $\Delta y = 2$. (Example solution: SILsq)

Programming Problem 9.2: Fit of a Power Law (Linear Case)

A power law

$$\eta = x t^w$$

is linear in the parameter x if w is a constant. This function is to be fitted to measurements (t_i, y_i) given by

$$t_i \quad = \quad t_0 + (i-1)\Delta t \quad , \qquad i = 1, \ldots, n \quad ,$$

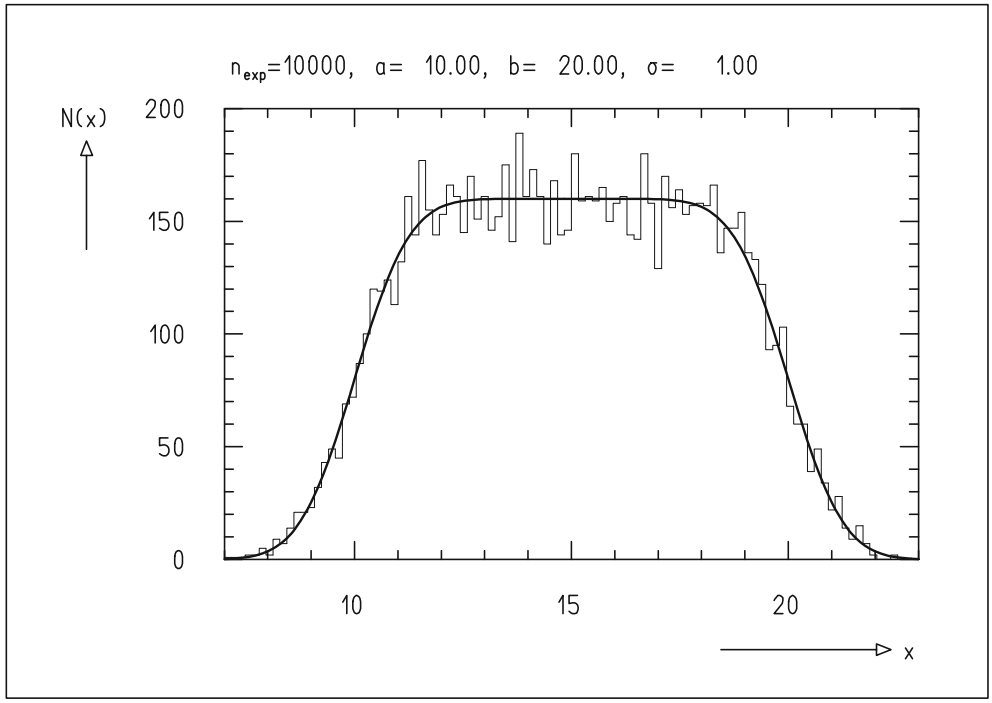

Fig. G.7: A histogram of 10 000 random numbers, each of which is the sum of a uniformly distributed random number and a normally distributed random number. The curve corresponds to the convolution of a uniform and a normal distribution.

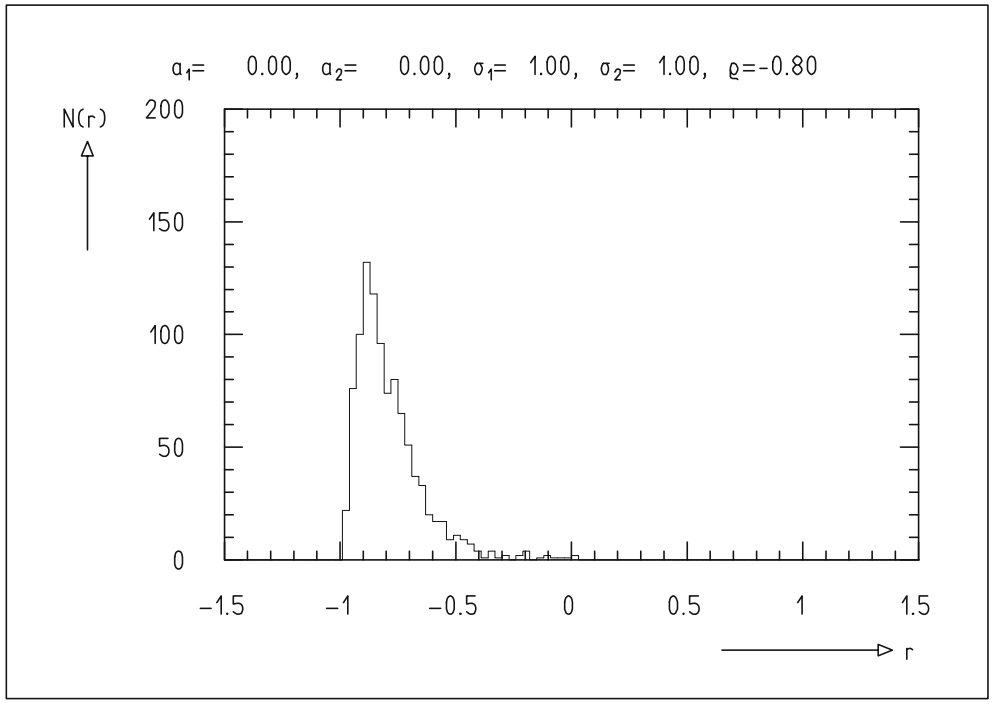

Fig. G.8: Histogram of the sample correlation coefficient computed for 1000 samples of size 10 from a bivariate Gaussian distribution with the correlation coefficient $\rho = -0.8$.

$$y_i \quad = \quad x t_i^w + \varepsilon_i \quad .$$

Here the ε_i follow a normal distribution centered about zero with width σ.

Write a class performing the following steps:

(i) Interactive input of n, t_0, Δt, x, w, σ.

(ii) Generation of measured points.

(iii) Fit of the power law.

(iv) Graphical display of the data and the fitted function, cf. Fig. G.10.

(Example solution: S2Lsq)

Programming Problem 9.3: Fit of a Power Law (Nonlinear Case)

If the power law has the form

$$\eta = x_1 t^{x_2} \quad ,$$

i.e., if the power itself is an unknown parameter, the problem becomes nonlinear. For the fit of a nonlinear function we have to start from a first approximation of the parameters. We limit ourselves to the case $t_i > 0$ for all i which occurs frequently in practice. Then one has $\ln \eta = \ln x_1 + x_2 \ln t$. If instead of (t_i, y_i) we now use

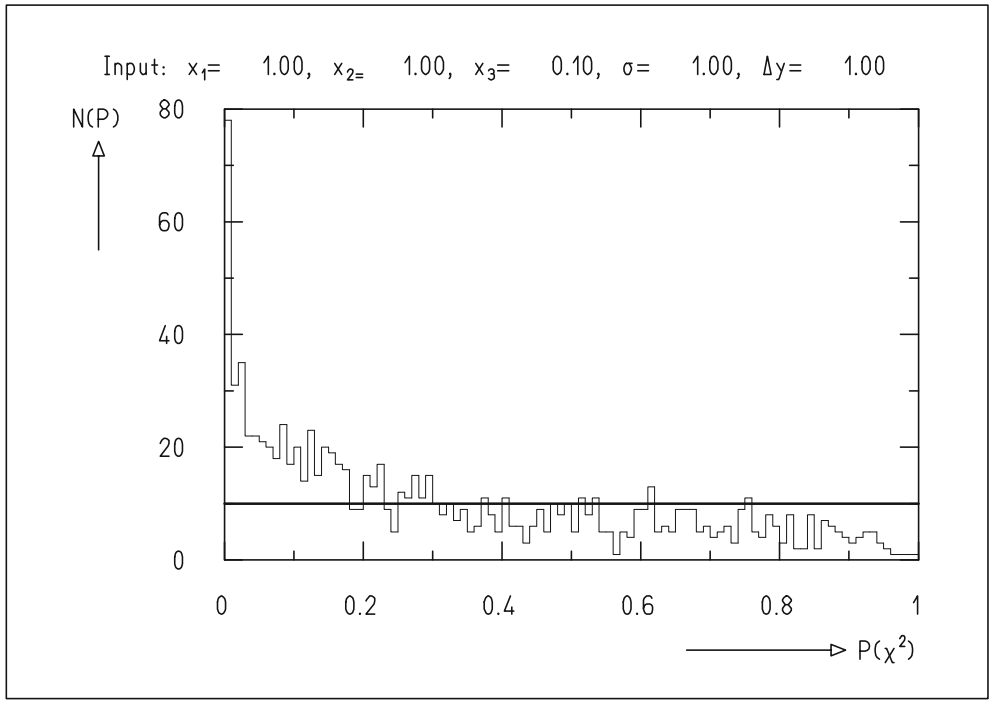

Fig. G.9: Histogram of the χ^2-probability for fits of a first-degree polynomial to 1000 data sets generated according to a second-degree polynomial.

$(\ln t_i, \ln y_i)$ as measured variables, we obtain a linear function in the parameters $\ln x_1$ and x_2. However, in this transformation the errors are distorted so that they are no longer Gaussian. We simply choose all errors to be of equal size and use the result of the linear fit as the first approximation of a nonlinear fit to the (t_i, y_i). We still have to keep in mind (in any case for $x_1 > 0$) that one always has $\eta > 0$ for $t > 0$. Because of measurement errors, however, measured values $y_i < 0$ can occur. Such points of course must not be used for the computation of the first approximation.

Write a class with the following steps:

(i) Interactive input of n, t_0, Δt, x_1, x_2, σ.

(ii) Generation of the n measured points

$$\begin{aligned} t_i &= t_0 + (i-1)\Delta t \quad , \qquad i = 1, \ldots, n \quad , \\ y_i &= x_1 t_i^{x_2} + \varepsilon_i \quad , \end{aligned}$$

where ε_i comes from a normal distribution centered about zero with width σ.

(iii) Computation of first approximations x_1, x_2 by fitting a linear function to $(\ln t_i, \ln y_i)$ with LsqLin.

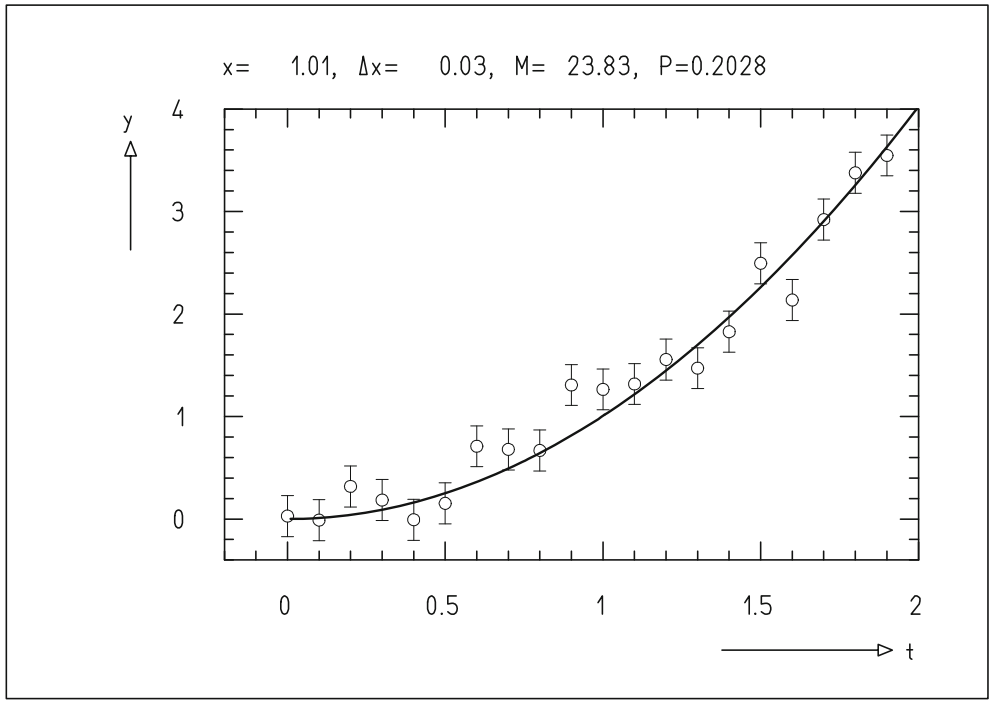

Fig. G.10: Result of the fit of a parabola $y = xt^2$ to 20 measured points.

(iv) Fit of a power law to (t_i, y_i) with LsqNon.

(v) Graphical display of the results, cf. Fig. G.11.

(Example solution: S3Lsq)

Programming Problem 9.4: Fit of a Breit–Wigner Function to Data Points with Errors

For the $N = 21$ values $t_i = -3, -2.7, \ldots, 3$ of the controlled variable the measured values

$$y_i = f(t_i) + \varepsilon_i \tag{G.3.1}$$

are to be simulated. Here,

$$f(t) = \frac{2}{\pi x_2} \frac{x_2^2}{4(t - x_1)^2 + x_2^2} \tag{G.3.2}$$

is the Breit–Wigner function (3.3.32) with $a = x_1$ and $\Gamma = x_2$. The measurement errors ε_i are to be taken from a normal distribution around zero with width σ. Choose $a = 0$ and $\Gamma = 1$. The points (t_i, y_i) scatter within the measurement errors around a bell-shaped curve with a maximum at $t = a$. A bell-shaped curve with a maximum at the same position, however, could be given by the Gaussian function,

$$f(t) = \frac{1}{x_2\sqrt{2\pi}} \exp\left\{-\frac{(t - x_1)^2}{2x_2^2}\right\} \quad . \tag{G.3.3}$$

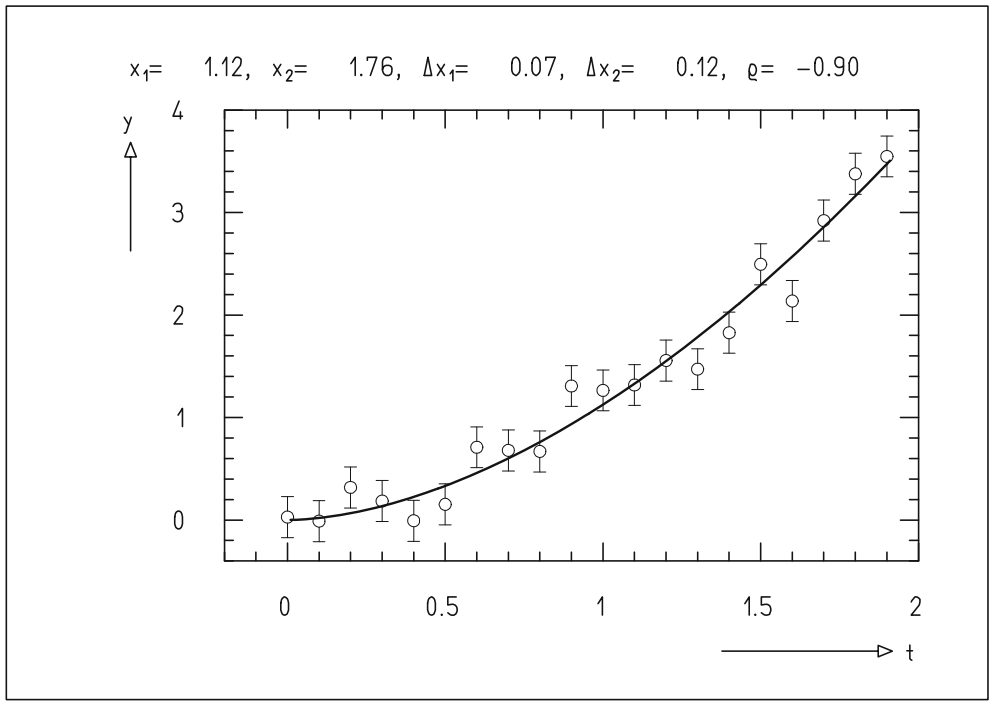

Fig. G.11: Fit of a function $y = x_1 t^{x_2}$ to 20 data points. The data points are identical to those in Fig. G.10.

Write a class with the following properties:

(i) Interactive input of σ and possibility to choose whether a Breit–Wigner function or a Gaussianis to be fitted.

(ii) Generation of the data, i.e., of the triplets of numbers $(t_i, y_i, \Delta y_i = \varepsilon_i)$.

(iii) Fit of the Breit–Wigner function (G.3.2) or of the Gaussian (G.3.3) to the data and computation of the minimum function M.

(iv) Graphical representation of the measured points with measurement errors and of the fitted function, cf. Fig. G.12.

Run the program using different values of σ and find out for which range of σ the data allow a clear discrimination between the Breit–Wigner and Gaussian functions. (Example solution: `S4Lsq`)

Programming Problem 9.5: Asymmetric Errors and Confidence Region for the Fit of a Breit–Wigner Function

Supplement the solution of Programming Problem 9.4 such that it yields a graphical representation for the parameters, their errors. covariance ellipse, asymmetric errors, and confidence region similar to Fig. 9.11. Discuss the differences obtained when fitting a Breit–Wigner or a Gaussian, respectively. For each case try $\sigma = 0.1$ and $\sigma = 1$. (Example solution: S5Lsq)

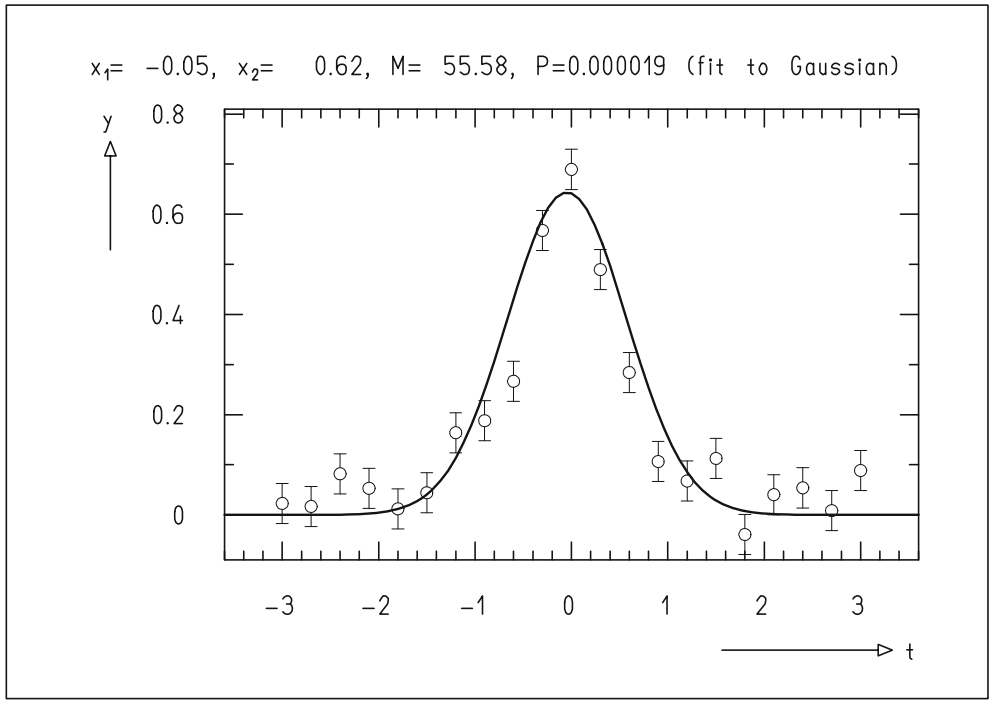

Fig. G.12: Fit of a Gaussian to data points that correspond to a Breit–Wigner function. The goodness-of-fit is poor.

Programming Problem 9.6: Fit of a Breit–Wigner Function to a Histogram

In the Programming Problem 9.4 we started from measurements $y_i = f(t_i) + \varepsilon_i$. Here $f(t)$ was a Breit–Wigner function (3.3.32), and the measurement errors ε_i corresponded to a Gaussian distribution centered about zero with width σ.

We now generate a sample of size n_{ev} from a Breit–Wigner distribution with mean $a = 0$ and full width at half maximum $\Gamma = 1$. We represent the sample by a histogram that is again characterized by triplets of numbers $(t_i, y_i, \Delta y_i)$. Now t_i is the center of the ith bin $t_i - \Delta t/2 \le t < t_i + \Delta t/2$, and y_i is the number of sample elements falling into this bin. For not too small y_i the corresponding statistical error is $\Delta y_i = \sqrt{y_i}$. For small values y_i this simple statement is problematic. It is completely wrong for $y_i = 0$. In the fit of a function to a histogram, care is therefore to be taken that empty bins (possibly also bins with few entries) are not to be considered as data points. The function to be fitted is

$$f(t) = x_3 \frac{2}{\pi x_2} \frac{x_2^2}{4(t - x_1)^2 + x_2^2} \quad . \tag{G.3.4}$$

The function is similar to (G.3.2) but there is an additional parameter x_3.

Write a class with the following steps:

(i) Interactive input of n_{ev} (sample size) and n_t (number of histogram bins). The lower limit of the histogram is to be fixed at $t = -3$, the upper limit at $t = 3$.

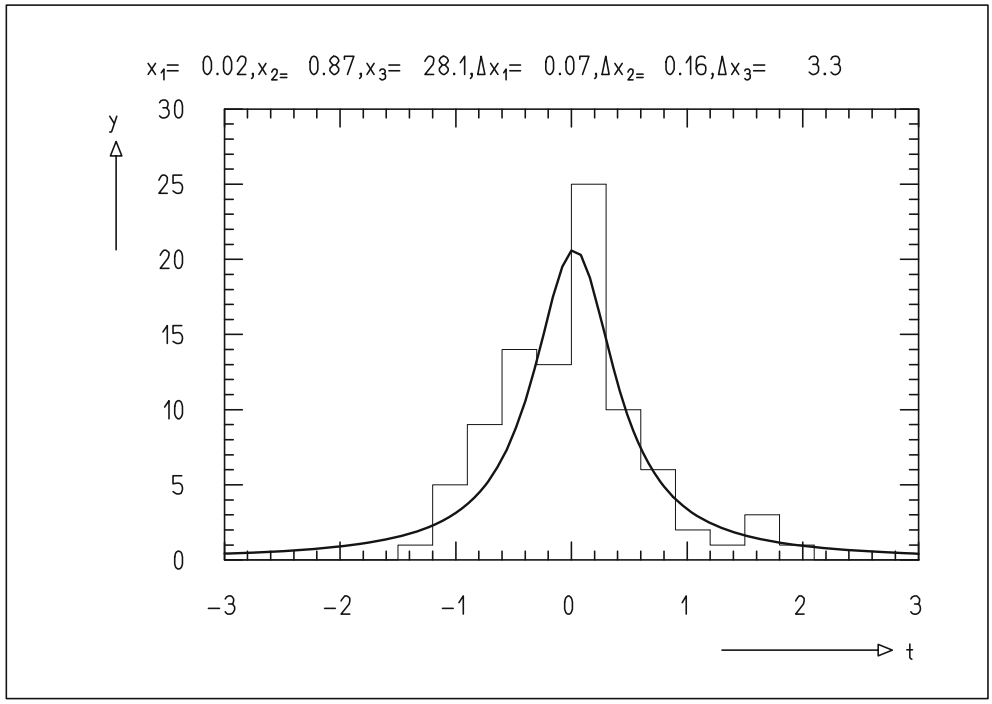

Fig. G.13: Fit of a Breit–Wigner function to a histogram.

(ii) Generation of the sample, cf. Programming Problem 4.1.

(iii) Construction of the histogram.

(iv) Construction of the triplets $(t_i, y_i, \Delta y_i)$ to be used for the fit.

(v) Fit of the function (G.3.4).

(vi) Output of the results in numerical and graphical form, cf. Fig. G.13.

Suggestions: Perform consecutive fits for the same sample but different numbers of bins. Try to find an optimum for the number of bins. (Example solution: `S6Lsq`)

Programming Problem 9.7: Fit of a Circle to Points with Measurement Errors in Abscissa and Ordinate

A total of m data points (s_i, t_i) are given in the (s, t) plane. The measurement errors are defined by 2×2 covariance matrices of the form

$$\begin{pmatrix} \Delta s_i^2 & c_i^2 \\ c_i^2 & \Delta t_i^2 \end{pmatrix} \tag{G.3.5}$$

as in Example 9.11. Here $c_i = \Delta s_i \Delta t_i \rho_i$ and ρ_i is the correlation coefficient between the measurement errors Δs_i, Δt_i. As in Example 9.11, construct the vector $\mathbf{y}$ of measurements from the s_i and t_i and construct the covariance matrix C_y. Set up

the equations $f_k(x, \boldsymbol{\eta}) = 0$ assuming that the true positions underlying the measured points lie on a circle with center (x_1, x_2) and radius x_3. Write a program with the following steps:

(i) Input of m, Δt, Δs, ρ.

(ii) Generation of m measured points (s_i, t_i) using bivariate normal distributions, the means of which are positioned at regular intervals on the unit circle ($x_1 = x_2 = 0$, $x_3 = 1$) and the covariance matrix of which is given by (G.3.5) with $\Delta s_i = \Delta s$, $\Delta t_i = \Delta t$, $c = \Delta s \Delta t \rho$.

(iii) Determination of a first approximation for x_1, x_2, x_3 by computation of the parameters of a circle through the first three measured points.

(iv) Fit to all measured points using LsqGen and a user function specially written for this problem.

(v) Graphical representation of the measured points and the fitted circle as in Fig. G.14.

(Example solution: `S6Lsq`)

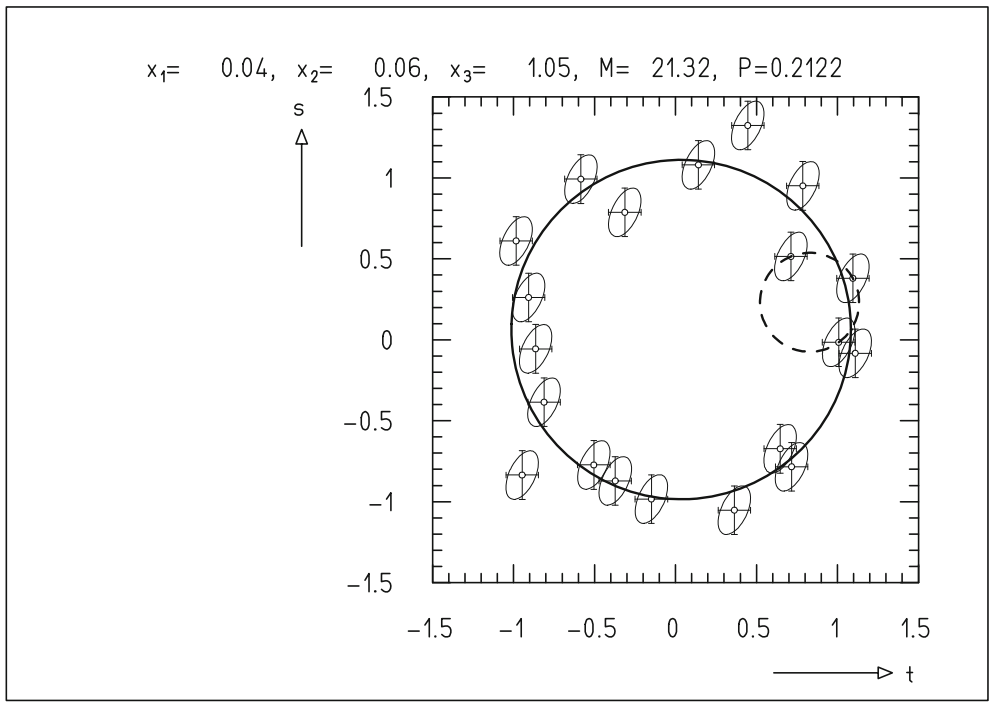

Fig. G.14: Measured points with covariance ellipses, circle of the first approximation which is given by 3 measured points (*broken line*), and circle fitted to all measured points.

Programming Problem 10.1: Monte Carlo Minimization to Choose a Good First Approximation

For some functions in Example Program 10.1 the choice of the point $\mathbf{x}_0$ defining the first approximation was decisive for success or failure of the minimization. If a function has several minima and if its value is smallest at one of them, that is if an "absolute minimum" exists, the following procedure will work. One uses the Monte Carlo method to determine a first approximation $\mathbf{x}_0$ of the absolute minimum in a larger region of the parameter space by generating points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ according to a uniform distribution in that region and by choosing that point at which the function has the smallest value.

On the basis of E1Min write class that determines the absolute minimum of the function $f_7(\mathbf{x})$ described in Sect. 10.1. A first approximation $\mathbf{x}_0$ within the region

$$-10 < x_{0i} < 10 \quad , \qquad i = 1, 2, 3 \quad ,$$

is to be determined by the Monte Carlo method. Perform the search for the first approximation with N points generated at random and allow for an interactive input of N. (Example solution: S1Min)

Programming Problem 10.2: Determination of the Parameters of a Breit–Wigner Distribution from the Elements of a Sample

By modifying a copy of E2Min produce a class that simulates a sample from a Breit–Wigner distribution with mean a and full width at half maximum Γ, and that subsequently determines the numerical values of these parameters by minimization of the negative log-likelihood function of the sample. Allow for interactive input of the sample size and the parameters a and Γ used in the simulation. (Example solution: `S2Min`)

Programming Problem 11.1: Two-Way Analysis of Variance with Crossed Classification

The model (11.2.16) for the data in an analysis of variance with crossed classification is

$$x_{ijk} = \mu + a_i + b_j + (ab)_{ij} + \varepsilon_{ijk} \quad .$$

The analysis of variance tests the null hypothesis

$$a_i = b_j = (ab)_{ij} = 0 \quad .$$

Data corresponding to this null hypothesis are generated by the program E2Anova, Sect. 11.2, and are subsequently analyzed.

Write a program similar to E2Anova which generates data x_{ijk} according to the above formula with

$$\begin{aligned} a_i &= \left(i - \frac{I+1}{2}\right) a \quad , \\ b_j &= \left(j - \frac{J+1}{2}\right) b \quad , \\ (ab)_{ij} &= \text{signum}(a_i) \quad \text{signum}(b_j) \quad ab \quad . \end{aligned}$$

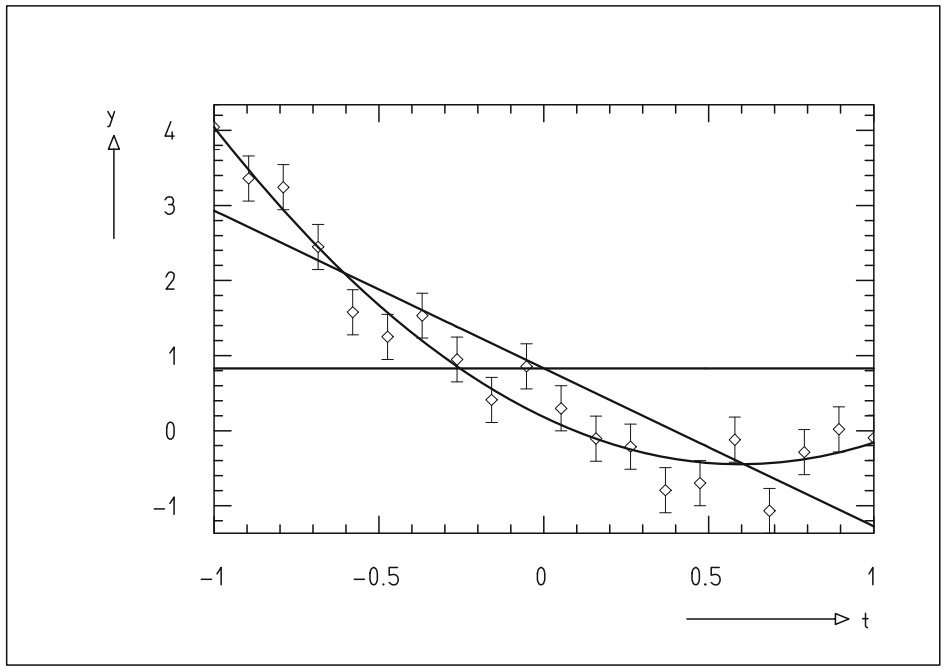

Fig. G.15: Data points with errors and regression polynomials of different degrees.

These relations fulfill the requirements (11.2.7). The ε_{ijk} as in E2Anova are to be drawn from a normal distribution with mean zero and standard deviation σ. Allow for interactive input of the quantities a, b, ab, σ, μ. Perform an analysis of variance on the simulated data. Study different cases, e.g., $a = 0$, $b \neq 0$, $ab = 0$; $a \neq 0$, $b = 0$, $ab = 0$; $a = 0$, $b = 0$, $ab \neq 0$; etc. (Example solution: S1Anova)

Programming Problem 11.2: Two-Way Analysis of Variance with Nested Classification

Modify Programming Problem 11.1 for the treatment of a nested classification with data of the form (11.2.22), e.g.,

$$x_{ijk} = \mu + a_i + b_{ij} + \varepsilon_{ijk} \quad ,$$

and use the relations

$$a_i = \left(i - \frac{I+1}{2}\right) a \quad , \qquad b_{ij} = \left(j - \frac{J+1}{2}\right) b \quad .$$

(Example solution: S2Anova)

Programming Problem 12.1: Simulation of Data and Plotting Regression Polynomials of Different Degrees

Write a class which generates n data points (t_i, y_i). The t_i are to be spread equidistantly over the interval $-1 \leq t \leq 1$; the y_i are to correspond to a polynomial with r

terms and to have errors with standard deviation σ. A regression analysis is to be performed and a plot as in Fig. G.15 of the data and the regression polynomials to be produced. Your program may be largely based on the classes E4Reg and E2Reg. (Example solution: S1Reg)

Programming Problem 12.2: Simulation of Data and Plotting the Regression Line with Confidence Limits

Extend the solution of Programming Problem 12.1 so that a regression polynomial of the desired degree together with confidence limits, corresponding to Sect. 12.2 is shown. (Example solution: S2Reg)

Programming Problem 13.1: Extrapolation in a Time Series Analysis

In Sect. 13.3 we have stressed that one must be very cautious with the interpretation of the results of a time series analysis at the edge of a time series, and of the extrapolation in regions outside the time series. In particular, we found that the extrapolation yields meaningless results if the data are not at least approximately described by a polynomial. The degree of this polynomial must be smaller than or equal to the degree of the polynomial used for the time series analysis.

Study these statements by simulating a number of time series and analyzing them. Write a program – starting from E2TimSer– that for $n = 200$, $i = 1, 2, \ldots, n$, generates data of the form

$$y_i = t_i^m + \varepsilon_i \quad , \qquad t_i = \frac{i - 100}{50} \quad .$$

Here the ε_i are to be generated according to a normal distribution with mean zero and standard deviation σ. Allow for the interactive input of m, σ, k, ℓ, and P. After generating the data perform a time series analysis and produce a plot as in E2TimSer. Study different combinations of m, k, and ℓ, and for each combination use small values of σ (e.g., $\sigma = 0.001$) and large values of σ (e.g., $\sigma = 0.1$). (Example solution: S1TimSer)

Programming Problem 13.2: Discontinuities in Time Series

In the development of time series analysis it was assumed that the measurements, apart from their statistical fluctuations, are continuous functions of time. We therefore expect unreliable results in regions where the measurements are discontinuous. Write a program that generates the following three types of time series, analyzes them, and displays the results graphically. One of them is continuous; the other two contain discontinuities.

Sine function:

$$y_i = \sin(\pi t_i/180) + \varepsilon_i \quad , \qquad t_i = i \quad , \qquad i = 1, 2, \ldots, n \quad .$$

Step function:

$$y_i = \begin{cases} -1+\varepsilon_i\,, & t_i \le 100 \\ 1+\varepsilon_i\,, & t_i > 100 \end{cases} \quad , \qquad t_i = i \bmod 200 \quad , \qquad i = 1,2,\ldots,n \quad .$$

Sawtooth function:

$$y_i = (t_i - 50)/100 + \varepsilon_i \quad , \qquad t_i = i \bmod 100 \quad , \qquad i = 1,2,\ldots,n \quad .$$

The ε_i are again to be generated according to normal distribution with mean zero and standard deviation σ. Allow for the choice of one of the functions and for the interactive input of n, σ, k, ℓ, and P. Study the time series using different values for the parameters and discuss the results. Figure G.16 shows an example. (Example solution: `S2TimSer`)

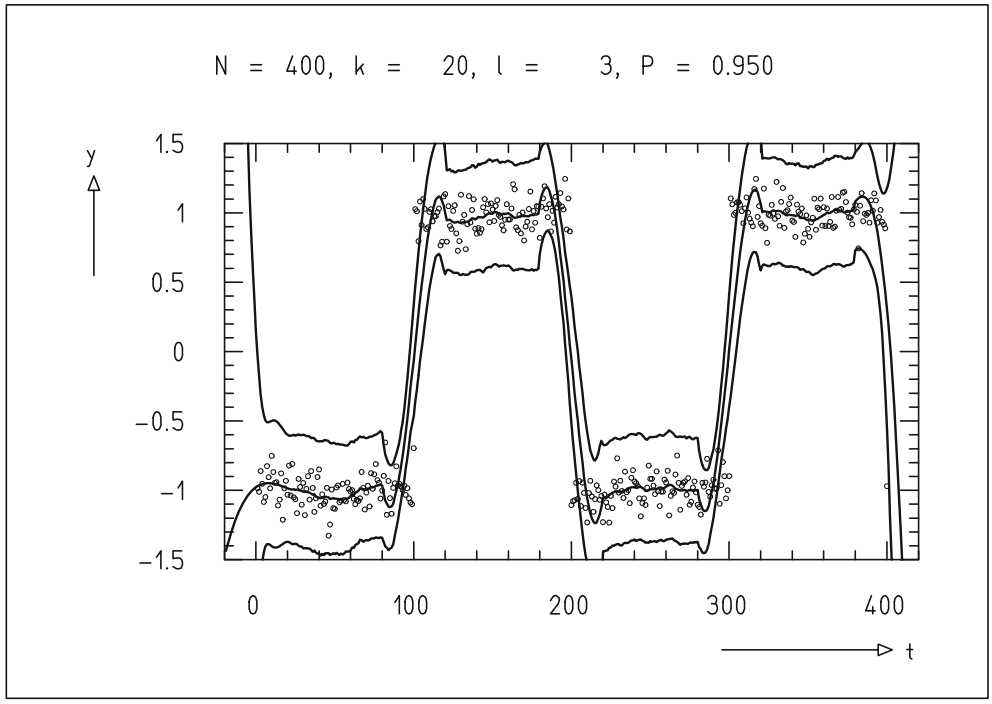

Fig. G.16: Time series corresponding to a step function with moving averages and confidence limits.

H. Collection of Formulas

Probability

$A, B, \ldots$ are *events*; $\bar{A}$ is the event "*not* A".

$(A+B)$ and (AB) combine events with *logical "or"* or *logical "and"*.

$P(A)$ is the *probability* of the event A.

$P(B|A) = P(AB)/P(A)$ is the probability for B given the condition A (*conditional probability)*.

The following rules hold:
For *every* event A

$$P(\bar{A}) = 1 - P(A) \quad ,$$

for *mutually exclusive* events $A, B, \ldots, Z$

$$P(A + B + \cdots + Z) = P(A) + P(B) + \cdots + P(Z) \quad ,$$

for *independent* events $A, B, \ldots, Z$

$$P(AB \cdots Z) = P(A)P(B) \cdots P(Z) \quad .$$

Single Random Variable

Distribution function: $\quad F(x) = P(\mathsf{x} < x)$

Probability density (for $F(x)$ differentiable):
$f(x) = F'(x) = \mathrm{d}f(x)/\mathrm{d}x$

Moments of order ℓ:

(a) About the point c: $\alpha_\ell = E\{(\mathsf{x} - c)^\ell\}$

(b) About the origin (central moments): $\lambda_\ell = E\{\mathsf{x}^\ell\}$

(c) About the mean: $\mu_\ell = E\{(\mathsf{x} - \widehat{x})^\ell\}$

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

Table H.1: Expectation values for discrete and continuous distributions.

	x discrete	x continuous; $F(x)$ differentiable
Probability density	–	$f(x) = F'(x) = \frac{\mathrm{d}F(x)}{\mathrm{d}x}$, $\int_{-\infty}^{\infty} f(x)\,\mathrm{d}x = 1$
Mean of x (expectation value)	$\widehat{x} = E(\mathsf{x})$ $= \sum_i x_i P(\mathsf{x} = x_i)$	$\widehat{x} = E(\mathsf{x})$ $= \int_{-\infty}^{\infty} x f(x)\,\mathrm{d}x$
Mean of the function $H(\mathsf{x})$	$E\{H(\mathsf{x})\}$ $= \sum_i H(x_i) P(\mathsf{x} = x_i)$	$E\{H(\mathsf{x})\}$ $= \int_{-\infty}^{\infty} H(x) f(x)\,\mathrm{d}x$

Variance: $\sigma^2(\mathsf{x}) = \mathrm{var}(\mathsf{x}) = \mu_2 = E\{(\mathsf{x} - \widehat{x})^2\}$

Standard deviation or *error* of x: $\Delta\mathsf{x} = \sigma(\mathsf{x}) = +\sqrt{\sigma^2(\mathsf{x})}$

Skewness: $\gamma = \mu_3/\sigma^3$

Reduced variable: $\mathsf{u} = (\mathsf{x} - \widehat{x})/\sigma(\mathsf{x})$; $E(\mathsf{u}) = 0$, $\sigma^2(\mathsf{u}) = 1$

Mode (most probable value) x_m defined by: $P(\mathsf{x} = x_\mathrm{m}) = \max$

Median $x_{0.5}$ defined by: $F(x_{0.5}) = P(\mathsf{x} < x_{0.5}) = 0.5$

Quantile x_q defined by: $F(x_q) = P(\mathsf{x} < x_q) = q$; $0 \le q \le 1$

Several Random Variables

Distribution function:

$$F(\mathbf{x}) = F(x_1, x_2, \ldots, x_n) = P(\mathsf{x}_1 < x_1, \mathsf{x}_2 < x_2, \ldots, \mathsf{x}_n < x_n)$$

Joint probability density (only for $F(\mathbf{x})$ differentiable with respect to all variables):

$$f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n) = \partial^n F(x_1, x_2, \ldots, x_n)/\partial x_1 \partial x_2 \ldots \partial x_n$$

Marginal probability density of the variable x_i:

$$g_i(x_i) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x_1, x_2, \ldots, x_n)\,\mathrm{d}x_1\,\mathrm{d}x_2 \ldots \mathrm{d}x_{i-1}\,\mathrm{d}x_{i+1} \ldots \mathrm{d}x_n$$

Expectation value of a function $H(\mathbf{x})$:

$$E\{H(\mathbf{x})\} = \int H(\mathbf{x}) f(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

Expectation value of the variables x_i:

$$\widehat{x_i} = E(\mathsf{x}_i) = \int x_i f(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \int_{-\infty}^{\infty} x_i g_i(x_i) \, \mathrm{d}x_i$$

The variables $\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n$ are *independent*, if

$$f(x_1, x_2, \ldots, x_n) = g_1(x_1) g_2(x_2) \ldots g_n(x_n)$$

Moments of order $\ell_1, \ell_2, \ldots, \ell_n$:

(a) About $\mathbf{c} = (c_1, c_2, \ldots, c_n)$:
$\alpha_{\ell_1 \ell_2 \ldots \ell_n} = E\{(\mathsf{x}_1 - c_1)^{\ell_1} (\mathsf{x}_2 - c_2)^{\ell_2} \cdots (\mathsf{x}_n - c_n)^{\ell_n}\}$

(b) About the origin: $\lambda_{\ell_1 \ell_2 \ldots \ell_n} = E\{\mathsf{x}_1^{\ell_1} \mathsf{x}_2^{\ell_2} \cdots \mathsf{x}_n^{\ell_n}\}$

(c) About $\widehat{\mathbf{x}}$: $\mu_{\ell_1 \ell_2 \ldots \ell_n} = E\{(\mathsf{x}_1 - \widehat{x}_1)^{\ell_1} (\mathsf{x}_2 - \widehat{x}_2)^{\ell_2} \cdots (\mathsf{x}_n - \widehat{x}_n)^{\ell_n}\}$

Variance of x_i: $\sigma^2(\mathsf{x}_i) = E\{(\mathsf{x}_i - \widehat{x}_i)^2\} = c_{ii}$

Covariance between x_i and x_j: $\operatorname{cov}(\mathsf{x}_i, \mathsf{x}_j) = E\{(\mathsf{x}_i - \widehat{x}_i)(\mathsf{x}_j - \widehat{x}_j)\} = c_{ij}$
For $\mathsf{x}_i, \mathsf{x}_j$ *independent*: $\operatorname{cov}(\mathsf{x}_i, \mathsf{x}_j) = 0$

Covariance matrix: $C = E\{(\mathbf{x} - \widehat{\mathbf{x}})(\mathbf{x} - \widehat{\mathbf{x}})^{\mathrm{T}}\}$

Correlation coefficient:

$$\rho(\mathsf{x}_1, \mathsf{x}_2) = \operatorname{cov}(\mathsf{x}_1, \mathsf{x}_2) / \sigma(\mathsf{x}_1) \sigma(\mathsf{x}_2) \quad ; \qquad -1 \le \rho \le 1$$

Rules of computation:

$$\sigma^2(c\mathsf{x}_i) = c^2 \sigma^2(\mathsf{x}_i) \quad ,$$
$$\sigma^2(a\mathsf{x}_i + b\mathsf{x}_j) = a^2 \sigma^2(\mathsf{x}_i) + b^2 \sigma^2(\mathsf{x}_j) + 2ab \operatorname{cov}(\mathsf{x}_i, \mathsf{x}_j) \quad ;$$
a, b, c are constants

Transformation of Variables

Original variables: $\mathbf{x} = (\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_n)$

Probability density: $f(\mathbf{x})$

Transformed variables: $\mathbf{y} = (\mathsf{y}_1, \mathsf{y}_2, \ldots, \mathsf{y}_n)$

Mapping: $\mathsf{y}_1 = \mathsf{y}_1(\mathbf{x})$, $\mathsf{y}_2 = \mathsf{y}_2(\mathbf{x})$, …, $\mathsf{y}_n = \mathsf{y}_n(\mathbf{x})$

Probability density: $g(\mathbf{y}) = |J| f(\mathbf{x})$

with the *Jacobian determinant*

$$J = J\left(\frac{x_1, x_2, \ldots, x_n}{y_1, y_2, \ldots, y_n}\right) = \begin{vmatrix} \frac{\partial x_1}{\partial y_1} & \frac{\partial x_2}{\partial y_1} & \cdots & \frac{\partial x_n}{\partial y_1} \\ \vdots & & & \\ \frac{\partial x_1}{\partial y_n} & \frac{\partial x_2}{\partial y_n} & \cdots & \frac{\partial x_n}{\partial y_n} \end{vmatrix}$$

Error Propagation

The original variables $\mathbf{x}$ have the covariance matrix C_x. The *covariance matrix of the transformed variables* $\mathbf{y}$ is

$$C_y = T C_x T^{\mathrm{T}} \qquad \text{with} \qquad T = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{pmatrix} .$$

The formula is only exact for a linear relation between $\mathbf{y}$ and $\mathbf{x}$, but is a good approximation for small deviations from linearity in a region around $\widehat{\mathbf{x}}$ of the magnitude of the standard deviation. *Only for vanishing covariances* in C_x does one have

$$\sigma(y_i) = \Delta y_i = \sqrt{\sum_{j=1}^{m} \left(\frac{\partial y_i}{\partial x_j}\right)^2 (\Delta x_j)^2} \quad .$$

The Law of Large Numbers

A total of n observations are carried out, which are characterized by the random variable x_i (= 1, if on the ith observation the event A occurs, otherwise = 0). The *frequency* of A is

$$\mathsf{h} = \frac{1}{n} \sum_{i=1}^{n} \mathsf{x}_i \quad .$$

For $n \to \infty$ this frequency is equal to the probability p for the occurrence of A,

$$E(\mathsf{h}) = h = p \quad , \qquad \sigma^2(\mathsf{h}) = \frac{1}{n} p(1-p) \quad .$$

Table H.2: Distributions of discrete variables.

Distribution	Probability for observing $\mathsf{x} = k$ ($\mathsf{x}_1 = k_1, \ldots, \mathsf{x}_l = k_l$)	Mean	Variance (elements of covariance matrix)
Binomial	$W_k^n = \binom{n}{k} p^k (1-p)^{n-k}$	$\widehat{x} = np$	$\sigma^2(\mathsf{x}) = np(1-p)$
Multinomial $\sum_{j=1}^{l} p_j = 1$	$W^n_{k_1,k_2,\ldots,k_l} = \frac{n!}{\prod_{j=1}^{l} k_j!} \prod_{j=1}^{l} p_j^{kj}$	$\widehat{x}_j = np_j$	$c_{ij} = np_i(\delta_{ij} - p_j)$
Hypergeometric $L = N - K$, $l = n - k$	$W_k = \binom{K}{k}\binom{L}{\ell} : \binom{N}{n}$	$\widehat{x} = n\frac{K}{N}$	$\sigma^2(\mathsf{x}) = \frac{nKL(N-n)}{N^2(N-1)}$
Poisson	$f(k) = \frac{\lambda^k}{k!}\mathrm{e}^{-\lambda}$	$\widehat{x} = \lambda$	$\sigma^2(\mathsf{x}) = \lambda$

Central Limit Theorem

If x_i are independent variables with mean a and variance σ^2, then $(1/n)$ $\sum_{i=1}^{n} \mathsf{x}_i$ for $n \to \infty$ follows a normal distribution with mean a and variance σ^2/n.

Convolutions of Distributions

The probability density of the sum $\mathsf{u} = \mathsf{x} + \mathsf{y}$ of two independent random variables x and y is

$$f_u(u) = \int_{-\infty}^{\infty} f_x(x) f_y(u-x)\,\mathrm{d}x = \int_{-\infty}^{\infty} f_y(y) f_x(u-y)\,\mathrm{d}y \quad .$$

- The convolution of two Poisson distributions with the parameters λ_1 and λ_2 is a Poisson distribution with the parameter $\lambda_1 + \lambda_2$.
- The convolution of two Gaussian distributions with means a_1, a_2 and variances σ_1^2, σ_2^2 is a Gaussian distribution with mean $a = a_1 + a_2$ and variance $\sigma^2 = \sigma_1^2 + \sigma_2^2$.

Table H.3: Distributions of continuous variables.

Distribution	Probability density	Mean	Variance (covariance matrix)
Uniform	$\begin{cases} 0;\ x > a,\ x \geq b \\ \frac{1}{b-a};\ a \leq x < b \end{cases}$	$\frac{1}{2}(b+a)$	$(b-a)^2/12$
Gaussian	$\frac{1}{b\sqrt{2\pi}} \exp\left(-\frac{(x-a)^2}{2b^2}\right)$	a	b^2
Stand. Gaussian	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$	0	1
Gaussian of several variables	$k \exp\left\{-\frac{(\mathbf{x}-\mathbf{a})^{\mathrm{T}} B (\mathbf{x}-\mathbf{a})}{2}\right\}$	$\mathbf{a}$	$C = B^{-1}$
χ^2	$\frac{1}{\Gamma(\frac{1}{2}f) 2^{\frac{1}{2}f}} (\chi^2)^{\frac{1}{2}f-1} \exp(-\frac{1}{2}\chi^2)$	f	$2f$
Fisher's F	$\left(\frac{f_1}{f_2}\right)^{\frac{1}{2}f_1} \frac{\Gamma(\frac{1}{2}(f_1+f_2))}{\Gamma(\frac{1}{2}f_1)\Gamma(\frac{1}{2}f_2)} \times F^{\frac{1}{2}f_1-1} \left(1+\frac{f_1}{f_2}F\right)^{-\frac{1}{2}(f_1+f_2)}$	$\frac{f_2}{f_2-2}$, $f_2 > 2$	$\frac{2f_2^2(f_1+f_2-2)}{f_1(f_2-2)^2(f_2-4)}$, $f_2 > 4$
Student's t	$\frac{\Gamma(\frac{1}{2}(f+1))}{\Gamma(\frac{1}{2}f)\sqrt{\pi}\sqrt{f}} \left(1+\frac{t^2}{f}\right)^{-\frac{1}{2}(f+1)}$	0	$\frac{f}{f-2}$, $f > 2$

– The convolution of two χ^2-distributions with f_1 and f_2 degrees of freedom is a χ^2-distribution with $f = f_1 + f_2$ degrees of freedom.

Samples

Population: An infinite (in some cases finite) set of elements described by a discrete or continuously distributed random variable x.

(Random) sample of *size* N: A selection of N elements $(\mathsf{x}^{(1)}, \mathsf{x}^{(2)}, \ldots, \mathsf{x}^{(N)})$ from the population. (For the requirements for a sample to be random see Sect. 6.1.)

Table H.4: Samples from different populations.

	Sample of size n from a continuously distributed population	Sample of size n from a discrete population of size N. Variable of the population is y, variable of the sample is x
Population mean	$E(\mathsf{x}) = \widehat{x}$	$\bar{y} = \frac{1}{N}\sum_{j=1}^{N} y_j$
Population variance	$\sigma^2(\mathsf{x})$	$\sigma^2(y) = \frac{1}{N-1}\sum_{j=1}^{N}(y_j - \bar{y})^2$
Sample mean	$\bar{\mathsf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathsf{x}_i$	$\bar{\mathsf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathsf{x}_i$
Sample variance mean	$\sigma^2(\bar{\mathsf{x}}) = \frac{1}{n}\sigma^2(\mathsf{x})$	$\sigma^2(\bar{\mathsf{x}}) = \frac{\sigma^2(y)}{n}\left(1 - \frac{n}{N}\right)$
Variance of the sample	$\mathsf{s}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\mathsf{x}_i - \bar{\mathsf{x}})^2$	$\mathsf{s}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\mathsf{x}_i - \bar{\mathsf{x}})^2$

Distribution function of the sample: $W_n(x) = n_x/N$,
where n_x is the number of elements in the sample for which $\mathsf{x} < x$.

Statistic: An arbitrary function of the elements of a sample

$$\mathsf{S} = \mathsf{S}(\mathsf{x}^{(1)}, \mathsf{x}^{(2)}, \ldots, \mathsf{x}^{(N)}) \quad .$$

Estimator: A statistic used to estimate a parameter λ of the population. An estimator is *unbiased*, if $E(\mathsf{S}) = \lambda$ and *consistent*, if

$$\lim_{N\to\infty} \sigma(\mathsf{S}) = 0 \quad .$$

Maximum-Likelihood Method

Consider a population described by the probability density $f(\mathbf{x}, \boldsymbol{\lambda})$, where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_p)$ is a set of parameters. If a sample $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, ..., $\mathbf{x}^{(N)}$ is obtained, then the *likelihood function* is $L = \prod_{j=1}^{N} f(\mathbf{x}^{(j)}, \boldsymbol{\lambda})$ and the *log-likelihood function* is $\ell = \ln L$. In order to determine the unknown parameters

$\boldsymbol{\lambda}$ from the sample the maximum-likelihood method prescribes the value of $\boldsymbol{\lambda}$ for which L (or ℓ) is a maximum. That is, one must solve the *likelihood equation* $\partial\ell/\partial\lambda_i = 0$; $i = 1, 2, \ldots, p$ or (for only one parameter) $\mathrm{d}\ell/\mathrm{d}\lambda = \ell' = 0$.

Information of a sample: $I(\lambda) = E(\ell'^2) = -E(\ell'')$

Information inequality: $\sigma^2(\mathsf{S}) \geq \{1 - B'(\lambda)\}^2 / I(\lambda)$
Here S is an *estimator* for λ, and $B(\lambda) = E(\mathsf{S}) - \lambda$ is its *bias*.

An estimator has *minimum variance* if $\ell' = A(\lambda)(\mathsf{S} - E(\mathsf{S}))$, where $A(\lambda)$ does not depend on the sample.

The maximum-likelihood estimator $\widetilde{\lambda}$, i.e., the solution of the likelihood equation, is *unique*, *asymptotically unbiased* (i.e., for $N \to \infty$), and has *minimum variance*.

The *asymptotic form* of the likelihood function for one parameter λ is

$$L = \text{const} \cdot \exp\left(-\frac{(\lambda - \widetilde{\lambda})^2}{2b^2}\right) \quad ,$$

$$b^2 = \sigma^2(\widetilde{\lambda}) = \frac{1}{E(\ell'^2(\widetilde{\lambda}))} = -\frac{1}{E(\ell''(\widetilde{\lambda}))}$$

and for several parameters

$$L = \text{const} \cdot \left\{-\frac{1}{2}(\boldsymbol{\lambda} - \widetilde{\boldsymbol{\lambda}})^{\mathrm{T}} B (\boldsymbol{\lambda} - \widetilde{\boldsymbol{\lambda}})\right\}$$

with the covariance matrix $C = B^{-1}$ and

$$B_{ij} = -E\left(\frac{\partial^2 \ell}{\partial\lambda_i \partial\lambda_j}\right)_{\boldsymbol{\lambda} = \widetilde{\boldsymbol{\lambda}}} \quad .$$

Testing Hypotheses

Null hypothesis $H_0(\boldsymbol{\lambda} = \boldsymbol{\lambda}_0)$: Assumption of values for the parameters $\boldsymbol{\lambda}$ that determine the probability distribution $f(\mathbf{x}, \boldsymbol{\lambda})$ of a population.

Alternative hypotheses $H_1(\boldsymbol{\lambda} = \boldsymbol{\lambda}_1)$, $H_2(\boldsymbol{\lambda} = \boldsymbol{\lambda}_2)$, ...: Other possibilities for $\boldsymbol{\lambda}$, against which the null hypothesis is to be tested by consideration of a sample $X = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)})$ from the population.

A hypothesis is *simple* if the parameters are completely determined, e.g., $H_0(\lambda_1 = 1, \lambda_2 = 5)$, otherwise it is *composite*, e.g., $H_1(\lambda_1 = 2, \lambda_2 < 7)$.

Test of a hypothesis H_0 with a *significance level* α or *confidence level* $1 - \alpha$: H_0 is rejected if $X \in S_c$, where S_c is the *critical region* in the sample space and

$$P(X \in S_c | H_0) = \alpha \quad .$$

Error of the first kind: Rejection of H_0, although H_0 is true. The probability of this error is α.

Error of the second kind: H_0 is not rejected although H_1 is true. The probability of this error is $P(X \notin S_c|H_1) = \beta$.

Power function:

$$M(S_c, \boldsymbol{\lambda}) = P(X \in S_c|H) = P(X \in S_c|\boldsymbol{\lambda}) \quad .$$

Operating characteristic function:

$$L(S_c, \boldsymbol{\lambda}) = 1 - M(S_c, \boldsymbol{\lambda}) \quad .$$

Most powerful test of H_0 with respect to H_1 has $M(S_c, \boldsymbol{\lambda}_1) = 1 - \beta = \max$. A *uniformly most powerful test* is a most powerful test with respect to all possible H_1.

An *unbiased test* has $M(S_c, \boldsymbol{\lambda}_1) \geq \alpha$ for all possible H_1.

NEYMAN–PEARSON LEMMA: A test of H_0 with respect to H_1 (both simple hypotheses) with the critical region S_c is a most powerful test if $f(X|H_0)/f(X|H_1) \leq c$ for $X \in S_c$ and $\geq c$ for $X \notin S_c$, where c is a constant only depending on α.

Test statistic $T(X)$: Scalar function of the sample X. By means of a mapping $X \to T(X)$, $S_c(X) \to U$ the question as to whether $X \in S_c$ can be reformulated as $T \in U$.

Likelihood-ratio test: If ω denotes the region in the parameter space corresponding to the null hypothesis and Ω denotes the entire possible parameter region, then the test statistic

$$T = f(x; \tilde{\boldsymbol{\lambda}}^{(\Omega)})/f(x; \tilde{\boldsymbol{\lambda}}^{(\omega)})$$

is used. Here $\tilde{\boldsymbol{\lambda}}^{(\Omega)}$ and $\tilde{\boldsymbol{\lambda}}^{(\omega)}$ are the maximum-likelihood estimators in the regions Ω and ω. H_0 is rejected if $T > T_{1-\alpha}$ with $P(T > T_{1-\alpha}|H_0) = \int_{T_{1-\alpha}}^{\infty} g(T)\,\mathrm{d}T = \alpha$; $g(T)$ is the conditional probability density of T for a given H_0.

WILKS *theorem* (holds for weak requirements concerning the probability density of the population): If H_0 specifies $p - r$ out of the p parameters, then $-2\ln T$ (where T is the likelihood-ratio test statistic) follows a χ^2-distribution with $f = p - r$ degrees of freedom in the limit $n \to \infty$.

χ^2-Test for Goodness-of-Fit

Hypothesis: The N measured values y_i with normally distributed errors σ_i are described by given quantities f_i.

Table H.5: Frequently used statistical tests for a sample from a normal distribution with mean λ and variance σ^2. (Case 1: σ known; case 2: σ unknown (Student's test); case 3: χ^2-test of the variance; case 4: Student's difference test of two samples of sizes N_1 and N_2 on the significance of s_Δ^2, cf. (8.3.19); case 5: F-test of two samples).

Case	Test statistic	Null hypothesis	Critical region for test statistic	Number of degrees of freedom
1	$T = \frac{\bar{\mathsf{x}}-\lambda_0}{\sigma/\sqrt{N}}$, $\bar{\mathsf{x}} = \frac{1}{N}\sum_{j=1}^{N} \mathsf{x}^{(j)}$	$\lambda = \lambda_0$ $\lambda \le \lambda_0$ $\lambda \ge \lambda_0$	$\lvert T\rvert > \Omega(1-\alpha/2)$ $T > \Omega(1-\alpha)$ $T < \Omega(\alpha)$	–
2	$T = \frac{\bar{\mathsf{x}}-\lambda_0}{\mathsf{s}/\sqrt{N}}$, $\mathsf{s}^2 = \frac{1}{N-1} \times \sum_{j=1}^{N} (\mathsf{x}^{(j)} - \bar{\mathsf{x}})^2$	$\lambda = \lambda_0$ $\lambda \le \lambda_0$ $\lambda \ge \lambda_0$	$\lvert T\rvert > t_{1-\alpha/2}$ $T > t_{1-\alpha}$ $T < -t_{1-\alpha} = t_\alpha$	$N-1$
3	$T = (N-1)\frac{\mathsf{s}^2}{\sigma_0^2}$	$\sigma^2 = \sigma_0^2$ $\sigma^2 \le \sigma_0^2$ $\sigma^2 \ge \sigma_0^2$	$\chi^2_{1-\alpha/2} < T < \chi^2_{\alpha/2}$ $T > \chi^2_{1-\alpha}$ $T < \chi^2_\alpha$	$N-1$
4	$T = \frac{\bar{\mathsf{x}}_1-\bar{\mathsf{x}}_2}{\mathsf{s}_\Delta}$	$\lambda_1 = \lambda_2$	$\lvert T\rvert > t_{1-\frac{1}{2}\alpha}$	N_1+N_2-2
5	$T = \mathsf{s}_1^2/\mathsf{s}_2^2$, $\mathsf{s}_i^2 = \frac{1}{N_i-1} \times \sum_{j=1}^{N_i} (\mathsf{x}_i^{(j)} - \bar{\mathsf{x}}_i)^2$	$\sigma_1^2 = \sigma_2^2$ $\sigma_1^2 \le \sigma_2^2$ $\sigma_1^2 \ge \sigma_2^2$	$F_{1-\alpha/2} < T < F_{\alpha/2}$ $T > F_{1-\alpha}$ $T < F_\alpha$	$f_1 = N_1 - 1$ $f_2 = N_2 - 1$

Test function: $T = \sum_{i=1}^{N} (y_i - f_i)^2 / \sigma_i^2$.

Critical region: $T > \chi^2_{1-\alpha}$.

Number of degrees of freedom: N (or $N - p$, if p parameters are determined from the measurements).

The Method of Least Squares

One considers a set of m *equations* $f_k(\mathbf{x}, \boldsymbol{\eta}) = 0$; $k = 1, \ldots, m$ relating the r-vector of the *unknowns* $\mathbf{x} = (x_1, x_2, \ldots, x_r)$ to the n-vector of *measurable quantities* $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n)$. Instead of $\boldsymbol{\eta}$, the quantities $\mathbf{y}$ are measured, which deviate from them by the *measurement errors* $\boldsymbol{\varepsilon}$, i.e., $\mathbf{y} = \boldsymbol{\eta} + \boldsymbol{\varepsilon}$. The quantities $\boldsymbol{\varepsilon}$ are assumed to be normally distributed about zero. This is expressed by the covariance matrix $C_y = G_y^{-1}$. In order to obtain the solution $\widetilde{\mathbf{x}}$, $\widetilde{\boldsymbol{\eta}}$, one expands the f_k for the *first approximations* $\mathbf{x}_0$, $\boldsymbol{\eta}_0 = \mathbf{y}$. Only the linear terms of the expansion are kept and the second approximation $\mathbf{x}_1 = \mathbf{x}_0 + \boldsymbol{\xi}$, $\boldsymbol{\eta}_1 = \boldsymbol{\eta} + \boldsymbol{\delta}$ is computed. The procedure is repeated iteratively until certain convergence criteria are met, for example, until a scalar function M no longer decreases. If the f_k are linear in $\mathbf{x}$ and $\boldsymbol{\eta}$, then only one step is necessary. The method can be interpreted as a procedure to minimize M. The function M corresponding to the solution depends on the measurement errors. It is a random variable and follows a χ^2-distribution with $f = m - r$ degrees of freedom. It can therefore be used for a χ^2-*test* of the goodness-of-fit or of other assumptions, in particular the assumption $C_y = G_y^{-1}$. If the errors $\boldsymbol{\varepsilon}$ are not normally distribution, but still distributed symmetrically about zero, then the least-squares solution $\widetilde{\mathbf{x}}$ still has the smallest possible variance, and one has $E(M) = m - r$ *(Gauss–Markov theorem)*.

Table H.6: Least squares in the general case and in the case of constrained measurements.

	General case	Constrained measurements
Equations	$f_k(\mathbf{x}, \boldsymbol{\eta}) = 0,\ k = 1, \ldots, m$	$f_k(\boldsymbol{\eta}) = 0$
First approximations	$\mathbf{x}_0,\ \boldsymbol{\eta}_0 = \mathbf{y}$ $\mathbf{f} = A\boldsymbol{\xi} + B\boldsymbol{\delta} + \mathbf{c} + \cdots$	$\boldsymbol{\eta}_0 = \mathbf{y}$ $\mathbf{f} = B\boldsymbol{\delta} + \mathbf{c} + \cdots$
Equations expanded	$\{A\}_{kl} = (\partial f_k/\partial x_l)_{\mathbf{x}_0, \boldsymbol{\eta}_0}$ $\{B\}_{kl} = (\partial f_k/\partial \eta_l)_{\mathbf{x}_0, \boldsymbol{\eta}_0}$ $\mathbf{c} = \mathbf{f}(\mathbf{x}_0,\ \boldsymbol{\eta}_0)$	$\{B\}_{kl} = (\partial f_k/\partial \eta_l)_{\boldsymbol{\eta}_0}$ $\mathbf{c} = \mathbf{f}(\boldsymbol{\eta}_0)$
Covariance matrix of the measurements	$C_y = G_y^{-1}$	$C_y = G_y^{-1}$
Corrections	$\widetilde{\boldsymbol{\xi}} = -(A^{\mathrm{T}} G_B A)^{-1} A^{\mathrm{T}} G_B \mathbf{c}$ $\widetilde{\boldsymbol{\delta}} = -G_y^{-1} B^{\mathrm{T}} G_B (A\widetilde{\boldsymbol{\xi}} + \mathbf{c})$ $G_B = (B G_y^{-1} B^{\mathrm{T}})^{-1}$	$\widetilde{\boldsymbol{\delta}} = -G_y^{-1} B^{\mathrm{T}} G_B \mathbf{c}$ $G_B = (B G_y^{-1} B^{\mathrm{T}})^{-1}$
Next step	$\mathbf{x}_1 = \mathbf{x}_0 + \widetilde{\boldsymbol{\xi}},\ \boldsymbol{\eta}_1 = \boldsymbol{\eta}_0 + \widetilde{\boldsymbol{\delta}}$, new values for A, B, $\mathbf{c}$, $\widetilde{\boldsymbol{\xi}}$, $\widetilde{\boldsymbol{\delta}}$	$\boldsymbol{\eta}_1 = \boldsymbol{\eta}_0 + \widetilde{\boldsymbol{\delta}}$, new values for B, $\mathbf{c}$, $\widetilde{\boldsymbol{\delta}}$
Solution (after s steps)	$\widetilde{\mathbf{x}} = \mathbf{x}_{s-1} + \widetilde{\boldsymbol{\xi}}$, $\widetilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_{s-1} + \widetilde{\boldsymbol{\delta}}$, $\widetilde{\boldsymbol{\varepsilon}} = \mathbf{y} - \widetilde{\boldsymbol{\eta}}$	$\widetilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_{s-1} + \widetilde{\boldsymbol{\delta}}$, $\widetilde{\boldsymbol{\varepsilon}} = \mathbf{y} - \widetilde{\boldsymbol{\eta}}$
Minimum function	$M = (B\widetilde{\boldsymbol{\varepsilon}})^{\mathrm{T}} G_B (B\widetilde{\boldsymbol{\varepsilon}})$	$M = (B\widetilde{\boldsymbol{\varepsilon}})^{\mathrm{T}} G_B (B\widetilde{\boldsymbol{\varepsilon}})$
Covariance matrices	$G_{\widetilde{x}}^{-1} = (A^{\mathrm{T}} G_B A)^{-1}$ $G_{\widetilde{\eta}}^{-1} = G_y^{-1}$ $-G_y^{-1} B^{\mathrm{T}} G_B B G_y^{-1}$ $+ G_y^{-1} B^{\mathrm{T}} G_B A (A^{\mathrm{T}} G_B A)^{-1}$ $\times A^{\mathrm{T}} G_B B G_y^{-1}$	$G_{\widetilde{\eta}}^{-1} = G_y^{-1}$ $- G_y^{-1} B^{\mathrm{T}} G_B B G_y^{-1}$

Table H.7: Least squares for indirect and direct measurements.

	Indirect measurements	Direct measurements of different accuracy
Equations	$f_k = \eta_k - g_k(x) = 0$	$f_k = \eta_k - x$
First approximations	$\mathbf{x}_0,\ \boldsymbol{\eta}_0 = \mathbf{y}$	$x_0 = 0,\ \boldsymbol{\eta}_0 = \mathbf{y}$
Equations expanded	$\mathbf{f} = A\widetilde{\boldsymbol{\xi}} - \boldsymbol{\varepsilon} + \mathbf{c} + \cdots$ $\{A\}_{kl} = \left(\frac{\partial f_k}{\partial x_l}\right)_{\mathbf{x}_0}$ $\mathbf{c} = \mathbf{y} - \mathbf{g}(\mathbf{x}_0)$	$\mathbf{f} = A\widetilde{\boldsymbol{\xi}} + \boldsymbol{\varepsilon} + \mathbf{c}$ $A = -\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ $\mathbf{c} = \mathbf{y}$
Covariance matrix of the measurements	$C_y = G_y^{-1}$	$C_y = G_y^{-1}$ $= \begin{pmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_n^2 \end{pmatrix}$
Corrections	$\widetilde{\boldsymbol{\xi}} = -(A^{\mathrm{T}} G_y A)^{-1} A^{\mathrm{T}} G_y \mathbf{c}$	–
Next step	$\mathbf{x}_1 = \mathbf{x}_0 + \widetilde{\boldsymbol{\xi}}$, new values for A, $\mathbf{c}$, $\widetilde{\boldsymbol{\xi}}$	–
Solution (after s steps)	$\widetilde{\mathbf{x}} = \mathbf{x}_{s-1} + \widetilde{\boldsymbol{\xi}}$, $\widetilde{\boldsymbol{\varepsilon}} = A\widetilde{\boldsymbol{\xi}} + \mathbf{c}$	$\widetilde{\xi} = \widetilde{x} = \frac{\sum_k y_k/\sigma_k^2}{\sum_k 1/\sigma_k^2}$, $\widetilde{\varepsilon}_k = y_k - \widetilde{x}$
Minimum function	$M = \widetilde{\boldsymbol{\varepsilon}}^{\mathrm{T}} G_y \widetilde{\boldsymbol{\varepsilon}}$	$M = \widetilde{\boldsymbol{\varepsilon}}^{\mathrm{T}} G_y \widetilde{\boldsymbol{\varepsilon}}$
Covariance matrices	$G_{\widetilde{x}}^{-1} = (A^{\mathrm{T}} G_y A)^{-1}$	$G_{\widetilde{x}}^{-1} = \sigma^2(\widetilde{x})$ $= \left(\sum_k \frac{1}{\sigma_k^2}\right)^{-1}$
	$G_{\widetilde{\eta}}^{-1} = A(A^{\mathrm{T}} G_y A) A^{\mathrm{T}}$	–

Analysis of Variance

The influence of external variables on a measured random variable x is investigated. One tries to decide by means of appropriately constructed F-tests whether x is independent of the *external variables*. Various *models* are constructed depending on the number of external variables and the assumptions concerning their influence.

A simple model is the *crossed two-way classification* with multiple observations. Two external variables are used to classify the observations into the classes A_i, B_j ($i = 1, 2, \ldots, I$; $j = 1, 2, \ldots, J$). Each class A_i, B_j contains K observations x_{ijk} ($k = 1, 2, \ldots, K$). One assumes the model

$$x_{ijk} = \mu + a_i + b_j + (ab)_{ij} + \varepsilon_{ijk} \quad ,$$

where the error of the observation ε_{ijk} is assumed to be normally distributed about zero and a_i, b_j, and $(ab)_{ij}$ are the influences of the *classifications* in A, B and their *interactions*. Three *null hypotheses*

$$H_0^{(A)}(a_i = 0,\, i = 1, \ldots, I) \quad , \qquad H_0^{(B)}(b_j = 0,\, j = 1, \ldots, J) \quad ,$$

$$H_0^{(AB)}((ab)_{ij} = 0,\, i = 1, \ldots, I,\, j = 1, \ldots, J)$$

can be tested with the ratios $F^{(A)}$, $F^{(B)}$, $F^{(AB)}$. They are summarized in an analysis of variance (ANOVA) table. For other models see Chap. 11.

Polynomial Regression

Problem: The true values $\eta(t)$, for which one has N *measurements* $y_i(t_i)$ with normally distributed measurement errors σ_i, are to be described by a *polynomial* of order $r - 1$ in the *controlled variable* t. Instead of $\eta(t) = x_1 + x_2 t + \cdots + x_r t^{r-1}$ one writes

$$\eta(t) = x_1 f_1(t) + x_2 f_2(t) + \cdots + x_r f_r(t) \quad .$$

Here the f_j are *orthogonal polynomials* of order $j - 1$,

$$f_j(t) = \sum_{k=1}^{j} b_{jk} t^{k-1} \quad ,$$

whose coefficients b_{jk} are determined by the *orthogonality conditions*

$$\sum_{i=1}^{N} g_i f_j(t_i) f_k(t_i) = \delta_{jk} \quad , \qquad g_i = 1/\sigma_i^2 \quad .$$

The *unknowns* x_j are obtained by least squares from

$$\sum_{i=1}^{N} g_i \left\{ y_i(t_i) - \sum_{j=1}^{r} x_j f_j(t_i) \right\}^2 = \min \quad .$$

The covariance matrix of the x_j is the r-dimensional unit matrix.

Time Series Analysis

One is given a series of *measured values* $y_i(t_i)$, $i = 1, \ldots, n$, which (in an unknown way) depend on a *controlled variable* t (usually time). One treats the y_i as the sum of a *trend* η_i and an *error* ε_i, $y_i = \eta_i + \varepsilon_i$. The measurements are carried out at regular time intervals, i.e., $t_i - t_{i-1} = \text{const}$. In order to minimize the errors ε_i, a *moving average* is constructed for every t_i ($i > k$, $i \leq n - k$), by fitting a polynomial of order ℓ to the $2k + 1$ measurements situated symmetrically about measurement i. The result of the fit at the point t_i is the moving average

$$\widetilde{\eta}_0(i) = a_{-k} y_{i-k} + a_{-k+1} y_{i-k+1} + \cdots + a_k y_{i+k} \quad .$$

The coefficients $a_{-k}, \ldots, a_k$ are given in Table 13.1 for low values of k and ℓ. For the beginning and end points t_i ($i < k$, $i > n - k$), the results of the fit can be used with caution also for points other than at the center of the interval of the $2k + 1$ measurements.

I. Statistical Tables

Table I.1: Quantiles $\lambda_P(k)$ of the Poisson distribution.

$$P = \sum_{n=0}^{k-1} \mathrm{e}^{-\lambda_P} \lambda_P^n / n!$$

	P						
k	0.0005	0.0010	0.0050	0.0100	0.0250	0.0500	0.1000
1	7.601	6.908	5.298	4.605	3.689	2.996	2.303
2	9.999	9.233	7.430	6.638	5.572	4.744	3.890
3	12.051	11.229	9.274	8.406	7.225	6.296	5.322
4	13.934	13.062	10.977	10.045	8.767	7.754	6.681
5	15.710	14.794	12.594	11.605	10.242	9.154	7.994
6	17.411	16.455	14.150	13.108	11.668	10.513	9.275
7	19.055	18.062	15.660	14.571	13.059	11.842	10.532
8	20.654	19.626	17.134	16.000	14.423	13.148	11.771
9	22.217	21.156	18.578	17.403	15.763	14.435	12.995
10	23.749	22.657	19.998	18.783	17.085	15.705	14.206
11	25.256	24.134	21.398	20.145	18.390	16.962	15.407
12	26.739	25.589	22.779	21.490	19.682	18.208	16.598
13	28.203	27.026	24.145	22.821	20.962	19.443	17.782
14	29.650	28.446	25.497	24.139	22.230	20.669	18.958
15	31.081	29.852	26.836	25.446	23.490	21.886	20.128
16	32.498	31.244	28.164	26.743	24.740	23.097	21.292
17	33.902	32.624	29.482	28.030	25.983	24.301	22.452
18	35.294	33.993	30.791	29.310	27.219	25.499	23.606
19	36.676	35.351	32.091	30.581	28.448	26.692	24.756
20	38.047	36.701	33.383	31.845	29.671	27.879	25.903
21	39.410	38.042	34.668	33.103	30.888	29.062	27.045
22	40.764	39.375	35.946	34.355	32.101	30.240	28.184
23	42.110	40.700	37.218	35.601	33.308	31.415	29.320
24	43.449	42.019	38.484	36.841	34.511	32.585	30.453
25	44.780	43.330	39.745	38.077	35.710	33.752	31.584

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

Table I.1: (continued)

$$P = \sum_{n=0}^{k-1} \mathrm{e}^{-\lambda_P} \lambda_P^n / n!$$

	P						
k	0.9000	0.9500	0.9750	0.9900	0.9950	0.9990	0.9995
1	0.105	0.051	0.025	0.010	0.005	0.001	0.001
2	0.532	0.355	0.242	0.149	0.103	0.045	0.032
3	1.102	0.818	0.619	0.436	0.338	0.191	0.150
4	1.745	1.366	1.090	0.823	0.672	0.429	0.355
5	2.433	1.970	1.623	1.279	1.078	0.739	0.632
6	3.152	2.613	2.202	1.785	1.537	1.107	0.967
7	3.895	3.285	2.814	2.330	2.037	1.520	1.348
8	4.656	3.981	3.454	2.906	2.571	1.971	1.768
9	5.432	4.695	4.115	3.507	3.132	2.452	2.220
10	6.221	5.425	4.795	4.130	3.717	2.961	2.699
11	7.021	6.169	5.491	4.771	4.321	3.491	3.202
12	7.829	6.924	6.201	5.428	4.943	4.042	3.726
13	8.646	7.690	6.922	6.099	5.580	4.611	4.269
14	9.470	8.464	7.654	6.782	6.231	5.195	4.828
15	10.300	9.246	8.395	7.477	6.893	5.794	5.402
16	11.135	10.036	9.145	8.181	7.567	6.405	5.990
17	11.976	10.832	9.903	8.895	8.251	7.028	6.590
18	12.822	11.634	10.668	9.616	8.943	7.662	7.201
19	13.671	12.442	11.439	10.346	9.644	8.306	7.822
20	14.525	13.255	12.217	11.082	10.353	8.958	8.453
21	15.383	14.072	12.999	11.825	11.069	9.619	9.093
22	16.244	14.894	13.787	12.574	11.792	10.288	9.741
23	17.108	15.719	14.580	13.329	12.521	10.964	10.397
24	17.975	16.549	15.377	14.089	13.255	11.647	11.060
25	18.844	17.382	16.179	14.853	13.995	12.337	11.730

Table I.2: Normal distribution $\psi_0(x)$.

$$P(\mathsf{x} < x) = \psi_0(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-x^2/2)\,\mathrm{d}x$$

x	0	1	2	3	4	5	6	7	8	9
−3.0	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
−2.9	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.001	0.001	0.001
−2.8	0.003	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
−2.7	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003
−2.6	0.005	0.005	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.004
−2.5	0.006	0.006	0.006	0.006	0.006	0.005	0.005	0.005	0.005	0.005
−2.4	0.008	0.008	0.008	0.008	0.007	0.007	0.007	0.007	0.007	0.006
−2.3	0.011	0.010	0.010	0.010	0.010	0.009	0.009	0.009	0.009	0.008
−2.2	0.014	0.014	0.013	0.013	0.013	0.012	0.012	0.012	0.011	0.011
−2.1	0.018	0.017	0.017	0.017	0.016	0.016	0.015	0.015	0.015	0.014
−2.0	0.023	0.022	0.022	0.021	0.021	0.020	0.020	0.019	0.019	0.018
−1.9	0.029	0.028	0.027	0.027	0.026	0.026	0.025	0.024	0.024	0.023
−1.8	0.036	0.035	0.034	0.034	0.033	0.032	0.031	0.031	0.030	0.029
−1.7	0.045	0.044	0.043	0.042	0.041	0.040	0.039	0.038	0.038	0.037
−1.6	0.055	0.054	0.053	0.052	0.051	0.049	0.048	0.047	0.046	0.046
−1.5	0.067	0.066	0.064	0.063	0.062	0.061	0.059	0.058	0.057	0.056
−1.4	0.081	0.079	0.078	0.076	0.075	0.074	0.072	0.071	0.069	0.068
−1.3	0.097	0.095	0.093	0.092	0.090	0.089	0.087	0.085	0.084	0.082
−1.2	0.115	0.113	0.111	0.109	0.107	0.106	0.104	0.102	0.100	0.099
−1.1	0.136	0.133	0.131	0.129	0.127	0.125	0.123	0.121	0.119	0.117
−1.0	0.159	0.156	0.154	0.152	0.149	0.147	0.145	0.142	0.140	0.138
−0.9	0.184	0.181	0.179	0.176	0.174	0.171	0.169	0.166	0.164	0.161
−0.8	0.212	0.209	0.206	0.203	0.200	0.198	0.195	0.192	0.189	0.187
−0.7	0.242	0.239	0.236	0.233	0.230	0.227	0.224	0.221	0.218	0.215
−0.6	0.274	0.271	0.268	0.264	0.261	0.258	0.255	0.251	0.248	0.245
−0.5	0.309	0.305	0.302	0.298	0.295	0.291	0.288	0.284	0.281	0.278
−0.4	0.345	0.341	0.337	0.334	0.330	0.326	0.323	0.319	0.316	0.312
−0.3	0.382	0.378	0.374	0.371	0.367	0.363	0.359	0.356	0.352	0.348
−0.2	0.421	0.417	0.413	0.409	0.405	0.401	0.397	0.394	0.390	0.386
−0.1	0.460	0.456	0.452	0.448	0.444	0.440	0.436	0.433	0.429	0.425
0.0	0.500	0.496	0.492	0.488	0.484	0.480	0.476	0.472	0.468	0.464

Table I.2: (continued)

$$P(\mathsf{x} < x) = \psi_0(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-x^2/2)\,\mathrm{d}x$$

x	0	1	2	3	4	5	6	7	8	9
0.0	0.500	0.504	0.508	0.512	0.516	0.520	0.524	0.528	0.532	0.536
0.1	0.540	0.544	0.548	0.552	0.556	0.560	0.564	0.567	0.571	0.575
0.2	0.579	0.583	0.587	0.591	0.595	0.599	0.603	0.606	0.610	0.614
0.3	0.618	0.622	0.626	0.629	0.633	0.637	0.641	0.644	0.648	0.652
0.4	0.655	0.659	0.663	0.666	0.670	0.674	0.677	0.681	0.684	0.688
0.5	0.691	0.695	0.698	0.702	0.705	0.709	0.712	0.716	0.719	0.722
0.6	0.726	0.729	0.732	0.736	0.739	0.742	0.745	0.749	0.752	0.755
0.7	0.758	0.761	0.764	0.767	0.770	0.773	0.776	0.779	0.782	0.785
0.8	0.788	0.791	0.794	0.797	0.800	0.802	0.805	0.808	0.811	0.813
0.9	0.816	0.819	0.821	0.824	0.826	0.829	0.831	0.834	0.836	0.839
1.0	0.841	0.844	0.846	0.848	0.851	0.853	0.855	0.858	0.860	0.862
1.1	0.864	0.867	0.869	0.871	0.873	0.875	0.877	0.879	0.881	0.883
1.2	0.885	0.887	0.889	0.891	0.893	0.894	0.896	0.898	0.900	0.901
1.3	0.903	0.905	0.907	0.908	0.910	0.911	0.913	0.915	0.916	0.918
1.4	0.919	0.921	0.922	0.924	0.925	0.926	0.928	0.929	0.931	0.932
1.5	0.933	0.934	0.936	0.937	0.938	0.939	0.941	0.942	0.943	0.944
1.6	0.945	0.946	0.947	0.948	0.949	0.951	0.952	0.953	0.954	0.954
1.7	0.955	0.956	0.957	0.958	0.959	0.960	0.961	0.962	0.962	0.963
1.8	0.964	0.965	0.966	0.966	0.967	0.968	0.969	0.969	0.970	0.971
1.9	0.971	0.972	0.973	0.973	0.974	0.974	0.975	0.976	0.976	0.977
2.0	0.977	0.978	0.978	0.979	0.979	0.980	0.980	0.981	0.981	0.982
2.1	0.982	0.983	0.983	0.983	0.984	0.984	0.985	0.985	0.985	0.986
2.2	0.986	0.986	0.987	0.987	0.987	0.988	0.988	0.988	0.989	0.989
2.3	0.989	0.990	0.990	0.990	0.990	0.991	0.991	0.991	0.991	0.992
2.4	0.992	0.992	0.992	0.992	0.993	0.993	0.993	0.993	0.993	0.994
2.5	0.994	0.994	0.994	0.994	0.994	0.995	0.995	0.995	0.995	0.995
2.6	0.995	0.995	0.996	0.996	0.996	0.996	0.996	0.996	0.996	0.996
2.7	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997	0.997
2.8	0.997	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998
2.9	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.999	0.999	0.999
3.0	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999

Table I.3: Normal distribution $2\psi_0(x) - 1$.

$$P(|\mathsf{x}| < x) = 2\psi_0(x) - 1 = \frac{1}{\sqrt{2\pi}} \int_{-x}^{x} \exp(-x^2/2)\,\mathrm{d}x$$

x	0	1	2	3	4	5	6	7	8	9
0.0	0.000	0.008	0.016	0.024	0.032	0.040	0.048	0.056	0.064	0.072
0.1	0.080	0.088	0.096	0.103	0.111	0.119	0.127	0.135	0.143	0.151
0.2	0.159	0.166	0.174	0.182	0.190	0.197	0.205	0.213	0.221	0.228
0.3	0.236	0.243	0.251	0.259	0.266	0.274	0.281	0.289	0.296	0.303
0.4	0.311	0.318	0.326	0.333	0.340	0.347	0.354	0.362	0.369	0.376
0.5	0.383	0.390	0.397	0.404	0.411	0.418	0.425	0.431	0.438	0.445
0.6	0.451	0.458	0.465	0.471	0.478	0.484	0.491	0.497	0.503	0.510
0.7	0.516	0.522	0.528	0.535	0.541	0.547	0.553	0.559	0.565	0.570
0.8	0.576	0.582	0.588	0.593	0.599	0.605	0.610	0.616	0.621	0.627
0.9	0.632	0.637	0.642	0.648	0.653	0.658	0.663	0.668	0.673	0.678
1.0	0.683	0.688	0.692	0.697	0.702	0.706	0.711	0.715	0.720	0.724
1.1	0.729	0.733	0.737	0.742	0.746	0.750	0.754	0.758	0.762	0.766
1.2	0.770	0.774	0.778	0.781	0.785	0.789	0.792	0.796	0.799	0.803
1.3	0.806	0.810	0.813	0.816	0.820	0.823	0.826	0.829	0.832	0.835
1.4	0.838	0.841	0.844	0.847	0.850	0.853	0.856	0.858	0.861	0.864
1.5	0.866	0.869	0.871	0.874	0.876	0.879	0.881	0.884	0.886	0.888
1.6	0.890	0.893	0.895	0.897	0.899	0.901	0.903	0.905	0.907	0.909
1.7	0.911	0.913	0.915	0.916	0.918	0.920	0.922	0.923	0.925	0.927
1.8	0.928	0.930	0.931	0.933	0.934	0.936	0.937	0.939	0.940	0.941
1.9	0.943	0.944	0.945	0.946	0.948	0.949	0.950	0.951	0.952	0.953
2.0	0.954	0.956	0.957	0.958	0.959	0.960	0.961	0.962	0.962	0.963
2.1	0.964	0.965	0.966	0.967	0.968	0.968	0.969	0.970	0.971	0.971
2.2	0.972	0.973	0.974	0.974	0.975	0.976	0.976	0.977	0.977	0.978
2.3	0.979	0.979	0.980	0.980	0.981	0.981	0.982	0.982	0.983	0.983
2.4	0.984	0.984	0.984	0.985	0.985	0.986	0.986	0.986	0.987	0.987
2.5	0.988	0.988	0.988	0.989	0.989	0.989	0.990	0.990	0.990	0.990
2.6	0.991	0.991	0.991	0.991	0.992	0.992	0.992	0.992	0.993	0.993
2.7	0.993	0.993	0.993	0.994	0.994	0.994	0.994	0.994	0.995	0.995
2.8	0.995	0.995	0.995	0.995	0.995	0.996	0.996	0.996	0.996	0.996
2.9	0.996	0.996	0.996	0.997	0.997	0.997	0.997	0.997	0.997	0.997
3.0	0.997	0.997	0.997	0.998	0.998	0.998	0.998	0.998	0.998	0.998

Table I.4: Quantiles $x_P = \Omega(P)$ of the normal distribution.

$$P = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_P} \exp(-x^2/2)\,\mathrm{d}x$$

P	0	1	2	3	4	5	6	7	8	9
0.0	$-\infty$	−2.33	−2.05	−1.88	−1.75	−1.64	−1.55	−1.48	−1.41	−1.34
0.1	−1.28	−1.23	−1.17	−1.13	−1.08	−1.04	−0.99	−0.95	−0.92	−0.88
0.2	−0.84	−0.81	−0.77	−0.74	−0.71	−0.67	−0.64	−0.61	−0.58	−0.55
0.3	−0.52	−0.50	−0.47	−0.44	−0.41	−0.39	−0.36	−0.33	−0.31	−0.28
0.4	−0.25	−0.23	−0.20	−0.18	−0.15	−0.13	−0.10	−0.08	−0.05	−0.03
0.5	0.00	0.03	0.05	0.08	0.10	0.13	0.15	0.18	0.20	0.23
0.6	0.25	0.28	0.31	0.33	0.36	0.39	0.41	0.44	0.47	0.50
0.7	0.52	0.55	0.58	0.61	0.64	0.67	0.71	0.74	0.77	0.81
0.8	0.84	0.88	0.92	0.95	0.99	1.04	1.08	1.13	1.17	1.23
0.9	1.28	1.34	1.41	1.48	1.55	1.64	1.75	1.88	2.05	2.33

Table I.5: Quantiles $x'_P = \Omega'(P)$ of the normal distribution.

$$P = \frac{1}{\sqrt{2\pi}} \int_{-x'_P}^{x'_P} \exp(-x^2/2)\,\mathrm{d}x$$

P	0	1	2	3	4	5	6	7	8	9
0.0	0.000	0.013	0.025	0.038	0.050	0.063	0.075	0.088	0.100	0.113
0.1	0.126	0.138	0.151	0.164	0.176	0.189	0.202	0.215	0.228	0.240
0.2	0.253	0.266	0.279	0.292	0.305	0.319	0.332	0.345	0.358	0.372
0.3	0.385	0.399	0.412	0.426	0.440	0.454	0.468	0.482	0.496	0.510
0.4	0.524	0.539	0.553	0.568	0.583	0.598	0.613	0.628	0.643	0.659
0.5	0.674	0.690	0.706	0.722	0.739	0.755	0.772	0.789	0.806	0.824
0.6	0.842	0.860	0.878	0.896	0.915	0.935	0.954	0.974	0.994	1.015
0.7	1.036	1.058	1.080	1.103	1.126	1.150	1.175	1.200	1.227	1.254
0.8	1.282	1.311	1.341	1.372	1.405	1.440	1.476	1.514	1.555	1.598
0.9	1.645	1.695	1.751	1.812	1.881	1.960	2.054	2.170	2.326	2.576

P	0	1	2	3	4	5	6	7	8	9
0.90	1.645	1.650	1.655	1.660	1.665	1.670	1.675	1.680	1.685	1.690
0.91	1.695	1.701	1.706	1.711	1.717	1.722	1.728	1.734	1.739	1.745
0.92	1.751	1.757	1.762	1.768	1.774	1.780	1.787	1.793	1.799	1.805
0.93	1.812	1.818	1.825	1.832	1.838	1.845	1.852	1.859	1.866	1.873
0.94	1.881	1.888	1.896	1.903	1.911	1.919	1.927	1.935	1.943	1.951
0.95	1.960	1.969	1.977	1.986	1.995	2.005	2.014	2.024	2.034	2.044
0.96	2.054	2.064	2.075	2.086	2.097	2.108	2.120	2.132	2.144	2.157
0.97	2.170	2.183	2.197	2.212	2.226	2.241	2.257	2.273	2.290	2.308
0.98	2.326	2.346	2.366	2.387	2.409	2.432	2.457	2.484	2.512	2.543
0.99	2.576	2.612	2.652	2.697	2.748	2.807	2.878	2.968	3.090	3.291

P	0	1	2	3	4	5	6	7	8	9
0.990	2.576	2.579	2.583	2.586	2.590	2.594	2.597	2.601	2.605	2.608
0.991	2.612	2.616	2.620	2.624	2.628	2.632	2.636	2.640	2.644	2.648
0.992	2.652	2.656	2.661	2.665	2.669	2.674	2.678	2.683	2.687	2.692
0.993	2.697	2.702	2.706	2.711	2.716	2.721	2.727	2.732	2.737	2.742
0.994	2.748	2.753	2.759	2.765	2.770	2.776	2.782	2.788	2.794	2.801
0.995	2.807	2.814	2.820	2.827	2.834	2.841	2.848	2.855	2.863	2.870
0.996	2.878	2.886	2.894	2.903	2.911	2.920	2.929	2.938	2.948	2.958
0.997	2.968	2.978	2.989	3.000	3.011	3.023	3.036	3.048	3.062	3.076
0.998	3.090	3.105	3.121	3.138	3.156	3.175	3.195	3.216	3.239	3.264
0.999	3.291	3.320	3.353	3.390	3.432	3.481	3.540	3.615	3.719	3.891

Table I.6: χ^2-distribution $F(\chi^2)$.

$$F(\chi^2) = \int_0^{\chi^2} f(\chi^2; f)\,\mathrm{d}\chi^2$$

	f									
χ^2	1	2	3	4	5	6	7	8	9	10
0.1	0.248	0.049	0.008	0.001	0.000	0.000	0.000	0.000	0.000	0.000
0.2	0.345	0.095	0.022	0.005	0.001	0.000	0.000	0.000	0.000	0.000
0.3	0.416	0.139	0.040	0.010	0.002	0.001	0.000	0.000	0.000	0.000
0.4	0.473	0.181	0.060	0.018	0.005	0.001	0.000	0.000	0.000	0.000
0.5	0.520	0.221	0.081	0.026	0.008	0.002	0.001	0.000	0.000	0.000
0.6	0.561	0.259	0.104	0.037	0.012	0.004	0.001	0.000	0.000	0.000
0.7	0.597	0.295	0.127	0.049	0.017	0.006	0.002	0.000	0.000	0.000
0.8	0.629	0.330	0.151	0.062	0.023	0.008	0.003	0.001	0.000	0.000
0.9	0.657	0.362	0.175	0.075	0.030	0.011	0.004	0.001	0.000	0.000
1.0	0.683	0.393	0.199	0.090	0.037	0.014	0.005	0.002	0.001	0.000
2.0	0.843	0.632	0.428	0.264	0.151	0.080	0.040	0.019	0.009	0.004
3.0	0.917	0.777	0.608	0.442	0.300	0.191	0.115	0.066	0.036	0.019
4.0	0.954	0.865	0.739	0.594	0.451	0.323	0.220	0.143	0.089	0.053
5.0	0.975	0.918	0.828	0.713	0.584	0.456	0.340	0.242	0.166	0.109
6.0	0.986	0.950	0.888	0.801	0.694	0.577	0.460	0.353	0.260	0.185
7.0	0.992	0.970	0.928	0.864	0.779	0.679	0.571	0.463	0.363	0.275
8.0	0.995	0.982	0.954	0.908	0.844	0.762	0.667	0.567	0.466	0.371
9.0	0.997	0.989	0.971	0.939	0.891	0.826	0.747	0.658	0.563	0.468
10.0	0.998	0.993	0.981	0.960	0.925	0.875	0.811	0.735	0.650	0.560
11.0	0.999	0.996	0.988	0.973	0.949	0.912	0.861	0.798	0.724	0.642
12.0	0.999	0.998	0.993	0.983	0.965	0.938	0.899	0.849	0.787	0.715
13.0	1.000	0.998	0.995	0.989	0.977	0.957	0.928	0.888	0.837	0.776
14.0	1.000	0.999	0.997	0.993	0.984	0.970	0.949	0.918	0.878	0.827
15.0	1.000	0.999	0.998	0.995	0.990	0.980	0.964	0.941	0.909	0.868
16.0	1.000	1.000	0.999	0.997	0.993	0.986	0.975	0.958	0.933	0.900
17.0	1.000	1.000	0.999	0.998	0.996	0.991	0.983	0.970	0.951	0.926
18.0	1.000	1.000	1.000	0.999	0.997	0.994	0.988	0.979	0.965	0.945
19.0	1.000	1.000	1.000	0.999	0.998	0.996	0.992	0.985	0.975	0.960
20.0	1.000	1.000	1.000	1.000	0.999	0.997	0.994	0.990	0.982	0.971

Table I.7: Quantiles χ^2_P of the χ^2-distribution.

$$P = \int_0^{\chi^2_P} f(\chi^2; f)\,\mathrm{d}\chi^2$$

	P				
f	0.900	0.950	0.990	0.995	0.999
1	2.706	3.841	6.635	7.879	10.828
2	4.605	5.991	9.210	10.597	13.816
3	6.251	7.815	11.345	12.838	16.266
4	7.779	9.488	13.277	14.860	18.467
5	9.236	11.070	15.086	16.750	20.515
6	10.645	12.592	16.812	18.548	22.458
7	12.017	14.067	18.475	20.278	24.322
8	13.362	15.507	20.090	21.955	26.124
9	14.684	16.919	21.666	23.589	27.877
10	15.987	18.307	23.209	25.188	29.588
11	17.275	19.675	24.725	26.757	31.264
12	18.549	21.026	26.217	28.300	32.909
13	19.812	22.362	27.688	29.819	34.528
14	21.064	23.685	29.141	31.319	36.123
15	22.307	24.996	30.578	32.801	37.697
16	23.542	26.296	32.000	34.267	39.252
17	24.769	27.587	33.409	35.718	40.790
18	25.989	28.869	34.805	37.156	42.312
19	27.204	30.144	36.191	38.582	43.820
20	28.412	31.410	37.566	39.997	45.315
30	40.256	43.773	50.892	53.672	59.703
40	51.805	55.758	63.691	66.766	73.402
50	63.167	67.505	76.154	79.490	86.661
60	74.397	79.082	88.379	91.952	99.607
70	85.527	90.531	100.425	104.215	112.317
80	80.000	101.879	112.329	116.321	124.839
90	107.565	113.145	124.116	128.299	137.208
100	118.498	124.342	135.807	140.169	149.449

Table I.8: Quantiles F_P of the F-distribution.

$$0.900 = P = \int_0^{F_P} f(F; f_1, f_2)\,\mathrm{d}F$$

	f_1									
f_2	1	2	3	4	5	6	7	8	9	10
1	39.86	49.50	53.59	55.83	57.24	58.20	58.91	59.44	59.86	60.19
2	8.526	9.000	9.162	9.243	9.293	9.326	9.349	9.367	9.381	9.392
3	5.538	5.462	5.391	5.343	5.309	5.285	5.266	5.252	5.240	5.230
4	4.545	4.325	4.191	4.107	4.051	4.010	3.979	3.955	3.936	3.920
5	4.060	3.780	3.619	3.520	3.453	3.405	3.368	3.339	3.316	3.297
6	3.776	3.463	3.289	3.181	3.108	3.055	3.014	2.983	2.958	2.937
7	3.589	3.257	3.074	2.961	2.883	2.827	2.785	2.752	2.725	2.703
8	3.458	3.113	2.924	2.806	2.726	2.668	2.624	2.589	2.561	2.538
9	3.360	3.006	2.813	2.693	2.611	2.551	2.505	2.469	2.440	2.416
10	3.285	2.924	2.728	2.605	2.522	2.461	2.414	2.377	2.347	2.323

$$0.950 = P = \int_0^{F_P} f(F; f_1, f_2)\,\mathrm{d}F$$

	f_1									
f_2	1	2	3	4	5	6	7	8	9	10
1	161.4	199.5	215.7	224.6	230.2	234.0	236.8	238.9	240.5	241.9
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40
3	10.13	9.552	9.277	9.117	9.013	8.941	8.887	8.845	8.812	8.786
4	7.709	6.944	6.591	6.388	6.256	6.163	6.094	6.041	5.999	5.964
5	6.608	5.786	5.409	5.192	5.050	4.950	4.876	4.818	4.772	4.735
6	5.987	5.143	4.757	4.534	4.387	4.284	4.207	4.147	4.099	4.060
7	5.591	4.737	4.347	4.120	3.972	3.866	3.787	3.726	3.677	3.637
8	5.318	4.459	4.066	3.838	3.687	3.581	3.500	3.438	3.388	3.347
9	5.117	4.256	3.863	3.633	3.482	3.374	3.293	3.230	3.179	3.137
10	4.965	4.103	3.708	3.478	3.326	3.217	3.135	3.072	3.020	2.978

Table I.8: (continued)

$$0.975 = P = \int_0^{F_P} f(F; f_1, f_2)\,\mathrm{d}F$$

f_2 \ f_1	1	2	3	4	5	6	7	8	9	10
1	647.8	799.5	864.2	899.6	921.8	937.1	948.2	956.7	963.3	968.6
2	38.51	39.00	39.17	39.25	39.30	39.33	39.36	39.37	39.39	39.40
3	17.44	16.04	15.44	15.10	14.88	14.73	14.62	14.54	14.47	14.42
4	12.22	10.65	9.979	9.605	9.364	9.197	9.074	8.980	8.905	8.844
5	10.01	8.434	7.764	7.388	7.146	6.978	6.853	6.757	6.681	6.619
6	8.813	7.260	6.599	6.227	5.988	5.820	5.695	5.600	5.523	5.461
7	8.073	6.542	5.890	5.523	5.285	5.119	4.995	4.899	4.823	4.761
8	7.571	6.059	5.416	5.053	4.817	4.652	4.529	4.433	4.357	4.295
9	7.209	5.715	5.078	4.718	4.484	4.320	4.197	4.102	4.026	3.964
10	6.937	5.456	4.826	4.468	4.236	4.072	3.950	3.855	3.779	3.717

$$0.990 = P = \int_0^{F_P} f(F; f_1, f_2)\,\mathrm{d}F$$

f_2 \ f_1	1	2	3	4	5	6	7	8	9	10
1	4052	5000	5403	5625	5764	5859	5928	5981	6022	6056
2	98.50	99.00	99.17	99.25	99.30	99.33	99.36	99.37	99.39	99.40
3	34.12	30.82	29.46	28.71	28.24	27.91	27.67	27.49	27.35	27.23
4	21.20	18.00	16.69	15.98	15.52	15.21	14.98	14.80	14.66	14.55
5	16.26	13.27	12.06	11.39	10.97	10.67	10.46	10.29	10.16	10.05
6	13.75	10.92	9.780	9.148	8.746	8.466	8.260	8.102	7.976	7.874
7	12.25	9.547	8.451	7.847	7.460	7.191	6.993	6.840	6.719	6.620
8	11.26	8.649	7.591	7.006	6.632	6.371	6.178	6.029	5.911	5.814
9	10.56	8.022	6.992	6.422	6.057	5.802	5.613	5.467	5.351	5.257
10	10.04	7.559	6.552	5.994	5.636	5.386	5.200	5.057	4.942	4.849

Table I.9: Quantiles t_P of Student's distribution.

$$P = \int_{-\infty}^{t_P} f(t; f)\,\mathrm{d}t$$

	P						
f	0.9000	0.9500	0.9750	0.9900	0.9950	0.9990	0.9995
1	3.078	6.314	12.706	31.821	63.657	318.309	636.619
2	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	1.325	1.725	2.086	2.528	2.845	3.552	3.850
30	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	1.303	1.684	2.021	2.423	2.704	3.307	3.551
50	1.299	1.676	2.009	2.403	2.678	3.261	3.496
60	1.296	1.671	2.000	2.390	2.660	3.232	3.460
70	1.294	1.667	1.994	2.381	2.648	3.211	3.435
80	1.292	1.664	1.990	2.374	2.639	3.195	3.416
90	1.291	1.662	1.987	2.368	2.632	3.183	3.402
100	1.290	1.660	1.984	2.364	2.626	3.174	3.390
200	1.286	1.653	1.972	2.345	2.601	3.131	3.340
500	1.283	1.648	1.965	2.334	2.586	3.107	3.310
1000	1.282	1.646	1.962	2.330	2.581	3.098	3.300

List of Computer Programs*

*The slanted numbers refer to the Appendix.

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*, DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014

Index*

*The slanted numbers refer to the Appendix.

S. Brandt, *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*,
DOI 10.1007/978-3-319-03762-2, © Springer International Publishing Switzerland 2014