Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

The CWI is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O).

CWI Syllabus 20

STATAL: statistical procedures in Algol 60, part 1

R. van der Horst, R.D. Gill (eds.)

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

ISBN 90 6196 358 3

NUGI-code: 815

STATAL

STATISTICAL PROCEDURES IN ALGOL 60

The library

STATAL

of programming in mathematical statistics

INTRODUCTION

STATAL is a library of statistical procedures, written in ALGOL-60, for use at the CDC CYBER 70 system of the Stichting Academisch Rekencentrum Amsterdam (SARA). At present (in the year 1988) STATAL works on the CYBER 170-750 (old) or the CYBER 180-990 (new) models under the NOS/BE operating system. The library was developed at the department of Mathematical Statistics of the Mathematisch Centrum at Amsterdam. The documentation first appeared in 1974 as a loose leaf manual. It was supplemented, revised and updated in 1976, 1977, 1978 and 1981. Further revisions lead to the present final form of the manual.

STATAL departs from other statistical libraries in that it does not contain complete *programs* for standard statistical techniques, but rather a quite complete collection of *procedures* for elementary, generally not simple statistical computations (e.g. computation of distribution functions, inverse distribution functions, random samples from distributions, tables, pictures, statistics). Using the STATAL-procedures it is easy for an ALGOL-programmer to write programs for various statistical problems. So, applications of STATAL will mainly be in non-standard consultation and in research (e.g. simulations).

The choice of the ALGOL language grew out of a tradition at the Mathematical Centrum. Furthermore, this language still describes the structure of the procedures better than many other programming languages currently in use. Thus we expect that the source texts of the procedures will be useful for programmers writing similar procedures in other languages.

STATAL is organised as follows. Each procedure of the library is identified by a name and a code number. The code number can be used in an ALGOL-60 program when reference is made to a pre-compiled procedure in the object code library. All procedures in STATAL are classified according to. These subjects are identified by a section number. This manual is ordered according to these section numbers. In order to find a particular procedure, there is a systematic index, in which all procedures are ordered by their section number. For cross referencing their is an index by code number, which has references to procedure name and section number.

On the SARA CDC CYBER models mentioned above, the object code of the procedures is available on disk; they are contained in the library file STATAL under ID=MATCEN, SN=M. This library file can be used when programs compiled under ALGOL 5 are loaded. Notice that the program must contain external references of all STATAL procedures used. (Such external references must contain the complete heading of the procedure with the correct parameter specifications and the statement "CODE" codenr.;). Compilation and execution of the program can be performed as follows:

```
ATTACH,A60.
ATTACH,ALG5LIB.

A60,I=program.
ATTACH,STATAL,ID=MATCEN,SN=M.
LIBRARY,STATAL.
LGO [, options].
```

Alternatively, one can also use the circumlude REFS with external references of all STATAL procedures. The program then need not contain external references. *By the way, this option is used in all examples of use.* Compilation and execution can be done by:

```
EFL,60000, or RFL,60000.
ATTACH,STATAL,ID=MATCEN,SN=M.
A60,I=program,S=STATAL-REFS.
LGO [, options].
```

A very few procedures use the library of numerical routines NUMAL or the library of graphical routines CALCOMP. For such procedures, these libraries have to be attached also. Documentation of these libraries are Hemker (1980) and SARA publicatie 11.

The procedures in STATAL are written in standard ALGOL-60 (that is according to Naur, 1964). However, in some STATAL procedures the use of input and output procedures, which are not defined in standard ALGOL-60, could not be avoided. So, most STATAL procedures are in principle independent of the computer and the compiler used. In any case they can be adapted by minor changes.

A part of the STATAL library has been translated into FORTRAN. This FORTRAN version of STATAL is called STAR and is also available on the SARA CDC CYBER models.

REFERENCES

[1] Hemker, P.W. (1980). *NUMAL numerical procedures in ALGOL-60.* MC Syllabus 47.1, 47.2,...,47.7. Mathematisch Centrum, Amsterdam.

[2] Naur, P. (ed) (1964). *Revised report on the algorithmic language ALGOL-60.* A/S Regnecentralen, Copenhagen.

[3] *Graphics,* SARA publicatie 11, Stichting Academisch Rekencentrum Amsterdam.

ACKNOWLEDGEMENTS

Most of the procedures and programs contained in STATAL were developed at the Mathematisch Centrum at Amsterdam. A few procedures, however, are adapted versions of procedures published elsewhere (cf. sections concerned). We are grateful to the following contributors and authors of procedures in STATAL:

J.G. Bethlehem
J.M. Buhrman
H. Elffers
A.J. van Es
M. van Gelderen
R.D. Gill
R. Kaas
E. Opperdoes
F.J.A. Overweel
C. van Putten
B.F. Schriever
I. van der Tweel
C.J. Warmer
R. Wiggers
D.T. Winter
A. Wolowitsj
A.C. IJsselstein

Furthermore, we want to thank Eline Meys and Mini Middelberg who took care of the typesetting of the documentation and Jan van der Steen for invaluable technical support.
J.G. Bethlehem was responsible for the original set-up in which prof. Hemelrijk played an inspiring role. B.F. Schriever was responsible for the final form. M.M. Voors and R. van der Horst prepared the source text of the procedures and ran the examples of use.

R. van der Horst (general editor)

SYSTEMATICAL INDEX

This index contains the names and code numbers of the procedures classified according to subject. (section number).

INDEX BY CODE NUMBER

1. DISTRIBUTIONS

1.1 DISCRETE DISTRIBUTIONS

This section contains procedures for computing the distribution function, the inverse distribution function, and the probability function of discrete distributions. The distribution functions are defined for all real values X of the argument. If the argument is smaller (larger) than the minimum (maximum) of the range, the value 0 (1) is assigned to the procedure identifier. If the argument X is not an integer but is within the range, the value of the distribution function with argument X is equal to the value with argument ENTIER(X). When the argument of a (discrete) probability function is not contained in the support of the distribution, the value 0 is assigned to the procedure identifier. The parameter list of the inverse distribution function contains a Boolean LEFT to indicate if either the left hand tail inverse or the right hand tail inverse has to be computed. The left hand tail inverse of the value PROB is defined as the maximum of all values Y of the support such that P(X≤Y)≤PROB, where X is a random variable with the distribution considered. The right hand tail inverse of the value PROB is defined as the minimum of all values Y of the support such that P(X≥Y)≤PROB. When it is possible to compute the inverse distribution function, a value larger than the maximum or smaller than the minimum of the support is assigned to the procedure identifier. (This value is maximum +1 or minimum -1 for most distributions, and is maximum +2 or minimum -2 for WILCOXINV and KENDALLINV).

We aimed for a precision of 10^{-10} in the computation of the procedures. In some cases it was not possible to obtain such a precision. The procedures which are computed 'exactly' have a precision equal to the machine precision 10^{-14}. Most procedures that approximate a function instead of computing it 'exactly' use the procedures INCOMPLETE BETA, LOGGAMMA or PHI.

The computation of the procedures WILCOX, WILCOXINV, WILCOXPROB, KENDALL, KENDALLINV and KENDALLPROB takes a considerable amount of time, so it is advised to use these procedures only with small samples, such as the ones in the examples of use.

TITLE: **Bin**

AUTHOR: J.H. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 740101

BRIEF DESCRIPTION

The procedure computes the binomial distribution function, i.e. the probability that the number of successes in N independent experiments is less than or equal to a given value X. Each of the experiments performed has the same, fixed probability P of succes.

KEYWORDS

Binomial distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" BIN (X, N, P);
"VALUE" X, N, P;
"REAL" X, N, P;
"CODE" 41000;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

N: <arithmetic expression>, number of experiments;

P: <arithmetic expression>, probability of success in a single experiment.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier BIN.

The following error messages may appear:

Errornumber 2 (if N is not an integer $\geqslant 0$)

Errornumber 3 (if $P < 0$ or $P > 1$)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

INCOMPLETE BETA STATAL 40401

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed exactly for N ⩽ 1000, and is approximated for N > 1000 by:

1 − INCOMPLETE BETA (P,ENTIER(X + 1),N − ENTIER(X), EPSILON),

where EPSILON, the precision of the incomplete BETA function, equals 10^{-12}.

The precision of the computation is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        BIN( 24, 100, .18),
        BIN(200, 400, .48),
        BIN(215, 700, .36))
"END"
```

Output:

```
.950439
.802564
.001841
```

SOURCE TEXT

```
"CODE" 41000;
"REAL" "PROCEDURE" BIN(X, N, P);
"VALUE" X, N, P; "REAL" X, N, P;
"BEGIN" "INTEGER" IX;


    IX:= ENTIER(X);
    "IF" N < 0 "OR" N > ENTIER(N) "THEN"
    BIN:= STATAL3 ERROR("("BIN")", 2, N) "ELSE"
    "IF" P < 0 "OR" P > 1 "THEN"
    BIN:= STATAL3 ERROR("("BIN")", 3, P) "ELSE"
    "IF" IX < 0 "THEN" BIN:= 0 "ELSE"
    "IF" IX >= N "THEN" BIN:= 1 "ELSE"
    "IF" P = 0 "THEN" BIN:= 1 "ELSE"
    "IF" P = 1 "THEN" BIN:= 0 "ELSE"
    "IF" N > 1000 "THEN"
    "BEGIN" "REAL" B;
        B:= 1 - INCOMPLETE BETA(P, IX + 1, N - IX, "-12);
        BIN:= "IF" B < 0 "THEN" 0 "ELSE" B;
    "END" "ELSE"
    "BEGIN" "REAL" "PROCEDURE" TAIL;
        "BEGIN" "INTEGER" I; "REAL" PROB, CUM, LAST;
            PROB:= CUM:= BINPROB(IX, N, P);
            "FOR" I:= IX - 1, I - 1 "WHILE" CUM > LAST "DO"
            "BEGIN" PROB:=
```

```
                PROB * (1 - P) / P * (I + 1) / (N - I);
                LAST:= CUM; CUM:= CUM + PROB
            "END";
            TAIL:= CUM
        "END";

        "IF" X > ENTIER(N / 2) "THEN"
        "BEGIN" IX:= N - IX - 1; P:= 1 - P;
            BIN:= 1 - TAIL
        "END" "ELSE" BIN:= TAIL
    "END";
"END" BIN;
         "EOP"
```

TITLE: **Bininv**

AUTHOR: J.H. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750401

BRIEF DESCRIPTION

The procedure computes the left (right) hand tail inverse of the binomial distribution, i.e. the largest (smallest) integer for which the left (right) hand tail probability of the distribution is less than or equal to a given value PROB. Each of the N experiments performed has the same, fixed probability P of succes.

KEYWORDS

Inverse binomial distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" BININV (PROB, N, P, LEFT);
"VALUE" PROB, N, P, LEFT;
"REAL" PROB, N, P;
"BOOLEAN" LEFT;
"CODE" 41001;
```

Formal parameters

PROB: <arithmetic expression>, tail probability of the value to be computed;

N: <arithmetic expression>, number of experiments;

P: <arithmetic expression>, probability of success in a single experiment;

LEFT: <Boolean expression>, Indicating if either the left hand tail inverse or the right hand tail inverse has to be computed. In the first (second) case left should have the value "TRUE" ("FALSE").

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier BININV. The value -1 ($N + 1$) is assigned if the probability of 0 (N) is larger than PROB.

The following error messages may appear:

Errornumber 1 (if PROB $\leqslant 0$ or PROB $\geqslant 1$)

Errornumber 2 (if N is not an integer $\geqslant 0$)

Errornumber 3 (if $P < 0$ or $P > 1$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
BIN	STATAL 41000
BINPROB	STATAL 41251
PHINV	STATAL 41501

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

In both cases the inverse distribution function is computed using the recurrent relation between successive binomial probabilities, starting with an estimated inverse obtained from a normal approximation.

The precision of the comparisons made is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    "BOOLEAN" LEFT;
    LEFT:= "TRUE";
    OUTPUT(61, "("6(+ZD,/)")",
        BININV(.25, 10, .4, LEFT),
        BININV(.52, 12, .6, LEFT),
        BININV(.01,  3, .5, LEFT),
        BININV(.74, 11, .7, "NOT" LEFT),
        BININV(.61,  8, .3, "NOT" LEFT),
        BININV(.02, 10, .5, "NOT" LEFT))
"END"
```

Output:

```
+2
+6
-1
+8
+3
+9
```

Source text

```
"CODE" 41001;
"REAL" "PROCEDURE" BININV(PROB, N, P, LEFT);
"VALUE" PROB, N, P, LEFT;
"REAL" PROB, N, P; "BOOLEAN" LEFT;
"BEGIN" "INTEGER" X; "REAL" PX, PCUM;

    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3 ERROR("("BININV")", 1, PROB) "ELSE"
    "IF" N > ENTIER(N) "OR" N < 0 "THEN"
    STATAL3 ERROR("("BININV")", 2, N) "ELSE"
    "IF" P < 0 "OR" P > 1 "THEN"
    STATAL3 ERROR("("BININV")", 3, P);

    "IF" P = 0 "OR" N = 0 "THEN"
        BININV:= ("IF" LEFT "THEN" -1 "ELSE" 1)
    "ELSE" "IF" P = 1 "THEN"
        BININV:= ("IF" LEFT "THEN" N - 1 "ELSE" N + 1)
    "ELSE" "IF" LEFT "THEN"
    "BEGIN" X:= PHINV(PROB) *
        SQRT(N * P * (1 - P)) - 0.5 + N * P;
        "IF" X < 0 "THEN" X:= 0
        "ELSE" "IF" X > N "THEN" X:= N;
        "IF" PROB < (1 - P) ** N "THEN" BININV:= -1 "ELSE"
        "BEGIN" PX:= BINPROB(X, N, P); PCUM:= BIN(X, N, P);
            "IF" PCUM > PROB "THEN"
            "BEGIN" "FOR" PCUM:= PCUM - PX
                "WHILE" PCUM > PROB "DO"
                "BEGIN" PX:= PX * X * (1 - P) /
                    (N - X + 1) / P;
                    X:= X - 1
                "END"; X:= X - 1
            "END" "ELSE"
            "BEGIN" "FOR" PX:=
                PX * (N - X) / (X + 1) * P / (1 - P)
                "WHILE" PCUM + PX < PROB "DO"
                "BEGIN" X:= X + 1; PCUM:= PCUM + PX "END"
            "END";
            BININV:= X
        "END"
    "END" "ELSE"
    "BEGIN" X:= PHINV(1 - PROB) *
        SQRT(N * P * (1 - P)) + 0.5 + N * P;
        "IF" X < 0 "THEN" X:= 0 "ELSE"
        "IF" X > N "THEN" X:= N;
        "IF" PROB < P ** N "THEN" BININV:= N + 1 "ELSE"
        "BEGIN" PCUM:= 1 - BIN(X - 1, N, P);
            PX:= BINPROB(X, N, P);
            "IF" PCUM < PROB "THEN"
            "BEGIN" "FOR" PX:=
                PX * X * (1 - P) / (N - X + 1) / P
                "WHILE" PCUM + PX < PROB "DO"
                "BEGIN" X:= X - 1; PCUM:= PCUM + PX "END"
```

```
            "END" "ELSE"
            "BEGIN" "FOR" PCUM:= PCUM - PX
                "WHILE" PCUM > PROB "DO"
                "BEGIN" PX:= PX * (N - X) * P /
                    (X + 1) / (1 - P);
                    X:= X + 1
                "END"; X:= X + 1
            "END";
            BININV:= X
        "END"
    "END"
"END" BININV;
        "EOP"
```

TITLE: **Binprob**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION

The procedure computes the binomial probability function, i.e. the probability that the number of successes in N independent experiments is equal to a given value X. Each of the experiments performed has the same, fixed probability P of success.

KEYWORDS

Binomial probability function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" BINPROB (X, N, P);
"VALUE" X, N, P;
"REAL" X, N, P;
"CODE" 41251;
```

Formal parameters

X: <arithmetic expression>, argument of the probability function;
N: <arithmetic expression>, number of experiments;
P: <arithmetic expression>, probability of success in a single experiment.

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier BINPROB.

The following error messages may appear:

Errornumber 2 (if N is not an integer ≥ 0)
Errornumber 3 (if $P < 0$ or $P > 1$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE

Algol 60

Method and performance

The probability function is computed as follows:

```
BINPROB(X,N,P) = EXP (LOGGAMMA(N + 1) − LOGGAMMA(X + 1)
−LOGGAMMA(N−X+1) + X * LN (P) + (N − X) * LN (1 − P)).
```

The precision is 10^{-10}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        BINPROB(2, 4, .50),
        BINPROB(0, 3, .25),
        BINPROB(4, 9, .90))
"END"
```

Output:

```
 .375000
 .421875
 .000827
```

Source text

```
"CODE" 41251;
"REAL" "PROCEDURE" BINPROB(X, N, P);
"VALUE" X, N, P; "REAL" X, N, P;
BINPROB:= "IF" N < 0 "OR" N > ENTIER(N)
          "THEN" STATAL3 ERROR("("BINPROB")", 2, N)
          "ELSE" "IF" P < 0 "OR" P > 1
          "THEN" STATAL3 ERROR("("BINPROB")", 3, P)
          "ELSE" "IF" X < 0 "OR" X > N "OR" X > ENTIER(X)
          "THEN" 0 "ELSE" "IF" P = 0 "OR" N = 0
          "THEN" ("IF" X = 0 "THEN" 1 "ELSE" 0)
          "ELSE" "IF" P = 1
          "THEN" ("IF" X = N "THEN" 1 "ELSE" 0)
          "ELSE" EXP(LOGGAMMA(N+1) - LOGGAMMA(X+1)
          - LOGGAMMA(N-X+1) + X * LN(P) + (N-X) * LN(1-P));
"EOP"
```

TITLE: **Hyperg**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750515

BRIEF DESCRIPTION

The procedure computes the hypergeometric distribution function, i.e. the probability that in a sample of size N, drawn (without replacement) from a population of size M, the number of elements with a given property is less than or equal to a given value X. R is the total number of elements with the given property.

KEYWORDS

Hypergeometric distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" HYPERG (X, N, R, M);
"VALUE" X, N, R, M;
"REAL" X, N, R, M;
"CODE" 41004;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

N: <arithmetic expression>, size of the sample;

R: <arithmetic expression>, number of elements in the population with the given property;

M: <arithmetic expression>, size of the population.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier HYPERG.

The following error messages may appear:

Errornumber 2	(if N is not an integer $\geq$ 0, or N > M)
Errornumber 3	(if R is not an integer $\geq$ 0, or R > M)
Errornumber 4	(if M is not an integer)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
HYPERGPROB	STATAL 41253
PHI	STATAL 41500

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed exactly for M $\leq$ 100000. Using the value of HYPERGPROB(X,N,R,M) and the recurrent relation of the hypergeometric distribution, the smallest tail of the distribution is computed. The iteration procedure is terminated when the change in the probability is less than 10^{-14}. If the right tail was computed, the result is subtracted from 1.
For M > 100000 a normal approximation is used. (Cf. Molenaar (1973) formula (2.45) on page 136 and formula (2.27) on page 126).
For M $\leq$ 100000 the precision is 10^{-10}, and for M > 100000 the precision is 10^{-5}.

REFERENCE

[1] W. Molenaar *Approximations to the poisson, binomial and hypergeometric distributions,* Mathematical Centre Tracts 31, Mathematical Centre, Amsterdam, 1973.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        HYPERG(  2,   5,  13,    18),
        HYPERG(  7,  18,  12,    60),
        HYPERG(229, 524, 500, 1000))
"END"
```

Output:

```
 .098739
 .996073
 .000019
```

SOURCE TEXT

```
"CODE" 41004;
"REAL" "PROCEDURE" HYPERG(X, N, R, NN);
"VALUE" X, N, R, NN; "REAL" X, N, R, NN;

"BEGIN" "INTEGER" I; "REAL" SUM, LAST, TERM; "BOOLEAN" LEFT;
    "IF" N < 0 "OR" N > NN "OR" N - ENTIER(N) ≠ 0 "THEN"
        STATAL3 ERROR("("HYPERG")", 2, N);
    "IF" R < 0 "OR" R > NN "OR" R - ENTIER(R) ≠ 0 "THEN"
        STATAL3 ERROR("("HYPERG")", 3, R);
    "IF" NN - ENTIER(NN) ≠ 0 "THEN"
        STATAL3 ERROR("("HYPERG")", 4, NN);
```

```
    LEFT:= "TRUE";
    "IF" N > NN / 2 "THEN"
    "BEGIN" LEFT:= "FALSE"; N:= NN - N; X:= R - X - 1 "END";
    "IF" R > NN / 2 "THEN"
    "BEGIN" LEFT:= "NOT" LEFT; R:= NN - R;
        X:= N - X - 1
    "END";
    "IF" N > R "THEN" "BEGIN" I:= N; N:= R; R:= I "END";
    "IF" X <  0 "THEN" HYPERG:= "IF" LEFT "THEN" 0 "ELSE" 1
    "ELSE"
    "IF" X >= N "THEN" HYPERG:= "IF" LEFT "THEN" 1 "ELSE" 0
    "ELSE" "IF" NN > "5 "THEN"
    "BEGIN" "REAL" BETA, TAU, CHI;
        TAU:= SQRT(R * N * (NN - N) * (NN - R) / NN) / NN;
        CHI:= (X + .5 - N * R / NN) / TAU;
        BETA:= (CHI * CHI + 2) / 12;
        X:= "IF" R <= NN / 4 "THEN"
            2 * (SQRT((X + .5 + BETA)
            * (NN - R - N + X + .5 + BETA))
            - SQRT((N - X - .5 + BETA) *
            (R - X - .5 + BETA))) /
            SQRT(NN + 1.5 - NN * NN / 2 / N / (NN - N))
                             "ELSE"
            CHI + (CHI * CHI - 1) *
            (2 * N -  NN) * (NN - 2 * R)
            / 6 / TAU / NN / NN + CHI *
            (1 - 3 * (NN - N) * N / NN / NN)
            / 48 / TAU / TAU;
        HYPERG:= PHI("IF" LEFT "THEN" X "ELSE" -X)
    "END" "ELSE"
    "BEGIN" X:= ENTIER(X);
        TERM:= SUM:= HYPERGPROB(X, N, R, NN);
        "IF" X > (N + 1) * (R + 1) / (NN + 2) "THEN"
        "BEGIN" LEFT:= "NOT" LEFT; SUM:= 0;
            "FOR" I:= X + 1, I + 1 "WHILE" LAST < SUM "DO"
            "BEGIN" TERM:= TERM * (N - I + 1) * (R - I + 1)
                    / I / (NN - R - N + I);
                LAST:= SUM; SUM:= SUM + TERM
            "END"
        "END" "ELSE"
        "FOR" I:= X, I - 1 "WHILE" LAST < SUM "DO"
        "BEGIN" TERM:= TERM * I * (NN - N - R + I)
            / (N - I + 1) / (R - I + 1);
            LAST:= SUM; SUM:= SUM + TERM
        "END";
        HYPERG:= "IF" LEFT "THEN" SUM "ELSE" 1 - SUM
    "END"
"END" HYPERG;
        "EOP"
```

TITLE: **Hyperginv**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750401

BRIEF DESCRIPTION

The procedure computes the left (right) hand tail inverse of the hypergeometric distribution i.e. the largest (smallest) integer for which the left (right) hand tail probability of the distribution is less than or equal to a given value PROB. M is the size of the population, N the size of the sample and R the number of elements in the population with a given property.

KEYWORDS

Inverse hypergeometric distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" HYPERGINV (PROB, N, R, M, LEFT);
"VALUE" PROB, N, R, M, LEFT;
"REAL" N, R, M, "BOOLEAN" LEFT;
"CODE" 41005;
```

Formal parameters

PROB: <arithmetic expression>, tail probability of the value to be computed;

N: <arithmetic expression>, size of the sample;

R: <arithmetic expression>, number of elements in the population with a given property;

M: <arithmetic expression>, size of the population;

LEFT: <boolean expression>, indicating if either the left hand tail inverse or the right hand tail inverse has to be computed. In the first (second) case left should have the value "TRUE" ("FALSE").

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier HYPERGINV. If left is "TRUE" and the probability of MAX(0,N+R−M) is greater than PROB, the value MAX(0,N+R−M)−1 is assigned. If left is "FALSE" and the probability of MIN(N,R) is greater than PROB, the value MIN(N,R)+1 is assigned.

The following error messages may appear:

Errornumber 1	(if PROB ⩽ 0 or PROB ⩾ 1)
Errornumber 2	(if N is not an integer ⩾ 0, or N > M)
Errornumber 3	(if R is not an integer ⩾ 0, or R > M)
Errornumber 4	(if M is not an integer ⩾ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
HYPERG	STATAL 41004
HYPERGPROB	STATAL 41253
PHINV	STATAL 41501

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
In both cases the value of the inverse distribution function is computed using the recurrent relation between successive hypergeometric probabilities, starting with an estimated inverse obtained from a normal approximation.

The precision of the comparisons made is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    "BOOLEAN" LEFT;
    LEFT:= "TRUE";
    OUTPUT(61, "("6(+ZD,/)")",
        HYPERGINV(.25, 10,  8, 12, LEFT),
        HYPERGINV(.52, 12, 14, 30, LEFT),
        HYPERGINV(.01,  3, 10, 15, LEFT),
        HYPERGINV(.74, 11, 10, 25, "NOT" LEFT),
        HYPERGINV(.61,  8,  8, 10, "NOT" LEFT),
        HYPERGINV(.02, 10, 10, 20, "NOT" LEFT))
"END"
```

Output:

```
 +5
 +5
 -1
 +5
 +7
 +8
```

SOURCE TEXT

```
"CODE" 41005;
"REAL" "PROCEDURE" HYPERGINV(PROB, N, R, M, LEFT);
"VALUE" PROB, N, R, M, LEFT;
"REAL" PROB, N, R, M; "BOOLEAN" LEFT;
"BEGIN" "INTEGER" X; "REAL" PX, PCUM, LOW, UP;

    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3 ERROR("("HYPERGINV")", 1, PROB) "ELSE"
    "IF" N > ENTIER(N) "OR" N < 0 "OR" N > M "THEN"
    STATAL3 ERROR("("HYPERGINV")", 2, N) "ELSE"
    "IF" R > ENTIER(R) "OR" R < 0 "OR" R > M "THEN"
    STATAL3 ERROR("("HYPERGINV")", 3, R) "ELSE"
    "IF" M > ENTIER(M) "OR" M < 0 "THEN"
    STATAL3 ERROR("("HYPERGINV")", 4, M);
    LOW:= "IF" N + R - M > 0 "THEN" N + R - M "ELSE" 0;
    UP:= "IF" N < R "THEN" N "ELSE" R;
    "IF" N = 0 "OR" R = 0 "THEN"
        HYPERGINV:= ("IF" LEFT "THEN" -1 "ELSE" +1)
    "ELSE" "IF" N = M "OR" R = M "THEN"
        HYPERGINV:= ("IF" LEFT "THEN" M - 1 "ELSE" M + 1)
    "ELSE" "IF" LEFT "THEN"
    "BEGIN" X:= PHINV(PROB) * SQRT((M - N) * N * R *
        (M - R) / (M * M *(M - 1))) + R * N / M + 0.5;
        "IF" X < LOW "THEN" X:= LOW "ELSE"
        "IF" X > UP "THEN" X:= UP;
        "IF" PROB < HYPERGPROB(LOW, N, R, M) "THEN"
            HYPERGINV:= LOW - 1
        "ELSE"
        "BEGIN" PX:= HYPERGPROB(X, N, R, M);
            PCUM:= HYPERG(X, N, R, M);
            "IF" PCUM > PROB "THEN"
            "BEGIN" "FOR" PCUM:= PCUM - PX
                    "WHILE" PCUM > PROB "DO"
                "BEGIN" PX:= PX * X * (M - N - R + X) /
                    (N - X + 1) / (R - X + 1); X:= X - 1
                "END"; X:= X - 1
            "END" "ELSE"
            "BEGIN" "FOR" PX:= PX * (N - X) * (R - X) /
                (X + 1) / (R - X + 1)
                "WHILE" PCUM + PX < PROB "DO"
                "BEGIN" X:= X + 1; PCUM:= PCUM + PX "END"
            "END";
            HYPERGINV:= X
        "END"
    "END" "ELSE"
    "BEGIN" X:= PHINV(1 - PROB) * SQRT((M - N) * N * R *
        (M - R) / (M * M * (M - 1))) + R * N / M - 0.5;
        "IF" X < LOW "THEN" X:= LOW "ELSE"
        "IF" X > UP "THEN" X:= UP;
        "IF" PROB < HYPERGPROB(UP, N, R, M) "THEN"
            HYPERGINV:= UP + 1
        "ELSE"
```

```
        "BEGIN" PCUM:= 1 - HYPERG(X - 1, N, R, M);
            PX:= HYPERGPROB(X, N, R, M);
            "IF" PCUM < PROB "THEN"
            "BEGIN" "FOR" PX:= PX * X * (M - N - R + X) /
                (N - X + 1) / (R - X + 1)
                "WHILE" PCUM + PX < PROB "DO"
                "BEGIN" X:= X - 1; PCUM:= PCUM + PX "END"
            "END" "ELSE"
            "BEGIN" "FOR" PCUM:= PCUM - PX
                "WHILE" PCUM > PROB "DO"
                "BEGIN" PX:= PX * (N - X) * (R - X) /
                    (X + 1) / (M - N - R + X + 1);
                    X:= X + 1
                "END"; X:= X + 1
            "END";
            HYPERGINV:= X
        "END"
    "END"
"END" HYPERGINV;
         "EOP"
```

TITLE: **Hypergprob**

AUTHORS: R. Kaas, J.M. Buhrman

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the hypergeometric probability function, i.e. the probability that in a sample of size N, drawn (without replacement) from a population of size M, the number of elements with a given property is equal to a given value X. R is the total number of elements with the given property.

KEYWORDS

Hypergeometric probability function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" HYPERGPROB (X, N, R, M);
"VALUE" X, N, R, M;
"REAL" X, N, R, M;
"CODE" 41253;
```

Formal parameters

X: <arithmetic expression>, argument of the probability function;

N: <arithmetic expression>, size of the sample;

R: <arithmetic expression>, number of elements in the population with the given property;

M: <arithmetic expression>, size of the population;

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier HYPERGPROB.

The following error messages may appear:

Errornumber 2	(if N is not an integer $\geqslant 0$, or $N > M$)
Errornumber 3	(if R is not an integer $\geqslant 0$, or $R > M$)
Errornumber 4	(if M is not an integer)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

Language
Algol 60

Method and performance
The probability function is computed as follows:
HYPERGROB(X,N,R,M)=

```
 BINCOEF(X,N)*BINCOEF(M-N,R-X)/BINCOEF(M,R)                    if M<=51,

 EXP(LOGGAMMA(N+1)+LOGGAMMA(M-N+1)+LOGGAMMA(R+1)+
 LOGGAMMA(M-R+1)-LOGGAMMA(N-X+1)-LOGGAMMA(X+1)-                 if M>51,
 LOGGAMMA(M-N-R+X+1)-LOGGAMMA(R-X+1)-LOGGAMMA(M+1))
```

where BINCOEF(K,L) is the binomial coefficient K!/(L!*(K-L)!).

The precision is 10^{-10}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        HYPERGPROB(3,  6,  6, 12),
        HYPERGPROB(2,  3,  9, 11),
        HYPERGPROB(8, 13, 44, 50))
"END"
```

Output:

```
.432900
.436364
.002997
```

Source text

```
"CODE" 41253;
"REAL" "PROCEDURE" HYPERGPROB(X, N, R, M);
"VALUE" X, N, R, M; "REAL" X, N, R, M;
"BEGIN"
    "INTEGER" "PROCEDURE" BINCOEF(N, K); "VALUE" N, K;
    "INTEGER" N, K;
    "BEGIN" "INTEGER" B, L, B1;
        B1:= "IF" K > N - K "THEN" N - K "ELSE" K; B:= 1;
        "FOR" L:= 1 "STEP" 1 "UNTIL" B1 "DO"
        B:= B * (N + 1 - L) // L;
        BINCOEF:= B
    "END";
    "IF" N < 0 "OR" N > M "OR" N > ENTIER(N)
    "THEN" STATAL3 ERROR("("HYPERGPROB")", 2, N)
    "ELSE" "IF" R < 0 "OR" R > M "OR" R > ENTIER(R)
    "THEN" STATAL3 ERROR("("HYPERGPROB")", 3, R)
```

```
    "ELSE" "IF" M > ENTIER(M)
    "THEN" STATAL3 ERROR("("HYPERGPROB")", 4, M);
    "IF" X < 0 "OR" X < N+R-M "OR" X > N "OR" X > R
    "OR" X > ENTIER(X)
    "THEN" HYPERGPROB:= 0 "ELSE"
    "IF" N = 0 "OR" M = 0 "THEN" HYPERGPROB:=
        ("IF" X = 0 "THEN" 1 "ELSE" 0) "ELSE"
    "IF" N = M "OR" R = M "THEN" HYPERGPROB:=
        ("IF" X = M "THEN" 1 "ELSE" 0) "ELSE"
    "IF" M <= 51 "THEN"
    HYPERGPROB:= (BINCOEF(N, X) * BINCOEF(M - N, R - X))
        / BINCOEF(M, R)
    "ELSE"
    "BEGIN" "INTEGER" I; "REAL" PROB; PROB:= 0;
        "FOR" I:= N, M-N, R, M-R "DO"
        PROB:= PROB + LOGGAMMA(I + 1);
        "FOR" I:= N - X, X, M - N - R + X, R - X, M "DO"
        PROB:= PROB - LOGGAMMA(I + 1);
        HYPERGPROB:= EXP(PROB)
    "END"
"END" HYPERGPROB;
         "EOP"
```

TITLE: **Negbin**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION
The procedure computes the negative binomial distribution function, i.e. the probability that the number of experiments, needed to obtain a given numbere of K successes, is less than or equal to a given value X. Each of the experiments performed has the same fixed probability P of success.

KEYWORDS
Negative binomial distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" NEGBIN (X, K, P);
"VALUE" X, K, P;
"REAL" X, K, P;
"CODE" 41009;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
K: <arithmetic expression>, required number of successes;
P: <arithmetic expression>, probability of success in a single experiment.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier NEGBIN
The following error messages may appear:
Errornumber 2 (if K is not an integer $\geq$ 0)
Errornumber 3 (if P $\leq$ 0, or P > 1)

PROCEDURES USED
STATAL3 ERROR STATAL 40100
BIN STATAL 41000

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:
NEGBIN(X,K,P)=1 − BIN(K−1, ENTIER(X), P).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        NEGBIN( 10.5,  7, .44),
        NEGBIN( 15.0, 11, .60),
        NEGBIN(100.0, 73, .50))
"END"
```

Output:

```
 .090843
 .217278
 .000002
```

SOURCE TEXT

```
"CODE" 41009;
"REAL" "PROCEDURE" NEGBIN(X, K, P);
"VALUE" X, K, P; "REAL" X, K, P;
NEGBIN:= "IF" K < 0 "OR" K > ENTIER(K)
         "THEN" STATAL3 ERROR("("NEGBIN")", 2, K)
         "ELSE" "IF" P <= 0 "OR" P > 1
         "THEN" STATAL3 ERROR("("NEGBIN")", 3, P)
         "ELSE" "IF" X >= K "THEN"
             1 - BIN(K - 1, ENTIER(X), P)
         "ELSE" 0;
"EOP"
```

TITLE: **Negbininv**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750401

BRIEF DESCRIPTION

The procedure computes the left (right) hand tail inverse of the negative binomial distribution, i.e. the largest (smallest) integer for which the left (right) hand tail probability of the distribution is less than or equal to a given value PROB. Each of the experiments performed has the same, fixed probability P of success.

KEYWORDS

Inverse negative binomial distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" NEGBININV (PROB, K, P, LEFT);
"VALUE" PROB, K, P, LEFT;
"REAL" PROB, K, P;
"BOOLEAN" LEFT;
"CODE" 41010;
```

Formal parameters

PROB: <arithmetic expression>, tail probability of the value to be computed;

K: <arithmetic expression>, required number of successes;

P: <arithmetic expression>, probability of success in a single experiment;

LEFT: <boolean expression>, indicating if either the left hand tail inverse or the right hand tail inverse has to be computed. In the first (second) case LEFT should have the value "TRUE" ("FALSE").

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier NEGBININV. The value K−1 is assigned if the probability of K is larger than PROB.

The following error messages may appear:

Errornumber 1 (if PROB $\leqslant$ 0 or PROB $\geqslant$ 1)

Errornumber 2 (if K is not an integer $\geqslant$ 0)

Errornumber 3 (if P $\leqslant$ 0, or P > 1)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
NEGBIN	STATAL 41009
NEGBINPROB	STATAL 41254
PHINV	STATAL 41501

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
In both cases the value of NEGBININV is computed using the recurrent relation between successive negative binomial probabilities, starting with an estimated inverse obtained from a normal approximation.

The precision of the comparisons made is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    "BOOLEAN" LEFT;
    LEFT:= "TRUE";
    OUTPUT(61, "("6(ZD,/)")",
        NEGBININV(.25, 10, .4, LEFT),
        NEGBININV(.52, 14, .6, LEFT),
        NEGBININV(.01,  3, .5, LEFT),
        NEGBININV(.74, 11, .7, "NOT" LEFT),
        NEGBININV(.61,  8, .3, "NOT" LEFT),
        NEGBININV(.02, 10, .5, "NOT" LEFT))
"END"
```

Output:

```
20
22
 2
15
25
32
```

Source text

```
"CODE" 41010;
"REAL" "PROCEDURE" NEGBININV(PROB, K, P, LEFT);
"VALUE" PROB, K, P, LEFT;
"REAL" PROB, K, P; "BOOLEAN" LEFT;
"BEGIN" "INTEGER" X; "REAL" PX, PCUM;

    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3 ERROR("("NEGBININV")", 1, PROB) "ELSE"
    "IF" K > ENTIER(K) "OR" K < 0 "THEN"
    STATAL3 ERROR("("NEGBININV")", 2, K) "ELSE"
    "IF" P <= 0 "OR" P > 1 "THEN"
    STATAL3 ERROR("("NEGBININV")", 3, P);

    "IF" P = 1 "OR" K = 0 "THEN"
        NEGBININV:= ("IF" LEFT "THEN" K - 1 "ELSE" K + 1)
    "ELSE" "IF" LEFT "THEN"
    "BEGIN" X:= (PHINV(PROB) *
        SQRT(K * (1 - P)) + K - P / 2) / P;
        "IF" X < K "THEN" X:= K;
        "IF" PROB < P ** K "THEN" NEGBININV:= K - 1 "ELSE"
        "BEGIN" PX:= NEGBINPROB(X, K, P);
            PCUM:= NEGBIN(X, K, P);
            "IF" PCUM > PROB "THEN"
            "BEGIN" "FOR" PCUM:= PCUM - PX
                "WHILE" PCUM > PROB "DO"
                "BEGIN" PX:= PX * (X - K) / (1 - P)
                    / (X - 1);
                    X:= X - 1
                "END"; X:= X - 1
            "END" "ELSE"
            "BEGIN" "FOR" PX:= PX * (1- P) * X / (X - K + 1)
                "WHILE" PCUM + PX < PROB "DO"
                "BEGIN" X:= X + 1; PCUM:= PCUM + PX "END"
            "END";
            NEGBININV:= X
        "END"
    "END" "ELSE"
    "BEGIN" X:= (PHINV(1 - PROB) *
        SQRT(K * (1 - P)) + K + P / 2) / P;
        "IF" X > K "THEN" X:= K;
        PCUM:= 1 - NEGBIN(X - 1, K, P);
        PX:= NEGBINPROB(X, K, P);
        "IF" PCUM < PROB "THEN"
        "BEGIN" "FOR" PX:= PX * (X - K) / (1 - P) / (X - 1)
            "WHILE" PCUM + PX < PROB "DO"
            "BEGIN" X:= X - 1; PCUM:= PCUM + PX "END"
        "END" "ELSE"
        "BEGIN" "FOR" PCUM:= PCUM - PX
            "WHILE" PCUM > PROB "DO"
            "BEGIN" PX:= PX * (1 - P) * X / (X - K + 1);
                X:= X + 1
            "END"; X:= X + 1
```

```
        "END";
        NEGBININV:= X
    "END"
"END" NEGBININV;
          "EOP"
```

TITLE: **Negbinprob**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION
The procedure computes the negative binomial probability function, i.e. the probability that the number of experiments, needed to obtain a given number of K successes, is equal to a given value X. Each of the experiments performed has the same, fixed probability P of success.

KEYWORDS
Negative binomial probability function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" NEGBINPROB (X, K, P);
"VALUE" X, K, P;
"REAL" X, K, P;
"CODE" 41254;
```
Formal parameters

X: <arithmetic expression>, argument of the probability function;
K: <arithmetic expression>, required number of successes;
P: <arithmetic expression>, probability of success in a single experiment.

DATA AND RESULTS
The value of the probability function is assigned to the procedure identifier NEGBINPROB.
The following error messages may appear:

Errornumber 2	(if K is not an integer ≥ 0)
Errornumber 3	(if $P \leq 0$, or $P > 1$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
BINPROB	STATAL 41251

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The probability function is computed as follows:

```
NEGBINPROB(X,K,P) = P * BINPROB (K-1, ENTIER(X-1), P).
```

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        NEGBINPROB( 5, 3, .50),
        NEGBINPROB( 4, 1, .25),
        NEGBINPROB(10, 5, .90))
"END"
```

Output:

```
 .187500
 .105469
 .000744
```

SOURCE TEXT

```
"CODE" 41254;
"REAL" "PROCEDURE" NEGBINPROB(X, K, P);
    "VALUE" X, K, P; "REAL" X, K, P;
NEGBINPROB:= "IF" K < 0 "OR" K > ENTIER(K)
             "THEN" STATAL3 ERROR("("NEGBINPROB")", 2, K)
             "ELSE" "IF" P <= 0 "OR" P > 1
             "THEN" STATAL3 ERROR("("NEGBINPROB")", 3, P)
             "ELSE" "IF" X < K "OR" X > ENTIER(X) "THEN" 0
             "ELSE" "IF" P = 0 "THEN" 0
             "ELSE" "IF" P = 1 "OR" K = 0
             "THEN" ("IF" X = K "THEN" 1 "ELSE" 0)
             "ELSE" P * BINPROB(K - 1, X - 1, P);
         "EOP"
```

TITLE: **Poisson**

AUTHORS: M. van Gelderen, J.M. Buhrman

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION
The procedure computes the Poisson distribution function, i.e. the probability that a random variable having a Poisson distribution with mean MU, is less than or equal to a given value X.

KEYWORDS
Poisson distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" POISSON (X, MU);
"VALUE" X, MU;
"REAL" X, MU;
"CODE" 41013;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
MU: <arithmetic expression>, mean of the distibution.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier POISSON.
The following error message may appear:
Errornumber 2 (if MU $\leq$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 41500

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed exactly for MU $\leq$ 1000, and is approximated by PHI(Y) for MU > 1000, where Y is the approximated normal deviate. (See Molenaar (1973), formula II.5.2 or Peizer & Pratt).

The precision is 10^{-10}.

REFERENCES

[1] W. Molenaar: *Approximations to the Poisson, binomial and hypergeometric distribution functions,* Mathematical Centre Tracts 31, Mathematical Centre, Amsterdam, 1973.

[2] D.B. Peizer & J.W. Pratt: *A normal approximation for binomial, f, beta and other common, related tail probabilities,* J. Amer. Stat. Assoc., 63, (1968) P.1416-1456.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        POISSON(2.958, 3.10),
        POISSON( 17.1, 20.5),
        POISSON(  950, 1100))
"END"
```

Output:

```
 .401163
 .260503
 .000002
```

SOURCE TEXT

```
"CODE" 41013;
"REAL" "PROCEDURE" POISSON(X, MU);
"VALUE" X, MU; "REAL" X, MU;
"BEGIN" "INTEGER" IX;
    "REAL" "PROCEDURE" KSI(K, L);
    "VALUE" K, L; "REAL" L; "INTEGER" K;
    "BEGIN" "REAL" U, U2, W; W:= SQRT(L);
        U:= 2 * (SQRT(K + 1) - W);
        U2:= U * U; KSI:=
        U + (U2 - 4) / 12 / W + (-U2 * U + 10 * U) / 72 /
        L + (21 * U2 * U2 - 371 * U2 - 52) / 6480 / L / W
    "END";

    IX:= ENTIER(X);
    "IF" IX < 0 "THEN" POISSON:= 0 "ELSE"
    "IF" MU <= 0 "THEN"
        POISSON:= STATAL3 ERROR("("POISSON")",2,MU)
    "ELSE"
    "IF" MU > 1000 "THEN" POISSON:= PHI(KSI(IX, MU))
    "ELSE"
    "BEGIN" "INTEGER" I, MODUS; "REAL" MODUSPROB, PROB, CUM;
        MODUS:= ENTIER(MU) + 1; "IF" IX < MODUS "THEN"
        "BEGIN" PROB:= CUM:= POISSONPROB(IX, MU);
            "FOR" I:= IX "STEP" -1 "UNTIL" 1 "DO"
            "BEGIN" PROB:= PROB * I / MU;
```

```
                CUM:= CUM + PROB
            "END"
        "END" "ELSE"
        "BEGIN" MODUSPROB:= PROB:= CUM:=
            POISSONPROB(MODUS, MU);
            "FOR" I:= MODUS "STEP" -1 "UNTIL" 1 "DO"
            "BEGIN" PROB:= PROB * I / MU;
                CUM:= CUM + PROB
            "END";  PROB:= MODUSPROB;
            "FOR" I:= MODUS + 1 "STEP" 1 "UNTIL" IX "DO"
            "BEGIN" PROB:= PROB * MU / I;
                CUM:= CUM + PROB
            "END"
        "END";
        POISSON:= CUM
    "END"
"END" POISSON;
         "EOP"
```

TITLE: **Poissoninv**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750401

BRIEF DESCRIPTION
The procedure computes the left (right) hand tail inverse of the Poisson distribution with mean MU, i.e. the largest (smallest) integer for which the left (right) hand tail probability of the distribution is less than or equal to a given value PROB.

KEYWORDS
Inverse Poisson distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" POISSONINV (PROB, MU, LEFT);
"VALUE" PROB, MU, LEFT;
"REAL" PROB, MU;
"BOOLEAN" LEFT;
"CODE" 41014;
```
Formal parameters

PROB: <arithmetic expression>, tail probability of the value to be computed;

MU: <arithmetic expression>, mean of the distibution;

LEFT: <boolean expression>, indicating if either the left hand tail inverse or the right hand tail inverse has to be computed. In the first (second) case left should have the value "TRUE" ("FALSE").

DATA AND RESULTS
The value of the inverse distribution function is assigned to the procedure identifier POISSONINV. The value -1 is assigned if the probability of 0 is larger than PROB.
The following error messages may appear:

Errornumber 1 (if PROB $\leqslant$ 0 or PROB $\geqslant$ 1)
Errornumber 2 (if MU $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
POISSON	STATAL 41013
POISSONPROB	STATAL 41252
PHINV	STATAL 41501

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
In both cases the inverse distribution function is computed using the recurrent relation between successive Poisson probabilities, starting with an estimated inverse obtained from a normal approximation.

The precision of the comparisons made is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    "BOOLEAN" LEFT;
    LEFT:= "TRUE";
    OUTPUT(61, "("6(+ZD,/)")",
        POISSONINV(.25,  1, LEFT),
        POISSONINV(.52,  2, LEFT),
        POISSONINV(.04, 17, LEFT),
        POISSONINV(.74, 11, "NOT" LEFT),
        POISSONINV(.61,  8, "NOT" LEFT),
        POISSONINV(.02,  4, "NOT" LEFT))
"END"
```

Output:

```
 -1
 +1
 +9
+10
 +8
+10
```

SOURCE TEXT

```
"CODE" 41014;
"REAL" "PROCEDURE" POISSONINV(PROB, MU, LEFT);
"VALUE" PROB, MU, LEFT;
"REAL" PROB, MU; "BOOLEAN" LEFT;
"BEGIN" "INTEGER" X; "REAL" PX, PCUM;

    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3 ERROR("("POISSONINV")", 1, PROB) "ELSE"
    "IF" MU <= 0 "THEN"
    STATAL3 ERROR("("POISSONINV")", 2, MU);

    "IF" LEFT "THEN"
    "BEGIN" X:= (PHINV(PROB) / 2 + SQRT(MU)) ** 2 - 1;
        "IF" X < 0 "THEN" X:= 0;
        "IF" PROB < EXP(-MU) "THEN" POISSONINV:= -1 "ELSE"
```

```
    "BEGIN" PX:= POISSONPROB(X, MU);
        PCUM:= POISSON(X, MU);
        "IF" PCUM > PROB "THEN"
        "BEGIN" "FOR" PCUM:= PCUM - PX
            "WHILE" PCUM > PROB "DO"
            "BEGIN" PX:= PX * X / MU; X:= X - 1 "END";
            X:= X - 1
        "END" "ELSE"
        "BEGIN" "FOR" PX:= PX * MU / (X + 1)
            "WHILE" PCUM + PX < PROB "DO"
            "BEGIN" X:= X + 1; PCUM:= PCUM + PX "END"
        "END";
        POISSONINV:= X
    "END"
"END" "ELSE"
"BEGIN" X:= (PHINV(1 - PROB) / 2 + SQRT(MU)) ** 2 + 1;
    "IF" X < 0 "THEN" X:= 0;
    PCUM:= 1 - POISSON(X - 1, MU);
    PX:= POISSONPROB(X, MU);
    "IF" PCUM < PROB "THEN"
    "BEGIN" "FOR" PX:= PX * X / MU
        "WHILE" PCUM + PX < PROB "DO"
        "BEGIN" X:= X - 1; PCUM:= PCUM + PX "END"
    "END" "ELSE"
    "BEGIN" "FOR" PCUM:= PCUM - PX
        "WHILE" PCUM > PROB "DO"
        "BEGIN" PX:= PX * MU / (X + 1); X:= X + 1 "END";
        X:= X + 1
    "END";
    POISSONINV:= X
"END"
"END" POISSONINV;
        "EOP"
```

TITLE: **Poissonprob**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION

The procedure computes the Poisson probability function, i.e. the probability that a random variable having a Poisson distribution with mean MU, is equal to a given value X.

KEYWORDS

Poisson probability function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" POISSONPROB (X, MU);
"VALUE" X, MU;
"REAL" X, MU;
"CODE" 41252;
```

Formal parameters

X: <arithmetic expression>, argument of the probability function;

MU: <arithmetic expression>, mean of the distribution.

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier POISSONPROB.

The following error message may appear:

Errornumber 2 (if $MU \leqslant 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The probability function is computed as follows:

```
POISSONPROB(X, MU) = EXP(-MU+X*LN(MU) - LOGGAMMA(X+1)).
```

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        POISSONPROB( 0,  1),
        POISSONPROB(15, 15),
        POISSONPROB(26, 30))
"END"
```

Output:

```
 .367879
 .102436
 .058979
```

SOURCE TEXT

```
"CODE" 41252;
"REAL" "PROCEDURE" POISSONPROB(X, MU);
"VALUE" X, MU; "REAL" X, MU;
POISSONPROB:= "IF" MU <= 0
             "THEN" STATAL3 ERROR("("POISSONPROB")", 2, MU)
             "ELSE" "IF" X < 0 "OR" X > ENTIER(X) "THEN" 0
             "ELSE" EXP(-MU + X * LN(MU) - LOGGAMMA(X+1));
         "EOP"
```

TITLE: **Wilcox**

AUTHORS: J.M. Burhman, J.G. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 780601

BRIEF DESCRIPTION

The procedure computes the distribution function of the test statistic `W` of Wilcoxon's two sample test (under the null-hypothesis), i.e. the probability that the value of `W` is less than or equal to a given value `X`. Let `M` and `N` be the sizes of two independent samples from two (possibly) different distributions. Wilcoxon's two sample test statistic `W` is the number of times that an observation from the first sample is larger than an observation from the second sample, multiplied by two. The procedure computes the distribution function of `W` under the null-hypothesis that the two samples are from the same (continuous) distribution. (Thus it is assumed that no equal observations occur in the joint sample).

KEYWORDS

Null-hypothesis distribution function of Wilcoxon's two sample test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" WILCOX (X, M, N);
"VALUE" X, M, N;
"REAL" X, M, N;
"CODE" 41020;
```

Formal parameters

`X`: <arithmetic expression>, argument of the distribution function;
`M`: <arithmetic expression>, size of the first sample;
`N`: <arithmetic expression>, size of the second sample.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier `WILCOX`.

The following error messages may appear:

Errornumber 2	(if `M` is not an integer $\geqslant 0$)
Errornumber 3	(if `N` is not an integer $\geqslant 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 41500
PHIDENS	STATAL 41752

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed exactly by using a recurrence relation when M*N<=400, and it is approximated by using an Edgeworth expansion when M*N>400 (cf. Fix and Hodges, 1955). In the trivial case that M=1 or N=1, the distribution function equals ENTIER(X/2+1)/(N+1) or ENTIER(X/2+1)/(M+1) respectively. The computation using the recurrence relation is exact, but would require too much time and/or memory for M*N> 400. The approximation requires little time and memory, but the precision is 10^{-5}. However, this precision is not even guaranteed if M or N is small (say 2 to 5), but M*N>400.

REFERENCE
[1] E. Fix and J.L. Hodges Jr.: *Significance probabilities of the Wilcoxon test,* Ann. Math. Stat., 26 (1955), P.301.

EXAMPLE OF USE
```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        WILCOX( 8,   3, 4),
        WILCOX(140, 140, 1),
        WILCOX(400,  50, 8))
"END"
```

Output:
```
 .314286
 .503546
 .504411
```

SOURCE TEXT
```
"CODE" 41020;
"REAL" "PROCEDURE" WILCOX(X,M,N);
"VALUE" X,M,N; "REAL" X,M,N;
"BEGIN"
    "INTEGER" "PROCEDURE" MIN(A,B);
    "VALUE" A,B; "INTEGER" A,B;
        MIN := "IF" A > B "THEN" B "ELSE" A;
    "INTEGER" "PROCEDURE" MAX(A,B);
    "VALUE" A,B; "INTEGER" A,B;
        MAX := "IF" A > B "THEN" A "ELSE" B;
    "REAL" WP1;    "BOOLEAN" X EVEN, RIGHT;
    "INTEGER" MN;
    "IF" M < 0 "OR" ENTIER(M) < M "THEN"
    STATAL3 ERROR("("WILCOX")", 2, M);
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("WILCOX")", 3, N);
    MN := M * N;
```

```
X := ENTIER(X/2);
"IF" X < MN/2 "THEN" RIGHT := "FALSE" "ELSE"
"BEGIN" RIGHT := "TRUE"; X := MN-X-1 "END";
X EVEN := ENTIER(X/2) * 2 = X;
M := MIN(M,N); N := MN/M;
"IF" X < 0 "THEN" WP1 := 0 "ELSE"
"IF" M = 1 "THEN" WP1 := (X+1) / (N+1) "ELSE"
"IF" M = 2 "THEN" WP1 := ("IF" X EVEN "THEN"
    (X+2)*(X+2) / (2*(N+1)*(N+2)) "ELSE"
    (X+1)*(X+3) / (2*(N+1)*(N+2)) )     "ELSE"
"IF" 2*X = MN - 1 "THEN" WP1 := .5 "ELSE"
"IF" MN > 400 "THEN"
"BEGIN" "INTEGER" NOEM, N2, N3, N4, M2, M3, M4;
    "REAL" FOY, F3Y, F5Y, F7Y, T3, T5, T7, Y, Y2;
    Y := (2*X + 1 - MN) / SQRT(MN * (M + N + 1) / 3);
    Y2 := Y * Y;
    NOEM := 10 * MN * (M + N + 1); FOY := PHIDENS(Y);
    N2 := N * N; N3 := N2 * N; N4 := N2 * N2;
    M2 := M * M; M3 := M2 * M; M4 := M2 * M2;
    F3Y := Y * (3 - Y2);
    F5Y := Y * (-15 + Y2 * (10 - Y2));
    F7Y := Y * (105 - Y2 * (105 - Y2 * (21 - Y2)));
    T3 := (M2 + N2 + MN + M + N) / NOEM / 2;
    T5 := (2 * (M4 + N4) + 4 *
         (M3 * N + N3 * M + M3 + N3) +
         6 * M2 * N2 + 7 * MN * (M + N) + M2 + N2 +
         2 * MN - M - N) / (NOEM * NOEM * 2.1);
    T7 := (M2 + N2 + MN + M + N) ** 2
         / (NOEM * NOEM * 8);
    WP1 := MAX(PHI(Y) - FOY *
         (T3 * F3Y - T5 * F5Y - T7 * F7Y), 0);
"END" "ELSE"
"BEGIN" "INTEGER" I,J,W,UP,UP1,UP2; "REAL" H1,H2;
    "IF" N * (X+1) <= 12000 "THEN"
    "BEGIN" M := N; N := MN / M "END";
    "BEGIN" "REAL" "ARRAY" WP[0:X, 1:M];
        UP2 := MIN(M, ENTIER((MN-X-1)/(N-1)));
        "FOR" I := MAX(2, -ENTIER(X/2-M)) "STEP" 1
        "UNTIL" UP2 "DO"
        "BEGIN" UP:= X-(M-I)*2; UP1 := MIN(UP,I-1);
            H1 := 1/(I+1);
            "FOR" W:= MAX(0, X-(M-I)*N) "STEP" 1
            "UNTIL" UP1 "DO"
            WP[W,I] := H1 * (W+1);
        "END";
        "FOR" J := 2 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" UP:= MIN(X-(M-1)*J , J-1);
          H2 := 1/(J+1);
          "FOR" W:= MAX(0, X-(M-2)*N-J) "STEP" 1
          "UNTIL" UP "DO"
          WP[W,1] := H2 * (W+1);
          UP2 := ("IF" J*M < X+1 "THEN"
            ENTIER((MN-X-1)/(N-J)) "ELSE" M);
          "FOR" I := MAX(2,-ENTIER(X/J-M)) "STEP" 1
```

```
              "UNTIL" UP2 "DO"
              "BEGIN" UP:= X - (M-I)*J;
                H1:= J/(I+J); H2 := I/(I+J);
                UP1 := MIN(UP,J-1);
                "FOR"  W := MAX(0, X-(M-I)*N) "STEP" 1
                "UNTIL" UP1 "DO"
                  WP[W,I] := WP[W,I]*H1;
                UP1 := MIN(UP,I*J-I-1);
                "FOR"  W := MAX(J, X-(M-I)*N) "STEP" 1
                "UNTIL" UP1 "DO"
                  WP[W,I] := WP[W,I]*H1 + WP[W-J,I-1]*H2;
                UP1 := MIN(UP,I*J-1);
                "FOR"  W:= MAX(I*J-I, X-(M-I)*N) "STEP" 1
                "UNTIL" UP1 "DO"
                  WP[W,I] := H1 + WP[W-J,I-1]*H2;
               "END"
             "END";
             WP1 := WP[X,M];
         "END";
    "END";
    WILCOX := "IF" RIGHT "THEN" 1-WP1 "ELSE" WP1;
"END"  WILCOXCDF;
          "EOP"
```

TITLE: **Wilcoxinv**

AUTHOR: J.M. Burhman

INSTITUTE: Mathematical Centre

RECEIVED: 780601

BRIEF DESCRIPTION

The procedure computes the left (right) hand tail inverse of the distribution of the test statistic W of Wilcoxon's two sample test under the null-hypothesis, i.e. the largest (smallest) even integer for which the left (right) hand tail probability of the distribution is less than or equal to a given value PROB. M and N are the sizes of the first and second sample, respectively.

KEYWORDS

Inverse null-hypothesis distribution function of Wilcoxon's two sample test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" WILCOXINV (PROB, M, N, LEFT);
"VALUE" PROB, M, N, LEFT;
"REAL" PROB, M, N;
"BOOLEAN" LEFT;
"CODE" 41021;
```

Formal parameters

PROB: <arithmetic expression>, tail probability of the value to be computed;

M: <arithmetic expression>, size of the first sample;

N: <arithmetic expression>, size of the second sample;

LEFT: <boolean expression>, indicating if either the left hand tail inverse or the right hand tail inverse has to be computed. In the first (second) case LEFT should have the value "TRUE" ("FALSE").

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier WILCOXINV. The value -2 (2 * M * N+2) is assigned if the probability of 0 (2 * M * N) is larger than PROB.

The following error messages may appear:

Errornumber 1 (if PROB $\leqslant$ 0 or PROB $\geqslant$ 1)

Errornumber 2 (if M is not an integer $\geqslant$ 0)

Errornumber 3 (if N is not an integer $\geqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
WILCOX	STATAL 41020
PHINV	STATAL 41501

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The computation is started by estimating the inverse using a simple normal approximation. If M*N≤400 or M=1 or N=1 the estimated inverse is used to find the correct inverse from the exact distribution, otherwise the estimated inverse is corrected as far as possible with use of the Edgeworth expansion mentioned in section 1.1.5.1.

The precision of the comparisons made is 10^{-14}.

EXAMPLE OF USE
```
"BEGIN"
   "BOOLEAN" LEFT;
   LEFT:= "TRUE";
   OUTPUT(61, "("6(-3ZD,/)")",
       WILCOXINV( .39,  4,  8, LEFT),
       WILCOXINV( .01, 26, 10, LEFT),
       WILCOXINV(.001,  4,  4, LEFT),
       WILCOXINV( .15,  8,  5, "NOT" LEFT),
       WILCOXINV( .01, 26, 10, "NOT" LEFT),
       WILCOXINV( .01, 40, 30, "NOT" LEFT))
"END"
```

Output:
```
  26
 128
  -2
  56
 392
1202
```

SOURCE TEXT

```
"CODE" 41021;
"REAL" "PROCEDURE" WILCOXINV(PROB, M, N, LEFT);
"VALUE" PROB, M, N, LEFT; "REAL" PROB, M, N; "BOOLEAN" LEFT;
"BEGIN"
    "INTEGER" X, W, WI, MN; "REAL" Z; "BOOLEAN" MN EVEN;
    MN := M * N;
    MN EVEN := ENTIER(MN/2) * 2 = MN;
    X := "IF" MN EVEN "THEN" MN/2 - 1 "ELSE" MN/2 - 1.5;
    PROB := PROB + "-13*(1-PROB);
    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3 ERROR("("WILCOXINV")", 1, PROB) "ELSE"
    "IF" M < 0 "OR" ENTIER(M) < M "THEN"
    STATAL3 ERROR("("WILCOXINV")", 2, M) "ELSE"
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("WILCOXINV")", 2, N) "ELSE"
    "IF" MN = 0 "THEN" WI := -2 "ELSE"
    "IF" PROB = .5 "THEN" WI := ENTIER((MN-1)/2) * 2 "ELSE"
    "IF" M = 1 "THEN" WI := ENTIER(PROB*(N+1)) * 2 - 2
    "ELSE"
    "IF" N = 1 "THEN" WI := ENTIER(PROB*(M+1)) * 2 - 2
    "ELSE" "IF" MN > 400 "OR" M=2 "OR" N=2 "THEN"
    "BEGIN" Z := PHINV(PROB) * SQRT(MN*(M+N+1)/3) + MN;
        WI := W := ENTIER(Z/2) * 2;
        "IF" WI < 0 "THEN" WI := W := 0;
        "IF" WI > 2*MN "THEN" WI := W := 2*MN;
        "IF" WILCOX(W, M, N) <= PROB "THEN"
        "BEGIN" "FOR" W := W + 2
          "WHILE" WILCOX(W, M, N) <= PROB "DO" WI := W
        "END" "ELSE"
        "BEGIN" "FOR" W := W - 2
           "WHILE" WILCOX(W, M, N) > PROB "DO" WI := W;
           WI := WI - 2;
        "END";
    "END" "ELSE"
    "BEGIN" "INTEGER" I,J,UP,UP1; "REAL" H1,H2;
        "BOOLEAN" RIGHT, EQUAL; "REAL" "ARRAY" WCMN[-1:X+2];
        "INTEGER" "PROCEDURE" MAX(A,B);
        "VALUE" A,B; "INTEGER" A,B;
            MAX := "IF" A > B "THEN" A "ELSE" B;
        "INTEGER" "PROCEDURE" MIN(A,B);
        "VALUE" A,B; "INTEGER" A,B;
            MIN := "IF" A > B "THEN" B "ELSE" A;
        RIGHT := PROB > .5;
        "IF" RIGHT "THEN" PROB := 1 - PROB;
        I := MAX(M,N); J := MIN(M,N);
        "IF" I * (X+1) > 12000 "THEN"
        "BEGIN" M := J; N := I "END" "ELSE"
        "BEGIN" M := I; N := J "END";
        "BEGIN" "REAL" "ARRAY" WP[0:X, 1:M];
            "FOR" I := MAX(2,-ENTIER(X/2-M)) "STEP" 1
            "UNTIL" M "DO"
            "BEGIN" UP:= X-(M-I)*2; UP1 := MIN(UP,I);
```

```
          H1 := 1/(I+1);
          "FOR" W:= 0 "STEP" 1 "UNTIL" UP1 "DO"
          WP[W,I] := H1;
        "END";
        "FOR" J := 2 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" UP:= MIN(X-(M-1)*J  , J); H2 := 1/(J+1);
          "FOR" W:= 0 "STEP" 1 "UNTIL" UP "DO"
          WP[W,1] := H2;
          "FOR" I := MAX(2,-ENTIER(X/J-M)) "STEP" 1
          "UNTIL" M "DO"
          "BEGIN" UP:= X - (M-I)*J;
            H1:= J/(I+J); H2 := I/(I+J);
            UP1 := MIN(UP,J-1);
            "FOR"  W := 0 "STEP" 1 "UNTIL" UP1 "DO"
              WP[W,I] := WP[W,I]*H1;
            UP1 := MIN(UP,I*J-I);
            "FOR"  W := J "STEP" 1 "UNTIL" UP1 "DO"
              WP[W,I] := WP[W,I]*H1 + WP[W-J,I-1]*H2;
            UP1 := MIN(UP,I*J);
            "FOR"  W := I*J-I+1 "STEP" 1 "UNTIL" UP1
            "DO"  WP[W,I] := WP[W-J,I-1]*H2;
          "END"
        "END";
        WCMN[-1] := 0;
        WCMN[0] := WP[0,M];
        "FOR" W := 1 "STEP" 1 "UNTIL" X "DO"
            WCMN[W] := WCMN[W-1] + WP[W,M];
        "IF" MN EVEN "THEN"
          "BEGIN" WCMN[X+1] := 1 - WCMN[X];
                  WCMN[X+2] := 1 - WCMN[X-1];
          "END"
        "ELSE"
          "BEGIN" WCMN[X+1] := .5;
                  WCMN[X+2] := 1 - WCMN[X];
          "END";
    "END";
    WI := PHINV(PROB) * SQRT(MN*(M+N+1)/3) + MN;
    W := ENTIER(WI/2);
    "IF" W < 0 "THEN" WI := W := 0 "ELSE" WI:= W;
    "IF" WCMN[W] <= PROB "THEN"
    "BEGIN" "FOR" W := W + 1
        "WHILE" WCMN[W] <= PROB "DO" WI := W
    "END" "ELSE"
    "BEGIN" "FOR" W := W - 1
        "WHILE" WCMN[W] > PROB "DO" WI := W;
        WI := WI - 1;
    "END";
    EQUAL := WCMN[WI] = PROB;
    "IF" RIGHT "THEN"
    "BEGIN" "IF" EQUAL "THEN" WI := 2 * (MN - WI - 1)
                          "ELSE" WI := 2 * (MN - WI - 2)
    "END"
    "ELSE" WI := 2 * WI;
"END";
```

```
   WILCOXINV := "IF" LEFT "THEN" WI "ELSE" 2*MN - WI;
"END" WILCOXINV;
        "EOP"
```

TITLE: **Wilcoxprob**

AUTHOR: J.M. Burhman

INSTITUTE: Mathematical Centre

RECEIVED: 780601

BRIEF DESCRIPTION

The procedure computes the probability function of the test statistic W of Wilcoxon's two sample test (under the null-hypothesis), i.e. the probability that the value of W is equal to a given value X. Let M and N be the sizes of two independent samples from two (possibly) different distributions. Wilcoxon's two sample test statistic W is the number of times that an observation from the first sample is larger than an observation from the second sample. The procedure computes the probability function of W under the null-hypothesis that the two samples are from the same (continuous) distribution. (Thus it is assumed that no equal observations occur in the joint sample).

KEYWORDS

Null-hypothesis probability function of Wilcoxon's two sample test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" WILCOXPROB (X, M, N);
"VALUE" X, M, N;
"REAL" X, M, N;
"CODE" 41022;
```

Formal parameters

X: <arithmetic expression>, argument of the probability function;

M: <arithmetic expression>, size of the first sample;

N: <arithmetic expression>, size of the second sample.

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier WILCOXPROB.

The following error messages may appear:

Errornumber 2 (if M is not an integer $\geqslant 0$)

Errornumber 3 (if N is not an integer $\geqslant 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
WILCOX	STATAL 41020

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The probability function is computed exactly, by using a recurrence relation, when M*N≤400, and is approximated as WILCOXPROB(X,M,N)= WILCOX(X,M,N) −WILCOX(X−2,M,N) when M*N<400. In the trivial cases that M=1 or N=1, the probability function equals 1/(N+1) or 1/(M+1), respectively. The computation using the recurrence relation is exact, but would require too much time and/or memory for M*N>400. The approximation requires little time and memory, but the precision is 10^{-5}. However, the precision is not even guaranteed if M or N is small (say 2 to 5), but M*N>400.

EXAMPLE OF USE

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        WILCOXPROB( 6,   4, 3),
        WILCOXPROB(140, 140, 1),
        WILCOXPROB(400,  50, 8))
"END"
```

Output:

```
 .085714
 .007092
 .008822
```

SOURCE TEXT

```
"CODE" 41022;
"REAL" "PROCEDURE" WILCOXPROB(X,M,N);
"VALUE" X,M,N; "REAL" X,M,N;
"BEGIN"
    "INTEGER" "PROCEDURE" MIN(A,B);
    "VALUE" A,B; "INTEGER" A,B;
        MIN := "IF" A > B "THEN" B "ELSE" A;
    "INTEGER" "PROCEDURE" MAX(A,B);
    "VALUE" A,B; "INTEGER" A,B;
        MAX := "IF" A > B "THEN" A "ELSE" B;
    "REAL" WP1;
    "INTEGER" MN;
    "IF" M < 0 "OR" ENTIER(M) < M "THEN"
    STATAL3 ERROR("("WILCOXPROB")", 2, M);
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("WILCOXPROB")", 3, N);
    MN := M * N;
    X := MIN(X/2, MN - X/2);
    M := MIN(M,N); N :=  MN/M;
    "IF" ENTIER(X) < X "THEN" WP1 := 0 "ELSE"
    "IF" X < 0 "THEN" WP1 := 0 "ELSE"
    "IF" MN = 0 "THEN" WP1 := 1 "ELSE"
```

```
"IF" M = 1 "THEN" WP1 := 1/(N+1) "ELSE"
"IF" M = 2 "THEN"
    WP1 := ENTIER(X/2+1) / ((N+1)*(N+2)/2) "ELSE"
"IF" MN > 400 "THEN"
    WP1 := WILCOX(2*X,M,N) - WILCOX(2*X-2,M,N) "ELSE"
"BEGIN" "INTEGER" I,J,W,UP,UP1,UP2; "REAL" H1,H2;
    "IF" N * (X+1) <= 12000 "THEN"
    "BEGIN" M := N; N := MN/M "END";
    "BEGIN" "REAL" "ARRAY" WP[0:X, 1:M];
        UP2 := MIN(M, ENTIER((MN-X)/(N-1)));
        "FOR" I := MAX(2,-ENTIER(X/2-M)) "STEP" 1
        "UNTIL" UP2 "DO"
        "BEGIN" UP:= X-(M-I)*2; UP1 := MIN(UP,I);
            H1 := 1/(I+1);
          "FOR" W:= MAX(0, X-(M-I)*N) "STEP" 1
          "UNTIL" UP1 "DO"
          WP[W,I] := H1;
        "END";
        "FOR" J := 2 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" UP:= MIN(X-(M-1)*J , J); H2 := 1/(J+1);
          "FOR" W:= MAX(0, X-(M-2)*N-J) "STEP" 1
          "UNTIL" UP "DO"
          WP[W,1] := H2;
          UP2 := "IF" J*M<X+1 "THEN"
              ENTIER((MN-X)/(N-J)) "ELSE" M;
          "FOR" I := MAX(2,-ENTIER(X/2-M)) "STEP" 1
          "UNTIL" UP2 "DO"
          "BEGIN" UP:= X - (M-I)*J;
            H1:= J/(I+J); H2 := I/(I+J);
            UP1 := MIN(UP,J-1);
            "FOR"  W := MAX(0, X-(M-I)*N) "STEP" 1
            "UNTIL" UP1 "DO"
              WP[W,I] := WP[W,I]*H1;
            UP1 := MIN(UP,I*J-I);
            "FOR"  W := MAX(J, X-(M-I)*N) "STEP" 1
            "UNTIL" UP1 "DO"
              WP[W,I] := WP[W,I]*H1 + WP[W-J,I-1]*H2;
            UP1 := MIN(UP,I*J);
            "FOR"  W:=MAX(I*J-I+1,X-(M-I)*N) "STEP" 1
            "UNTIL" UP1 "DO"
              WP[W,I] := WP[W-J,I-1]*H2;
          "END"
        "END";
        WP1 := WP[X,M];
    "END";
"END";
WILCOXPROB := WP1
"END"  WILCOXPROB;
      "EOP"
```

TITLE: **Run**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750701

BRIEF DESCRIPTION

The procedure computes the distribution function of the test statistic W of the two sample run test (under the nul-hypothesis), i.e. the probability that the value of the test statistic is less than or equal to a given value X. The test statistic is equal to the number of runs in the ordered joint sample. M and N are the sizes of the first and second sample, respectively. The procedure computes the distribution function under the null-hypothesis that the two samples are from the same (continuous) distribution. (Thus it is assumed that no equal observations occur in the joint sample).

KEYWORDS

Null-hypothesis distribution function of the two sample run test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" RUN (X, M, N);
"VALUE" X, M, N;
"REAL" X, M, N;
"CODE" 41023;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;
M: <arithmetic expression>, size of the first sample;
N: <arithmetic expression>, size of the second sample.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier RUN.

The following error messages may appear:

Errornumber 2	(if M is not an integer $\geqslant 0$)
Errornumber 3	(if N is not an integer $\geqslant 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
RUNPROB	STATAL 40125

Language
Algol 60

Method and performance
The distribution function is computed exactly, using the value of RUNPROB and a recurrent relation between successive probabilities.

The precision is 10^{-10}.

Reference

[1] A. Wald and J. Wolfowitz: *On a test whether two samples are from the same population,* Ann. Math. Stat., 11, 1949, p.147-162.

Example of use

```
"BEGIN"
   OUTPUT(61, "("3(Z.6D,/)")",
       RUN(4,  3,  4),
       RUN(9,  8,  6),
       RUN(8, 17, 22))
"END"
```

Output:

```
.542857
.820513
.000036
```

Source text

```
"CODE" 41023;
"REAL" "PROCEDURE" RUN(X, M, N);
"VALUE" X, M, N; "REAL" X, M, N;
"BEGIN"
    "REAL" P, PCUM; "INTEGER" IX, I, K, UP;
    "IF" M < 0 "OR" ENTIER(M) < M "THEN"
    STATAL3 ERROR("("RUN")", 2, M) "ELSE"
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("RUN")", 3, N) "ELSE"
    "IF" M > N "THEN" "BEGIN" K:= M; M:= N; N:= K "END";
    "IF" M = 0 "THEN" RUN:= ("IF" X < 1 "THEN" 0 "ELSE" 1)
    "ELSE" "IF" X < 2 "THEN" RUN:= 0 "ELSE"
    "IF" M = N "AND" X >= M * 2 "THEN" RUN:= 1 "ELSE"
    "IF" X > M * 2 "THEN" RUN:= 1 "ELSE"
    "BEGIN" IX:= ENTIER(X); "IF" IX // 2 * 2 < IX "THEN"
        "BEGIN" PCUM:= P:= RUNPROB(IX, M, N);
            K:= (IX - 1) / 2;
            P:= P * K * 2 / (M + N - K * 2); PCUM:= PCUM + P
        "END" "ELSE"
        "BEGIN" K:= IX / 2;
            P:= PCUM:= RUNPROB(IX, M, N)
        "END";
```

```
        "FOR" I:= K - 1 "STEP" -1 "UNTIL" 1 "DO"
        "BEGIN" P:= P * (M + N - I * 2) * I  / (N - I) /
            (M - I) / 2; PCUM:= PCUM + P;
            P:= P * 2 * I / (M + N - 2 * I); PCUM:= PCUM + P
        "END";
        RUN:= PCUM
    "END"
"END" RUN;
         "EOP"
```

TITLE: **Runinv**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750701

BRIEF DESCRIPTION

The procedure computes the left (right) hand tail inverse of the distribution of the test statistic of the run test, i.e. the largest (smallest) integer for which the left (right) hand tail probability of the distribution is less than or equal to a given value PROB. M and N are the sizes of the first and second sample respectively.

KEYWORDS

Inverse null-hypothesis distribution function of the run test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" RUNINV (PROB, M, N, LEFT);
"VALUE" PROB, M, N, LEFT;
"REAL" PROB, M, N;
"BOOLEAN" LEFT;
"CODE" 41024;
```

Formal parameters

PROB: <arithmetic expression>, tail probability of the value to be computed;

M: <arithmetic expression>, size of the first sample;

N: <arithmetic expression>, size of the second sample;

LEFT: <boolean expression>, indicating if either the "LEFT" hand tail inverse or the right hand tail inverse has to be computed. In the first (second) case LEFT should have the value "TRUE" ("FALSE").

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier RUNINV. If no value within the support can be computed the value 1 (if LEFT is "TRUE") or the value 2*MIN(M,N)+1 (if LEFT is "FALSE" and M=N) or the value 2*MIN(M,N)+2 (if LEFT is "FALSE" and M≠N) is assigned.

The following error messages may appear:

Errornumber 1 (if PROB ⩽ 0 or PROB ⩾ 1)

Errornumber 2 (if M is not an integer ⩾ 0)

Errornumber 3 (if N is not an integer ⩾ 0)

Procedures used

STATAL3 ERROR	STATAL 40100
RUN	STATAL 41023
RUNPROB	STATAL 41025
PHINV	STATAL 41501

Language
Algol 60

Method and performance
In both cases the inverse distribution function is computed, using the recurrent relation between successive probabilities of the run test statistic, starting with an estimated inverse obtained from a normal approximation.

The precision of the comparisons made is 10^{-10}.

Example of use
```
"BEGIN"
    "BOOLEAN" LEFT;
    LEFT:= "TRUE";
    OUTPUT(61, "("6(+ZD,/)")",
        RUNINV(.39, 4, 8, LEFT),
        RUNINV(.48, 3, 6, LEFT),
        RUNINV(.77, 5, 8, LEFT),
        RUNINV(.15, 7, 4, "NOT" LEFT),
        RUNINV(.92, 3, 5, "NOT" LEFT),
        RUNINV(.50, 6, 5, "NOT" LEFT))
"END"
```

Output:

```
+5
+4
+7
+9
+4
+7
```

SOURCE TEXT

```
"CODE" 41024;
"REAL" "PROCEDURE" RUNINV(PROB, M, N, LEFT);
"VALUE" PROB, M, N, LEFT; "REAL" PROB, M, N; "BOOLEAN" LEFT;
"BEGIN"
    "INTEGER" H, R, MN; "REAL" P, PCUM;
    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3 ERROR("("RUNINV")", 1, PROB) "ELSE"
    "IF" M < 0 "OR" ENTIER(M) < M "THEN"
    STATAL3 ERROR("("RUNINV")", 2, M) "ELSE"
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("RUNINV")", 3, N);
    MN:= M + N;
    "IF" M > N "THEN" "BEGIN" H:= M; M:= N; N:= H "END";
    "IF" M = 0 "THEN"
        RUNINV:= ("IF" LEFT "THEN" 0 "ELSE" 2) "ELSE"
    "IF" LEFT "THEN"
    "BEGIN" R:= PHINV(PROB) *
        SQRT((MN ** 3 - MN) / (2 * M * N *
        (2 * M * N - MN))) + .5 + 2 * M * N / MN;
        "IF" R < 2 "THEN" R:= 2 "ELSE"
        "IF" R > M * 2 "THEN" R:= M * 2;
        "IF" PROB < RUNPROB(2, M, N)
        "THEN" RUNINV:= +1 "ELSE"
        "BEGIN" PCUM:= RUN(R, M, N);
            "IF" PCUM <= PROB "THEN"
            "BEGIN" "FOR" P:= RUNPROB(R + 1, M, N)
                "WHILE" PCUM + P <= PROB "DO"
                "BEGIN" R:= R + 1; PCUM:= PCUM + P "END"
            "END" "ELSE"
            "BEGIN" "FOR" P:= RUNPROB(R, M, N)
                "WHILE" PCUM - P > PROB "DO"
                "BEGIN" R:= R - 1; PCUM:= PCUM - P "END";
                R:= R - 1
            "END"; RUNINV:= R
        "END"
    "END" "ELSE"
    "BEGIN" R:= PHINV(1 - PROB) *
        SQRT((MN ** 3 - MN) / (2 * M * N *
        (2 * M * N - MN))) + 1.5 + 2 * M * N / MN;
        "IF" R < 2 "THEN" R:= 2 "ELSE"
        "IF" R > M * 2 "THEN" R:= M * 2;
        MN:= "IF" M = N "THEN" 2 * M "ELSE" 2 * M + 1;
        "IF" PROB < RUNPROB(MN, M, N)
        "THEN" RUNINV:= MN + 1 "ELSE"
        "BEGIN" PCUM:= 1 - RUN(R - 1, M, N);
            "IF" PCUM <= PROB "THEN"
            "BEGIN" "FOR" P:= RUNPROB(R - 1, M, N)
                "WHILE" PCUM + P <= PROB "DO"
                "BEGIN" R:= R - 1; PCUM:= PCUM + P "END"
            "END" "ELSE"
            "BEGIN" "FOR" P:= RUNPROB(R, M, N)
                "WHILE" PCUM - P > PROB "DO"
```

```
            "BEGIN" R:= R + 1; PCUM:= PCUM - P "END";
            R:= R + 1
        "END";
        RUNINV:= R
    "END"
  "END"
"END" RUNINV;
      "EOP"
```

TITLE: **Runprob**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750701

BRIEF DESCRIPTION

The procedure computes the probability function of the test statistic of the two sample run test, i.e. the probability that the value of the test statistic is equal to a given value X. The test statistic is equal to the number of runs in the ordered joint sample. M and N are the sizes of the first and second sample, respectively. The procedure computes the distribution function under the null-hypothesis that the two samples are from the same (continuous) distribution. (Thus it is assumed that no equal observations occur in the joined sample).

KEYWORDS

Null-hypothesis probability function of the two sample run test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" RUNPROB (X, M, N);
"VALUE" X, M, N;
"REAL" X, M, N;
"CODE" 41025;
```

Formal parameters

X: <arithmetic expression>, argument of the probability function;
M: <arithmetic expression>, size of the first sample;
N: <arithmetic expression>, size of the second sample.

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier RUNPROB.

The following error messages may appear:

Errornumber 2	(if M is not an integer $\geqslant 0$)
Errornumber 3	(if N is not an integer $\geqslant 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The probability function is computed exactly, using the procedure LOGGAMMA for the computation of LN(N!).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        RUNPROB( 5, 3,  4),
        RUNPROB(11, 7,  7),
        RUNPROB(14, 8, 12))
"END"
```

Output:

```
 .257143
 .052448
 .051346
```

SOURCE TEXT

```
"CODE" 41025;
"REAL" "PROCEDURE" RUNPROB(X, M, N); "VALUE" X, M, N;
"REAL" X, M, N;
"BEGIN" "INTEGER" K; "REAL" P; "BOOLEAN" EVEN;

    "IF" M < 0 "OR" ENTIER(M) < M "THEN"
    STATAL3 ERROR("("RUNPROB")", 2, M) "ELSE"
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("RUNPROB")", 3, N);
    "IF" M > N "THEN" "BEGIN" K:= N; N:= M; M:= K "END";
    EVEN:= ENTIER(X / 2) * 2 = X;
    K:= "IF" EVEN "THEN" X / 2 "ELSE" (X + 1) / 2;
    RUNPROB:=
    "IF" N = 0 "THEN" 0 "ELSE"
    "IF" M = 0 "THEN" ("IF" X = 1 "THEN" 1 "ELSE" 0) "ELSE"
    "IF" X < 2 "OR" X > 2 * M + 1 "OR" ENTIER(X) < X
    "THEN" 0 "ELSE"
    "IF" EVEN "THEN"
    2 * M * N * EXP(2 * (LOGGAMMA(M) + LOGGAMMA(N) -
    LOGGAMMA(K)) - LOGGAMMA(M - K + 1) - LOGGAMMA(N - K + 1)
    - LOGGAMMA(M + N + 1)) "ELSE"
    (M + N - X + 1) * M * N / (K - 1) * EXP(2 * (LOGGAMMA(M)
    + LOGGAMMA(N) - LOGGAMMA(K - 1)) - LOGGAMMA(M + N + 1)
    - LOGGAMMA(M - K + 2) - LOGGAMMA(N - K + 2))
"END" RUNPROB;
        "EOP"
```

TITLE: **Kendall**

AUTHORS: J. Bethlehem, J.M. Buhrman

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the distribution function of the test statistic S of Kendall's test of independence under the null-hypothesis, i.e. the probability that the value of S is less than or equal to a given value X. N is the number of pairs of observations (X[1],Y[1]),...,(X[N],Y[N]). The test statistic S is equal to the sum of SIGN((X[I]-X[J])*(Y[I]-Y[J])); I,J=1,...,N. The assumption is made that no equal values occur among the X[I] or the Y[J].

KEYWORDS

Null-hypothesis distribution function of Kendall's test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" KENDALL (X, N);
"VALUE" X, N;
"REAL" X, N;
"CODE" 41026;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

N: <arithmetic expression>, number of pairs of observations.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier KENDALL.

The following error message may appear:

Errornumber 2 (if N is not an integer $\geqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
KENDALLPROB	STATAL 41028
PHI	STATAL 41500

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed exactly as the sum of probabilities, using KENDALLPROB, for N ⩽ 9, and is approximated by a normal distribution with continuity correction for N > 9.
For N ⩽ 9 the computation is exact but slow (approximately proportional to X*N!); for N > 9 the computation is fast but the precision is $5*10^{-3}$.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        KENDALL( 2, 5),
        KENDALL(-1, 3),
        KENDALL(12, 7))
"END"
```

Output:

```
 .758333
 .500000
 .965476
```

SOURCE TEXT

```
"CODE" 41026;
"REAL" "PROCEDURE" KENDALL(X, N); "VALUE" X, N; "REAL" X, N;
"BEGIN"
    "INTEGER" I, G, IX; "REAL" P;
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("KENDALL")", 2, N);
    G:= N * (N - 1) / 2; IX:= G + ENTIER(-(G - X) / 2) * 2;
    "IF" IX >= G "THEN" KENDALL:= 1 "ELSE"
    "IF" IX < -G "THEN" KENDALL:= 0 "ELSE"
    "IF" N > 9 "THEN"
    KENDALL:= PHI(IX + 1 / SQRT(N * (N - 1) * (N+N+5) / 18))
    "ELSE" "IF" IX > 0 "THEN"
    "BEGIN" P:= 0; "FOR" I:= G "STEP" -2 "UNTIL" IX + 2 "DO"
        P:= P + KENDALLPROB(I, N); KENDALL:= 1 - P
    "END" "ELSE"
    "BEGIN" P:= 0; "FOR" I:= -G "STEP" 2 "UNTIL" IX "DO"
        P:= P + KENDALLPROB(I, N); KENDALL:= P
    "END"
"END" KENDALL;
         "EOP"
```

TITLE: **Kendallinv**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750701

BRIEF DESCRIPTION

The procedure computes the left (right) hand tail inverse of the distribution of Kendall's test statistic, i.e. the largest (smallest) integer with the same parity as the values with positive probability for which the left (right) hand tail probability is less than or equal to a given value PROB. N is the number of pairs of observations.

KEYWORDS

Inverse null-hypothesis distribution function of Kendall's test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" KENDALLINV (PROB, N, LEFT);
"VALUE" PROB, N, LEFT;
"REAL" PROB, N;
"BOOLEAN" LEFT;
"CODE" 41027;
```

Formal parameters

PROB: <arithmetic expression>, tail probability of the value to be computed;

N: <arithmetic expression>, number of pairs of observations;

LEFT: <boolean expression>, indicating if either the left hand tail inverse or the right hand tail inverse has to be computed. In the first (second) case LEFT should have the value "TRUE" ("FALSE").

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier KENDALLINV. If no value within the support can be computed, the value $-\frac{1}{2}N(N-1)-2$ (if LEFT is "TRUE") or the value $\frac{1}{2}N(N-1)+2$ (if LEFT is "FALSE") is assigned.

The following error messages may appear:

Errornumber 1 (if $PROB \leq 0$ or $PROB \geq 1$)

Errornumber 2 (if N is not an integer ≥ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
KENDALL	STATAL 41026
KENDALLPROB	STATAL 41028
PHINV	STATAL 41501

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
In both cases the inverse distribution function is computed exactly, starting with an estimated inverse obtained from a normal approximation. The computing time is approximately proportional to ABS(PHINV(PROB))*N^2.

The precision of the comparisons made is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    "BOOLEAN" LEFT;
    LEFT:= "TRUE";
    OUTPUT(61, "("6(+ZD,/)")",
        KENDALLINV(.93, 4, LEFT),
        KENDALLINV(.84, 6, LEFT),
        KENDALLINV(.77, 6, LEFT),
        KENDALLINV(.51, 4, "NOT" LEFT),
        KENDALLINV(.29, 5, "NOT" LEFT),
        KENDALLINV(.05, 6, "NOT" LEFT))
"END"
```

Output:

```
 +2
 +3
 +3
 +2
 +4
+11
```

SOURCE TEXT

```
"CODE" 41027;
"REAL" "PROCEDURE" KENDALLINV(PROB, N, LEFT);
"VALUE" PROB, N, LEFT; "REAL" PROB, N; "BOOLEAN" LEFT;
"BEGIN" "INTEGER" S, G; "REAL" P, PCUM;

    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3 ERROR("("KENDALLINV")", 1, PROB) "ELSE"
    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("KENDALLINV")", 2, N);
    G:= N * (N - 1) / 2;
    "IF" N = 0 "THEN"
        KENDALLINV:= ("IF" LEFT "THEN" -1 "ELSE" +1)
    "ELSE" "IF" LEFT "THEN"
    "BEGIN" S:= PHINV(PROB) * SQRT(G * (N * 2 + 5) / 9);
        S:= "IF" ABS(S) > G "THEN" G * SIGN(S) "ELSE"
        G + ENTIER(-(G - S) / 2) * 2;
        "IF" PROB < KENDALLPROB(-G, N) "THEN"
            KENDALLINV:= -G - 2
        "ELSE"
        "BEGIN" PCUM:= KENDALL(S, N);
            "IF" PCUM <= PROB "THEN"
            "BEGIN" "FOR" P:= KENDALLPROB(S + 2, N)
                "WHILE" PCUM + P <= PROB "DO"
                "BEGIN" S:= S + 2; PCUM:= PCUM + P "END"
            "END" "ELSE"
            "BEGIN" "FOR" P:= KENDALLPROB(S, N)
                "WHILE" PCUM - P > PROB "DO"
                "BEGIN" S:= S - 2; PCUM:= PCUM - P "END";
                S:= S - 2
            "END"; KENDALLINV:= S
        "END"
    "END" "ELSE"
    "BEGIN" S:= PHINV(1 - PROB) * SQRT(G * (N * 2 + 5) / 9);
        S:= "IF" ABS(S) > G "THEN" G * SIGN(S) "ELSE"
        G - ENTIER((G - S) / 2) * 2;
        "IF" PROB < KENDALLPROB(G, N) "THEN"
        KENDALLINV:= G + 2 "ELSE"
        "BEGIN" PCUM:= 1 - KENDALL(S - 2, N);
            "IF" PCUM <= PROB "THEN"
            "BEGIN" "FOR" P:= KENDALLPROB(S - 2, N)
                "WHILE" PCUM + P <= PROB "DO"
                "BEGIN" S:= S - 2; PCUM:= PCUM + P "END"
            "END" "ELSE"
            "BEGIN" "FOR" P:= KENDALLPROB(S, N)
                "WHILE" PCUM - P > PROB "DO"
                "BEGIN" S:= S + 2; PCUM:= PCUM - P "END";
                S:= S + 2
            "END";
            KENDALLINV:= S
        "END"
    "END"
"END" KENDALLINV;
        "EOP"
```

TITLE: **Kendallprob**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the probability function of the test statistic S of Kendall's test of independence under the null-hypothesis, i.e. the probability that the value of S is equal to a given value X. N is the number of pairs of observations (X[1],Y[1]),...,(X[N],Y[N]). The test statistic S is equal to the sum of SIGN((X[I]-X[J])*(Y[I]-Y[J])); I,J=1,...,N. The assumption is made that no equal values occur among the X[I] or the Y[J].

KEYWORDS

Null-hypothesis probability function of Kendall's test statistic

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" KENDALLPROB (X, N);
"VALUE" X, N;
"REAL" X, N;
"CODE" 41028;
```

Formal parameters

X: <arithmetic expression>, argument of the probability function;

N: <arithmetic expression>, number of pairs of observations.

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier KENDALLPROB.

The following error message may appear:

Errornumber 2 (if N is not an integer $\geqslant 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 41500

LANGUAGE

Algol 60

Method and performance

The probability function is computed exactly, using a recurrent relation, for `N` ⩽ `9`, and is approximated by a normal distribution with continuity correction for `N` > `9`. For `N` ⩽ `9` the computation is exact but slow (approximately proportional to `N!`); for `N` > `9` the computation is fast, but the precision is $5*10^{-3}$.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        KENDALLPROB( 3,  6),
        KENDALLPROB(-1,  7),
        KENDALLPROB(25, 10))
"END"
```

Output:

```
 .125000
 .113690
 .000000
```

Source text

```
"CODE" 41028;
"REAL" "PROCEDURE" KENDALLPROB(X, N);
"VALUE" X, N; "REAL" X, N;
"BEGIN" "INTEGER" G, IX;

    "REAL" "PROCEDURE" PROB(S, N);
    "VALUE" S, N; "INTEGER" S, N;
    "BEGIN" "INTEGER" I; "REAL" P;
        "IF" N = 2 "THEN" PROB:= ("IF" ABS(S) = 1 "THEN" .5
        "ELSE" 0) "ELSE"
        "BEGIN" P:= 0;
            "FOR" I:= 0 "STEP" 1 "UNTIL" N - 1 "DO"
            P:= P + PROB(S - N + 1 + I * 2, N - 1);
            PROB:= P / N
        "END"
    "END";

    "IF" N < 0 "OR" ENTIER(N) < N "THEN"
    STATAL3 ERROR("("KENDALLPROB")", 2, N);
    G:= N * (N - 1) / 2; IX:= ENTIER(X);
    "IF" N > 9 "THEN"
    "BEGIN" "REAL" S; S:= SQRT(N * (N-1) * (N+N+5) / 18);
        KENDALLPROB:= PHI(IX + 1/S) - PHI(IX - 1/S)
    "END" "ELSE"
    KENDALLPROB:= "IF" IX < X "OR" ABS(IX) > G "OR"
    (G - IX) // 2 * 2 < G - IX "THEN" 0 "ELSE" PROB(IX, N)
"END" KENDALLPROB;
        "EOP"
```

TITLE: **Mulnomprob**

AUTHORS: J.M. Buhrman, R. Kaas, I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the multinomial probability function, i.e. the probability that a random vector with a multinomial distribution with parameters N, P[1],...,P[K] is equal to X[1],...,X[K]. N denotes the number of independent experiments performed and P[1],...,P[K] the respective probabilities of the events 1,...,K. The result of each experiment is one of these events.

KEYWORDS

Multinomial probability function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" MULNOMPROB (X, N, K, P);
"VALUE" N, K;
"REAL" N, K;
"ARRAY" X, P;
"CODE" 41255;
```

Formal parameters

X: <arithmetic expression>, argument of the probability function;
N: <arithmetic expression>, number of experiments;
K: <arithmetic expression>, number of events;
P: <array identifier>, vector containing the probabilities.

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier MULNOMPROB.

The following error messages may appear:

Errornumber 2	(if N is not an integer > 0)
Errornumber 3	(if K is not an integer > 1)
Errornumber 4	(if $\sum$P[I]$\neq$1, or $\sum$P[I]<0).

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The probability is computed as follows:
MULNOMPROB(X,N,K,P)=

$$\begin{cases} 0 & \text{if } \sum X[I] \neq N, \\ 0 & \text{if } X[I] \text{ is not a non-negative integer for some } 1 \leq I \leq K, \\ 0 & \text{if } P[I]=0 \text{ and } X[I]>0 \text{ for some } 1 \leq I \leq K, \\ EXP(LOGGAMMA(N+1)+\sum_{I=1}^{K} (X[I]*LN(P[I])/LOGGAMMA(X[I]+1))) & \\ & \text{in all other cases.} \end{cases}$$

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    "REAL" "ARRAY" X, Z, P1[1:3], Y, P2[1:4];
    INARRAY(60, X); INARRAY(60, P1); INARRAY(60, Y);
    INARRAY(60, P2); INARRAY(60, Z);
    OUTPUT(61, "("3(Z.6D,/)")",
        MULNOMPROB(X, 10, 3, P1),
        MULNOMPROB(Y,  6, 4, P2),
        MULNOMPROB(Z,  6, 3, P1))
"END"
```

Input:

```
 3   5   2
.2  .6  .2
 0   4   1   1
.1  .4  .4  .1
 5   3   2
```

Output:

```
.062706
.030720
.000000
```

SOURCE TEXT

```
"CODE" 41255;
"REAL" "PROCEDURE" MULNOMPROB(XVEC, N, K, PVEC);
    "VALUE" N,K; "REAL" N,K; "ARRAY" XVEC, PVEC;
"BEGIN" "REAL" XL, PL, LNPR, EPS, PSUM;
    "INTEGER" J, XSUM;

    "IF" N > ENTIER(N) "OR" N < 1 "THEN"
    STATAL3 ERROR("("MULNOMPROB")",2,N) "ELSE"
    "IF" K > ENTIER(K) "OR" K < 2 "THEN"
    STATAL3 ERROR("("MULNOMPROB")",3,K) "ELSE"
    "BEGIN" PSUM := 0; XSUM := 0; EPS := "-14;
        "FOR" J := 1 "STEP" 1 "UNTIL" K "DO"
        "BEGIN" PL := PVEC[J];
            "IF" PL < 0 "THEN"
            STATAL3 ERROR("("MULNOMPROB")",4,PL) "ELSE"
              PSUM := PSUM + PL
        "END";
        "IF" ABS(PSUM-1) > EPS "THEN"
        STATAL3 ERROR("("MULNOMPROB")",4,PSUM) "ELSE"
        "BEGIN" "FOR" J := 1 "STEP" 1 "UNTIL" K "DO"
            "BEGIN" XL := XVEC[J];
                "IF" XL > ENTIER(XL) "OR" XL < 0 "THEN"
                "BEGIN" MULNOMPROB := 0; "GOTO" OUT "END";
                XSUM := XSUM + XL;
            "END";
            "IF" XSUM ^= N "THEN"
            "BEGIN" MULNOMPROB := 0; "GOTO" OUT "END" "ELSE"
            "BEGIN" LNPR := LOGGAMMA(N+1);
                "FOR" J := 1 "STEP" 1 "UNTIL" K "DO"
                "IF" PVEC[J] = 0 "THEN"
                "BEGIN" "IF" XVEC[J] ^= 0 "THEN"
                    "BEGIN" MULNOMPROB := 0;
                        "GOTO" OUT
                    "END"
                "END" "ELSE"
                LNPR := LNPR - LOGGAMMA(XVEC[J]+1) +
                        XVEC[J] * LN(PVEC[J]);
                MULNOMPROB := EXP(LNPR);
            "END";
        "END";
    "END"; OUT:
"END" MULNOMPROB;
        "EOP"
```

TITLE: **Mulhypergprob**

AUTHOR: J.M. Buhrman

INSTITUTE: Mathematical Centre

RECEIVED: 780601

BRIEF DESCRIPTION

The procedure computes the multihypergeometric probability function, i.e. the probability that in a random sample of size N drawn without replacement from a population consisting of R[1] elements of type 1,...,R[K] elements of type K, occur X[1] elements of type 1,...,X[K] elements of type K.

KEYWORDS

Multihypergeometric probability function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" MULHYPERGPROB (X, N, K, R);
"VALUE" N, K;
"REAL" N, K;
"ARRAY" X, R;
"CODE" 41256;
```

Formal parameters

X: <array identifier>, argument of the probability function;

N: <arithmetic expression>, sample size;

K: <arithmetic expression>, number of different types of elements in the population;

R: <array identifier>, vector containing the numbers of elements of each type in the population .

DATA AND RESULTS

The value of the probability function is assigned to the procedure identifier MULHYPERGPROB.

The following error messages may appear:

Errornumber 2	(if N is not an integer > 0)
Errornumber 3	(if K is not an integer > 0)
Errornumber 4	(if some R[I] is not an integer ⩾ 0).

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

Language

Algol 60

Method and performance

The probability is computed as follows:

MULHYPERGPROB(X,N,K,R)=

$$\begin{cases} 0 & \text{if } \sum_{I=1}^{K} X[I] \neq N, \\ 0 & \text{if } X[I] \text{ is not a non-negative integer for some } 1 \leq I \leq K, \\ 0 & \text{if } X[I] > R[I] \text{ for some } 1 \leq I \leq K, \\ \text{EXP(LOGGAMMA(N+1)} & \\ \text{+LOGGAMMA(M-N+1)} & \\ \text{-LOGGAMMA(M+1)} & \\ +\sum_{I=1}^{K} \text{(LOGGAMMA(R[I]+1)} & \\ \text{-LOGGAMMA(X[I]+1)} & \\ \text{-LOGGAMMA(R[I]-X[I]+1)))} & \text{in all other cases.} \end{cases}$$

where $M=\Sigma R[I]$ is the number of elements in the population.

The precision is 10^{-10}.

Example of use

Program:

```
"BEGIN"
    "REAL" "ARRAY" Y1, Y2, S1[1:3], Y3, S2[1:4];
    INARRAY(60, Y1); INARRAY(60, Y2); INARRAY(60, S1);
    INARRAY(60, Y3); INARRAY(60, S2);
    OUTPUT(61, "("3(Z.6D,/)")",
        MULHYPERGPROB(Y1, 7, 3, S1),
        MULHYPERGPROB(Y2, 8, 3, S1),
        MULHYPERGPROB(Y3, 6, 4, S2))
"END"
```

Input:

```
 4  2  1
 2  3  3
13 14 14
 0  2  2  2
 4  4  4  4
```

Output:

```
.040517
.108162
.026973
```

SOURCE TEXT

```
"CODE" 41256;
"REAL" "PROCEDURE" MULHYPERGPROB(X,N,K,R);
"VALUE" N,K; "REAL" N,K; "ARRAY" X,R;
"BEGIN" "INTEGER" I,J,L,SR,SX; "REAL" MHP,XJ,RJ;
    "IF" N < 1 "OR" N > ENTIER(N) "THEN"
        STATAL3 ERROR("("MULHYPERGPROB")",2,N);
    "IF" K < 1 "OR" K > ENTIER(K) "THEN"
        STATAL3 ERROR("("MULHYPERGPROB")",3,K);
    SX := SR := 0;
    MHP := 0;
    "FOR" J := 1 "STEP" 1 "UNTIL" K  "DO"
    "BEGIN"  RJ := R[J]; XJ := X[J];
        "IF" RJ < 0 "OR" RJ > ENTIER(RJ) "THEN"
            STATAL3 ERROR("("MULHYPERGPROB")",4,RJ);
        SR := SR + RJ; SX := SX + XJ;
        "IF" XJ > RJ "OR" XJ < 0 "OR" XJ > ENTIER(XJ) "THEN"
        "BEGIN" MULHYPERGPROB := 0; "GOTO" OUT "END";
        MHP := MHP + LOGGAMMA(RJ+1) - LOGGAMMA(XJ+1)
        - LOGGAMMA(RJ-XJ+1);
    "END" J ;
    "IF" SX ^= N "THEN"
    "BEGIN" MULHYPERGPROB := 0; "GOTO" OUT "END";
    MHP := MHP + LOGGAMMA(N+1) + LOGGAMMA(SR-N+1)
          - LOGGAMMA(SR+1);
    MULHYPERGPROB := EXP(MHP);
  OUT:
"END" MULHYPERGPROB;
        "EOP"
```

1.2 CONTINUOUS DISTRIBUTIONS

This section contains procedures for computing the distribution function, the inverse distribution function, and the density function of continuous distributions. The distribution and density functions are defined for all real values of the argument. Calling an inverse distribution function with a value of the argument PROB smaller than 10^{-14} or larger than $1-10^{-14}$ causes an error message. It is anyway advised not to call the inverse distribution function with a value of the argument too close to 1 or too close to 0, say $< 10^{-10}$ or $> 1-10^{-10}$.

We aimed for a precision of 10^{-10} in the computation of the procedures. In some cases it was not possible to obtain this precision.

TITLE: **Uniform**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750515

BRIEF DESCRIPTION
The procedure computes the uniform distribution function, i.e. the probability that a random variable with a uniform (A,B) distribution is less than or equal to a given value X. The parameters A and B are the lower and the upper bound of the range.

KEYWORDS
Uniform distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" UNIFORM (X, A, B);
"VALUE" X, A, B;
"REAL" X, A, B;
"CODE" 41567;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
A: <arithmetic expression>, lower bound of the range;
B: <arithmetic expression>, upper bound of the range.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier UNIFORM.
The following error message may appear:
Errornumber 2 (if $B \leqslant A$)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed as follows:
UNIFORM(X,A,B)=

$$\begin{cases} 0 & \text{if } X<A, \\ (X-A)/(B-A) & \text{if } A\leqslant X<B, \\ 1 & \text{if } X\geqslant B. \end{cases}$$

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        UNIFORM( 2.5, 2.1,  4.3),
        UNIFORM(1.96,   0, 2.45),
        UNIFORM( 150, 105,  218))
"END"
```

Output:

```
 .181818
 .800000
 .398230
```

SOURCE TEXT

```
"CODE" 41567;
"REAL" "PROCEDURE" UNIFORM(X, A, B);
"VALUE" X, A, B; "REAL" X, A, B;
"BEGIN"
    "IF" B <= A "THEN"
        STATAL3ERROR("("UNIFORM")", 2, B);
    UNIFORM:= "IF" X <= A "THEN" 0 "ELSE"
        "IF" X >= B "THEN" 1 "ELSE" (X - A) / (B - A)
"END" UNIFORM;
          "EOP"
```

TITLE: **Uniforminv**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750515

BRIEF DESCRIPTION

The procedure computes the argument X, for which the uniform (A,B) distribution function has a given value PROB. The parameters A and B are the lower and the upper bound of the range.

KEYWORDS

Inverse uniform distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" UNIFORMINV (PROB, A, B);
"VALUE" PROB, A, B;
"REAL" PROB, A, B;
"CODE" 41568;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

A: <arithmetic expression>, lower bound of the range;

B: <arithmetic expression>, upper bound of the range.

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier UNIFORMINV.

The following error messages may appear:

Errornumber 1	(if PROB $\leqslant 0$ or PROB $\geqslant 1$)
Errornumber 2	(if B $\leqslant$ A)

PROCEDURES USED

STATAL3 ERROR   STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The inverse distribution function is computed as follows:

UNIFORMINV(PROB,A,B)=A+PROB*(B−A).

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        UNIFORMINV(.250, -5.1, 1.08),
        UNIFORMINV( 2/3, 2.97, 5.03),
        UNIFORMINV(.514, 3.43, 6.86))
"END"
```

Output:

```
 -3.555000
 +4.343333
 +5.193020
```

SOURCE TEXT

```
"CODE" 41568;
"REAL" "PROCEDURE" UNIFORMINV(PROB, A, B);
"VALUE" PROB, A, B; "REAL" PROB, A, B;
"BEGIN"
    "IF" B <= A "THEN" STATAL3ERROR("("UNIFORMINV")", 2, B);
    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
    STATAL3ERROR("("UNIFORMINV")", 1, PROB);
    UNIFORMINV:= (B - A) * PROB + A
"END" UNIFORMINV;
          "EOP"
```

TITLE: **Uniformdens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750515

BRIEF DESCRIPTION

The procedure computes the density function of the uniform (A,B) distribution for a given argument X. The parameters A and B are the lower and the upper bound of the range.

KEYWORDS

Uniform density function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" UNIFORMDENS (X, A, B);
"VALUE" X, A, B;
"REAL" X, A, B;
"CODE" 41751;
```

Formal parameters

X: <arithmetic expression>, argument of the density function;
A: <arithmetic expression>, lower bound of the range;
B: <arithmetic expression>, upper bound of the range.

DATA AND RESULTS

The value of the density function is assigned to the procedure identifier UNIFORMDENS

The following error message may appear:

Errornumber 2 (if $B \leq A$)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The inverse function is computed ad follows:

UNIFORMDENS(X,A,B)=

$$\begin{cases} 1/(B-A) & \text{if } A<X\leq B, \\ 0 & \text{if } X\leq A \text{ or } X>B. \end{cases}$$

The precision is 10^{-14}.

EXAMPLE OF USE

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        UNIFORMDENS( .5,    0,    1),
        UNIFORMDENS(.543,   .2,   .7),
        UNIFORMDENS(.388, .309, .471))
"END"
```

Output:

```
1.000000
2.000000
6.172840
```

SOURCE TEXT

```
"CODE" 41751;
"REAL" "PROCEDURE" UNIFORMDENS(X, A, B);
"VALUE" X, A, B; "REAL" X,A,B;
"BEGIN"
    "IF" B <= A
    "THEN" STATAL3ERROR("("UNIFORMDENS")", 2, B);
    UNIFORMDENS:= "IF" X <= A "OR" X > B
    "THEN" 0 "ELSE" 1 / (B - A)
"END" UNIFORMDENS;
         "EOP"
```

TITLE: **Phi**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 740101

BRIEF DESCRIPTION

The procedure computes the standard nomal distribution function, i.e. the probability that a random variable with a standard normal distribution is less than or equal to a given value X.

KEYWORDS

Standard normal distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" PHI (X);
"VALUE" X;
"REAL" X;
"CODE" 41500;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier PHI.

PROCEDURES USED

None.

LANGUAGE

Compass

METHOD AND PERFORMANCE

The distribution function is computed using rational Chebyshev approximations.

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        PHI(-1.96),
        PHI( 0.00),
        PHI( 1.68))
"END"
```

Output:

```
 .024998
 .500000
 .953521
```

Source text

The procedure is written in COMPASS; an equivalent ALGOL 60 text is given.

```
"CODE" 41500;
"REAL" "PROCEDURE" PHI(X); "VALUE" X; "REAL" X;
"BEGIN" "REAL" ABSX, ERF, ERFC, C, P, Q;
    X:= X * .70710 67811 8655; ABSX:= ABS(X);
    "IF" X > 5.5 "THEN" PHI:= 1 "ELSE" "IF" X < -5.5
    "THEN" PHI:= 0 "ELSE" "IF" ABSX <= 0.5 "THEN"
    "BEGIN" C:= X * X;
        P:= ((-0.35609 84370 1815"-1  * C +
        0.69963 83488 6191"+1) * C + 0.21979 26161 8294"+2)
        * C + 0.24266 79552 3053"+3;
        Q:= ((C +
        0.15082 79763 0408"+2) * C +
        0.91164 90540 4515"+2) * C +
        0.21505 88758 6986"+3;
        PHI:= .5 * X * P / Q + .5
    "END" "ELSE"
    "BEGIN" "IF" ABSX < 4 "THEN"
        "BEGIN" C:= ABSX;
            P:= ((((((-0.13686 48573 8272"-6  * C +
            0.56419 55174 7897"+0) * C +
            0.72117 58250 8831"+1) * C +
            0.43162 22722 2057"+2) * C +
            0.15298 92850 4694"+3) * C +
            0.33932 08167 3434"+3) * C +
            0.45191 89537 1187"+3) * C +
            0.30045 92610 2016"+3;
            Q:= ((((((C +
            0.12782 72731 9629"+2) * C +
            0.77000 15293 5230"+2) * C +
            0.27758 54447 4399"+3) * C +
            0.63898 02644 6563"+3) * C +
            0.93135 40948 5061"+3) * C +
            0.79095 09253 2790"+3) * C +
            0.30045 92609 5698"+3;
```

```
            C:= P / Q
        "END" "ELSE"
        "BEGIN" C:= 1 / X / X;
            P:= (((0.22319 24597 3419"-1  * C +
            0.27866 13086 0965"-0) * C +
            0.22695 65935 3969"-0) * C +
            0.49473 09106 2325"-1) * C +
            0.29961 07077 0354"-2;
            Q:= (((C +
            0.19873 32018 1714"+1) * C +
            0.10516 75107 0679"+1) * C +
            0.19130 89261 0783"+0) * C +
            0.10620 92305 2847"-1;
            C:= (C * (-P) / Q + 0.56418 95835 4776) / ABSX
        "END";
        PHI:= .5 + .5 * SIGN(X) * (1 - C * EXP(-X * X))
    "END"
"END" PHI;
"EOP"
```

TITLE: **Phinv**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION
The procedure computes the argument X, for which the standard normal distribution function has a given value PROB.

KEYWORDS
Inverse standard normal distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" PHINV (PROB);
"VALUE" PROB;
"REAL" PROB;
"CODE" 41501;
```
Formal parameters
PROB: <arithmetic expression>, left hand tail probability of the value to be computed.

DATA AND RESULTS
The value of the inverse distribution function is assigned to the procedure identifier PHINV.
The following error message may appear:
Errornumber 1 (if PROB $\leq 10^{-14}$ or PROB $\geq 1-10^{-14}$).

PROCEDURES USED
STATAL3ERROR STATAL 40100

LANGUAGE
Compass

METHOD AND PERFORMANCE
The inverse distribution function is computed using rational Chebyshev approximations.

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        PHINV(.900),
        PHINV(.025),
        PHINV(.950))
"END"
```

Output:

```
 +1.281552
 -1.959964
 +1.644854
```

Source text

The procedure is written in COMPASS; an equivalent ALGOL 60 text is given.

```
"CODE" 41501;
"REAL" "PROCEDURE" PHINV(PROB); "VALUE" PROB; "REAL" PROB;
"BEGIN" "REAL" EPS;
    "REAL" "PROCEDURE" INVERF(X); "VALUE" X; "REAL" X;
    "BEGIN" "REAL" ABSX, P, BETAX;
        "REAL" "ARRAY" A[0 : 23];
        "REAL" "PROCEDURE" CHEPOLSER(N, X, A);
        "VALUE" N, X; "INTEGER" N; "REAL" X; "ARRAY" A;
        "BEGIN" "INTEGER" K; "REAL" H, R, S, TX;
            TX:= X + X; R:= A[N];
            H:= A[N - 1] + R * TX;
            "FOR" K:= N - 2 "STEP" -1 "UNTIL" 1 "DO"
            "BEGIN" S:= R; R:= H;
                H:= A[K] + R * TX - S
            "END";
            CHEPOLSER:= A[0]- R + H * X
        "END" CHEPOLSER;

        ABSX:= ABS(X);
        "IF" ABSX <= 0.8 "THEN"
        "BEGIN"
            A[ 0]:= 0.99288 53766 1894;
            A[ 1]:= 0.12046 75161 4310;
            A[ 2]:= 0.01607 81993 4210;
            A[ 3]:= 0.00268 67044 3716;
            A[ 4]:= 0.00049 96347 3024;
            A[ 5]:= 0.00009 88982 1860;
            A[ 6]:= 0.00002 03918 1276;
            A[ 7]:= 0.00000 43272 7162;
            A[ 8]:= 0.00000 09380 8141;
            A[ 9]:= 0.00000 02067 3472;
            A[10]:= 0.00000 00461 5970;
            A[11]:= 0.00000 00104 1668;
            A[12]:= 0.00000 00023 7150;
```

```
        A[13]:= 0.00000 00005 4393;
        A[14]:= 0.00000 00001 2555;
        A[15]:= 0.00000 00000 2914;
        A[16]:= 0.00000 00000 0680;
        A[17]:= 0.00000 00000 0159;
        A[18]:= 0.00000 00000 0037;
        A[19]:= 0.00000 00000 0009;
        A[20]:= 0.00000 00000 0002;
        A[21]:= 0.00000 00000 0001;
        INVERF:= CHEPOLSER(21, X * X / 0.32 - 1, A) * X
    "END" "ELSE"
    "IF" 1 - ABSX >= 25"-4 "THEN"
    "BEGIN"
        A[ 0]:=  0.91215 88034 1755;
        A[ 1]:= -0.01626 62818 6766;
        A[ 2]:=  0.00043 35564 7295;
        A[ 3]:=  0.00021 44385 7007;
        A[ 4]:=  0.00000 26257 5108;
        A[ 5]:= -0.00000 30210 9105;
        A[ 6]:= -0.00000 00124 0606;
        A[ 7]:=  0.00000 00624 0661;
        A[ 8]:= -0.00000 00005 4013;
        A[ 9]:= -0.00000 00014 2321;
        A[10]:=  0.00000 00000 3438;
        A[11]:=  0.00000 00000 3358;
        A[12]:= -0.00000 00000 0146;
        A[13]:= -0.00000 00000 0081;
        A[14]:=  0.00000 00000 0005;
        A[15]:=  0.00000 00000 0002;
        BETAX:= SQRT(- LN((1 + ABSX) * (1 - ABSX)));
        P:= -1.54881 30423 7326 * BETAX +
            2.56549 01231 4782;
        P:= CHEPOLSER(15, P, A);
        INVERF:= "IF" X < 0 "THEN" - BETAX * P
            "ELSE" BETAX * P
    "END" "ELSE"
    "BEGIN"
        A[ 0]:=  0.95667 97090 2049;
        A[ 1]:= -0.02310 70043 0907;
        A[ 2]:= -0.00437 42360 9751;
        A[ 3]:= -0.00057 65034 2265;
        A[ 4]:= -0.00001 09610 2231;
        A[ 5]:=  0.00002 51085 4703;
        A[ 6]:=  0.00001 05623 3607;
        A[ 7]:=  0.00000 27544 1233;
        A[ 8]:=  0.00000 04324 8450;
        A[ 9]:= -0.00000 00205 3034;
        A[10]:= -0.00000 00438 9154;
        A[11]:= -0.00000 00176 8401;
        A[12]:= -0.00000 00039 9129;
        A[13]:= -0.00000 00001 8693;
        A[14]:=  0.00000 00002 7292;
        A[15]:=  0.00000 00001 3282;
        A[16]:=  0.00000 00000 3183;
```

```
            A[17]:=  0.00000 00000 0167;
            A[18]:= -0.00000 00000 0204;
            A[19]:= -0.00000 00000 0097;
            A[20]:= -0.00000 00000 0022;
            A[21]:= -0.00000 00000 0001;
            A[22]:=  0.00000 00000 0001;
            A[23]:=  0.00000 00000 0001;
            BETAX:= SQRT(- LN((1 + ABSX) * (1 - ABSX)));
            P:= -0.55945 76313 29832 * BETAX +
                2.28791 57162 6336;
            P:= CHEPOLSER(23, P, A);
            INVERF:= "IF" X < 0 "THEN" - BETAX * P
                "ELSE" BETAX * P
        "END"
    "END" INVERSE ERROR FUNCTION;

    EPS:= "-14;
    "IF" PROB < EPS "OR" 1 - PROB < EPS "THEN"
    STATAL3 ERROR("("PHINV")", 1, PROB);
    PHINV:= INVERF(2 * PROB - 1) * 1.41421356237310
"END" PHINV;
"EOP"
```

TITLE: **Phidens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION
The procedure computes the density function of the standard normal distribution for a given argument X.

KEYWORDS
Standard normal density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" PHIDENS (X);
"VALUE" X;
"REAL" X;
"CODE" 41752;
```
Formal parameters
X: <arithmetic expression>, argument of the density function.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier PHIDENS.

PROCEDURES USED
None

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
$PHIDENS(X)=1/\sqrt{(2*\pi)}*EXP(-\frac{1}{2}X^2)$.

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        PHIDENS( 0.000),
        PHIDENS( 1.737),
        PHIDENS(-1.500))
"END"
```

Output:

```
 .398942
 .088255
 .129518
```

SOURCE TEXT

```
"CODE" 41752;
"REAL" "PROCEDURE" PHIDENS(X); "VALUE" X; "REAL" X;
PHIDENS:= .39894228040143 * EXP(- X * X / 2);
         "EOP"
```

TITLE: **Normal**

AUTHOR: E. Opperdoes

INSTITUTE: Mathematical Centre

RECEIVED: 740401

BRIEF DESCRIPTION
The procedure computes the normal distribution function, i.e. the probability that a random variable with a normal distribution is less than or equal to a given value X, the parameters MU and SIGMA are the mean and the standard deviation of the distribution.

KEYWORDS
Normal distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" NORMAL (X, MU,SIGMA );
"VALUE" X, MU,SIGMA;
"REAL" X, MU,SIGMA;
"CODE" 41502;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
MU: <arithmetic expression>, mean of the distribution;
SIGMA: <arithmetic expression>, standard deviation of the distribution.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier NORMAL.
The following error message may appear:
Errornumber 3 (if SIGMA $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 41500

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed as follows:
NORMAL(X,MU,SIGMA)=PHI((X-MU)/SIGMA).

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        NORMAL(  2.5, 2.1, 4.3),
        NORMAL(-1.96,   0,   2),
        NORMAL(  150, 100, 100))
"END"
```

Output:

```
 .537057
 .163543
 .691462
```

Source text

```
"CODE" 41502;
"REAL" "PROCEDURE" NORMAL(X, MU, SIGMA);
    "VALUE" X, MU, SIGMA; "REAL" X, MU, SIGMA;
NORMAL:= "IF" SIGMA <= 0
          "THEN" STATAL3 ERROR("("NORMAL")", 3, SIGMA)
          "ELSE" PHI((X - MU) / SIGMA);
          "EOP"
```

TITLE: **Normalinv**

AUTHOR: E. Opperdoes

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the argument X, for which the normal distribution function has a given value PROB. The parameters MU and SIGMA are the mean and the standard deviation of the distribution.

KEYWORDS

Inverse normal distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" NORMALINV (PROB,MU,SIGMA );
"VALUE" PROB,MU,SIGMA;
"REAL" PROB,MU,SIGMA;
"CODE" 41503;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

MU: <arithmetic expression>, mean of the distribution;

SIGMA: <arithmetic expression>, standard deviation of the distribution.

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier NORMALINV.

The following error messages may appear:

Errornumber 1 (if PROB $< 10^{-14}$ or PROB$>1-10^{-14}$)

Errornumber 3 (if SIGMA ≤ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHINV	STATAL 41501

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:
`NORMALINV(PROB,MU,SIGMA)=MU+PHINV(PROB)*SIGMA)`.

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        NORMALINV(.25,  10,  12),
        NORMALINV(.95, 2.5, 4.1),
        NORMALINV(.49,  .5,  .2))
"END"
```

Output:

```
 +1.906123
 +9.243900
 +0.494986
```

SOURCE TEXT

```
"CODE" 41503;
"REAL" "PROCEDURE" NORMALINV(PROB, MU, SIGMA);
    "VALUE" PROB, MU, SIGMA; "REAL" PROB, MU, SIGMA;
NORMALINV:= "IF" SIGMA <= 0
            "THEN" STATAL3 ERROR("("NORMALINV")", 3, SIGMA)
            "ELSE" "IF" PROB < "-14 "OR" PROB > 1 - "-14
            "THEN" STATAL3 ERROR("("NORMALINV")", 1, PROB)
            "ELSE" MU + PHINV(PROB) * SIGMA;
         "EOP"
```

TITLE: **Normaldens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION

The procedure computes the density function of the normal distribution for a given argument X. The parameters MU and SIGMA are the mean and the standard deviation of the distribution.

KEYWORDS

Normal density function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" NORMALDENS (X,MU,SIGMA );
"VALUE" X,MU,SIGMA;
"REAL" X,MU,SIGMA;
"CODE" 41753;
```

Formal parameters

X: <arithmetic expression>, argument of the density function;

MU: <arithmetic expression>, mean of the distribution;

SIGMA: <arithmetic expression>, standard deviation of the distribution.

DATA AND RESULTS

The value of the density function is assigned to the procedure identifier NORMALDENS.

The following error message may appear:

Errornumber 3 (if $SIGMA \leqslant 0$)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

$$NORMALDENS(X,MU,SIGMA)=EXP(-\frac{1}{2}Y^2)/(\sqrt{2\pi}*SIGMA),$$

where $Y=(X-MU)/SIGMA$.

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        NORMALDENS( .00,  .000, 1.00),
        NORMALDENS(1.50,  .178, 1.03),
        NORMALDENS(3.18, 2.500, 4.00))
"END"
```

Output:

```
 .398942
 .169963
 .098305
```

SOURCE TEXT

```
"CODE" 41753;
"REAL" "PROCEDURE" NORMALDENS(X, MU, SIGMA);
    "VALUE" X, MU, SIGMA; "REAL" X, MU, SIGMA;
NORMALDENS:= "IF" SIGMA <= 0
              "THEN"
                STATAL3 ERROR("("NORMALDENS")", 3, SIGMA)
              "ELSE" EXP(-(((X - MU) / SIGMA) ** 2) / 2)
                     * .39894228040143 / SIGMA;
         "EOP"
```

TITLE: **Bivanorm**

AUTHORS: R. Kaas, F.J.A. Overweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the standard bivariate normal distribution function, i.e. the probability that two variables with a standard bivariate normal distribution with correlation coefficient RHO are less than or equal to X and Y, respectively.

KEYWORDS

Bivariate standard normal distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" BIVANORM (X, Y , RHO );
"VALUE" X, Y , RHO;
"REAL" X, Y , RHO;
"CODE" 41558;
```

Formal parameters

X: <arithmetic expression>, first argument of the distribution function;

Y: <arithmetic expression>, second argument of the distribution function;

RHO: <arithmetic expression>, correlation coefficient of the two variables.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier BIVANORM.

The following error message may appear:

Errornumber 3 (if ABS(RHO) > 1)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 41500

LANGUAGE

Algol 60

Method and performance

The computation of the distribution function is based on formula (31) on p.97 and formula (28) on p.96 from Johnson and Kotz (1972).

The precision is 10^{-14}.

Reference

[1] N.L. Johnson and S. Kotz: *Continuous multivariate distributions*, Houghton Mifflin Company, Boston, 1972.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        BIVANORM(-0.5, -1.0, -.90),
        BIVANORM( 0.3,  0.7,  .40),
        BIVANORM( 0.5,  1.4,  .60))
"END"
```

Output:

```
 .000018
 .518849
 .672725
```

Source text

```
"CODE" 41558;
"REAL" "PROCEDURE" BIVANORM(H, K, RHO); "VALUE" H, K, RHO;
    "REAL" H, K, RHO;
    "BEGIN" "REAL" B;

    "REAL" "PROCEDURE" V(H, K, EPS); "VALUE" H, K, EPS;
        "REAL" H, K, EPS;
        "IF" H = 0 "OR" K = 0 "THEN" V:= 0 "ELSE"
        "IF" ABS(H) < ABS(K) "THEN"
        V:= (PHI(H) - .5) * (PHI(K) - .5) - V(K, H, EPS)
        "ELSE"
        "IF" ABS(K)> 8 "THEN"
            V:=.15915 49430 9189 * ARCTAN (K/H)
        "ELSE"
    "BEGIN" "REAL" M, L, L2, S, R, T, SS, TSN; "INTEGER" N;
        L:= K / H; M:= H * H / 2; L2:= L * L; R:= EXP(-M);
        S:= 1 - R; T:= L; SS:= T * S;
        "FOR" N:= 1, N + 1 "WHILE" ABS(TSN) >= EPS "DO"
        "BEGIN" R:= R * M / N; S:= S - R; T:= -T * L2;
            TSN:= S * T / (2 * N + 1);
            SS:= SS + TSN
        "END";
```

```
        V:= SS * .15915 49430 9189
    "END" V;

    "IF" H < -8 "OR" K < -8 "THEN" B:=0 "ELSE"
    "IF" H > 8 "AND" K > 8 "THEN" B:=1 "ELSE"
    B:= "IF" ABS(RHO) > 1 "THEN"
            STATAL3 ERROR("("BIVANORM")", 3, RHO)
        "ELSE" "IF" ABS(RHO) = 1 "THEN"
           ("IF" RHO = 1 "THEN" ("IF" K <= H "THEN" PHI(K)
                                              "ELSE" PHI(H))
                             "ELSE" ("IF" H <= -K "THEN" 0
                                     "ELSE" PHI(K) - PHI(H)))
        "ELSE" V(H,(K - RHO * H)/ SQRT(1 - RHO ** 2), "-14)
             + V(K,(H - RHO * K)/ SQRT(1 - RHO ** 2), "-14)
             + .5 * (PHI(H) + PHI(K))
             - .15915 49430 9189 * ARCCOS(RHO);
    "IF" B < 0 "THEN" BIVANORM:=0 "ELSE" BIVANORM:=B;
"END" BIVANORM;
        "EOP"
```

TITLE: **Lognormal**

AUTHOR: E. Opperdoes

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION

The procedure computes the lognormal distribution function, i.e. the probability that a random variable with a lognormal distribution is less than or equal to a given value X. The parameters MU and SIGMA are the mean and the standard deviation of the logarithm of the random variable.

KEYWORDS

Lognormal distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" LOGNORMAL (X, MU , SIGMA );
"VALUE" X, MU , SIGMA;
"REAL" X, MU , SIGMA;
"CODE" 41539;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

MU: <arithmetic expression>, mean of the logarithm of the random variable;

SIGMA: <arithmetic expression>, standard deviation of the logarithm of the random variable.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier LOGNORMAL.

The following error message may appear:

Errornumber 3 (if SIGMA $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 41500

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

```
LOGNORMAL(X,MU,SIGMA)=
      {0                           if X<=0,
      {PHI((LN(X)-MU)/SIGMA)       if X>0.
```

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        LOGNORMAL(4, 0, 1),
        LOGNORMAL(1, 0, 1),
        LOGNORMAL(2, 2, 4))
"END"
```

Output:

```
 .917171
 .500000
 .371942
```

SOURCE TEXT

```
"CODE" 41539;
"REAL" "PROCEDURE" LOGNORMAL(X, MU, SIGMA);
    "VALUE" X, MU, SIGMA; "REAL" X, MU, SIGMA;
LOGNORMAL:= "IF" SIGMA <= 0
            "THEN" STATAL3 ERROR("("LOGNORMAL")", 3, SIGMA)
            "ELSE" "IF" X <= 0 "THEN" 0
            "ELSE" PHI((LN(X) - MU) / SIGMA);
         "EOP"
```

TITLE: **Lognormalinv**

AUTHOR: E. Opperdoes

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION
The procedure computes the argument X, for which the lognormal distribution function has a given value PROB. The parameters MU and SIGMA are the mean and the standard deviation of the logarithm of the random variable.

KEYWORDS
Inverse lognormal distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" LOGNORMALINV (PROB, MU , SIGMA );
"VALUE" PROB, MU , SIGMA;
"REAL" PROB, MU , SIGMA;
"CODE" 41540;
```
Formal parameters
PROB: <arithmetic expression>, left hand tail probability of the value to be computed;
MU: <arithmetic expression>, mean of the logarithm of the random variable;
SIGMA: <arithmetic expression>, standard deviation of the logarithm of the random variable.

DATA AND RESULTS
The value of the inverse distribution function is assigned to the procedure identifier LOGNORMALINV.
The following error messages may appear:
Errornumber 1 (if PROB $< 10^{-14}$ or PROB $> 1-10^{-14}$)
Errornumber 3 (if SIGMA ≤ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100
PHINV STATAL 41501

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:
LOGNORMALINV(PROB,MU,SIGMA)=EXP(PHINV(PROB)*SIGMA+MU).

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        LOGNORMALINV(.36, 0, 1),
        LOGNORMALINV(.88, 0, 1),
        LOGNORMALINV(.62, 2, 4))
"END"
```

Output:

```
 +0.698752
 +3.238100
+25.076299
```

SOURCE TEXT

```
"CODE" 41540;
"REAL" "PROCEDURE" LOGNORMALINV(PROB, MU, SIGMA);
"VALUE" PROB, MU, SIGMA; "REAL" PROB, MU, SIGMA;
LOGNORMALINV:= "IF" SIGMA <= 0
    "THEN" STATAL3 ERROR("("LOGNORMALINV")", 3, SIGMA)
    "ELSE" "IF" PROB < "-14 "OR" PROB > 1 - "-14
    "THEN" STATAL3 ERROR("("LOGNORMALINV")", 1, PROB)
    "ELSE" EXP(PHINV(PROB) * SIGMA + MU);
         "EOP"
```

TITLE: **Lognormaldens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION
The procedure computes the density function of the lognormal distribution for a given argument X. The parameters MU and SIGMA are the mean and the standard deviation of the logarithm of the random variable.

KEYWORDS
Lognormal density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" LOGNORMALDENS (X, MU , SIGMA);
"VALUE" X, MU , SIGMA;
"REAL" X, MU , SIGMA;
"CODE" 41574;
```
Formal parameters
X: <arithmetic expression>, argument of the density function;
MU: <arithmetic expression>, mean of the logarithm of the random variable;
SIGMA: <arithmetic expression>, standard deviation of the logarithm of the random variable.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier LOGNORMALDENS.
The following error message may appear:
Errornumber 3 (if SIGMA $\leq$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:

$$\text{LOGNORMALDENS(X,MU,SIGMA)} = \begin{cases} 1/\sqrt{(2*\pi)}*\text{EXP}(-\frac{1}{2}Y^2)/(X*\text{SIGMA}) & \text{if } X > 0, \\ 0 & \text{if } X \leq 0, \end{cases}$$

where Y=(LN(X)-MU)/SIGMA.

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
   OUTPUT(61, "("3(Z.6D,/)")",
        LOGNORMALDENS(4, 0, 1),
        LOGNORMALDENS(1, 0, 1),
        LOGNORMALDENS(2, 2, 4))
"END"
```

Output:

```
.038153
.398942
.047276
```

SOURCE TEXT

```
"CODE" 41754;
"REAL" "PROCEDURE" LOGNORMALDENS(X, MU, SIGMA);
    "VALUE" X, MU, SIGMA; "REAL" X, MU, SIGMA;
LOGNORMALDENS:= "IF" SIGMA <= 0
                "THEN"
                STATAL3 ERROR("("LOGNORMALDENS")", 3, SIGMA)
                "ELSE" "IF" X <= 0 "THEN" 0
                "ELSE"
                  EXP(-(((LN(X) - MU) / SIGMA) ** 2) / 2)
                       * .39894228040143 / X / SIGMA;
        "EOP"
```

TITLE: **Chisq**

AUTHOR: M. van Gelderen

INSTITUTE: Mathematical Centre

RECEIVED: 750501

BRIEF DESCRIPTION
The procedure computes the χ^2 distribution function, i.e. the probability that a random variable having a χ^2 distribution with DF degrees of freedom is less than or equal to a given value X.

KEYWORDS
χ^2 distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" CHISQ (X, DF);
"VALUE" X, DF;
"REAL" X, DF;
"CODE" 41506;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
DF: <arithmetic expression>, number of degrees of freedom of the distribution.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier CHISQ.
The following error message may appear:
Errornumber 2 (if DF $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
GAMMA	STATAL 41513

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed as follows:
CHISQ(X,DF)=GAMMA(X,DF/2,2).

The precision is 10^{-10}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        CHISQ( 1,   1),
        CHISQ(9.2,  11),
        CHISQ(8.4, 8.4))
"END"
```

Output:

```
 .682689
 .396563
 .564927
```

Source text

```
"CODE" 41506;
"REAL" "PROCEDURE" CHISQ(X, DF);
"VALUE" X, DF; "REAL" X, DF;
CHISQ:= "IF" DF <= 0 "THEN"
        STATAL3 ERROR("("CHISQ")", 2, DF)
        "ELSE" "IF" X <= 0 "THEN" 0 "ELSE"
        GAMMA(X, DF / 2, 2);
         "EOP"
```

TITLE: **Chisqinv**

AUTHOR: E. Opperdoes

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION
The procedure computes the argument X, for which the χ^2 distribution function with DF degrees of freedom has a given value PROB.

KEYWORDS
Inverse χ^2 distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" CHISQINV (PROB, DF);
"VALUE" PROB, DF;
"REAL" PROB, DF;
"CODE" 41507;
```
Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

DF: <arithmetic expression>, number of degrees of freedom of the distribution.

DATA AND RESULTS
The argument of the inverse distribution function is assigned to the procedure identifier CHISQINV.
The following error messages may appear:

Errornumber 1	(if PROB $\leqslant 10^{-10}$ or PROB $\geqslant 1-10^{-10}$)
Errornumber 2	(if DF $\leqslant 0$)

PROCEDURES USED

INVERSE	STATAL 40001
STATAL3 ERROR	STATAL 40100
CHISQ	STATAL 41506

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The inverse distribution function is computed as follows:
CHISQINV(PROB,DF)=INVERSE(X,CHISQ(X,DF),PROB,10^{-10}).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(2ZD.6D,/)")",
        CHISQINV(.598,   9),
        CHISQINV(.500, 150),
        CHISQINV(.375,  32))
"END"
```

Output:

```
  9.390906
149.333863
 28.884946
```

SOURCE TEXT

```
"CODE" 41507;
"REAL" "PROCEDURE" CHISQINV(PROB, DF);
    "VALUE" PROB, DF; "REAL" PROB, DF;
"IF" PROB < "-10 "OR" PROB > 1 - "-10
"THEN" STATAL3 ERROR("("CHISQINV")", 1, PROB)
"ELSE" "IF" DF <= 0
"THEN" STATAL3 ERROR("("CHISQINV")", 2, DF)
"ELSE"
"BEGIN" "REAL" X;
    X:= PHINV(PROB) * SQRT(2 * DF) + DF;
    CHISQINV:= INVERSE(X, CHISQ(X, DF), PROB, "-10)
"END" CHISQINV;
         "EOP"
```

TITLE: **Chisqdens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750601

BRIEF DESCRIPTION
The procedure computes the density function of the χ^2 distribution with DF degrees of freedom for a given argument X.

KEYWORDS
χ^2 density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" CHISQDENS (X, DF);
"VALUE" X, DF;
"REAL" X, DF;
"CODE" 41758;
```
Formal parameters
X: <arithmetic expression>, argument of the density function;
DF: <arithmetic expression>, number of degrees of freedom of the distribution.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier CHISQDENS.
The following error message may appear:
Errornumber 2 (if DF $\leq$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:

$$\text{CHISQDENS(X,DF)} = \begin{cases} \text{EXP((DF/2-1)*LN(X)-X/2-DF*LN(2)/2} \\ \text{-LOGGAMMA(DF/2))} & \text{if } X > 0, \\ 0 & \text{if } X \leq 0. \end{cases}$$

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        CHISQDENS(5.5,  10),
        CHISQDENS(  4,  15),
        CHISQDENS(  8, 4.9))
"END"
```

Output:

```
 .076169
 .003273
 .053228
```

SOURCE TEXT

```
"CODE" 41758;
"REAL" "PROCEDURE" CHISQDENS(X, DF);
"VALUE" X, DF; "REAL" X, DF;
CHISQDENS:= "IF" DF <= 0
            "THEN" STATAL3ERROR("("CHISQDENS")", 2, DF)
            "ELSE" "IF" X <= 0 "THEN" 0 "ELSE"
            EXP((DF / 2 - 1) * LN(X) - X / 2 -
                DF * LN(2) / 2 - LOGGAMMA(DF / 2));
         "EOP"
```

TITLE: **Ncchisq**

AUTHORS: J.M. Buhrman, R. van der Horst

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION
The procedure computes the distribution function of the non-central χ^2 distribution, i.e. the probability that a random variable with a non-central χ^2 distribution with DF degrees of freedom and non-centrality parameter DELTA is less than or equal to a given value X.

KEYWORDS
Non-central χ^2 distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" NCCHISQ (X, DF, DELTA);
"VALUE" X, DF, DELTA;
"REAL" X, DF, DELTA;
"CODE" 41509;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
DF: <arithmetic expression>, number of degrees of freedom of the distribution;
DELTA: <arithmetic expression>, non-centrality parameter.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier NCCHISQ.
The following error messages may appear:

Errornumber 2	(if DF is not an integer > 0)
Errornumber 3	(if DELTA < 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400
CHISQ	STATAL 41500

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The computation of the distribution function is based on formula 28.3.2 from Johnson and Kotz (1969) and formula 26.4.8 from Abramowitz and Stegun (1970).

The precision is 10^{-9}.

REFERENCES

[1] M. Abramowitz and D.A. Stegun: *Handbook of mathematical functions,* Dover Publications, New York, 1970.

[2] N.L. Johnson and S. Kotz: *Continuous univariate distributions - 2,* Houghton Mifflin Company, Boston, 1969.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        NCCHISQ(1, 4, 8),
        NCCHISQ(7, 7, 3),
        NCCHISQ(9, 5, 2))
"END"
```

Output:

```
 .002999
 .313851
 .733269
```

SOURCE TEXT

```
"CODE" 41509;
"REAL" "PROCEDURE" NCCHISQ(X, DF, DELTA);
"VALUE" X, DF, DELTA; "REAL" X, DF, DELTA;
"BEGIN" "REAL" FACTOR1, FACTOR2, PROB, SUM, TERM;
    "INTEGER" M;
    "IF" DF < 1 "OR" DF > ENTIER(DF) "THEN"
    STATAL3ERROR("("NCCHISQ")", 2, DF);
    "IF" DELTA < 0 "THEN"
        STATAL3ERROR("("NCCHISQ")", 3, DELTA);

    "IF" X <= 0 "THEN" NCCHISQ:= 0 "ELSE"
    "IF" DELTA = 0 "THEN" NCCHISQ:= CHISQ(X, DF) "ELSE"
    "BEGIN" PROB:= CHISQ(X, DF); X:= X / 2; DF:= DF / 2;
        DELTA:= DELTA / 2; FACTOR1:= EXP(-DELTA);
        FACTOR2:= EXP(DF * LN(X) - X - LOGGAMMA(DF + 1));
        TERM:= SUM:= PROB * FACTOR1; M:= 0;
        "FOR" M:= M + 1
        "WHILE" "NOT"( TERM < "-9 "AND" M > DELTA ) "DO"
        "BEGIN" FACTOR1:= FACTOR1 * DELTA / M;
```

```
            PROB:= PROB - FACTOR2;
            FACTOR2:= FACTOR2 * X / (DF + M);
            TERM:= PROB * FACTOR1; SUM:= SUM + TERM
        "END";
        NCCHISQ:= SUM
    "END";
"END" NCCHISQ;
         "EOP"
```

TITLE: **Student**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 740121

BRIEF DESCRIPTION
The procedure computes the Student's t-distribution function, i.e. the probability that a random variable having a Student's t-distribution with DF degrees of freedom is less than or equal to a given value X.

KEYWORDS
Student's t-distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" STUDENT (X,DF,);
"VALUE" X,DF;
"REAL" X,DF;
"CODE" 41530;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
DF: <arithmetic expression>, number of degrees of freedom of the distribution.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier STUDENT.
The following error message may appear:
Errornumber 2 (if DF $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
INCOMPLETE BETA	STATAL 40401

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed as follows:
STUDENT(X,DF)=

$$\begin{cases} 1-IB/2 & \text{if } X \geqslant 0, \\ IB/2 & \text{if } X < 0, \end{cases}$$

where IB=INCOMPLETE BETA(DF/(DF+X*X), DF/2,.5, EPS)/2.
EPS, the precision of the incomplete beta function is 10^{-12}.

The precision of the computation is 10^{-12}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        STUDENT(-1.3,  2),
        STUDENT(13.8, 10),
        STUDENT( 4.2,  5))
"END"
```

Output:

```
 .161624
1.000000
 .995755
```

SOURCE TEXT

```
"CODE" 41530;
"REAL" "PROCEDURE" STUDENT(X, DF); "VALUE" X, DF;
    "REAL" X, DF;
"BEGIN" "REAL"IB;

    "IF" DF <= 0 "THEN"
        STUDENT:= STATAL3 ERROR("("STUDENT")",2,DF)
    "ELSE"
    "BEGIN" IB:=
        INCOMPLETE BETA(DF/(DF + X * X),DF/2,0.5,"-12);
        "IF" IB < 0 "THEN" IB:= 0
        "ELSE" "IF" IB > 2 "THEN" IB:= 2;
        STUDENT:= "IF" X < 0 "THEN" IB / 2 "ELSE" 1 - IB / 2
    "END"
"END" STUDENT;
         "EOP"
```

TITLE: **Studentinv**

AUTHOR: M. van Gelderen

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the argument X, for which the Student's t-distribution function with DF degrees of freedom has a given value PROB.

KEYWORDS

Inverse Student's t-distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" STUDENTINV (PROB,DF,);
"VALUE" PROB,DF,;
"REAL" PROB,DF,;
"CODE" 41531;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

DF: <arithmetic expression>, number of degrees of freedom of the distribution;

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier STUDENTINV.

The following error messages may appear:

Errornumber 1	(if PROB $< 10^{-10}$ or PROB $> 1-10^{-10}$)
Errornumber 2	(if DF ≤ 0)

PROCEDURES USED

INVERSE	STATAL 40001
STATAL3 ERROR	STATAL 40100
STUDENT	STATAL 41530

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The inverse distribution function is computed as follows:
STUDENTINV(PROB,DF)=INVERSE(X,STUDENT(X,DF),PROB,10^{-10}).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        STUDENTINV(.747,   3),
        STUDENTINV(.685,  14),
        STUDENTINV(.332, 189))
"END"
```

Output:

```
 +0.753294
 +0.492499
 -0.435081
```

SOURCE TEXT

```
"CODE" 41531;
"REAL" "PROCEDURE" STUDENTINV(PROB, DF);
    "VALUE" PROB, DF; "REAL" PROB, DF;
"BEGIN"
    "IF" PROB < "-10 "OR" PROB > 1 - "-10
    "THEN" STATAL3 ERROR("("STUDENTINV")", 1, PROB)
    "ELSE" "IF" DF <= 0
    "THEN" STATAL3 ERROR("("STUDENTINV")", 2, DF)
    "ELSE"
    "BEGIN" "REAL" X, U, U2;
        U:= PHINV(PROB); U2:= U * U;
        X:= U * (1 + (U2 + 1) / 4 / DF +
                (3 + U2 * (U2 * 5 + 16)) / 96 / DF / DF);
        STUDENTINV:= INVERSE(X, STUDENT(X, DF), PROB, "-10)
    "END"
"END" STUDENTINV;
         "EOP"
```

TITLE: **Studentdens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750601

BRIEF DESCRIPTION
The procedure computes the density function of the Student's t-distribution with DF degrees of freedom for a given argument X.

KEYWORDS
Student's t-density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" STUDENTDENS (X,DF);
"VALUE" X,DF,;
"REAL" X,DF;
"CODE" 41762;
```
Formal parameters
X: <arithmetic expression>, argument of the density function;
DF: <arithmetic expression>, number of degrees of freedom of the distribution;

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier STUDENTDENS.
The following error message may appear:
Errornumber 2 (if DF $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
```
STUDENTDENS(X,DF)=EXP(LOGGAMMA((DF+1)/2)-LOGGAMMA(DF/2)-
                 (DF+1)/2*LN(1+X²/DF)-LN(DF)/2-LN(π)/2).
```

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        STUDENTDENS( 1, 10),
        STUDENTDENS( 2,  7),
        STUDENTDENS(21, 26))
"END"
```

Output:

```
 .230362
 .063135
 .000000
```

SOURCE TEXT

```
"CODE" 41762;
"REAL" "PROCEDURE" STUDENTDENS(X, DF);
"VALUE" X, DF; "REAL" X, DF;
STUDENTDENS:= "IF" DF <= 0
              "THEN" STATAL3 ERROR("("STUDENTDENS")", 2, DF)
              "ELSE"
              EXP(LOGGAMMA((DF + 1) / 2) - LOGGAMMA(DF / 2) -
               (DF + 1) / 2 * LN(1 + X * X / DF) -
                LN(DF) / 2 - .57236494299247);
          "EOP"
```

TITLE: **Ncstudent**

AUTHOR: H. Elffers

INSTITUTE: Mathematical Centre

RECEIVED: 750601

BRIEF DESCRIPTION

The procedure computes the non-central Student's t-distribution function, i.e. the probability that a random variable having a non-central Student's t-distribution with DF degrees of freedom is less than or equal to a given value X. DELTA is the non-centrality parameter.

KEYWORDS

Non-central Student's t-distribution

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" NCSTUDENT (X,DF, DELTA);
"VALUE" X,DF, DELTA;
"REAL" X,DF, DELTA;
"CODE" 41533;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

DF: <arithmetic expression>, number of degrees of freedom of the distribution;

DELTA: <arithmetic expression>, non-centrality parameter.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier NCSTUDENT.

The following error messages may appear:

Errornumber 0	(if it is impossible to compute the distribution function with prescribed precision)
Errornumber 2	(if DF is not an integer > 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 40500

LANGUAGE

Algol 60

Method and performance

An exact representation, according to Owen (1968), p.464, is used for the computation of the distribution function. For odd DF an integral has to be evaluated numerically over a finite interval; details can be found in Elffers (unpublished).

The precision is 10^{-8}.

References

[1] D.B. Owen: *A survey of properties and applications of the noncentral t-distribution,* Technometrics, 10, (1968), p.445-78.
[2] H. Elffers: *Cornisch-Fisher expansie in exacte procedures voor de berekening van de niet-centrale studentverdeling,* (unpublished report).

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        NCSTUDENT(-1.81, 10,    1),
        NCSTUDENT( 2.93,  8,  2.5),
        NCSTUDENT( 4.12, 27, 13.7))
"END"
```

Output:

```
 +0.004946
 +0.605456
 +0.000000
```

Source text

```
"CODE" 41533;
"REAL" "PROCEDURE" NCSTUDENT(X, DF, DELTA);
"VALUE" X, DF, DELTA; "REAL" X, DF, DELTA;
"BEGIN" "REAL" A, B, A2, WB, D2, TOL, TOLI, H, HELP, RESULT;
    "BOOLEAN" DFEVEN;

    "REAL" "PROCEDURE" INTEGRATE(Y0, Y4, F0, F2, F4);
    "VALUE" Y0, Y4, F0, F2, F4; "REAL" Y0, Y4, F0, F2, F4;
    "BEGIN" "REAL" F1, F3, Y2, TEE, Y;
        Y2:=(Y0 + Y4)/2;
        Y :=(Y0 + Y2)/2; F1:= EXP(H*(1 + Y*Y))/(1 + Y*Y);
        Y :=(Y2 + Y4)/2; F3:= EXP(H*(1 + Y*Y))/(1 + Y*Y);
        TEE:=6*F2 - 4*(F1 + F3) + F0 +F4;
        INTEGRATE:="IF" ABS(TEE) < TOLI
            "THEN" (Y4 - Y0)*(4*(F1 + F3) + 2*F2 +
                   F0 + F4 - TEE/15)
            "ELSE" INTEGRATE(Y0, Y2, F0, F1, F2) +
                   INTEGRATE(Y2, Y4, F2, F3, F4);
```

```
"END" INTEGRATE;

"REAL" "PROCEDURE" SUMMATION OF FACTORS M;
"BEGIN" "INTEGER" I;
    "REAL" MSUM, COEF, MIMIN2, MIMIN1, MI;
    "BOOLEAN" ADD;
    MSUM:= 0;
    "IF" DF > 1 "THEN"
    "BEGIN"
        MIMIN2:= A*WB * EXP(H) * PHI(HELP*WB) *
                 .3989422804;
        "IF" DFEVEN "THEN" MSUM:= MSUM + MIMIN2;
        "IF" DF > 2 "THEN"
        "BEGIN" COEF:= 1;
            MIMIN1:= B*(HELP*MIMIN2 +
                     A*.1591549431*EXP(-.5*D2));
            "IF" ^ DFEVEN "THEN" MSUM:= MSUM + MIMIN1;
            ADD:= DFEVEN;
            "FOR" I:= 2 "STEP" 1 "UNTIL" DF - 2 "DO"
            "BEGIN" MI:= (I - 1)/I*B*
                         (COEF*HELP*MIMIN1 + MIMIN2);
                "IF" ADD "THEN" MSUM:= MSUM + MI;
                ADD:= ^ ADD; COEF:= 1/(I - 1)/COEF;
                MIMIN2:= MIMIN1; MIMIN1:= MI;
            "END" I;
        "END" DF>2;
    "END" DF>1;
    SUMMATION OF FACTORS M:= MSUM;
"END" SUMMATION OF FACTORS M;

"PROCEDURE" INITIALISATION;
"BEGIN" TOL:= "-8;
    "IF" DF < 1 "OR" ENTIER(DF) ^= DF "THEN"
    STATAL3 ERROR("("NCSTUDENT")", 2, DF);
    DFEVEN:= ENTIER(DF/2) = DF/2;
    A:= X/SQRT(DF); A2:= A*A; D2:= DELTA*DELTA;
    HELP:= DELTA*A;
    B:=DF/(DF + X*X); WB:= SQRT(B); H:=-D2*B*.5;
    "IF" ABS(A) > TOL "THEN" TOLI:= 180 * TOL / ABS(A);
"END" INITIALISATION;

INITIALISATION;

RESULT:=
"IF" DFEVEN "THEN"
    PHI(-DELTA) + SUMMATION OF FACTORS M * 2.5066282746
"ELSE"
    PHI(-DELTA*WB) + SUMMATION OF FACTORS M * 2 +
    ("IF" ABS(A) <= TOL "THEN" 0 "ELSE"
    .31830 98862 * INTEGRATE(0, A, EXP(H),
    EXP(H*(1 + A2/4))/(1 + A2/4),
    EXP(H*(1 + A2))/(1 + A2)) / 12);

NCSTUDENT:=
```

```
    "IF" TOL <= RESULT "AND" RESULT <= 1 - TOL "THEN" RESULT
    "ELSE"
    "IF" ABS(RESULT) < TOL "THEN" 0 "ELSE"
    "IF" ABS(RESULT - 1) < TOL "THEN" 1 "ELSE"
    STATAL3 ERROR("("NCSTUDENT")", 0, RESULT);
"END" NCSTUDENT;
        "EOP"
```

TITLE: **Ncstudentinv**

AUTHOR: H. Elffers

INSTITUTE: Mathematical Centre

RECEIVED: 750601

BRIEF DESCRIPTION
The procedure computes the argument X, for which the non-central Student's t-distribution function with DF degrees of freedom has a given value PROB. DELTA is the non-centrality parameter.

KEYWORDS
Inverse non-central Student's t-distribution

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" NCSTUDENTINV (PROB, DF, DELTA);
"VALUE" PROB, DF, DELTA;
"REAL" PROB, DF, DELTA;
"CODE" 41534;
```
Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

DF: <arithmetic expression>, number of degrees of freedom of the distribution;

DELTA: <arithmetic expression>, non-centrality parameter.

DATA AND RESULTS
The value of the inverse distribution function is assigned to the procedure identifier NCSTUDENTINV.
The following error messages may appear:

Errornumber 0	(if it is impossible to find a solution)
Errornumber 1	(if PROB $< 10^{-7}$ or PROB $> 1-10^{-7}$)
Errornumber 2	(if DF is not an integer > 0)

PROCEDURES USED

ZEROIN	NUMAL 34150
STATAL3 ERROR	STATAL 40100
PHINV	STATAL 41501
NCSTUDENT	STATAL 41533

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The procedure searches a solution of the equation in T: NCSTUDENT(T,DF,DELTA)=PROB. This is performed by a call of the procedure ZEROIN in an interval which is constructed by means of an initial Cornish-Fisher approximation to the solution. For details see reference [2] of section 1.2.5.4.

The precision is 10^{-7}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        NCSTUDENTINV(.05, 19, 8),
        NCSTUDENTINV(.52, 15, 4),
        NCSTUDENTINV(.81, 10, 0))
"END"
```

Output:

```
 +5.803159
 +4.139614
 +0.918456
```

SOURCE TEXT

```
"CODE" 41534;
"REAL" "PROCEDURE" NCSTUDENTINV(PROB, DF, DELTA);
    "VALUE" PROB, DF, DELTA; "REAL" PROB, DF, DELTA;
"BEGIN" "REAL" X, Y, TOL;

    "PROCEDURE" CORNISH FISHER EXPANSION;
    "BEGIN" "REAL" UA, UA2, UA3, UA4, UA5;
        "INTEGER" DF4, DFDF;
        UA:=PHINV(PROB);
        UA2:=UA*UA; UA3:=UA2*UA; UA4:=UA2*UA2;
        UA5:=UA4*UA; DF4:=DF*4; DFDF:=DF*DF;
        X:=-UA/DFDF/32;
        X:=X*DELTA - (UA2 - 1)/DFDF/24;
        X:=X*DELTA + UA/DF4 + (UA3 + UA*4)/DFDF/16;
        X:=X*DELTA + 1 + (UA2*2 + 1)/DF4 +
           (UA4*4 + UA2*12 + 1)/DFDF/32;
        X:=X*DELTA + UA + (UA3 + UA)/DF4 + (UA5*5 + UA3*16
           + UA*3)/DFDF/96;
    "END" INITIAL APPROXIMATION BY CORNISH-FISHER METHOD;
```

```
    TOL:= "-7;
    "IF" PROB < TOL "OR" PROB > 1 - TOL "THEN"
    STATAL3 ERROR("("NCSTUDENTINV")", 1, PROB);
    "IF" DF < 1 "OR" ENTIER(DF) ^= DF "THEN"
    STATAL3 ERROR("("NCSTUDENTINV")", 2, DF);

    CORNISH FISHER EXPANSION;

    NCSTUDENTINV:=
        INVERSE(X, NCSTUDENT(X, DF, DELTA), PROB, TOL)
"END" NCSTUDENTINV;
         "EOP"
```

TITLE: **Fisher**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 740114

BRIEF DESCRIPTION
The procedure computes the Fisher distribution function, i.e. the probability that a random varianble having a Fisher distribution with DF1 and DF2 degrees of freedom is less than or equal to a given value X.

KEYWORDS
Fisher distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" FISHER (X,DF1,DF2);
"VALUE" X,DF1,DF2;
"REAL" X,DF1,DF2;
"CODE" 41521;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution;
DF1: <arithmetic expression>, number of degrees of freedom of the numerator;
DF2: <arithmetic expression>, number of degrees of freedom of the denominator.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier FISHER.
The following error messages may appear:
Errornumber 2 (if DF1 $\leqslant$ 0)
Errornumber 2 (if DF2 $\leqslant$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100
INCOMPLETE BETA STATAL 40401

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

FISHER(X,DF1,DF2)=

$$\begin{cases} 1-\text{INCOMPLETE BETA}(DF2/(DF2+DF1*X), DF2/2, DF1/2, EPS) & \text{if } X > 0, \\ 0 & \text{if } X \leq 0. \end{cases}$$

EPS, the precision of the incomplete beta function, is 10^{-12}.

The precision of the computation is 10^{-12}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        FISHER( 1, 3, 2),
        FISHER(23, 3, 4),
        FISHER(49, 3, 9))
"END"
```

Output:

```
.464758
.994474
.999993
```

SOURCE TEXT

```
"CODE" 41521;
"REAL" "PROCEDURE" FISHER(X, DF1, DF2); "VALUE" X, DF1, DF2;
    "REAL" X, DF1, DF2;
"BEGIN" "REAL"IB;

    "IF" DF1 <= 0 "THEN"
        FISHER:= STATAL3 ERROR("("FISHER")",2,DF1)
    "ELSE"
    "IF" DF2 <= 0 "THEN"
        FISHER:= STATAL3 ERROR("("FISHER")",3,DF2)
    "ELSE"
    "IF" X <= 0 "THEN" FISHER:= 0
    "ELSE"
    "BEGIN" IB:= INCOMPLETE BETA(DF2/(DF2 + DF1 * X)
        , DF2/2,DF1/2,"-12);
        "IF" IB < 0 "THEN" IB:= 0
        "ELSE" "IF" IB > 1 "THEN" IB:= 1;
        FISHER:= 1 - IB
    "END"
"END" FISHER;
         "EOP"
```

TITLE: **Fisherinv**

AUTHOR: M. van Gelderen

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the argument X, for which the Fisher distribution function with DF1 and DF2 degrees of freedom has a given value PROB.

KEYWORDS

Inverse Fisher distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" FISHERINV (PROB,DF1,DF2);
"VALUE" PROB,DF1,DF2;
"REAL" PROB,DF1,DF2;
"CODE" 41522;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

DF1: <arithmetic expression>, number of degrees of freedom of the numerator;

DF2: <arithmetic expression>, number of degrees of freedom of the denominator.

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier FISHERINV.

The following error messages may appear:

Errornumber 1	(if $PROB < 10^{-10}$ or $PROB > 1-10^{-10}$)
Errornumber 2	(if $DF1 \leqslant 0$)
Errornumber 3	(if $DF2 \leqslant 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
INVERSE	STATAL 40001
FISHER	STATAL 41521

Language
Algol 60

Method and performance
The inverse distribution function is computed as follows:
FISHERINV(X,DF1,DF2)=INVERSE(X,FISHER(X,DF1,DF2),PROB,10^{-10}).

The precision is 10^{-10}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        FISHERINV(.950,   6,  22),
        FISHERINV(.338,  10,  18),
        FISHERINV(.712, 100, 120))
"END"
```

Output:

```
2.549061
 .762042
1.111899
```

Source text

```
"CODE" 41522;
"REAL" "PROCEDURE" FISHERINV(PROB, DF1, DF2);
    "VALUE" PROB, DF1, DF2; "REAL" PROB, DF1, DF2;
"BEGIN"
    "IF" PROB < "-10 "OR" PROB > 1 - "-10
    "THEN" STATAL3 ERROR("("FISHERINV")", 1, PROB)
    "ELSE" "IF" DF1 <= 0
    "THEN" STATAL3 ERROR("("FISHERINV")", 2, DF1)
    "ELSE" "IF" DF2 <= 0
    "THEN" STATAL3 ERROR("("FISHERINV")", 3, DF2)
    "ELSE"
    "BEGIN" "REAL" X;
        X:= "IF" PROB <= .5 "THEN" .5 "ELSE"
            "IF" DF2 <= 4 "THEN" 1 "ELSE"
            DF2 / (DF2 - 2) + PHINV(PROB) *
            SQRT(2 * DF2 * DF2 * (DF1 + DF2 - 2) /
                 (DF1 * (DF2 - 4) * (DF2- 2) * (DF2 - 2)));
        FISHERINV:=
            INVERSE(X, FISHER(X, DF1, DF2), PROB, "-10)
    "END"
"END" FISHERINV;
        "EOP"
```

TITLE: **Fisherdens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750601

BRIEF DESCRIPTION

The procedure computes the density function of the Fisher distribution with DF1 and DF2 degrees of freedom for a given argument X.

KEYWORDS

Fisher density function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" FISHERDENS (X, DF1, DF2);
"VALUE" X, DF1, DF2;
"REAL" X, DF1, DF2;
"CODE" 41761;
```

Formal parameters

X: <arithmetic expression>, argument of the density function;

DF1: <arithmetic expression>, number of degrees of freedom of the numerator;

DF2: <arithmetic expression>, number of degrees of freedom of the denominator.

DATA AND RESULTS

The value of the density function is assigned to the procedure identifier FISHERDENS.

The following error messages may appear:

Errornumber 2 (if DF1 $\leq$ 0)
Errornumber 3 (if DF2 $\leq$ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100
LOGGAMMA STATAL 40400

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The density function is computed as follows:

FISHERDENS(X,DF1,DF2)=

```
{EXP(LOGGAMMA((DF1+DF2)/2)-LOGGAMMA(DF1/2)-
 LOGGAMMA(DF2/2)+
 (DF1*LN(DF1)+DF2*LN(DF2))/
 2+(DF1/2-1)*LN(X)-(DF1+DF2)/2*
 LN(DF1*X+DF2))                          if X > 0,
 0                                       if X <= 0.
```

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        FISHERDENS(10,  11,  12),
        FISHERDENS( 1, 3.5, 5.7),
        FISHERDENS(15,  20,   2))
"END"
```

Output:

```
.000097
.391929
.004131
```

SOURCE TEXT

```
"CODE" 41761;
"REAL" "PROCEDURE" FISHERDENS(X, DF1, DF2);
"VALUE" X, DF1, DF2;
    "REAL" X, DF1, DF2;
FISHERDENS:= "IF" DF1 <= 0
             "THEN" STATAL3 ERROR("("FISHERDENS")", 2, DF1)
             "ELSE"
             "IF" DF2 <= 0
             "THEN" STATAL3 ERROR("("FISHERDENS")", 3, DF2)
             "ELSE"
             "IF" X <= 0 "THEN" 0 "ELSE"
             EXP(LOGGAMMA((DF1 + DF2) / 2) -
              LOGGAMMA(DF1 / 2) - LOGGAMMA(DF2 / 2) +
               (DF1 * LN(DF1) + DF2 * LN(DF2)) / 2 +
                (DF1 / 2 - 1) * LN(X) - (DF1 + DF2) / 2 *
                  LN(DF1 * X + DF2));
         "EOP"
```

TITLE: **Ncfisher**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770901

BRIEF DESCRIPTION

The procedure computes the non-central Fisher distribution function, i.e. the probability that a random variable having a non-central Fisher distribution with DF1 and DF2 degrees of freedom and non-centrality parameter DELTA is less than or equal to a given value X.

KEYWORDS

Non-central Fisher distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" NCFISHER (X, DF1, DF2, "DELTA");
"VALUE" X, DF1, DF2, DELTA;
"REAL" X, DF1, DF2, "DELTA";
"CODE" 41525;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

DF1: <arithmetic expression>, number of degrees of freedom of the numerator;

DF2: <arithmetic expression>, number of degrees of freedom of the denominator;

DELTA: <arithmetic expression>, non-centrality parameter.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier NCFISHER.

The following error messages may appear:

Errornumber 2	(if $DF1 \leqslant 0$)
Errornumber 3	(if $DF2 \leqslant 0$)
Errornumber 4	(if $DELTA < 0$)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
INCOMPLETE BETA	STATAL 40401

Language
Algol 60

Method and performance
The computation of the distribution function is based on formula 26.6.20 on p.947 in Abramowitz and Stegun (1970).
If DF1 $\leq$ 1000 and DF2 $\leq$ 1000 the precision of the computation is 10^{-12}.

Reference
[1] M. Abramowitz & I.A. Stegun: *Handbook of mathematical functions*, Dover publications, New York, 1970.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        NCFISHER(33.677, 1, 19, 64.00),
        NCFISHER( .750, 4,  5,  1.20),
        NCFISHER( 2.750, 3, 12,  2.25))
"END"
```

Output:

```
 .050003
 .301548
 .750618
```

Source text

```
"CODE" 41525;
"REAL" "PROCEDURE" NCFISHER(X,DF1,DF2,DELTA);
"VALUE" X,DF1,DF2,DELTA; "REAL" X,DF1,DF2,DELTA;
"BEGIN" "INTEGER" J; "REAL" XX,FAKTOR1,FAKTOR2,EPS,SUM;

  "IF" DF1 <= 0 "THEN" STATAL3 ERROR("("NCFISHER")",2,DF1)
  "ELSE"
  "IF" DF2 <= 0 "THEN" STATAL3 ERROR("("NCFISHER")",3,DF2)
  "ELSE"
  "IF" DELTA < 0 "THEN"
      STATAL3 ERROR("("NCFISHER")",4,DELTA)
  "ELSE" "IF" X <= 0 "THEN" NCFISHER:= 0 "ELSE"
  "BEGIN" XX:= (DF1 * X) / (DF1 * X + DF2); EPS:= "-12;
    DF1:= DF1 / 2; DF2:= DF2 / 2; DELTA:= DELTA / 2;
    FAKTOR1:= 1;
    FAKTOR2:= SUM:= INCOMPLETE BETA(XX,DF1,DF2,EPS);
    "IF" DELTA = 0 "THEN" "GOTO" UIT;
    J:= 0; "FOR" J:= J + 1 "WHILE" FAKTOR2 > EPS "DO"
    "BEGIN" FAKTOR1:= FAKTOR1 * DELTA / J;
      FAKTOR2:= FAKTOR1 *
```

```
                 INCOMPLETE BETA(XX,DF1 + J,DF2,EPS);
      SUM:= SUM + FAKTOR2
    "END";
UIT: NCFISHER:= EXP(-DELTA) * SUM
  "END"
"END" NCFISHER;
          "EOP"
```

TITLE: **Expon**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 741206

BRIEF DESCRIPTION

The procedure computes the distribution function of the exponential distribution, i.e. the probability that a random variable with an exponential distribution is less than or equal to a given value X. The scale parameter LAMBDA is the inverse of the mean of the distribution.

KEYWORDS

Exponential distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" EXPON (X, LAMBDA);
"VALUE" X, LAMBDA";
"REAL" X, LAMBDA;
"CODE" 41561;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

LAMBDA: <arithmetic expression>, inverse scale parameter.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier EXPON.

The following error message may appear:

Errornumber 2 (if LAMBDA ⩽ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

EXPON(X,LAMBDA)=

$$\begin{cases} 0 & \text{if } X \leqslant 0, \\ 1-EXP(-X*LAMBDA) & \text{if } X > 0. \end{cases}$$

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        EXPON( 2.0, 1.0),
        EXPON( 3.1,  .5),
        EXPON(-1.2, 4.0))
"END"
```

Output:

```
 .864665
 .787752
 .000000
```

SOURCE TEXT

```
"CODE" 41561;
"REAL" "PROCEDURE" EXPON (X, LAMBDA);
"VALUE" X, LAMBDA; "REAL" X, LAMBDA;
EXPON:= "IF" LAMBDA <= 0
        "THEN" STATAL3 ERROR("("EXPON")", 2, LAMBDA)
        "ELSE" "IF" X <= 0 "THEN" 0
        "ELSE" 1 - EXP(-LAMBDA * X);
         "EOP"
```

TITLE: **Exponinv**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 741206

BRIEF DESCRIPTION
The procedure computes the argument X, for which the exponential distribution function has a given value PROB. The scale parameter LAMBDA is the inverse of the mean of the distribution.

KEYWORDS
Inverse exponential distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" EXPONINV (PROB, LAMBDA);
"VALUE" PROB, LAMBDA;
"REAL" PROB, LAMBDA;
"CODE" 41562;
```
Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

LAMBDA: <arithmetic expression>, inverse scale parameter.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier EXPONINV.
The following error messages may appear:

Errornumber 1 (if PROB $< 10^{-14}$ or PROB $> 1-10^{-14}$)
Errornumber 2 (if LAMBDA $\leqslant 0$)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The inverse distribution function is computed as follows:
EXPONINV(PROB,LAMBDA)=−LN(1−PROB)/LAMBDA.

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        EXPONINV(.90, 1.8),
        EXPONINV(.71, 5.0),
        EXPONINV(.25, 2.3))
"END"
```

Output:

```
 +1.279214
 +0.247575
 +0.125079
```

Source text

```
"CODE" 41562;
"REAL" "PROCEDURE" EXPONINV(PROB, LAMBDA);
"VALUE" PROB, LAMBDA; "REAL" PROB, LAMBDA;
EXPONINV:= "IF" LAMBDA <= 0
           "THEN" STATAL3 ERROR("("EXPONINV")",2,LAMBDA)
           "ELSE" "IF" PROB <= 0 "OR" PROB >= 1
           "THEN" STATAL3 ERROR("("EXPONINV")",1,PROB)
           "ELSE" - LN(1 - PROB) / LAMBDA;
         "EOP"
```

TITLE: **Expondens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750601

BRIEF DESCRIPTION
The procedure computes the density function of the exponential distribution for a given argument X. The scale parameter LAMBDA is the inverse of the mean of the distribution.

KEYWORDS
Exponential density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" EXPONDENS (X, LAMBDA);
"VALUE" X, LAMBDA;
"REAL" X, LAMBDA;
"CODE" 41755;
```
Formal parameters
X: <arithmetic expression>, argument of the density function;
LAMBDA: <arithmetic expression>, inverse scale parameter.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier EXPONDENS.
The following error message may appear:
Errornumber 2 (if LAMBDA $\leqslant$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
EXPONDENS(X,LAMBDA)=

$$\begin{cases} \text{LAMBDA*EXP(}-\text{LAMBDA*X)} & \text{if } X > 0, \\ 0 & \text{if } X \leqslant 0. \end{cases}$$

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        EXPONDENS(3, 4),
        EXPONDENS(2, 5),
        EXPONDENS(4, 9))
"END"
```

Output:

```
 .000025
 .000227
 .000000
```

SOURCE TEXT

```
"CODE" 41755;
"REAL" "PROCEDURE" EXPONDENS(X, LAMBDA); "VALUE" X, LAMBDA;
    "REAL" X, LAMBDA;
EXPONDENS:= "IF" LAMBDA <= 0
            "THEN"
               STATAL3 ERROR("("EXPONDENS")", 2, LAMBDA)
            "ELSE" "IF" X <= 0 "THEN" 0
                   "ELSE" LAMBDA * EXP(- LAMBDA * X);
         "EOP"
```

TITLE: **Logistic**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750501

BRIEF DESCRIPTION

The procedure computes the logistic distribution function. i.e. the probability that a random variable with a logistic distribution is less than or equal to a given value X. The location parameter LOC is the mean of the distribution, the scale parameter SCALE is proportional to the standard deviation of the distribution.

KEYWORDS

Logistic distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" LOGISTIC (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41550;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;
LOC: <arithmetic expression>, location parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier LOGISTIC.

The following error message may occur:

Errornumber 3 (if SCALE $\leq$ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

```
LOGISTIC(X,LOC,SCALE)=1/(1+EXP(-(X-LOC)/SCALE)).
```

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        LOGISTIC(  0, 0, 1),
        LOGISTIC(.37, 1, 2),
        LOGISTIC(.88, 0, 1))
"END"
```

Output:

```
 .500000
 .421895
 .706822
```

Source text

```
"CODE" 41550;
"REAL" "PROCEDURE" LOGISTIC(X, MU, SIGMA);
"VALUE" X, MU, SIGMA; "REAL" X, MU, SIGMA;
LOGISTIC:= "IF" SIGMA <= 0
           "THEN" STATAL3 ERROR("("LOGISTIC")", 3, SIGMA)
           "ELSE" 1 / (1 + EXP(-(X - MU) / SIGMA));
         "EOP"
```

TITLE: **Logisticinv**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750501

BRIEF DESCRIPTION

The procedure computes the argument X, for which the logistic distribution function has a given value PROB. The location parameter LOC is the mean of the distribution, the scale parameter SCALE is proportional to the standard deviation of the distribution.

KEYWORDS

Inverse logistic distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" LOGISTICINV (PROB, LOC, SCALE);
"VALUE" PROB, LOC, SCALE;
"REAL" PROB, LOC, SCALE;
"CODE" 41551;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

LOC: <arithmetic expression>, location parameter;

SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier LOGISTICINV.

The following error messages may occur:

Errornumber 1 (if PROB $\leqslant$ 0 or PROB $\geqslant$ 1)

Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

Method and performance

The inverse distribution function is computed as follows:
`LOGISTICINV(PROB,LOC,SCALE)=-SCALE*LN((1-PROB)/PROB)+LOC.`

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        LOGISTICINV(0.3, 0, 1),
        LOGISTICINV(0.1, 1, 2),
        LOGISTICINV(0.8, 2, 4))
"END"
```

Output:

```
 -0.847298
 -3.394449
 +7.545177
```

Source text

```
"CODE" 41551;
"REAL" "PROCEDURE" LOGISTICINV (PROB, MU, SIGMA);
"VALUE" PROB, MU, SIGMA; "REAL" PROB, MU, SIGMA;
LOGISTICINV:= "IF" SIGMA <= 0
    "THEN" STATAL3 ERROR("("LOGISTICINV")", 3, SIGMA)
    "ELSE" "IF" PROB <= 0 "OR" PROB >= 1
    "THEN" STATAL3 ERROR("("LOGISTICINV")", 1, PROB)
    "ELSE" - SIGMA * LN((1 - PROB) / PROB) + MU;
        "EOP"
```

TITLE: **Logisticdens**

AUTHOR: J. Bethlehem

INSTITUTE: Mathematical Centre

RECEIVED: 750501

BRIEF DESCRIPTION
The procedure computes the logistic density function for a given argument X. The location parameter LOC is the mean of the density, the scale parameter SCALE is proportional to the standard deviation of the distribution.

KEYWORDS
Logistic density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" LOGISTICDENS (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41765;
```
Formal parameters
X: <arithmetic expression>, argument of the density function;
LOC: <arithmetic expression>, location parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier LOGISTICDENS.
The following error message may occur:
Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
LOGISTICDENS(X,LOC,SCALE)=Y/((1+Y)^2*SCALE),
where Y=EXP(−(X−LOC)/SCALE).

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        LOGISTICDENS( 0, 0, 1),
        LOGISTICDENS( 1, 1, 2),
        LOGISTICDENS(-1, 2, 4))
"END"
```

Output:

```
 .250000
 .125000
 .054474
```

Source text

```
"CODE" 41765;
"REAL" "PROCEDURE" LOGISTICDENS(X, MU, SIGMA);
    "VALUE" X, MU, SIGMA; "REAL" X, MU, SIGMA;
"BEGIN"
    "IF" SIGMA <= 0
    "THEN" STATAL3 ERROR("("LOGISTICDENS")", 3, SIGMA);
    X:= EXP(-(X - MU) / SIGMA);
    LOGISTICDENS:= X / ((1 + X) * (1 + X) * SIGMA)
"END" LOGISTICDENS;
          "EOP"
```

TITLE: **Gamma**

AUTHOR: M. van Gelderen

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the gamma distribution function, i.e. the probability that a random variable having a gamma distribution with shape and scale parameters ALPHA and SCALE is less than or equal to a given value X.

KEYWORDS

Gamma distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" GAMMA (X, ALPHA, SCALE);
"VALUE" X, ALPHA, SCALE;
"REAL" X, ALPHA, SCALE;
"CODE" 41513;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;
ALPHA: <arithmetic expression>, shape parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier GAMMA.

The following error messages may appear:

Errornumber 2 (if ALPHA ≤ 0)
Errornumber 3 (if SCALE ≤ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The algorithm used is based upon the recurrence formula 6.5.21 from Abramowitz and Stegun (1970), which shows that any GAMMA integral can be reduced to the sum of:

1) A series of terms which can be directly evaluated, and

2) A GAMMA integral with: 1 < ALPHA < 2, which is evaluated by pade approximations.
For ALPHA > 500, a normal approximation is used.

The precision of the computation depends on the parameters ALPHA and SCALE. For moderate values of the parameters the precision is 10^{-10}.

REFERENCE

[1] M. Abramowitz and I.A. Stegun: *Handbook of mathematical functions*, Dover Publications, New York, 1970.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        GAMMA(  20,  10,  3),
        GAMMA( 105, 6.8, 21),
        GAMMA(2000, 100, 25))
"END"
```

Output:

```
.137372
.263863
.017108
```

SOURCE TEXT

```
"CODE" 41513;
"REAL" "PROCEDURE" GAMMA(X, ALPHA, SCALE);
    "VALUE" X, ALPHA, SCALE; "REAL" X, ALPHA, SCALE;
"BEGIN" "INTEGER" DELTA, UPP;
    "REAL" BETA, START, SUM, TERM;

    "REAL" "PROCEDURE" INCGAM(X, A, EPS);
    "VALUE" X, A, EPS; "REAL" X, A, EPS;
    "BEGIN" "REAL" C0, C1, C2, D0, D1, D2, X2, AX, P,
        Q, R, S, R1, R2, SCF; "INTEGER" N;
        S:= EXP(-X + A * LN(X)); SCF:= "+300;
        "IF" X <= 1 "THEN"
        "BEGIN" X2:= X * X; AX:= A * X;
            D0:= 1; P:= A; C0:= S;
            D1:= (A + 1) * (A + 2 - X);
            C1:= (D1 + AX + 2 * X) * S;
            R2:= C1 / D1;
            "FOR" N:= 1, N + 1
            "WHILE" ABS((R2 - R1) / R2) > EPS "DO"
            "BEGIN" P:= P + 2;
                Q:= (P + 1) * (P * (P + 2) - AX);
                R:= N * (N + A) * (P + 2) * X2;
                C2:= (Q * C1 + R * C0) / P;
                D2:= (Q * D1 + R * D0) / P;
```

```
                R1:= R2; R2:= C2 / D2;
                C0:= C1; C1:= C2; D0:= D1; D1:= D2;
                "IF" ABS(C1) > SCF "OR" ABS(D1) > SCF "THEN"
                "BEGIN" C0:= C0 / SCF; C1:= C1 / SCF;
                    D0:= D0 / SCF; D1:= D1 / SCF
                "END"
            "END"; INCGAM:= R2 / A / EXP(LOGGAMMA(A))
        "END" "ELSE"
        "BEGIN" C0:= A * S; C1:= (1 + X) * C0;
            Q:= X + 2 - A;
            D0:= X; D1:= X * Q; R2:= C1 / D1;
            "FOR" N:= 1, N + 1
            "WHILE" ABS((R2 - R1) / R2) > EPS "DO"
            "BEGIN" Q:= Q + 2; R:= N * (N + 1 - A);
                C2:= Q * C1 - R * C0; D2:= Q * D1 - R * D0;
                R1:= R2; R2:= C2 / D2;
                C0:= C1; C1:= C2; D0:= D1; D1:= D2;
                "IF" ABS(C1) > SCF "OR" ABS(D1) > SCF "THEN"
                "BEGIN" C0:= C0 / SCF; C1:= C1 / SCF;
                    D0:= D0 / SCF; D1:= D1 / SCF
                "END"
            "END"; INCGAM:= 1 - R2 / A / EXP(LOGGAMMA(A))
        "END"
    "END" INCGAM;

    "IF" ALPHA <= 0 "THEN"
        STATAL3 ERROR("("GAMMA")", 2, ALPHA)
    "ELSE" "IF" SCALE <= 0 "THEN"
        STATAL3 ERROR("("GAMMA")", 3, SCALE)
    "ELSE" "IF" X <= 0 "THEN" GAMMA:= 0 "ELSE"
    "IF" ALPHA >= 500 "THEN"
    "BEGIN"
        GAMMA:= PHI(((X / SCALE / ALPHA) ** .33333333333333
                - 1 + 1 / (9 * ALPHA)) * 3 * SQRT(ALPHA))
    "END" "ELSE"
    "BEGIN" X:= X / SCALE; BETA:= ALPHA - ENTIER(ALPHA) + 1;
        START:= "IF" X >= 40 "THEN" 1 "ELSE"
                INCGAM(X, BETA, "-12);
        "IF" ALPHA < 1 "THEN"
            GAMMA:= START + EXP(-X + ALPHA * LN(X)
                    - LOGGAMMA(ALPHA + 1))
        "ELSE" "IF" ALPHA < 2 "THEN" GAMMA:= START
        "ELSE" "IF" X > 700 "THEN" GAMMA:= 1
        "ELSE"
        "BEGIN" UPP:= ENTIER(ALPHA) - 2; SUM:= TERM:=
            EXP(-X + (ALPHA - 1) * LN(X) - LOGGAMMA(ALPHA));
            "FOR" DELTA:= 1 "STEP" 1 "UNTIL" UPP "DO"
            "BEGIN" TERM:= TERM * (ALPHA - DELTA) / X;
                SUM:= SUM + TERM
            "END";
            GAMMA:= START - SUM
        "END"
    "END"
"END" GAMMA;
      "EOP"
```

TITLE: **Gammainv**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the argument X for which the gamma distribution function has a given value PROB. ALPHA and SCALE are the shape and scale parameters of the distribution.

KEYWORDS

Inverse gamma distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" GAMMAINV (PROB, ALPHA, SCALE);
"VALUE" PROB, ALPHA, SCALE;
"REAL" PROB, ALPHA, SCALE;
"CODE" 41514;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

ALPHA: <arithmetic expression>, shape parameter;

SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the inverse gamma distribution is assigned to the procedure identifier GAMMAINV.

The following error messages may appear:

Errornumber 1	(if PROB $\leqslant 10^{-10}$ or PROB$\geqslant 1-10^{-10}$)
Errornumber 2	(if ALPHA $\leqslant 0$)
Errornumber 3	(if SCALE $\leqslant 0$)

PROCEDURES USED

INVERSE	STATAL 40001
STATAL3 ERROR	STATAL 40100
GAMMA	STATAL 41513

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The inverse distribution function is computed as follows:
GAMMAINV(PROB,ALPHA,SCALE)=INVERSE(X,GAMMA(X,ALPHA,SCALE), PROB,10^{-10}).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(3ZD.6D,/)")",
        GAMMAINV(.372,  10,  3),
        GAMMAINV(.863, 6.8, 21),
        GAMMAINV(.108, 100, 25))
"END"
```

Output:

```
  26.074090
 202.669137
2195.583645
```

SOURCE TEXT

```
"CODE" 41514;
"REAL" "PROCEDURE" GAMMAINV(PROB,ALPHA,SCALE);
"VALUE" PROB,ALPHA,SCALE; "REAL" PROB,ALPHA,SCALE;
"BEGIN" "REAL" X,TOL;

  TOL:= "-10;
  "IF" ALPHA <= 0 "THEN"
      STATAL3 ERROR("("GAMMAINV")",2,ALPHA) "ELSE"
  "IF" SCALE <= 0 "THEN"
      STATAL3 ERROR("("GAMMAINV")",3,SCALE) "ELSE"
  "IF" PROB <= TOL "OR" PROB >= 1 - TOL "THEN"
      STATAL3 ERROR("("GAMMAINV")",1,PROB);
  X:= ALPHA * SCALE;
  GAMMAINV:= INVERSE(X,GAMMA(X,ALPHA,SCALE),PROB,TOL)
"END" GAMMAINV;
        "EOP"
```

TITLE: **Gammadens**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION
The procedure computes the gamma density function for a given argument X. ALPHA and SCALE are the shape and scale parameters of the distribution.

KEYWORDS
Gamma density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" GAMMADENS (X, ALPHA, SCALE);
"VALUE" X, ALPHA, SCALE;
"REAL" X, ALPHA, SCALE;
"CODE" 41756;
```
Formal parameters
X: <arithmetic expression>, argument of the density function;
ALPHA: <arithmetic expression>, shape parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier GAMMADENS.
The following error messages may appear:
Errornumber 2 (if ALPHA ≤ 0)
Errornumber 3 (if SCALE ≤ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
GAMMADENS(X,ALPHA,SCALE)=

```
 ( 0                                  if X <= 0,
 | EXP(-ALPHA*LN(SCALE)-
 { LOGGAMMA(ALPHA)-X/SCALE+
 | (ALPHA-1)*LN(X))                   if X > 0.
 (
```

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        GAMMADENS(7, 5.5, 4),
        GAMMADENS(4, 6.8, 3),
        GAMMADENS(5, 3.2, 2))
"END"
```

Output:

```
 .010298
 .000939
 .127109
```

Source text

```
"CODE" 41756;
"REAL" "PROCEDURE" GAMMADENS(X,ALPHA,SCALE);
"VALUE" X,ALPHA,SCALE; "REAL" X,ALPHA,SCALE;
"IF" X <= 0 "THEN" GAMMADENS:= 0 "ELSE"
"BEGIN"
  "IF" ALPHA <= 0
  "THEN" STATAL3 ERROR("("GAMMADENS")",2,ALPHA)
  "ELSE" "IF" SCALE <= 0
         "THEN" STATAL3 ERROR("("GAMMADENS")",3,SCALE);

  GAMMADENS:=
    EXP(- ALPHA * LN(SCALE) - LOGGAMMA(ALPHA) -
       X / SCALE + (ALPHA - 1) * LN(X))
"END" GAMMADENS;
         "EOP"
```

TITLE: **Beta**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION

The procedure computes the beta distribution function, i.e. the probability that a random variable having a beta distribution with parameters ALPHA1 and ALPHA2 is less than or equal to a given value X.

KEYWORDS

Beta distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" BETA (X, ALPHA1, ALPHA2);
"VALUE" X, ALPHA1, ALPHA2;
"REAL" X, ALPHA1, ALPHA2;
"CODE" 41517;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;
ALPHA1: <arithmetic expression>, first shape parameter;
ALPHA2: <arithmetic expression>, second shape parameter.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier BETA.

The following error messages may appear:

Errornumber 2 (if ALPHA1 $\leqslant$ 0)
Errornumber 3 (if ALPHA2 $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100
INCOMPLETE BETA STATAL 40401

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

BETA(X,ALPHA1,ALPHA2)=INCOMPLETE BETA(X,ALPHA1,ALPHA2,EPS).

EPS, the precision of the incomplete beta function, is 10^{-12}.

The precision is 10^{-12}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        BETA(.5, 5, 5),
        BETA(.1, 2, 3),
        BETA(.6, 1, 1))
"END"
```

Output:

```
 .500000
 .052300
 .600000
```

SOURCE TEXT

```
"CODE" 41517;
"REAL" "PROCEDURE" BETA(X, ALPHA1, ALPHA2);
   "VALUE" X, ALPHA1, ALPHA2; "REAL" X, ALPHA1, ALPHA2;
BETA:= "IF" ALPHA1 <= 0
       "THEN" STATAL3 ERROR("("BETA")", 2, ALPHA1)
       "ELSE" "IF" ALPHA2 <= 0
       "THEN" STATAL3 ERROR("("BETA")", 3, ALPHA2)
       "ELSE" "IF" X <= 0 "THEN" 0
       "ELSE" "IF" X >= 1 "THEN" 1
       "ELSE" INCOMPLETE BETA(X, ALPHA1, ALPHA2, "-12);
         "EOP"
```

TITLE: **Betainv**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the argument X, for which the beta distribution function has a given value PROB. ALPHA1 and ALPHA2 are the parameters of the distribution.

KEYWORDS

Inverse beta distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" BETAINV (PROB, ALPHA1, ALPHA2);
"VALUE" PROB, ALPHA1, ALPHA2;
"REAL" PROB, ALPHA1, ALPHA2;
"CODE" 41518;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

ALPHA1: <arithmetic expression>, first shape parameter;

ALPHA2: <arithmetic expression>, second shape parameter.

DATA AND RESULTS

The value of the inverse beta distribution function is assigned to the procedure identifier BETAINV.

The following error messages may appear:

Errornumber 1	(if PROB $\leq 10^{-10}$ or PROB $\geq 1-10^{-10}$)
Errornumber 2	(if ALPHA1 ≤ 0)
Errornumber 3	(if ALPHA2 ≤ 0)

PROCEDURES USED

INVERSE	STATAL 40001
STATAL3 ERROR	STATAL 40100
BETA	STATAL 41517

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The inverse beta distribution function is computed as follows:
BETAINV(PROB,ALPHA1,ALPHA2)=INVERSE(X,BETA(X,ALPHA1,ALPHA2), PROB,10^{-10}).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        BETAINV(.5  ,  5, 5),
        BETAINV(.523,  2, 3),
        BETAINV(.9  , .5, 5))
"END"
```

Output:

```
 .500000
 .398959
 .247272
```

SOURCE TEXT

```
"CODE" 41518;
"REAL" "PROCEDURE" BETAINV(PROB,ALPHA1,ALPHA2);
"VALUE" PROB,ALPHA1,ALPHA2; "REAL" PROB,ALPHA1,ALPHA2;
"BEGIN" "REAL" X,Y,TOL;

  "COMMENT" DEFINE ACCURACY;
  TOL:= "-10;

  "COMMENT" TEST FOR ADMISSIBILITY OF PARAMETERS;
  "IF" ALPHA1 <= 0 "THEN"
      STATAL3 ERROR("("BETAINV")",2,ALPHA1)
  "ELSE"
  "IF" ALPHA2 <= 0 "THEN"
      STATAL3 ERROR("("BETAINV")",3,ALPHA2)
  "ELSE"
  "IF" PROB <= TOL "OR" PROB >= 1 - TOL "THEN"
    STATAL3 ERROR("("BETAINV")",1,PROB);

  X:= 0;
  BETAINV:= INVERSE(X,BETA(X,ALPHA1,ALPHA2),PROB,TOL)
"END" BETAINV;
        "EOP"
```

TITLE: **Betadens**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the beta density function for a given argument X. ALPHA1 and ALPHA2 are the parameters of the distribution.

KEYWORDS

Beta density function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" BETADENS (X,ALPHA1,ALPHA2);
"VALUE" X,ALPHA1,ALPHA2;
"REAL" X,ALPHA1,ALPHA2;
"CODE" 41760;
```

Formal parameters

X: <arithmetic expression>, argument of the density function;

ALPHA1: <arithmetic expression>, first shape parameter;

ALPHA2: <arithmetic expression>, second shape parameter.

DATA AND RESULTS

The value of the density function is assigned to the procedure identifier BETADENS.

The following error messages may appear:

Errornumber 2	(if ALPHA1 $\leq$ 0)
Errornumber 3	(if ALPHA2 $\leq$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The density function is computed as follows:

```
BETADENS(X,ALPHA1,ALPHA2)=
            BETA*EXP((ALPHA1-1)*LN(X)+(ALPHA2-1)*LN(1-X)),
```

where BETA is the reciprocal of the complete beta function.

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        BETADENS(.5,  5, 5),
        BETADENS(.1,  2, 3),
        BETADENS(.1, .5, 5))
"END"
```

Output:

```
2.460938
 .972000
2.552940
```

SOURCE TEXT

```
"CODE" 41760;
"REAL" "PROCEDURE" BETADENS(X,ALPHA1,ALPHA2);
"VALUE" X,ALPHA1,ALPHA2; "REAL" X,ALPHA1,ALPHA2;
"BEGIN" "REAL" BET;

  "IF" ALPHA1 <= 0
  "THEN" STATAL3 ERROR("("BETADENS")",2,ALPHA1);
  "IF" ALPHA2 <= 0
  "THEN" STATAL3 ERROR("("BETADENS")",3,ALPHA2);

  "IF" X <= 0 "OR" X >= 1 "THEN" BETADENS:= 0 "ELSE"
  "BEGIN" BET:= EXP(LOGGAMMA(ALPHA1 + ALPHA2) -
                LOGGAMMA(ALPHA1) - LOGGAMMA(ALPHA2));
      BETADENS:= BET * EXP((ALPHA1 - 1) * LN(X) +
                  (ALPHA2 - 1 ) * LN(1 - X))
  "END";
"END" BETADENS;
        "EOP"
```

TITLE: **Cauchy**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION

The procedure computes the Cauchy distribution function, i.e. the probability that a random variable having a Cauchy distribution is less than or equal to a given value X. The location parameter LOC is the median of the distribution.

KEYWORDS

Cauchy distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" CAUCHY (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41541;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;
LOC: <arithmetic expression>, location parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier Cauchy.
The following error message may appear:
Errornumber 3 (if SCALE $\leq$ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

```
CAUCHY(X,LOC,SCALE)=.5+ARCTAN((X-LOC)/SCALE)/π.
```

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        CAUCHY(10,  0, 1    ),
        CAUCHY( 0,  0, 1    ),
        CAUCHY( 0, 10, 0.002))
"END"
```

Output:

```
 .968274
 .500000
 .000064
```

SOURCE TEXT

```
"CODE" 41541;
"REAL" "PROCEDURE" CAUCHY(X, LOC, SCALE);
"VALUE" X, LOC, SCALE; "REAL" X, LOC, SCALE;
CAUCHY:= "IF" SCALE <= 0
    "THEN" STATAL3 ERROR("("CAUCHY")", 3, SCALE)
    "ELSE" ARCTAN((X - LOC) / SCALE) * .31830988618379 + .5;
        "EOP"
```

TITLE: **Cauchyinv**

AUTHOR: E. Opperdoes

INSTITUTE: Mathematical Centre

RECEIVED: 751001

BRIEF DESCRIPTION

The procedure computes the argument X, for which the Cauchy distribution function has a given value PROB. The location parameter LOC is the median of the distribution.

KEYWORDS

Inverse Cauchy distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" CAUCHYINV (PROB, LOC, SCALE);
"VALUE" PROB, LOC, SCALE;
"REAL" PROB, LOC, SCALE;
"CODE" 41542;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

LOC: <arithmetic expression>; location parameter;

SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the inverse Cauchy distribution function is assigned to the procedure identifier CAUCHYINV.

The following error messages may appear:

Errornumber 1	(if PROB $\leqslant 0$ or PROB $\geqslant 1$)
Errornumber 3	(if SCALE $\leqslant 0$)

PROCEDURES USED

STATAL3 ERROR   STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The inverse distribution function is computed as follows:

```
CAUCHYINV=LOC-SCALE*COS(π*PROB)/SIN(π*PROB).
```

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        CAUCHYINV(.35,   0,   1),
        CAUCHYINV(.05,   2, 1.3),
        CAUCHYINV(.93, 0.1,   1))
"END"
```

Output:

```
 -0.509525
 -6.207877
 +4.573743
```

Source text

```
"CODE" 41542;
"REAL" "PROCEDURE" CAUCHYINV (PROB, LOC, SCALE);
"VALUE" PROB, LOC, SCALE; "REAL" PROB, LOC, SCALE;
"BEGIN" "REAL" ARG;
    ARG:= 3.1415 92653 5898 * PROB;
    CAUCHYINV:= "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
            STATAL3 ERROR ("("CAUCHYINV")", 1, PROB)
        "ELSE" "IF" SCALE <= 0 "THEN"
            STATAL3 ERROR ("("CAUCHYINV")", 3, SCALE)
        "ELSE"
            -SCALE * COS (ARG) / SIN (ARG) + LOC
"END" CAUCHYINV;
         "EOP"
```

TITLE: **Cauchydens**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION
The procedure computes the Cauchy density function for a given argument X.
LOC and SCALE are the parameters of the distribution.

KEYWORDS
Cauchy density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" CAUCHYDENS (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41763;
```
Formal parameters

X:	<arithmetic expression>, argument of the density function;
LOC:	<arithmetic expression>, location parameter;
SCALE:	<arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier CAUCHYDENS.
The following error message may appear:
Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
CAUCHYDENS(X,LOC,SCALE)=1/(π*SCALE*(1+((X−LOC)/SCALE)^2)).

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        CAUCHYDENS( 1, 0, 1),
        CAUCHYDENS( 2, 0, 1),
        CAUCHYDENS(10, 0, 1))
"END"
```

Output:

```
 .159155
 .063662
 .003152
```

SOURCE TEXT

```
"CODE" 41763;
"REAL" "PROCEDURE" CAUCHYDENS(X,LOC,SCALE);
"VALUE" X,LOC,SCALE; "REAL" X,LOC,SCALE;
"BEGIN" "REAL" PI,Q;

  "IF" SCALE <= 0
  "THEN" STATAL3 ERROR("("CAUCHYDENS")",3,SCALE);
  PI:= 3.1415926535898;
  Q:= (X - LOC) / SCALE;
  CAUCHYDENS:= 1 / (PI * SCALE * (1 + Q * Q))
"END" CAUCHYDENS;
         "EOP"
```

TITLE: **Weibull**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 760901

BRIEF DESCRIPTION

The procedure computes the Weibull distribution function, i.e the probability that a random variable with a Weibull distribution is less than or equal to a given value X. LOC,SCALE and ALPHA are the parameters of the distribution.

KEYWORDS

Weibull distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" WEIBULL (X, LOC, SCALE, ALPHA);
"VALUE" X, LOC, SCALE, ALPHA;
"REAL" X, LOC, SCALE, ALPHA;
"CODE" 41545;
```

Formal parameters

X:	<arithmetic expression>, argument of the distribution function;
LOC:	<arithmetic expression>, location parameter;
SCALE:	<arithmetic expression>, scale parameter;
ALPHA:	<arithmetic expression>, exponent.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier WEIBULL.

The following error messages may appear:

Errornumber 3	(if SCALE $\leq$ 0)
Errornumber 4	(if ALPHA $\leq$ 0)

PROCEDURES USED

STATAL3 ERROR   STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

WEIBULL(X,LOC,SCALE,ALPHA) =

$$\begin{cases} 0 & \text{if } X \leq LOC, \\ 1-EXP(-((X-LOC)/SCALE)^{ALPHA}) & \text{if } X > LOC. \end{cases}$$

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        WEIBULL( 4,  1, 1.0, 2),
        WEIBULL( 2,  1, 0.5, 1),
        WEIBULL(10, 10, 0.1, 5))
"END"
```

Output:

```
 .999877
 .864665
 .000000
```

SOURCE TEXT

```
"CODE" 41545;
"REAL" "PROCEDURE" WEIBULL(X, LOC, SCALE, ALPHA);
"VALUE" X, LOC, SCALE, ALPHA; "REAL" X, LOC, SCALE, ALPHA;
WEIBULL:= "IF" SCALE <= 0
          "THEN" STATAL3 ERROR("("WEIBULL")", 3, SCALE)
          "ELSE" "IF" ALPHA <= 0
          "THEN" STATAL3 ERROR("("WEIBULL")", 4, ALPHA)
          "ELSE" "IF" X <= LOC "THEN" 0
          "ELSE" 1 - EXP(-((X - LOC) / SCALE) ** ALPHA);
         "EOP"
```

TITLE: **Weibullinv**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the argument X, for which the Weibull distribution function has a given value PROB. LOC,SCALE and ALPHA are the parameters of the distribution.

KEYWORDS

Inverse Weibull distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" WEIBULLINV (PROB, LOC, SCALE, ALPHA);
"VALUE" PROB, LOC, SCALE, ALPHA;
"REAL" PROB, LOC, SCALE, ALPHA;
"CODE" 41546;
```

Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;

LOC: <arithmetic expression>, location parameter;

SCALE: <arithmetic expression>, scale parameter;

ALPHA: <arithmetic expression>, exponent.

DATA AND RESULTS

The value of the inverse distribution function is assigned to the procedure identifier WEIBULLINV.

The following error messages may appear:

Errornumber 1 (if PROB $\leqslant 10^{-10}$ or PROB $\geqslant 1-10^{-10}$)

Errornumber 3 (if SCALE $\leqslant 0$)

Errornumber 4 (if ALPHA $\leqslant 0$)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

Method and performance

The inverse distribution function is computed as follows:
WEIBULLINV(PROB,LOC,SCALE,ALPHA)=LOC+SCALE*(−LN(1−PROB))$^{(1/ALPHA)}$.

The precision is 10^{-10}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(ZD.6D,/)")",
        WEIBULLINV(.987, 1, 1 , 2  ),
        WEIBULLINV(.466, 1, .5, 1  ),
        WEIBULLINV(.812, 2, 2 , 1.5))
"END"
```

Output:

```
 3.083940
 1.313680
 4.816665
```

Source text

```
"CODE" 41546;
"REAL" "PROCEDURE" WEIBULLINV(PROB,LOC,SCALE,ALPHA);
"VALUE" PROB,LOC,SCALE,ALPHA; "REAL" PROB,LOC,SCALE,ALPHA;
"BEGIN"
  "IF" PROB <= "-10 "OR" PROB >= 1 - "-10 "THEN"
    STATAL3 ERROR("("WEIBULLINV")",1,PROB) "ELSE"
  "IF" SCALE <= 0 "THEN"
      STATAL3 ERROR("("WEIBULLINV")",3,SCALE)
  "ELSE"
  "IF" ALPHA <= 0 "THEN"
      STATAL3 ERROR("("WEIBULLINV")",4,ALPHA);

  WEIBULLINV:= LOC + SCALE * (-LN(1 - PROB)) ** (1 / ALPHA)
"END" WEIBULLINV;
         "EOP"
```

TITLE: **Weibulldens**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION
The procedure computes the Weibull density function for a given argument X. LOC,SCALE and ALPHA are the parameters of the distribution.

KEYWORDS
Weibull density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" WEIBULLDENS (X, LOC, SCALE, ALPHA);
"VALUE" X, LOC, SCALE, ALPHA;
"REAL" X, LOC, SCALE, ALPHA;
"CODE" 41759;
```
Formal parameters

X:	<arithmetic expression>, argument of the density function;
LOC:	<arithmetic expression>, location parameter;
SCALE:	<arithmetic expression>, scale parameter;
ALPHA:	<arithmetic expression>, exponent.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier WEIBULLDENS.
The following error messages may appear:

Errornumber 3	(if SCALE $\leqslant$ 0)
Errornumber 4	(if ALPHA $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
WEIBULLDENS(X,LOC,SCALE,ALPHA)=

$$\begin{cases} 0 & \text{if } X \leqslant LOC, \\ \text{(ALPHA/SCALE)*EXP((ALPHA-1)*LN(Y)-Y**ALPHA)} & \text{if } X > LOC, \end{cases}$$

where Y=(X−LOC)/SCALE.

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        WEIBULLDENS(4, 1,  1,   2),
        WEIBULLDENS(2, 1, .5,   2),
        WEIBULLDENS(3, 2,  2, 1.5))
"END"
```

Output:

```
 .000740
 .146525
 .372392
```

SOURCE TEXT

```
"CODE" 41759;
"REAL" "PROCEDURE" WEIBULLDENS(X,LOC,SCALE,ALPHA);
"VALUE" X,LOC,SCALE,ALPHA; "REAL" X,LOC,SCALE,ALPHA;
"BEGIN"
  "IF" SCALE <= 0
  "THEN" STATAL3 ERROR("("WEIBULLDENS")",3,SCALE) "ELSE"
  "IF" ALPHA <= 0
  "THEN" STATAL3 ERROR("("WEIBULLDENS")",4,ALPHA);

  WEIBULLDENS:= "IF" X <= LOC "THEN" 0 "ELSE"
    (ALPHA / SCALE) * EXP((ALPHA - 1) *
      LN((X - LOC) / SCALE) - ((X - LOC) / SCALE) ** ALPHA)
"END" WEIBULLDENS;
         "EOP"
```

TITLE: **Laplace**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION

The procedure computes the Laplace distribution function, i.e. the probability that a random variable with a Laplace distribution is less than or equal to a given value X. The location parameter LOC is the mean of the distribution, the scale parameter SCALE is proportional to the standard deviation.

KEYWORDS

Laplace distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" LAPLACE (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41565;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

LOC: <arithmetic expression>, location parameter;

SCALE: <arithmetic expression>, scale parameter;

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier LAPLACE.

The following error message may appear:

Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The distribution function is computed as follows:

LAPLACE(X,LOC,SCALE)=

$$\begin{cases} 1-\mathrm{EXP}(-(X-\mathrm{LOC})/\mathrm{SCALE})/2 & \text{if } X > \mathrm{LOC}, \\ \mathrm{EXP}(+(X-\mathrm{LOC})/\mathrm{SCALE})/2 & \text{if } X \leqslant \mathrm{LOC}. \end{cases}$$

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        LAPLACE( 100, 100, 100),
        LAPLACE(   1,   0,   1),
        LAPLACE(-100,   0,  55))
"END"
```

Output:

```
 .500000
 .816060
 .081160
```

SOURCE TEXT

```
"CODE" 41565;
"REAL" "PROCEDURE" LAPLACE(X, MU, SIGMA);
"VALUE" X, MU, SIGMA; "REAL" X, MU, SIGMA;
"BEGIN"
    "IF" SIGMA <= 0 "THEN"
        STATAL3 ERROR("("LAPLACE")", 3, SIGMA);
    X:= (X - MU) / SIGMA;
    LAPLACE:= .5 * (1 + (1 - EXP(-ABS(X))) * SIGN(X))
"END" LAPLACE;
         "EOP"
```

TITLE: **Laplaceinv**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 770301

BRIEF DESCRIPTION
The procedure computes the argument X. for which the Laplace distribution function has a given value PROB. The location parameter LOC is the mean of the distribution, the scale parameter SCALE is proportional to the standard deviation.

KEYWORDS
Inverse Laplace distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" LAPLACEINV (PROB, LOC, SCALE);
"VALUE" PROB, LOC, SCALE;
"REAL" PROB, LOC, SCALE;
"CODE" 41566;
```
Formal parameters

PROB: <arithmetic expression>, left hand tail probability of the value to be computed;
LOC: <arithmetic expression>, location parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the inverse distribution function is assigned to the procedure identifier LAPLACEINV.
The following error messages may appear:
Errornumber 1 (if PROB $\leqslant$ 0 or PROB $\geqslant$ 1)
Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The inverse distribution function is computed as follows:

LAPLACEINV(PROB,LOC,SCALE)=

$$\begin{cases} \text{LOC+LN(2*PROB)*SCALE} & \text{if PROB} \leq \frac{1}{2}, \\ \text{LOC−LN(2*(1−PROB))*SCALE} & \text{if } \frac{1}{2} < \text{PROB} < 1. \end{cases}$$

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        LAPLACEINV(.546, 5.5, 1.0),
        LAPLACEINV(.872, 4.3, 2.2),
        LAPLACEINV(.930, 1.2, 3.5))
"END"
```

Output:

```
 +5.596511
 +7.297671
 +8.081395
```

SOURCE TEXT

```
"CODE" 41566;
"REAL" "PROCEDURE" LAPLACEINV(PROB,LOC,SCALE);
"VALUE" PROB, LOC, SCALE; "REAL" PROB, LOC, SCALE;
"BEGIN"
    "IF" SCALE  <= 0 "THEN"
        STATAL3 ERROR("("LAPLACEINV")",3,SCALE);
    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
        STATAL3 ERROR("("LAPLACEINV")",1,PROB);
    "IF" PROB <= .5 "THEN"
        LAPLACEINV :=  LOC + LN(2 * PROB) * SCALE
    "ELSE"
        LAPLACEINV :=  LOC - LN(2 * (1 - PROB)) * SCALE;
"END" LAPLACEINV;
"EOP"
```

TITLE: **Laplacedens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 770301

BRIEF DESCRIPTION

The procedure computes the density function of the Laplace distribution for a given argument X. The location parameter LOC is the mean of the distribution, the scale parameter SCALE is proportional to the standard deviation.

KEYWORDS

Laplace density function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" LAPLACEDENS (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41764;
```

Formal parameters

X: <arithmetic expression>, argument of the density function;
LOC: <arithmetic expression>, location parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the density function is assigned to the procedure identifier LAPLACEDENS.

The following error message may appear:

Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The density function is computed as follows:

```
LAPLACEDENS(X,LOC,SCALE)=EXP(-ABS((X-LOC)/SCALE)/(2*SCALE).
```

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        LAPLACEDENS(  .1, 1.0, 5.3),
        LAPLACEDENS( 2.0, 2.0, 8.9),
        LAPLACEDENS(15.9, 4.5, 7.7))
"END"
```

Output:

```
 .079606
 .056180
 .014774
```

SOURCE TEXT

```
"CODE" 41764;
"REAL" "PROCEDURE" LAPLACEDENS(X,LOC,SCALE);
    "VALUE" X, LOC, SCALE; "REAL" X, LOC, SCALE;
"BEGIN"
    "IF" SCALE <= 0
    "THEN" STATAL3 ERROR("("LAPLACEDENS")",3,SCALE);
    X := (X - LOC) / SCALE;
    LAPLACEDENS := .5 / SCALE * EXP(-ABS(X));
"END" LAPLACEDENS;
"EOP"
```

TITLE: **Erlang**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 750201

BRIEF DESCRIPTION
The procedure computes the Erlang distribution function, i.e. the probability that a random variable having an Erlang distribution with parameters ALPHA and SCALE is less than or equal to a given value X.

KEYWORDS
Erlang distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" ERLANG (X, ALPHA, SCALE);
"VALUE" X, ALPHA, SCALE;
"REAL" X, ALPHA, SCALE;
"CODE" 41563;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
ALPHA: <arithmetic expression>, shape parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier ERLANG.
The following error messages may appear:
Errornumber 2 (if ALPHA is not an integer > 0)
Errornumber 3 (if SCALE ≤ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100
GAMMA STATAL 41513

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed as follows:
ERLANG(X,ALPHA,SCALE)=GAMMA(X,ALPHA,1/SCALE).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        ERLANG(2,  2, 0.5),
        ERLANG(1, 10, 2.5),
        ERLANG(2,  1, 1.0))
"END"
```

Output:

```
 .908422
 .000000
 .864665
```

SOURCE TEXT

```
"CODE" 41563;
"REAL" "PROCEDURE" ERLANG(X, N, SCALE);
"VALUE" X, N, SCALE; "REAL" X, N, SCALE;
ERLANG:= "IF" SCALE <= 0
         "THEN" STATAL3 ERROR("("ERLANG")", 3, SCALE)
         "ELSE" "IF" N <= 0 "OR" ENTIER(N) < N
         "THEN" STATAL3 ERROR("("ERLANG")", 2, N)
         "ELSE" GAMMA(X, N, SCALE);
         "EOP"
```

TITLE: **Erlanginv**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770301

BRIEF DESCRIPTION
The procedure computes the argument X, for which the Erlang distribution function with parameters "ALPHA" and SCALE has a given value PROB.

KEYWORDS
Inverse Erlang distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" ERLANGINV (PROB, ALPHA, SCALE);
"VALUE" PROB, ALPHA, SCALE;
"REAL" PROB, ALPHA, SCALE;
"CODE" 41564;
```
Formal parameters
PROB: <arithmetic expression>, left hand tail probability of the value to be computed;
ALPHA: <arithmetic expression>, shape parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the inverse distribution function is assigned to the procedure identifier ERLANGINV.
The following error messages may appear:

Errornumber 1	(if PROB $\leq 10^{-10}$ or PROB $\geq 1-10^{-10}$)
Errornumber 2	(if ALPHA is not an integer > 0)
Errornumber 3	(if SCALE ≤ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
GAMMAINV	STATAL 41514

LANGUAGE
Algol 60

METHOD AND PERFORMANCE

The inverse distribution function is computed as follows:
ERLANGINV(PROB,ALPHA,SCALE)=GAMMAINV(PROB,ALPHA,1/SCALE).

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+ZD.6D,/)")",
        ERLANGINV(.384, 1, 3.5),
        ERLANGINV(.119, 2, 1.5),
        ERLANGINV(.508, 7, 2.0))
"END"
```

Output:

```
 +1.695779
 +0.886744
+13.442688
```

SOURCE TEXT

```
"CODE" 41564;
"REAL" "PROCEDURE" ERLANGINV(PROB,ALPHA,SCALE);
"VALUE" PROB,ALPHA,SCALE; "REAL" PROB,ALPHA,SCALE;
"IF" PROB <= "-10 "OR" PROB >= 1 - "-10 "THEN"
    STATAL3 ERROR("("ERLANGINV")",1,PROB)
"ELSE" "IF" ALPHA <= 0 "OR" ENTIER(ALPHA) < ALPHA "THEN"
    STATAL3 ERROR("("ERLANGINV")",2,ALPHA)
"ELSE" "IF" SCALE <= 0 "THEN"
    STATAL3 ERROR("("ERLANGINV")",3,SCALE)
"ELSE" ERLANGINV:= GAMMAINV(PROB,ALPHA, SCALE);
         "EOP"
```

TITLE: **Erlangdens**

AUTHOR: I. van der Tweel

INSTITUTE: Mathematical Centre

RECEIVED: 770301

BRIEF DESCRIPTION

The procedure computes the density function of the Erlang distribution with parameters ALPHA and SCALE for a given argument X.

KEYWORDS

Erlang density function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" ERLANGDENS (X, ALPHA, SCALE);
"VALUE" X, ALPHA, SCALE;
"REAL" X, ALPHA, SCALE;
"CODE" 41757;
```

Formal parameters

X: <arithmetic expression>, argument of the density function;
ALPHA: <arithmetic expression>, shape parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS

The value of the density function is assigned to the procedure identifier ERLANGDENS.

The following error messages may appear:

Errornumber 2 (if ALPHA is not an integer > 0)
Errornumber 3 (if SCALE ⩽ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

The density function is computed as follows:

```
ERLANGDENS(X,ALPHA,SCALE)=
  0                                              if X ⩽ 0,
  EXP(-ALPHA*LN(SCALE)-LOGGAMMA(ALPHA)
  -X/SCALE+(ALPHA-1)*LN(X))                      if X > 0.
```

The precision is 10^{-10}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        ERLANGDENS(2, 8, 1),
        ERLANGDENS(3, 7, 2),
        ERLANGDENS(8, 1, 4))
"END"
```

Output:

```
 .003437
 .001765
 .033834
```

SOURCE TEXT

```
"CODE" 41757;
"REAL" "PROCEDURE" ERLANGDENS(X,ALPHA,SCALE);
"VALUE" X,ALPHA,SCALE; "REAL" X,ALPHA,SCALE;
"IF" X < 0
"THEN" STATAL3 ERROR("("ERLANGDENS")",1,X) "ELSE"
"IF" ALPHA <= 0 "OR" ENTIER(ALPHA) < ALPHA "THEN"
STATAL3 ERROR("("ERLANGDENS")",2,ALPHA) "ELSE"
"IF" SCALE <= 0
"THEN" STATAL3 ERROR("("ERLANGDENS")",3,SCALE)
"ELSE" ERLANGDENS:= "IF" X = 0 "THEN" 0 "ELSE"
EXP(-ALPHA * LN(SCALE) - LOGGAMMA(ALPHA) - X / SCALE +
    (ALPHA - 1) * LN(X));
         "EOP"
```

TITLE: **Extval**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION
The procedure computes the extreme value distribution function, i.e. the probability that a random variable with a type 1 extreme value distribution, is less than or equal to a given value X. LOC and SCALE are the parameters of the distribution.

KEYWORDS
Extreme value distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" EXTVAL (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41571;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
LOC: <arithmetic expression>, location parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier EXTVAL.
The following error message may appear:
Errornumber 3 (if SCALE $\leq$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The distribution function is computed as follows:
EXTVAL(X,LOC,SCALE)=EXP(−EXP(−(X−LOC)/SCALE)).

The precision is 10^{-14}.

REFERENCE

[1] N.L. Johnson & S. Kotz: *Continuous univariate distributions - 1*, Houghton Mifflin Company, Boston, 1970.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        EXTVAL(  5,   0,  1),
        EXTVAL(100, 100, 10),
        EXTVAL( 55,  45,  3))
"END"
```

Output:

```
 .993285
 .367879
 .964955
```

SOURCE TEXT

```
"CODE" 41571;
"REAL" "PROCEDURE" EXTVAL(X, LOC, SCALE);
"VALUE" X, LOC, SCALE; "REAL" X, LOC, SCALE;
"BEGIN"
    "IF" SCALE <= 0 "THEN"
        STATAL3 ERROR("("EXTVAL")", 3, SCALE);
    X:= -(X - LOC) / SCALE;
    EXTVAL:= "IF" X > LN(-LN(MINREAL))
                "THEN" 0
                "ELSE" EXP(-EXP(X));
"END" EXTVAL;
"EOP"
```

TITLE: **Extvalinv**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 770301

BRIEF DESCRIPTION
The procedure computes the argument X, for which the type 1 extreme value distribution with parameters LOC and SCALE has a given value PROB.

KEYWORDS
Inverse Extreme value distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" EXTVALINV (PROB, LOC, SCALE;
"VALUE" PROB, LOC, SCALE;
"REAL" PROB, LOC, SCALE;
"CODE" 41572;
```
Formal parameters
PROB: <arithmetic expression>, left hand tail probability of the value to be computed;
LOC: <arithmetic expression>, location parameter;
SCALE: <arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the inverse distribution function is assigned to the procedure identifier EXTVALINV.
The following error messages may appear:
Errornumber 1 (if PROB $\leqslant$ 0 or PROB $\geqslant$ 1)
Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The inverse distribution function is computed as follows:
EXTVALINV(PROB,LOC,SCALE)=−SCALE*LN(−LN(PROB))+LOC.

The precision is 10^{-14}.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(+2ZD.6D,/)")",
        EXTVALINV(.747,   3,  5),
        EXTVALINV(.685,  14,  8),
        EXTVALINV(.332, 189, 13))
"END"
```

Output:

```
  +9.160317
 +21.775771
+187.730037
```

Source text

```
"CODE" 41572;
"REAL" "PROCEDURE" EXTVALINV(PROB, LOC, SCALE);
"VALUE" PROB, LOC, SCALE; "REAL" PROB, LOC, SCALE;
"BEGIN"
    "IF" SCALE <= 0 "THEN"
      STATAL3 ERROR("("EXTVALINV")", 3, SCALE);
    "IF" PROB <= 0 "OR" PROB >= 1 "THEN"
      STATAL3 ERROR("("EXTVALINV")", 1, PROB);
    EXTVALINV:= -SCALE * LN(-LN(PROB)) + LOC
"END" EXTVALINV;
         "EOP"
```

TITLE: **Extvaldens**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 770301

BRIEF DESCRIPTION
The procedure computes the density function of the type 1 extreme value distribution with parameters LOC and SCALE for a given argument X.

KEYWORDS
Extreme value density function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" EXTVALDENS (X, LOC, SCALE);
"VALUE" X, LOC, SCALE;
"REAL" X, LOC, SCALE;
"CODE" 41766;
```
Formal parameters

X:	<arithmetic expression>, argument of the density function;
LOC:	<arithmetic expression>, location parameter;
SCALE:	<arithmetic expression>, scale parameter.

DATA AND RESULTS
The value of the density function is assigned to the procedure identifier EXTVALDENS.
The following error message may appear:
Errornumber 3 (if SCALE $\leqslant$ 0)

PROCEDURES USED
STATAL3 ERROR STATAL 40100

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
The density function is computed as follows:
EXTVALDENS(X,LOC,SCALE)=EXP(−(Y+EXP(−Y)))/SCALE,
where Y=(X−LOC)/SCALE.

The precision is 10^{-14}.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        EXTVALDENS( 8, 4, 1),
        EXTVALDENS( 9, 3, 2),
        EXTVALDENS(14, 7, 8))
"END"
```

Output:

```
 .017983
 .023685
 .034345
```

SOURCE TEXT

```
"CODE" 41766;
"REAL" "PROCEDURE" EXTVALDENS(X, LOC, SCALE);
    "VALUE" X, LOC, SCALE; "REAL" X, LOC, SCALE;
"BEGIN"
    "IF" SCALE <= 0 "THEN"
    STATAL3 ERROR("("EXTVALDENS")", 3, SCALE);
    X:= (X - LOC) / SCALE;
    EXTVALDENS:= EXP(-(X + EXP(-X))) / SCALE
"END" EXTVALDENS;
          "EOP"
```

TITLE: **Studrange**

AUTHOR: J.M. Buhrman

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the probability that a studentized range statistic (of which the numerator is the range of a sample of size N from the standard normal distribution and the denominator is the square root of an independent χ^2 distributed random variable with DF degrees of freedom, divided by DF) is less than or equal to a given value X.

KEYWORDS

Studentized range distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" STUDRANGE (X, N, DF);
"VALUE" X, N, DF;
"REAL" X, N, DF;
"CODE" 41560;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;

N: <arithmetic expression>, sample size associated with the numerator of the studentized range statistic;

DF: <arithmic expression>, number of degrees of freedom associated with the denominator of the studentized range statistic.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier STUDRANGE.

The following error messages may appear:

Errornumber 2 (if N is not an integer > 1)

Errornumber 3 (if DF $\leqslant$ 0)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
LOGGAMMA	STATAL 40400
PHI	STATAL 41500
QADRAT	NUMAL 32070

Language
Algol 60

Method and performance
The distribution function is computed by integration of the density, which is obtained by integration. The formula used can be found in the references below.

The precision is 10^{-4}.

Due to the repeated integration each call of STUDRANGE requires much time (5 - 10 cps).

References

[1] H.L. Harter, D.S. Clemm and E.H. Guthrie: *The probability integrals of the range and of the studentized range, 1 and 2,* Wright Air Development Center, Ohio, Techn. Report 58 - 484 (1959).

[2] E.S Pearson and H.O. Hartley: *Biometrika tables for statisticians,* (1969) p.43, p.53

[3] Joyce M. May: *Extended and corrected tables of the upper percentage point of the 'studentized' range.* Biometrica 39 (1952), p.192 - 193.

Example of use

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.4D,/)")",
        STUDRANGE(4.0,  2,  1),
        STUDRANGE(5.0, 50,  2),
        STUDRANGE(0.1,  2, 40))
"END"
```

Output:

```
 .7837
 .4499
 .0560
```

SOURCE TEXT

```
"CODE" 41560;
"REAL" "PROCEDURE" STUDRANGE(Q,N,NU);
"VALUE" Q,N,NU; "REAL" Q,N,NU;
"BEGIN" "REAL" X, PI, LN4, LNSQRT2PI, LNSQRTPI4;
    "ARRAY" E[1 : 3];

    "REAL" "PROCEDURE" POWER(X)TO:(N); "VALUE" X, N;
    "REAL" X; "INTEGER" N;
    "BEGIN" "INTEGER" N2; "REAL" Y;
        Y:= 1;
    WHILE POS N:
        "IF" N <= 0 "THEN" "GOTO" END WHILE POS N;
        N2:= N // 2;
      WHILE EVEN N:
        "IF" N2 * 2 ^= N "THEN" "GOTO" END WHILE EVEN N;
        N:= N2; X:= X * X; N2:= N // 2;
        "GOTO" WHILE EVEN N;
      END WHILE EVEN N:
        N:= N - 1; Y:= Y * X;
        "GOTO" WHILE POS N;
    END WHILE POS N:
        POWER:= Y
    "END" POWER;

    "REAL" "PROCEDURE" RANGE(T,N); "VALUE" T,N; "REAL" T,N;
    "BEGIN" "REAL" U; "REAL" "ARRAY" E[1:3];
      E[1]:= E[2]:= "-7;
      RANGE:= N * QADRAT(U, -5, +5,
        PHIDENS(U) * POWER(PHI(U + T) - PHI(U), N - 1), E);
    "END" RANGE;

    "REAL" "PROCEDURE" INTEGRAND(X); "VALUE" X; "REAL" X;
    "BEGIN" "REAL" XQ;
        XQ:= X / Q;
        INTEGRAND:= EXP(NU * (LN4 + LN(XQ) - LNSQRT2PI
            - XQ * XQ / 2)) * (1 - RANGE(X, N)) / X;
    "END" INTEGRAND;

    "IF" N < 2 "OR" ENTIER(N) < N "THEN"
      STATAL ERROR("("STUDRANGE")", 2, N);
    "IF" NU < 1 "OR" ENTIER(NU) < NU "THEN"
      STATAL ERROR("("STUDRANGE")", 3, NU);

    E[1]:= E[2]:= "-6; PI:= ARCTAN(1) * 4;
    LNSQRT2PI:= .5 * LN(2 * PI); LN4:= LN(4);
    LNSQRTPI4:= .5 * LN(PI) - LN4;
    STUDRANGE:= "IF" Q <= 0 "THEN" 0 "ELSE"
      1 - 2 * EXP(NU * (LN(NU) / 2 + LNSQRTPI4)
      - LOGGAMMA(NU / 2)) *
      QADRAT(X, "-6, Q * 7, INTEGRAND(X), E);
"END" STUDRANGE;
        "EOP"
```

TITLE: **Binorcor**

AUTHOR: J.M. Buhrman

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION
The procedure computes the probability that the correlation coefficient in a sample of size N drawn from a bivariate normal distribution with correlation coefficient RHO, is less than or equal to a given value X.

KEYWORDS
Sample correlation coefficient distribution function

CALLING SEQUENCE
Heading
```
"REAL" "PROCEDURE" BINORCOR (X, RHO, N);
"VALUE" X, RHO, N;
"REAL" X, RHO, N;
"CODE" 41569;
```
Formal parameters
X: <arithmetic expression>, argument of the distribution function;
RHO: <arithmetic expression>, correlation coefficient of the normal distribution from which the sample is drawn;
N: <arithmetic expression>, sample size.

DATA AND RESULTS
The value of the distribution function is assigned to the procedure identifier BINORCOR.
The following error messages may appear:
Errornumber 2 (if ABS(RHO) ⩾ 1)
Errornumber 3 (if N is not an integer > 2)

PROCEDURES USED

STATAL3 ERROR	STATAL 40100
PHI	STATAL 41500
STUDENT	STATAL 41530
QADRAT	NUMAL 32070

LANGUAGE
Algol 60

METHOD AND PERFORMANCE
If RHO = 0 the distribution function is computed exactly using the Student distribution.
If RHO ≠ 0 and
- 3 ≤ N ≤ 8 exact formulae (cf. Garwood, 1939) are used,
- 9 ≤ N ≤ 500 the exact (N < 15) or approximated (N ≥ 15) density is integrated,
- N > 500 normal approximations are used (see all references below).

If ABS(RHO) ≤ .9 and N > 500 the precision is at most 10^{-5}.
For values of RHO close to 1 or -1 the precision is uncertain.

REFERENCES

[1] F. Garwood: *The probability integral of the correlation coefficient in samples from a normal bi-variate population,* Biometrica 25 (1939), p.71 - 78.
[2] N.L. Johnson and S. Kotz: *Continuous unvariate distributions 2, (1970),* p.230 (form. (19) corrected).
[3] H. Hotelling: *New light on the correlation coefficient and its transforms,* Journal of the Royal Statistical Society, Series B, 15 (1953), p.213 - 217.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.4D,/)")",
        BINORCOR(-0.1  ,  0.1  ,  25),
        BINORCOR( .701,   .701, 600),
        BINORCOR( 0    , -0.2  ,  10))
"END"
```

Output:

```
.1687
.4943
.7223
```

SOURCE TEXT

```
"CODE" 41569;
"REAL" "PROCEDURE" BINORCOR(R,RHO,N);
"VALUE" R, RHO, N; "REAL" R, RHO, N;
"BEGIN"
    "REAL" "PROCEDURE" SAMCORBIVNORDEN(R,RHO,N);
    "VALUE" R, RHO, N; "REAL" R, RHO, N;
    "BEGIN"
        "IF" ABS(R) >= 1 "THEN" SAMCORBIVNORDEN := 0
        "ELSE"
        "BEGIN"
            "REAL" W1, W3, Y1, Y2, Y3, Y4, N1, R2, RRHO,
              R2RHO2, W2, PI, RHO2;
            R2 := R * R;
            RHO2 := RHO * RHO;
            RRHO := R * RHO;
            R2RHO2 := R2 * RHO2;
            W1 := SQRT(1 - R2);
            W2 := SQRT(1 - RHO2);
            W3 := SQRT(1 - RHO2 * R2);
            PI := ARCTAN(1) * 4;
            N1 := N - 1;
            "IF" N < 15 "THEN"
            "BEGIN"
              "REAL" "ARRAY" SB[3:N]; "INTEGER" I;
              SB[3] :=
                (1-RHO2) / PI / W1 * (1 + RRHO *
                ARCCOS(-RRHO) / W3) / (1 - R2RHO2);
              "IF" N =3 "THEN" SAMCORBIVNORDEN := SB[3]
              "ELSE"
              "BEGIN"
                SB[4] :=
                  (1 - RHO2) * W2 / PI * (3 * RRHO +
                  (1 + 2 * R2RHO2) * ARCCOS(-RRHO) / W3)
                  / (1 - R2RHO2) / (1 - R2RHO2);
                "IF" N = 4 "THEN" SAMCORBIVNORDEN := SB[4]
                "ELSE"
                "BEGIN" "FOR" I := 5 "STEP" 1 "UNTIL" N "DO"
                  SB[I] :=
                    (2 * I - 5) / (I - 3) * RRHO * W1 * W2
                    / (1 - R2RHO2) * SB[I-1] +
                    (I - 3) / (I - 4) * (1 - RHO2) *
                    (1 - R2) / (1 - R2RHO2) * SB[I-2];
                  SAMCORBIVNORDEN := SB[N];
                "END";
              "END";
        "END"
            "ELSE"
            "BEGIN"
              Y1 := (RRHO + 2) / 8;
              Y2 := (3 * RRHO + 2) * (3 * RRHO + 2) / 128;
              Y3 := (((15 * RRHO + 18) * RRHO - 4)
                  * RRHO - 8) * 5 / 1024;
```

```
          Y4 := ((((3675 * RRHO + 4200) * RRHO - 2520)
            * RRHO - 3360) * RRHO - 336) / 32768;
          SAMCORBIVNORDEN :=
            (N - 2) / SQRT(N - 1) * (1 - RHO2) * W2
            * (W1 * W2 / (1 - RRHO)) ** N1 *
            SQRT((1 - RRHO) / 2 / PI)
            / ((1 - R2) * W1 * (1 - RHO2) * W2)
            * ((((Y4 / N1 + Y3) / N1 + Y2) / N1 + Y1)
            / N1 + 1);
        "END";
      "END";
"END" SAMCORBIVNORDEN;
"IF" ABS(RHO) >= 1 "THEN"
    STATAL3 ERROR("("BINORCOR")", 2, RHO)
"ELSE" "IF" N > ENTIER(N) "OR" N < 3 "THEN"
    STATAL3 ERROR("("BINORCOR")", 3, N)
"ELSE" "IF" R <= -1 "THEN" BINORCOR := 0
"ELSE" "IF" R >= 1 "THEN" BINORCOR := 1
"ELSE" "IF" RHO = 0 "THEN"
    BINORCOR :=
    STUDENT(R * SQRT((N - 2) / (1 - R * R)), N - 2)
"ELSE"
"IF" N <= 500 "THEN"
"BEGIN" "REAL" W1, W2, W3, PI, R2, RHO2, RHO3,
    RHO4, RRHO, R2RHO2;
    R2 := R * R;
    RHO2 := RHO * RHO;
    RHO3 := RHO2 * RHO;
    RHO4 := RHO2 * RHO2;
    RRHO := R * RHO;
    R2RHO2 := R2 * RHO2;
    W1 := SQRT(1 - R2);
    W2 := SQRT(1 - RHO2);
    W3 := SQRT(1 - RHO2 * R2);
    PI := ARCTAN(1) * 4;
    "IF" N = 3 "THEN"
      BINORCOR :=
      (ARCCOS(-R) - RHO * W1 / W3 * ARCCOS(-RRHO)) / PI
    "ELSE" "IF" N = 4 "THEN"
      BINORCOR :=
      W1 * W2 * SAMCORBIVNORDEN(R, RHO, 3) / RHO -
      (W2 / RHO - ARCCOS(RHO)) / PI
    "ELSE" "IF" N = 5 "THEN"
      BINORCOR :=
      W1 * W2 * SAMCORBIVNORDEN(R, RHO, 4) / 2 / RHO
      - R * (1 - R2) / 2 * SAMCORBIVNORDEN(R, RHO, 3)
      - W1 * (1 + RHO2) / (2 * PI * RHO) * ARCCOS(-RRHO)
      / W3 + ARCCOS(-R) / PI
    "ELSE"
    "IF" N = 6 "THEN"
      BINORCOR :=
      W2 * (1 - 4 * RHO2) / (3 * PI * RHO3)
      +  ARCCOS(RHO) / PI
      - (1 - RHO2) * W1 * W2 / 3 / RHO3 *
```

```
        SAMCORBIVNORDEN(R, RHO, 3)
        + (1 - RHO2) * R / 3 / RHO2 *
        SAMCORBIVNORDEN(R, RHO, 4)
        + W1 * W2 / 3 / RHO * SAMCORBIVNORDEN(R, RHO, 5)
      "ELSE" "IF" N = 7 "THEN"
        BINORCOR :=
        ARCCOS(-R) / PI - (3 + 6 * RHO2 - RHO4) *
        ARCCOS(-RRHO) / W3 * W1 / (8 * PI * RHO)
        - R * (1 - R2) * (4 - 3 * RHO2 + 3 * RHO4) / 8 /
        RHO2 * SAMCORBIVNORDEN(R, RHO, 3)
        - R2 * W1 * W2 * (2 - RHO2) / 8 / RHO *
        SAMCORBIVNORDEN(R, RHO, 4)
        + (1 - RHO2) * R / 4 / RHO2 *
        SAMCORBIVNORDEN(R, RHO, 5)
        + W1 * W2 / 4 / RHO * SAMCORBIVNORDEN(R, RHO, 6)
      "ELSE" "IF" N = 8 "THEN"
        BINORCOR := ARCCOS(RHO)
        / PI - W2 * (3 - 11 * RHO2 + 23 * RHO4)
        / 15 / PI / RHO4 / RHO
        + (W2 / RHO) ** 5 * W1 / 5 *
        SAMCORBIVNORDEN(R, RHO, 3)
        - R * (1-RHO2) * (1-RHO2) / 5 / RHO4 *
        SAMCORBIVNORDEN(R, RHO, 4)
        + (3 * R2 - 1) * (1 - RHO2) * W2 / W1 / 15 / RHO3
        * SAMCORBIVNORDEN(R, RHO, 5)
        + (1 - RHO2) * R / 5 / RHO2 *
        SAMCORBIVNORDEN(R, RHO, 6)
        + W1 * W2 / 5 / RHO * SAMCORBIVNORDEN(R, RHO, 7)
      "ELSE"
      "BEGIN" "REAL" "ARRAY" E[1:3]; "REAL" X;
          E[1] := E [2] := "-6;
          BINORCOR :=
          STUDENT(-RHO * SQRT(N - 1) / W2, N - 1)
          + QADRAT(X, 0, R, SAMCORBIVNORDEN(X, RHO, N), E)
      "END";
  "END"
  "ELSE"
  "BEGIN" "REAL" R2, RHO2, RHO3; "INTEGER" N1, N2, N3;
      N1 := N - 1; N2 := N1 * N1; N3 := N1 * N2;
      R2 := R * R;
      RHO2 := RHO * RHO;
      RHO3 := RHO2 * RHO;
      BINORCOR :=
      "IF" ABS(RHO) <= .7 "THEN"
          PHI((R * SQRT((N - 2.5) / (1 - R2)) -
          RHO * SQRT((N - 1.5) / (1 - RHO2)))
          / SQRT(1 + RHO2 / 2 / (1 - RHO2) + R2 /
          2 / (1 - R2)))
      "ELSE"
          PHI((.5 * LN((1 + R) * (1 - RHO) / (1 - R) /
          (1 + RHO)) - RHO / 2 / N1 -
          (5 * RHO + RHO3) / 8 / N2 - (11 * RHO +
          (2 + RHO2) * 3 * RHO3) / 16 / N3)
          / SQRT(1 / N1 + (4 - RHO2) / 2 / N2 +
```

```
            (22 - (6 - 3 * RHO2) * RHO2) / 6 / N3));
    "END";
"END"  BINORCOR;
          "EOP"
```

TITLE: **Kolsmir**

AUTHOR: R. Kaas

INSTITUTE: Mathematical Centre

RECEIVED: 770101

BRIEF DESCRIPTION

The procedure computes the distribution function of the two-sided Kolmogorov-Smirnov test-statistic under the hypothesis that the two empirical distribution functions F_1 and F_2 stem from the same population. The two-sided Kolmogorov-Smirnov test-statistic equals the maximum over all numbers Z of D = ABS(F_1(Z)−F_2(Z)).

KEYWORDS

Kolmogorov-Smirnov distribution function

CALLING SEQUENCE

Heading

```
"REAL" "PROCEDURE" KOLSMIR (X, M, N, EPS);
"VALUE" X, M, N, EPS;
"REAL" X, M, N, EPS;
"CODE" 41556;
```

Formal parameters

X: <arithmetic expression>, argument of the distribution function;
M: <arithmetic expression>, size of the first sample;
N: <arithmetic expression>, size of the second sample;
EPS: <arithmetic expression>, precision of the value to be computed.

DATA AND RESULTS

The value of the distribution function is assigned to the procedure identifier KOLSMIR.

The following error messages may appear:

Errornumber 2 (if M is not an integer > 0)
Errornumber 3 (if N is not an integer > 0)

PROCEDURES USED

STATAL3 ERROR STATAL 40100

LANGUAGE

Algol 60

METHOD AND PERFORMANCE

If $0 \leq$ EPS $< 10^{-3}$ and (M + N)!/(M!*N!) $\leq$ the maximal representable real number, the distribution function is computed exactly by counting configurations leading to a test-statistic less than or equal to X. In all other cases the distribution function is computed approximately, using the asymptotic relation

$$\text{KOLSMIR(X,M,N,EPS)} = 1 + 2\sum_{k=1}^{\infty}(-1)^k \,\text{EXP}(-\text{YK}^2),$$

where $Y = X^X MN/(2M+2N)$. The summation is terminated as soon as a term is less than EPS.

REFERENCE

[1] J.H.B. Kemperman: *Exacte en asymptotische formules voor de Kolmogorov-Smirnov toets,* Report SD 29, Mathematical Centre, Amsterdam, 1958.

EXAMPLE OF USE

Program:

```
"BEGIN"
    OUTPUT(61, "("3(Z.6D,/)")",
        KOLSMIR( .1,  11,  11,     0),
        KOLSMIR( .1, 100, 150, 0.001),
        KOLSMIR(.15,  15,  17,     0))
"END"
```

Output:

```
 .002903
 .486545
 .030058
```

SOURCE TEXT

```
"CODE" 41556;
"REAL" "PROCEDURE" KOLSMIR(D, XSIZE, YSIZE, EPS);
"VALUE" D, XSIZE, YSIZE, EPS; "REAL" D, XSIZE, YSIZE, EPS;
"BEGIN" "INTEGER" I, KGV, M, N;

    "INTEGER" "PROCEDURE" GGD(M, N);
    "VALUE" M, N; "INTEGER" M, N;
    GGD:= "IF" N = 0 "THEN" M "ELSE" GGD(N, M - M // N * N);

    "PROCEDURE" APPROX;
    "BEGIN" "INTEGER" K; "REAL" SUM, X, TERM, THETA;
        SUM:= .5; THETA:= (1 + (M / KGV) ** 1.2) / (M + N);
        X:= (I / KGV + THETA) ** 2 * 2 / (1 / M + 1 / N);
        "FOR" K:= 1, K + 2 "WHILE" TERM > EPS "DO"
```

```
        "BEGIN" TERM:= EXP(-X * K * K);
            SUM:= SUM - TERM * (1 - EXP(-X * (2 * K + 1)))
        "END";
        KOLSMIR:= 2 * SUM
    "END";

    "PROCEDURE" EXACT;
    "BEGIN" "INTEGER" "ARRAY" LOW[0:N]; "ARRAY" H[0:M];
        "INTEGER" DMN, MN1, X, Y, UPP; "REAL" SUM, BINCOEF;
        BINCOEF:= 1; LOW[0]:= 0; MN1:= M + N + 1;
        DMN:= I * M * N / KGV;
        "FOR" X:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" "INTEGER" T, TN;
            T:= M * X - DMN; TN:= T // N;
            LOW[X]:= "IF" T <= 0 "THEN" 0 "ELSE"
                "IF" TN = T / N "THEN" TN "ELSE" TN + 1;
            BINCOEF:= BINCOEF * (MN1 - X) / X;
            "IF" BINCOEF > "318 "THEN"
            "BEGIN" EPS:= "-4;"GOTO" L "END"
        "END";
        H[0]:= 1;
        "FOR" Y:= 1 "STEP" 1 "UNTIL" M "DO" H[Y]:= 0;
        "FOR" X:= 0 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" Y:= LOW[X]; SUM:= H[Y];
            UPP:= M - LOW[N - X];
            "FOR" Y:= Y + 1 "STEP" 1 "UNTIL" UPP "DO"
            H[Y]:= SUM:= SUM + H[Y]
        "END";
        KOLSMIR:= SUM / BINCOEF
    "END";

    "IF" XSIZE <= 0 "OR" XSIZE > ENTIER(XSIZE) "THEN"
        STATAL3 ERROR("("KOLSMIR")", 2, XSIZE);
    "IF" YSIZE <= 0 "OR" YSIZE > ENTIER(YSIZE) "THEN"
        STATAL3 ERROR("("KOLSMIR")", 3, YSIZE);
    "IF" XSIZE < YSIZE "THEN"
    "BEGIN" N:= XSIZE; M:= YSIZE "END"
    "ELSE" "BEGIN" M:= XSIZE; N:= YSIZE "END";
    "IF" EPS < 0 "OR" EPS > "-2 "THEN" EPS:= "-3;
    "IF" XSIZE < YSIZE "THEN"
    "BEGIN" N:= XSIZE; M:= YSIZE "END"
    "ELSE" "BEGIN" M:= XSIZE; N:= YSIZE "END";
    KGV:= M * N / GGD(M, N);
    I:= ENTIER((1 + "-14) * D * KGV);
    "IF" EPS >= "-3 "THEN" L: APPROX "ELSE" EXACT;
"END" KOLSMIR;
         "EOP"
```

CWI SYLLABI

1 Vacantiecursus 1984: *Hewet - plus wiskunde.* 1984.

2 E.M. de Jager, H.G.J. Pijls (eds.). *Proceedings Seminar 1981-1982. Mathematical structures in field theories.* 1984.

3 W.C.M. Kallenberg, et al. *Testing statistical hypotheses: worked solutions.* 1984.

4 J.G. Verwer (ed.). *Colloquium topics in applied numerical analysis, volume 1.* 1984.

5 J.G. Verwer (ed.). *Colloquium topics in applied numerical analysis, volume 2.* 1984.

6 P.J.M. Bongaarts, J.N. Buur, E.A. de Kerf, R. Martini, H.G.J. Pijls, J.W. de Roever. *Proceedings Seminar 1982-1983. Mathematical structures in field theories.* 1985.

7 Vacantiecurus 1985: *Variatierekening.* 1985.

8 G.M. Tuynman. *Proceedings Seminar 1983-1985. Mathematical structures in field theories, Vol.1 Geometric quantization.* 1985.

9 J. van Leeuwen, J.K. Lenstra (eds.). *Parallel computers and computations.* 1985.

10 Vacantiecursus 1986: *Matrices.* 1986.

11 P.W.H. Lemmens. *Discrete wiskunde: tellen, grafen, spelen en codes.* 1986.

12 J. van de Lune. *An introduction to Tauberian theory: from Tauber to Wiener.* 1986.

13 G.M. Tuynman, M.J. Bergvelt, A.P.E. ten Kroode. *Proceedings Seminar 1983-1985. Mathematical structures in field theories, Vol.2.* 1987.

14 Vacantiecursus 1987: *De personal computer en de wiskunde op school.* 1987.

15 Vacantiecursus 1983: *Complexe getallen.* 1987.

16 P.J.M. Bongaarts, E.A. de Kerf, P.H.M. Kersten. *Proceedings Seminar 1984-1986. Mathematical structures in field theories, Vol.1.* 1988.

17 F. den Hollander, H. Maassen (eds.). *Mark Kac seminar on probability and physics. Syllabus 1985-1987.* 1988.

18 Vacantiecursus 1988. *Differentierekening.* 1988.

19 R. de Bruin, C.G. van der Laan, J.R. Luyten, H.F. Vogt. *Publiceren met LATEX.* 1988.

20 R. van der Horst, R.D. Gill (eds.). *STATAL: statistical procedures in Algol 60, part 1.* 1988.

21 R. van der Horst, R.D. Gill (eds.). *STATAL: statistical procedures in Algol 60, part 2.* 1988.

22 R. van der Horst, R.D. Gill (eds.). *STATAL: statistical procedures in Algol 60, part 3.* 1988.

MC SYLLABI

1.1 F. Göbel, J. van de Lune. *Leergang besliskunde, deel 1: wiskundige basiskennis.* 1965.

1.2 J. Hemelrijk, J. Kriens. *Leergang besliskunde, deel 2: kansberekening.* 1965.

1.3 J. Hemelrijk, J. Kriens. *Leergang besliskunde, deel 3: statistiek.* 1966.

1.4 G. de Leve, W. Molenaar. *Leergang besliskunde, deel 4: Markovketens en wachttijden.* 1966.

1.5 J. Kriens, G. de Leve. *Leergang besliskunde, deel 5: inleiding tot de mathematische besliskunde.* 1966.

1.6a B. Dorhout, J. Kriens. *Leergang besliskunde, deel 6a: wiskundige programmering 1.* 1968.

1.6b B. Dorhout, J. Kriens, J.Th. van Lieshout. *Leergang besliskunde, deel 6b: wiskundige programmering 2.* 1977.

1.7a G. de Leve. *Leergang besliskunde, deel 7a: dynamische programmering 1.* 1968.

1.7b G. de Leve, H.C. Tijms. *Leergang besliskunde, deel 7b: dynamische programmering 2.* 1970.

1.7c G. de Leve, H.C. Tijms. *Leergang besliskunde, deel 7c: dynamische programmering 3.* 1971.

1.8 J. Kriens, F. Göbel, W. Molenaar. *Leergang besliskunde, deel 8: minimaxmethode, netwerkplanning, simulatie.* 1968.

2.1 G.J.R. Förch, P.J. van der Houwen, R.P. van de Riet. *Colloquium stabiliteit van differentieschema's, deel 1.* 1967.

2.2 L. Dekker, T.J. Dekker, P.J. van der Houwen, M.N. Spijker. *Colloquium stabiliteit van differentieschema's, deel 2.* 1968.

3.1 H.A. Lauwerier. *Randwaardeproblemen, deel 1.* 1967.

3.2 H.A. Lauwerier. *Randwaardeproblemen, deel 2.* 1968.

3.3 H.A. Lauwerier. *Randwaardeproblemen, deel 3.* 1968.

4 H.A. Lauwerier. *Representaties van groepen.* 1968.

5 J.H. van Lint, J.J. Seidel, P.C. Baayen. *Colloquium discrete wiskunde.* 1968.

6 K.K. Koksma. *Cursus ALGOL 60.* 1969.

7.1 *Colloquium moderne rekenmachines, deel 1.* 1969.

7.2 *Colloquium moderne rekenmachines, deel 2.* 1969.

8 H. Bavinck, J. Grasman. *Relaxatietrillingen.* 1969.

9.1 T.M.T. Coolen, G.J.R. Förch, E.M. de Jager, H.G.J. Pijls. *Colloquium elliptische differentiaalvergelijkingen, deel 1.* 1970.

9.2 W.P. van den Brink, T.M.T. Coolen, B. Dijkhuis, P.P.N. de Groen, P.J. van der Houwen, E.M. de Jager, N.M. Temme, R.J. de Vogelaere. *Colloquium elliptische differentiaalvergelijkingen, deel 2.* 1970.

10 J. Fabius, W.R. van Zwet. *Grondbegrippen van de waarschijnlijkheidsrekening.* 1970.

11 H. Bart, M.A. Kaashoek. H.G.J. Pijls, W.J. de Schipper, J. de Vries. *Colloquium halfalgebra's en positieve operatoren.* 1971.

12 T.J. Dekker. *Numerieke algebra.* 1971.

13 F.E.J. Kruseman Aretz. *Programmeren voor rekenautomaten; de MC ALGOL 60 vertaler voor de EL X8.* 1971.

14 H. Bavinck, W. Gautschi, G.M. Willems. *Colloquium approximatietheorie.* 1971.

15.1 T.J. Dekker, P.W. Hemker, P.J. van der Houwen. *Colloquium stijve differentiaalvergelijkingen, deel 1.* 1972.

15.2 P.A. Beentjes, K. Dekker, H.C. Hemker, S.P.N. van Kampen, G.M. Willems. *Colloquium stijve differentiaalvergelijkingen, deel 2.* 1973.

15.3 P.A. Beentjes, K. Dekker, P.W. Hemker, M. van Veldhuizen. *Colloquium stijve differentiaalvergelijkingen, deel 3.* 1975.

16.1 L. Geurts. *Cursus programmeren, deel 1: de elementen van het programmeren.* 1973.

16.2 L. Geurts. *Cursus programmeren, deel 2: de programmeertaal ALGOL 60.* 1973.

17.1 P.S. Stobbe. *Lineaire algebra, deel 1.* 1973.

17.2 P.S. Stobbe. *Lineaire algebra, deel 2.* 1973.

17.3 N.M. Temme. *Lineaire algebra, deel 3.* 1976.

18 F. van der Blij, H. Freudenthal, J.J. de Iongh, J.J. Seidel, A. van Wijngaarden. *Een kwart eeuw wiskunde 1946-1971, syllabus van de vakantiecursus 1971.* 1973.

19 A. Hordijk, R. Potharst, J.Th. Runnenburg. *Optimaal stoppen van Markovketens.* 1973.

20 T.M.T. Coolen, P.W. Hemker, P.J. van der Houwen, E. Slagt. *ALGOL 60 procedures voor begin- en randwaardeproblemen.* 1976.

21 J.W. de Bakker (red.). *Colloquium programmacorrectheid.* 1975.

22 R. Helmers, J. Oosterhoff, F.H. Ruymgaart, M.C.A. van Zuylen. *Asymptotische methoden in de toetsingstheorie; toepassingen van naburigheid.* 1976.

23.1 J.W. de Roever (red.). *Colloquium onderwerpen uit de biomathematica, deel 1.* 1976.

23.2 J.W. de Roever (red.). *Colloquium onderwerpen uit de biomathematica, deel 2.* 1977.

24.1 P.J. van der Houwen. *Numerieke integratie van differentiaalvergelijkingen, deel 1: eenstapsmethoden.* 1974.

25 *Colloquium structuur van programmeertalen.* 1976.

26.1 N.M. Temme (ed.). *Nonlinear analysis, volume 1.* 1976.

26.2 N.M. Temme (ed.). *Nonlinear analysis, volume 2.* 1976.

27 M. Bakker, P.W. Hemker, P.J. van der Houwen, S.J. Polak, M. van Veldhuizen. *Colloquium discretiseringsmethoden.* 1976.

28 O. Diekmann, N.M. Temme (eds.). *Nonlinear diffusion problems.* 1976.

29.1 J.C.P. Bus (red.). *Colloquium numerieke programmatuur, deel 1A, deel 1B.* 1976.

29.2 H.J.J. te Riele (red.). *Colloquium numerieke programmatuur, deel 2.* 1977.

30 J. Heering, P. Klint (red.). *Colloquium programmeeromgevingen.* 1983.

31 J.H. van Lint (red.). *Inleiding in de coderingstheorie.* 1976.

32 L. Geurts (red.). *Colloquium bedrijfsystemen.* 1976.

33 P.J. van der Houwen. *Berekening van waterstanden in zeeën en rivieren.* 1977.

34 J. Hemelrijk. *Oriënterende cursus mathematische statistiek.* 1977.

35 P.J.W. ten Hagen (red.). *Colloquium computer graphics.* 1978.

36 J.M. Aarts, J. de Vries. *Colloquium topologische dynamische systemen.* 1977.

37 J.C. van Vliet (red.). *Colloquium capita datastructuren.* 1978.

38.1 T.H. Koornwinder (ed.). *Representations of locally compact groups with applications, part I.* 1979.

38.2 T.H. Koornwinder (ed.). *Representations of locally compact groups with applications, part II.* 1979.

39 O.J. Vrieze, G.L. Wanrooy. *Colloquium stochastische spelen.* 1978.

40 J. van Tiel. *Convexe analyse.* 1979.

41 H.J.J. te Riele (ed.). *Colloquium numerical treatment of integral equations.* 1979.

42 J.C. van Vliet (red.). *Colloquium capita implementatie van programmeertalen.* 1980.

43 A.M. Cohen, H.A. Wilbrink. *Eindige groepen (een inleidende cursus).* 1980.

44 J.G. Verwer (ed.). *Colloquium numerical solution of partial differential equations.* 1980.

45 P. Klint (red.). *Colloquium hogere programmeertalen en computerarchitectuur.* 1980.

46.1 P.M.G. Apers (red.). *Colloquium databankorganisatie, deel 1.* 1981.

46.2 P.G.M. Apers (red.). *Colloquium databankorganisatie, deel 2.* 1981.

47.1 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60: general information and indices.* 1981.

47.2 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 1: elementary procedures; vol. 2: algebraic evaluations.* 1981.

47.3 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 3A: linear algebra, part I.* 1981.

47.4 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 3B: linear algebra, part II.* 1981.

47.5 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 4: analytical evaluations; vol. 5A: analytical problems, part I.* 1981.

47.6 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 5B: analytical problems, part II.* 1981.

47.7 P.W. Hemker (ed.). *NUMAL, numerical procedures in ALGOL 60, vol. 6: special functions and constants; vol. 7: interpolation and approximation.* 1981.

48.1 P.M.B. Vitányi, J. van Leeuwen, P. van Emde Boas (red.). *Colloquium complexiteit en algoritmen, deel 1.* 1982.

48.2 P.M.B. Vitányi, J. van Leeuwen, P. van Emde Boas (red.). *Colloquium complexiteit en algoritmen, deel 2.* 1982.

49 T.H. Koornwinder (ed.). *The structure of real semisimple Lie groups.* 1982.

50 H. Nijmeijer. *Inleiding systeemtheorie.* 1982.

51 P.J. Hoogendoorn (red.). *Cursus cryptografie.* 1983.