MC SYLLABUS 47.3

NUMAL

NUMERICAL PROCEDURES IN ALGOL 60

VOLUME 3A, LINEAR ALGEBRA, PART 1

P.W. HEMKER (ed.)

MATHEMATISCH CENTRUM AMSTERDAM 1981

1980 Mathematics subject classification: 65XX04, 68B99

ISBN 90 6196 217 X

AUTHORS: J. C. P. BUS AND T. J. DEKKER.

CONTRIBUTOR: J.C.P. BUS AND P. A. BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730830.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS SIX PROCEDURES:

A: DEC PERFORMS A TRIANGULAR DECOMPOSITION WITH PARTIAL PIVOTING;

B: GSSELM PERFORMS A TRIANGULAR DECOMPOSITION WITH A COMBINATION OF PARTIAL AND COMPLETE PIVOTING;

C: ONENRMINV DELIVERS THE 1-NORM OF THE INVERSE OF A MATRIX WHOSE TRIANGULARLY DECOMPOSED FORM HAS BEEN DELIVERED BY DEC OR GSSELM;

D: ERBELM CALCULATES A ROUGH UPPERBOUND FOR THE SOLUTION OF A LINEAR SYSTEM WHOSE MATRIX IS TRIANGULARLY DECOMPOSED BY GSSELM;

E: GSSERB PERFORMS A TRIANGULAR DECOMPOSTION OF THE MATRIX OF A LINEAR SYSTEM AND CALCULATES AN UPPERBOUND FOR THE RELATIVE ERROR OF THE SOLUTION OF THAT SYSTEM;

F: GSSNRI PERFORMS A TRIANGULAR DECOMPOSITION AND CALCULATES THE 1-NORM OF THE INVERSE MATRIX;

THE METHOD USED IN DEC AND GSSELM YIELDS A LOWER-TRIANGULAR MATRIX L AND A UNIT UPPER-TRIANGULAR MATRIX U SUCH THAT THE PRODUCT LU EQUALS THE GIVEN MATRIX WITH PERMUTED ROWS (DEC) OR ROWS AND COLUMNS (GSSELM); IN DEC, ONLY PARTIAL PIVOTING IS USED ([3], [4, P.115], [5, P.201]); THE PIVOTING STRATEGY IN GSSELM IS A COMBINATION OF PARTIAL AND COMPLETE PIVOTING ([2], [1]); IN THIS STRATEGY THE PROCESS WILL SWITCH TO COMPLETE PIVOTING IF PARTIAL PIVOTING MIGHT NOT YIELD STABLE RESULTS; SO IN GSSELM THE EFFICIENCY OF PARTIAL PIVOTING IS COMBINED WITH THE STABILITY OF COMPLETE PIVOTING;

SINCE, IN EXCEPTIONAL CASES, PARTIAL PIVOTING MAY YIELD USELESS RESULTS, EVEN FOR WELL-CONDITIONED MATRICES, THE USER IS ADVISED TO USE GSSELM; HOWEVER, IF THE NUMBER OF VARIABLES IS SMALL RELATIVE TO THE NUMBER OF BINARY DIGITS IN THE MANTISSA (48 FOR THE CYBER), THEN DEC MAY ALSO BE USED;

KEYWORDS:

LU DECOMPOSITION,
TRIANGULAR DECOMPOSITION,
GAUSSIAN ELIMINATION.

SUBSECTION: DEC.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DEC(A, N, AUX, P); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX; "INTEGER" "ARRAY" P;"CODE" 34300;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N,1:N];
            ENTRY:THE MATRIX;
            EXIT:THE  CALCULATED  LOWER-TRIANGULAR  MATRIX  AND  UNIT
            UPPERTRIANGULAR  MATRIX  WITH ITS  UNIT  DIAGONAL  OMITTED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[1:3];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                    VALUE  IS AN ESTIMATE  OF THE RELATIVE PRECISION OF
                    THE  MATRIX ELEMENTS;  HOWEVER,  IT  SHOULD NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION; SEE
                    METHOD AND PERFORMANCE;
            EXIT:
            AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE  AUX[3]),   THEN   AUX[1]  EQUALS  1   IF  THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX OF  ORDER  R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                    AUX[3] < N THEN THE PROCESS IS BROKEN OFF BECAUSE
                    THE  SELECTED  PIVOT  IS  TOO  SMALL  RELATIVE  TO
                    THE MAXIMUM OF THE EUCLIDEAN NORMS OF THE ROWS OF
                    THE GIVEN MATRIX;
    P:      <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" P[1:N];
            EXIT:THE PIVOTAL INDICES.
```

PROCEDURES USED:

MATMAT = CP34013,
MATTAM = CP34015,
ICHROW = CP34032.

REQUIRED CENTRAL MEMORY:

A REAL ARRAY OF ORDER N IS DECLARED.

RUNNING TIME: PROPORTIONAL TO N ** 3.

METHOD AND PERFORMANCE:

THE METHOD USED IN DEC IS TRIANGULAR DECOMPOSITION WITH STABILIZING ROW INTERCHANGES, ALSO CALLED "PARTIAL PIVOTING"; SEE ALSO [3,P.19] AND [5,P.201].

EXAMPLE OF USE: SEE DECSOL (SECTION 3.1.1.1.1.1.3).

SUBSECTION: GSSELM .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX; "INTEGER" "ARRAY" RI, CI;"CODE" 34231;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N,1:N];
            ENTRY:THE N-TH ORDER MATRIX;
            EXIT:THE  CALCULATED  LOWER-TRIANGULAR  MATRIX  AND  UNIT
            UPPERTRIANGULAR  MATRIX  WITH ITS UNIT  DIAGONAL  OMITTED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[1:7];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                    VALUE  IS  AN ESTIMATE OF THE RELATIVE PRECISION OF
                    THE MATRIX ELEMENTS;  HOWEVER,  IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION; SEE
                    METHOD AND PERFORMANCE;
            AUX[4]: A  VALUE  WHICH  IS USED  FOR CONTROLLING PIVOTING;
                    USUALLY, AUX[4] = 8 WILL GIVE GOOD RESULTS; SEE
                    METHOD AND PERFORMANCE;
            EXIT:
            AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE  AUX[3]),   THEN   AUX[1]  EQUALS   1  IF  THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX  OF ORDER  R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                    AUX[3] < N THEN THE PROCESS HAS BEEN BROKEN OFF,
                    BECAUSE THE SELECTED PIVOT IS TOO SMALL RELATIVE TO
                    THE MAXIMUM OF THE MODULI OF ELEMENTS OF THE GIVEN
                    MATRIX;
            AUX[5]: THE  MODULUS  OF  AN  ELEMENT  WHICH  IS OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX WHICH HAD BEEN GIVEN
                    IN ARRAY A;
            AUX[7]: AN UPPER BOUND FOR THE GROWTH (I. E. THE MODULUS OF
                    AN ELEMENT  WHICH IS OF MAXIMUM ABSOLUTE VALUE  FOR
                    THE MATRICES OCCURRING DURING ELIMINATION);
    RI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" RI[1:N];
            EXIT: THE PIVOTAL ROW-INDICES;
    CI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" CI[1:N];
            EXIT: THE PIVOTAL COLUMN-INDICES;
```

PROCEDURES USED:

```
ROWCST      = CP31132,
ELMROW      = CP34024,
MAXELMROW   = CP34025,
ICHCOL      = CP34031,
ICHROW      = CP34032,
ABSMAXMAT   = CP31069.
```

REQUIRED CENTRAL MEMORY: NO EXTRA ARRAYS ARE DECLARED.

RUNNING TIME: PROPORTIONAL TO N ** 3.

METHOD AND PERFORMANCE:

THE PROCESS OF GAUSSIAN ELIMINATION IS PERFORMED IN AT MOST N STEPS, WHERE N DENOTES THE ORDER OF THE MATRIX; PARTIAL PIVOTING WILL BE USED AS LONG AS THE CALCULATED UPPER BOUND FOR THE GROWTH ([2], [1]), IS LESS THAN A CRITICAL VALUE THAT EQUALS AUX[4] * N TIMES THE MODULUS OF AN ELEMENT WHICH IS OF MAXIMUM ABSOLUTE VALUE FOR THE GIVEN MATRIX;
IN THE PARTIAL PIVOTING STRATEGY, THAT ELEMENT IS CHOSEN AS PIVOT IN THE K-TH STEP, WHOSE ABSOLUTE VALUE IS MAXIMAL FOR THE K-TH COLUMN OF THE LOWER-TRIANGULAR MATRIX L; HOWEVER, IF THE UPPER BOUND FOR THE GROWTH EXCEEDS THIS CRITICAL VALUE IN THE K-TH STEP, THEN A PIVOT IS SELECTED IN THE J-TH STEP (J = K,..., N), IN SUCH A WAY, THAT ITS ABSOLUTE VALUE IS MAXIMAL FOR THE REMAINING SUBMATRIX OF ORDER N - K + 1 (COMPLETE PIVOTING);
SINCE IN PRACTICE, IF WE CHOOSE AUX[4] PROPERLY, THE UPPER BOUND FOR THE GROWTH RARELY EXCEEDS THIS CRITICAL VALUE ([2], [4]), WE WILL USUALLY TAKE ADVANTAGE OF THE GREATER SPEED OF PARTIAL PIVOTING (ORDER N - K + 1 IN THE K-TH STEP), WHILE IN A FEW DOUBTFUL CASES NUMERICAL DIFFICULTIES WILL BE RECOGNIZED AND THE PROCESS WILL SWITCH TO COMPLETE PIVOTING (ORDER (N - K + 1) ** 2 IN THE K-TH STEP);
USING GSSELM,THE UPPER BOUND FOR THE RELATIVE ERROR IN THE SOLUTION OF A LINEAR SYSTEM ([4], [5]), WILL BE AT MOST AUX[4] * N TIMES THE UPPER BOUND USING GAUSSIAN ELIMINATION WITH COMPLETE PIVOTING ONLY; USUALLY, HOWEVER, THIS WILL BE A CRUDE OVERESTIMATE;
THE CHOICE AUX[4] < 1 / N WILL RESULT IN COMPLETE PIVOTING ONLY, WHILE PARTIAL PIVOTING WILL BE USED IN EVERY STEP IF WE CHOOSE AUX[4] > (2 ** (N - 1)) / N; USUALLY, AUX[4] = 8 WILL GIVE GOOD RESULTS ([2], [1]);
THE PROCESS WILL ALSO SWITCH TO COMPLETE PIVOTING IF THE MODULUS OF THE PIVOT OBTAINED WITH PARTIAL PIVOTING IS LESS THAN A CERTAIN TOLERANCE, WHICH EQUALS THE GIVEN RELATIVE TOLERANCE AUX[2] TIMES THE MODULUS OF AN ELEMENT WHICH IS OF MAXIMUM ABSOLUTE VALUE FOR THE GIVEN MATRIX; IF ALL ELEMENTS IN THE REMAINING SUBMATRIX ARE SMALLER IN ABSOLUTE VALUE THAN THIS TOLERANCE THEN THE PROCESS IS BROKEN OFF AND THE PREVIOUS STEPNUMBER IS DELIVERED IN AUX[3];
IN CONTRAST WITH THE METHOD USED IN DEC (THIS SECTION), NO EQUILIBRATING IS DONE IN THIS PIVOTING STRATEGY; THE USER HIMSELF HAS TO TAKE CARE FOR A REASONABLE SCALING OF THE MATRIX ELEMENTS.

EXAMPLE OF USE: SEE GSSSOL (SECTION 3.1.1.1.1.1.3).

SUBSECTION: ONENRMINV.

CALLING SEQUENCE:

THE HEADING OF THIS PROCEDURE IS
"REAL" "PROCEDURE" ONENRMINV(A, N); "VALUE" N;
"INTEGER" N; "ARRAY" A; "CODE" 34240;

ONENRMINV:= THE 1-NORM OF THE CALCULATED INVERSE OF THE MATRIX, WHOSE TRIANGULARLY DECOMPOSED FORM IS GIVEN IN ARRAY A;

THE MEANING OF THE FORMAL PARAMETERS IS:

A: <ARRAY IDENTIFIER>;
"ARRAY" A[1:N,1:N];
ENTRY:
THE TRIANGULARLY DECOMPOSED FORM OF A MATRIX, AS DELIVERED BY GSSELM OR DEC (THIS SECTION);

N: <ARITHMETIC EXPRESSION>;
THE ORDER OF THE MATRIX, WHOSE TRIANGULARLY DECOMPOSED FORM HAS BEEN GIVEN IN ARRAY A.

PROCEDURES USED:

MATVEC = CP34011.

REQUIRED CENTRAL MEMORY:

ONE REAL ARRAY OF ORDER N IS DECLARED.

RUNNING TIME: PROPORTIONAL TO N ** 3.

METHOD AND PERFORMANCE:

THE INVERSE OF THE MATRIX WHOSE TRIANGULARLY DECOMPOSED FORM, AS DELIVERED BY GSSELM OR DEC, HAS BEEN GIVEN IN ARRAY A, IS CALCULATED WITH FORWARD AND BACK SUBSTITUTION ([3],[4],[5]); ONLY THE 1-NORM OF THIS INVERSE IS DELIVERED BY ONENRMINV; THE ELEMENTS OF ARRAY A REMAIN UNALTERED.

EXAMPLE OF USE: SEE GSSSOLERB (SECTION 3.1.1.1.1.1.3).

SUBSECTION: ERBELM.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" ERBELM(N, AUX, NRMINV); "VALUE" N, NRMINV;
    "INTEGER" N; "REAL" NRMINV; "ARRAY" AUX; "CODE" 34241;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE LINEAR SYSTEM IN CONSIDERATION;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[0:11];
            ENTRY:
            AUX[0]: THE MACHINE PRECISION;
            AUX[5]: THE MODULUS  OF  AN ELEMENT,  WHICH  IS  OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX OF THE LINEAR SYSTEM;
                    THIS VALUE IS DELIVERED  BY GSSELM  IN AUX[5] (THIS
                    SECTION);
            AUX[6]: AN  UPPER BOUND  FOR  THE  RELATIVE  ERROR  IN  THE
                    ELEMENTS OF THE MATRIX OF THE LINEAR SYSTEM;
            AUX[7]: AN  UPPER BOUND  FOR  THE  GROWTH  DURING  GAUSSIAN
                    ELIMINATION;  THIS VALUE IS DELIVERED IN  AUX[7] BY
                    GSSELM (THIS SECTION);
            EXIT:
            AUX[9]: THE VALUE OF NRMINV;
            AUX[11]: A ROUGH UPPER BOUND  FOR THE RELATIVE ERROR IN THE
                    SOLUTION   OF  A  LINEAR  SYSTEM   WHEN   GAUSSIAN
                    ELIMINATION  IS USED  FOR  THE CALCULATION  OF THIS
                    SOLUTION; IF NO USE CAN BE MADE OF THE FORMULA  FOR
                    THE  ERROR BOUND  (SEE:  METHOD  AND  PERFORMANCE),
                    BECAUSE OF A VERY BAD CONDITION OF THE MATRIX, THEN
                    AUX[11]:= -1;
    NRMINV: <ARITHMETIC EXPRESSION>;
            THE 1-NORM  OF  THE INVERSE  OF THE MATRIX  OF  THE  LINEAR
            SYSTEM MUST BE GIVEN IN NRMINV; THIS VALUE MAY BE OBTAINED
            BY ONENRMINV (THIS SECTION).
```

PROCEDURES USED: NONE.

REQUIRED CENTRAL MEMORY: NO EXTRA ARAYS ARE DECLARED.

METHOD AND PERFORMANCE:

WHEN CALLED AFTER GSSELM, ERBELM WILL CALCULATE A ROUGH UPPER BOUND FOR THE RELATIVE ERROR IN THE SOLUTION OF THE LINEAR SYSTEM, WHOSE MATRIX HAS BEEN DECOMPOSED INTO TRIANGULAR FORM BY GSSELM (THIS SECTION), BY ([3], [4], [5]):

```
        NORM(DX) / NORM(X) <= P / (1 - P),
WHERE : P = Q * NORM(C) / (1 - Q * NORM(C)),
        Q = G * (.75 * N ** 3 + 4.5 * N ** 2) * EPS + EPSA,
        C IS THE CALCULATED INVERSE OF THE MATRIX,
        G   THE   UPPER   BOUND   FOR   THE   GROWTH   DURING   GAUSSIAN
        ELIMINATION, AS DELIVERED BY GSSELM (THIS SECTION),
        N THE ORDER OF THE MATRIX,
        EPSA  AN UPPER BOUND  FOR  THE RELATIVE ERROR IN THE MATRIX
        ELEMENTS,
        EPS THE MACHINE PRECISION AND
        NORM(.) DENOTES THE 1-NORM.
```

THIS PROCEDURE IS USED IN E.G. GSSERB (THIS SECTION) AND GSSINVERB (SECTION 3.1.1.1.1.1.4)

EXAMPLE OF USE: SEE GSSSOLERB (SECTION 3.1.1.1.1.1.3).

SUBSECTION: GSSERB .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" GSSERB(A, N, AUX, RI, CI); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX; "INTEGER" "ARRAY" RI, CI; "CODE" 34242;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N,1:N];
            ENTRY:THE MATRIX TO BE DECOMPOSED;
            EXIT: THE CALCULATED LOWER-TRIANGULAR MATRIX AND UNIT UPPER
            TRIANGULAR MATRIX,WITH ITS UNIT DIAGONAL OMITTED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[0:11];
            ENTRY:(SEE ALSO GSSELM IN THIS SECTION);
            AUX[0]: THE MACHINE PRECISION;
            AUX[2]: A RELATIVE TOLERANCE;
            AUX[4]: A VALUE WHICH IS USED FOR CONTROLLING PIVOTING;
            AUX[6]: AN UPPER BOUND  FOR  THE RELATIVE PRECISION  OF THE
                    MATRIX ELEMENTS;
            EXIT:
            AUX[1]: IF  R IS THE NUMBER OF ELIMINATION STEPS PERFORMED,
                    AUX[1] EQUALS 1 IF THE DETERMINANT OF THE PRINCIPAL
                    SUBMATRIX OF ORDER R IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
            AUX[5]: THE  MODULUS  OF  AN ELEMENT,  WHICH  IS OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX WHICH HAS BEEN GIVEN
                    IN ARRAY A;
            AUX[7]: AN UPPER BOUND FOR THE GROWTH;
            AUX[9]: IF AUX[3] = N, THEN AUX[9] WILL EQUAL THE 1-NORM OF
                    THE INVERSE MATRIX, ELSE AUX[9] WILL BE UNDEFINED;
            AUX[11]: IF AUX[3] = N, THEN THE VALUE OF AUX[11] WILL BE A
                    ROUGH  UPPER BOUND  FOR  THE RELATIVE ERROR  IN THE
                    SOLUTION OF LINEAR SYSTEMS  WITH A MATRIX  AS GIVEN
                    IN ARRAY A, ELSE AUX[11] WILL BE UNDEFINED;  IF  NO
                    USE CAN BE MADE OF THE FORMULA FOR  THE ERROR BOUND
                    AS GIVEN ABOVE (SUBSECTION  ERBELM),  BECAUSE OF  A
                    VERY BAD CONDITION OF THE MATRIX, THEN AUX[11]:=-1;
    RI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" RI[1:N];
            EXIT: THE PIVOTAL ROW-INDICES.
    CI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" CI[1:N];
            EXIT: THE PIVOTAL COLUMN-INDICES.
```

PROCEDURES USED:

```
    GSSELM      = CP34231,
    ONENRMINV   = CP34240,
    ERBELM      = CP34241.
```

REQUIRED CENTRAL MEMORY: NO EXTRA ARRAYS ARE DECLARED.

RUNNING TIME: PROPORTIONAL TO N ** 3.

METHOD AND PERFORMANCE:

GSSERB USES GSSELM (THIS SECTION) TO PERFORM THE TRIANGULAR DECOMPOSITION OF THE MATRIX GIVEN IN ARRAY A AND ERBELM AND ONENRMINV (THIS SECTION) TO CALCULATE AN UPPER BOUND FOR THE RELATIVE ERROR IN THE SOLUTION OF LINEAR SYSTEMS WITH A MATRIX AS GIVEN IN ARRAY A; IF AUX[3] < N, THEN THE EFFECT OF GSSERB IS MERELY THAT OF GSSELM.

EXAMPLE OF USE: SEE GSSSOLERB (SECTION 3.1.1.1.1.1.3).

SUBSECTION: GSSNRI .

CALLING SEQUENCE:

```
THE HEADING OF THIS PROCEDURE IS:
"PROCEDURE" GSSNRI(A, N, AUX, RI, CI); "VALUE" N;
"INTEGER" N; "ARRAY" A, AUX; "INTEGER" "ARRAY" RI, CI; "CODE" 34252;

THE MEANING OF THE FORMAL PARAMETERS IS:
A:      <ARRAY IDENTIFIER>;
        "ARRAY" A[1:N,1:N];
        ENTRY:THE MATRIX TO BE DECOMPOSED;
        EXIT: THE CALCULATED LOWER-TRIANGULAR MATRIX AND UNIT UPPER
        TRIANGULAR MATRIX,WITH ITS DIAGONAL OMITTED;
N:      <ARITHMETIC EXPRESSION>;
        THE ORDER OF THE MATRIX;
AUX:    <ARRAY IDENTIFIER>;
        "ARRAY" AUX[1:9];
        ENTRY:(SEE ALSO GSSELM IN THIS SECTION);
        AUX[2]: A RELATIVE TOLERANCE;
        AUX[4]: A VALUE USED FOR CONTROLLING PIVOTING;
        EXIT:
        AUX[1]: IF R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED,
                THEN AUX[1]  EQUALS  1  IF THE DETERMINANT  OF  THE
                PRINCIPAL SUBMATRIX OF ORDER  R  IS POSITIVE,  ELSE
                AUX[1] = -1;
        AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
        AUX[5]: THE  MODULUS  OF  AN  ELEMENT,  WHICH IS OF MAXIMUM
                ABSOLUTE VALUE FOR THE MATRIX WHICH HAD BEEN GIVEN
                IN ARRAY A;
        AUX[7]: AN UPPER BOUND FOR THE GROWTH;
        AUX[9]: IF AUX[3] = N, THEN AUX[9] WIL EQUAL  THE 1-NORM OF
                THE INVERSE MATRIX, ELSE AUX[9] WILL BE UNDEFINED;
```

```
    RI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" RI[1:N];
            EXIT: THE PIVOTAL ROW INDICES.
    CI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" CI[1:N];
            EXIT:THE PIVOTAL COLUMN INDICES.
```

PROCEDURES USED:

```
    GSSELM       = CP34231,
    ONENRMINV    = CP34240.
```

REQUIRED CENTRAL MEMORY: NO EXTRA ARRAYS ARE DECLARED.

RUNNING TIME: PROPORTIONAL TO N ** 3.

METHOD AND PERFORMANCE:

GSSNRI USES GSSELM (THIS SECTION) TO PERFORM THE TRIANGULAR DECOMPOSITION OF THE MATRIX GIVEN IN ARRAY A AND ONENRMINV (THIS SECTION) TO CALCULATE THE 1-NORM OF THE INVERSE MATRIX; IF AUX[3] < N, THEN THE EFFECT OF GSSNRI IS MERELY THAT OF GSSELM (THIS SECTION).

EXAMPLE OF USE: SEE GSSITISOLERB (SECTION 3.1.1.1.1.1.5).

REFERENCES:

[1] BUS, J. C. P.
LINEAR SYSTEMS WITH CALCULATION OF ERROR BOUNDS AND ITERATIVE REFINEMENT (DUTCH).
MATHEMATICAL CENTRE, AMSTERDAM, LR 3. 4. 19 (1972).

[2] BUSINGER, P. A.
MONITORING THE NUMERICAL STABILITY OF GAUSSIAN ELIMINATION.
NUMER. MATH. 16, 360-361 (1971).

[3] DEKKER, T. J.
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1.
MATHEMATICAL CENTRE, AMSTERDAM, TRACT 22 (1968).

[4] WILKINSON, J. H.
ROUNDING ERRORS IN ALGEBRAIC PROCESSES.
LONDON (1963).

[5] WILKINSON, J. H.
THE ALGEBRAIC EIGENVALUE PROBLEM.
OXFORD (1965).

SOURCE TEXT(S):

```
"CODE" 34300;
    "PROCEDURE" DEC(A, N, AUX, P); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX; "INTEGER" "ARRAY" P;
    "BEGIN" "INTEGER" I, K, K1, PK, D;
        "REAL" R, S, EPS;
        "ARRAY" V[1:N];
        "REAL" "PROCEDURE" MATMAT(L, U, I, J, A, B); "CODE" 34013;
        "REAL" "PROCEDURE" MATTAM(L, U, I, J, A, B); "CODE" 34015;
        "PROCEDURE" ICHROW(L, U, I, J, A); "CODE" 34032;
        R:= -1;
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" S:= SQRT(MATTAM(1,N,I,I,A,A));
            "IF" S > R "THEN" R:= S; V[I]:= 1/S
        "END";
        EPS:= AUX[2] * R; D:= 1;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" R:= - 1; K1:= K - 1;
            "FOR" I:= K "STEP" 1 "UNTIL" N "DO"
            "BEGIN" A[I,K]:= A[I,K] - MATMAT(1, K1, I, K, A, A);
                S:= ABS(A[I,K]) * V[I]; "IF" S > R "THEN"
                "BEGIN" R:= S; PK:= I "END"
            "END" LOWER;
            P[K]:= PK; V[PK]:= V[K]; S:= A[PK,K];
            "IF" ABS(S) < EPS "THEN" "GOTO" END;
            "IF" S < 0 "THEN" D:= -D; "IF" PK ^= K "THEN"
            "BEGIN" D:= - D; ICHROW(1, N, K, PK, A) "END";
            "FOR" I:= K + 1 "STEP" 1 "UNTIL" N "DO"
            A[K,I]:= (A[K,I] - MATMAT(1, K1, K, I, A, A)) / S
        "END" LU;
        K:= N + 1;
     END: AUX[1]:= D; AUX[3]:= K - 1
    "END" DEC;
        "EOP"

"CODE" 34231;
    "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "INTEGER" "ARRAY" RI, CI;
    "BEGIN" "INTEGER" I, J, P, Q, R, R1, JPIV, RANK, SIGNDET;
        "REAL" CRIT, PIVOT, RGROW, MAX, AID, MAX1, EPS;
        "BOOLEAN" PARTIAL;
        "PROCEDURE" ELMROW(L, U, I, J, A, B, X); "CODE" 34024;
        "INTEGER" "PROCEDURE" MAXELMROW(L, U, I, J, A, B, X);
        "CODE" 34025;
        "PROCEDURE" ICHCOL(L, U, I, J, A); "CODE" 34031;
        "PROCEDURE" ICHROW(L, U, I, J, A); "CODE" 34032;
                                                              "COMMENT"
```

```
    "PROCEDURE" ROWCST(L, U, I, A, X); "CODE" 31132;
    "REAL" "PROCEDURE" ABSMAXMAT(LR, UR, LC, UC, I, J, A);
    "CODE" 31069;
    AUX[5]:= RGROW:= ABSMAXMAT(1, N, 1, N, I, J, A);
    CRIT:= N * RGROW * AUX[4]; EPS:= RGROW * AUX[2]; MAX:= 0;
    RANK:= N; SIGNDET:= 1; PARTIAL:= RGROW ^= 0;
    "FOR" Q:= 1 "STEP" 1 "UNTIL" N "DO" "IF" Q ^= J "THEN"
    "BEGIN" AID:= ABS(A[I,Q]);
        "IF" AID > MAX "THEN" MAX:= AID
    "END";
    RGROW:= RGROW + MAX;
    "FOR" R:= 1 "STEP" 1 "UNTIL" N "DO"
    "BEGIN" R1:= R + 1; "IF" I ^= R "THEN"
        "BEGIN" SIGNDET:= - SIGNDET; ICHROW(1, N, R, I, A) "END";
        "IF" J ^= R "THEN"
        "BEGIN" SIGNDET:= - SIGNDET; ICHCOL(1, N, R, J, A) "END";
        RI[R]:= I; CI[R]:= J; PIVOT:= A[R,R];
        "IF" PIVOT < 0 "THEN" SIGNDET:= - SIGNDET;
        "IF" PARTIAL "THEN"
        "BEGIN" MAX:= MAX1:= 0; J:= R1;
            ROWCST(R1, N, R, A, 1 / PIVOT);
            "FOR" P:= R1 "STEP" 1 "UNTIL" N "DO"
            "BEGIN" ELMROW(R1, N, P, R, A, A, - A[P,R]);
                AID:= ABS(A[P,R1]); "IF" MAX < AID "THEN"
                "BEGIN" MAX:= AID; I:= P "END";
             "END";
            "FOR" Q:= R1 + 1 "STEP" 1 "UNTIL" N "DO"
            "BEGIN" AID:= ABS(A[I,Q]);
                "IF" MAX1 < AID "THEN" MAX1:= AID
            "END";
            AID:= RGROW; RGROW:= RGROW + MAX1;
            "IF" RGROW > CRIT "OR" MAX < EPS "THEN"
            "BEGIN" PARTIAL:= "FALSE"; RGROW:= AID;
                MAX:= ABSMAXMAT(R1, N, R1, N, I, J, A)
            "END"
        "END" PARTIAL PIVOTINGSTEP
        "ELSE"
        "BEGIN" "IF" MAX <= EPS "THEN"
            "BEGIN" RANK:= R - 1;
                "IF" PIVOT < 0 "THEN" SIGNDET:= - SIGNDET;"GOTO"OUT
            "END";
            MAX:= - 1;
            ROWCST(R1, N, R, A, 1 / PIVOT);
            "FOR" P:= R1 "STEP" 1 "UNTIL" N "DO"
            "BEGIN" JPIV:= MAXELMROW(R1, N, P, R, A, A, - A[P,R]);
                AID:= ABS(A[P,JPIV]); "IF" MAX < AID "THEN"
                "BEGIN" MAX:= AID; I:= P; J:= JPIV "END"
            "END";
            "IF" RGROW < MAX "THEN" RGROW:= MAX
        "END" COMPLETE PIVOTINGSTEP
    "END" ELIMINATIONSTEP;
OUT: AUX[1]:= SIGNDET; AUX[3]:= RANK; AUX[7]:= RGROW
"END" GSSELM;
    "EOP"
```

```
"CODE" 34240;
    "REAL" "PROCEDURE" ONENRMINV(A, N); "VALUE" N; "INTEGER" N;
    "ARRAY" A;
    "BEGIN" "INTEGER" I, J;
        "REAL" NORM, MAX, AID;
        "ARRAY" Y[1:N];
        "REAL" "PROCEDURE" MATVEC(L, U, I, A, B); "CODE" 34011;
        NORM:= 0;
        "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO" Y[I]:= "IF" I < J
            "THEN" 0 "ELSE" "IF" I = J "THEN" 1 / A[I,I] "ELSE"
            - MATVEC(J, I - 1, I, A, Y) / A[I,I];
            MAX:= 0;
            "FOR" I:= N "STEP" - 1 "UNTIL" 1 "DO"
            "BEGIN" AID:= Y[I]:= Y[I] - MATVEC(I + 1, N, I, A, Y);
                MAX:= MAX + ABS(AID)
            "END";
            "IF" NORM < MAX "THEN" NORM:= MAX
        "END";
        ONENRMINV:= NORM
    "END" ONENRMINV;
        "EOP"
"CODE" 34241;
    "PROCEDURE" ERBELM(N, AUX, NRMINV); "VALUE" N, NRMINV;
    "INTEGER" N; "REAL" NRMINV;
    "ARRAY" AUX;
    "BEGIN" "REAL" AID, EPS;
        EPS:= AUX[0]; AID:= (1.06 * EPS * (.75 * N + 4.5) * N ** 2
        * AUX[7] + AUX[5] * AUX[6]) * NRMINV;
        AUX[11]:= "IF" 2 * AID >= (1 - EPS) "THEN" - 1 "ELSE"
        AID / (1 - 2 * AID); AUX[9]:= NRMINV
    "END" ERBELM;
        "EOP"
"CODE" 34242;
    "PROCEDURE" GSSERB(A, N, AUX, RI, CI); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "INTEGER" "ARRAY" RI, CI;
    "BEGIN" "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "CODE" 34231;
        "REAL" "PROCEDURE" ONENRMINV(A, N); "CODE" 34240;
        "PROCEDURE" ERBELM(N, AUX, NRMINV); "CODE" 34241;
        GSSELM(A, N, AUX, RI, CI);
        "IF" AUX[3] = N "THEN" ERBELM(N, AUX, ONENRMINV(A, N))
    "END" GSSERB;
        "EOP"
"CODE" 34252;
    "PROCEDURE" GSSNRI(A, N, AUX, RI, CI); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "INTEGER" "ARRAY" RI, CI;
    "BEGIN" "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "CODE" 34231;
        "REAL" "PROCEDURE" ONENRMINV(A, N); "CODE" 34240;
        GSSELM(A, N, AUX, RI, CI);
        "IF" AUX[3] = N "THEN" AUX[9]:= ONENRMINV(A, N)
    "END" GSSNRI;
        "EOP"
```

AUTHOR: J. C. P. BUS.

CONTRIBUTOR: J.C.P. BUS AND P. A. BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730831.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS A PROCEDURE FOR CALCULATING THE DETERMINANT OF A TRIANGULAR DECOMPOSED MATRIX;

KEYWORDS:

DETERMINANT.

CALLING SEQUENCE:

THE HEADING OF THIS PROCEDURE IS:

```
"REAL" "PROCEDURE" DETERM(A, N, SIGN); "VALUE" N, SIGN;
"INTEGER" N, SIGN; "ARRAY" A;
```

DETERM: DELIVERS THE CALCULATED VALUE OF THE DETERMINANT OF THE MATRIX;

THE MEANING OF THE FORMAL PARAMETERS IS:

```
A:      <ARRAY IDENTIFIER>;
        "ARRAY" A[1:N, 1:N];
        ENTRY:  THE DIAGONAL ELEMENTS OF THE
                LOWER-TRIANGULAR MATRIX L, OBTAINED BY TRIANGULAR
                DECOMPOSITION OF THE MATRIX, HAS TO BE GIVEN IN
                A[I,I], I= 1, ..., N;
N:      <ARITHMETIC EXPRESSION>;
        THE ORDER OF THE MATRIX, WHOSE DETERMINANT HAS TO BE
        CALCULATED;
SIGN:   <ARITHMETIC EXPRESSION>;
        ENTRY:  IF THE DETERMINANT OF THE MATRIX IS POSITIVE THEN
                THE VALUE OF SIGN SHOULD BE +1, ELSE -1; THIS VALUE
                IS DELIVERED BY GSSELM OR DEC IN AUX[1], (SECTION
                3.1.1.1.1.1.1).
```

PROCEDURES USED: NONE.

RUNNING TIME: PROPORTIONAL TO N.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

A LOWER-TRIANGULAR MATRIX L HAS TO BE GIVEN, SUCH THAT FOR SOME UNIT UPPER-TRIANGULAR MATRIX U THE PRODUCT LU EQUALS THE MATRIX (WITH PERMUTED ROWS AND COLUMNS); THE SIGN OF THE DETERMINANT ALSO HAS TO BE GIVEN; THESE DATA ARE DELIVERED IN THE MATRIX AND AUX[1] BY THE PROCEDURES GSSELM OR DEC (SECTION 3.1.1.1.1.1.1) AND THE PROCEDURES GSSERB, GSSNRI (SECTION 3.1.1.1.1.1.1), DECSOL, GSSSOL, GSSSOLERB (SECTION 3.1.1.1.1.1.3), GSSITISOL AND GSSITISOLERB (SECTION 3.1.1.1.1.1.5), WHICH MAKE USE OF GSSELM OR DEC.
THE CALCULATION OF THE DETERMINANT IS DONE STRAIGTH ON BY CALCULATING THE PRODUCT OF THE DIAGONAL ELEMENTS OF THE LOWER-TRIANGULAR MATRIX GIVEN IN ARRAY A; THE USER IS WARNED, THAT OVERFLOW MAY OCCUR IF THE ORDER OF THE MATRIX IS LARGE.

EXAMPLE OF USE:

THE DETERMINANT OF THE FOURTH ORDER SEGMENT OF THE HILBERT MATRIX MAY BE OBTAINED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "INTEGER" I, J; "REAL" D; "INTEGER" "ARRAY" RI, CI[1:4];
    "ARRAY" A[1:4, 1:4], AUX[1:7];
    "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "CODE" 34231;
    "REAL" "PROCEDURE" DETERM(A, N, SIGN); "CODE" 34303;
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO" A[I, J]:= 1 / (I + J - 1);
    AUX[2]:= "-14; AUX[4]:= 8;
    GSSELM(A, 4, AUX, RI, CI);
    D:= "IF" AUX[3] = 4 "THEN" DETERM(A, 4, AUX[1]) "ELSE" 0;
    OUTPUT(71, "("("DETERMINANT =")"B+.15D"+3D")", D)
"END"
```

RESULT:

```
DETERMINANT = +.165343915345370"-006
```

SOURCE TEXT(S):

```
"CODE" 34303;
    "REAL" "PROCEDURE" DETERM(A, N, SIGN); "VALUE" N, SIGN;
    "INTEGER" N, SIGN; "ARRAY" A;
    "BEGIN" "INTEGER" I; "REAL" DET;
        DET:= 1;
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO" DET:= A[I, I] * DET;
        DETERM:= SIGN * ABS(DET)
    "END" DETERM;
        "EOP"
```

AUTHORS: J. C. P. BUS AND T. J. DEKKER.

CONTRIBUTOR: J.C.P. BUS AND P. A. BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730915.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS FIVE PROCEDURES:
SOL SOLVES THE LINEAR SYSTEM WHOSE MATRIX HAS BEEN TRIANGULARLY DECOMPOSED BY DEC;
DECSOL SOLVES A LINEAR SYSTEM WHOSE ORDER IS SMALL RELATIVE TO THE NUMBER OF BINARY DIGITS IN THE NUMBER REPRSENTATION;
SOLELM SOLVES A LINEAR SYSTEM WHOSE MATRIX HAS BEEN TRIANGULARLY DECOMPOSED BY GSSELM OR GSSERB(SECTION 3.1.1.1.1.1.1.).
GSSSOL SOLVES A LINEAR SYSTEM;
GSSSOLERB SOLVES A LINEAR SYSTEM AND CALCULATES A ROUGH UPPERBOUND FOR THE RELATIVE ERROR IN THE CALCULATED SOLUTION;

THE DIFFERENCE BETWEEN DECSOL ON THE ONE SIDE AND GSSSOL AND GSSSOLERB ON THE OTHER SIDE LIES IN THE METHOD USED FOR TRIANGULAR DECOMPOSITION, PARTICULARLY IN THE PIVOTING STRATEGY; DECSOL USES DEC, GSSSOL AND GSSSOLERB USE GSSELM TO PERFORM THE TRIANGULAR DECOMPOSITION (SECTION 3.1.1.1.1.1.1); SINCE, IN EXCEPTIONAL CASES, DEC MAY YIELD USELESS RESULTS, ONE IS ADVISED TO USE GSSSOL OR GSSSOLERB; HOWEVER, IF THE ORDER OF THE LINEAR SYSTEM IS VERY SMALL RELATIVE TO THE NUMBER OF BINARY DIGITS IN THE NUMBER REPRESENTATION, THEN DECSOL ALSO MAY BE USED.

KEYWORDS:

ALGEBRAIC EQUATIONS,
LINEAR SYSTEMS.

SUBSECTION: SOL .

CALLING SEQUENCE:

THE HEADING OF THIS PROCEDURE IS:
"PROCEDURE" SOL(A, N, P, B): "VALUE" N:
"INTEGER" N: "ARRAY" A, B; "INTEGER" "ARRAY" P;

THE MEANING OF THE FORMAL PARAMETERS IS:

A: <ARRAY IDENTIFIER>:
"ARRAY" A[1:N, 1:N]:
ENTRY: THE TRIANGULARLY DECOMPOSED FORM OF THE MATRIX OF THE LINEAR SYSTEM AS PRODUCED BY DEC (SECTION 3.1.1.1.1.1.1):

N: <ARITHMETIC EXPRESSION>:
THE ORDER OF THE MATRIX:

P: <ARRAY IDENTIFIER>;
"INTEGER""ARRAY" P[1:N]:
ENTRY:THE PIVOTAL INDICES, AS PRODUCED BY DEC.

B: <ARRAY IDENTIFIER>;
"ARRAY" B[1:N];
ENTRY: THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM:
EXIT: THE SOLUTION OF THE LINEAR SYSTEM.

PROCEDURES USED:

MATVEC = CP34011.

RUNNING TIME: PROPORTIONAL TO N ** 2.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

SOL SHOULD BE CALLED AFTER DEC (SECTION 3.1.1.1.1.1.1) AND SOLVES THE LINEAR SYSTEM WITH A MATRIX, WHOSE TRIANGULARLY DECOMPOSED FORM AS PRODUCED BY DEC IS GIVEN IN ARRAY A, AND A RIGHT-HAND SIDE AS GIVEN IN ARRAY B: SOL LEAVES A AND P UNALTERED, SO, AFTER ONE CALL OF DEC, SEVERAL CALLS OF SOL MAY FOLLOW FOR SOLVING SEVERAL SYSTEMS HAVING THE SAME MATRIX BUT DIFFERENT RIGHT-HAND SIDES.

EXAMPLE OF USE: SEE DECSOL (THIS SECTION).

SUBSECTION: DECSOL .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DECSOL(A, N, AUX, B); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N, 1:N];
            ENTRY:THE N-TH ORDER MATRIX;
            EXIT: THE CALCULATED LOWER-TRIANGULAR MATRIX AND UNIT
            UPPERTRIANGULAR MATRIX WITH ITS UNIT DIAGONAL OMITTED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[1:3];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE: A REASONABLE CHOICE FOR THIS
                    VALUE IS AN ESTIMATE OF THE RELATIVE PRECISION OF
                    THE MATRIX ELEMENTS; HOWEVER, IT SHOULD NOT BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            EXIT:
            AUX[1]: IF R IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE AUX[3]), THEN AUX[1] EQUALS 1 IF THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX OF ORDER R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED; IF
                    AUX[3] < N THEN THE PROCESS IS TERMINATED AND NO
                    SOLUTION WILL BE CALCULATED;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   IF AUX[3] = N, THEN THE CALCULATED SOLUTION OF THE
                    LINEAR SYSTEM IS OVERWRITTEN ON B, ELSE B REMAINS
                    UNALTERED.
```

PROCEDURES USED:

DEC = CP34300,
SOL = CP34051.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: DECSOL DECLARES AN AUXILIARY ARRAY OF TYPE INTEGER AND ORDER N.

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

DECSOL USES DEC TO PERFORM THE TRIANGULAR DECOMPOSITION OF THE MATRIX AND SOL TO CALCULATE THE SOLUTION WITH FORWARD AND BACK SUBSTITUTION: SINCE DECSOL MAY YIELD USELESS RESULTS, EVEN FOR WELL-CONDITIONED MATRICES (SEE DEC, SECTION 3.1.1.1.1.1.1), DECSOL SHOULD ONLY BE USED IF THE ORDER OF THE MATRIX IS SMALL RELATIVE TO THE NUMBER OF BINARY DIGITS IN THE NUMBER REPRESENTATION: IF AUX[3] < N, THEN THE EFFECT OF DECSOL IS MERELY THAT OF DEC.

EXAMPLE OF USE:

LET A BE THE FOURTH ORDER SEGMENT OF THE HILBERT MATRIX AND B THE THIRD COLUMN OF A, THEN THE SOLUTION OF THE LINEAR SYSTEM AX = B IS GIVEN BY THE THIRD UNIT VECTOR AND MAY BE CALCULATED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "INTEGER" I, J;
    "ARRAY" A[1:4, 1:4], B[1:4], AUX[1:3];
    "PROCEDURE" DECSOL(A, N, AUX, B); "CODE" 34301;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(B[I]);
        "FOR" I:= 1 "STEP" 2 "UNTIL" 3 "DO" ITEM(AUX[I])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("*, "("SOLUTION:")"B+.15D"+3D,/,3(10B+.15D"+3D,/),
    "("SIGN(DET) = ")"+D,/,"("NUMBER OF ELIMINATIONSTEPS = ")"
    +D")");

    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        A[I,J]:= 1 / (I + J - 1); B[I]:= A[I,3]
    "END";
    AUX[2]:= "-14;
    DECSOL(A, 4, AUX, B);
    OUTLIST(71, LAYOUT, LIST)
"END"
```

RESULTS:

```
SOLUTION: +.000000000000000"+000
          +.000000000000000"+000
          +.100000000000000"+001
          +.000000000000000"+000
SIGN(DET) = +1
NUMBER OF ELIMINATIONSTEPS = +4
```

SUBSECTION: SOLELM .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" SOLELM(A, N, RI, CI, B); "VALUE" N;
    "INTEGER" N; "ARRAY" A, B; "INTEGER" "ARRAY" RI, CI;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N, 1:N];
            ENTRY:  THE TRIANGULARLY  DECOMPOSED FORM  OF THE MATRIX OF
                    THE LINEAR SYSTEM AS  PRODUCED BY  GSSELM  (SECTION
                    3.1.1.1.1.1.1);
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    RI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" RI[1:N];
            ENTRY:THE PIVOTAL ROW INDICES,  AS PRODUCED BY  GSSELM;
    CI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" CI[1:N];
            ENTRY:THE PIVOTAL COLUMN INDICES, AS PRODUCED BY  GSSELM;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   THE SOLUTION OF THE LINEAR SYSTEM.
```

PROCEDURES USED:

SOL = CP34051.

RUNNING TIME: PROPORTIONAL TO N ** 2.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

SOLELM SHOULD BE CALLED AFTER GSSELM OR GSSERB (SECTION 3.1.1.1.1.1.1) AND SOLVES THE LINEAR SYSTEM WITH THE MATRIX, WHOSE TRIANGULARLY DECOMPOSED FORM AS PRODUCED BY GSSELM IS GIVEN IN ARRAY A, AND A RIGHT-HAND SIDE AS GIVEN IN ARRAY B; SOLELM LEAVES A, RI AND CI UNALTERED, SO, AFTER ONE CALL OF GSSELM OR GSSERB, SEVERAL CALLS OF SOLELM MAY FOLLOW FOR SOLVING SEVERAL SYSTEMS HAVING THE SAME MATRIX BUT DIFFERENT RIGHT-HAND SIDES.

EXAMPLE OF USE: SEE GSSSOL OR GSSSOLERB (THIS SECTION).

SUBSECTION: GSSSOL .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" GSSSOL(A, N, AUX, B); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N, 1:N];
            ENTRY:THE N-TH ORDER MATRIX;
            EXIT: THE  CALCULATED  LOWER-TRIANGULAR  MATRIX  AND  UNIT
            UPPERTRIANGULAR  MATRIX  WITH ITS  UNIT  DIAGONAL  OMITTED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[1:7];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE:  A REASONABLE CHOICE FOR THIS
                    VALUE  IS  AN ESTIMATE OF THE RELATIVE PRECISION OF
                    THE MATRIX ELEMENTS;  HOWEVER,  IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            AUX[4]: A VALUE WHICH IS USED FOR CONTROLLING PIVOTING (SEE
                    GSSELM, SECTION 3.1.1.1.1.1.1);
            EXIT:
            AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE  AUX[3]),   THEN   AUX[1]  EQUALS   1  IF  THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX  OF ORDER  R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                    AUX[3] < N  THEN  THE PROCESS  IS TERMINATED AND NO
                    SOLUTION WILL HAVE BEEN CALCULATED;
            AUX[5]: THE  MODULUS  OF  AN ELEMENT  WHICH  IS  OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX GIVEN IN ARRAY A;
            AUX[7]: AN UPPER BOUND FOR THE GROWTH (SEE GSSELM,  SECTION
                    3.1.1.1.1.1.1);
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   IF AUX[3] = N,  THEN THE CALCULATED SOLUTION OF THE
                    LINEAR SYSTEM IS OVERWRITTEN ON B,  ELSE  B REMAINS
                    UNALTERED.
```

PROCEDURES USED:

```
    SOLELM  = CP34061,
    GSSELM  = CP34231.
```

REQUIRED CENTRAL MEMORY:

```
    EXECUTION FIELD LENGTH: GSSSOL  DECLARES  TWO  AUXILIARY  ARRAYS OF
                            TYPE INTEGER AND ORDER N.
```

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

GSSSOL USES GSSELM (SECTION 3.1.1.1.1.1.1) TO PERFORM A TRIANGULAR DECOMPOSITION OF THE MATRIX AND SOLELM (THIS SECTION) TO CALCULATE THE SOLUTION OF THE GIVEN LINEAR SYSTEM; IF AUX[3] < N, THEN THE EFFECT OF GSSSOL IS MERELY THAT OF GSSELM.

EXAMPLE OF USE:

LET A BE THE FOURTH ORDER SEGMENT OF THE HILBERT MATRIX AND B THE THIRD COLUMN OF A, THEN THE SOLUTION OF THE LINEAR SYSTEM AX = B IS GIVEN BY THE THIRD UNIT VECTOR AND MAY BE CALCULATED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "INTEGER" I, J;
    "ARRAY" A[1:4, 1:4], B[1:4], AUX[1:7];
    "PROCEDURE" GSSSOL(A, N, AUX, B); "CODE" 34232;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(B[I]);
        "FOR" I:= 1 "STEP" 2 "UNTIL" 7 "DO" ITEM(AUX[I])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("*, "("SOLUTION:")"B+.15D"+3D,/,3(10B+.15D"+3D,/),
    "("SIGN(DET) = ")"+D,/,"("NUMBER OF ELIMINATIONSTEPS = ")"
    +D,/,"("MAX(ABS(A[I,J]))= ")"+.15D"+3D,/,
    "("UPPER BOUND GROWTH: ")"+.15D"+3D")");

    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        A[I,J]:= 1 / (I + J - 1); B[I]:= A[I,3]
    "END";
    AUX[2]:= "-14; AUX[4]:= 8;
    GSSSOL(A, 4, AUX, B);
    OUTLIST(71, LAYOUT, LIST)
"END"
```

RESULTS:

```
SOLUTION: +.888178419700120"-014
          -.497379915032070"-013
          +.100000000000010"+001
          +.000000000000000"+000
SIGN(DET) = +1
NUMBER OF ELIMINATIONSTEPS = +4
MAX(ABS(A[I,J]))= +.100000000000000"+001
UPPER BOUND GROWTH: +.159619047619050"+001
```

SUBSECTION: GSSSOLERB.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" GSSSOLERB(A, N, AUX, B); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N, 1:N];
            ENTRY:THE N-TH ORDER MATRIX;
            EXIT: THE  CALCULATED  LOWER-TRIANGULAR  MATRIX  AND  UNIT
            UPPERTRIANGULAR  MATRIX  WITH ITS  UNIT  DIAGONAL  OMITTED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[0:11];
            ENTRY:
            AUX[0]: THE MACHINE PRECISION;
            AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                    VALUE  IS  AN ESTIMATE OF THE RELATIVE PRECISION OF
                    THE MATRIX ELEMENTS;  HOWEVER,  IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            AUX[4]: A VALUE WHICH IS USED FOR CONTROLLING PIVOTING (SEE
                    GSSELM, SECTION 3.1.1.1.1.1.1);
            AUX[6]: AN UPPER BOUND  FOR  THE RELATIVE PRECISION  OF THE
                    GIVEN MATRIX ELEMENTS;
            EXIT:
            AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE  AUX[3]),   THEN   AUX[1]  EQUALS   1  IF  THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX  OF ORDER  R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                    AUX[3] < N  THEN  THE PROCESS  IS TERMINATED AND NO
                    SOLUTION OR ERROR BOUND WILL HAVE BEEN CALCULATED;
            AUX[5]: THE  MODULUS  OF  AN ELEMENT  WHICH  IS  OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX GIVEN IN ARRAY A;
            AUX[7]: AN UPPER BOUND FOR THE GROWTH (SEE GSSELM,  SECTION
                    3.1.1.1.1.1.1);
            AUX[9]: IF AUX[3] = N, THEN AUX[9] WILL EQUAL THE 1-NORM OF
                    THE INVERSE MATRIX, ELSE AUX[9] WILL BE UNDEFINED;
            AUX[11]: IF AUX[3] = N THEN THE VALUE OF AUX[11]  WILL BE A
                    ROUGH  UPPER BOUND  FOR  THE RELATIVE ERROR  IN THE
                    CALCULATED  SOLUTION  OF  THE  GIVEN LINEAR SYSTEM,
                    ELSE  AUX[11]  WILL BE UNDEFINED;  IF NO USE CAN BE
                    MADE OF THE FORMULA FOR THE ERROR BOUND AS GIVEN IN
                    SECTION 3.1.1.1.1.1.1 (SUBSECTION ERBELM),  BECAUSE
                    OF  A  VERY  BAD  CONDITION  OF  THE  MATRIX,  THEN
                    AUX[11]:= -1;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   IF AUX[3] = N,  THEN THE CALCULATED SOLUTION OF THE
                    LINEAR SYSTEM, ELSE  B REMAINS UNALTERED.
```

PROCEDURES USED:

SOLELM = CP34061,
GSSERB = CP34242.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: GSSSOLERB DECLARES TWO AUXILIARY ARRAYS OF TYPE INTEGER AND ORDER N.

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

GSSSOLERB USES GSSERB (SECTION 3.1.1.1.1.1.1) TO PERFORM THE TRIANGULAR DECOMPOSITION OF THE MATRIX AND TO CALCULATE AN UPPER BOUND FOR THE RELATIVE ERROR IN THE CALCULATED SOLUTION OF THE GIVEN LINEAR SYSTEM, AND SOLELM (THIS SECTION) TO CALCULATE THIS SOLUTION; IF AUX[3] < N, THEN THE EFFECT OF GSSSOLERB IS MERELY THAT OF GSSELM (SECTION 3.1.1.1.1.1.1).

EXAMPLE OF USE:
LET A BE THE FOURTH ORDER SEGMENT OF THE HILBERT MATRIX AND B THE THIRD COLUMN OF A, THEN THE SOLUTION OF THE LINEAR SYSTEM AX = B IS GIVEN BY THE THIRD UNIT VECTOR AND THIS SOLUTION, AS WELL AS AN UPPER BOUND FOR THE RELATIVE ERROR IN THE CALCULATED ONE , MAY BE OBTAINED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "INTEGER" I, J;
    "ARRAY" A[1:4, 1:4], B[1:4], AUX[0:11];
    "PROCEDURE" GSSSOLERB(A, N, AUX, B); "CODE" 34243;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(B[I]);
        "FOR" I:= 1 "STEP" 2 "UNTIL" 11 "DO" ITEM(AUX[I])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("*, "("SOLUTION:")"B+.15D"+3D,/,3(10B+.15D"+3D,/),
    "("SIGN(DET) = ")"+D,/,"("NUMBER OF ELIMINATIONSTEPS = ")"
    +D,/,"("MAX(ABS(A[I,J]))= ")"+.15D"+3D,/,
    "("UPPER BOUND GROWTH: ")"+.15D"+3D,/,
    "("1-NORM OF THE INVERSE MATRIX:")"B+.15D"+3D,/,
    "("UPPER BOUND REL. ERR. IN THE CALC. SOL.")"
    B+.15D"+3D")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        A[I,J]:= 1 / (I + J - 1); B[I]:= A[I,3]
    "END";
    AUX[0]:= AUX[2]:= "-14; AUX[4]:= 8; AUX[6]:= "-14;
    GSSSOLERB(A, 4, AUX, B);
    OUTLIST(71, LAYOUT, LIST)
"END"
```

```
RESULTS:

SOLUTION: +.888178419700120"-014
          -.497379915032070"-013
          +.100000000000010"+001
          +.000000000000000"+000
SIGN(DET) = +1
NUMBER OF ELIMINATIONSTEPS = +4
MAX(ABS(A[I,J]))= +.100000000000000"+001
UPPER BOUND GROWTH: +.159619047619050"+001
1-NORM OF THE INVERSE MATRIX: +.136199999998790"+005
UPPER BOUND REL. ERR. IN THE CALC. SOL. +.277896269157090"-007
```

REFERENCES:

[1] BUS, J. C. P.
LINEAR SYSTEMS WITH CALCULATION OF ERROR BOUNDS AND ITERATIVE REFINEMENT (DUTCH).
MATHEMATICAL CENTRE, AMSTERDAM, LR 3. 4. 19 (1972).

[2] DEKKER, T. J.
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1.
MATHEMATICAL CENTRE, AMSTERDAM, TRACT 22 (1968).

SOURCE TEXT(S):

```
"CODE" 34051;
    "PROCEDURE" SOL(A, N, P, B); "VALUE" N; "INTEGER" N; "ARRAY" A, B;
    "INTEGER" "ARRAY" P;
    "BEGIN" "INTEGER" K, PK;
        "REAL" R;
        "REAL" "PROCEDURE" MATVEC(L, U, I, A, B); "CODE" 34011;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" R:= B[K]; PK:= P[K];
            B[K]:= (B[PK] - MATVEC(1, K - 1, K, A, B)) / A[K,K];
            "IF" PK ^= K "THEN" B[PK]:= R
        "END";
        "FOR" K:= N "STEP" - 1 "UNTIL" 1 "DO"
        B[K]:= B[K] - MATVEC(K + 1, N, K, A, B)
    "END" SOL;
        "EOP"
```

```
"CODE" 34301;
     "PROCEDURE" DECSOL(A, N, AUX, B); "VALUE" N; "INTEGER" N;
     "ARRAY" A, AUX, B;
     "BEGIN" "INTEGER" "ARRAY" P[1:N];
          "PROCEDURE" SOL(A, N, P, B); "CODE" 34051;
          "PROCEDURE" DEC(A, N, AUX, P); "CODE" 34300;
          DEC(A, N, AUX, P);
          "IF" AUX[3] = N "THEN" SOL(A, N, P, B)
     "END" DECSOL;
          "EOP"

"CODE" 34061;
     "PROCEDURE" SOLELM(A, N, RI, CI, B); "VALUE" N; "INTEGER" N;
     "ARRAY" A, B;
     "INTEGER" "ARRAY" RI, CI;
     "BEGIN" "INTEGER" R, CIR;
          "REAL" W;
          "PROCEDURE" SOL(A, N, P, B); "CODE" 34051;
          SOL(A, N, RI, B);
          "FOR" R:= N "STEP" - 1 "UNTIL" 1 "DO"
          "BEGIN" CIR:= CI[R]; "IF" CIR ^= R "THEN"
               "BEGIN" W:= B[R]; B[R]:= B[CIR]; B[CIR]:= W "END"
          "END"
     "END" SOLELM;
          "EOP"
"CODE" 34232;
     "PROCEDURE" GSSSOL(A, N, AUX, B); "VALUE" N; "INTEGER" N;
     "ARRAY" A, AUX, B;
     "BEGIN" "INTEGER" "ARRAY" RI, CI[1:N];
          "PROCEDURE" SOLELM(A, N, RI, CI, B); "CODE" 34061;
          "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "CODE" 34231;
          GSSELM(A, N, AUX, RI, CI);
          "IF" AUX[3] = N "THEN" SOLELM(A, N, RI, CI, B)
     "END" GSSSOL;
          "EOP"

"CODE" 34243;
     "PROCEDURE" GSSSOLERB(A, N, AUX, B); "VALUE" N; "INTEGER" N;
     "ARRAY" A, AUX, B;
     "BEGIN" "INTEGER" "ARRAY" RI, CI[1:N];
          "PROCEDURE" SOLELM(A, N, RI, CI, B); "CODE" 34061;
          "PROCEDURE" GSSERB(A, N, AUX, RI, CI); "CODE" 34242;
          GSSERB(A, N, AUX, RI, CI);
          "IF" AUX[3] = N "THEN" SOLELM(A, N, RI, CI, B)
     "END" GSSSOLERB;
          "EOP"
```

AUTHORS: J. C. P. BUS AND T. J. DEKKER.

CONTRIBUTOR: J.C.P. BUS AND P.A. BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730920.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS FIVE PROCEDURES FOR INVERSION OF MATRICES:
INV CALCULATES THE INVERSE OF A MATRIX THAT HAS BEEN TRIANGULARLY DECOMPOSED BY DEC;
DECINV CALCULATES THE INVERSE OF A MATRIX WHOSE ORDER IS SMALL RELATIVE TO THE NUMBER OF BINARY DIGITS IN THE NUMBER REPRESENTATION;
INV1 CALCULATES THE INVERSE OF A MATRIX THAT HAS BEEN TRIANGULARLY DECOMPOSED BY GSSELM OR GSSERB.THE 1-NORM OF THE INVERSE MATRIX MIGHT ALSO BE CALCULATED
GSSINV CALCULATES THE INVERSE OF A MATRIX:
GSSINVERB CALCULATES THE INVERSE OF A MATRIX AND ITS 1-NORM.
A ROUGH UPPERBOUND FOR THE RELATIVE ERROR IN THE CALCULATED INVERSE MATRIX IS ALSO GIVEN;

THE DIFFERENCE BETWEEN DECINV ON THE ONE SIDE AND GSSINV AND GSSINVERB ON THE OTHER SIDE LIES IN THE METHOD USED FOR TRIANGULAR DECOMPOSITION, PARTICULARLY IN THE PIVOTING STRATEGY; DECINV USES DEC, GSSINV AND GSSINVERB USE GSSELM TO PERFORM THE TRIANGULAR DECOMPOSITION; THE USER IS ADVISED TO USE GSSINV OR GSSINVERB (SEE SECTION 3.1.1.1.1.1.1).

KEYWORDS:

MATRIX INVERSION.

SUBSECTION: INV .

CALLING SEQUENCE:

```
THE HEADING OF THIS PROCEDURE IS:
"PROCEDURE" INV(A, N, P); "VALUE" N;
"INTEGER" N; "ARRAY" A; "INTEGER" "ARRAY" P;

THE MEANING OF THE FORMAL PARAMETERS IS:
A:      <ARRAY IDENTIFIER>;
        "ARRAY" A[1:N, 1:N];
        ENTRY:  THE TRIANGULARLY  DECOMPOSED FORM  OF THE MATRIX AS
                PRODUCED BY  DEC  (SECTION 3.1.1.1.1.1.1);
        EXIT:   THE CALCULATED INVERSE MATRIX;
N:      <ARITHMETIC EXPRESSION>;
        THE ORDER OF THE MATRIX;
P:      <ARRAY IDENTIFIER>;
        "INTEGER""ARRAY P[1:N];
        ENTRY:THE PIVOTAL INDICES, AS PRODUCED BY DEC;
```

PROCEDURES USED:

```
MATMAT     = CP34013,
ICHCOL     = CP34031,
DUPCOLVEC  = CP31034.
```

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: INV DECLARES AN AUXILIARY ARRAY OF TYPE REAL AND ORDER N.

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

INV SHOULD BE CALLED AFTER DEC (SECTION 3.1.1.1.1.1.1) AND CALCULATES THE INVERSE OF THE MATRIX, WHOSE TRIANGULARLY DECOMPOSED FORM AS PRODUCED BY DEC IS GIVEN IN ARRAY A; THE INVERSE MATRIX IS OVERWRITTEN ON A.

EXAMPLE OF USE: SEE DECINV (THIS SECTION).

SUBSECTION: DECINV .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DECINV(A, N, AUX); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N, 1:N];
            ENTRY:  THE MATRIX, WHOSE INVERSE HAS TO BE CALCULATED;
            EXIT:   IF  AUX[3] = N, THEN THE CALCULATED INVERSE MATRIX;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[1:3];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                    VALUE  IS  AN ESTIMATE OF THE RELATIVE PRECISION OF
                    THE MATRIX ELEMENTS;  HOWEVER,  IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            EXIT:
            AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE  AUX[3]),   THEN   AUX[1]  EQUALS   1  IF  THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX  OF ORDER  R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                    AUX[3] < N, THEN  THE PROCESS  IS TERMINATED AND NO
                    INVERSE WILL HAVE BEEN CALCULATED.
```

PROCEDURES USED:

```
    DEC = CP34300,
    INV = CP34053.
```

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: DECINV DECLARES AN AUXILIARY ARRAY OF TYPE INTEGER AND ORDER N.

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

DECINV USES DEC (SECTION 3.1.1.1.1.1.1) TO PERFORM THE TRIANGULAR DECOMPOSITION OF A MATRIX AND INV TO CALCULATE ITS INVERSE: DECINV SHOULD ONLY BE USED IF THE ORDER OF THE MATRIX IS SMALL RELATIVE TO THE NUMBER OF BINARY DIGITS IN THE NUMBER REPRESENTATION (SEE DEC); IF AUX[3] < N, THEN THE EFFECT OF DECINV IS MERELY THAT OF DEC.

EXAMPLE OF USE:

THE FOLLOWING PROGRAM CALCULATES THE INVERSE OF THE INPUT MATRIX AND PRINTS THE RESULTS:

```
"BEGIN"
    "ARRAY" A[1:4, 1:4], AUX[1:3];
    "PROCEDURE" DECINV(A, N, AUX); "CODE" 34302;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I, J;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
        "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(A[I,J])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("4(4B,4(B+ZDB),/),/")");

    INLIST(70, LAYOUT, LIST); AUX[2]:= "-14;
    OUTPUT(71,"("/,"("CALCULATED INVERSE:")",/")");
    DECINV(A, 4, AUX);
    OUTLIST(71, LAYOUT, LIST);
    OUTPUT(71, "(""("AUX[1]=")"B+D,/,"("AUX[3]=")"B+D")",
    AUX[1], AUX[3])
"END"
```

INPUT:

```
 + 4   + 2   + 4   + 1
 +30   +20   +45   +12
 +20   +15   +36   +10
 +35   +28   +70   +20
```

RESULTS:

```
CALCULATED INVERSE:
  +4    -2    +4    -1
 -30   +20   -45   +12
 +20   -15   +36   -10
 -35   +28   -70   +20

AUX[1]= +1
AUX[3]= +4
```

SUBSECTION: INV1 .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "REAL" "PROCEDURE" INV1(A, N, RI, CI, WITHNORM);
    "VALUE" N, WITHNORM; "INTEGER" N; "BOOLEAN" WITHNORM; "ARRAY" A;
    "INTEGER" "ARRAY" RI, CI;

    INV1:   IF THE VALUE OF  WITHNORM  IS TRUE, THEN THE VALUE OF  INV1
            WILL EQUAL THE  1-NORM  OF THE CALCULATED  INVERSE  MATRIX,
            ELSE INV1:= 0;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N, 1:N];
            ENTRY:  THE TRIANGULARLY  DECOMPOSED FORM  OF THE MATRIX AS
                    PRODUCED BY  GSSELM  (SECTION 3.1.1.1.1.1.1);
            EXIT:   THE CALCULATED INVERSE MATRIX;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    RI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" RI[1:N];
            ENTRY:THE PIVOTAL ROW INDICES,  AS PRODUCED BY  GSSELM;
    CI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" CI[1:N];
            ENTRY:THE PIVOTAL COLUMN INDICES, AS PRODUCED BY  GSSELM;
    WITHNORM: <BOOLEAN EXPRESSION>;
            IF THE VALUE OF  WITHNORM  IS TRUE,  THEN THE 1-NORM OF THE
            INVERSE MATRIX  WILL  BE CALCULATED  AND  ASSIGNED TO INV1,
            ELSE INV1:= 0;
```

PROCEDURES USED:

```
    ICHROW = CP34032,
    INV    = CP34053.
```

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

INV1 SHOULD BE CALLED AFTER GSSELM OR GSSERB (SECTION 3.1.1.1.1.1.1), WHICH DELIVERS THE TRIANGULARLY DECOMPOSED FORM OF A MATRIX; INV1 CALCULATES THE INVERSE MATRIX AND ALSO ITS 1-NORM MIGHT BE CALCULATED; THE INVERSE MATRIX IS OVERWRITTEN ON A.

EXAMPLE OF USE: SEE GSSINV AND GSSINVERB (THIS SECTION).

```
SUBSECTION: GSSINV   .

CALLING SEQUENCE:

    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" GSSINV(A, N, AUX); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N, 1:N];
            ENTRY:  THE MATRIX, WHOSE INVERSE HAS TO BE CALCULATED;
            EXIT:   IF  AUX[3] = N, THEN THE CALCULATED INVERSE MATRIX;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[1:9];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE:  A REASONABLE CHOICE FOR THIS
                    VALUE  IS  AN ESTIMATE OF THE RELATIVE PRECISION OF
                    THE MATRIX ELEMENTS;  HOWEVER,  IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION:
            AUX[4]: A VALUE WHICH IS USED FOR CONTROLLING PIVOTING (SEE
                    GSSELM, SECTION 3.1.1.1.1.1.1);
            EXIT:
            AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE  AUX[3]),   THEN   AUX[1]  EQUALS   1  IF  THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX  OF ORDER  R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                    AUX[3] < N  THEN  THE PROCESS  IS TERMINATED AND NO
                    SOLUTION WILL BE CALCULATED:
            AUX[5]: THE  MODULUS  OF  AN ELEMENT  WHICH  IS  OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX GIVEN IN ARRAY A;
            AUX[7]: AN UPPER BOUND FOR THE GROWTH (SEE GSSELM,  SECTION
                    3.1.1.1.1.1.1):
            AUX[9]: IF AUX[3] = N, THEN AUX[9] WILL EQUAL THE 1-NORM OF
                    THE CALCULATED INVERSE MATRIX, ELSE AUX[9]  WILL BE
                    UNDEFINED.

PROCEDURES USED:

    INV1    = CP34235,
    GSSELM  = CP34231.

REQUIRED CENTRAL MEMORY:

    EXECUTION FIELD LENGTH: GSSINV  DECLARES  TWO  AUXILIARY  ARRAYS OF
                            TYPE INTEGER AND ORDER N.

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE:   ALGOL 60.
```

METHOD AND PERFORMANCE:

GSSINV USES GSSELM (SECTION 3.1.1.1.1.1.1) TO PERFORM A TRIANGULAR DECOMPOSITION OF THE MATRIX AND INV1 (THIS SECTION) TO CALCULATE THE INVERSE MATRIX; IF AUX[3] = N, THEN THE EFFECT OF GSSINV IS MERELY THAT OF GSSELM.

EXAMPLE OF USE:

THE FOLLOWING PROGRAM CALCULATES THE INVERSE OF THE INPUT MATRIX AND PRINTS THE RESULTS:

```
"BEGIN"
    "ARRAY" A[1:4, 1:4], AUX[1:9];
    "PROCEDURE" GSSINV(A, N, AUX); "CODE" 34236;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I, J;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
        "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(A[I,J])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("4(4B,4(B+ZDB),/),/")");

    INLIST(70, LAYOUT, LIST); AUX[2]:= "-14; AUX[4]:= 8;
    GSSINV(A, 4, AUX);
    OUTPUT(71,"("/,"("CALCULATED INVERSE:")",/")");
    OUTLIST(71, LAYOUT, LIST);
    OUTPUT(71, "("4B"("AUX ELEMENTS:")",/,2(4B+D,/),
    3(4B+.15D"+3D,/)")", AUX[1], AUX[3], AUX[5], AUX[7], AUX[9])
"END"
```

INPUT:

```
 + 4   + 2   + 4   + 1
 +30   +20   +45   +12
 +20   +15   +36   +10
 +35   +28   +70   +20
```

RESULTS:

```
CALCULATED INVERSE:
  +4    -2    +4    -1
 -30   +20   -45   +12
 +20   -15   +36   -10
 -35   +28   -70   +20

AUX ELEMENTS:
+1
+4
+.700000000000000"+002
+.112528571428570"+003
+.154999999999730"+003
```

SUBSECTION: GSSINVERB.

CALLING SEQUENCE:

THE HEADING OF THIS PROCEDURE IS:
"PROCEDURE" GSSINVERB(A, N, AUX); "VALUE" N;
"INTEGER" N; "ARRAY" A, AUX;

THE MEANING OF THE FORMAL PARAMETERS IS:

A: <ARRAY IDENTIFIER>;
"ARRAY" A[1:N, 1:N];
ENTRY: THE MATRIX, WHOSE INVERSE HAS TO BE CALCULATED;
EXIT: IF AUX[3] = N, THEN THE CALCULATED INVERSE MATRIX;

N: <ARITHMETIC EXPRESSION>;
THE ORDER OF THE MATRIX;

AUX: <ARRAY IDENTIFIER>;
"ARRAY" AUX[0:11];
ENTRY:
AUX[0]: THE MACHINE PRECISION;
AUX[2]: A RELATIVE TOLERANCE; A REASONABLE CHOICE FOR THIS VALUE IS AN ESTIMATE OF THE RELATIVE PRECISION OF THE MATRIX ELEMENTS; HOWEVER, IT SHOULD NOT BE CHOSEN SMALLER THAN THE MACHINE PRECISION;
AUX[4]: A VALUE WHICH IS USED FOR CONTROLLING PIVOTING (SEE GSSELM, SECTION 3.1.1.1.1.1.1);
AUX[6]: AN UPPER BOUND FOR THE RELATIVE PRECISION OF THE GIVEN MATRIX ELEMENTS;
EXIT:
AUX[1]: IF R IS THE NUMBER OF ELIMINATION STEPS PERFORMED (SEE AUX[3]), THEN AUX[1] EQUALS 1 IF THE DETERMINANT OF THE PRINCIPAL SUBMATRIX OF ORDER R IS POSITIVE, ELSE AUX[1] = -1;
AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED; IF AUX[3] < N THEN THE PROCESS IS TERMINATED AND NO SOLUTION WILL HAVE BEEN CALCULATED;
AUX[5]: THE MODULUS OF AN ELEMENT WHICH IS OF MAXIMUM ABSOLUTE VALUE FOR THE MATRIX GIVEN IN ARRAY A;
AUX[7]: AN UPPER BOUND FOR THE GROWTH (SEE GSSELM, SECTION 3.1.1.1.1.1.1);
AUX[9]: IF AUX[3] = N, THEN AUX[9] WILL EQUAL THE 1-NORM OF THE INVERSE MATRIX, ELSE AUX[9] WILL BE UNDEFINED;
AUX[11]: IF AUX[3] = N THEN THE VALUE OF AUX[11] WILL BE A ROUGH UPPER BOUND FOR THE RELATIVE ERROR IN THE CALCULATED INVERSE MATRIX, ELSE AUX[11] WILL BE BE UNDEFINED; IF NO USE CAN BE MADE OF THE FORMULA FOR THE ERROR BOUND AS GIVEN IN SECTION 3.1.1.1.1.1.1 (SUBSECTION ERBELM), BECAUSE OF A VERY BAD CONDITION OF THE MATRIX, THEN AUX[11]:=-1.

PROCEDURES USED:

```
INV1    = CP34235,
GSSELM  = CP34231,
ERBELM  = CP34241.
```

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: GSSINVERB DECLARES TWO AUXILIARY ARRAYS OF TYPE INTEGER AND ORDER N.

RUNNING TIME: PROPORTIONAL TO N ** 3.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

GSSINVERB USES GSSELM (SECTION 3.1.1.1.1.1.1) TO PERFORM THE TRIANGULAR DECOMPOSITION OF THE MATRIX, INV1 (THIS SECTION) TO CALCULATE THE INVERSE MATRIX AND ITS 1-NORM AND ERBELM (SECTION 3.1.1.1.1.1) TO CALCULATE AN UPPER BOUND FOR THE RELATIVE ERROR IN THE CALCULATED INVERSE: IF AUX[3] < N, THEN THE EFFECT OF GSSINVERB IS MERELY THAT OF GSSELM.

EXAMPLE OF USE:

THE FOLLOWING PROGRAM CALCULATES THE INVERSE OF THE INPUT MATRIX WITH AN UPPER BOUND FOR THE RELATIVE ERROR IN IT AND PRINTS THE RESULTS:

```
"BEGIN"
    "ARRAY" A[1:4, 1:4], AUX[0:11];
    "PROCEDURE" GSSINVERB(A, N, AUX); "CODE" 34244;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I, J;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
        "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(A[I,J])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("4(4B,4(B+ZDB),/),/")");

    INLIST(70, LAYOUT, LIST); AUX[0]:= AUX[2]:= AUX[6]:= "-14;
    AUX[4]:= 8; GSSINVERB(A, 4, AUX);
    OUTPUT(71,"("/,"("CALCULATED INVERSE:")",/")");
    OUTLIST(71, LAYOUT, LIST);
    OUTPUT(71, "("4B"("AUX ELEMENTS:")",/,2(4B+D,/),
    4(4B+.15D"+3D,/)")", AUX[1], AUX[3], AUX[5], AUX[7], AUX[9],
    AUX[11])
"END"
```

INPUT:

```
+ 4   + 2   + 4   + 1
+30   +20   +45   +12
+20   +15   +36   +10
+35   +28   +70   +20
```

RESULTS:

CALCULATED INVERSE:

```
 +4    -2    +4    -1
-30   +20   -45   +12
+20   -15   +36   -10
-35   +28   -70   +20
```

AUX ELEMENTS:

```
+1
+4
+.700000000000000"+002
+.112528571428570"+003
+.154999999999730"+003
+.222946341369190"-007
```

REFERENCES:

[1] BUS, J. C. P.
LINEAR SYSTEMS WITH CALCULATION OF ERROR BOUNDS AND ITERATIVE REFINEMENT (DUTCH).
MATHEMATICAL CENTRE, AMSTERDAM, LR 3. 4. 19 (1972).

[2] DEKKER, T. J.
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1.
MATHEMATICAL CENTRE, AMSTERDAM, TRACT 22 (1968).

SOURCE TEXT(S):

```
"CODE" 34053;
    "PROCEDURE" INV(A, N, P); "VALUE" N; "INTEGER" N; "ARRAY" A;
    "INTEGER" "ARRAY" P;
    "BEGIN" "INTEGER" J, K, K1;
        "REAL" R;
        "ARRAY" V[1:N];
        "REAL" "PROCEDURE" MATMAT(L, U, I, J, A, B); "CODE" 34013;
        "PROCEDURE" ICHCOL(L, U, I, J, A); "CODE" 34031;
        "PROCEDURE" DUPCOLVEC(L, U, J, A, B); "CODE" 31034;
        "FOR" K:= N "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" K1:= K + 1;
            "FOR" J:= N "STEP" - 1 "UNTIL" K1 "DO"
            "BEGIN" A[J,K1]:= V[J];
                V[J]:= - MATMAT(K1, N, K, J, A, A)
            "END";
            R:= A[K,K];
            "FOR" J:= N "STEP" - 1 "UNTIL" K1 "DO"
            "BEGIN" A[K,J]:= V[J];
                V[J]:= - MATMAT(K1, N, J, K, A, A) / R
            "END";
            V[K]:= (1 - MATMAT(K1, N, K, K, A, A)) / R
        "END";
        DUPCOLVEC(1, N, 1, A, V);
        "FOR" K:= N - 1 "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" K1:= P[K]; "IF" K1 ^= K "THEN"
            ICHCOL(1, N, K, K1, A)
        "END"
    "END" INV;
        "EOP"

"CODE" 34302;
    "PROCEDURE" DECINV(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "BEGIN" "INTEGER" "ARRAY" P[1:N];
        "PROCEDURE" DEC(A, N, AUX, P); "CODE" 34300;
        "PROCEDURE" INV(A, N, P); "CODE" 34053;
        DEC(A, N, AUX, P); "IF" AUX[3] = N "THEN" INV(A, N, P)
    "END" DECINV;
        "EOP"
```

```
"CODE" 34235;
    "REAL" "PROCEDURE" INV1(A, N, RI, CI, WITHNORM);
    "VALUE" N, WITHNORM; "INTEGER" N; "BOOLEAN" WITHNORM;
    "ARRAY" A; "INTEGER" "ARRAY" RI, CI;
    "BEGIN" "INTEGER" L, K, K1;
        "REAL" AID, NRMINV;
        "PROCEDURE" ICHROW(L, U, I, J, A); "CODE" 34032;
        "PROCEDURE" INV(A, N, P); "CODE" 34053;
        INV(A, N, RI); NRMINV:= 0; "IF" WITHNORM "THEN"
        "FOR" L:= 1 "STEP" 1 "UNTIL" N "DO"
        NRMINV:= NRMINV + ABS(A[L,N]);
        "FOR" K:= N - 1 "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" "IF" WITHNORM "THEN"
            "BEGIN" AID:= 0;
                "FOR" L:= 1 "STEP" 1 "UNTIL" N "DO"
                AID:= AID + ABS(A[L,K]);
                "IF" NRMINV < AID "THEN" NRMINV:= AID
            "END";
            K1:= CI[K]; "IF" K1 ^= K "THEN" ICHROW(1, N, K, K1, A)
        "END";
        INV1:= NRMINV
    "END" INV1;
        "EOP"

"CODE" 34236;
    "PROCEDURE" GSSINV(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "BEGIN" "INTEGER" "ARRAY" RI, CI[1:N];
        "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "CODE" 34231;
        "REAL" "PROCEDURE" INV1(A, N, RI, CI, WITHNORM); "CODE" 34235;
        GSSELM(A, N, AUX, RI, CI);
        "IF" AUX[3] = N "THEN" AUX[9]:= INV1(A, N, RI, CI, "TRUE")
    "END" GSSINV;
        "EOP"

"CODE" 34244;
    "PROCEDURE" GSSINVERB(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "BEGIN" "INTEGER" "ARRAY" RI, CI[1:N];
        "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "CODE" 34231;
        "REAL" "PROCEDURE" INV1(A, N, RI, CI, WITHNORM); "CODE" 34235;
        "PROCEDURE" ERBELM(N, AUX, NRMINV); "CODE" 34241;
        GSSELM(A, N, AUX, RI, CI);
        "IF" AUX[3] = N "THEN"
        ERBELM(N, AUX, INV1(A, N, RI, CI, "TRUE"))
    "END" GSSINVERB;
        "EOP"
```

AUTHOR: J. C. P. BUS.

CONTRIBUTOR: J.C.P. BUS AND P. A. BEENTJES.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 731008.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS FOUR PROCEDURES FOR CALCULATING AN ITERATIVELY IMPROVED SOLUTION OF A SYSTEM OF LINEAR EQUATIONS:
ITISOL SOLVES A LINEAR SYSTEM WHOSE MATRIX HAS BEEN TRIANGULARLY DECOMPOSED BY GSSELM OR GSSERB. THIS SOLUTION THUS OBTAINED NUMERICALLY, IS IMPROVED ITERATIVELY;
GSSITISOL SOLVES A LINEAR SYSTEM AND THIS SOLUTION THUS OBTAINED NUMERICALLY, IS IMPROVED ITERATIVELY;
ITISOLERB SOLVES A LINEAR SYSTEM WHOSE MATRIX HAS BEEN TRIANGULARLY DECOMPOSED BY GSSNRI.THIS SOLUTION IS IMPROVED ITERATIVELY.MOREOVER A REALISTIC UPPERBOUND FOR THE RELATIVE ERROR IN THE SOLUTION IS CALCULATED.
GSSITISOLERB SOLVES A LINEAR SYSTEM.THIS SOLUTION IS IMPROVED ITERATIVELY AND A REALISTIC UPPERBOUND FOR THE RELATIVE ERROR IN THE SOLUTION IS CALCULATED;

KEYWORDS:

ALGEBRAIC EQUATIONS,
LINEAR SYSTEMS,
ITERATIVE REFINEMENT.

SUBSECTION: ITISOL .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" ITISOL(A, LU, N, AUX, RI, CI, B); "VALUE" N;
    "INTEGER" N; "ARRAY" A, LU, AUX, B; "INTEGER" "ARRAY" RI, CI;
    "CODE" 34250;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N,1:N];
            ENTRY:THE MATRIX OF THE LINEAR SYSTEM;
    LU:     <ARRAY IDENTIFIER>;
            "ARRAY" LU[1:N, 1:N];
            ENTRY:THE TRIANGULARLY DECOMPOSED FORM OF THE MATRIX GIVEN
            IN A, AS DELIVERED BY GSSELM  (SECTION 3.1.1.1.1.1.1);
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[10:13];
            ENTRY:
            AUX[10]: A RELATIVE TOLERANCE  FOR THE SOLUTION VECTOR;  IF
                     THE  1-NORM  OF THE VECTOR  OF CORRECTIONS  TO  THE
                     SOLUTION,  DIVIDED BY  THE 1-NORM OF THE CALCULATED
                     SOLUTION, IS SMALLER THAN AUX[10], THEN THE PROCESS
                     WILL STOP; THE USER SHOULD NOT CHOOSE  THE VALUE OF
                     AUX[10] SMALLER THAN THE RELATIVE PRECISION  OF THE
                     ELEMENTS OF THE MATRIX  AND THE RIGHT-HAND SIDE  OF
                     THE LINEAR SYSTEM;
            AUX[12]: THE MAXIMUM NUMBER  OF ITERATIONS  ALLOWED FOR THE
                     REFINEMENT  OF  THE  SOLUTION;  IF  THE  NUMBER  OF
                     ITERATIONS EXCEEDS THE VALUE OF  AUX[12],  THEN THE
                     PROCESS  WILL  BE BROKEN OFF;  USUALLY  AUX[12] = 5
                     WILL GIVE GOOD RESULTS;
            EXIT:
            AUX[11]: THE 1-NORM  OF  THE VECTOR  OF  CORRECTIONS TO THE
                     SOLUTION IN THE LAST ITERATION STEP, DIVIDED BY THE
                     1-NORM OF THE CALCULATED SOLUTION;
                     IF  AUX[11] > AUX[10],  THEN  THE PROCESS  HAS BEEN
                     BROKEN OFF,  BECAUSE  THE  NUMBER  OF  ITERATIONS
                     EXCEEDED THE VALUE GIVEN IN AUX[12];
            AUX[13]: THE 1-NORM OF THE RESIDUAL VECTOR  (SEE METHOD AND
                     PERFORMANCE;
    RI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" RI[1:N];
            ENTRY:THE PIVOTAL ROW-INDICES,  AS PRODUCED BY  GSSELM;
    CI:     <ARRAY IDENTIFIER>;
            "INTEGER""ARRAY" CI[1:N];
            ENTRY:THE PIVOTAL COLUMN-INDICES, AS PRODUCED BY  GSSELM;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   THE CALCULATED  SOLUTION OF THE LINEAR SYSTEM.
```

PROCEDURES USED:

```
    SOLELM      = CP34061,
    INIVEC      = CP31010,
    DUPVEC      = CP31030,
    LNGMATVEC   = CP34411.
```

REQUIRED CENTRAL MEMORY:

TWO REAL ARRAYS, BOTH OF ORDER N, ARE DECLARED.

RUNNING TIME: THE NUMBER OF ARITHMETICAL OPERATIONS IN EACH ITERATION STEP IS PROPORTIONAL TO N ** 2.

METHOD AND PERFORMANCE:

ITISOL SHOULD BE CALLED AFTER GSSELM OR GSSERB (SECTION 3.1.1.1.1.1.1) AND SOLVES THE LINEAR SYSTEM WITH A MATRIX GIVEN IN ARRAY A, AND A RIGHT-HAND SIDE GIVEN IN ARRAY B; ONCE A SOLUTION IS CALCULATED WITH SOLELM (SECTION 3.1.1.1.1.1.3), THIS SOLUTION WILL BE REFINED ITERATIVELY UNTIL THE CALCULATED RELATIVE CORRECTION TO THIS SOLUTION WILL BE LESS THAN A PRESCRIBED VALUE (SEE AUX[10]);
EACH ITERATION OF THE REFINEMENT PROCESS CONSISTS OF THE FOLLOWING THREE STEPS (SEE [1], [2], [3]):

1 CALCULATE, IN DOUBLE PRECISION, THE RESIDUAL VECTOR R, DEFINED BY: R = AX - B,
WHERE X DENOTES THE SOLUTION, OBTAINED IN THE PREVIOUS ITERATION, B THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM, GIVEN IN B[1:N], AND A THE MATRIX GIVEN IN A[1:N, 1:N];
2 CALCULATE THE SOLUTION C, SAY, OF THE LINEAR SYSTEM: AC = R, WITH THE AID OF THE TRIANGULARLY DECOMPOSED MATRIX AS GIVEN IN LU[1:N, 1:N];
3 CALCULATE THE NEW SOLUTION: XNEW = X - C;

CONDITION OF THE MATRIX IS NOT TOO BAD, THEN THE PRECISION OF THE CALCULATED SOLUTION WILL BE OF THE ORDER OF THE PRECISION ASKED FOR IN AUX[10]; HOWEVER, IF THE CONDITION OF THE MATRIX IS VERY BAD, THEN THIS PROCESS WILL POSSIBLY NOT CONVERGE OR, IN EXCEPTIONAL CASES, CONVERGE TO A USELESS RESULT; IF THE USER WANTS TO MAKE CERTAIN ABOUT THE PRECISION OF THE CALCULATED SOLUTION, THEN HE HAS TO USE ITISOLERB (THIS SECTION), WHICH NEEDS THE CALCULATION (OF ORDER N ** 3) OF THE INVERSE MATRIX TO GET AN UPPER BOUND FOR THE CONDITION NUMBER AND WHICH GIVES A REALISTIC UPPER BOUND FOR THE RELATIVE ERROR IN THE CALCULATED SOLUTION;
ITISOL LEAVES A, LU, RI AND CI UNALTERED, SO AFTER ONE CALL OF GSSELM SEVERAL CALLS OF ITISOL MAY FOLLOW TO CALCULATE THE SOLUTION OF SEVERAL LINEAR SYSTEMS WITH THE SAME MATRIX BUT DIFFERENT RIGHT-HAND SIDES.

EXAMPLE OF USE: SEE GSSITISOL (THIS SECTION).

SUBSECTION: GSSITISOL .

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" GSSITISOL(A, N, AUX, B); "VALUE" N;
    "INTEGER" N; "ARRAY" A, AUX, B; "CODE" 34251;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N,1:N];
            ENTRY:THE N-TH ORDER MATRIX;
            EXIT: THE  CALCULATED  LOWER-TRIANGULAR  MATRIX  AND  UNIT
            UPPERTRIANGULAR  MATRIX  WITH  ITS  UNIT  DIAGONAL OMITTED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[1:13];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE FOR THE PROCESS OF TRIANGULAR
                    DECOMPOSITION; A REASONABLE CHOICE FOR THIS VALUE IS
                    AN ESTIMATE OF THE RELATIVE PRECISION OF THE MATRIX
                    ELEMENTS; HOWEVER,IT SHOULD NOT BE CHOSEN SMALLER
                    THANN THE MACHINE PRECISION;
            AUX[4]: A VALUE WHICH IS USED FOR CONTROLLING PIVOTING (SEE
                    GSSELM, SECTION 3.1.1.1.1.1.1);
            AUX[10]: A RELATIVE TOLERANCE  FOR THE SOLUTION VECTOR;  IF
                    THE  1-NORM  OF THE VECTOR  OF CORRECTIONS  TO  THE
                    SOLUTION,  DIVIDED BY  THE 1-NORM OF THE CALCULATED
                    SOLUTION, IS SMALLER THAN AUX[10], THEN THE PROCESS
                    WILL STOP; THE USER SHOULD NOT CHOOSE  THE VALUE OF
                    AUX[10] SMALLER THAN THE RELATIVE PRECISION  OF THE
                    ELEMENTS OF THE MATRIX  AND THE RIGHT-HAND SIDE  OF
                    THE LINEAR SYSTEM;
            AUX[12]: THE MAXIMUM NUMBER  OF ITERATIONS  ALLOWED FOR THE
                    REFINEMENT  OF  THE  SOLUTION;  IF  THE  NUMBER  OF
                    ITERATIONS EXCEEDS THE VALUE OF  AUX[12],  THEN THE
                    PROCESS  WILL  BE BROKEN OFF;  USUALLY  AUX[12] = 5
                    WILL GIVE GOOD RESULTS;
            EXIT:
            AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                    (SEE  AUX[3]),   THEN   AUX[1]  EQUALS   1  IF  THE
                    DETERMINANT OF THE PRINCIPAL SUBMATRIX  OF ORDER  R
                    IS POSITIVE, ELSE AUX[1] = -1;
            AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                    AUX[3] < N THEN THE PROCESS HAS BEEN BROKEN OFF AND
                    NO SOLUTION WILL HAVE BEEN CALCULATED;
            AUX[5]: THE  MODULUS  OF  AN ELEMENT  WHICH  IS  OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX WHICH HAD BEEN GIVEN
                    IN ARRAY A;
```

```
          AUX[7]: AN UPPER BOUND FOR THE GROWTH (SEE GSSELM, SECTION
                  3.1.1.1.1.1.1);
          AUX[11]: IF AUX[3] < N, THEN AUX[11] WILL BE UNDEFINED,
                  ELSE AUX[11] EQUALS THE 1-NORM OF THE VECTOR OF
                  CORRECTIONS TO THE SOLUTION IN THE LAST STEP,
                  DIVIDED BY THE 1-NORM OF THE CALCULATED SOLUTION;
                  IF AUX[11] > AUX[10], THEN THE PROCESS HAS BEEN
                  BROKEN OFF, BECAUSE THE NUMBER OF ITERATIONS
                  EXCEEDED THE VALUE GIVEN IN AUX[12];
          AUX[13]: IF AUX[3] = N, THEN THE VALUE OF AUX[13] WILL
                  EQUAL THE 1-NORM OF THE RESIDUAL VECTOR (SEE ITISOL
                  IN THIS SECTION), ELSE AUX[13] WILL BE UNDEFINED;
    B:    <ARRAY IDENTIFIER>;
          "ARRAY" B[1:N];
          ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
          EXIT:   IF AUX[3] = N, THEN THE CALCULATED SOLUTION OF THE
                  LINEAR SYSTEM IS OVERWRITTEN ON B, ELSE B REMAINS
                  UNALTERED.
```

PROCEDURES USED:

```
    DUPMAT  = CP31035,
    GSSELM  = CP34231,
    ITISOL  = CP34250.
```

REQUIRED CENTRAL MEMORY:

THREE REAL ARRAYS, TWO OF ORDER N AND ONE OF ORDER N ** 2, ARE DECLARED. FURTHERMORE, TWO INTEGER ARRAYS OF ORDER N ARE USED.

RUNNING TIME: PROPORTIONAL TO N ** 3.

METHOD AND PERFORMANCE:

GSSITISOL USES GSSELM (SECTION 3.1.1.1.1.1.1) TO PERFORM A TRIANGULAR DECOMPOSITION OF THE MATRIX AND ITISOL (THIS SECTION) TO CALCULATE AN ITERATIVELY REFINED SOLUTION OF THE GIVEN LINEAR SYSTEM; IF AUX[3] < N, THEN THE EFFECT OF GSSITISOL IS MERELY THAT OF GSSELM; IF THE CONDITION OF THE MATRIX IS VERY BAD, THEN, IN EXCEPTIONAL CASES, THE CALCULATED SOLUTION MAY BE USELESS (SEE ITISOL, IN THIS SECTION).

EXAMPLE OF USE:

LET A BE THE FOURTH ORDER SEGMENT OF THE HILBERT MATRIX, MULTIPLIED WITH 840 TO GET INTEGER ELEMENTS, AND B THE THIRD COLUMN OF A, THEN THE SOLUTION OF THE LINEAR SYSTEM AX = B IS GIVEN BY THE THIRD UNIT VECTOR AND MAY BE OBTAINED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "INTEGER" I, J;
    "ARRAY" A[1:4, 1:4], B[1:4], AUX[1:13];
    "PROCEDURE" GSSITISOL(A, N, AUX, B); "CODE" 34251;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(B[I]);
        "FOR" I:= 1 "STEP" 2 "UNTIL" 7 "DO" ITEM(AUX[I]);
        ITEM(AUX[11]); ITEM(AUX[13])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("*, "("SOLUTION:")"B+.15D"+3D,/,3(10B+.15D"+3D,/),
    "("SIGN(DET) = ")"+D,/,"("NUMBER OF ELIMINATIONSTEPS = ")"
    +D,/,"("MAX(ABS(A[I,J]))= ")"+.15D"+3D,/,
    "("UPPER BOUND GROWTH: ")"+.15D"+3D,/,
    "("NORM LAST CORRECTION VECTOR: ")"+.15D"+3D,/,
    "("NORM RESIDUAL VECTOR: ")"+.15D"+3D")");

    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        A[I,J]:= 840 // (I + J - 1); B[I]:= A[I,3]
    "END";
    AUX[2]:= "-14; AUX[4]:= 8; AUX[10]:= "-14; AUX[12]:= 5;
    GSSITISOL(A, 4, AUX, B);
    OUTLIST(71, LAYOUT, LIST)
"END"
```

RESULTS:

```
SOLUTION: +.000000000000000"+000
          +.000000000000000"+000
          +.100000000000000"+001
          +.000000000000000"+000
SIGN(DET) = +1
NUMBER OF ELIMINATIONSTEPS = +4
MAX(ABS(A[I,J]))= +.840000000000000"+003
UPPER BOUND GROWTH: +.134080000000000"+004
NORM LAST CORRECTION VECTOR: +.000000000000000"+000
NORM RESIDUAL VECTOR: +.000000000000000"+000
```

SUBSECTION: ITISOLERB.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" ITISOLERB(A, LU, N, AUX, RI, CI, B); "VALUE" N;
    "INTEGER" N; "ARRAY" A, LU, AUX, B; "INTEGER" "ARRAY" RI, CI;
    "CODE" 34253;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY"[1:N,1:N];
            ENTRY:THE  MATRIX  OF  THE  LINEAR  SYSTEM;
    LU:     <ARRAY IDENTIFIER>;
            "ARRAY" LU[1:N,1:N];
            ENTRY:THE TRIANGULARLY DECOMPOSED FORM OF THE MATRIX GIVEN
            IN A AS DELIVERED BY GSSNRI  (SECTION 3.1.1.1.1.1.1);
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[0:13];
            ENTRY:
            AUX[0]: THE MACHINE PRECISION;
            AUX[5]: THE  MODULUS  OF  AN  ELEMENT,  WHICH IS OF MAXIMUM
                    ABSOLUTE VALUE FOR THE MATRIX OF THE LINEAR SYSTEM;
                    THIS  VALUE  IS  DELIVERED   BY   GSSNRI   (SECTION
                    3.1.1.1.1.1.1) IN AUX[5];
            AUX[6]: AN  UPPER  BOUND  FOR  THE  RELATIVE  ERROR  IN THE
                    ELEMENTS OF THE MATRIX OF THE LINEAR SYSTEM;
            AUX[7]: AN  UPPER  BOUND  FOR  THE GROWTH  DURING  GAUSSIAN
                    ELIMINATION; THIS VALUE IS DELIVERED BY  GSSNRI  IN
                    AUX[7];
            AUX[8]: AN  UPPER  BOUND  FOR  THE  RELATIVE  ERROR  IN THE
                    ELEMENTS  OF  THE  RIGHT-HAND  SIDE  OF  THE LINEAR
                    SYSTEM;
            AUX[9]: THE 1-NORM  OF  THE INVERSE MATRIX;  THIS  VALUE IS
                    DELIVERED BY GSSNRI IN AUX[9];
            AUX[10]: A RELATIVE TOLERANCE  FOR THE SOLUTION VECTOR;  IF
                    THE  1-NORM  OF THE VECTOR  OF CORRECTIONS  TO  THE
                    SOLUTION,  DIVIDED BY  THE 1-NORM OF THE CALCULATED
                    SOLUTION, IS SMALLER THAN AUX[10], THEN THE PROCESS
                    WILL STOP; THE USER SHOULD NOT CHOOSE  THE VALUE OF
                    AUX[10] SMALLER THAN THE RELATIVE PRECISION  OF THE
                    ELEMENTS OF THE MATRIX  AND THE RIGHT-HAND SIDE  OF
                    THE LINEAR SYSTEM, GIVEN IN AUX[6] AND AUX[8];
            AUX[12]: THE MAXIMUM NUMBER  OF ITERATIONS  ALLOWED FOR THE
                    REFINEMENT  OF  THE  SOLUTION;  IF  THE  NUMBER  OF
                    ITERATIONS EXCEEDS THE VALUE OF  AUX[12],  THEN THE
                    PROCESS  WILL  BE BROKEN OFF;  USUALLY  AUX[12] = 5
                    WILL GIVE GOOD RESULTS;
```

```
          EXIT:
          AUX[11]:A REALISTIC UPPERBOUND FOR THE RELATIVE ERROR IN
                  THE CALCULATED SOLUTION; HOWEVER, IF NO USE CAN BE
                  MADE OF THE ERROR-FORMULA, THEN AUX[11] := -1;
          AUX[13]:THE 1-NORM OF THE RESIDUAL VECTOR  (SEE METHOD AND
                  PERFORMANCE);
    RI:   <ARRAY IDENTIFIER>;
          "INTEGER""ARRAY" RI[1:N];
          ENTRY:THE PIVOTAL ROW-INDICES,  AS PRODUCED BY  GSSNRI;
    CI:   <ARRAY IDENTIFIER>;
          "INTEGER""ARRAY" CI[1:N];
          EXIT:THE PIVOTAL COLUMN-INDICES, AS PRODUCED BY  GSSNRI;
    B:    <ARRAY IDENTIFIER>;
          "ARRAY" B[1:N];
          ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
          EXIT:   THE SOLUTION OF THE LINEAR SYSTEM.
```

PROCEDURES USED:

ITISOL = CP34250.

REQUIRED CENTRAL MEMORY:TWO REAL AARRYS, BOTH OF ORDER N, ARE DECLARED.

RUNNING TIME: THE NUMBER OF ARITHMETICAL OPERATIONS IN EACH ITERATION STEP OF THE REFINEMENT PROCESS IS PROPORTIONAL TO N ** 2.

METHOD AND PERFORMANCE:

ITISOLERB SHOULD BE CALLED AFTER GSSNRI (SECTION 3.1.1.1.1.1.1), WHICH DELIVERS THE TRIANGULARLY DECOMPOSED FORM OF THE MATRIX AND THE PROPER VALUES FOR THE ODD ELEMENTS OF ARRAY AUX; ITISOLERB CALCULATES, WITH THE USE OF ITISOL (THIS SECTION), AN ITERATIVELY IMPROVED SOLUTION OF THE LINEAR SYSTEM WITH A MATRIX AS GIVEN IN ARRAY A AND A RIGHT-HAND SIDE AS GIVEN IN B; MOREOVER, ITISOLERB CALCULATES A REALISTIC UPPER BOUND FOR THE RELATIVE ERROR IN THE CALCULATED SOLUTION, BY (SEE [1], [2]):

```
     NORM(DX) / NORM(X) <= P / (1 - P),
WHERE : P = ( NORM(R) / NORM(X) + DB / NORM(X) + DA )
     * NORM(C) / (1 - Q * NORM(C) )
     FOR Q SEE SECTION 3.1.1.1.1.1.1 (SUBSECTION ERBELM),
     R IS THE RESIDUAL VECTOR (SEE ITISOL IN THIS SECTION),
     X IS THE CALCULATED SOLUTION,
     DB IS THE UPPER BOUND FOR THE RELATIVE ERROR  IN THE RIGHT-HAND
     SIDE ELEMENTS,
     DA IS THE UPPER BOUND FOR THE RELATIVE ERROR IN THE MATRIX
     ELEMENTS. C IS THE CALCULATED INVERSE MATRIX,
     AND THE 1-NORM OF A VECTOR OR A MATRIX IS DENOTED BY: NORM(.)
```

IF 1 - P < AUX[0], THEN THE VALUE -1 IS DELIVERED IN AUX[11]; ITISOLERB LEAVES A, LU, RI AND CI UNALTERED, SO AFTER ONE CALL OF GSSNRI SEVERAL CALLS OF ITISOLERB MAY FOLLOW, TO CALCULATE THE SOLUTION OF SEVERAL LINEAR SYSTEMS WITH THE SAME MATRIX BUT DIFFERENT RIGHT-HAND SIDES.

EXAMPLE OF USE: SEE GSSITISOLERB (THIS SECTION).

SUBSECTION: GSSITISOLERB.

CALLING SEQUENCE:

THE HEADING OF THIS PROCEDURE IS:

```
"PROCEDURE" GSSITISOLERB(A, N, AUX, B); "VALUE" N;
"INTEGER" N; "ARRAY" A, AUX, B; "CODE" 34254;
```

THE MEANING OF THE FORMAL PARAMETERS IS:

```
A:      <ARRAY IDENTIFIER>;
        "ARRAY" A[1:N,1:N];
        ENTRY:THE N-TH ORDER MATRIX;
        EXIT: THE  CALCULATED  LOWER-TRIANGULAR  MATRIX  AND  UNIT
        UPPER-TRIANGULAR  MATRIX WITH ITS  UNIT  DIAGONAL  OMITTED;
N:      <ARITHMETIC EXPRESSION>;
        THE ORDER OF THE MATRIX;
AUX:    <ARRAY IDENTIFIER>;
        "ARRAY" AUX[0:13];
        ENTRY:
        AUX[0]: THE MACHINE PRECISION;
        AUX[2]: A RELATIVE TOLERANCE:  A REASONABLE CHOICE FOR THIS
                VALUE  IS  AN ESTIMATE OF THE RELATIVE PRECISION OF
                THE MATRIX ELEMENTS; HOWEVER, IT SHOULD  NOT  BE
                CHOSEN SMALLER THAN THE MACHINE PRECISION;
        AUX[4]: A VALUE WHICH IS USED FOR CONTROLLING PIVOTING (SEE
                GSSELM, SECTION 3.1.1.1.1.1.1);
        AUX[6]: AN UPPER BOUND FOR THE RELATIVE ERROR IN THE MATRIX
                ELEMENTS OF THE LINEAR SYSTEM;
        AUX[8]: AN  UPPER  BOUND  FOR  THE  RELATIVE  ERROR  IN THE
                ELEMENTS OF THE RIGHT-HAND SIDE;
        AUX[10]: A RELATIVE TOLERANCE  FOR THE SOLUTION VECTOR;  IF
                THE  1-NORM  OF THE VECTOR  OF CORRECTIONS  TO  THE
                SOLUTION,  DIVIDED BY  THE 1-NORM OF THE CALCULATED
                SOLUTION, IS SMALLER THAN AUX[10], THEN THE PROCESS
                WILL STOP; THE USER SHOULD NOT CHOOSE  THE VALUE OF
                AUX[10] SMALLER THAN THE RELATIVE PRECISION  OF THE
                ELEMENTS OF THE MATRIX  AND THE RIGHT-HAND SIDE  OF
                THE LINEAR SYSTEM (AUX[10] >= AUX[2]);
        AUX[12]: THE MAXIMUM NUMBER  OF ITERATIONS  ALLOWED FOR THE
                REFINEMENT  OF  THE  SOLUTION;  IF  THE  NUMBER  OF
                ITERATIONS EXCEEDS THE VALUE OF  AUX[12],  THEN THE
                PROCESS  WILL  BE BROKEN OFF;  USUALLY  AUX[12] = 5
                WILL GIVE GOOD RESULTS;
```

```
          EXIT:
          AUX[1]: IF  R  IS THE NUMBER OF ELIMINATION STEPS PERFORMED
                  (SEE  AUX[3]),   THEN   AUX[1]  EQUALS  1  IF  THE
                  DETERMINANT OF THE PRINCIPAL SUBMATRIX  OF ORDER  R
                  IS POSITIVE, ELSE AUX[1] = -1;
          AUX[3]: THE  NUMBER  OF  ELIMINATION  STEPS  PERFORMED;  IF
                  AUX[3] < N  THEN  THE PROCESS  HAS BEEN BROKEN OFF
                  AND NO SOLUTION WILL HAVE BEEN CALCULATED;
          AUX[5]: THE  MODULUS  OF  AN ELEMENT  WHICH  IS  OF MAXIMUM
                  ABSOLUTE VALUE FOR THE MATRIX WHICH HAD BEEN GIVEN
                  IN ARRAY A;
          AUX[7]: AN UPPER BOUND FOR THE GROWTH (SEE GSSELM,  SECTION
                  3.1.1.1.1.1.1);
          AUX[9]: IF AUX[3] = N THEN AUX[9] EQUALS THE 1-NORM  OF THE
                  CALCULATED  INVERSE  MATRIX,  ELSE  AUX[9]  WILL BE
                  UNDEFINED;
          AUX[11]: IF AUX[3] < N,  THEN  AUX[11]  WILL  BE UNDEFINED,
                  ELSE THE VALUE OF AUX[11]  EQUALS A REALISTIC UPPER
                  BOUND  FOR  THE RELATIVE ERROR  IN  THE  CALCULATED
                  SOLUTION; HOWEVER,  IF NO USE  CAN BE MADE  OF THE
                  ERROR FORMULA (SEE ITISOLERB IN THIS SECTION), THEN
                  AUX[11]:= -1;
          AUX[13]: IF AUX[3] = N,  THEN  AUX[13] EQUALS THE 1-NORM OF
                  THE RESIDUAL VECTOR  (SEE  ITISOL IN THIS SECTION),
                  ELSE AUX[13] WILL BE UNDEFINED;
    B:    <ARRAY IDENTIFIER>;
          "ARRAY" B[1:N];
          ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
          EXIT:   IF AUX[3] = N,  THEN THE CALCULATED SOLUTION OF THE
                  LINEAR SYSTEM IS OVERWRITTEN ON B,  ELSE  B REMAINS
                  UNALTERED.
```

PROCEDURES USED:

```
    DUPMAT       = CP31035,
    GSSNRI       = CP34252,
    ITISOLERB    = CP34253.
```

REQUIRED CENTRAL MEMORY:

THREE REAL ARRAYS, TWO OF ORDER N AND ONE OF ORDER N ** 2 , ARE USED. FURTHERMORE, TWO INTEGER ARRAYS OF ORDER N ARE USED.

RUNNING TIME: PROPORTIONAL TO N ** 3.

METHOD AND PERFORMANCE:

GSSITISOLERB USES GSSNRI (SECTION 3.1.1.1.1.1.1) TO PERFORM A TRIANGULAR DECOMPOSITION OF THE MATRIX AND ITISOLERB (THIS SECTION) TO CALCULATE AN ITERATIVELY REFINED SOLUTION OF THE GIVEN LINEAR SYSTEM AND A REALISTIC UPPER BOUND FOR THE RELATIVE ERROR IN THIS SOLUTION; IF AUX[3] < N, THEN THE EFFECT OF GSSITISOLERB IS MERELY THAT OF GSSELM (SECTION 3.1.1.1.1.1.1).

EXAMPLE OF USE:

LET A BE THE FOURTH ORDER SEGMENT OF THE HILBERT MATRIX, MULTIPLIED WITH 840 TO GET INTEGER ELEMENTS, AND B THE THIRD COLUMN OF A, THEN THE SOLUTION OF THE LINEAR SYSTEM AX = B IS GIVEN BY THE THIRD UNIT VECTOR AND THIS SOLUTION, AS WELL AS A REALISTIC UPPER BOUND FOR THE RELATIVE ERROR IN IT, MAY BE OBTAINED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "INTEGER" I, J;
    "ARRAY" A[1:4, 1:4], B[1:4], AUX[0:13];
    "PROCEDURE" GSSITISOLERB(A, N, AUX, B); "CODE" 34254;
    "PROCEDURE" LIST(ITEM); "PROCEDURE" ITEM;
    "BEGIN" "INTEGER" I;
        "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO" ITEM(B[I]);
        "FOR" I:= 1 "STEP" 2 "UNTIL" 13 "DO" ITEM(AUX[I])
    "END" LIST;
    "PROCEDURE" LAYOUT;
    FORMAT("("*, "("SOLUTION:")"B+.15D"+3D,/,3(10B+.15D"+3D,/),
    "("SIGN(DET) = ")"+D,/,"("NUMBER OF ELIMINATIONSTEPS = ")"
    +D,/,"("MAX(ABS(A[I,J]))= ")"+.15D"+3D,/,
    "("UPPER BOUND GROWTH: ")"+.15D"+3D,/,
    "("NORM CALCULATED INVERSE MATRIX: ")"+.15D"+3D,/,
    "("UPPER BOUND FOR THE RELATIVE ERROR: ")"+.15D"+3D,/,
    "("NORM RESIDUAL VECTOR: ")"+.15D"+3D")");

    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        A[I,J]:= 840 // (I + J - 1); B[I]:= A[I,3]
    "END";
    AUX[0]:= AUX[2]:= "-14; AUX[4]:= 8; AUX[6]:= AUX[8]:= 0;
    AUX[10]:= "-14; AUX[12]:= 5;
    GSSITISOLERB(A, 4, AUX, B);
    OUTLIST(71, LAYOUT, LIST)
"END"
```

RESULTS:

```
SOLUTION: +.000000000000000"+000
          +.000000000000000"+000
          +.100000000000000"+001
          +.000000000000000"+000
SIGN(DET) = +1
NUMBER OF ELIMINATIONSTEPS = +4
MAX(ABS(A[I,J]))= +.840000000000000"+003
UPPER BOUND GROWTH: +.134080000000000"+004
NORM CALCULATED INVERSE MATRIX: +.162142857143540"+002
UPPER BOUND FOR THE RELATIVE ERROR: +.000000000000000"+000
NORM RESIDUAL VECTOR: +.000000000000000"+000
```

REFERENCES:

[1] BUS, J. C. P.
LINEAR SYSTEMS WITH CALCULATION OF ERROR BOUNDS AND ITERATIVE REFINEMENT (DUTCH).
MATHEMATICAL CENTRE, AMSTERDAM, LR 3.4.19, (1972).

[2] DEKKER, T. J.
NUMERICAL ALGEBRA (DUTCH).
MATHEMATICAL CENTRE, AMSTERDAM, SYLLABUS 12, (1971).

[3] WILKINSON, J. H.
THE ALGEBRAIC EIGENVALUE PROBLEM.
OXFORD (1965).

SOURCE TEXT(S):

```
"CODE" 34250;
    "PROCEDURE" ITISOL(A, LU, N, AUX, RI, CI, B); "VALUE" N;
    "INTEGER" N;
    "ARRAY" A, LU, AUX, B; "INTEGER" "ARRAY" RI, CI;
    "BEGIN" "INTEGER" I, ITER, MAXITER;
        "REAL" MAXERX, ERX, NRMRES, NRMSOL, R, RR;
        "ARRAY" RES, SOL[1:N];
        "PROCEDURE" SOLELM(A, N, RI, CI, B); "CODE" 34061;
        "PROCEDURE" INIVEC(L, U, A, X); "CODE" 31010;
        "PROCEDURE" DUPVEC(L, U, SHIFT, A, B); "CODE" 31030;
        "PROCEDURE" LNGMATVEC(A, B, C, D, E, F, G, H, I);
        "CODE" 34411;
        MAXERX:= ERX:= AUX[10]; MAXITER:= AUX[12];
        INIVEC(1, N, SOL, 0); DUPVEC(1, N, 0, RES, B);
        "FOR" ITER:= 1, ITER + 1 "WHILE" ITER <= MAXITER &
        MAXERX < ERX "DO"
        "BEGIN" SOLELM(LU, N, RI, CI, RES); ERX:= NRMSOL:= NRMRES:= 0;
            "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
            "BEGIN" R:= RES[I]; ERX:= ERX + ABS(R); RR:= SOL[I] + R;
                SOL[I]:= RR; NRMSOL:= NRMSOL + ABS(RR)
            "END";
            ERX:= ERX / NRMSOL;
            "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
            "BEGIN" LNGMATVEC(1, N, I, A, SOL, - B[I], 0, R, RR);
                R:= - (R + RR); RES[I]:= R; NRMRES:= NRMRES + ABS(R)
            "END"
        "END" ITERATION;
        DUPVEC(1, N, 0, B, SOL);
        AUX[11]:= ERX; AUX[13]:= NRMRES
    "END" ITISOL;
        "EOP"
```

```
"CODE" 34251;
    "PROCEDURE" GSSITISOL(A, N, AUX, B); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX, B;
    "BEGIN" "INTEGER" I, J;
        "ARRAY" AA[1:N,1:N];
        "INTEGER" "ARRAY" RI, CI[1:N];
        "PROCEDURE" GSSELM(A, N, AUX, RI, CI); "CODE" 34231;
        "PROCEDURE" ITISOL(A, LU, N, AUX, RI, CI, B); "CODE" 34250;
        "PROCEDURE" DUPMAT(L, U, I, J, A, B); "CODE" 31035;
        DUPMAT(1, N, 1, N, AA, A);
        GSSELM(A, N, AUX, RI, CI);
        "IF" AUX[3] = N "THEN" ITISOL(AA, A, N, AUX, RI, CI, B)
    "END" GSSITISOL;
        "EOP"

"CODE" 34253;
    "PROCEDURE" ITISOLERB(A, LU, N, AUX, RI, CI, B); "VALUE" N;
    "INTEGER" N;
    "ARRAY" A, LU, AUX, B; "INTEGER" "ARRAY" RI, CI;
    "BEGIN" "INTEGER" I;
        "REAL" NRMSOL, NRMINV, NRMB, ALFA, TOLA, EPS;
        "PROCEDURE" ITISOL(A, LU, N, AUX, RI, CI, B); "CODE" 34250;
        EPS:= AUX[0];
        NRMINV:= AUX[9]; TOLA:= AUX[5] * AUX[6]; NRMB:= NRMSOL:= 0;
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO" NRMB:= NRMB + ABS(B[I]);
        ITISOL(A, LU, N, AUX, RI, CI, B);
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
        NRMSOL:= NRMSOL + ABS(B[I]);
        ALFA:= 1 - (1.06 * EPS * AUX[7] * (.75 * N + 4.5) * N ** 2
                + TOLA) * NRMINV;
        "IF" ALFA < EPS "THEN" AUX[11]:= - 1 "ELSE"
        "BEGIN" ALFA:= ((AUX[13] + AUX[8] * NRMB) / NRMSOL + TOLA) *
                       NRMINV / ALFA;
            AUX[11]:= "IF" 1 - ALFA < EPS "THEN" - 1 "ELSE"
                      ALFA / (1 - ALFA)
        "END"
    "END" ITISOLERB;
        "EOP"

"CODE" 34254;
    "PROCEDURE" GSSITISOLERB(A, N, AUX, B); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX, B;
    "BEGIN" "INTEGER" I, J;
        "ARRAY" AA[1:N,1:N];
        "INTEGER" "ARRAY" RI, CI[1:N];
        "PROCEDURE" GSSNRI(A, N, AUX, RI, CI); "CODE" 34252;
        "PROCEDURE" ITISOLERB(A, LU, N, AUX, RI, CI, B); "CODE" 34253;
        "PROCEDURE" DUPMAT(L, U, I, J, A, B); "CODE" 31035;
        DUPMAT(1, N, 1, N, AA, A);
        GSSNRI(A, N, AUX, RI, CI);
        "IF" AUX[3] = N "THEN" ITISOLERB(AA, A, N, AUX, RI, CI, B)
    "END" GSSITISOLERB;
        "EOP"
```

AUTHOR: T.J. DEKKER.

CONTRIBUTORS: S.P.N. VAN KAMPEN, J. KOK.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 731015.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS TWO PROCEDURES:
A) CHLDEC2 CALCULATES THE CHOLESKY DECOMPOSITION OF A POSITIVE DEFINITE SYMMETRIC MATRIX WHOSE UPPER TRIANGLE IS GIVEN IN A TWO-DIMENSIONAL ARRAY;
B) CHLDEC1 CALCULATES THE CHOLESKY DECOMPOSITION OF A POSITIVE DEFINITE SYMMETRIC MATRIX WHOSE UPPER TRIANGLE IS GIVEN COLUMNWISE IN A ONE-DIMENSIONAL ARRAY.

KEYWORDS:

LINEAR EQUATIONS,
POSITIVE DEFINITE SYMMETRIC MATRIX,
CHOLESKY DECOMPOSITION.

SUBSECTION: CHLDEC2.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS:
"PROCEDURE" CHLDEC2(A, N, AUX); "VALUE" N; "INTEGER" N;
"ARRAY" A, AUX;

THE MEANING OF THE FORMAL PARAMETERS IS:

A: <ARRAY IDENTIFIER>;
"ARRAY" A[1:N,1:N];
ENTRY: THE UPPER TRIANGLE OF THE POSITIVE DEFINITE MATRIX MUST BE GIVEN IN THE UPPER-TRIANGULAR PART OF A (THE ELEMENTS A[I,J], I <= J);
EXIT: THE CHOLESKY DECOMPOSITION OF THE MATRIX IS DELIVERED IN THE UPPER TRIANGLE OF A;

N: <ARITHMETIC EXPRESSION>;
THE ORDER OF THE MATRIX;

AUX: <ARRAY IDENTIFIER>;
"ARRAY" AUX[2:3];
ENTRY: AUX[2]: A RELATIVE TOLERANCE USED TO CONTROL THE CALCULATION OF THE DIAGONAL ELEMENTS;
NORMAL EXIT: AUX[3]:= N;
ABNORMAL EXIT: IF THE DECOMPOSITION CANNOT BE CARRIED OUT BECAUSE THE MATRIX IS (NUMERICALLY) NOT POSITIVE DEFINITE, AUX[3]:= K - 1, WHERE K IS THE LAST STAGE NUMBER.

PROCEDURES USED: TAMMAT = CP34014.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

CHLDEC2 PERFORMS THE CHOLESKY DECOMPOSITION OF A SYMMETRIC POSITIVE DEFINITE MATRIX.
THE METHOD USED IS CHOLESKY'S SQUARE ROOT METHOD WITHOUT PIVOTING (SEE REF[1] AND [2]). IF THE GIVEN SYMMETRIC MATRIX IS POSITIVE DEFINITE, THE METHOD YIELDS AN UPPER-TRIANGULAR MATRIX U SUCH THAT U'U EQUALS THE GIVEN MATRIX.
THE PROCESS IS TERMINATED AT STAGE K, IF THE K-TH DIAGONAL ELEMENT OF THE GIVEN MATRIX MINUS THE SUM OF THE SQUARED ELEMENTS OF THE K-TH COLUMN OF U IS LESS THAN A TOLERANCE TIMES THE MAXIMUM DIAGONAL ELEMENT OF THE GIVEN MATRIX. IN THIS CASE THE MATRIX, POSSIBLY MODIFIED BY ROUNDING ERRORS, IS NOT POSITIVE DEFINITE.

REFERENCES:

[1]. T.J. DEKKER.
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1.
MC TRACT 22, 1968, MATH. CENTR., AMSTERDAM.

[2]. J.H. WILKINSON.
THE ALGEBRAIC EIGENVALUE PROBLEM.
CLARENDON PRESS, OXFORD, 1965.

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLINV2, SECTION 3.1.1.1.1.2.4.

SUBSECTION: CHLDEC1.

CALLING SEQUENCE:

```
    THE HEADING OF THE PROCEDURE IS:
    "PROCEDURE" CHLDEC1(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1 : (N + 1) * N // 2];
            ENTRY: THE UPPER-TRIANGULAR PART OF THE  POSITIVE  DEFINITE
                   SYMMETRIC MATRIX MUST BE GIVEN COLUMNWISE IN ARRAY A
                   (THE (I,J)-TH ELEMENT OF THE MATRIX MUST BE GIVEN IN
                   A[(J - 1) * J // 2 + I] FOR 1 <= I <= J <= N);
            EXIT:  THE  CHOLESKY  DECOMPOSITION  OF  THE   MATRIX  IS
                   DELIVERED COLUMNWISE IN A.
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[2:3];
            ENTRY: AUX[2]: A  RELATIVE TOLERANCE  USED TO  CONTROL  THE
                   CALCULATION OF THE DIAGONAL ELEMENTS;
            NORMAL EXIT:   AUX[3]:= N;
            ABNORMAL EXIT: IF THE  DECOMPOSITION CANNOT BE  CARRIED OUT
                   BECAUSE  THE  MATRIX  IS (NUMERICALLY) NOT  POSITIVE
                   DEFINITE,  AUX[3]:= K - 1, WHERE K IS THE LAST STAGE
                   NUMBER.
```

PROCEDURES USED: VECVEC = CP34010.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

CHLDEC1 PERFORMS THE CHOLESKY DECOMPOSITION OF A SYMMETRIC POSITIVE DEFINITE MATRIX, WHOSE UPPER TRIANGLE IS STORED IN A ONE-DIMENSIONAL ARRAY, BY CHOLESKY'S SQUARE ROOT METHOD WITHOUT PIVOTING.
SEE ALSO METHOD AND PERFORMANCE OF CHLDEC2, (THIS SECTION).

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLINV1, SECTION 3.1.1.1.1.2.4.

SOURCE TEXT(S) :

```
"CODE" 34310;
    "PROCEDURE" CHLDEC2(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "BEGIN" "INTEGER" K, J; "REAL" R, EPSNORM;
        "REAL" "PROCEDURE" TAMMAT(L, U, I, J, A, B); "CODE" 34014;

        R:= 0;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "IF" A[K,K] > R "THEN" R:= A[K,K];
        EPSNORM:= AUX[2] * R;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" R:= A[K,K] - TAMMAT(1, K - 1, K, K, A, A);
            "IF" R <= EPSNORM "THEN"
            "BEGIN" AUX[3]:= K - 1; "GOTO" END "END";
            A[K,K]:= R:= SQRT(R);
            "FOR" J:= K + 1 "STEP" 1 "UNTIL" N "DO"
            A[K,J]:= (A[K,J] - TAMMAT(1, K - 1, J, K, A, A)) / R
        "END";
        AUX[3]:= N;
    END:
    "END" CHLDEC2;
        "EOP"

"CODE" 34311;
    "PROCEDURE" CHLDEC1(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "BEGIN" "INTEGER" J, K, KK, KJ, LOW, UP; "REAL" R, EPSNORM;
        "REAL" "PROCEDURE" VECVEC(L, U, SHIFT, A, B); "CODE" 34010;

        R:= 0; KK:= 0;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" KK:= KK + K; "IF" A[KK] > R "THEN" R:= A[KK] "END";
        EPSNORM:= AUX[2] * R; KK:= 0;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" KK:= KK + K; LOW:= KK - K + 1; UP:= KK - 1;
            R:= A[KK] - VECVEC(LOW, UP, 0, A, A);
            "IF" R <= EPSNORM "THEN"
            "BEGIN" AUX[3]:= K - 1; "GOTO" END "END";
            A[KK]:= R:= SQRT(R); KJ:= KK + K;
            "FOR" J:= K + 1 "STEP" 1 "UNTIL" N "DO"
            "BEGIN" A[KJ]:= (A[KJ] -
                    VECVEC(LOW, UP, KJ - KK, A, A)) / R;
                KJ:= KJ + J
            "END"
        "END";
        AUX[3]:= N;
    END:
    "END" CHLDEC1;
        "EOP"
```

CONTRIBUTORS: S.P.N. VAN KAMPEN, J. KOK.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 731015.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS TWO PROCEDURES:
CHLDETERM2, FOR THE CALCULATION OF THE DETERMINANT OF A SYMMETRIC POSITIVE DEFINITE MATRIX WHOSE CHOLESKY MATRIX IS GIVEN IN THE UPPER TRIANGLE OF A TWO-DIMENSIONAL ARRAY;
CHLDETERM1, FOR THE CALCULATION OF THE DETERMINANT OF A SYMMETRIC POSITIVE DEFINITE MATRIX WHOSE CHOLESKY MATRIX IS GIVEN COLUMNWISE IN A ONE-DIMENSIONAL ARRAY.

KEYWORDS:

DETERMINANT,
POSITIVE DEFINITE SYMMETRIC MATRIX,
CHOLESKY DECOMPOSITION.

SUBSECTION: CHLDETERM2.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS:

```
"REAL" "PROCEDURE" CHLDETERM2(A, N); "VALUE" N; "INTEGER" N;
"ARRAY" A; "CODE" 34312;
```

CHLDETERM2 := THE DETERMINANT OF THE SYMMETRIC POSITIVE DEFINITE MATRIX OF WHICH THE CHOLESKY MATRIX IS STORED IN A;

THE MEANING OF THE FORMAL PARAMETERS IS:

```
A:      <ARRAY IDENTIFIER>;
        "ARRAY" A[1:N,1:N];
        ENTRY: THE  UPPER-TRIANGULAR PART OF THE  CHOLESKY MATRIX
               AS  PRODUCED BY CHLDEC2 (SECTION  3.1.1.1.1.2.1) OR
               CHLDECSOL2 (SECTION  3.1.1.1.1.2.3)  MUST  BE GIVEN
               IN THE UPPER TRIANGLE OF A;
        EXIT:  THE CONTENTS OF A ARE NOT CHANGED;
N:      <ARITHMETIC EXPRESSION>;
        THE ORDER OF THE MATRIX.
```

PROCEDURES USED: NONE.

REQUIRED CENTRAL MEMORY: NO EXTRA ARRAYS ARE DECLARED.

RUNNING TIME: PROPORTIONAL TO N.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLDETERM2 SHOULD BE CALLED AFTER A SUCCESSFUL CALL
OF CHLDEC2 OR CHLDECSOL2, I.E. IF AUX[3] = N;
CHLDETERM2 SHOULD NOT BE CALLED IF OVERFLOW IS TO BE EXPECTED.

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLDECINV2 (SECTION 3.1.1.1.1.2.4)

SUBSECTTON: CHLDETERM1.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS:
"REAL" "PROCEDURE" CHLDETERM1(A, N); "VALUE" N; "INTEGER" N;
"ARRAY" A; "CODE" 34313;

CHLDETERM1 := THE DETERMINANT OF THE SYMMETRIC POSITIVE
DEFINITE MATRIX OF WHICH THE CHOLESKY MATRIX IS STORED IN A;

THE MEANING OF THE FORMAL PARAMETERS IS:
A: <ARRAY IDENTIFIER>;
"ARRAY" A[1 : (N + 1) * N // 2];
ENTRY: THE UPPER-TRIANGULAR PART OF THE CHOLESKY MATRIX
AS PRODUCED BY CHLDEC1 (SECTION 3.1.1.1.1.2.1) OR
CHLDECSOL1 (SECTION 3.1.1.1.1.2.3) MUST BE GIVEN
COLUMNWISE IN ARRAY A;
EXIT: THE CONTENTS OF A ARE NOT CHANGED;
N: <ARITHMETIC EXPRESSION>;
THE ORDER OF THE MATRIX.

PROCEDURES USED: NONE.

REQUIRED CENTRAL MEMORY: NO EXTRA ARRAYS ARE DECLARED.

RUNNING TIME: PROPORTIONAL TO N.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLDETERM1 SHOULD BE CALLED AFTER A SUCCESSFUL CALL OF CHLDEC1 OR CHLDECSOL1, I.E. IF AUX[3] = N; CHLDETERM1 SHOULD NOT BE CALLED IF OVERFLOW IS TO BE EXPECTED.

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLDECINV1, SECTION 3.1.1.1.1.2.4.

SOURCE TEXT(S) :

```
"CODE" 34312;
    "REAL" "PROCEDURE" CHLDETERM2(A, N); "VALUE" N; "INTEGER" N;
    "ARRAY" A;
    "BEGIN" "INTEGER" K; "REAL" D;
        D:= 1;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO" D:= A[K,K] * D;
        CHLDETERM2:= D * D
    "END" CHLDETERM2;
        "EOP"

"CODE" 34313;
    "REAL" "PROCEDURE" CHLDETERM1(A, N); "VALUE" N; "INTEGER" N;
    "ARRAY" A;
    "BEGIN" "INTEGER" K, KK; "REAL" D;
        D:= 1; KK:= 0;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" KK:= KK + K; D:= A[KK] * D "END";
        CHLDETERM1:= D * D
    "END" CHLDETERM1;
        "EOP"
```

AUTHOR: T.J. DEKKER.

CONTRIBUTORS: S.P.N. VAN KAMPEN, J. KOK.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 731015.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS FOUR PROCEDURES:
A) CHLSOL2, FOR THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS IF THE COEFFICIENT MATRIX HAS BEEN DECOMPOSED BY A CALL OF THE PROCEDURE CHLDEC2, SECTION 3.1.1.1.1.2.1., OR CHLDECSOL2;
B) CHLSOL1, FOR THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS IF THE COEFFICIENT MATRIX HAS BEEN DECOMPOSED BY A CALL OF THE PROCEDURE CHLDEC1, SECTION 3.1.1.1.1.2.1., OR CHLDECSOL1;
C) CHLDECSOL2, FOR THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS BY CHOLESKY'S SQUARE ROOT METHOD;
THE COEFFICIENT MATRIX HAS TO BE SYMMETRIC POSITIVE DEFINITE AND MUST BE GIVEN IN THE UPPER TRIANGLE OF A TWO-DIMENSIONAL ARRAY;
D) CHLDECSOL1, FOR THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS BY CHOLESKY'S SQUARE ROOT METHOD;
THE COEFFICIENT MATRIX HAS TO BE SYMMETRIC POSITIVE DEFINITE AND MUST BE GIVEN COLUMNWISE IN A ONE-DIMENSIONAL ARRAY.

KEYWORDS:

LINEAR EQUATIONS,
POSITIVE DEFINITE SYMMETRIC MATRIX,
CHOLESKY DECOMPOSITION.

SUBSECTION: CHLSOL2.

CALLING SEQUENCE:

```
    THE HEADING OF THE PROCEDURE IS:
    "PROCEDURE" CHLSOL2(A, N, B); "VALUE" N; "INTEGER" N; "ARRAY" A, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N,1:N];
            ENTRY: THE  UPPER-TRIANGULAR PART OF THE  CHOLESKY MATRIX
                   AS  PRODUCED BY CHLDEC2, SECTION  3.1.1.1.1.2.1., OR
                   CHLDECSOL2  (THIS  SECTION), MUST  BE  GIVEN IN  THE
                   UPPER TRIANGLE OF A;
            EXIT:  THE CONTENTS OF A ARE NOT CHANGED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY: THE  RIGHT HAND  SIDE  OF  THE  SYSTEM  OF  LINEAR
                   EQUATIONS;
            EXIT:  THE SOLUTION OF THE SYSTEM.
```

PROCEDURES USED:

MATVEC = CP34011,
TAMVEC = CP34012.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N SQUARED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLSOL2 CALCULATES THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS, PROVIDED THAT THE COEFFICIENT MATRIX HAS BEEN DECOMPOSED BY A SUCCESSFUL CALL OF CHLDEC2 OR CHLDECSOL2;
THE SOLUTION IS OBTAINED BY CARRYING OUT THE FORWARD AND BACK SUBSTITUTION WITH THE CHOLESKY MATRIX AND THE RIGHT HAND SIDE.
THE RIGHT HAND SIDE IS OVERWRITTEN BY THE SOLUTION BUT THE ELEMENTS OF THE CHOLESKY MATRIX ARE NOT CHANGED, THUS SEVERAL SYSTEMS OF LINEAR EQUATIONS WITH THE SAME COEFFICIENT MATRIX BUT DIFFERENT RIGHT HAND SIDES CAN BE SOLVED BY SUCCESSIVE CALLS OF CHLSOL2. SEE ALSO REF [1].

REFERENCES:

[1]. T.J. DEKKER.
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1.
MC TRACT 22, 1968, MATH. CENTR., AMSTERDAM.

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLINV2, SECTION 3.1.1.1.1.2.4.

SUBSECTION: CHLSOL1.

CALLING SEQUENCE:

```
    THE HEADING OF THE PROCEDURE IS:
    "PROCEDURE" CHLSOL1(A, N, B); "VALUE" N; "INTEGER" N; "ARRAY" A, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1 : (N + 1) * N // 2];
            ENTRY: THE  UPPER-TRIANGULAR PART OF THE  CHOLESKY MATRIX
                   AS  PRODUCED BY CHLDEC1, SECTION  3.1.1.1.1.2.1., OR
                   CHLDECSOL1 (THIS SECTION), MUST  BE GIVEN COLUMNWISE
                   IN ARRAY A;
            EXIT:  THE CONTENTS OF A ARE NOT CHANGED;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY: THE  RIGHT HAND   SIDE  OF  THE   SYSTEM  OF  LINEAR
                   EQUATIONS;
            EXIT:  THE SOLUTION OF THE SYSTEM.
```

PROCEDURES USED:

    VECVEC = CP34010,
    SEQVEC = CP34016.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N SQUARED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLSOL1 CALCULATES THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS, PROVIDED THAT THE COEFFICIENT MATRIX HAS BEEN DECOMPOSED BY A SUCCESSFUL CALL OF CHLDEC1 OR CHLDECSOL1;
SEVERAL SYSTEMS WITH THE SAME COEFFICIENT MATRIX BUT DIFFERENT RIGHT HAND SIDES CAN BE SOLVED BY SUCCESSIVE CALLS OF CHLSOL1.
SEE ALSO METHOD AND PERFORMANCE OF CHLSOL2 (THIS SECTION).

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLINV1, SECTION 3.1.1.1.1.2.4.

SUBSECTION: CHLDECSOL2.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS:

```
"PROCEDURE" CHLDECSOL2(A, N, AUX, B); "VALUE" N; "INTEGER" N;
"ARRAY" A, AUX, B;
```

THE MEANING OF THE FORMAL PARAMETERS IS:

A: <ARRAY IDENTIFIER>;
"ARRAY" A[1:N,1:N];
ENTRY: THE UPPER TRIANGLE OF THE POSITIVE DEFINITE MATRIX MUST BE GIVEN IN THE UPPER-TRIANGULAR PART OF A (THE ELEMENTS A[I,J], I <= J);
EXIT: THE CHOLESKY DECOMPOSITION OF THE MATRIX IS DELIVERED IN THE UPPER TRIANGLE OF A;

N: <ARITHMETIC EXPRESSION>;
THE ORDER OF THE MATRIX;

AUX: <ARRAY IDENTIFIER>;
"ARRAY" AUX[2:3];
ENTRY: AUX[2]: A RELATIVE TOLERANCE USED TO CONTROL THE CALCULATION OF THE DIAGONAL ELEMENTS; (SEE METHOD AND PERFORMANCE OF CHLDEC2, SECTION 3.1.1.1.1.2.1);
NORMAL EXIT: AUX[3]:= N;
ABNORMAL EXIT: IF THE DECOMPOSITION CANNOT BE CARRIED OUT BECAUSE THE MATRIX IS (NUMERICALLY) NOT POSITIVE DEFINITE, AUX[3]:= K - 1, WHERE K IS THE LAST STAGE NUMBER.

B: <ARRAY IDENTIFIER>;
"ARRAY" B[1:N];
ENTRY: THE RIGHT HAND SIDE OF THE SYSTEM OF LINEAR EQUATIONS;
EXIT: THE SOLUTION OF THE SYSTEM.

PROCEDURES USED:
CHLDEC2 = CP34310,
CHLSOL2 = CP34390.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLDECSOL2 SOLVES A SYSTEM OF LINEAR EQUATIONS WITH A SYMMETRIC POSITIVE DEFINITE COEFFICIENT MATRIX BY CALLING CHLDEC2, SECTION 3.1.1.1.1.2.1., AND, IF THIS CALL WAS SUCCESSFUL, CHLSOL2 (THIS SECTION).
SEE ALSO CHLDEC2, SECTION 3.1.1.1.1.2.1.

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLDECINV2, SECTION 3.1.1.1.1.2.4.

SUBSECTION: CHLDECSOL1.

CALLING SEQUENCE:

```
    THE HEADING OF THE PROCEDURE IS:
    "PROCEDURE" CHLDECSOL1(A, N, AUX, B); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1 : (N + 1) * N // 2];
            ENTRY: THE UPPER-TRIANGULAR PART OF THE  POSITIVE  DEFINITE
                   SYMMETRIC MATRIX MUST BE GIVEN COLUMNWISE IN ARRAY A
                   (THE (I,J)-TH ELEMENT OF THE MATRIX MUST BE GIVEN IN
                   A[(J - 1) * J // 2 + I] FOR 1 <= I <= J <= N);
            EXIT:  THE  CHOLESKY  DECOMPOSITION   OF  THE  MATRIX  IS
                   DELIVERED COLUMNWISE IN A.
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[2:3];
            ENTRY: AUX[2]: A  RELATIVE TOLERANCE  USED TO  CONTROL  THE
                   CALCULATION OF THE DIAGONAL ELEMENTS;
            NORMAL EXIT:   AUX[3]:= N;
            ABNORMAL EXIT: IF THE  DECOMPOSITION CANNOT BE  CARRIED OUT
                   BECAUSE  THE  MATRIX  IS (NUMERICALLY) NOT  POSITIVE
                   DEFINITE,  AUX[3]:= K - 1, WHERE K IS THE LAST STAGE
                   NUMBER.
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY: THE  RIGHT HAND  SIDE  OF  THE  SYSTEM  OF  LINEAR
                   EQUATIONS;
            EXIT:  THE SOLUTION OF THE SYSTEM.
```

PROCEDURES USED:

CHLDEC1 = CP34311,
CHLSOL1 = CP34391.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLDECSOL1 SOLVES A SYSTEM OF LINEAR EQUATIONS WITH A SYMMETRIC POSITIVE DEFINITE COEFFICIENT MATRIX BY CALLING CHLDEC1, SECTION 3.1.1.1.1.2.1., AND, IF THIS CALL WAS SUCCESSFUL, CHLSOL1 (THIS SECTION).
THE UPPER TRIANGLE OF THE COEFFICIENT MATRIX MUST BE STORED COLUMN-WISE IN A ONE-DIMENSIONAL ARRAY.
SEE ALSO CHLDEC1, SECTION 3.1.1.1.1.2.1.

EXAMPLE OF USE:

SEE EXAMPLE OF USE OF CHLDECINV1, SECTION 3.1.1.1.1.2.4.

SOURCE TEXT(S) :

```
"CODE" 34390;
    "PROCEDURE" CHLSOL2(A, N, B); "VALUE" N; "INTEGER" N; "ARRAY" A, B;
    "BEGIN" "INTEGER" I;
        "REAL" "PROCEDURE" MATVEC(L, U, I, A, B); "CODE" 34011;
        "REAL" "PROCEDURE" TAMVEC(L, U, I, A, B); "CODE" 34012;

        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
        B[I]:= (B[I] - TAMVEC(1, I - 1, I, A, B)) / A[I,I];
        "FOR" I:= N "STEP" - 1 "UNTIL" 1 "DO"
        B[I]:= (B[I] - MATVEC(I + 1, N, I, A, B)) / A[I,I]
    "END" CHLSOL2;
        "EOP"
"CODE" 34391;
    "PROCEDURE" CHLSOL1(A, N, B); "VALUE" N; "INTEGER" N; "ARRAY" A, B;
    "BEGIN" "INTEGER" I, II;
        "REAL" "PROCEDURE" VECVEC(L, U, SHIFT, A, B); "CODE" 34010;
        "REAL" "PROCEDURE" SEQVEC(L, U, I1, SHIFT, A, B); "CODE" 34016;

        II:= 0;
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" II:= II + I;
            B[I]:= (B[I] - VECVEC(1, I - 1, II - I, B, A)) / A[II]
        "END";
        "FOR" I:= N "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" B[I]:= (B[I] -
                SEQVEC(I + 1, N, II + I, 0, A, B)) / A[II];
            II:= II - I
        "END"
    "END" CHLSOL1;
        "EOP"
"CODE" 34392;
    "PROCEDURE" CHLDECSOL2(A, N, AUX, B); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX, B;
    "BEGIN"
        "PROCEDURE" CHLDEC2(A, N, AUX); "CODE" 34310;
        "PROCEDURE" CHLSOL2(A, N, B); "CODE" 34390;

        CHLDEC2(A, N, AUX);
        "IF" AUX[3] = N "THEN" CHLSOL2(A, N, B)
    "END" CHLDECSOL2;
        "EOP"
"CODE" 34393;
    "PROCEDURE" CHLDECSOL1(A, N, AUX, B); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX, B;
    "BEGIN"
        "PROCEDURE" CHLDEC1(A, N, AUX); "CODE" 34311;
        "PROCEDURE" CHLSOL1(A, N, B); "CODE" 34391;

        CHLDEC1(A, N, AUX);
        "IF" AUX[3] = N "THEN" CHLSOL1(A, N, B)
    "END" CHLDECSOL1;
        "EOP"
```

AUTHOR : T.J. DEKKER.

CONTRIBUTORS: S.P.N. VAN KAMPEN, J. KOK.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 731015.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS FOUR PROCEDURES:
A) CHLINV2, FOR THE INVERSION OF A SYMMETRIC POSITIVE DEFINITE MATRIX, IF THE MATRIX HAS BEEN DECOMPOSED BY A CALL OF THE PROCEDURE CHLDEC2, SECTION 3.1.1.1.1.2.1., OR CHLDECSOL2, SECTION 3.1.1.1.1.2.3.;
B) CHLINV1, FOR THE INVERSION OF A SYMMETRIC POSITIVE DEFINITE MATRIX, IF THE MATRIX HAS BEEN DECOMPOSED BY A CALL OF THE PROCEDURE CHLDEC1, SECTION 3.1.1.1.1.2.1., OR CHLDECSOL1, SECTION 3.1.1.1.1.2.3.;
C) CHLDECINV2, FOR THE INVERSION OF A MATRIX BY CHOLESKY'S SQUARE ROOT METHOD;
THE COEFFICIENT MATRIX HAS TO BE SYMMETRIC POSITIVE DEFINITE AND MUST BE GIVEN IN THE UPPER TRIANGLE OF A TWO-DIMENSIONAL ARRAY;
D) CHLDECINV1, FOR THE INVERSION OF A MATRIX BY CHOLESKY'S SQUARE ROOT METHOD;
THE COEFFICIENT MATRIX HAS TO BE SYMMETRIC POSITIVE DEFINITE AND MUST BE GIVEN COLUMNWISE IN A ONE-DIMENSIONAL ARRAY.

KEYWORDS:

MATRIX INVERSION,
POSITIVE DEFINITE SYMMETRIC MATRIX,
CHOLESKY DECOMPOSITION.

SUBSECTION: CHLINV2.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS:

```
"PROCEDURE" CHLINV2(A, N); "VALUE" N; "INTEGER" N; "ARRAY" A;
```

THE MEANING OF THE FORMAL PARAMETERS IS:

```
A:      <ARRAY IDENTIFIER>;
        "ARRAY" A[1:N,1:N];
        ENTRY: THE  UPPER-TRIANGULAR PART OF THE  CHOLESKY MATRIX
               AS PRODUCED  BY CHLDEC2, SECTION  3.1.1.1.1.2.1., OR
               CHLDECSOL2,  SECTION  3.1.1.1.1.2.3., MUST  BE GIVEN
               IN THE UPPER TRIANGLE OF A;
        EXIT:  THE  UPPER-TRIANGULAR PART  OF THE INVERSE MATRIX IS
               DELIVERED IN THE UPPER TRIANGLE OF A;
N:      <ARITHMETIC EXPRESSION>;
        THE ORDER OF THE MATRIX.
```

PROCEDURES USED:

```
MATVEC    = CP34011.
TAMVEC    = CP34012.
DUPVECROW = CP31031.
```

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: N.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLINV2 CALCULATES THE INVERSE OF A MATRIX, PROVIDED THAT THE MATRIX HAS BEEN DECOMPOSED BY A SUCCESSFUL CALL OF CHLDEC2 OR CHLDECSOL2;
THE INVERSE, X, OF U'U, WHERE U IS THE CHOLESKY MATRIX, IS OBTAINED FROM THE CONDITIONS THAT X BE SYMMETRIC AND UX BE A LOWER-TRIANGULAR MATRIX WHOSE MAIN DIAGONAL ELEMENTS ARE THE RECIPROCALS OF THE DIAGONAL ELEMENTS OF U. HEREWITH THE UPPER-TRIANGULAR ELEMENTS OF X ARE CALCULATED BY BACK SUBSTITUTION.
THE UPPER TRIANGLE OF THE INVERSE MATRIX IS DELIVERED IN THE UPPER TRIANGLE OF THE GIVEN ARRAY. SEE ALSO REF[1].

REFERENCES:

[1]. T.J. DEKKER.
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1.
MC TRACT 22, 1968, MATH. CENTR., AMSTERDAM.

EXAMPLE OF USE:

THE SYMMETRIC POSITIVE DEFINITE COEFFICIENT MATRIX (THE PASCAL MATRIX OF ORDER 4) OF THE SYSTEM OF EQUATIONS

```
X1 +     X2 +      X3 +       X4 =  2
X1 + 2 * X2 +  3 * X3 +  4 * X4 =  4
X1 + 3 * X2 +  6 * X3 + 10 * X4 =  8
X1 + 4 * X2 + 10 * X3 + 20 * X4 = 16
```

IS STORED IN THE TWO-DIMENSIONAL ARRAY PASCAL2.
THE INVERSE OF THE COEFFICIENT MATRIX AND THE SOLUTION OF THE LINEAR SYSTEM ARE CALCULATED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "COMMENT" TEST CHLDEC2, CHLSOL2 AND CHLINV2;
    "INTEGER" I, J;
    "ARRAY" PASCAL2[1:4,1:4], B[1:4], AUX[2:3];
    "PROCEDURE" CHLDEC2(A, N, AUX); "CODE" 34310;
    "PROCEDURE" CHLSOL2(A, N, B); "CODE" 34390;
    "PROCEDURE" CHLINV2(A, N); "CODE" 34400;
    "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" PASCAL2[1,J]:= 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" J "DO"
        PASCAL2[I,J]:= "IF" I = J "THEN" PASCAL2[I-1,J] * 2 "ELSE"
                       PASCAL2[I,J-1] + PASCAL2[I-1,J];
        B[J]:= 2 ** J
    "END";
    AUX[2]:= "-11;
    CHLDEC2(PASCAL2, 4, AUX);
    "IF" AUX[3] = 4 "THEN"
    "BEGIN" CHLSOL2(PASCAL2, 4, B); CHLINV2(PASCAL2, 4) "END"
    "ELSE" OUTPUT(61, "(""("MATRIX NOT POSITIVE DEFINITE")", /")");
    OUTPUT(61, "("4B")");
    OUTPUT(61, "(""("SOLUTION WITH CHLDEC2 AND CHLSOL2:")", /")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    OUTPUT(61, "("4B+D.5D")", B[I]);
    OUTPUT(61, "("//, 4B")");
    OUTPUT(61, "(""("INVERSE MATRIX WITH CHLINV2:")", /, 4B")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        "IF" J < I "THEN" OUTPUT(61, "("12B")") "ELSE"
        OUTPUT(61, "("+ZD.5D3B")", PASCAL2[I,J]);
        OUTPUT(61, "("/, 4B")")
    "END"
"END"
```

THIS PROGRAM DELIVERS:

```
SOLUTION WITH CHLDEC2 AND CHLSOL2:
+0.00000    +4.00000    -4.00000    +2.00000

INVERSE MATRIX WITH CHLINV2:
 +4.00000    -6.00000    +4.00000    -1.00000
            +14.00000   -11.00000    +3.00000
                        +10.00000    -3.00000
                                     +1.00000
```

SUBSECTION: CHLINV1.

CALLING SEQUENCE:

```
    THE HEADING OF THE PROCEDURE IS:
    "PROCEDURE" CHLINV1(A, N); "VALUE" N; "INTEGER" N; "ARRAY" A;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:(N + 1) * N // 2];
            ENTRY: THE  UPPER-TRIANGULAR PART OF THE CHOLESKY  MATRIX
                   AS PRODUCED  BY CHLDEC1, SECTION  3.1.1.1.1.2.1., OR
                   CHLDECSOL1,  SECTION  3.1.1.1.1.2.3., MUST  BE GIVEN
                   COLUMNWISE IN ARRAY A;
            EXIT:  THE  UPPER-TRIANGULAR  PART OF THE INVERSE MATRIX IS
                   DELIVERED COLUMNWISE IN ARRAY A;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX.
```

PROCEDURES USED:

```
    SEQVEC    = CP34016,
    SYMMATVEC = CP34018.
```

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: N.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLINV1 CALCULATES THE INVERSE OF A MATRIX, PROVIDED THAT THE MATRIX HAS BEEN DECOMPOSED BY A SUCCESSFUL CALL OF CHLDEC1 OR CHLDECSOL1;
THE UPPER TRIANGLE OF THE INVERSE MATRIX IS DELIVERED COLUMNWISE IN THE ONE-DIMENSIONAL ARRAY.
SEE ALSO METHOD AND PERFORMANCE OF CHLINV2 (THIS SECTION).

EXAMPLE OF USE:

THE SYMMETRIC POSITIVE DEFINITE COEFFICIENT MATRIX (THE PASCAL MATRIX OF ORDER 4) OF THE SYSTEM OF EQUATIONS

```
        X1 +     X2 +      X3 +      X4 =  2
        X1 + 2 * X2 +  3 * X3 +  4 * X4 =  4
        X1 + 3 * X2 +  6 * X3 + 10 * X4 =  8
        X1 + 4 * X2 + 10 * X3 + 20 * X4 = 16
```

IS STORED IN THE ONE-DIMENSIONAL ARRAY PASCAL1.
THE INVERSE OF THE COEFFICIENT MATRIX AND THE SOLUTION OF THE LINEAR SYSTEM ARE CALCULATED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "COMMENT" TEST CHLDEC1, CHLSOL1 AND CHLINV1;
    "INTEGER" I, J, JJ;
    "ARRAY" PASCAL1[1:(4 + 1) * 4 // 2], B[1:4], AUX[2:3];

    "PROCEDURE" CHLDEC1(A, N, AUX); "CODE" 34311;
    "PROCEDURE" CHLSOL1(A, N, B); "CODE" 34391;
    "PROCEDURE" CHLINV1(A, N); "CODE" 34401;

    JJ:= 1;
    "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" PASCAL1[JJ]:= 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" J "DO"
        PASCAL1[JJ + I - 1]:= "IF" I = J "THEN"
        PASCAL1[JJ + I - 2] * 2 "ELSE"
        PASCAL1[JJ + I - 2] + PASCAL1[JJ + I - J];
        B[J]:= 2 ** J;
        JJ:= JJ + J
    "END";

    AUX[2]:= "-11;
    CHLDEC1(PASCAL1, 4, AUX);
    "IF" AUX[3] = 4 "THEN"
    "BEGIN" CHLSOL1(PASCAL1, 4, B); CHLINV1(PASCAL1, 4) "END"
    "ELSE" OUTPUT(61, "(""("MATRIX NOT POSITIVE DEFINITE")", /")");

    OUTPUT(61,"("4B")");
    OUTPUT(61, "(""("SOLUTION WITH CHLDEC1 AND CHLSOL1:")", /")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    OUTPUT(61, "("4B+D.5D")", B[I]);
    OUTPUT(61, "("2/, 4B")");
    OUTPUT(61, "(""("INVERSE MATRIX WITH CHLINV1:")", /, 4B")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        "IF" J < I "THEN" OUTPUT(61, "("12B")") "ELSE"
        OUTPUT(61, "("+ZD.5D3B")", PASCAL1[(J - 1) * J // 2 + I]);
        OUTPUT(61, "("/, 4B")")
    "END"
"END"
```

THIS PROGRAM DELIVERS:

```
SOLUTION WITH CHLDEC1 AND CHLSOL1:
+0.00000     +4.00000     -4.00000     +2.00000

INVERSE MATRIX WITH CHLINV1:
 +4.00000     -6.00000     +4.00000     -1.00000
             +14.00000    -11.00000     +3.00000
                          +10.00000     -3.00000
                                        +1.00000
```

SUBSECTION: CHLDECINV2.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS:
```
"PROCEDURE" CHLDECINV2(A, N, AUX); "VALUE" N; "INTEGER" N;
"ARRAY" A, AUX;
```

THE MEANING OF THE FORMAL PARAMETERS IS:

A: <ARRAY IDENTIFIER>;
"ARRAY" A[1:N,1:N];
ENTRY: THE UPPER TRIANGLE OF THE POSITIVE DEFINITE MATRIX MUST BE GIVEN IN THE UPPER TRIANGLE OF A (THE ELEMENTS A[I,J], I <= J);
EXIT: THE UPPER-TRIANGULAR PART OF THE INVERSE MATRIX IS DELIVERED IN THE UPPER TRIANGLE OF A.

N: <ARITHMETIC EXPRESSION>;
THE ORDER OF THE MATRIX;

AUX: <ARRAY IDENTIFIER>;
"ARRAY" AUX[2:3];
ENTRY: AUX[2]: A RELATIVE TOLERANCE USED TO CONTROL THE CALCULATION OF THE DIAGONAL ELEMENTS;
NORMAL EXIT: AUX[3]:= N;
ABNORMAL EXIT: IF THE DECOMPOSITION CANNOT BE CARRIED OUT BECAUSE THE MATRIX IS (NUMERICALLY) NOT POSITIVE DEFINITE, AUX[3]:= K - 1, WHERE K IS THE LAST STAGE NUMBER.

PROCEDURES USED:

CHLDEC2 = CP34310,
CHLINV2 = CP34400.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLDECINV2 CALCULATES THE INVERSE OF A SYMMETRIC POSITIVE DEFINITE MATRIX BY CALLING CHLDEC2 AND, IF THIS CALL WAS SUCCESSFUL, CHLINV2.
THE UPPER TRIANGLE OF THE INVERSE MATRIX IS DELIVERED IN THE UPPER TRIANGLE OF THE GIVEN ARRAY.
SEE ALSO METHOD AND PERFORMANCE OF CHLINV2 (THIS SECTION) AND CHLDEC2, SECTION 3.1.1.1.1.2.1.

EXAMPLE OF USE:

THE SYMMETRIC POSITIVE DEFINITE COEFFICIENT MATRIX (THE PASCAL MATRIX OF ORDER 4) OF THE SYSTEM OF EQUATIONS

```
X1 +     X2 +      X3 +      X4 =  2
X1 + 2 * X2 +  3 * X3 +  4 * X4 =  4
X1 + 3 * X2 +  6 * X3 + 10 * X4 =  8
X1 + 4 * X2 + 10 * X3 + 20 * X4 = 16
```

IS STORED IN THE TWO-DIMENSIONAL ARRAY PASCAL2.
THE DETERMINANT AND THE INVERSE OF THE COEFFICIENT MATRIX AND THE SOLUTION OF THE LINEAR SYSTEM ARE CALCULATED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "COMMENT" TEST CHLDECSOL2, CHLDETERM2 AND CHLDECINV2;
    "INTEGER" I, J;
    "ARRAY" PASCAL2[1:4,1:4], B[1:4], AUX[2:3];
    "REAL" DETERMINANT;
    "PROCEDURE" CHLDECSOL2(A, N, AUX, B); "CODE" 34392;
    "REAL" "PROCEDURE" CHLDETERM2(A, N); "CODE" 34312;
    "PROCEDURE" CHLDECINV2(A, N, AUX); "CODE" 34402;
    "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" PASCAL2[1,J]:= 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" J "DO"
        PASCAL2[I,J]:= "IF" I = J "THEN" PASCAL2[I-1,J] * 2 "ELSE"
                       PASCAL2[I,J-1] + PASCAL2[I-1,J];
        B[J]:= 2 ** J
    "END";
    AUX[2]:= "-11;
    CHLDECSOL2(PASCAL2, 4, AUX, B);
    "IF" AUX[3] = 4 "THEN" DETERMINANT:= CHLDETERM2(PASCAL2, 4)
    "ELSE" OUTPUT(61, "(""("MATRIX NOT POSITIVE DEFINITE")", /")");
    OUTPUT(61, "("4B")");
    OUTPUT(61, "(""("SOLUTION WITH CHLDECSOL2:")", /")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    OUTPUT(61, "("4B+D.5D")", B[I]);
    OUTPUT(61, "("//, 4B, "("DETERMINANT WITH CHLDETERM2: ")",
               +D.5D, 2/, 4B")", DETERMINANT);
    "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" PASCAL2[1,J]:= 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" J "DO"
        PASCAL2[I,J]:= "IF" I = J "THEN" PASCAL2[I-1,J] * 2 "ELSE"
                       PASCAL2[I,J-1] + PASCAL2[I-1,J]
    "END";
    CHLDECINV2(PASCAL2, 4, AUX);
    OUTPUT(61, "(""("INVERSE MATRIX WITH CHLDECINV2:")", /, 4B")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        "IF" J < I "THEN" OUTPUT(61, "("12B")") "ELSE"
        OUTPUT(61, "("+ZD.5D3B")", PASCAL2[I,J]);
        OUTPUT(61, "("/, 4B")")
    "END"
"END"
```

```
    THIS PROGRAM DELIVERS:

    SOLUTION WITH CHLDECSOL2:
    +0.00000     +4.00000     -4.00000     +2.00000

    DETERMINANT WITH CHLDETERM2: +1.00000

    INVERSE MATRIX WITH CHLDECINV2:
     +4.00000     -6.00000     +4.00000     -1.00000
                 +14.00000    -11.00000     +3.00000
                              +10.00000     -3.00000
                                            +1.00000
```

SUBSECTION: CHLDECINV1.

CALLING SEQUENCE:

```
    THE HEADING OF THE PROCEDURE IS:
    "PROCEDURE" CHLDECINV1(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    A:      <ARRAY IDENTIFIER>;
            "ARRAY" A[1:(N + 1) * N // 2];
             ENTRY: THE UPPER-TRIANGULAR PART OF THE SYMMETRIC POSITIVE
                    DEFINITE MATRIX  MUST BE GIVEN COLUMNWISE IN ARRAY A
                    (THE (I,J)-TH ELEMENT OF THE MATRIX MUST BE GIVEN IN
                    A[(J - 1) * J // 2 + I] FOR 1 <= I <= J <= N);
            EXIT:   THE  UPPER-TRIANGULAR  PART OF THE INVERSE MATRIX IS
                    DELIVERED COLUMNWISE IN ARRAY A;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[2:3];
            ENTRY: AUX[2]: A  RELATIVE TOLERANCE  USED TO  CONTROL  THE
                    CALCULATION OF THE DIAGONAL ELEMENTS; (SEE METHOD
                    AND PERFORMANCE OF CHLDEC2, SECTION 3.1.1.1.1.2.1);
            NORMAL EXIT:    AUX[3]:= N;
            ABNORMAL EXIT: IF THE  DECOMPOSITION CANNOT BE  CARRIED OUT
                    BECAUSE  THE  MATRIX  IS (NUMERICALLY) NOT  POSITIVE
                    DEFINITE,  AUX[3]:= K - 1, WHERE K IS THE LAST STAGE
                    NUMBER.
```

PROCEDURES USED:

    CHLDEC1 = CP34311,
    CHLINV1 = CP34401.

RUNNING TIME: ROUGHLY PROPORTIONAL TO N CUBED.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE PROCEDURE CHLDECINV1 CALCULATES THE INVERSE OF A SYMMETRIC POSITIVE DEFINITE MATRIX BY CALLING CHLDEC1 AND, IF THIS CALL WAS SUCCESSFUL, CHLINV1.
THE UPPER TRIANGLE OF THE INVERSE MATRIX IS DELIVERED COLUMNWISE IN THE GIVEN ONE-DIMENSIONAL ARRAY.
SEE ALSO METHOD AND PERFORMANCE OF CHLINV2, (THIS SECTION) AND CHLDEC1, SECTION 3.1.1.1.1.2.1.

EXAMPLE OF USE:

THE SYMMETRIC POSITIVE DEFINITE COEFFICIENT MATRIX (THE PASCAL MATRIX OF ORDER 4) OF THE SYSTEM OF EQUATIONS

```
X1 +     X2 +      X3 +      X4 =  2
X1 + 2 * X2 +  3 * X3 +  4 * X4 =  4
X1 + 3 * X2 +  6 * X3 + 10 * X4 =  8
X1 + 4 * X2 + 10 * X3 + 20 * X4 = 16
```

IS STORED IN THE ONE-DIMENSIONAL ARRAY PASCAL1.
THE DETERMINANT AND THE INVERSE OF THE COEFFICIENT MATRIX AND THE SOLUTION OF THE LINEAR SYSTEM ARE CALCULATED BY THE FOLLOWING PROGRAM:

```
"BEGIN" "COMMENT" TEST CHLDECSOL1, CHLDETERM1 AND CHLDECINV1;
    "INTEGER" I, J, JJ;
    "ARRAY" PASCAL1[1:(4 + 1) * 4 // 2], B[1:4], AUX[2:3];
    "REAL" DETERMINANT;

    "PROCEDURE" CHLDECSOL1(A, N, AUX, B); "CODE" 34393;
    "REAL" "PROCEDURE" CHLDETERM1(A, N); "CODE" 34313;
    "PROCEDURE" CHLDECINV1(A, N, AUX); "CODE" 34403;

    JJ:= 1;
    "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" PASCAL1[JJ]:= 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" J "DO"
        PASCAL1[JJ + I - 1]:= "IF" I = J "THEN"
        PASCAL1[JJ + I - 2] * 2 "ELSE"
        PASCAL1[JJ + I - 2] + PASCAL1[JJ + I - J];
        B[J]:= 2 ** J;
        JJ:= JJ + J
    "END";

    AUX[2]:= "-11;
    CHLDECSOL1(PASCAL1, 4, AUX, B);
    "IF" AUX[3] = 4 "THEN" DETERMINANT:= CHLDETERM1(PASCAL1, 4)
    "ELSE" OUTPUT(61, "(""("MATRIX NOT POSITIVE DEFINITE")", /")");

    OUTPUT(61, "("4B")");
    OUTPUT(61, "(""("SOLUTION WITH CHLDECSOL1:")", /")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    OUTPUT(61, "("4B+D.5D")", B[I]);
    OUTPUT(61, "("//, 4B, "("DETERMINANT WITH CHLDETERM1: ")",
              +D.5D, 2/, 4B")", DETERMINANT);

    JJ:= 1;
    "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" PASCAL1[JJ]:= 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" J "DO"
        PASCAL1[JJ + I - 1]:= "IF" I = J "THEN"
        PASCAL1[JJ + I - 2] * 2 "ELSE"
        PASCAL1[JJ + I - 2] + PASCAL1[JJ + I - J];
        JJ:= JJ + J
    "END";


    CHLDECINV1(PASCAL1, 4, AUX);

    OUTPUT(61, "(""("INVERSE MATRIX WITH CHLDECINV1:")", /, 4B")");
    "FOR" I:= 1 "STEP" 1 "UNTIL" 4 "DO"
    "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" 4 "DO"
        "IF" J < I "THEN" OUTPUT(61, "("12B")") "ELSE"
        OUTPUT(61, "("+ZD.5D3B")", PASCAL1[(J - 1) * J // 2 + I]);
        OUTPUT(61, "("/, 4B")")
    "END"
"END"
```

```
    THIS PROGRAM DELIVERS:

    SOLUTION WITH CHLDECSOL1:
    +0.00000     +4.00000     -4.00000     +2.00000

    DETERMINANT WITH CHLDETERM1: +1.00000

    INVERSE MATRIX WITH CHLDECINV1:
     +4.00000     -6.00000     +4.00000     -1.00000
                 +14.00000    -11.00000     +3.00000
                              +10.00000     -3.00000
                                            +1.00000
```

SOURCE TEXT(S) :

```
"CODE" 34400;
    "PROCEDURE" CHLINV2(A, N); "VALUE" N; "INTEGER" N; "ARRAY" A;
    "BEGIN" "REAL" R; "INTEGER" I, J, I1;
        "ARRAY" U[1:N];
        "PROCEDURE" DUPVECROW(L, U, I, A, B); "CODE" 31031;
        "REAL" "PROCEDURE" MATVEC(L, U, I, A, B); "CODE" 34011;
        "REAL" "PROCEDURE" TAMVEC(L, U, I, A, B); "CODE" 34012;

        "FOR" I:= N "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" R:= 1 / A[I,I]; I1:= I + 1;
            DUPVECROW(I1, N, I, U, A);
            "FOR" J:= N "STEP" - 1 "UNTIL" I1 "DO" A[I,J]:=
            - (TAMVEC(I1, J, J, A, U) + MATVEC(J + 1, N, J, A, U)) * R;
            A[I,I]:= (R - MATVEC(I1, N, I, A, U)) * R
        "END"
    "END" CHLINV2;
        "EOP"

"CODE" 34401;
    "PROCEDURE" CHLINV1(A, N); "VALUE" N; "INTEGER" N; "ARRAY" A;
    "BEGIN" "INTEGER" I, II, I1, J, IJ; "REAL" R;
        "ARRAY" U[1:N];
        "REAL" "PROCEDURE" SEQVEC(L, U, I1, SHIFT, A, B); "CODE" 34016;
        "REAL" "PROCEDURE" SYMMATVEC(L, U, I, A, B); "CODE" 34018;

        II:= (N + 1) * N // 2;
        "FOR" I:= N "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" R:= 1 / A[II]; I1:= I + 1; IJ:= II + I;
            "FOR" J:= I1 "STEP" 1 "UNTIL" N "DO"
            "BEGIN" U[J]:= A[IJ]; IJ:= IJ + J "END";
            "FOR" J:= N "STEP" - 1 "UNTIL" I1 "DO"
            "BEGIN" IJ:= IJ - J; A[IJ]:= -SYMMATVEC(I1, N, J, A, U) * R
            "END";
            A[II]:= (R - SEQVEC(I1, N, II + I, 0, A, U)) * R;
            II:= II - I
        "END"
    "END" CHLINV1;
        "EOP"
```

```
"CODE" 34402;
    "PROCEDURE" CHLDECINV2(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "BEGIN"
        "PROCEDURE" CHLDEC2(A, N, AUX); "CODE" 34310;
        "PROCEDURE" CHLINV2(A, N); "CODE" 34400;

        CHLDEC2(A, N, AUX);
        "IF" AUX[3] = N "THEN" CHLINV2(A, N)
    "END" CHLDECINV2;
        "EOP"

"CODE" 34403;
    "PROCEDURE" CHLDECINV1(A, N, AUX); "VALUE" N; "INTEGER" N;
    "ARRAY" A, AUX;
    "BEGIN"
        "PROCEDURE" CHLDEC1(A, N, AUX); "CODE" 34311;
        "PROCEDURE" CHLINV1(A, N); "CODE" 34401;

        CHLDEC1(A, N, AUX);
        "IF" AUX[3] = N "THEN" CHLINV1(A, N)
    "END" CHLDECINV1;
        "EOP"
```

AUTHORS : J.R.BUNCH,L,KAUFMAN,B.N.PARLETT.

CONTRIBUTOR : C.H.CONVENT.

INSTITUTE : UNIVERSITY OF AMSTERDAM.

RECEIVED : 770712.

BRIEF DESCRIPTION :

DECSYM2 CALCULATES THE LDL' DECOMPOSITION OF A SYMMETRIC MATRIX.
THE MATRIX MAY BE INDEFINITE AND/OR SINGULAR;

KEYWORDS :

GENERAL SYMMETRIC MATRIX,
LDL' DECOMPOSITION,
BLOCK DIAGONAL PIVOTING;

CALLING SEQUENCE :

```
THE HEADING OF THE PROCEDURE READS :
"PROCEDURE" DECSYM2(A,N,TOL,AUX,P,DETAUX);
"VALUE" N;"INTEGER" N;"REAL" TOL;
"ARRAY" A,DETAUX;"INTEGER" "ARRAY" AUX,P;
"CODE" 34291;

THE MEANING OF THE FORMAL PARAMETERS IS :
A :     <ARRAY IDENTIFIER>;
        "ARRAY" A[1:N,1:N];
        ENTRY : THE SYMMETRIC COEFFICIENT MATRIX;
        EXIT  : THE ELEMENTS OF THE LDL' DECOMPOSITION OF A ARE
                STORED IN THE UPPER TRIANGULAR PART OF A. HERE D
                IS A BLOCK DIAGONAL MATRIX WITH BLOCKS OF ORDER 1
                OR 2. FOR A BLOCK OF ORDER 2 WE ALWAYS HAVE
                D[I,I+1]^=0 AND L[I+1,I]=0,SO THAT D AND L' FIT
                IN THE UPPER TRIANGULAR PART OF A.
                THE STRICTLY LOWER TRIANGULAR PART OF A IS LEFT
                UNDISTURBED.
N :     <ARITHMETIC EXPRESSION>;
        THE ORDER OF THE MATRIX;
```

```
TOL :   <ARITHMETIC EXPRESSION>;
        A RELATIVE TOLERANCE, USED TO CONTROL THE
        CALCULATION OF THE BLOCK DIAGONAL ELEMENTS;
AUX :    <ARRAY IDENTIFIER>;
         "INTEGER" "ARRAY" AUX[2:5];
         EXIT :
         AUX[2] : IF THE MATRIX IS SYMMETRIC THEN 1, OTHER-
                  WISE 0;IN THE LAST CASE NO DECOMPOSITION
                  IS PERFORMED;
         AUX[3] : IF THE MATRIX IS SYMMETRIC THEN THE
                  NUMBER OF ITS POSITIVE EIGENVALUES,
                  OTHERWISE 0. IF AUX[3]=N THEN THE
                  MATRIX IS POSITIVE DEFINITE;
         AUX[4] : IF THE MATRIX IS SYMMETRIC THEN THE
                  NUMBER OF ITS NEGATIVE EIGENVALUES,
                  OTHERWISE 0. IF AUX[4]=N THEN THE
                  MATRIX IS NEGATIVE DEFINITE;
         AUX[5] : IF THE MATRIX IS SYMMETRIC THEN THE
                  NUMBER OF ITS ZERO EIGENVALUES, OTHERWISE
                  N; SO, IF AUX[5]=0 THEN THE MATRIX IS
                  SYMMETRIC AND NON-SINGULAR;
P :      <ARRAY IDENTIFIER>;
         "INTEGER" "ARRAY" P[1:N];
         EXIT :
         A VECTOR RECORDING
         1) THE INTERCHANGES PERFORMED ON A DURING THE
            COMPUTATION OF THE DECOMPOSITION AND
         2) THE BLOCK STRUCTURE OF D.
         IF P[I]>0 AND P[I+1]=0 A 2*2 BLOCK HAS BEEN
         FOUND I.E. D[I,I+1]^=0 AND L[I+1,I]=0;
DETAUX : <ARRAY IDENTIFIER>;
          "ARRAY" DETAUX[1:N];
          EXIT :
          IF P[I]>0 AND P[I+1]>0 THEN DETAUX[I] EQUALS
          THE EXIT-VALUE OF A[I,I].
          IF P[I]>0 AND P[I+1]=0 THEN DETAUX[I]=1 AND
          DETAUX[I+1] EQUALS THE VALUE OF THE DETERMINANT
          OF THE CORRESPONDING 2*2 DIAGONAL BLOCK AS
          DETERMINED BY DECSYM2;
```

PROCEDURES USED :

```
ELMROW=CP34030.
ICHROW=CP34032.
ICHROWCOL=CP34033.
```

RUNNING TIME : ROUGHLY PROPORTIONAL TO N**3.

METHOD AND PERFORMANCE :

THE PROCEDURE DECSYM2 COMPUTES THE LDL' DECOMPOSITION OF A SYMMETRIC MATRIX,ACCORDING TO A METHOD DUE TO BUNCH,KAUFMAN AND PARLETT (SEE [1],[2]). IT USES BLOCK DIAGONAL PIVOTING.
THE BLOCK DIAGONAL MATRIX D IS DELIVERED IN THE BLOCK DIAGONAL OF A. IF P[I]>0 AND P[I+1]=0 A 2*2 BLOCK HAS BEEN FOUND AND FURTHERMORE : L[I+1,I]=0 WHEN D[I,I+1]^=0.
THE STRICTLY UPPER TRIANGULAR PART OF L' IS DELIVERED IN THE STRICTLY UPPER TRIANGULAR PART OF A. FOR THE INERTIA PROBLEM IT IS IMPORTANT THAT DECSYM2 CAN ACCEPT SINGULAR MATRICES. NOTE, HOWEVER, THAT IN ORDER TO FIND THE NUMBER OF ZERO EIGENVALUES OF SINGULAR MATRICES, THE SINGULAR VALUE DECOMPOSITION MIGHT BE PREFERRED.
BEFORE THE DECOMPOSITION IS PERFORMED A CHECK IS MADE TO SEE WHETHER THE MATRIX IS SYMMETRIC. IF THE MATRIX IS ASYMMETRIC THEN NO DECOMPOSITION IS PERFORMED;

REFERENCES :

1) J.R.BUNCH,L.KAUFMAN.
SOME STABLE METHODS FOR CALCULATING INERTIA AND SOLVING SYMMETRIC LINEAR SYSTEMS.
MATHEMATICS OF COMPUTATION 31,P 163-180,1977.

2) J.R.BUNCH,L.KAUFMAN,B.N.PARLETT.
DECOMPOSITION OF A SYMMETRIC MATRIX.
NUMERISCHE MATHEMATIK 27,P 95-109,1976.

SOURCE TEXT :

```
"CODE" 34291;
"PROCEDURE" DECSYM2(A,N,TOL,AUX,P,DETAUX);
"VALUE" N;"INTEGER" N;"REAL" TOL;
"ARRAY" A,DETAUX;"INTEGER" "ARRAY" P,AUX;
"BEGIN" "INTEGER" I,J,K,M,IP1,IP2,DUMMY;"BOOLEAN" ONEBYONE,SYM;
   "REAL" DET,S,T,ALPHA,LAMBDA,SIGMA,AII,AIP1,AIP1I;
   "PROCEDURE" ELMROW(L,U,I,J,A,B,X);"CODE" 34024;
   "PROCEDURE" ICHROW(L,U,I,J,A);"CODE" 34032;
   "PROCEDURE" ICHROWCOL(L,U,I,J,A);"CODE" 34033;
   AUX[3]:=AUX[4]:=0;SYM:="TRUE";I:=0;
   "FOR" DUMMY:=0 "WHILE" SYM "AND" (I<N) "DO"
   "BEGIN" I:=I+1;J:=I;
      "FOR" M:=0 "WHILE" SYM "AND" (J<N) "DO"
      "BEGIN" J:=J+1;SYM:=SYM "AND" (A[I,J]=A[J,I]) "END";
   "END";
   "IF" SYM "THEN" AUX[2]:=1
        "ELSE" "BEGIN" AUX[2]:=0;"GOTO" ENDDEC "END";
   ALPHA:=(1+SQRT(17))/8;P[N]:=N;I:=1;
   "FOR" DUMMY:=0 "WHILE" I<N "DO"
   "BEGIN" IP1:=I+1;IP2:=I+2;AII:=ABS(A[I,I]);P[I]:=I;
      LAMBDA:=ABS(A[I,IP1]);J:=IP1;
      "FOR" M:=IP2 "STEP" 1 "UNTIL" N "DO"
      "IF" ABS(A[I,M])>LAMBDA "THEN"
      "BEGIN" J:=M;LAMBDA:=ABS(A[I,M]) "END";
      T:=ALPHA*LAMBDA;ONEBYONE:="TRUE";
      "IF" AII<T "THEN"
      "BEGIN" SIGMA:=LAMBDA;
         "FOR" M:=IP1 "STEP" 1 "UNTIL" J-1 "DO"
            "IF" ABS(A[M,J])>SIGMA "THEN" SIGMA:=ABS(A[M,J]);
         "FOR" M:=J+1 "STEP" 1 "UNTIL" N "DO"
            "IF" ABS(A[J,M])>SIGMA "THEN" SIGMA:=ABS(A[J,M]);
         "IF" SIGMA*AII<LAMBDA "THEN"
         "BEGIN" "IF" ALPHA*SIGMA<ABS(A[J,J]) "THEN"
            "BEGIN" ICHROW(J+1,N,I,J,A);ICHROWCOL(IP1,J-1,I,J,A);
               T:=A[I,I];A[I,I]:=A[J,J];A[J,J]:=T;P[I]:=J
            "END"
            "ELSE"
            "BEGIN" "IF" J>IP1 "THEN"
              "BEGIN" ICHROW(J+1,N,IP1,J,A);ICHROWCOL(IP2,J-1,IP1,J,A);
                  T:=A[I,I];A[I,I]:=A[J,J];A[J,J]:=T;
                  T:=A[I,J];A[I,J]:=A[I,IP1];A[I,IP1]:=T
               "END";
               DET:=A[I,I]*A[IP1,IP1]-A[I,IP1]**2;AIP1I:=A[I,IP1]/DET;
               AII:=A[I,I]/DET;AIP1:=A[IP1,IP1]/DET;P[I]:=J;P[IP1]:=0;
               DETAUX[I]:=1;DETAUX[IP1]:=DET;                    "COMMENT"
```

```
                 "FOR" J:=IP2 "STEP" 1 "UNTIL" N "DO"
                 "BEGIN" S:=AIP1I*A[IP1,J]-AIP1*A[I,J];
                    T:=AIP1I*A[I,J]-AII*A[IP1,J];ELMROW(J,N,J,I,A,A,S);
                    ELMROW(J,N,J,IP1,A,A,T);A[I,J]:=S;A[IP1,J]:=T
                 "END";
                 AUX[3]:=AUX[3]+1;AUX[4]:=AUX[4]+1;I:=IP2;
                 ONEBYONE:="FALSE"
              "END"
           "END"
        "END";
        "IF" ONEBYONE "THEN"
        "BEGIN" "IF" TOL<ABS(A[I,I]) "THEN"
           "BEGIN" AII:=A[I,I];DETAUX[I]:=A[I,I];
              "IF" AII>0 "THEN" AUX[3]:=AUX[3]+1 "ELSE" AUX[4]:=AUX[4]+1;
              "FOR" J:=IP1 "STEP" 1 "UNTIL" N "DO"
             "BEGIN" S:=-A[I,J]/AII;ELMROW(J,N,J,I,A,A,S);A[I,J]:=S "END"
           "END";I:=IP1
        "END"
     "END" WHILE I;
     "IF" I=N "THEN"
     "BEGIN" "IF" TOL<ABS(A[N,N]) "THEN"
        "BEGIN" "IF" A[N,N]>0 "THEN" AUX[3]:=AUX[3]+1
              "ELSE" AUX[4]:=AUX[4]+1
        "END";DETAUX[N]:=A[N,N]
     "END";
   ENDDEC:
     AUX[5]:=N-AUX[3]-AUX[4]
"END" DECSYM2;
       "EOP"
```

AUTHORS : J.R.BUNCH,L.KAUFMAN,B.N.PARLETT.

CONTRIBUTOR : C.H.CONVENT.

INSTITUTE : UNIVERSITY OF AMSTERDAM.

RECEIVED : 770712.

BRIEF DESCRIPTION :

DETERMSYM2 CALCULATES THE DETERMINANT OF A SYMMETRIC MATRIX.
THE LDL' DECOMPOSITION OF THE MATRIX, AS PRODUCED BY DECSYM2,
SHOULD BE AVAILABLE.

KEYWORDS :

GENERAL SYMMETRIC MATRIX,
LDL' DECOMPOSITION,
BLOCK DIAGONAL PIVOTING;

CALLING SEQUENCE :

```
THE HEADING OF THE PROCEDURE READS :
"REAL" "PROCEDURE" DETERMSYM2(DETAUX,N,AUX);
"VALUE" N;"INTEGER" N;
"ARRAY" DETAUX;"INTEGER" "ARRAY" AUX;
"CODE" 34294;

DETERMSYM2 : DELIVERS THE CALCULATED VALUE OF THE DETERMINANT OF
             THE MATRIX;

THE MEANING OF THE FORMAL PARAMETERS IS :
DETAUX :    <ARRAY IDENTIFIER>;
            "ARRAY" DETAUX[1:N];
            ENTRY : THE ARRAY DETAUX AS PRODUCED BY DECSYM2;
N      :    <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE ARRAY DETAUX
                ( = THE ORDER OF THE MATRIX );
AUX    :    <ARRAY IDENTIFIER>;
            "INTEGER" "ARRAY" AUX[2:5];
            ENTRY : THE ARRAY AUX AS PRODUCED BY DECSYM2;
```

PROCEDURES USED : NONE.

RUNNING TIME : PROPORTIONAL TO N;

METHOD AND PERFORMANCE :

FIRST OF ALL DECSYM2 SHOULD BE CALLED TO PERFORM THE LDL' DECOMPOSITION OF THE SYMMETRIC MATRIX,ACCORDING TO A METHOD DUE TO BUNCH,KAUFMAN AND PARLETT (SEE [1],[2]).
IF A 1*1 BLOCK HAS BEEN COMPUTED FOR D THEN DETAUX[I] CONTAINS THE VALUE OF D[I]. IF A 2*2 BLOCK HAS BEEN COMPUTED FOR D THEN DETAUX[I]=1 AND DETAUX[I+1] CONTAIN THE VALUE OF THE DETERMINANT OF THE CORRESPONDING 2*2 BLOCK.
THE COMPUTATION OF THE DETERMINANT IS DONE BY CALCULATING THE PRODUCT OF THE ELEMENTS OF DETAUX.

REFERENCES :

1) J.R.BUNCH,L.KAUFMAN.
SOME STABLE METHODS FOR CALCULATING INERTIA AND SOLVING SYMMETRIC LINEAR SYSTEMS.
MATHEMATICS OF COMPUTATION 31,P 163-180,1977.
2) J.R.BUNCH,L.KAUFMAN,B.N.PARLETT.
DECOMPOSITION OF A SYMMETRIC MATRIX.
NUMERISCHE MATHEMATIK 27,P 95-109,1976.

EXAMPLE OF USE :

```
"BEGIN" "COMMENT" EXAMPLE OF USE OF THE PROCEDURE DETERMSYM2;
        "INTEGER" I,J;"REAL" TOL,DETERMINANT;
        "REAL" "ARRAY" A[1:5,1:5],DETAUX[1:5];
        "INTEGER" "ARRAY" AUX[2:5],P[1:5];
        "PROCEDURE" DECSYM2(A,N,TOL,AUX,P,DETAUX);"CODE" 34291;
        "REAL" "PROCEDURE" DETERMSYM2(DETAUX,N,AUX);"CODE" 34294;

        A[1,1]:=A[1,2]:=-3;A[1,3]:=-18;A[1,4]:=-30;A[1,5]:=18;
        A[2,2]:=-1;A[2,3]:=-4;A[2,4]:=-48;A[2,5]:=8;
        A[3,3]:=-6;A[3,4]:=-274;A[3,5]:=6;
        A[4,4]:=119;A[4,5]:=19;
        A[5,5]:=216;
        "FOR" I:=1 "STEP" 1 "UNTIL" 5 "DO"
        "FOR" J:=I+1 "STEP" 1 "UNTIL" 5 "DO" A[J,I]:=A[I,J];
                                                              "COMMENT"
```

```
        OUTPUT(61,"(""("THE COEFFICIENT MATRIX :")",/")");
        "FOR" I:=1 "STEP" 1 "UNTIL" 5 "DO"
        "BEGIN" "FOR" J:=1 "STEP" 1 "UNTIL" 5 "DO"
                OUTPUT(61,"("-2ZD,4B")",A[I,J]);
                OUTPUT(61,"("/")")
        "END";
        TOL:="-14;
        DECSYM2(A,5,TOL,AUX,P,DETAUX);
        "IF" AUX[2]=1 "THEN"
             OUTPUT(61,"("/,"("THE MATRIX IS SYMMETRIC")"")")
        "ELSE" OUTPUT(61,"("/,"("THE MATRIX IS ASYMMETRIC.THE ")",
                          "("RESULTS ARE MEANINGLESS")"")");
        DETERMINANT:=DETERMSYM2(DETAUX,5,AUX);
        OUTPUT(61,"("/,"("THE DETERMINANT OF THE MATRIX : ")",
                   3Z3D.2D")",DETERMINANT)
"END";
```

THIS DELIVERS AS RESULT :

```
   THE COEFFICIENT MATRIX :
     -3       -3      -18     -30      18
     -3       -1       -4     -48       8
    -18       -4       -6    -274       6
    -30      -48     -274     119      19
     18        8        6      19     216

   THE MATRIX IS SYMMETRIC.
   THE DETERMINANT OF THE MATRIX :     168.00
```

SOURCE TEXT :

```
"CODE" 34294;
"REAL" "PROCEDURE" DETERMSYM2(DETAUX,N,AUX);
"VALUE" N;"INTEGER" N;
"ARRAY" DETAUX;"INTEGER" "ARRAY" AUX;
"BEGIN" "INTEGER" I;"REAL" DET;
        "IF" AUX[5]>0 "THEN" DET:=0 "ELSE"
        "BEGIN" DET:=1;
                "FOR" I:=1 "STEP" 1 "UNTIL" N "DO" DET:=DET*DETAUX[I]
        "END";
        DETERMSYM2:=DET
"END" DETERMSYM2;
```

AUTHORS : J.R.BUNCH,L.KAUFMAN,B.N.PARLETT.

CONTRIBUTOR : C.H.CONVENT.

INSTITUTE : UNIVERSITY OF AMSTERDAM.

RECEIVED : 770712.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS TWO PROCEDURES :
A) SOLSYM2 SOLVES A SYMMETRIC SYSTEM OF LINEAR EQUATIONS, ASSUMING THAT THE MATRIX HAS BEEN DECOMPOSED INTO LDL' FORM BY A CALL OF DECSYM2;
B) DECSOLSYM2 CALCULATES THE LDL' DECOMPOSITION OF A SYMMETRIC MATRIX; MOREOVER, IF THIS MATRIX IS NON-SINGULAR, THEN IT SOLVES A CORRESPONDING SYSTEM OF LINEAR EQUATIONS;

KEYWORDS :

GENERAL SYMMETRIC MATRIX,
LDL' DECOMPOSITION,
BLOCK DIAGONAL PIVOTING;

SUBSECTION : SOLSYM2.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE READS :

```
"PROCEDURE" SOLSYM2(A,N,B,P,DETAUX);
"VALUE" N;"INTEGER" N;
"ARRAY" A,B,DETAUX;"INTEGER" "ARRAY" P;
"CODE" 34292;
```

```
THE MEANING OF THE FORMAL PARAMETERS IS :
A :      <ARRAY IDENTIFIER>;
         "ARRAY" A[1:N,1:N];
         ENTRY : THE LDL' DECOMPOSITION OF A AS PRODUCED BY
                 DECSYM2;
N :      <ARITHMETIC EXPRESSION>;
         THE ORDER OF THE MATRIX;
B :      <ARRAY IDENTIFIER>;
         "ARRAY" B[1:N];
         ENTRY : THE RIGHT-HANDSIDE OF A SYSTEM OF LINEAR
                 EQUATIONS;
         EXIT  : THE CALCULATED SOLUTION VECTOR;
P :      <ARRAY IDENTIFIER>;
         "INTEGER" "ARRAY" P[1:N];
         ENTRY : A VECTOR RECORDING THE INTERCHANGES PERFORMED ON
                 THE MATRIX BY THE PROCEDURE DECSYM2. P ALSO CONTAINS
                 INFORMATION ON THE BLOCKSTRUCTURE OF THE MATRIX AS
                 DECOMPOSED BY DECSYM2;
DETAUX : <ARRAY IDENTIFIER>;
          "ARRAY" DETAUX[1:N];
          ENTRY : THE ARRAY DETAUX AS PRODUCED BY DECSYM2;
```

PROCEDURES USED :

MATVEC=CP34011.
ELMVECROW=CP34026.

RUNNING TIME : ROUGHLY PROPORTIONAL TO N**2.

METHOD AND PERFORMANCE :

THE PROCEDURE SOLSYM2 COMPUTES THE SOLUTION OF A SYMMETRIC SYSTEM OF LINEAR EQUATIONS,ASSUMING THAT THE MATRIX HAS BEEN DECOMPOSED INTO LDL' FORM BY A CALL OF DECSYM2. B IS OVERWRITTEN WITH THE SOLUTION VECTOR.

REFERENCES :

1) J.R.BUNCH,L.KAUFMAN.
SOME STABLE METHODS FOR CALCULATING INERTIA AND SOLVING SYMMETRIC LINEAR SYSTEMS.
MATHEMATICS OF COMPUTATION 31,P 163-180,1977.

2) J.R.BUNCH,L.KAUFMAN,B.N.PARLETT.
DECOMPOSITION OF A SYMMETRIC MATRIX.
NUMERISCHE MATHEMATIK 27,P 95-109,1976.

SUBSECTION : DECSOLSYM2.

CALLING SEQUENCE :

```
    THE HEADING OF THE PROCEDURE READS :
    "PROCEDURE" DECSOLSYM2(A,N,B,TOL,AUX);
    "VALUE" N;"INTEGER" N;"REAL" TOL;
    "ARRAY" A,B;"INTEGER" "ARRAY" AUX;
    "CODE" 34293;

    THE MEANING OF THE FORMAL PARAMETERS IS :
    A :     <ARRAY IDENTIFIER>;
            "ARRAY" A[1:N,1:N];
            ENTRY : SEE DECSYM2;
            EXIT  : SEE DECSYM2;
    N :     <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    B :     <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY : SEE SOLSYM2;
            EXIT  : THE CALCULATED SOLUTION VECTOR,WHEN A WAS FOUND
                    TO BE NON-SINGULAR.
                    B IS LEFT UNDISTURBED OTHERWISE;
    TOL :   <ARITHMETIC EXPRESSION>;
            ENTRY : SEE DECSYM2;
    AUX :   <ARRAY IDENTIFIER>;
            "INTEGER" "ARRAY" AUX[2:5];
            EXIT  : SEE DECSYM2;
```

PROCEDURES USED :

```
    DECSYM2=CP34291.
    SOLSYM2=CP34292.
```

RUNNING TIME : ROUGHLY PROPORTIONAL TO N**3.

METHOD AND PERFORMANCE :

THE PROCEDURE DECSOLSYM2 COMPUTES THE SOLUTION OF A SYMMETRIC SYSTEM OF LINEAR EQUATIONS. IT DOES SO BY FIRST CALLING THE PROCEDURE DECSYM2 TO COMPUTE THE LDL' DECOMPOSITION OF THE SYMMETRIC MATRIX,ACCORDING TO A METHOD DUE TO BUNCH,KAUFMAN AND PARLETT (SEE [1],[2]).
WHEN THE MATRIX IS FOUND TO BE NON-SINGULAR THE PROCEDURE SOLSYM2 IS CALLED TO COMPUTE THE SOLUTION VECTOR,AND THE LATTER OVERWRITES B. WHEN THE MATRIX IS FOUND TO BE SINGULAR THE PROCEDURE SOLSYM2 IS NOT CALLED AND B IS LEFT UNDISTURBED.

REFERENCES :

1) J.R.BUNCH,L.KAUFMAN.
SOME STABLE METHODS FOR CALCULATING INERTIA AND SOLVING SYMMETRIC LINEAR SYSTEMS.
MATHEMATICS OF COMPUTATION 31,P 163-180,1977.
2) J.R.BUNCH,L.KAUFMAN,B.N.PARLETT.
DECOMPOSITION OF A SYMMETRIC MATRIX.
NUMERISCHE MATHEMATIK 27,P 95-109,1976.

EXAMPLES OF USE :

```
"BEGIN" "COMMENT" EXAMPLE OF USE OF THE PROCEDURE DECSOLSYM2;
        "INTEGER" I,J;"REAL" TOL;
        "REAL" "ARRAY" A[1:5,1:5],B[1:5];
        "INTEGER" "ARRAY" AUX[2:5];
        "PROCEDURE" DECSOLSYM2(A,N,B,TOL,AUX);"CODE" 34293;

        A[1,1]:=A[1,2]:=-3;A[1,3]:=-18;A[1,4]:=-30;A[1,5]:=18;
        A[2,2]:=-1;A[2,3]:=-4;A[2,4]:=-48;A[2,5]:=8;
        A[3,3]:=-6;A[3,4]:=-274;A[3,5]:=6;
        A[4,4]:=119;A[4,5]:=19;
        A[5,5]:=216;
        "FOR" I:=1 "STEP" 1 "UNTIL" 5 "DO"
        "FOR" J:=I+1 "STEP" 1 "UNTIL" 5 "DO" A[J,I]:=A[I,J];

        OUTPUT(61,"("""("THE COEFFICIENTMATRIX :")",/")");
        "FOR" I:=1 "STEP" 1 "UNTIL" 5 "DO"
        "BEGIN" "FOR" J:=1 "STEP" 1 "UNTIL" 5 "DO"
                OUTPUT(61,"("-2ZD,4B")",A[I,J]);
                OUTPUT(61,"("/")")
        "END";                                             "COMMENT"
```

```
        OUTPUT(61,"("/,"("THE RHS-VECTOR :")",/")");
        "FOR" I:=1 "STEP" 1 "UNTIL" 5 "DO"
        "BEGIN" INPUT(60,"("")",B[I]);
                OUTPUT(61,"("-Z3D,4B")",B[I])
        "END";
        TOL:="-14;
        DECSOLSYM2(A,5,B,TOL,AUX);
        OUTPUT(61,"("2/")");
        "IF" AUX[2]=1 "THEN"
             OUTPUT(61,"(""("THE MATRIX IS SYMMETRIC")",/")")
        "ELSE" OUTPUT(61,"(""("THE MATRIX IS ASYMMETRIC.THE ")",
                          "("RESULTS ARE MEANINGLESS")",/")");
        OUTPUT(61,"("/,"("INERTIA : <")",D,"(",")",D,"(",")",D,
                  "(">")",/")",AUX[3],AUX[4],AUX[5]);
        OUTPUT(61,"("/,"("THE COMPUTED SOLUTION :")",/")");
        "FOR" I:=1 "STEP" 1 "UNTIL" 5 "DO"
        OUTPUT(61,"("-D.5D,4B")",B[I])
"END";
```

THIS DELIVERS AS RESULT :

```
   THE COEFFICIENT MATRIX :
     -3       -3      -18     -30       18
     -3       -1       -4     -48        8
    -18       -4       -6    -274        6
    -30      -48     -274     119       19
     18        8        6      19      216

   THE RHS-VECTOR :
    327      291     1290     275     1720

   THE MATRIX IS SYMMETRIC.
   INERTIA : <3,2,0>

   THE COMPUTED SOLUTION :
   -7.00000    -2.00000    -1.00000    -4.00000     9.00000
```

SOURCE TEXT(S) :

```
"CODE" 34292;
"PROCEDURE" SOLSYM2(A,N,B,P,DETAUX);
"VALUE" N;"INTEGER" N;
"ARRAY" A,B,DETAUX;"INTEGER" "ARRAY" P;
"BEGIN" "INTEGER" I,II,J,K,IP1,PI,PII,DUMMY;
   "REAL" DET,TEMP,SAVE;
   "REAL" "PROCEDURE" MATVEC(L,U,I,A,B);"CODE" 34011;
   "PROCEDURE" ELMVECROW(L,U,I,A,B,X);"CODE" 34026;
   I:=1;
   "FOR" DUMMY:=0 "WHILE" I<N "DO"
   "BEGIN" IP1:=I+1;PI:=P[I];SAVE:=B[PI];
      "IF" P[IP1]>0 "THEN"
      "BEGIN" B[PI]:=B[I];B[I]:=SAVE/A[I,I];
          ELMVECROW(IP1,N,I,B,A,SAVE);I:=IP1
      "END"
      "ELSE"
      "BEGIN" TEMP:=B[I];B[PI]:=B[IP1];DET:=DETAUX[IP1];
          B[I]:=(TEMP*A[IP1,IP1]-SAVE*A[I,IP1])/DET;
          B[IP1]:=(SAVE*A[I,I]-TEMP*A[I,IP1])/DET;
          ELMVECROW(I+2,N,I,B,A,TEMP);ELMVECROW(I+2,N,IP1,B,A,SAVE);
          I:=I+2
      "END"
   "END" WHILE I;
   "IF" I=N "THEN" "BEGIN" B[I]:=B[I]/A[I,I];I:=N-1 "END"
      "ELSE" I:=N-2;
   "FOR" DUMMY:=0 "WHILE" I>0 "DO"
   "BEGIN" "IF" P[I]=0 "THEN" II:=I-1 "ELSE" II:=I;
      "FOR" K:=II "STEP" 1 "UNTIL" I "DO"
      "BEGIN" SAVE:=B[K];SAVE:=SAVE+MATVEC(I+1,N,K,A,B);
         B[K]:=SAVE
      "END";
      PII:=P[II];B[I]:=B[PII];B[PII]:=SAVE;I:=II-1
   "END" WHILE I
"END" SOLSYM2;
        "EOP"


"CODE" 34293;
"PROCEDURE" DECSOLSYM2(A,N,B,TOL,AUX);
"VALUE" N;"INTEGER" N;"REAL" TOL;
"ARRAY" A,B;"INTEGER" "ARRAY" AUX;
"BEGIN" "REAL" "ARRAY" DETAUX[1:N];"INTEGER" "ARRAY" P[1:N];
        "PROCEDURE" DECSYM2(A,N,TOL,AUX,P,DETAUX);"CODE" 34291;
        "PROCEDURE" SOLSYM2(A,N,B,P,DETAUX);"CODE" 34292;
        DECSYM2(A,N,TOL,AUX,P,DETAUX);
        "IF" AUX[5]=0 "THEN" SOLSYM2(A,N,B,P,DETAUX)
"END" DECSOLSYM2;
        "EOP"
```

AUTHOR : T.J. DEKKER.

CONTRIBUTOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 731015.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS TWO PROCEDURES :
A) LSQORTDEC, FOR THE HOUSEHOLDER TRIANGULARIZATION WITH COLUMN INTERCHANGES OF THE COEFFICIENT MATRIX OF A LINEAR LEAST SQUARES PROBLEM;
B) LSQDGLINV, FOR THE CALCULATION OF THE DIAGONAL ELEMENTS OF THE INVERSE OF M'M, WHERE M IS THE COEFFICIENT MATRIX OF A LINEAR LEAST SQUARES PROBLEM.

KEY WORDS :

LINEAR LEAST SQUARES PROBLEM,
HOUSEHOLDER TRIANGULARIZATION.

SUBSECTION : LSQORTDEC.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" LSQORTDEC(A, N, M, AUX, AID, CI); "VALUE" N, M;
"INTEGER" N, M; "INTEGER""ARRAY" CI; "ARRAY" A, AUX, AID;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

A : <ARRAY IDENTIFIER>;
"ARRAY" A[1 : N,1 : M];
ENTRY :THE COEFFICIENT MATRIX OF THE LINEAR LEAST SQUARES PROBLEM;
EXIT : IN THE UPPER TRIANGLE OF A (THE ELEMENTS A[I,J] WITH I < J) THE SUPERDIAGONAL ELEMENTS OF THE UPPER-TRIANGULAR MATRIX, PRODUCED BY THE HOUSEHOLDER TRANSFORMATION; IN THE OTHER PART OF THE COLUMNS OF A THE SIGNIFICANT ELEMENTS OF THE GENERATING VECTORS OF THE HOUSEHOLDER MATRICES USED FOR THE HOUSEHOLDER TRIANGULARIZATION;

N : <ARITHMETIC EXPRESSION>;
NUMBER OF ROWS OF THE MATRIX;
M : <ARITHMETIC EXPRESSION>;
NUMBER OF COLUMNS OF THE MATRIX (N >= M);
AUX : <ARRAY IDENTIFIER>;
"ARRAY" AUX[2 : 5];
ENTRY : AUX[2] CONTAINS A RELATIVE TOLERANCE USED FOR CALCULATING THE DIAGONAL ELEMENTS OF THE UPPER-TRIANGULAR MATRIX;
EXIT :
AUX[3] DELIVERS THE NUMBER OF THE DIAGONAL ELEMENTS OF THE UPPER-TRIANGULAR MATRIX WHICH ARE FOUND NOT NEGLIGIBLE;
NORMAL EXIT AUX[3] = M;
AUX[5] := THE MAXIMUM OF THE EUCLIDEAN NORMS OF THE COLUMNS OF THE GIVEN MATRIX;
AID : <ARRAY IDENTIFIER>;
"ARRAY" AID[1 : M];
NORMAL EXIT (AUX[3] = M) : AID CONTAINS THE DIAGONAL ELEMENTS OF THE UPPER-TRIANGULAR MATRIX PRODUCED BY THE HOUSEHOLDER TRIANGULARIZATION;
CI : <ARRAY IDENTIFIER>;
"INTEGER""ARRAY" CI[1 : M];
EXIT : CI CONTAINS THE PIVOTAL INDICES OF THE INTERCHANGES OF THE COLUMNS OF THE GIVEN MATRIX.

PROCEDURES USED :

TAMMAT = CP34014,
ELMCOL = CP34023,
ICHCOL = CP34031.

REQUIRED CENTRAL MEMORY :

EXECUTION FIELD LENGTH : AN ARRAY OF M ELEMENTS IS DECLARED.

RUNNING TIME:

(C1 * M + C2) * M * (N - M / 3);
THE CONSTANTS C1 AND C2 DEPEND ON THE ARITHMETIC OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

THE PROCEDURE LSQORTDEC IS A MODIFICATION OF THE PROCEDURE LSQDEC DUE TO T.J. DEKKER (SEE REF [1]), WHERE A DERIVATION IS GIVEN OF A SET OF PROCEDURES BY P. BUSINGER AND G.H. GOLUB (SEE REF [2]). THE METHOD IS HOUSEHOLDER TRIANGULARIZATION WITH COLUMN INTERCHANGES.
LET M DENOTE THE GIVEN MATRIX. LSQORTDEC PRODUCES AN N-TH ORDER ORTHOGONAL MATRIX Q AND AN N * M UPPER-TRIANGULAR MATRIX R SUCH THAT R EQUALS QM WITH PERMUTED COLUMNS, Q IS THE PRODUCT OF AT MOST M HOUSEHOLDER MATRICES WHICH ARE REPRESENTED BY THEIR GENERATING VECTORS. M IS REDUCED TO R IN AT MOST M STAGES : AT THE K-TH STAGE THE K-TH COLUMN OF THE (ALREADY MODIFIED) MATRIX IS INTERCHANGED WITH THE COLUMN OF MAXIMUM EUCLIDEAN NORM (THE PIVOTAL COLUMN); THEN THE MATRIX IS MULTIPLIED WITH A HOUSEHOLDER MATRIX SUCH, THAT THE SUBDIAGONAL ELEMENTS OF THE K-TH COLUMN BECOME ZERO, WHILE THE FIRST K - 1 COLUMNS REMAIN UNCHANGED.
THE PROCESS TERMINATES PREMATURELY, IF AT SOME STAGE THE EUCLIDEAN NORM OF THE PIVOTAL COLUMN IS LESS THAN SOME TOLERANCE, VIZ. A GIVEN TOLERANCE (AUX[2]) TIMES THE MAXIMUM OF THE EUCLIDEAN NORMS OF THE COLUMNS OF THE GIVEN MATRIX.
LSQORTDEC DELIVERS THE SIGNIFICANT ELEMENTS OF THE GENERATING VECTOR OF THE K-TH HOUSEHOLDER MATRIX (THE FIRST K - 1 ELEMENTS OF THIS VECTOR BEING ZERO) IN THE LOWER TRIANGLE PART OF THE K-TH COLUMN OF THE ARRAY A (A[I,K] FOR I >= K).
OF THE RESULTING UPPER-TRIANGULAR MATRIX THE DIAGONAL ELEMENTS ARE DELIVERED SEPARATELY IN AN ARRAY AID, AND THE REMAINING ELEMENTS IN THE SUPER-TRIANGULAR PART OF THE ARRAY A.
FOR THE SOLUTION OF LEAST SQUARES PROBLEMS, ONLY CALLS WITH N >= M ARE USEFUL.

REFFRENCES :

[1] DEKKER, T.J. :
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1,
MC TRACT 22, 1968, MATHEMATISCH CENTRUM, AMSTERDAM.
[2] BUSINGER, P. AND G.H. GOLUB :
LINEAR LEAST SQUARES SOLUTION BY HOUSEHOLDER TRANSFORMATIONS,
NUM. MATH. 7 (1965), PP. 269 - 276.

EXAMPLE OF USE :

SEE EXAMPLE OF USE OF LSQSOL.

SUBSECTION : LSQDGLINV.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" LSQDGLINV(A, M, AID, CI, DIAG); "VALUE" M; "INTEGER" M;
"INTEGER""ARRAY" CI; "ARRAY" A, AID, DIAG;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

A, M, AID, CI :
SEE CALLING SEQUENCE OF LSQORTDEC (THIS SECTION);
THE CONTENTS OF A, AID AND CI SHOULD BE PRODUCED BY A
SUCCESSFUL CALL OF LSQORTDEC (AUX[3] = M) .

DIAG : <ARRAY IDENTIFIER>; "ARRAY" DIAG[1 : M];
EXIT : THE DIAGONAL ELEMENTS OF THE INVERSE OF M'M
WHERE M IS THE MATRIX OF THE LINEAR LEAST SQUARES
PROBLEM.

PROCEDURES USED :
VECVEC = CP34010,
TAMVEC = CP34012.

RUNNING TIME :
(C3 * M + C4) * M * M;
THE CONSTANTS C3 AND C4 DEPEND ON THE ARITHMETIC
OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

LSQDGLINV SHOULD BE CALLED AFTER A SUCCESSFUL CALL OF LSQORTDEC,
I.E. IF AUX[3] = M. LSQDGLINV CALCULATES THE DIAGONAL ELEMENTS
OF THE INVERSE OF M'M, WHERE M IS THE MATRIX OF A LINEAR
LEAST SQUARES PROBLEM.
THESE VALUES CAN BE USED FOR THE COMPUTATION OF THE STANDARD
DEVIATIONS OF LEAST SQUARES SOLUTIONS.

EXAMPLE OF USE :

SEE EXAMPLE OF USE OF LSQSOL.

SOURCE TEXT(S) :

```
"CODE" 34134;
    "PROCEDURE" LSQORTDEC(A, N, M, AUX, AID, CI); "VALUE" N, M;
    "INTEGER" N, M; "ARRAY" A, AUX, AID; "INTEGER" "ARRAY" CI;
    "BEGIN" "INTEGER" J, K, KPIV;
        "REAL" BETA, SIGMA, NORM, W, EPS, AKK, AIDK;
        "ARRAY" SUM[1:M];
        "REAL""PROCEDURE" TAMMAT(L, U, I, J, A, B);"CODE" 34014;
        "PROCEDURE" ELMCOL(L, U, I, J, A, B, X);"CODE" 34023;
        "PROCEDURE" ICHCOL(L, U, I, J, A);"CODE" 34031;

        NORM:= 0; AUX[3]:= M;
        "FOR" K:= 1 "STEP" 1 "UNTIL" M "DO"
        "BEGIN" W:= SUM[K]:= TAMMAT(1, N, K, K, A, A);
            "IF" W > NORM "THEN" NORM:= W
        "END";
        W:= AUX[5]:= SQRT(NORM); EPS:= AUX[2] * W;
        "FOR" K:= 1 "STEP" 1 "UNTIL" M "DO"
        "BEGIN" SIGMA:= SUM[K]; KPIV:= K;
            "FOR" J:= K + 1 "STEP" 1 "UNTIL" M "DO"
            "IF" SUM[J] > SIGMA "THEN"
            "BEGIN" SIGMA:= SUM[J]; KPIV:= J "END";
            "IF" KPIV ^= K "THEN"
            "BEGIN" SUM[KPIV]:= SUM[K]; ICHCOL(1, N, K, KPIV, A) "END";
            CI[K]:= KPIV; AKK:= A[K,K];
            SIGMA:= TAMMAT(K, N, K, K, A, A); W:= SQRT(SIGMA);
            AIDK:= AID[K]:= "IF" AKK < 0 "THEN" W "ELSE" - W;
            "IF" W < EPS "THEN"
            "BEGIN" AUX[3]:= K - 1; "GO TO" ENDDEC "END";
            BETA:= 1 / (SIGMA - AKK * AIDK); A[K,K]:= AKK - AIDK;
            "FOR" J:= K + 1 "STEP" 1 "UNTIL" M "DO"
            "BEGIN" ELMCOL(K, N, J, K, A, A, - BETA * TAMMAT(K, N,
                K, J, A, A)); SUM[J]:= SUM[J] - A[K,J] ** 2
            "END"
        "END" FOR K;
    ENDDEC:
    "END" LSQORTDEC;
        "EOP"
```

```
"CODE" 34132;
    "PROCEDURE" LSQDGLINV(A, M, AID, CI, DIAG); "VALUE" M; "INTEGER" M;
    "ARRAY" A, AID, DIAG; "INTEGER" "ARRAY" CI;
    "BEGIN" "INTEGER" J, K, CIK;
        "REAL" W;
        "REAL""PROCEDURE" VECVEC(L, U, S, A, B);"CODE" 34010;
        "REAL""PROCEDURE" TAMVEC(L, U, I, A, B);"CODE" 34012;

        "FOR" K:= 1 "STEP" 1 "UNTIL" M "DO"
        "BEGIN" DIAG[K]:= 1 / AID[K];
            "FOR" J:= K + 1 "STEP" 1 "UNTIL" M "DO"
            DIAG[J]:= - TAMVEC(K, J - 1, J, A, DIAG) / AID[J];
            DIAG[K]:= VECVEC(K, M, 0, DIAG, DIAG)
        "END";
        "FOR" K:= M "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" CIK:= CI[K]; "IF" CIK ^= K "THEN"
            "BEGIN" W:= DIAG[K]; DIAG[K]:= DIAG[CIK]; DIAG[CIK]:= W
            "END"
        "END"
    "END" LSQDGLINV;
        "EOP"
```

AUTHOR   :   T.J. DEKKER.

CONTRIBUTOR :   J. KOK.

INSTITUTE   :   MATHEMATICAL CENTRE.

RECEIVED   :   731015.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS TWO PROCEDURES :
A) LSQSOL, FOR THE SOLUTION OF A LINEAR LEAST SQUARES PROBLEM IF THE COEFFICIENT MATRIX HAS BEEN DECOMPOSED BY LSQORTDEC (SECTION 3.1.1.2.1.1.);
B) LSQORTDECSOL, FOR THE SOLUTION OF A LINEAR LEAST SQUARES PROBLEM BY HOUSEHOLDER TRIANGULARIZATION WITH COLUMN INTERCHANGES AND FOR THE CALCULATION OF THE DIAGONAL OF THE INVERSE OF M'M, WHERE M IS THE COEFFICIENT MATRIX.

KEY WORDS :

LINEAR LEAST SQUARES PROBLEM,
HOUSEHOLDER TRIANGULARIZATION.

SUBSECTION :   LSQSOL.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" LSQSOL(A, N, M, AID, CI, B); "VALUE" N, M;
"INTEGER" N, M; "INTEGER""ARRAY" CI; "ARRAY" A, AID, B;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

A, N, M, AID, CI : SEE CALLING SEQUENCE OF LSQORTDEC (SECTION 3.1.1.2.1.1.); THE CONTENTS OF THE ARRAYS A, AID AND CI SHOULD BE PRODUCED BY A SUCCESSFUL CALL OF LSQORTDEC, I.E. IF AUX[3] = M;

B : <ARRAY IDENTIFIER>;
"ARRAY" B[1 : N];
ENTRY : B CONTAINS THE RIGHT HAND SIDE OF A LINEAR LEAST SQUARES PROBLEM;
EXIT : B[1 : M] CONTAINS THE SOLUTION OF THE PROBLEM;
B[M + 1 : N] CONTAINS A VECTOR WITH EUCLIDEAN LENGTH EQUAL TO THE EUCLIDEAN LENGTH OF THE RESIDUE VECTOR.

PROCEDURES USED :

```
MATVEC    = CP34011,
TAMVEC    = CP34012,
ELMVECCOL = CP34021.
```

RUNNING TIME:

(C5 * M + C6) * N;
THE CONSTANTS C5 AND C6 DEPEND UPON THE ELEMENTARY ARITHMETIC OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

LSQSOL SHOULD BE CALLED AFTER A SUCCESSFUL CALL OF LSQORTDEC (SECTION 3.1.1.2.1.1.), I.E. IF AUX[3] = M. LSQSOL YIELDS THE LEAST SQUARES SOLUTION OF THE OVERDETERMINED SYSTEM WITH THE DECOMPOSED COEFFICIENT MATRIX IN ARRAY A AND THE RIGHT HAND SIDE IN ARRAY B.
FIRST THE ORTHOGONAL TRANSFORMATION WITH THE HOUSEHOLDER MATRICES IS PERFORMED ON THE RIGHT HAND SIDE. NEXT THE SYSTEM OF THE FIRST M EQUATIONS AND WITH AN UPPER-TRIANGULAR COEFFICIENT MATRIX IS SOLVED BY BACK SUBSTITUTION, YIELDING A SOLUTION WITH M PERMUTED COMPONENTS DUE TO THE COLUMN INTERCHANGES OF THE TRIANGULARIZATION. FINALLY THE ORDER OF THE M COMPONENTS IS RESTORED. SEE ALSO METHOD AND PERFORMANCE OF LSQORTDEC (SECTION 3.1.1.2.1.1.).
THE LEAST SQUARES SOLUTIONS OF SEVERAL OVERDETERMINED SYSTEMS WITH THE SAME COEFFICIENT MATRIX CAN BE SOLVED BY SUCCESSIVE CALLS OF LSQSOL WITH DIFFERENT RIGHT HAND SIDES.

EXAMPLE OF USE :

THE NEXT PROGRAM SOLVES THE SYSTEM

```
- 2 * X1   +    X2  =  0
-     X1   +    X2  =  1
      X1   +    X2  =  2
  2 * X1   +    X2  =  2
      X1 + 2 *  X2  =  3
```

```
"BEGIN""COMMENT" 730912, TEST LSQORTDEC, LSQSOL, LSQDGLINV;
   "ARRAY" A, C[1 : 5,1 : 2], B, X[1 : 5], DIAG, AID[1 : 2],
   AUX[2 : 5];
   "INTEGER""ARRAY" PIV[1 : 2];
   "INTEGER" I, J;
   "REAL" H;

   "REAL""PROCEDURE" VECVEC(L, U, S, A, B);"CODE" 34010;
   "PROCEDURE" LSQORTDEC(A, N, M, AUX, AID, CI);"CODE" 34134;
   "PROCEDURE" LSQSOL(A, N, M, AID, CI, B);"CODE" 34131;
   "PROCEDURE" LSQDGLINV(A, M, AID, CI, DIAG);"CODE" 34132;

   "REAL""PROCEDURE" SUM(I, A, B, X); "VALUE" A, B;
   "INTEGER" I, A, B; "REAL" X;
   "BEGIN""REAL" S; S:= 0; "FOR" I:= A "STEP" 1 "UNTIL" B "DO"
      S:= S + X; SUM:= S
   "END" SUM;

   AUX[2]:= "-12; I:= J:= 1;
   "FOR" H:= - 2, - 1, 1, 2, 1, 1, 1, 1, 1, 2 "DO"
   "BEGIN" A[I,J]:= C[I,J]:= H; "IF" I < 5 "THEN" I:= I + 1 "ELSE"
      "BEGIN" I:= 1; J:= J + 1 "END"
   "END";
   "FOR" H:= 0, 1, 2, 2, 3 "DO"
   "BEGIN" B[I]:= X[I]:= H; I:= I + 1 "END";

   LSQORTDEC(A, 5, 2, AUX, AID, PIV);
   "IF" AUX[3] = 2 "THEN"
   "BEGIN" LSQSOL(A, 5, 2, AID, PIV, X);
      LSQDGLINV(A, 2, AID, PIV, DIAG);
      OUTPUT(61, "("/, "("AUX[2, 3, 5] = ")" +.4D"+DD5B, 3ZD5B,
         +.4D"+DD/, "("LSQ SOLUTION  :")", 2(2B+.8D"+DD), /
         "("RESIDUE (DELIVERED) :")" +.8D"+DD/,
         "("RESIDUE (CHECKED)   :")" +.8D"+DD/,
         "("DIAGONAL OF INVERSE M'M  :")", 2(2B+.8D"+DD)")",
         AUX[2], AUX[3], AUX[5], X[1], X[2],
         SQRT(VECVEC(3, 5, 0, X, X)),
         SQRT(SUM(I, 1, 5, (B[I] - C[I,1] * X[1] - C[I,2] * X[2])
          ** 2)), DIAG[1], DIAG[2])
   "END"
"END"
     "EOP"    END OF PROGRAM
```

DELIVERS :

```
AUX[2, 3, 5] = +.1000"-11        2      +.3317"+01
LSQ SOLUTION  :  +.50000000"+00  +.12500000"+01
RESIDUE (DELIVERED) :+.50000000"+00
RESIDUE (CHECKED)   :+.50000000"+00
DIAGONAL OF INVERSE M'M  :  +.95238095"-01  +.13095238"+00
```

SUBSECTION : LSQORTDECSOL.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" LSQORTDECSOL(A, N, M, AUX, DIAG, B); "VALUE" N, M;
"INTEGER" N, M; "ARRAY" A, AUX, DIAG, B;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

```
    A    :   <ARRAY IDENTIFIER>;
             "ARRAY" A[1 : N,1 : M];
             ENTRY  : A CONTAINS THE COEFFICIENT MATRIX OF THE
             LINEAR LEAST SQUARES PROBLEM;
             EXIT   : IN THE UPPER TRIANGLE OF A (THE ELEMENTS
             A[I,J] WITH I < J) THE SUPERDIAGONAL
             ELEMENTS OF THE UPPER-TRIANGULAR MATRIX, PRODUCED BY
             THE HOUSEHOLDER TRANSFORMATION; IN THE OTHER PART OF
             THE COLUMNS OF A THE SIGNIFICANT ELEMENTS OF THE
             GENERATING VECTORS OF THE HOUSEHOLDER MATRICES USED
             FOR THE HOUSEHOLDER TRIANGULARIZATION;
    N     :  <ARITHMETIC EXPRESSION>;
             NUMBER OF ROWS OF THE MATRIX;
    M     :  <ARITHMETIC EXPRESSION>;
             NUMBER OF COLUMNS OF THE MATRIX (N >= M);
    AUX   :  <ARRAY IDENTIFIER>;
             "ARRAY" AUX[2 : 5];
             ENTRY  : AUX[2] CONTAINS A RELATIVE TOLERANCE USED FOR
             CALCULATING THE DIAGONAL ELEMENTS OF THE
             UPPER-TRIANGULAR MATRIX;
             EXIT  :
             AUX[3] DELIVERS THE NUMBER OF THE DIAGONAL ELEMENTS OF
             THE UPPER-TRIANGULAR MATRIX WHICH ARE FOUND NOT
             NEGLIGIBLE; NORMAL EXIT AUX[3] = M;
             AUX[5] := THE MAXIMUM OF THE EUCLIDEAN NORMS OF THE
             COLUMNS OF THE GIVEN MATRIX;
    DIAG :   <ARRAY IDENTIFIER>;
             "ARRAY" DIAG[1 : M];
             EXIT : THE DIAGONAL ELEMENTS OF THE INVERSE OF M'M
             WHERE M IS THE MATRIX OF THE LINEAR LEAST SQUARES
             PROBLEM;
    B :      <ARRAY IDENTIFIER>;
             "ARRAY" B[1 : N];
             ENTRY : B CONTAINS THE RIGHT HAND SIDE OF A LINEAR
             LEAST SQUARES PROBLEM;
             EXIT : B[1 : M] CONTAINS THE SOLUTION OF THE PROBLEM;
             B[M + 1 : N] CONTAINS A VECTOR WITH EUCLIDEAN LENGTH
             EQUAL TO THE EUCLIDEAN LENGTH OF THE RESIDUE VECTOR.
```

PROCEDURES USED :

```
    LSQORTDEC = CP34134,
    LSQDGLINV = CP34132,
    LSQSOL    = CP34131.
```

REQUIRED CENTRAL MEMORY :

EXECUTION FIELD LENGTH : AN INTEGER ARRAY AND A REAL ARRAY, BOTH OF M ELEMENTS, ARE DECLARED.

RUNNING TIME : ROUGHLY PROPORTIONAL TO N * M ** 2.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

LSQORTDECSOL SOLVES AN OVERDETERMINED SYSTEM OF N LINEAR EQUATIONS IN M UNKNOWNS BY CALLING LSQORTDEC AND, IF THIS CALL WAS SUCCESSFUL LSQDGLINV AND LSQSOL. LSQORTDECSOL DELIVERS THE LEAST SQUARES SOLUTION AND THE DIAGONAL OF THE INVERSE OF M'M, WHERE M IS THE COEFFICIENT MATRIX OF THE SYSTEM. SEE SECTION 3.1.1.2.1.1., AND LSQSOL (THIS SECTION).

EXAMPLE OF USE :

THE PROGRAM

```
"BEGIN""COMMENT" 730914, TEST LSQORTDECSOL;
   "ARRAY" A, C[1 : 5,1 : 2], B, X[1 : 5], DIAG[1 : 2],
   AUX[2 : 5];
   "INTEGER" I, J;
   "REAL" H;
   "REAL""PROCEDURE" VECVEC(L, U, S, A, B);"CODE" 34010;
   "PROCEDURE" LSQORTDECSOL(A, N, M, AUX, DIAG, B);"CODE" 34135;
   "REAL""PROCEDURE" SUM(I, A, B, X); "VALUE" A, B;
   "INTEGER" I, A, B; "REAL" X;
   "BEGIN""REAL" S; S:= 0; "FOR" I:= A "STEP" 1 "UNTIL" B "DO"
      S:= S + X; SUM:= S
   "END" SUM;
   AUX[2]:= "-12; I:= J:= 1;
   "FOR" H:= - 2, - 1, 1, 2, 1, 1, 1, 1, 1, 2 "DO"
   "BEGIN" A[I,J]:= C[I,J]:= H; "IF" I < 5 "THEN" I:= I + 1 "ELSE"
      "BEGIN" I:= 1; J:= J + 1 "END"
   "END";
   "FOR" H:= 0, 1, 2, 2, 3 "DO"
   "BEGIN" B[I]:= X[I]:= H; I:= I + 1 "END";
   LSQORTDECSOL(A, 5, 2, AUX, DIAG, X);
   "IF" AUX[3] = 2 "THEN"
   OUTPUT(61, "("/, "("AUX[2, 3, 5] = ")" +.4D"+DD5B, 3ZD5B,
         +.4D"+DD/, "("LSQ SOLUTION  :")", 2(2B+.8D"+DD), /
         "("RESIDUE (DELIVERED) :")" +.8D"+DD/,
         "("RESIDUE (CHECKED)   :")" +.8D"+DD/,
         "("DIAGONAL OF INVERSE M'M  :")", 2(2B+.8D"+DD)")",
         AUX[2], AUX[3], AUX[5], X[1], X[2],
         SQRT(VECVEC(3, 5, 0, X, X)),
         SQRT(SUM(I, 1, 5, (B[I] - C[I,1] * X[1] - C[I,2] * X[2])
          ** 2)), DIAG[1], DIAG[2])
"END"
     "EOP"    END OF PROGRAM
```

WHICH SOLVES THE PROBLEM OF THE EXAMPLE OF USE OF LSQSOL, DELIVERS :

```
AUX[2, 3, 5] = +.1000"-11          2      +.3317"+01
LSQ SOLUTION  :  +.50000000"+00  +.12500000"+01
RESIDUE (DELIVERED) :+.50000000"+00
RESIDUE (CHECKED)   :+.50000000"+00
DIAGONAL OF INVERSE M'M  :  +.95238095"-01  +.13095238"+00
```

SOURCE TEXT(S) :

```
"CODE" 34131;
    "PROCEDURE" LSQSOL(A, N, M, AID, CI, B); "VALUE" N, M;
    "INTEGER" N, M; "ARRAY" A, AID, B; "INTEGER" "ARRAY" CI;
    "BEGIN" "INTEGER" K, CIK;
        "REAL" W;
        "REAL""PROCEDURE" MATVEC(L, U, I, A, B);"CODE" 34011;
        "REAL""PROCEDURE" TAMVEC(L, U, I, A, B);"CODE" 34012;
        "PROCEDURE" ELMVECCOL(L, U, I, A, B, X);"CODE" 34021;

        "FOR" K:= 1 "STEP" 1 "UNTIL" M "DO" ELMVECCOL(K, N, K, B, A,
        TAMVEC(K, N, K, A, B) / (AID[K] * A[K,K]));
        "FOR" K:= M "STEP" - 1 "UNTIL" 1 "DO" B[K]:= (B[K] - MATVEC
        (K + 1, M, K, A, B)) / AID[K];
        "FOR" K:= M "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" CIK:= CI[K]; "IF" CIK ^= K "THEN"
            "BEGIN" W:= B[K]; B[K]:= B[CIK]; B[CIK]:= W "END"
        "END"
    "END" LSQSOL;
        "EOP"

"CODE" 34135;
    "PROCEDURE" LSQORTDECSOL(A, N, M, AUX, DIAG, B); "VALUE" N, M;
    "INTEGER" N, M; "ARRAY" A, AUX, DIAG, B;
    "BEGIN" "ARRAY" AID[1:M];
        "INTEGER" "ARRAY" CI[1:M];
        "PROCEDURE" LSQORTDEC(A, N, M, AUX, AID, CI);"CODE" 34134;
        "PROCEDURE" LSQDGLINV(A, M, AID, CI, DIAG);"CODE" 34132;
        "PROCEDURE" LSQSOL(A, N, M, AID, CI, B);"CODE" 34131;

        LSQORTDEC(A, N, M, AUX, AID, CI);
        "IF" AUX[3] = M "THEN"
        "BEGIN" LSQDGLINV(A, M, AID, CI, DIAG);
            LSQSOL(A, N, M, AID, CI, B)
        "END"
    "END" LSQORTDECSOL;
        "EOP"
```

CONTRIBUTOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 740617.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS ONE PROCEDURE, LSQINV, FOR THE CALCULATION OF THE INVERSE OF THE MATRIX S'S, WHERE S IS THE COEFFICIENT MATRIX OF A LINEAR LEAST SQUARES PROBLEM.

KEYWORDS :

INVERSE MATRIX,
LEAST SQUARES PROBLEM.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE READS :

```
"PROCEDURE" LSQINV(A, M, AID, C); "VALUE" M; "INTEGER" M;
"ARRAY" A, AID; "INTEGER""ARRAY" C;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

A : <ARRAY IDENTIFIER>;
"ARRAY" A[1 : M, 1 : M];
ENTRY : IN THE UPPER TRIANGLE OF A (THE ELEMENTS A[I,J] WITH 1 <= I < J <= M) THE SUPERDIAGONAL ELEMENTS SHOULD BE GIVEN OF THE UPPERTRIANGULAR MATRIX THAT IS PRODUCED BY THE HOUSEHOLDER TRIANGULARIZATION IN A CALL OF THE PROCEDURE LSQORTDEC (SECTION 3.1.1.2.1.1.) WITH A NORMAL EXIT (AUX[3] = M).
SEE ALSO THE MEANING OF THE PARAMETER AID;
EXIT : THE UPPER TRIANGLE OF THE (SYMMETRIC) INVERSE MATRIX IS DELIVERED IN THE UPPERTRIANGULAR ELEMENTS OF THE ARRAY A (A[I,J] FOR 1 <= I <= J <= M);

M : <ARITHMETIC EXPRESSION>;
NUMBER OF COLUMNS OF THE MATRIX OF THE LINEAR LEAST SQUARES PROBLEM;

AID : <ARRAY IDENTIFIER>;
"ARRAY" AID[1 : M];
ENTRY : AID CONTAINS THE DIAGONAL ELEMENTS OF THE UPPERTRIANGULAR MATRIX THAT IS PRODUCED BY LSQORTDEC;

C : <ARRAY IDENTIFIER>;
"INTEGER""ARRAY" C[1 : M];
ENTRY : C CONTAINS THE PIVOTAL INDICES AS PRODUCED BY A CALL OF LSQORTDEC.

PROCEDURES USED :

CHLINV2 = CP34400.
ICHCOL = CP34031.
ICHROW = CP34032.
ICHROWCOL = CP34033.

REQUIRED CENTRAL MEMORY :

EXECUTION FIELD LENGTH : A REAL ARRAY OF M ELEMENTS IS DECLARED (IN THE CALL OF CHLINV2).

RUNNING TIME : PROPORTIONAL TO M CUBED.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

LSQINV SHOULD BE CALLED AFTER A SUCCESSFUL CALL OF LSQORTDEC (SECTION 3.1.1.2.1.1.). LSQINV CAN BE USED FOR THE CALCULATION OF THE COVARIANCE MATRIX OF A LINEAR LEAST SQUARES PROBLEM.
LET S BE THE MATRIX OF THE LEAST SQUARES SYSTEM WITH PERMUTED COLUMNS AND Q * R THE HOUSEHOLDER DECOMPOSITION OF S. THEN THE INVERSE OF S'S ALSO IS THE INVERSE OF R'R. SINCE R IS THE CHOLESKY MATRIX OF R'R, THE INVERSE MATRIX IS COMPUTED IN A CALL OF CHLINV2 (SECTION 3.1.1.1.1.2.4.). AFTERWARDS THE COVARIANCE MATRIX IS OBTAINED BY INTERCHANGES OF THE COLUMNS AND ROWS OF THE INVERSE MATRIX.

EXAMPLE OF USE :

THE FOLLOWING PROGRAM COMPUTES THE INVERSE T OF S'S, WHERE S IS THE COEFFICIENT MATRIX OF THE SYSTEM IN THE EXAMPLE OF USE OF LSQORTDEC AND LSQDGLINV (SECTION 3.1.1.2.1.1.). THE DIAGONAL OF S CAN BE COMPARED WITH THE RESULT OF LSQDGLINV. TO CHECK THE ANSWERS S' * (S * T) IS PRINTED.

```
"BEGIN""COMMENT" JKOK, 740530, EXAMPLE OF USE OF  LSQORTDEC AND
    LSQINV;

    "ARRAY" A, C[1 : 5, 1 : 2], AID[1 : 2], T[1 : 2, 1 : 2],
    AUX[2 : 5];
    "INTEGER""ARRAY" PIV[1 : 2];
    "INTEGER" I, J; "REAL" H;

    "REAL""PROCEDURE" MATMAT(L, U, I, J, A, B); "CODE" 34013;
    "REAL""PROCEDURE" TAMMAT(L, U, I, J, A, B); "CODE" 34014;
    "PROCEDURE" LSQORTDEC(A, N, M, AUX, AID, CI); "CODE" 34134;
    "PROCEDURE" LSQINV(A, M, AID, CI); "CODE" 34136;

    AUX[2]:= "-12; I:= J:= 1;
    "FOR" H:= - 2, - 1, 1, 2, 1, 1, 1, 1, 1, 2 "DO"
    "BEGIN" A[I,J]:= C[I,J]:= H; "IF" I < 5 "THEN" I:= I + 1 "ELSE"
        "BEGIN" I:= 1; J:= J + 1 "END"
    "END";

    LSQORTDEC(A, 5, 2, AUX, AID, PIV); "IF" AUX[3] = 2 "THEN"
    "BEGIN" LSQINV(A, 2, AID, PIV);
        T[1,1]:= A[1,1]; T[2,2]:= A[2,2]; T[2,1]:= T[1,2]:= A[1,2];
        "FOR" J:= 1, 2 "DO""FOR" I:= 1 "STEP" 1 "UNTIL" 5 "DO"
        A[I,J]:= MATMAT(1, 2, I, J, C, T);
        OUTPUT(61, "("/4B, "(" AUX[2, 3, 5] = ")",
            -.4D"+DD5B, 37D5B, +.4D"+DD/,
            2(/4B, 30S, /, 2(/4B, 2(2B+.8D"+DD)), /) ")",
            AUX[2], AUX[3], AUX[5],
            "(" INVERSE :")", ((T[I,J], J:= 1 : 2), I:= 1 : 2),
            "(" CHECK : S' * (S * T) :")",
            ((TAMMAT(1, 5, I, J, C, A), J:= 1 : 2), I:= 1 : 2) )
    "END"
"END"
    "EOP"    END OF PROGRAM

OUTPUT  :

 AUX[2, 3, 5] =  .1000"-11         2      +.3317"+01

 INVERSE :

  +.95238095"-01  -.23809524"-01
  -.23809524"-01  +.13095238"+00

 CHECK : S' * (S * T) :

  +.10000000"+01  +.17763568"-14
  +.00000000"+00  +.10000000"+01
```

SOURCE TEXT(S):

```
"CODE" 34136;
     "PROCEDURE" LSQINV(A, M, AID, C); "VALUE" M; "INTEGER" M;
     "ARRAY" A, AID; "INTEGER""ARRAY" C;
     "BEGIN""INTEGER" I, CI;
        "REAL" W;

        "PROCEDURE" CHLINV2(A, N); "CODE" 34400;
        "PROCEDURE" ICHCOL(L, U, I, J, A); "CODE" 34031;
        "PROCEDURE" ICHROW(L, U, I, J, A); "CODE" 34032;
        "PROCEDURE" ICHROWCOL(L, U, I, J, A); "CODE" 34033;

        "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO" A[I,I]:= AID[I];
        CHLINV2(A, M);
        "FOR" I:= M "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" CI:= C[I]; "IF" CI ^= I "THEN"
           "BEGIN" ICHCOL(1, I - 1, I, CI, A); ICHROWCOL(I + 1, CI - 1,
              I, CI, A); ICHROW(CI + 1, M, I, CI, A);
              W:= A[I,I]; A[I,I]:= A[CI,CI]; A[CI,CI]:= W
           "END"
        "END"
     "END" LSQINV;
         "EOP"
```

AUTHORS: A.BJOERCK AND G.H.GOLUB.

CONTRIBUTOR: J.KOOPMAN.

INSTITUTE: UNIVERSITY OF AMSTERDAM.

RECEIVED: 780701 .

BRIEF DESCRIPTION:

```
    THIS SECTION CONTAINS TWO PROCEDURES FOR SOLVING A LINEAR LEAST
    SQUARES PROBLEM SUBJECT TO LINEAR CONSTRAINTS:
    LSQDECOMP , FOR THE QR-DECOMPOSITION OF A LEAST SQUARES MATRIX,
                WHERE  THIS MATRIX ALSO CONTAINS THE COEFFICIENTS OF
                THE LINEAR CONSTRAINTS (LINEAR EQUATIONS);
    LSQREFSOL , FOR THE SOLUTION OF THIS LEAST SQUARES PROBLEM, IF
                THE MATRIX HAS BEEN DECOMPOSED BY LSQDECOMP.
```

KEYWORDS:

LEAST SQUARES,
LINEAR CONSTRAINTS,
HOUSEHOLDER TRIANGULARIZATION,
ITERATIVE REFINEMENT.

SUBSECTION: LSQDECOMP.

CALLING SEQUENCE:

```
    THE HEADING OF THE PROCEDURE READS:
    "PROCEDURE" LSQDECOMP(A, N, M, N1, AUX, AID, CI);
    "VALUE"N,M,N1;"INTEGER"N,M,N1;"ARRAY" A,AUX,AID;
    "INTEGER""ARRAY" CI;
    "CODE" 34137;
```

```
THE MEANING OF THE FORMAL PARAMETERS IS:

A       :<ARRAY IDENTIFIER>;
         "ARRAY" A[1:N,1:M];
         ENTRY: THE ORIGINAL LEAST SQUARES MATRIX, WHERE THE FIRST
                N1 ROWS SHOULD FORM THE CONSTRAINT MATRIX (I.E. THE
                FIRST N1 EQUATIONS ARE TO BE STRICTLY SATISFIED);
         EXIT : IN THE UPPER TRIANGLE OF A (THE ELEMENTS A[I,J] WITH
                I<J) THE SUPERDIAGONAL PART OF THE UPPER TRIANGULAR
                MATRIX, PRODUCED BY HOUSEHOLDER TRANSFORMATIONS; IN
                THE OTHER PART OF THE COLUMNS OF A THE SIGNIFICANT
                ELEMENTS OF THE GENERATING VECTORS OF THE HOUSEHOLDER
                MATRICES USED FOR THE TRIANGULARIZATION;
N       :<ARITHMETIC EXPRESSION>;
         NUMBER OF ROWS OF THE MATRIX;
M       :<ARITHMETIC EXPRESSION>;
         NUMBER OF COLUMNS OF THE MATRIX;
N1      :<ARITHMETIC EXPRESSION>;
         NUMBER OF LINEAR CONSTRAINTS, I.E. THE FIRST N1 ROWS OF A
         SET UP A SYSTEM OF N1 LINEAR EQUATIONS THAT MUST BE
         STRICTLY SATISFIED (OF COURSE, IF THERE ARE NO CON-
         STRAINTS, N1 MUST BE CHOSEN ZERO);
AUX     :<ARRAY IDENTIFIER>;
         "ARRAY" AUX[2:7];
         ENTRY: AUX[2] CONTAINS A RELATIVE TOLERANCE FOR CALCULATING
                THE DIAGONAL ELEMENTS OF THE UPPER TRIANGULAR MATRIX;
         EXIT:  AUX[3] CONTAINS THE NUMBER OF DIAGONAL ELEMENTS WHICH
                ARE NOT NEGLIGIBLE, NORMAL EXIT AUX[3]=M;
AID     :<ARRAY IDENTIFIER>;
         "ARRAY" AID[1:M];
         NORMAL EXIT (AUX[3]=M): THE DIAGONAL ELEMENTS OF THE UPPER
         TRIANGULAR MATRIX PRODUCED BY THE HOUSEHOLDER TRANSFORMATION
CI      :<ARRAY IDENTIFIER>;
         "INTEGER""ARRAY" CI[1:M];
         EXIT: THE PIVOTAL INDICES OF THE INTERCHANGES OF THE COLUMNS
         OF THE GIVEN MATRIX;
```

PROCEDURES USED:

```
MATMAT = CP34013.
TAMMAT = CP34014.
ELMCOL = CP34023.
ICHCOL = CP34031.
```

RUNNING TIME:

ROUGHLY PROPORTIONAL TO N*M**2-M**3/3.

METHOD AND PERFORMANCE:

LET A DENOTE THE GIVEN MATRIX. LSQDECOMP PRODUCES AN N-TH ORDER ORTHOGONAL MATRIX Q AND AN N*M UPPER TRIANGULAR MATRIX R SUCH THAT R EQUALS QA WITH PERMUTED COLUMNS.
THE ORTHOGONAL MATRIX Q IS FORMED AS A PRODUCT OF AT MOST M TRANSFORMATIONS OF THE FORM (I-BETA U U'). THESE HOUSEHOLDER MATRICES REDUCE A TO THE MATRIX R: AT THE K-TH STAGE THE K-TH COLUMN OF THE (ALREADY MODIFIED) MATRIX IS INTERCHANGED WITH THE COLUMN OF MAXIMUM EUCLIDEAN NORM.THESE INTERCHANGES ARE RECORDED IN THE ARRAY CI. PREMATURE TERMINATION OCCURS IF AT SOME STAGE THE EUCLIDEAN NORM OF THE PIVOTAL COLUMN IS LESS THAN SOME TOLERANCE (AUX[2]) TIMES THE MAXIMUM OF THE EUCLIDEAN NORMS OF THE COLUMNS OF THE MATRIX.
LSQDECOMP DELIVERS THE UPPER TRIANGULAR MATRIX, WHERE THE DIAGONAL ELEMENTS ARE STORED IN THE ARRAY AID AND THE OFF-DIAGONAL ELEMENTS IN THE UPPER TRIANGULAR PART OF A. THE SIGNIFICANT ELEMENTS OF THE GENERATING VECTORS OF THE HOUSEHOLDER TRANSFORMATIONS ARE STORED IN THE COLUMNS OF A. I.E. ON AND BELOW THE LEADING DIAGONAL OF A. IT SHOULD BE NOTED THAT FOR THE SOLUTION OF LEAST SQUARES PROBLEMS. ONLY CALLS WITH N>=M ARE USEFUL.

REFERENCES:

A.BJOERCK AND G.H.GOLUB:
ITERATIVE REFINEMENT OF LEAST SQUARES SOLUTIONS BY HOUSEHOLDER TRANSFORMATION. BIT 7 (1967), PP. 322-337

SUBSECTION: LSQREFSOL.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE READS:

```
"PROCEDURE" LSQREFSOL(A, QR, N, M, N1, AUX,AID,CI,B,LDX,X,RES);
"VALUE"N,M,N1;"INTEGER"N,M,N1;"INTEGER""ARRAY"CI;"REAL" LDX;
"ARRAY"A,QR,AUX,AID,B,X,RES;
"CODE" 34138;
```

```
THE MEANING OF THE FORMAL PARAMETERS IS:
A      :<ARRAY IDENTIFIER>;
        "ARRAY" A[1:N,1:M];
        THE ORIGINAL LEAST SQUARES MATRIX, WHERE THE FIRST N1
        ROWS SET UP A SYSTEM OF LINEAR EQUATIONS THAT MUST BE
        STRICTLY SATISFIED;
QR     :<ARRAY IDENTIFIER>;
        "ARRAY" QR[1:N,1:M];
        THE QR-DECOMPOSITION OF THE ORIGINAL LEAST SQUARES
        MATRIX AS DELIVERED BY A SUCCESSFUL CALL OF LSQDECOMP;
N      :<ARITHMETIC EXPRESSION>;
        NUMBER OF ROWS OF THE MATRICES A AND QR;
M      :<ARITHMETIC EXPRESSION>;
        NUMBER OF COLUMNS OF THE MATRICES A AND QR;
N1     :<ARITHMETIC EXPRESSION>;
        NUMBER OF LINEAR CONSTRAINTS;
AUX    :<ARRAY IDENTIFIER>;
        "ARRAY" AUX[2:7];
        ENTRY: AUX[2] CONTAINS A RELATIVE TOLERANCE AS A
               CRITERION TO STOP ITERATIVE REFINING: IF THE
               EUCLIDEAN NORM OF THE CORRECTION IS SMALLER
               THAN AUX[2] TIMES THE CURRENT APPROXIMATION OF
               THE SOLUTION, THE ITERATIVE REFINING IS STOPPED;
               AUX[6]: MAXIMUM NUMBER OF ITERATIONS ALLOWED
               (USUALLY AUX[6]=5 WILL BE SUFFICIENT);
        EXIT : AUX[7]: THE NUMBER OF ITERATIONS PERFORMED;
AID    :<ARRAY IDENTIFIER>;
        "ARRAY" AID[1:M];
        THE DIAGONAL ELEMENTS OF THE UPPER TRIANGULAR MATRIX
        AS DELIVERED BY A SUCCESSFUL CALL OF LSQDECOMP;
CI     :<ARRAY IDENTIFIER>;
        "INTEGER""ARRAY" CI[1:M];
        THE PIVOTAL INDICES AS PRODUCED BY LSQDECOMP;
B      :<ARRAY IDENTIFIER>;
        "ARRAY" B[1:N];
        THE RIGHT-HAND SIDE OF THE LEAST SQUARES PROBLEM;
        FIRST N1 ELEMENTS FORM THE RIGHT HAND SIDES OF THE
        CONSTRAINTS;
LDX    :<REAL VARIABLE>;
        THE EUCLIDEAN NORM OF THE LAST CORRECTION OF THE
        SOLUTION;
X      :<ARRAY IDENTIFIER>;
        "ARRAY" X[1:M];
        EXIT: THE SOLUTION VECTOR;
RES    :<ARRAY IDENTIFIER>;
        "ARRAY" RES[1:N];
        EXIT: THE RESIDUAL VECTOR CORRESPONDING TO THE SOLUTION;
```

PROCEDURES USED:

VECVEC = CP34010.
MATVEC = CP34011.
TAMVEC = CP34012.
ELMVECCOL = CP34021.
ICHCOL = CP34031.
LNG SUB = CP31106.
LNGMATVEC = CP34411.
LNGTAMVEC = CP34412.

RUNNING TIME:

ROUGHLY PROPORTIONAL TO C1*M*N-C2*M**2, WHERE C1 AND C2 ARE CONSTANTS.

METHOD AND PERFORMANCE:

LSQREFSOL SHOULD BE CALLED AFTER A SUCCESSFUL CALL OF LSQDECOMP (I.E. AUX[3]=M). LSQREFSOL YIELDS THE LEAST SQUARES SOLUTION OF THE OVERDETERMINED SYSTEM WITH THE DECOMPOSED COEFFICIENT MATRIX IN THE ARRAY A AND THE RIGHT-HAND SIDE IN ARRAY B. THE ORIGINAL LEAST SQUARES MATRIX ALSO CONTAINS THE LINEAR CONSTRAINTS (THE FIRST N1 ROWS OF THIS MATRIX SET UP A SYSTEM THAT MUST BE STRICTLY SATISFIED). FIRST, THE ORTHOGONAL TRANSFORMATION WITH THE HOUSEHOLDER MATRICES IS PERFORMED ON THE RIGHT-HAND SIDE. NEXT THE SYSTEM IS SOLVED BY MEANS OF A BACK SUBSTITUTION. IN THIS WAY THE FIRST APPROXIMATION OF THE SOLUTION (WITH PERMUTED COLUMNS) IS OBTAINED.
AFTER THIS AN ITERATIVE PROCESS REFINES THE APPROXIMATION UNTIL THE EUCLIDEAN NORM OF THE CORRECTION VECTOR IS NEGLIGIBLY SMALL COMPARED TO THE APPROXIMATION OR UNTIL THE MAXIMUM NUMBER OF ITERATIONS IS REACHED. FOR A MORE DETAILED DESCRIPTION OF THE ITERATIVE IMPROVEMENT, SEE REF[1].
AFTER THE ITERATIVE PROCESS AN APPROXIMATION TO THE SOLUTION IS FOUND. HOWEVER, THE ORDER OF THE COMPONENTS POSSIBLY IS NOT CORRECT. THEREFORE THIS ORDER IS RESTORED AT THE END OF THE PROCEDURE (SEE ALSO METHOD AND PERFORMANCE OF LSQDECOMP).
THE LEAST SQUARES SOLUTIONS OF SEVERAL OVERDETERMINED SYSTEMS WITH THE SAME CONSTRAINTS AND COEFFICIENT MATRIX CAN BE SOLVED BY SUCCESSIVE CALLS OF LSQREFSOL WITH DIFFERENT RIGHT-HAND SIDES.

REFERENCES:

[1] A.BJOERCK AND G.H.GOLUB:
ITERATIVE REFINEMENT OF LEAST SQUARES SOLUTIONS BY HOUSEHOLDER TRANSFORMATION, BIT 7 (1967), PP. 322-337.

EXAMPLE OF USE:

THE FOLLOWING PROGRAM SOLVES THE PROBLEM:

MINIMIZE //B - A*X//

UNDER X1 + 1000*X2 + 5*X3 = 2016 ,WHERE
A IS THE MATRIX

```
 1    0    8
 0    3    2
 1    2    "-5
 0    0    0
```

AND THE VECTOR B = (25, 12, 5.00003, 1)' , X = (X1, X2, X3)' ;

```
"BEGIN" "INTEGER" N,M,N1,I,J;
        N := 5; M := 3; N1 := 1;
        "BEGIN""INTEGER""ARRAY" CI[1:M];
            "ARRAY" AUX[2:7],QR,A[1:N,1:M],B,RES[1:N],AID,X[1:M];
            "REAL" LDX;
            "PROCEDURE" LSQDECOMP(A,N,M,N1,AUX,AID,CI);"CODE"34137;
            "PROCEDURE" LSQREFSOL(A,QR,N,M,N1,AUX,AID,CI,B,LDX,X,RES);
            "CODE" 34138;
            A[1,1] := 1; A[1,2] := 1000; A[1,3] := 5;
            A[2,1] := 1; A[2,2] := 0; A[2,3] := 8;
            A[3,1] := 0; A[3,2] := 3; A[3,3] := 2;
            A[4,1] := 1; A[4,2] := 2; A[4,3] := "-5;
            A[5,1]:=A[5,2]:=A[5,3]:=0;
            B[1] := 2016; B[2] :=25; B[3] :=12; B[4] := 5.00003;
            B[5] := 1;  AUX[2] := "-14; AUX[6] := 5;
            "FOR" I:=1 "STEP" 1 "UNTIL" N "DO"
            "FOR" J:=1 "STEP" 1 "UNTIL" M "DO" QR[I,J] := A[I,J];

            LSQDECOMP(QR,N,M,N1,AUX,AID,CI);
            LSQREFSOL(A,QR,N,M,N1,AUX,AID,CI,B,LDX,X,RES);

OUTPUT(61,"("("THE SOLUTION VECTOR: ")",//")");
"FOR" I:=1 "STEP" 1 "UNTIL" M "DO" OUTPUT(61,"("/")",X[I]);
OUTPUT(61,"("//,"("THE RESIDUAL VECTOR: ")",//")");
"FOR" J:=N1+1 "STEP" 1 "UNTIL" N "DO" OUTPUT(61,"("/")",RES[J]);
OUTPUT(61,"("///,"("NUMBER OF ITERATIONS: ")",D,//,
        "("NORM LAST CORRECTION OF X: ")",N")",AUX[7],LDX)
        "END"
END"
```

DELIVERS:

THE SOLUTION VECTOR:

```
+1.0000000000000"+000
+2.0000000000000"+000
+3.0000000000000"+000
```

THE RESIDUAL VECTOR:

```
-5.2734444477081"-016
+2.1280091641666"-015
+5.3479806840033"-016
+1.0000000000000"+000
```

NUMBER OF ITERATIONS: 2

NORM LAST CORRECTION OF X: +2.1657844626990"-015

SOURCE TEXT(S):

```
"CODE" 34137;
"PROCEDURE" LSQDECOMP( A, N ,M ,N1 ,AUX ,AID ,CI );
"VALUE" N , M ,N1;"INTEGER" N,M,N1;"ARRAY" A,AUX, AID;
"INTEGER""ARRAY" CI;
"BEGIN""INTEGER"J,K,KPIV,NR,S;"BOOLEAN" FSUM;
    "REAL" BETA,SIGMA,NORM,AIDK,AKK,W,EPS;
    "ARRAY" SUM[1:M];
    "REAL""PROCEDURE" MATMAT(L ,U ,I ,J ,A ,B );"CODE" 34013;
    "REAL""PROCEDURE" TAMMAT(L ,U ,I ,J ,A ,B );"CODE" 34014;
    "PROCEDURE" ELMCOL(L ,U ,I ,J ,A ,B ,X );"CODE" 34023;
    "PROCEDURE" ICHCOL(L ,U ,I ,J ,A );"CODE" 34031;

    NORM:=0 ; AUX[3]:=M;NR:=N1;FSUM:="TRUE";
    "FOR" K:=1 "STEP" 1 "UNTIL" M "DO"
        "BEGIN" "IF" K=N1+1 "THEN" "BEGIN" FSUM:="TRUE"; NR:=N "END";
        "IF" FSUM "THEN"
        "FOR" J:=K "STEP" 1 "UNTIL" M "DO"
        "BEGIN" W:=SUM[J]:= TAMMAT(K ,NR ,J ,J ,A ,A);
                "IF" W>NORM "THEN" NORM:=W
        "END"; FSUM:="FALSE";EPS:=AUX[2]*SQRT(NORM);
        SIGMA:=SUM[K]; KPIV:=K;
        "FOR" J:=K+1 "STEP" 1 "UNTIL" M "DO"
        "IF" SUM[J]>SIGMA "THEN"
        "BEGIN" SIGMA:=SUM[J]; KPIV:=J "END";
                                                    "COMMENT"
```

```
        "IF" KPIV^=K "THEN"
        "BEGIN" SUM[KPIV]:=SUM[K]; ICHCOL( 1 ,N ,K ,KPIV ,A) "END";
        CI[K]:=KPIV; AKK:=A[K,K];
        SIGMA:=TAMMAT(K ,NR ,K ,K ,A ,A); W:=SQRT(SIGMA);
        AIDK:=AID[K]:="IF" AKK<0 "THEN" W "ELSE" -W;
        "IF" W<EPS "THEN"
        "BEGIN" AUX[3]:=K-1;"GOTO" ENDDEC "END";
        BETA:= 1/(SIGMA-AKK*AIDK); A[K,K]:=AKK-AIDK;
        "FOR" J:=K+1 "STEP" 1 "UNTIL" M "DO"
        "BEGIN" ELMCOL(K ,NR ,J ,K ,A ,A ,-BETA*TAMMAT(K ,NR ,
        K ,J ,A ,A)); SUM[J]:=SUM[J]-A[K,J]**2
        "END";
        "IF" K=N1 "THEN"
        "FOR" J:=N1+1 "STEP" 1 "UNTIL" N "DO"
        "FOR" S:=1 "STEP" 1 "UNTIL" M "DO"
        "BEGIN" NR:="IF" S>N1 "THEN" N1 "ELSE" S-1;
        W:=A[J,S]-MATMAT(1 ,NR , J ,S ,A ,A);
        A[J,S]:="IF" S>N1 "THEN" W "ELSE" W/AID[S]
        "END"
    "END" FOR K;
  ENDDEC:
"END" LSQDECOMP;
        "EOP"
"CODE" 34138;
"PROCEDURE" LSQREFSOL(A, QR, N, M, N1, AUX, AID, CI, B,LDX,X,RES);
"VALUE"N,M,N1;"INTEGER"N,M,N1;"INTEGER""ARRAY"CI;"REAL"LDX;
"ARRAY"QR,A,AID,AUX,X,RES,B;
"BEGIN""INTEGER"I,J,K,S;
        "REAL"C1,NEXVE,NDX,NDR,D,DD,OP,OPL,CORRNORM;
        "ARRAY"F[1:N],G[1:M];
        "REAL""PROCEDURE"VECVEC(L,U,S,A,B);"CODE"34010;
        "REAL""PROCEDURE"MATVEC(L,U,S,A,B);"CODE"34011;
        "REAL""PROCEDURE"TAMVEC(L,U,S,A,B);"CODE"34012;
        "PROCEDURE" ICHCOL(L,U,I,J,A);"CODE" 34031;
        "PROCEDURE"LNG SUB(A,AA,B,BB,C,CC);"CODE"31106;
        "PROCEDURE"ELMVECCOL(L,U,I,A,B,X);"CODE"34021;
        "PROCEDURE"LNGTAMVEC(L,U,I,A,B,C,CC,D,DD);"CODE"34412;
        "PROCEDURE"LNGMATVEC(L,U,I,A,B,C,CC,D,DD);"CODE"34411;
        "PROCEDURE" HOUSEHOLDER(P, Q, R, E);
        "VALUE" P,Q,R,E;"INTEGER" P,Q,R,E;
        "BEGIN" "FOR" S:=P "STEP" Q "UNTIL" R "DO"
                ELMVECCOL(S, E, S, F, QR,TAMVEC(S, E, S,QR, F)/(QR[S,S]*
                AID[S]))
        "END";
        "FOR" J:=1 "STEP" 1 "UNTIL" M "DO"
        "BEGIN" S:=CI[J];"IF" S^=J"THEN" ICHCOL(1,N,J,S,A) "END";
        "FOR"J:=1"STEP"1"UNTIL"M"DO"X[J]:=G[J]:=0;
        "FOR"I:=1"STEP"1"UNTIL"N"DO"
        "BEGIN"RES[I]:=0;F[I]:=B[I]"END";
                                                            "COMMENT"
```

```
    "FOR"K:=0,1,K+1
       "WHILE" (CORRNORM>AUX[2]*NEXVE & K<=AUX[6])
    "DO"
    "BEGIN"NDX:=NDR:=0;
       "IF"K^=0"THEN"
       "BEGIN""FOR"I:=1"STEP"1"UNTIL"N"DO"RES[I]:=RES[I]+F[I];
              "FOR"S:=1"STEP"1"UNTIL"M"DO"
              "BEGIN" X[S]:=X[S]+G[S];
                      LNGTAMVEC(1,N,S,A,RES,0,0,D,DD);
                      G[S]:=(-D-TAMVEC(1,S-1,S,QR,G))/AID[S]
              "END";
              "FOR"I:=1"STEP"1"UNTIL"N"DO"
              "BEGIN" LNGMATVEC(1, M, I, A, X,
                       "IF" I>N1 "THEN" RES[I] "ELSE" 0, 0, D, DD);
                      LNG SUB(B[I],0,D,DD,OP,OPL);
                      F[I]:=OP
              "END"
       "END";
       NEXVE:=SQRT(VECVEC(1,M,0,X,X)+VECVEC(1,N,0,RES,RES));
       HOUSEHOLDER(1, 1, N1, N1);
       "FOR" I:=N1+1 "STEP" 1 "UNTIL" N "DO"
       F[I]:=F[I]-MATVEC(1,N1, I, QR, F);
       HOUSEHOLDER(N1+1, 1, M, N);
       "FOR" I:=1 "STEP" 1 "UNTIL" M "DO"
       "BEGIN" C1:=F[I];F[I]:=G[I];
           G[I]:="IF" I>N1 "THEN" C1-G[I] "ELSE" C1
       "END";
       "FOR"S:=M"STEP"-1"UNTIL"1"DO"
       "BEGIN"G[S]:=(G[S]-MATVEC(S+1,M,S,QR,G))/AID[S];
              NDX:=NDX+G[S]**2
       "END";
       HOUSEHOLDER(M, -1, N1+1, N);
       "FOR" S:=1 "STEP" 1 "UNTIL" N1 "DO"
       F[S]:=F[S]-TAMVEC(N1+1, N, S, QR, F);
       HOUSEHOLDER(N1, -1, 1, N1);
       AUX[7]:=K;
       "FOR"I:=1"STEP"1"UNTIL"N"DO"NDR:=NDR+F[I]**2;
       CORRNORM:=SQRT(NDX+NDR)
    "END"FOR K;
    LDX:=SQRT(NDX);
    "FOR" S:=M "STEP" -1 "UNTIL" 1 "DO"
    "BEGIN" J:=CI[S];"IF" J^=S "THEN"
       "BEGIN" C1:=X[J];X[J]:=X[S];X[S]:=C1;ICHCOL(1,N,J,S,A)"END"
    "END"
"END" LSQREFSOL;
```

AUTHOR : D.T.WINTER

INSTITUTE : MATHEMATICAL CENTRE

RECEIVED : 731217

BRIEF DESCRIPTION :

THIS SECTION CONTAINS 2 PROCEDURES FOR THE SOLUTION OF AN OVERDETERMINED SYSTEM OF LINEAR EQUATIONS:
SOLSVDOVR SOLVES AN OVERDETERMINED SYSTEM OF LINEAR EQUATIONS , MULTIPLYING THE RIGHT-HAND SIDE BY THE PSEUDO-INVERSE OF THE GIVEN MATRIX; THE SINGULAR VALUES DECOMPOSITION SHOULD BE AVAILABLE.
SOLOVR CALCULATES THE SINGULAR VALUES DECOMPOSITION AND SOLVES AN OVERDETERMINED SYSTEM OF LINEAR EQUATIONS.

KEYWORDS :
BEST LEAST-SQUARES SOLUTION
SINGULAR VALUES
PSEUDO-INVERSE

SUBSECTION : SOLSVDOVR

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :
"PROCEDURE" SOLSVDOVR(U, VAL, V, M, N, X, EM);
"VALUE" M,N; "INTEGER" M,N; "ARRAY" U, VAL, V, X, EM; "CODE" 34280;

THE MEANING OF THE FORMAL PARAMETERS IS :
U: <ARRAY IDENTIFIER>;
    "ARRAY" U[1:M,1:N];
    ENTRY:THE MATRIX U IN THE SINGULAR VALUES DECOMPOSITION U*S*V'.
VAL: <ARRAY IDENTIFIER>;
    "ARRAY" VAL[1:N];
    ENTRY:THE SINGULAR VALUES.
V: <ARRAY IDENTIFIER>;
    "ARRAY" V[1:N,1:N];
    ENTRY:THE MATRIX V IN THE SINGULAR VALUES DECOMPOSITION.
N: <ARITHMETIC EXPRESSION>;
    THE NUMBER OF UNKNOWNS.
M: <ARITHMETIC EXPRESSION>;
    THE LENGTH OF THE RIGHT-HAND SIDE VECTOR, N SHOULD SATISFY
    N <= M.
X: <ARRAY IDENTIFIER>;
    "ARRAY" X[1:M];
    ENTRY: THE RIGHT-HAND SIDE VECTOR;
    EXIT: THE SOLUTION VECTOR IN X[1:N].
EM: <ARRAY IDENTIFIER>;
    "ARRAY" EM[6:6];
    ENTRY: EM[6]: THE MINIMAL NON-NEGLECTABLE SINGULAR VALUE.

```
PROCEDURES USED :
    MATVEC = CP34011
    TAMVEC = CP34012

REQUIRED CENTRAL MEMORY : AN AUXILIARY ARRAY OF N REALS IS DECLARED.

RUNNING TIME : ROUGHLY PROPORTIONAL TO (M + N) * N

METHOD AND PERFORMANCE :

    THE SOLUTION IS FOUND IN THREE STEPS :
    1.  U' * X = X1 IS CALCULATED,
    2.  VAL+ * X1 = X2  IS CALCULATED,  HERE VAL+  DENOTES THE DIAGONAL
        MATRIX OBTAINED FROM VAL BY SETTING VAL+[I,I] = 1/VAL[I]  IF
        VAL[I] GREATER THAN OR EQUAL TO EM[6], AND 0 OTHERWISE,
    3.  THE SOLUTION V * X2 IS CALCULATED.


SUBSECTION : SOLOVR

CALLING SEQUENCE :

    THE HEADING OF THE PROCEDURE IS :
    "INTEGER" "PROCEDURE" SOLOVR(A, M, N, X, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, X, EM; "CODE" 34281;

    SOLOVR:= THE NUMBER OF SINGULAR VALUES NOT FOUND,  I.E. ZERO IF ALL
        SINGULAR VALUES ARE CALCULATED.

    THE MEANING OF THE FORMAL PAREMETERS IS :
    A:  <ARRAY IDENTIFIER>;
        "ARRAY" A[1:M,1:N];
        ENTRY: THE MATRIX OF THE SYSTEM;
    M:  <ARITHMETIC EXPRESSION>;
        THE NUMBER OF ROWS OF A;
    N:  <ARITHMETIC EXPRESSION>;
        THE NUMBER OF COLUMNS OF A, N <= M;
    X:  <ARRAY IDENTIFIER>;
        "ARRAY" X[1:M];
        ENTRY: THE RIGHT-HAND SIDE VECTOR;
        EXIT: THE SOLUTION VECTOR IN X[1:N];
    EM: <ARRAY IDENTIFIER>;
        "ARRAY" EM[0:7];
        ENTRY: EM[0]: THE MACHINE PRECISION;
            EM[2]: THE RELATIVE PRECISION OF THE SINGULAR VALUES;
            EM[4]: THE MAXIMAL NUMBER OF ITERATIONS TO BE PERFORMED IN
                   THE SINGULAR VALUES DECOMPOSITION;
            EM[6]: THE MINIMAL NON-NEGLECTABLE SINGULAR VALUE;
        EXIT:EM[1]: THE INFINITY NORM OF THE MATRIX;
            EM[3]: THE MAXIMAL NEGLECTED SUPERDIAGONAL ELEMENT;
            EM[5]: THE NUMBER OF ITERATIONS  PERFORMED IN THE SINGULAR
                   VALUES DECOMPOSITION;
            EM[7]: THE NUMERICAL RANK OF THE MATRIX, I.E. THE NUMBER OF
                   SINGULAR VALUES GREATER THAN OR EQUAL TO EM[6].
```

PROCEDURES USED :

    QRISNGVALDEC = CP34273
    SOLSVDOVR   = CP34280

REQUIRED CENTRAL MEMORY :

    AUXILIARY ARRAYS ARE DECLARED TO A TOTAL OF (N + 2) * N REALS

RUNNING TIME : ROUGHLY PROPORTIONAL TO (M + N) * N * N

METHOD AND PERFORMANCE :

THE SOLUTION IS FOUND IN TWO STEPS :

1. THE SINGULAR VALUES DECOMPOSITION IS CALCULATED BY MEANS OF THE PROCEDURE QRISNGVALDEC (SECTION 3.5.1.2);
2. THE SOLUTION IS CALCULATED BY MEANS OF THE PROCEDURE SOLSVDOVR, (THIS SECTION);

REFERENCES :

WILKINSON, J.H. AND C.REINSCH
HANDBOOK OF AUTOMATIC COMPUTATION, VOL. 2 (CONTRIBUTION I-10)
LINEAR ALGEBRA
HEIDELBERG (1971)

EXAMPLE OF USE :

FIRST A PROGRAM IS GIVEN, AND THEN THE RESULTS OF THIS PROGRAM :

```
"BEGIN" "ARRAY" A[1:8,1:5], B[1:8], EM[0:7];
    "INTEGER" I;
    "INTEGER" "PROCEDURE" SOLOVR(A, M, N, X, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, X, EM;
   "CODE" 34281;
    A[1,1]:=22; A[1,2]:= A[2,3]:=10; A[1,3]:= A[7,1]:= A[8,5]:=2;
    A[1,4]:= A[3,5]:=3; A[1,5]:= A[2,2]:=7; A[2,1]:=14; A[2,5]:=8;
    A[2,4]:= A[8,3]:=0; A[3,1]:= A[3,3]:= A[6,5]:=-1; A[3,2]:=13;
    A[3,4]:=-11; A[4,1]:=-3; A[4,2]:= A[4,4]:= A[5,4]:= A[8,4]:=-2;
    A[4,3]:=13; A[4,5]:= A[5,5]:= A[8,1]:=4; A[5,1]:= A[6,1]:=9;
    A[5,2]:=8; A[5,3]:= A[6,2]:= A[7,5]:=1; A[6,3]:=-7;
    A[6,4]:= A[7,4]:= A[8,2]:=5; A[7,2]:=-6; A[7,3]:=6;
    B[1]:=-1; B[2]:=2; B[3]:= B[7]:=1; B[4]:=4; B[5]:= B[8]:=0;
    B[6]:=-3; EM[0]:="-14; EM[2]:="-12; EM[4]:=80; EM[6]:="-10;
    I:= SOLOVR(A, 8, 5, B, EM);
    OUTPUT(61, "("4B, "("NUMBER SINGULAR VALUES NOT FOUND : ")",
    3ZD,/, 4B, "("NORM : ")", N,/, 4B, "("MAX NEGL SUBD ELEM : ")",
    N,/, 4B, "("NUMBER ITERATIONS : ")", 3ZD, /, 4B, "("RANK : ")",
    3ZD, /")", I, EM[1], EM[3], EM[5], EM[7]);
    OUTPUT(61, "("/, 4B, "("SOLUTION VECTOR")",/,/, 5(4B, N, /)")",
    B[1], B[2], B[3], B[4], B[5])
"END"
```

```
      NUMBER SINGULAR VALUES NOT FOUND :     0
      NORM : +4.4000000000000"+001
      MAX NEGL SUBD ELEM : +4.3977072741076"-014
      NUMBER ITERATIONS :     6
      RANK :     3

      SOLUTION VECTOR

      -8.3333333333334"-002
      +1.0989227456287"-015
      +2.5000000000000"-001
      -8.3333333333332"-002
      +8.3333333333334"-002
```

SOURCE TEXT(S):

```
"CODE" 34280;
"PROCEDURE" SOLSVDOVR(U, VAL, V, M, N, X, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, X, EM;
"BEGIN" "INTEGER" I;
     "REAL" MIN;
     "ARRAY" X1[1:N];
     "REAL" "PROCEDURE" MATVEC(L, U, I, A, B);
     "VALUE" L, U, I; "INTEGER" L, U, I; "ARRAY" A, B;
    "CODE" 34011;
     "REAL" "PROCEDURE" TAMVEC(L, U, I, A, B);
     "VALUE" L, U, I; "INTEGER" L, U, I; "ARRAY" A, B;
    "CODE" 34012;
     MIN:= EM[6];
     "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
     X1[I]:= "IF" VAL[I] <= MIN "THEN" 0 "ELSE" TAMVEC(1, M, I, U, X) /
     VAL[I];
     "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
     X[I]:= MATVEC(1, N, I, V, X1)
"END" SOLSVDOVR;
         "EOP"

"CODE" 34281;
"INTEGER" "PROCEDURE" SOLOVR(A, M, N, X, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" A, X, EM;
"BEGIN" "INTEGER" I;
     "ARRAY" VAL[1:N], V[1:N,1:N];
     "INTEGER" "PROCEDURE" QRISNGVALDEC(A, M, N, VAL, V, EM);
     "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, VAL, V, EM;
    "CODE" 34273;
     "PROCEDURE" SOLSVDOVR(U, VAL, V, M, N, X, EM);
     "VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, X, EM;
    "CODE" 34280;

     SOLOVR:= I:= QRISNGVALDEC(A, M, N, VAL, V, EM);
     "IF" I = 0 "THEN" SOLSVDOVR(A, VAL, V, M, N, X, EM)
"END" SOLOVR;
         "EOP"
```

AUTHOR : D.T.WINTER

INSTITUTE : MATHEMATICAL CENTRE

RECEIVED : 731217

BRIEF DESCRIPTION :

THIS SECTION CONTAINS 2 PROCEDURES FOR THE SOLUTION OF AN UNDERDETERMINED SYSTEM OF LINEAR EQUATIONS:
SOLUND EXPECTS AS INPUT THE MATRIX OF THE SYSTEM OF EQUATIONS, CALCULATES THE SINGULAR VALUES DECOMPOSITION BY MEANS OF THE PROCEDURE QRISNGVALDEC, AND SOLVES THE SYSTEM BY MEANS OF THE PROCEDURE SOLSVDUND.
SOLSVDUND ASSUMES THAT THE MATRIX IS ALREADY DECOMPOSED AND SOLVES THE SYSTEM OF EQUATIONS, MULTIPLYING THE RIGHT-HAND SIDE BY THE PSEUDO-INVERSE OF THE GIVEN MATRIX.

KEYWORDS :

BEST LEAST-SQUARES SOLUTION
SINGULAR VALUES
PSEUDO-INVERSE

SUBSECTION : SOLSVDUND

CALLING SEQUENCE :

```
    THE HEADING OF THE PROCEDURE IS :
    "PROCEDURE" SOLSVDUND(U, VAL, V, M, N, X, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, X, EM;

    THE MEANING OF THE FORMAL PARAMETERS IS :
    U:   <ARRAY IDENTIFIER>;
         "ARRAY" U[1:M,1:N];
         ENTRY:THE MATRIX U IN THE SINGULAR VALUES DECOMPOSITION V*S*U'.
    VAL: <ARRAY IDENTIFIER>;
         "ARRAY" VAL[1:N];
         ENTRY:THE SINGULAR VALUES;
    V:   <ARRAY IDENTIFIER>;
         "ARRAY" V[1:N,1:N];
         ENTRY:THE MATRIX V IN THE SINGULAR VALUES DECOMPOSITION.
    N:   <ARITHMETIC EXPRESSION>;
         THE LENGTH OF THE RIGHT-HAND SIDE VECTOR;
    M:   <ARITHMETIC EXPRESSION>;
         THE NUMBER OF UNKNOWNS, N SHOULD SATISFY N <= M;
    X:   <ARRAY IDENTIFIER>;
         "ARRAY" X[1:M];
         ENTRY: THE RIGHT-HAND SIDE VECTOR IN X[1:N];
         EXIT: THE SOLUTION VECTOR IN X[1:M];
    EM:  <ARRAY IDENTIFIER>;
         "ARRAY" EM[6:6];
         ENTRY: EM[6]: THE MINIMAL NON-NEGLECTABLE SINGULAR VALUE.
```

PROCEDURES USED :

MATVEC = CP34011
TAMVEC = CP34012

REQUIRED CENTRAL MEMORY : AN AUXILIARY ARRAY OF N REALS IS DECLARED

RUNNING TIME : ROUGHLY PROPORTIONAL TO (M + N) * N

METHOD AND PERFORMANCE :

THE SOLUTION IS FOUND IN THREE STEPS :

1. V' * X = X1 IS CALCULATED,
2. VAL+ * X1 = X2 IS CALCULATED, HERE VAL+ DENOTES THE DIAGONAL MATRIX OBTAINED FROM VAL BY SETTING VAL+[I,I] = 1/VAL[I] IF VAL[I] GREATER THAN OR EQUAL TO EM[6], AND 0 OTHERWISE,
3. THE SOLUTION U * X2 IS CALCULATED.

LANGUAGE : ALGOL 60

SUBSECTION : SOLUND

CALLING SEQUENCE :

```
    THE HEADING OF THE PROCEDURE IS :
    "INTEGER" "PROCEDURE" SOLUND(A, M, N, X, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, X, EM;

    SOLUND:= THE NUMBER OF SINGULAR VALUES NOT FOUND,  I.E. ZERO IF ALL
        SINGULAR VALUES ARE CALCULATED.

    THE MEANING OF THE FORMAL PAREMETERS IS :
    A:  <ARRAY IDENTIFIER>;
        "ARRAY" A[1:M,1:N];
        ENTRY: THE TRANSPOSE OF THE MATRIX;
    M:  <ARITHMETIC EXPRESSION>;
        THE NUMBER OF ROWS OF A;
    N:  <ARITHMETIC EXPRESSION>;
        THE NUMBER OF COLUMNS OF A, N <= M;
    X:  <ARRAY IDENTIFIER>;
        "ARRAY" X[1:M];
        ENTRY: THE RIGHT-HAND SIDE VECTOR IN X[1:N];
        EXIT: THE SOLUTION VECTOR.
    EM: <ARRAY IDENTIFIER>;
        "ARRAY" EM[0:7];
        ENTRY: EM[0]: THE MACHINE PRECISION;
            EM[2]: THE RELATIVE PRECISION FOR THE SINGULAR VALUES;
            EM[4]: THE MAXIMAL NUMBER OF ITERATIONS TO BE PERFORMED IN
                   THE SINGULAR VALUES DECOMPOSITION;
            EM[6]: THE MINIMAL NON-NEGLECTABLE SINGULAR VALUE;
        EXIT: EM[1]: THE INFINITY NORM OF THE MATRIX;
            EM[3]: THE MAXIMAL NEGLECTED SUPERDIAGONAL ELEMENT;
            EM[5]: THE NUMBER OF ITERATIONS PERFORMED  IN THE SINGULAR
                   VALUES DECOMPOSITION;
            EM[7]: THE NUMERICAL RANK OF THE MATRIX, I.E. THE NUMBER OF
                   SINGULAR VALUES GREATER THAN OR EQUAL TO EM[6].
```

PROCEDURES USED :

```
    QRISNGVALDEC = CP34273
    SOLSVDUND    = CP34282
```

REQUIRED CENTRAL MEMORY :

```
    AUXILIARY ARRAYS ARE DECLARED TO A TOTAL OF (N + 1) * N REALS
```

METHOD AND PERFORMANCE :

```
    THE SOLUTION IS FOUND IN TWO STEPS :
    1.  THE SINGULAR VALUES DECOMPOSITION IS CALCULATED BY MEANS OF THE
        PROCEDURE QRISNGVALDEC;
    2.  THE SOLUTION IS CALCULATED BY MEANS OF THE PROCEDURE SOLSVDUND.
```

RUNNING TIME : ROUGHLY PROPORTIONAL TO (M + N) * N * N

LANGUAGE : ALGOL-60

REFERENCES :

WILKINSON, J.H. AND C.REINSCH
HANDBOOK OF AUTOMATIC COMPUTATION, VOL. 2
LINEAR ALGEBRA
HEIDELBERG (1971)

EXAMPLE OF USE :

FIRST WE GIVE A PROGRAM, AND THAN THE RESULTS OF THIS PROGRAM :

```
"BEGIN" "ARRAY" A[1:8,1:5], B[1:8], EM[0:7];
    "INTEGER" I;
    "INTEGER" "PROCEDURE" SOLUND(A, M, N, X, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, X, EM;
   "CODE" 34283;
    A[1,1]:=22; A[1,2]:= A[2,3]:=10; A[1,3]:= A[7,1]:= A[8,5]:=2;
    A[1,4]:= A[3,5]:=3; A[1,5]:= A[2,2]:=7; A[2,1]:=14; A[2,5]:=8;
    A[2,4]:= A[8,3]:=0; A[3,1]:= A[3,3]:= A[6,5]:=-1; A[3,2]:=13;
    A[3,4]:=-11; A[4,1]:=-3; A[4,2]:= A[4,4]:= A[5,4]:= A[8,4]:=-2;
    A[4,3]:=13; A[4,5]:= A[5,5]:= A[8,1]:=4; A[5,1]:= A[6,1]:=0;
    A[5,2]:=8; A[5,3]:= A[6,2]:= A[7,5]:=1; A[6,3]:=-7;
    A[6,4]:= A[7,4]:= A[8,2]:=5; A[7,2]:=-6; A[7,3]:=6;
    B[1]:=-1; B[2]:=2; B[3]:=1; B[4]:=4; B[5]:=0;
    EM[0]:="-14; EM[2]:="-12; EM[4]:=80; EM[6]:="-10;
    I:= SOLUND(A, 8, 5, B, EM);
    OUTPUT(61, "("4B, "("NUMBER SINGULAR VALUES NOT FOUND : ")",
    3ZD, /, 4B, "("NORM : ")",N, /, 4B, "("MAX NEGL SUBD ELEM : ")",
    N,/, 4B, "("NUMBER ITERATIONS : ")", 3ZD, /, 4B, "("RANK : ")",
    3ZD, /")", I, EM[1], EM[3], EM[5], EM[7]);
    OUTPUT(61, "("/, 4B, "("SOLUTION VECTOR")", /,/, 8(4B, N,/)")",
    B[1], B[2], B[3], B[4], B[5], B[6], B[7], B[8])
"END"
```

```
 NUMBER SINGULAR VALUES NOT FOUND :    0
 NORM : +4.4000000000000"+001
 MAX NEGL SUBD ELEM : +4.3977072741076"-014
 NUMBER ITERATIONS :    6
 RANK :    3

 SOLUTION VECTOR

 +1.6410256410255"-002
 +1.4807692307694"-002
 -4.8397435897438"-002
 +1.0000000000002"-002
 -6.7948717948740"-003
 +1.1602564102565"-002
 +2.9999999999996"-002
 -8.3974358974328"-003
```

SOURCE TEXT(S):

```
"CODE" 34282;
"PROCEDURE" SOLSVDUND(U, VAL, V, M, N, X, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, X, EM;
"BEGIN" "INTEGER" I;
     "REAL" MIN;
     "ARRAY" X1[1:N];

     "REAL" "PROCEDURE" MATVEC(L, U, I, A, B);
     "VALUE" L, U, I; "INTEGER" L, U, I; "ARRAY" A, B;
    "CODE" 34011;

     "REAL" "PROCEDURE" TAMVEC(L, U, I, A, B);
     "VALUE" L, U, I; "INTEGER" L, U, I; "ARRAY" A, B;
    "CODE" 34012;

     MIN:= EM[6];
     "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
     X1[I]:= "IF" VAL[I] <= MIN "THEN" 0 "ELSE" TAMVEC(1, N, I, V, X) /
     VAL[I];
     "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO"
     X[I]:= MATVEC(1, N, I, U, X1)
"END" SOLSVDUND;
        "EOP"

"CODE" 34283;
"INTEGER" "PROCEDURE" SOLUND(A, M, N, X, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" A, X, EM;
"BEGIN" "INTEGER" I;
     "ARRAY" VAL[1:N], V[1:N,1:N];

     "INTEGER" "PROCEDURE" QRISNGVALDEC(A, M, N, VAL, V, EM);
     "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, VAL, V, EM;
    "CODE" 34273;

     "PROCEDURE" SOLSVDUND(U, VAL, V, M, N, X, EM);
     "VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, X, EM;
    "CODE" 34282;

     SOLUND:= I:= QRISNGVALDEC(A, M, N, VAL, V, EM);
     "IF" I = 0 "THEN" SOLSVDUND(A, VAL, V, M, N, X, EM)
"END" SOLUND;
        "EOP"
```

AUTHOR : D.T.WINTER

INSTITUTE : MATHEMATICAL CENTRE

RECEIVED : 731217

BRIEF DESCRIPTION :

THIS SECTION CONTAINS 2 PROCEDURES FOR THE
CALCULATION OF THE HOMOGENEOUS EQUATIONS A * X = 0 AND X' * A = 0,
WHERE A DENOTES A MATRIX, AND X A VECTOR;
HOMSOLSVD ASSUMES THAT THE SINGULAR VALUES DECOMPOSITION OF A HAS
BEEN GIVEN.
HOMSOL FIRST CALCULATES THE SINGULAR VALUES
DECOMPOSITION BY MEANS OF THE PROCEDURE QRISNGVALDEC.

KEYWORDS :

HOMOGENEOUS SOLUTION
SINGULAR VALUES

SUBSECTION : HOMSOLSVD

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :
"PROCEDURE" HOMSOLSVD(U, VAL, V, M, N);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V;

THE MEANING OF THE FORMAL PARAMETERS IS :

U: <ARRAY IDENTIFIER>;
"ARRAY" U[1:M,1:N];
ENTRY:THE MATRIX U IN THE SINGULAR VALUES DECOMPOSITION U*S*V'.
EXIT: THE COLUMNS OF U THAT CORRESPOND TO THE ELEMENTS
OF VAL WITH A VALUE SMALLER THAN SOME SMALL CONSTANT MAY BE
SEEN AS THE SOLUTIONS OF X' * A = 0;

VAL: <ARRAY IDENTIFIER>;
"ARRAY" VAL[1:N];
ENTRY:THE SINGULAR VALUES;
EXIT :THE ARRAY WILL BE ORDERED IN SUCH A WAY THAT
VAL[I] < VAL[J] IF J < I;

V: <ARRAY IDENTIFIER>;
"ARRAY" V[1:N,1:N];
ENTRY:THE MATRIX V IN THE SINGULAR VALUES DECOMPOSITION;
EXIT:THE COLUMNS OF V THAT CORRESPOND TO THE ELEMENTS OF VAL
THAT ARE SMALLER THAN SOME SMALL CONSTANT MAY BE SEEN AS THE
SOLUTIONS OF THE EQUATION A * X = 0;

M: <ARITHMETIC EXPRESSION>;
THE NUMBER OF ROWS OF U;

N: <ARITHMETIC EXPRESSION>;
THE NUMBER OF COLUMNS OF U;

PROCEDURES USED :

    ICHCOL = CP34031

RUNNING TIME : PROPORTIONAL TO N ' 2

METHOD AND PERFORMANCE :

THE PROCEDURE DOES NOTHING MORE THAN A SIMPLE SORTING PROCESS ON THE ELEMENTS OF THE ARRAY VAL, AT THE SAME TIME THE COLUMNS OF U AND V ARE INTERCHANGED, ACCORDING TO THE INTERCHANGING OF THE ELEMENTS VAL.

LANGUAGE : ALGOL 60

SUBSECTION : HOMSOL

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"INTEGER" "PROCEDURE" HOMSOL(A, M, N, V, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" A, V, EM;
```

HOMSOL:= THE NUMBER OF SINGULAR VALUES NOT FOUND, I.E. ZERO IF ALL SINGULAR VALUES ARE CALCULATED.

THE MEANING OF THE FORMAL PAREMETERS IS :

```
A:  <ARRAY IDENTIFIER>;
    "ARRAY" A[1:M,1:N];
    ENTRY: THE MATRIX;
    EXIT:  THE COLUMNS OF A THAT  CORRESPOND TO THE ELEMENTS OF
        VAL THAT ARE SMALLER THAN SOME SMALL CONSTANT,  MAY BE SEEN
        AS THE SOLUTIONS OF THE EQUATION X' * A = 0.
M:  <ARITHMETIC EXPRESSION>;
    THE NUMBER OF ROWS OF A.
N:  <ARITHMETIC EXPRESSION>;
    THE NUMBER OF COLUMNS OF A.
```

V: <ARRAY IDENTIFIER>;
"ARRAY" V[1:N,1:N];
EXIT: THE COLUMNS OF V THAT CORRESPOND TO ELEMENTS OF VAL SMALLER THAN SOME SMALL CONSTANT MAY BE SEEN AS THE SOLUTIONS OF THE EQUATION A * X = 0.

EM: <ARRAY IDENTIFIER>;
"ARRAY" EM[0:7];
ENTRY: EM[0]: THE MACHINE PRECISION;
EM[2]: THE RELATIVE PRECISION FOR THE SINGULAR VALUES;
EM[4]: THE MAXIMAL NUMBER OF ITERATIONS TO BE PERFORMED IN THE SINGULAR VALUES DECOMPOSITION.
EM[6]: THE MINIMAL NON-NEGLECTABLE SINGULAR VALUE;
EXIT: EM[1]: THE INFINITY NORM OF THE MATRIX;
EM[3]: THE MAXIMAL NEGLECTED SUPERDIAGONAL ELEMENT;
EM[5]: THE NUMBER OF ITERATIONS PERFORMED IN THE SINGULAR VALUES DECOMPOSITION;
EM[7]: THE NUMERICAL RANK OF THE MATRIX, I.E. THE NUMBER OF SINGULAR VALUES GREATER THAN OR EQUAL TO EM[6].

PROCEDURES USED :

QRISNGVALDEC = CP34273
HOMSOLSVD = CP34284

METHOD AND PERFORMANCE :

THE SOLUTION IS FOUND IN TWO STEPS :

1. THE SINGULAR VALUES DECOMPOSITION IS CALCULATED BY MEANS OF THE PROCEDURE QRISNGVALDEC;
2. THE SINGULAR VALUES ARE ORDERED BY MEANS OF THE PROCEDURE HOMSOLSVD.

RUNNING TIME : ROUGHLY PROPORTIONAL TO (M + N) * N * N

LANGUAGE : ALGOL 60

REFERENCES :

WILKINSON, J.H. AND C.REINSCH
HANDBOOK OF AUTOMATIC COMPUTATION, VOL. 2
LINEAR ALGEBRA
HEIDELBERG (1971)

EXAMPLE OF USE :

FIRST WE GIVE A PROGRAM, AND THAN THE RESULTS OF THIS PROGRAM :

```
"BEGIN" "ARRAY" A[1:8,1:5], V[1:5,1:5], EM[0:7];
    "INTEGER" I, J;
    A[1,1]:=22; A[1,2]:= A[2,3]:=10; A[1,3]:= A[7,1]:= A[8,5]:=2;
    A[1,4]:= A[3,5]:=3; A[1,5]:= A[2,2]:=7; A[2,1]:=14; A[2,5]:=8;
    A[2,4]:= A[8,3]:=6; A[3,1]:= A[3,3]:= A[6,5]:=-1; A[3,2]:=13;
    A[3,4]:=-11; A[4,1]:=-3; A[4,2]:= A[4,4]:= A[5,4]:= A[8,4]:=-2;
    A[4,3]:=13; A[4,5]:= A[5,5]:= A[8,1]:=4; A[5,1]:= A[6,1]:=9;
    A[5,2]:=8; A[5,3]:= A[6,2]:= A[7,5]:=1; A[6,3]:=-7;
    A[6,4]:= A[7,4]:= A[8,2]:=5; A[7,2]:=-6; A[7,3]:=6;
    EM[0]:="-14; EM[2]:="-12; EM[4]:=80; EM[6]:="-10;
    I:= HOMSOL(A, 8, 5, V, EM);
    OUTPUT(61, "("4B, "("NUMBER SINGULAR VALUES NOT FOUND : ")",
    3ZD,/, 4B, "("NORM : ")", N,/, 4B, "("MAX NEGL SUBD ELEM : ")",
    N,/, 4B, "("NUMBER ITERATIONS : ")", 3ZD, /, 4B, "("RANK : ")",
    3ZD, /")", I, EM[1], EM[3], EM[5], EM[7]);
    "FOR" J:= EM[7] + 1 "STEP" 1 "UNTIL" 5 "DO"
    OUTPUT(61, "("/, 4B, "("COLUMN NUMBER : ")", D, 5(/ ,4B, 2(N)),
    3(/, 4B, N), /")", J, A[1,J], V[1,J], A[2,J], V[2,J], A[3,J],
    V[3,J], A[4,J], V[4,J], A[5,J], V[5,J], A[6,J], A[7,J], A[8,J])
"END"
```

```
 NUMBER SINGULAR VALUES NOT FOUND :     0
 NORM : +4.4000000000000"+001
 MAX NEGL SUBD ELEM : +4.3977072741076"-014
 NUMBER ITERATIONS :     6
 RANK :     3

 COLUMN NUMBER : 4
 +3.4708599800002"-001  -4.1909548511171"-001
 -6.0723369623011"-001  +4.4050912303713"-001
 +1.2207461910546"-001  -5.2004549247434"-002
 +6.1882574433898"-001  +6.7605914021670"-001
 -4.6344371870996"-003  +4.1297730284731"-001
 +3.3409859838125"-001
 -3.3528410857408"-002
 -1.3547246422274"-002

 COLUMN NUMBER : 5
 -2.5533109413182"-001  +0.0000000000000"+000
 -1.7359809248754"-001  -4.1854806384909"-001
 -2.2081225414163"-001  -3.4879005320758"-001
 +4.1165471593410"-002  -2.4415303724531"-001
 +9.2044247057656"-001  +8.0221712237742"-001
 -2.8895953996492"-002
 +6.1327596621994"-002
 -4.9058079025100"-002
```

SOURCE TEXT(S):

```
"CODE" 34284;
"PROCEDURE" HOMSOLSVD(U, VAL, V, M, N);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V;
"BEGIN" "INTEGER" I, J;
    "REAL" X;

    "PROCEDURE" ICHCOL(L, U, I, J, A);
    "VALUE" L, U, I, J; "INTEGER" L, U, I, J; "ARRAY" A;
   "CODE" 34031;

    "FOR" I:= N "STEP" - 1 "UNTIL" 2 "DO"
    "FOR" J:= I - 1 "STEP" - 1 "UNTIL" 1 "DO"
    "IF" VAL[I] > VAL[J] "THEN"
    "BEGIN" X:= VAL[I]; VAL[I]:= VAL[J]; VAL[J]:= X;
        ICHCOL(1, M, I, J, U); ICHCOL(1, N, I, J, V)
    "END"
"END" HOMSOLSVD;
        "EOP"

"CODE" 34285;
"INTEGER" "PROCEDURE" HOMSOL(A, M, N, V, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" A, V, EM;
"BEGIN" "INTEGER" I;
    "ARRAY" VAL[1:N];

    "INTEGER" "PROCEDURE" QRISNGVALDEC(A, M, N, VAL, V, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, VAL, V, EM;
   "CODE" 34273;

    "PROCEDURE" HOMSOLSVD(U, VAL, V, M, N);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V;
   "CODE" 34284;

    HOMSOL:= I:= QRISNGVALDEC(A, M, N, VAL, V, EM);
    "IF" I = 0 "THEN" HOMSOLSVD(A, VAL, V, M, N)
"END" HOMSOL;
        "EOP"
```

AUTHOR : D.T.WINTER

INSTITUTE : MATHEMATICAL CENTRE

RECEIVED : 731217

BRIEF DESCRIPTION :

THIS SECTION CONTAINS TWO PROCEDURES FOR THE CALCULATION OF THE PSEUDO-INVERSE OF A MATRIX:
PSDINVSVD ASSUMES THAT THE MATRIX IS GIVEN AS SINGULAR VALUES DECOMPOSITION.
PSDINV FIRST CALCULATES THIS DECOMPOSITION.

KEYWORDS :

PSEUDO-INVERSE
SINGULAR VALUES

SUBSECTION : PSDINVSVD

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" PSDINVSVD(U, VAL, V, M, N, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, EM; "CODE" 34286;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

```
U:   <ARRAY IDENTIFIER>;
     "ARRAY" U[1:M,1:N];
     ENTRY: THE MATRIX U IN THE SINGULAR VALUES DECOMPOSITION
          U * S * V';
     EXIT: THE TRANSPOSE OF THE PSEUDO-INVERSE.
VAL: <ARRAY IDENTIFIER>;
     "ARRAY" VAL[1:N];
     THE SINGULAR VALUES.
V:   <ARRAY IDENTIFIER>;
     "ARRAY" V[1:N,1:N];
     THE MATRIX V IN THE SINGULAR VALUES DECOMPOSITION U * S * V'.
M:   <ARITHMETIC EXPRESSION>;
     THE NUMBER OF ROWS OF U.
N:   <ARITHMETIC EXPRESSION>;
     THE NUMBER OF COLUMNS OF V.
EM:  <ARRAY IDENTIFIER>;
     "ARRAY" EM[6:6];
     ENTRY: EM[6]:THE MINIMAL NON-NEGLECTABLE SINGULAR VALUE.
```

```
PROCEDURES USED :

    MATVEC = CP34011

REQUIRED CENTRAL MEMORY : AN AUXILIARY ARRAY OF N REALS IS DECLARED

RUNNING TIME : ROUGHLY PROPORTIONAL TO M * N * N


METHOD AND PERFORMANCE :

    THE PSEUDO-INVERSE IS CALCULATED IN TWO STEPS :
    1.  THE MATRIX X = VAL+ * U' IS CALCULATED,  WHERE VAL+ DENOTES THE
        DIAGONAL MATRIX OBTAINED FROM VAL BY PUTTING
        VAL+[I,I] = 1/VAL[I] IF VAL[I] GREATER THAN OR EQUAL TO EM[6],
        AND VAL+[I,I] = 0 OTHERWISE.
    2.  THE PSEUDO INVERSE (V * X) IS CALCULATED.


SUBSECTION : PSDINV


CALLING SEQUENCE :

    THE HEADING OF THE PROCEDURE IS :
    "INTEGER" "PROCEDURE" PSDINV(A, M, N, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, EM; "CODE" 34287;

    PSDINV:= THE NUMBER OF SINGULAR VALUES NOT FOUND,  I.E. ZERO IF ALL
        SINGULAR VALUES ARE CALCULATED.

    THE MEANING OF THE FORMAL PARAMETERS IS :
    A   <ARRAY IDENTIFIER>;
        "ARRAY" A[1:M,1:N];
        ENTRY : THE GIVEN MATRIX;
        EXIT : THE TRANSPOSE OF THE PSEUDO-INVERSE;
    M:  <ARITHMETIC EXPRESSION>;
        THE NUMBER OF ROWS OF A;
    N:  <ARITHMETIC EXPRESSION>;
        THE NUMBER OF COLUMNS OF A, N<= M;
    EM: <ARRAY IDENTIFIER>;
        "ARRAY" EM[0:7];
     ENTRY: EM[0]: THE MACHINE PRECISION;
            EM[2]: THE RELATIVE PRECISION FOR THE SINGULAR VALUES;
            EM[4]: THE MAXIMAL NUMBER OF ITERATIONS TO BE PERFORMED;
            EM[6]: THE MINIMAL NON-NEGLECTABLE SINGULAR VALUE;
      EXIT: EM[1]: THE INFINITY NORM OF THE MATRIX;
            EM[3]: THE MAXIMAL NEGLECTED SUPERDIAGONAL ELEMENT;
            EM[5]: THE NUMBER OF ITERATIONS  PERFORMED IN THE SINGULAR
                   VALUES DECOMPOSITION;
            EM[7]: THE NUMERICAL RANK OF THE MATRIX, I.E. THE NUMBER OF
                   SINGULAR VALUES GREATER THAN OR EQUAL TO EM[6];
```

PROCEDURES USED :

```
    QRISNGVALDEC = CP34273
    PSDINVSVD    = CP34286
```

REQUIRED CENTRAL MEMORY :

AUXILIARY ARRAY'S ARE DECLARED TO A TOTAL OF (N + 1) * N REALS

RUNNING TIME : ROUGHLY PROPORTIONAL TO (2M + N) * N * N

METHOD AND PERFORMANCE :

FIRST THE SINGULAR VALUES DECOMPOSITION IS CALCULATED, AND THAN THE PSEUDO-INVERSE IS CALCULATED BY PSDINVSVD.

REFERENCES :

WILKINSON, J.H. AND C.REINSCH
HANDBOOK OF AUTOMATIC COMPUTATION, VOL.2
LINEAR ALGEBRA
HEIDELBERG (1971)

EXAMPLE OF USE :

FIRST WE GIVE A PROGRAM, AND THEN THE RESULTS OF THIS PROGRAM :

```
"BEGIN" "ARRAY" A[1:8,1:5], EM[0:7];
    "INTEGER" I, J;

    "INTEGER" "PROCEDURE" PSDINV(A, M, N, EM);
    "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, EM;
   "CODE" 34287;

    A[1,1]:=22; A[1,2]:= A[2,3]:=10; A[1,3]:= A[7,1]:= A[8,5]:=2;
    A[1,4]:= A[3,5]:=3; A[1,5]:= A[2,2]:=7; A[2,1]:=14; A[2,5]:=8;
    A[2,4]:= A[8,3]:=0; A[3,1]:= A[3,3]:= A[6,5]:=-1; A[3,2]:=13;
    A[3,4]:=-11; A[4,1]:=-3; A[4,2]:= A[4,4]:= A[5,4]:= A[8,4]:=-2;
    A[4,3]:=13; A[4,5]:= A[5,5]:= A[8,1]:=4; A[5,1]:= A[6,1]:=9;
    A[5,2]:=8; A[5,3]:= A[6,2]:= A[7,5]:=1; A[6,3]:=-7;
    A[6,4]:= A[7,4]:= A[8,2]:=5; A[7,2]:=-6; A[7,3]:=6;
    EM[0]:="-14; EM[2]:="-12; EM[4]:=80; EM[6]:="-10;
    I:= PSDINV(A, 8, 5, EM);
    OUTPUT(61, "("4B, "("NUMBER SINGULAR VALUES NOT FOUND : ")",
    3ZD,/, 4B, "("NORM : ")", N,/, 4B, "("MAX NEGL SUBD ELEM : ")",
    N,/, 4B, "("NUMBER ITERATIONS : ")", 3ZD, /, 4B, "("RANK : ")",
    3ZD, /")", I, EM[1], EM[3], EM[5], EM[7]);
    OUTPUT(61, "("/, 4B, "("TRANSPOSE OF PSEUDO-INVERSE")", /,
    4B, "("FIRST THREE COLUMNS")", /, /, 8(4B, 3(N), /), /, /,
    4B, "("LAST TWO COLUMNS")", /, /, 8(15B, 2(N), /)")",
    A[1,1], A[1,2], A[1,3], A[2,1], A[2,2], A[2,3], A[3,1], A[3,2],
    A[3,3], A[4,1], A[4,2], A[4,3], A[5,1], A[5,2], A[5,3], A[6,1],
    A[6,2], A[6,3], A[7,1], A[7,2], A[7,3], A[8,1], A[8,2], A[8,3],
    A[1,4], A[1,5], A[2,4], A[2,5], A[3,4], A[3,5], A[4,4], A[4,5],
    A[5,4], A[5,5], A[6,4], A[6,5], A[7,4], A[7,5], A[8,4], A[8,5])
"END"
```

```
 NUMBER SINGULAR VALUES NOT FOUND :     0
 NORM : +4.4000000000000"+001
 MAX NEGL SUBD ELEM : +4.3977072741076"-014
 NUMBER ITERATIONS :     6
 RANK :     3

 TRANSPOSE OF PSEUDO-INVERSE
 FIRST THREE COLUMNS

+2.1129807692308"-002  +4.6153846153850"-003  -2.1073717948727"-003
+9.3108974358974"-003  +2.2115384615376"-003  +2.0528846153848"-002
-1.1097756410256"-002  +2.7403846153848"-002  -3.8862179487199"-003
-7.9166666666667"-003  -5.0000000000007"-003  +3.3750000000001"-002
+5.5128205128205"-003  +9.8076923076935"-003  -8.9743589743826"-004
+1.4318910256410"-002  -2.5961538461548"-003  -2.0136217948716"-002
+4.8958333333335"-003  -1.4999999999998"-002  +1.5312499999996"-002
+1.5064102564102"-003  +7.4038461538447"-003  -1.6987179487147"-003
```

LAST TWO COLUMNS

```
+7.6041666666662"-003  +3.8060897435894"-003
-2.0833333333295"-004  +1.0016025641026"-002
-2.7604166666667"-002  +4.2067307692303"-003
-5.4166666666662"-003  +1.0416666666667"-002
-5.0000000000005"-003  +3.2051282051275"-003
+1.2812500000000"-002  -6.2099358974354"-003
+1.2395833333332"-002  +2.6041666666656"-003
-4.9999999999993"-003  +1.6025641025649"-003
```

SOURCE TEXT(S):

```
"CODE" 34286;
"PROCEDURE" PSDINVSVD(U, VAL, V, M, N, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, EM;
"BEGIN" "INTEGER" I, J;
     "REAL" MIN, VALI;
     "ARRAY" X[1:N];
     "REAL" "PROCEDURE" MATVEC(L, U, I, A, B);
     "VALUE" L, U, I; "INTEGER" L, U, I; "ARRAY" A, B;
    "CODE" 34011;
     MIN:= EM[6];
     "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
     "IF" VAL[I] > MIN "THEN"
     "BEGIN" VALI:= 1 / VAL[I];
        "FOR" J:= 1 "STEP" 1 "UNTIL" M "DO" U[J,I]:= U[J,I] * VALI
     "END"
     "ELSE" "FOR" J:= 1 "STEP" 1 "UNTIL" M "DO" U[J,I]:= 0;
     "FOR" I:= 1 "STEP" 1 "UNTIL" M "DO"
     "BEGIN" "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO" X[J]:= U[I,J];
         "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO"
         U[I,J]:= MATVEC(1, N, J, V, X)
     "END"
"END" PSDINVSVD;
         "EOP"

"CODE" 34287;
"INTEGER" "PROCEDURE" PSDINV(A, M, N, EM);
"VALUE" M, N; "INTEGER" M, N; "ARRAY" A, EM;
"BEGIN" "INTEGER" I;
     "ARRAY" VAL[1:N], V[1:N,1:N];
     "INTEGER" "PROCEDURE" QRISNGVALDEC(A, M, N, VAL, V, EM);
     "VALUE" M, N; "INTEGER" M, N; "ARRAY" A, VAL, V, EM;
    "CODE" 34273;
     "PROCEDURE" PSDINVSVD(U, VAL, V, M, N, EM);
     "VALUE" M, N; "INTEGER" M, N; "ARRAY" U, VAL, V, EM;
    "CODE" 34286;

     PSDINV:= I:= QRISNGVALDEC(A, M, N, VAL, V, EM);
     "IF" I = 0 "THEN" PSDINVSVD(A, VAL, V, M, N, EM)
"END" PSDINV;
         "EOP"
```

AUTHOR : T.J. DEKKER.

REVISOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 730903.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS THE PROCEDURE DECBND
FOR THE DECOMPOSITION OF A BAND MATRIX BY GAUSSIAN ELIMINATION
WITH STABILIZING ROW INTERCHANGES (PARTIAL PIVOTING).

KEY WORDS :

LINEAR EQUATIONS,
PARTIAL PIVOTING,
GAUSSIAN ELIMINATION,
BAND MATRIX.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" DECBND(A, N, LW, RW, AUX, M, P); "VALUE" N, LW, RW;
"INTEGER" N, LW, RW; "INTEGER""ARRAY" P; "ARRAY" A, M, AUX;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

```
A     :  <ARRAY IDENTIFIER>;
         "ARRAY" A[1 : (LW + RW) * (N - 1) + N];
         ENTRY  : A CONTAINS ROWWISE THE BAND ELEMENTS OF THE
          BAND MATRIX IN SUCH A WAY THAT THE (I,J)-TH ELEMENT OF
          THE MATRIX IS GIVEN IN A[( LW + RW ) * (I - 1) + J],
          I=1,...,N AND J=MAX(1,I-LW),...,MIN(N,I+RW).
          THE  VALUES  OF  THE  REMAINING  ELEMENTS  OF  A  ARE
          IRRELEVANT.
         EXIT   : THE BAND ELEMENTS OF THE GAUSSIAN ELIMINATED
          MATRIX, WHICH IS AN UPPERTRIANGULAR BAND MATRIX WITH
          (LW + RW) CODIAGONALS, ARE ROWWISE DELIVERED IN A  AS
          FOLLOWS: THE (I,J)-TH ELEMENT OF U IS
          A [ ( LW + RW ) * (I - J) + J ],I=1,...,N AND
          J=I,...,MIN(N,I + LW + RW).
N     :  <ARITHMETIC EXPRESSION>;
         ORDER OF THE BAND MATRIX;
LW    :  <ARITHMETIC EXPRESSION>;
         NUMBER OF  LEFT CODIAGONALS OF A;
```

```
        RW    :   <ARITHMETIC EXPRESSION>;
                  NUMBER OF RIGHT CODIAGONALS OF A;
        AUX   :   <ARRAY IDENTIFIER>;
                  "ARRAY" AUX[1 : 5];
                  ENTRY  :AUX[2] = EPS IS A RELATIVE TOLERANCE TO CONTROL
                    THE ELIMINATION; THE PROCESS IS DISCONTINUED IF
                    (EPS > PIVOT[I] / EUCLIDEAN NORM OF I-TH ROW)
                    IN THE I-TH ELIMINATION STEP;
                  NORMAL EXIT  :
                    AUX[1] = SIGN OF THE DETERMINANT OF THE MATRIX
                    (+1 OR -1);
                    AUX[3] = N;
                    AUX[5] = MINIMUM ABSOLUTE VALUE OF
                    PIVOT[I] / EUCLIDEAN NORM OF THE I-TH ROW;
                  ABNORMAL EXIT : IF THE ELIMINATION CANNOT BE CARRIED
                    OUT, I.E. IF TEMP (THE QUANTITY
                    ABS(PIVOT[I] / EUCLIDEAN NORM OF THE I-TH ROW))
                    IS TOO SMALL IN THE I-TH ELIMINATION STEP :
                    AUX[3] = I - 1,
                    AUX[5] = TEMP;
        M   :     <ARRAY IDENTIFIER>;
                  "ARRAY" M[1 : LW * (N - 2) + 1];
                  EXIT :  THE GAUSSIAN MULTIPLIERS OF ALL ELIMINATIONS
                   IN SUCH A WAY THAT THE I-TH MULTIPLIER OF THE J-TH
                   STEP  IS  M [ LW * (J - 1) + I - J ].
        P   :     <ARRAY IDENTIFIER>;
                  "INTEGER""ARRAY" P[1 : N];
                  EXIT :  THE PIVOTAL INDICES.
```

PROCEDURES USED   :

```
    VECVEC = CP34010,
    ELMVEC = CP34020,
    ICHVEC = CP34030.
```

REQUIRED CENTRAL MEMORY   :

EXECUTION FIELD LENGTH : A REAL ARRAY OF N ELEMENTS IS DECLARED.

RUNNING TIME   :

```
    (C1 * LW + C2) * (LW + RW + 1) * N;
    THE CONSTANTS C1 AND C2 DEPEND UPON THE
    ARITHMETIC OF THE COMPUTER.
```

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

DECBND PERFORMS THE DECOMPOSITION OF A MATRIX WHOSE NON-ZERO ELEMENTS ARE IN BAND FORM, AND WHOSE BAND ELEMENTS ARE STORED ROWWISE IN A ONE-DIMENSIONAL ARRAY.
THE METHOD USED IS GAUSSIAN ELIMINATION WITH STABILIZING ROW INTERCHANGES (PARTIAL PIVOTING).
THE GAUSSIAN ELIMINATION IS PERFORMED IN N STEPS. IN THE K-TH STEP, K = 1, ... , N, A PIVOT IS SELECTED IN THE K-TH COLUMN OF THE REMAINING SUBMATRIX OF ORDER N - K + 1 (THIS COLUMN CONTAINS AT MOST LW + 1 NON-ZERO ELEMENTS); THEN THE PIVOTAL ROW IS INTERCHANGED WITH THE K-TH ROW; SUBSEQUENTLY THE K-TH UNKNOWN IS ELIMINATED IN THE LAST N - K ROWS (ONLY THE FIRST LW OF THESE LAST ROWS ARE INVOLVED HERE).
THE PIVOT IS SELECTED IN SUCH A WAY THAT ITS ABSOLUTE VALUE DIVIDED BY THE EUCLIDEAN NORM OF THE CORRESPONDING ROW OF THE MATRIX IS MAXIMAL. THUS, THE MATRIX IS EQUILIBRATED IN THIS PIVOTING STRATEGY SUCH, THAT THE ROWS EFFECTIVELY OBTAIN UNIT EUCLIDEAN NORM.
THE PROCEDURE DELIVERS THE BAND ELEMENTS OF THE ELIMINATED MATRIX (WHICH IS AN UPPER TRIANGULAR MATRIX WITH LW + RW SUPERDIAGONALS) AND THE GAUSSIAN MULTIPLIERS FOR EACH ELIMINATION .
THE ELIMINATION CANNOT BE CARRIED OUT IF THE ABSOLUTE VALUE OF THE PIVOT IS LESS THAN A GIVEN RELATIVE TOLERANCE (AUX[2]) TIMES THE EUCLIDEAN NORM OF THE CORRESPONDING ROW OF THE MATRIX. THEN THE PREVIOUS STEP NUMBER OF THE ELIMINATION IS DELIVERED (IN AUX[3], WHICH ELSE TAKES THE VALUE N). SEE ALSO REF [1], SECTION 212.

REFERENCE :

[1] DEKKER, T.J. :
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1,
MC TRACT 22, 1968, MATHEMATISCH CENTRUM, AMSTERDAM.

EXAMPLE OF USE :

SEE EXAMPLE OF USE OF SOLBND.

SOURCE TEXT(S) :

```
"CODE" 34320;
"PROCEDURE" DECBND(A, N, LW, RW, AUX, M, P); "VALUE" N, LW, RW;
"INTEGER" N, LW, RW; "INTEGER" "ARRAY" P; "ARRAY" A, M, AUX;
"BEGIN" "INTEGER" I, J, K, KK, KK1, PK, MK, IK, LW1, F, Q, W, W1,
        W2, NRW, IW, SDET;
        "REAL" R, S, EPS, MIN;
        "ARRAY" V[1:N];

        "REAL""PROCEDURE" VECVEC(A, B, C, D, E); "CODE" 34010;
        "PROCEDURE" ELMVEC(A, B, C, D, E, F); "CODE" 34020;
        "PROCEDURE" ICHVEC(A, B, C, D); "CODE" 34030;
        F:= LW; W1:= LW + RW; W:= W1 + 1; W2:= W - 2; IW:= 0; SDET:= 1;
        NRW:= N - RW; LW1:= LW + 1; Q:= LW - 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" LW "DO"
        "BEGIN" Q:= Q - 1; IW:= IW + W1;
            "FOR" J:= IW - Q "STEP" 1 "UNTIL" IW "DO" A[J]:= 0
        "END";
        IW:= - W2; Q:= - LW;
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" IW:= IW + W; "IF" I <= LW1 "THEN" IW:= IW - 1;
            Q:= Q + W; "IF" I > NRW "THEN" Q:= Q - 1;
            V[I]:= SQRT(VECVEC(IW, Q, 0, A, A))
        "END";
        EPS:= AUX[2]; MIN:= 1; KK:= - W1; MK:= - LW;
        "IF" F > NRW "THEN" W2:= W2 + NRW - F;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN""IF" F < N "THEN" F:= F + 1; IK:= KK:= KK + W;
            MK:= MK + LW; S:= ABS(A[KK]) / V[K]; PK:= K; KK1:= KK + 1;
            "FOR" I:= K + 1 "STEP" 1 "UNTIL" F "DO"
            "BEGIN" IK:= IK + W1; M[MK + I - K]:= R:= A[IK]; A[IK]:= 0;
                R:= ABS(R) / V[I]; "IF" R > S "THEN"
                "BEGIN" S:= R; PK:= I "END"
            "END";
            "IF" S < MIN "THEN" MIN:= S; "IF" S < EPS "THEN"
            "BEGIN" AUX[3]:= K - 1; AUX[5]:= S; "GO TO" END "END";
            "IF" K + W2 >= N "THEN" W2:= W2 - 1;
            P[K]:= PK; "IF" PK ^= K "THEN"
            "BEGIN" V[PK]:= V[K];
                PK:= PK - K; ICHVEC(KK1, KK1 + W2, PK * W1, A);
                SDET:= - SDET; R:= M[MK + PK]; M[MK + PK]:= A[KK];
                A[KK]:= R
            "END""ELSE" R:= A[KK]; "IF" R < 0 "THEN" SDET:= - SDET;
            IW:= KK1; LW1:= F - K + MK;
            "FOR" I:= MK + 1 "STEP" 1 "UNTIL" LW1 "DO"
            "BEGIN" M[I]:= S:= M[I] / R; IW:= IW + W1;
                ELMVEC(IW, IW + W2, KK1 - IW, A, A, - S)
            "END"
        "END";
        AUX[3]:= N; AUX[5]:= MIN;
    END: AUX[1]:= SDET
    "END" DECBND;
        "EOP"
```

CONTRIBUTOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 730903.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS THE PROCEDURE DETERMBND
FOR THE CALCULATION OF THE DETERMINANT OF A BAND MATRIX.

KEY WORDS :

DETERMINANT,
BAND MATRIX.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"REAL""PROCEDURE" DETERMBND(A, N, LW, RW, SGNDET); "VALUE" N, LW,
RW, SGNDET; "INTEGER" N, LW, RW, SGNDET; "ARRAY" A;
```

DETERMBND DELIVERS THE DETERMINANT OF THE MATRIX.

THE MEANING OF THE FORMAL PARAMETERS IS :

```
A, N, LW, RW  :   SEE CALLING SEQUENCE OF DECBND
             (SECTION 3.1.2.1.1.1.1.1.);
             ENTRY : THE CONTENTS OF A ARE AS PRODUCED BY DECBND OR
             DECSOLBND (SECTION 3.1.2.1.1.1.1.3.);
SGNDET  :    <ARITHMETIC EXPRESSION>;
             ENTRY : THE SIGN OF THE DETERMINANT AS DELIVERED IN
             AUX[1] BY DECBND, IF THE ELIMINATION BY DECBND WAS
             SUCCESSFUL.
```

PROCEDURES USED : NONE.

RUNNING TIME : PROPORTIONAL TO N.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

DETERMBND CAN BE CALLED AFTER DECBND OR DECSOLBND ONLY IF THE GAUSSIAN ELIMINATION WAS SUCCESSFUL, I.E. IF AUX[3] = N.
THE FUNCTION VALUE OF DETERMBND IS
THE DETERMINANT OF THE GAUSSIAN ELIMINATED UPPER TRIANGULAR MATRIX PROVIDED WITH THE CORRECT SIGN THAT IS DELIVERED BY DECBND OR DECSOLBND IN AUX[1].
DETERMBND SHOULD NOT BE CALLED WHEN OVERFLOW CAN BE EXPECTED.

EXAMPLE OF USE :

SEE EXAMPLES OF USE OF SOLBND AND DECSOLBND.

SOURCE TEXT(S) :

```
"CODE" 34321;
"REAL""PROCEDURE" DETERMBND(A, N, LW, RW, SGNDET);
"VALUE" N, LW, RW, SGNDET; "INTEGER" N, LW, RW, SGNDET; "ARRAY" A;
"BEGIN""INTEGER" I, L; "REAL" P;
        L:= 1; P:= 1; LW:= LW + RW + 1;
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" P:= A[L] * P; L:= L + LW "END";
        DETERMBND:= ABS(P) * SGNDET
    "END" DETERMBND;
        "EOP"
```

AUTHOR : T.J. DEKKER.

REVISOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 730903.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS TWO PROCEDURES.
A) SOLBND, FOR THE SOLUTION OF ONE OR MORE SYSTEMS OF LINEAR EQUATIONS WITH THE SAME COEFFICIENT MATRIX, IF THIS MATRIX HAS BEEN DECOMPOSED BY A CALL OF THE PROCEDURE DECBND (SECTION 3.1.2.1.1.1.1.1.).
B) DECSOLBND, FOR THE SOLUTION OF ONE SYSTEM OF LINEAR EQUATIONS BY GAUSSIAN ELIMINATION WITH STABILIZING ROW INTERCHANGES (PARTIAL PIVOTING) IF THE COEFFICIENT MATRIX IS IN BAND FORM AND IS STORED ROWWISE IN A ONE-DIMENSIONAL ARRAY.

KEY WORDS :

LINEAR EQUATIONS,
PARTIAL PIVOTING,
GAUSSIAN ELIMINATION,
BAND MATRIX.

SUBSECTION : SOLBND.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" SOLBND(A, N, LW, RW, M, P, B); "VALUE" N, LW, RW;
"INTEGER" N, LW, RW; "INTEGER""ARRAY" P; "ARRAY" A, M, B;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

A, N, LW, RW, M, P : SEE CALLING SEQUENCE OF DECBND,
ENTRY : THE CONTENTS OF THE ARRAYS A, M, P ARE AS PRODUCED BY DECBND;

B : <ARRAY IDENTIFIER>;
"ARRAY" B[1 : N];
ENTRY : THE RIGHT HAND SIDE OF THE SYSTEM OF LINEAR EQUATIONS;
EXIT : THE SOLUTION OF THE SYSTEM.

PROCEDURES USED :
    VECVEC = CP34010,
    ELMVEC = CP34020.

RUNNING TIME :
    (C3 * LW + C4 * RW + C5) * N;
    THE CONSTANTS C3, C4 AND C5 DEPEND UPON THE
    ARITHMETIC OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

SOLBND CALCULATES THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS, PROVIDED THAT THE MATRIX WAS DECOMPOSED BY A SUCCESSFUL CALL OF DECBND (SECTION 3.1.2.1.1.1.1.1.).
THE SOLUTION OF THE LINEAR SYSTEM IS OBTAINED BY CARRYING OUT THE ELIMINATIONS, FOR WHICH THE GAUSSIAN MULTIPLIERS ARE SAVED, ON THE RIGHT HAND SIDE, AND BY SOLVING THE NEW SYSTEM WITH THE UPPER TRIANGULAR BAND MATRIX , AS PRODUCED BY DECBND, BY BACK SUBSTITUTION. THE SOLUTIONS OF SEVERAL SYSTEMS WITH THE SAME COEFFICIENT MATRIX CAN BE OBTAINED BY SUCCESSIVE CALLS OF SOLBND.

EXAMPLE OF USE :
    THE FOLLOWING PROGRAM SOLVES THE SYSTEM OF SIMULTANEOUS EQUATIONS

```
  2 * X1  -    X2                                    = 1
  -    X1 + 2 * X2  -    X3                          = 0
            -    X2 + 2 * X3  -    X4                = 0
                      -    X3 + 2 * X4  -    X5      = 0
                                -    X4 + 2 * X5     = 1
```

```
"BEGIN""COMMENT" 730822, TEST DECBND, SOLBND AND DETERMBND;
   "PROCEDURE" DECBND(A, N, LW, RW, AUX, M, P); "CODE" 34320;
   "PROCEDURE" SOLBND(A, N, LW, RW, M, P, B); "CODE" 34071;
   "REAL""PROCEDURE" DETERMBND(A, N, LW, RW, SGNDET); "CODE" 34321;
   "INTEGER" I; "INTEGER""ARRAY" ROWIND[1 : 5];
   "ARRAY" BAND[1 : 13], MULT[1 : 4], RIGHT, AUX[1 : 5];
   "FOR" I:= 1 "STEP" 1 "UNTIL" 13 "DO"
   BAND[I]:= "IF" (I + 1) // 3 * 3 < I "THEN" 2 "ELSE" - 1;
   RIGHT[1]:= RIGHT[5]:= 1;
   "FOR" I:= 2, 3, 4 "DO" RIGHT[I]:= 0; AUX[2]:= "- 12;
   DECBND(BAND, 5, 1, 1, AUX, MULT, ROWIND);
   "IF" AUX[3] = 5 "THEN"
   "BEGIN" SOLBND(BAND, 5, 1, 1, MULT, ROWIND, RIGHT);
      OUTPUT(61, "("5(+2Z.4D2B), /"("DETERMINANT IS  ")" +.8D"+DD
      ")", (RIGHT[I], I:= 1 : 5), DETERMBND(BAND, 5, 1, 1, AUX[1]))
   "END"
"END"
```

DELIVERS :

```
 +1.0000   +1.0000   +1.0000   +1.0000   +1.0000
DETERMINANT IS  +.60000000"+01
```

SUBSECTION : DECSOLBND.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" DECSOLBND(A, N, LW, RW, AUX, B); "VALUE" N, LW, RW;
"INTEGER" N, LW, RW; "ARRAY" A, AUX, B;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

```
A, N, LW, RW, AUX  : SEE DECBND (SECTION : 3.1.2.1.1.1.1.1);
B                  : SEE SOLBND (THIS SECTION).
```

PROCEDURES USED :

```
VECVEC = CP34010,
ELMVEC = CP34020,
ICHVEC = CP34030.
```

REQUIRED CENTRAL MEMORY :

EXECUTION FIELD LENGTH : A REAL ARRAY OF N ELEMENTS AND A REAL ARRAY OF LW + 1 ELEMENTS ARE DECLARED.

RUNNING TIME :

(C1 * LW + C6) * (LW + RW + 1) * N;
THE CONSTANTS C1 AND C6 DEPEND UPON THE ARITHMETIC OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

DECSOLBND PERFORMS GAUSSIAN ELIMINATION IN THE SAME WAY AS DECBND , MEANWHILE ALSO CARRYING OUT THE ELIMINATION WITH THE GIVEN RIGHT HAND SIDE. THE SOLUTION OF THE ELIMINATED SYSTEM IS OBTAINED BY BACK SUBSTITUTION.

EXAMPLE OF USE :

THE PROGRAM

```
"BEGIN""COMMENT" 730822, TEST DECSOLBND AND DETERMBND;
   "PROCEDURE" DECSOLBND(A, N, LW, RW, AUX, B); "CODE" 34322;
   "REAL""PROCEDURE" DETERMBND(A, N, LW, RW, SGNDET); "CODE" 34321;
   "INTEGER" I;
   "ARRAY" BAND[1 : 13], RIGHT, AUX[1 : 5];

   "FOR" I:= 1 "STEP" 1 "UNTIL" 13 "DO"
   BAND[I]:= "IF" (I + 1) // 3 * 3 < I "THEN" 2 "ELSE" - 1;
   RIGHT[1]:= RIGHT[5]:= 1;
   "FOR" I:= 2, 3, 4 "DO" RIGHT[I]:= 0; AUX[2]:= "- 12;
   DECSOLBND(BAND, 5, 1, 1, AUX, RIGHT);
   "IF" AUX[3] = 5 "THEN"
   "BEGIN"
      OUTPUT(61, "("5(+ZZ.4D2B), /"("DETERMINANT IS  ")" +.8D"+DD
      ")", (RIGHT[I], I:= 1 : 5), DETERMBND(BAND, 5, 1, 1, AUX[1]))
   "END"
"END"
```

WHICH SOLVES THE SAME PROBLEM AS THE PROGRAM IN THE EXAMPLE OF USE OF SOLBND, DELIVERS :

```
 +1.0000   +1.0000   +1.0000   +1.0000   +1.0000
DETERMINANT IS  +.60000000"+01
```

SOURCE TEXT(S) :

```
"CODE" 34071;
"PROCEDURE" SOLBND(A, N, LW, RW, M, P, B); "VALUE" N, LW, RW;
"INTEGER" N, LW, RW; "INTEGER" "ARRAY" P; "ARRAY" A, B, M;
"BEGIN" "INTEGER" F, I, K, KK, W, W1, W2, SHIFT;
        "REAL" S;

        "REAL""PROCEDURE" VECVEC(A, B, C, D, E); "CODE" 34010;
        "PROCEDURE" ELMVEC(A, B, C, D, E, F); "CODE" 34020;

        F:= LW; SHIFT:= - LW; W1:= LW - 1;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN""IF" F < N "THEN" F:= F + 1; SHIFT:= SHIFT + W1;
            I:=P[K]; S:= B[I]; "IF" I ^= K "THEN"
            "BEGIN" B[I]:= B[K]; B[K]:= S "END";
            ELMVEC(K + 1, F, SHIFT, B, M, - S)
        "END";
        W1:= LW + RW; W:= W1 + 1; KK:= (N + 1) * W - W1; W2:= - 1;
        SHIFT:= N * W1;
        "FOR" K:= N "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" KK:= KK - W; SHIFT:= SHIFT - W1;
            "IF" W2 < W1 "THEN" W2:= W2 + 1;
            B[K]:= (B[K] - VECVEC(K + 1, K + W2, SHIFT, B, A)) / A[KK]
        "END"
    "END" SOLBND;
        "EOP"


"CODE" 34322;
"PROCEDURE" DECSOLBND(A, N, LW, RW, AUX, B); "VALUE" N, LW, RW;
"INTEGER" N, LW, RW; "ARRAY" A, B, AUX;
"BEGIN""INTEGER" I, J, K, KK, KK1, PK, IK, LW1, F, Q, W, W1, W2,IW,
        NRW, SHIFT, SDET;
        "REAL" R, S, EPS, MIN; "ARRAY" M[0:LW], V[1:N];

        "REAL""PROCEDURE" VECVEC(A, B, C, D, E); "CODE" 34010;
        "PROCEDURE" ELMVEC(A, B, C, D, E, F); "CODE" 34020;
        "PROCEDURE" ICHVEC(A, B, C, D); "CODE" 34030;

        F:= LW; SDET:= 1; W1:= LW + RW; W:= W1 + 1; W2:= W - 2; IW:= 0;
        NRW:= N - RW; LW1:= LW + 1; Q:= LW - 1;
        "FOR" I:= 2 "STEP" 1 "UNTIL" LW "DO"
        "BEGIN" Q:= Q - 1; IW:= IW + W1;
            "FOR" J:= IW - Q "STEP" 1 "UNTIL" IW "DO" A[J]:= 0
        "END";
                                                            "COMMENT"
```

```
        IW:= - W2; Q:= - LW;
        "FOR" I:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" IW:= IW + W; "IF" I <= LW1 "THEN" IW:= IW - 1;
            Q:= Q + W; "IF" I > NRW "THEN" Q:= Q - 1;
            V[I]:= SQRT(VECVEC(IW, Q, O, A, A))
        "END";
        EPS:= AUX[2]; MIN:= 1; KK:= - W1;
        "IF" F > NRW "THEN" W2:= W2 + NRW - F;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN""IF" F < N "THEN" F:= F + 1; IK:= KK:= KK + W;
            S:= ABS(A[KK]) / V[K]; PK:= K; KK1:= KK + 1;
            "FOR" I:= K + 1 "STEP" 1 "UNTIL" F "DO"
            "BEGIN" IK:= IK + W1; M[I - K]:= R:= A[IK]; A[IK]:= 0;
                R:= ABS(R) / V[I]; "IF" R > S "THEN"
                "BEGIN" S:= R; PK:= I "END"
            "END";
            "IF" S < MIN "THEN" MIN:= S; "IF" S < EPS "THEN"
            "BEGIN" AUX[3]:= K - 1; AUX[5]:= S; "GO TO" END "END";
            "IF" K + W2 >= N "THEN" W2:= W2 - 1; "IF" PK ^= K "THEN"
            "BEGIN" V[PK]:= V[K];
                PK:= PK - K; ICHVEC(KK1, KK1 + W2, PK * W1, A);
                SDET:= - SDET; R:= B[K]; B[K]:= B[PK + K];
                B[PK + K]:= R; R:= M[PK]; M[PK]:= A[KK]; A[KK]:= R
            "END"
            "ELSE" R:= A[KK]; IW:= KK1; LW1:= F - K;
            "IF" R < 0 "THEN" SDET:= - SDET;
            "FOR" I:= 1 "STEP" 1 "UNTIL" LW1 "DO"
            "BEGIN" M[I]:= S:= M[I] / R; IW:= IW + W1;
                ELMVEC(IW, IW + W2, KK1 - IW, A, A, - S);
                B[K + I]:= B[K + I] - B[K] * S
            "END"
        "END";
        AUX[3]:= N; AUX[5]:= MIN;
        KK:= (N + 1) * W - W1; W2:= - 1; SHIFT:= N * W1;
        "FOR" K:= N "STEP" - 1 "UNTIL" 1 "DO"
        "BEGIN" KK:= KK - W; SHIFT:= SHIFT - W1;
            "IF" W2 < W1 "THEN" W2:= W2 + 1;
            B[K]:= (B[K] - VECVEC(K + 1, K + W2, SHIFT, B, A)) / A[KK]
        "END";
    END: AUX[1]:= SDET
    "END" DECSOLBND;
        "EOP"
```

AUTHOR: W. HOFFMANN.

CONTRIBUTOR: J. C. P. BUS.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 731210.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS TWO PREPARATORY PROCEDURES FOR THE SOLUTION OF SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS WITH A TRIDIAGONAL MATRIX:
DECTRI PERFORMS A TRIANGULAR DECOMPOSITION OF A TRIDIAGONAL MATRIX.
DECTRIPIV PERFORMS A TRIANGULAR DECOMPOSITION OF A TRIDIAGONAL MATRIX, USING PARTIAL PIVOTING TO STABILIZE THE PROCESS.

KEYWORDS:

LU DECOMPOSITION,
TRIANGULAR DECOMPOSITION,
TRIDIAGONAL MATRIX.

SUBSECTION: DECTRI.

CALLING SEQUENCE:

THE HEADING OF THIS PROCEDURE IS:
```
"PROCEDURE" DECTRI(SUB, DIAG, SUPER, N, AUX);
"VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AUX;
```

THE MEANING OF THE FORMAL PARAMETERS IS:
```
SUB:     <ARRAY IDENTIFIER>;
         "ARRAY" SUB[1: N - 1];
         ENTRY:  THE SUBDIAGONAL OF THE GIVEN MATRIX T, SAY:
                 T[I + 1, I] SHOULD BE GIVEN IN SUB[I], I = 1,
                 ..., N - 1;
         EXIT:   SUPPOSE L DENOTES THE LOWER-BIDIAGONAL MATRIX, SUCH
                 THAT LU = T, FOR SOME UPPER-BIDIAGONAL MATRIX U,
                 WITH UNIT DIAGONAL ELEMENTS, THEN L[I + 1, I] WILL
                 BE DELIVERED IN SUB[I], I = 1, ..., AUX[3] - 1;
```

DIAG: <ARRAY IDENTIFIER>;
"ARRAY" DIAG[1: N];
ENTRY: THE DIAGONAL OF T;
EXIT: L[I, I] WILL BE DELIVERED IN DIAG[I], I = 1, ..., AUX[3];

SUPER: <ARRAY IDENTIFIER>;
"ARRAY" SUPER[1: N - 1];
ENTRY: THE SUPERDIAGONAL OF T: T[I, I + 1] SHOULD BE GIVEN IN SUPER[I], I = 1, ..., N - 1;
EXIT: U[I, I + 1] WILL BE DELIVERED IN SUPER[I], I = 1, ..., AUX[3] - 1;

N: <ARITHMETICAL EXPRESSION>;
THE ORDER OF THE MATRIX;

AUX: <ARRAY IDENTIFIER>;
"ARRAY" AUX[2:5];
ENTRY :
AUX[2]: A RELATIVE TOLERANCE; A REASONABLE CHOICE FOR THIS VALUE IS AN ESTIMATE OF THE RELATIVE PRECISION OF THE MATRIX ELEMENTS, HOWEVER, IT SHOULD NOT BE CHOSEN SMALLER THAN THE MACHINE PRECISION;
EXIT:
AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
AUX[5]: IF AUX[3] = N, THEN AUX[5] WILL EQUAL THE INFINITY-NORM OF THE MATRIX, ELSE AUX[5] IS SET EQUAL TO THE VALUE OF THAT ELEMENT WHICH CAUSES THE BREAKDOWN OF THE DECOMPOSITION.

PROCEDURES USED: NONE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE METHOD USED IN DECTRI YIELDS A LOWER-BIDIAGONAL MATRIX L AND A UNIT UPPER-BIDIAGONAL MATRIX U, SUCH THAT THE PRODUCT LU EQUALS THE GIVEN TRIDIAGONAL MATRIX; THE PROCESS IS TERMINATED IN THE K-TH STEP, IF THE MODULUS OF THE K-TH DIAGONAL ELEMENT IS SMALLER THAN A CERTAIN SMALL VALUE, WHICH IS GIVEN BY AUX[2] MULTIPLIED BY THE 1-NORM OF THE K-TH ROW; IN THIS CASE AUX[3] WILL BE GIVEN THE VALUE K - 1 AND AUX[5] WILL BE GIVEN THE VALUE OF THE K-TH DIAGONAL ELEMENT.

EXAMPLE OF USE: SEE DECSOLTRI (SECTION 3.1.2.1.1.1.2.3).

SUBSECTION: DECTRIPIV.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DECTRIPIV(SUB, DIAG, SUPER, N, AID, AUX, PIV);
    "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AID, AUX;
    "BOOLEAN" "ARRAY" PIV;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    SUB:    <ARRAY IDENTIFIER>;
            "ARRAY" SUB[1: N - 1];
            ENTRY:  THE  SUBDIAGONAL   OF  THE  GIVEN  MATRIX  T,  SAY;
                    T[I+1,I] SHOULD BE GIVEN IN SUB[I],I > 1,...,N - 2
            EXIT:   LET  T'  DENOTE THE MATRIX  T  WITH  PERMUTED ROWS;
                    SUPPOSE L DENOTES THE LOWER-BIDIAGONAL MATRIX, SUCH
                    THAT LU = T', FOR SOME UNIT UPPER-TRIANGULAR MATRIX
                    U, THEN  L[I + 1, I]  WILL BE DELIVERED IN  SUB[I],
                    I = 1, ..., AUX[3] - 1;  NOTE  THAT  U  HAS  TWO
                    CODIAGONALS, BECAUSE OF THE PARTIAL PIVOTING DURING
                    THE DECOMPOSITION;
    DIAG:   <ARRAY IDENTIFIER>;
            "ARRAY" DIAG[1: N];
            ENTRY:  THE DIAGONAL OF T;
            EXIT:   L[I,I] WILL BE DELIVERED IN DIAG[I],I=1,...,AUX[3];
    SUPER:  <ARRAY IDENTIFIER>;
            "ARRAY" SUPER[1: N - 1];
            ENTRY:  THE SUPERDIAGONAL OF T: T[I, I + 1] SHOULD BE GIVEN
                    IN SUPER[I], I = 1, ..., N - 1;
            EXIT:   U[I, I + 1]  WILL BE DELIVERED IN  SUPER[I], I = 1,
                    ..., AUX[3] - 1;
    N:      <ARITHMETICAL EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AID:    <ARRAY IDENTIFIER>;
            "ARRAY" AID[1: N - 2];
            EXIT:U[I,I+2] WILL BE DELIVERED IN AID[I],I=1,...,AUX[3]-2;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[2:5];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                    VALUE IS AN ESTIMATE OF  THE RELATIVE PRECISION  OF
                    THE  MATRIX ELEMENTS,  HOWEVER, IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            EXIT:
            AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
            AUX[5]: IF AUX[3] = N, THEN AUX[5] WILL EQUAL THE INFINITY-
                    NORM OF THE MATRIX,  ELSE  AUX[5]  IS SET  EQUAL TO
                    THE  VALUE  OF  THAT  ELEMENT  WHICH  CAUSES  THE
                    BREAKDOWN OF THE DECOMPOSITION.
    PIV:    <ARRAY IDENTIFIER>;
            "BOOLEAN""ARRAY" PIV[1 : N - 1];
            THE VALUE OF PIV[I] WILL BE TRUE IF THE I-TH AND (I + 1)-TH
            ROW ARE INTERCHANGED, I = 1, ..., MIN(AUX[3], N - 1),  ELSE
            PIV[I] WILL BE FALSE.
```

PROCEDURES USED: NONE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE METHOD USED IN DECTRIPIV YIELDS A LOWER-BIDIAGONAL MATRIX L AND A UNIT UPPER-TRIANGULAR MATRIX U WITH TWO CODIAGONALS, SUCH THAT THE PRODUCT LU EQUALS THE GIVEN TRIDIAGONAL MATRIX WITH PERMUTED ROWS; PARTIAL PIVOTING IS USED DURING THE TRIANGULAR DECOMPOSITION, I.E. THAT ELEMENT OF THE K-TH COLUMN OF L IS CHOSEN AS PIVOT IN THE K-TH STEP, WHOSE MODULUS DIVIDED BY THE 1-NORM OF THE CORRESPONDING ROW OF THE GIVEN MATRIX IS MAXIMAL; THE PROCESS IS TERMINATED IN THE K-TH STEP, IF THE MODULUS OF THE K-TH PIVOT ELEMENT IS LESS THAN A CERTAIN SMALL VALUE, WHICH IS GIVEN BY AUX[2] MULTIPLIED BY THE 1-NORM OF THE CORRESPONDING ROW; IN THIS CASE AUX[3] WILL BE GIVEN THE VALUE K - 1, AND AUX[5] WILL BE GIVEN THE VALUE OF THE K-TH PIVOT ELEMENT.

EXAMPLE OF USE: SEE SOLTRIPIV (SECTION 3.1.2.1.1.1.2.3).

SOURCE TEXTS:

```
"CODE" 34423;
  "PROCEDURE" DECTRI(SUB, DIAG, SUPER, N, AUX);
  "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AUX;
  "BEGIN" "INTEGER" I, N1;
      "REAL" D, R, S, U, NORM, NORM1, TOL;
      TOL:= AUX[2]; D:= DIAG[1]; R:= SUPER[1];
      NORM:= NORM1:= ABS(D) + ABS(R);
      "IF" ABS(D) <= NORM1 * TOL "THEN"
      "BEGIN" AUX[3]:= 0; AUX[5]:= D; "GOTO" EXIT "END";
      U:= SUPER[1]:= R / D; S:= SUB[1]; N1:= N - 1;
      "FOR" I:= 2 "STEP" 1 "UNTIL" N1 "DO"
      "BEGIN" D:= DIAG[I]; R:= SUPER[I];
         NORM1:= ABS(S) + ABS(D) + ABS(R);
         D:= DIAG[I]:= D - U * S;
         "IF" ABS(D) <= NORM1 * TOL "THEN"
         "BEGIN" AUX[3]:= I - 1; AUX[5]:= D; "GOTO" EXIT "END";
         U:= SUPER[I]:= R / D; S:= SUB[I];
         "IF" NORM1 > NORM "THEN" NORM:= NORM1
      "END";
      D:= DIAG[N]; NORM1:= ABS(D) + ABS(S);
      D:= DIAG[N]:= D - U * S;
      "IF" ABS(D) <= NORM1 * TOL "THEN"
      "BEGIN" AUX[3]:= N1; AUX[5]:= D; "GOTO" EXIT "END";
      "IF" NORM1 > NORM "THEN" NORM:= NORM1;
      AUX[3]:= N; AUX[5]:= NORM;
 EXIT:
 "END" DECTRI;
         "EOP"
```

```
"CODE" 34426;
  "PROCEDURE" DECTRIPIV(SUB, DIAG, SUPER, N, AID, AUX, PIV);
  "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AID, AUX;
  "BOOLEAN" "ARRAY" PIV;
  "BEGIN" "INTEGER" I, I1, N1, N2;
     "REAL" D, R, S, U, T, Q, V, W, NORM, NORM1, NORM2, TOL;
     TOL:= AUX[2]; D:= DIAG[1]; R:= SUPER[1];
     NORM:= NORM2:= ABS(D) + ABS(R); N2:= N - 2;
     "FOR" I:= 1 "STEP" 1 "UNTIL" N2 "DO"
     "BEGIN" I1:= I + 1; S:= SUB[I]; T:= DIAG[I1]; Q:= SUPER[I1];
        NORM1:= NORM2; NORM2:= ABS(S) + ABS(T) + ABS(Q);
        "IF" NORM2 > NORM "THEN" NORM:= NORM2;
        "IF" ABS(D) * NORM2 < ABS(S) * NORM1 "THEN"
        "BEGIN" "IF" ABS(S) <= TOL * NORM2 "THEN"
           "BEGIN" AUX[3]:= I - 1; AUX[5]:= S; "GOTO" EXIT "END";
           DIAG[I]:= S; U:= SUPER[I]:= T / S;
           V:= AID[I]:= Q / S; SUB[I]:= D;
           W:= SUPER[I1]:= -V * D; D:= DIAG[I1]:= R - U * D;
           R:= W; NORM2:= NORM1; PIV[I]:= "TRUE"
        "END" "ELSE"
        "BEGIN" "IF" ABS(D) <= TOL * NORM1 "THEN"
           "BEGIN" AUX[3]:= I - 1; AUX[5]:= D; "GOTO" EXIT "END";
           U:= SUPER[I]:= R / D; D:= DIAG[I1]:= T - U * S;
           AID[I]:= 0; PIV[I]:= "FALSE"; R:= Q
        "END"
     "END";
     N1:= N - 1; S:= SUB[N1]; T:= DIAG[N]; NORM1:= NORM2;
     NORM2:= ABS(S) + ABS(T); "IF" NORM2 > NORM "THEN" NORM:= NORM2;
     "IF" ABS(D) * NORM2 < ABS(S) * NORM1 "THEN"
     "BEGIN" "IF" ABS(S) <= TOL * NORM2 "THEN"
        "BEGIN" AUX[3]:= N2; AUX[5]:= S; "GOTO" EXIT "END";
        DIAG[N1]:= S; U:= SUPER[N1]:= T / S; SUB[N1]:= D;
        D:= DIAG[N]:= R - U * D; NORM2:= NORM1; PIV[N1]:= "TRUE"
     "END" "ELSE"
     "BEGIN" "IF" ABS(D) <= TOL * NORM1 "THEN"
        "BEGIN" AUX[3]:= N2; AUX[5]:= D; "GOTO" EXIT "END";
        U:= SUPER[N1]:= R / D; D:= DIAG[N]:= T - U * S;
        PIV[N1]:= "FALSE"
     "END";
     "IF" ABS(D) <= TOL * NORM2 "THEN"
     "BEGIN" AUX[3]:= N1; AUX[5]:= D; "GOTO" EXIT "END";
     AUX[3]:= N; AUX[5]:= NORM;
 EXIT:
 "END" DECTRIPIV;
        "EOP"
```

AUTHOR: W. HOFFMANN.

CONTRIBUTOR: J. C. P. BUS.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 731210.

BRIEF DESCRIPTION:

THIS SECTION CONTAINS FOUR PROCEDURES
FOR SOLVING A SYSTEM OF LINEAR EQUATIONS WITH A
TRIDIAGONAL MATRIX:
SOLTRI CALCULATES A SOLUTION BY FORWARD AND BACK SUBSTITUTION IF
THE TRIANGULAR DECOMPOSED FORM AS DELIVERED BY DECTRI IS GIVEN.
DECSOLTRI PERFORMS THE TRIANGULAR DECOMPOSITION OF THE GIVEN
MATRIX (NOT USING ANY PIVOTING STRATEGY DURING THE PROCESS) AND
CALCULATES THE SOLUTION BY FORWARD AND BACK SUBSTITUTION.
SOLTRIPIV CALCULATES A SOLUTION BY FORWARD AND BACK SUBTITUTION,IF
THE TRIANGULAR DECOMPOSED FORM AS DELIVERED BY DECTRIPIV IS GIVEN.
DECSOLTRIPIV PERFORMS THE TRIANGULAR DECOMPOSITION OF THE GIVEN
MATRIX (USING PARTIAL PIVOTING) AND CALCULATES THE SOLUTION BY
FORWARD AND BACK SUBSTITUTION.

KEYWORDS:

ALGEBRAIC EQUATIONS,
LINEAR SYSTEMS,
TRIDIAGONAL MATRIX,
FORWARD AND BACK SUBSTITUTION.

SUBSECTION: SOLTRI.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" SOLTRI(SUB, DIAG, SUPER, N, B);
    "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    SUB:      <ARRAY IDENTIFIER>;
              "ARRAY" SUB[1, N - 1];
              ENTRY : THE SUBDIAGONAL OF THE
              LOWER-BIDIAGONAL MATRIX, AS DELIVERED BY DECTRI (SECTION
              3.1.2.1.1.1.2.1);
    DIAG:     <ARRAY IDENTIFIER>;
              "ARRAY" DIAG[1:N];
              ENTRY : THE DIAGONAL OF THE LOWER-
              BIDIAGONAL MATRIX, AS DELIVERED BY DECTRI;
    SUPER:    <ARRAY IDENTIFIER>;
              "ARRAY" SUPER[1: N - 1];
              ENTRY : THE SUPERDIAGONAL OF THE
              UPPER-BIDIAGONAL MATRIX AS DELIVERED BY DECTRI;
    N:        <ARITHMETICAL EXPRESSION>;
              THE ORDER OF THE MATRIX;
    B:        <ARRAY IDENTIFIER>;
              "ARRAY" B[1:N];
              ENTRY: THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
              EXIT:  THE CALCULATED SOLUTION OF THE LINEAR SYSTEM.
```

PROCEDURES USED: NONE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

SOLTRI CALCULATES THE SOLUTION OF A LINEAR SYSTEM WITH A TRIDIAGONAL MATRIX. WITH FORWARD AND BACK SUBSTITUTION; THE TRIANGULARLY DECOMPOSED FORM OF THE MATRIX, AS PRODUCED BY DECTRI (SECTION 3.1.2.1.1.1.2.1), SHOULD BE GIVEN; ONE CALL OF DECTRI FOLLOWED BY SEVERAL CALLS OF SOLTRI MAY BE USED TO SOLVE SEVERAL LINEAR SYSTEMS HAVING THE SAME TRIDIAGONAL MATRIX, BUT DIFFERENT RIGHT-HAND SIDES.

EXAMPLE OF USE: SEE DECSOLTRI (THIS SECTION).

SUBSECTION: DECSOLTRI.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DECSOLTRI(SUB, DIAG, SUPER, N, AUX, B);
    "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AUX, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    SUB:    <ARRAY IDENTIFIER>;
            "ARRAY" SUB[1: N - 1];
            ENTRY:  THE  SUBDIAGONAL   OF  THE  GIVEN  MATRIX  T,  SAY:
                    T[I + 1, I]  SHOULD   BE  GIVEN  IN  SUB[I], I = 1,
                    ..., N - 1;
            EXIT:   SUPPOSE L DENOTES THE LOWER-BIDIAGONAL MATRIX, SUCH
                    THAT  LU = T,  FOR SOME UPPER-BIDIAGONAL MATRIX  U,
                    WITH UNIT DIAGONAL ELEMENTS, THEN L[I + 1, I]  WILL
                    BE DELIVERED IN SUB[I], I = 1, ..., AUX[3] - 1;
    DIAG:   <ARRAY IDENTIFIER>;
            "ARRAY" DIAG[1: N];
            ENTRY:  THE DIAGONAL OF T;
            EXIT:   L[I, I] WILL BE DELIVERED IN  DIAG[I],  I = 1, ...,
                    AUX[3];
    SUPER:  <ARRAY IDENTIFIER>;
            "ARRAY" SUPER[1: N - 1];
            ENTRY:  THE SUPERDIAGONAL OF T; T[I, I + 1] SHOULD BE GIVEN
                    IN SUPER[I], I = 1, ..., N - 1;
            EXIT:   U[I, I + 1]  WILL BE DELIVERED IN  SUPER[I], I = 1,
                    ..., AUX[3] - 1;
    N:      <ARITHMETICAL EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[2:5];
            ENTRY :
            AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                    VALUE IS AN ESTIMATE OF  THE RELATIVE PRECISION  OF
                    THE  MATRIX ELEMENTS,  HOWEVER, IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            EXIT :
            AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
            AUX[5]: IF AUX[3] = N, THEN AUX[5] WILL EQUAL THE INFINITY-
                    NORM OF THE MATRIX  (SEE  SECTION  3.1.2.1.1.1.2.1,
                    SUBSECTION DECTRI);
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   IF AUX[3] = N,  THEN  THE  SOLUTION  OF  THE LINEAR
                    SYSTEM  IS  OVERWRITTEN  ON  B,  ELSE   B   REMAINS
                    UNALTERED.
```

PROCEDURES USED:

DECTRI = CP34423,
SOLTRI = CP34424.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

DECSOLTRI CALCULATES THE SOLUTION OF A LINEAR SYSTEM WITH A TRIDIAGONAL MATRIX; THE TRIANGULAR DECOMPOSITION IS DONE BY CALLING DECTRI (SECTION 3.1.2.1.1.1.2.1) AND THE FORWARD AND BACK SUBSTITUTION BY CALLING SOLTRI (THIS SECTION); IF AUX[3] < N, THEN THE EFFECT OF DECSOLTRI IS MERELY THAT OF DECTRI.

EXAMPLE OF USE:

LET T BE A TRIDIAGONAL MATRIX WITH SUBDIAGONAL AND SUPERDIAGONAL ELEMENTS I * 2 AND I RESPECTIVELY (I = 1, ..., N - 1), AND DIAGONAL ELEMENTS I + 10 (I = 1, ..., N); LET B BE THE SECOND COLUMN OF T; THEN THE SOLUTION OF THE LINEAR SYSTEM TX = B IS GIVEN BY THE SECOND UNIT VECTOR; BY THE FOLLOWING PROGRAM WE MAY SOLVE THIS SYSTEM AND PRINT THE ERROR IN THE CALCULATED SOLUTION.

```
"BEGIN"
    "PROCEDURE" DECSOLTRI(L, D, U, N, A, B); "CODE" 34425;
    "REAL" "PROCEDURE" VECVEC(L, U, S, A, B); "CODE" 34010;
    "INTEGER" I;
    "ARRAY" D, SUB, SUPER, B[1:30], AUX[2:5];
    "FOR" I:= 1 "STEP" 1 "UNTIL" 30 "DO"
    "BEGIN" SUB[I]:= I * 2; SUPER[I]:= I; D[I]:= I + 10;
        B[I]:= 0
    "END"; B[1]:= 1; B[2]:= 12; B[3]:= 4;
    AUX[2]:= "-14;
    DECSOLTRI(SUB, D, SUPER, 30, AUX, B);
    OUTPUT(71, "("/,"("AUX[3] AND AUX[5]:")",2(/,N)")",
    AUX[3], AUX[5]);
    B[2]:= B[2] - 1;
    OUTPUT(71, "("//"("ERROR IN THE SOLUTION: ")", N")",
    SQRT(VECVEC(1, 30, 0, B, B)))
"END"
```

RESULTS:

```
AUX[3] AND AUX[5]:
+3.0000000000000"+001
+1.2400000000000"+002

ERROR IN THE SOLUTION: +0.0000000000000"+000
```

SUBSECTION: SOLTRIPIV.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" SOLTRIPIV(SUB, DIAG, SUPER, N, AID, PIV, B);
    "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AID, B;
    "BOOLEAN" "ARRAY" PIV;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    SUB:    <ARRAY IDENTIFIER>;
            "ARRAY" SUB[1, N - 1];
            ENTRY : THE SUBDIAGONAL OF THE
            LOWER-BIDIAGONAL MATRIX, AS DELIVERED BY DECTRIPIV (SECTION
            3.1.2.1.1.1.2.1);
    DIAG:   <ARRAY IDENTIFIER>;
            "ARRAY" DIAG[1:N];
            ENTRY: THE DIAGONAL OF THE LOWER-
            BIDIAGONAL MATRIX, AS DELIVERED BY DECTRIPIV;
    SUPER:  <ARRAY IDENTIFIER>;
            "ARRAY" SUPER[1: N - 1];
            ENTRY : THE FIRST CODIAGONAL OF
            THE UPPER-TRIANGULAR MATRIX AS DELIVERED BY DECTRIPIV;
    N:      <ARITHMETICAL EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AID:    <ARRAY IDENTIFIER>;
            "ARRAY" AID[1: N - 2];
            ENTRY:  THE SECOND CODIAGONAL OF
            THE UPPER-TRIANGULAR MATRIX AS DELIVERED BY DECTRIPIV;
    PIV:    <ARRAY IDENTIFIER>;
            "BOOLEAN""ARRAY" PIV[1: N-1];
            ENTRY:  THE PIVOT-
            INFORMATION AS DELIVERED BY DECTRIPIV;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   THE CALCULATED SOLUTION OF THE LINEAR SYSTEM.
```

PROCEDURES USED: NONE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

SOLTRIPIV CALCULATES THE SOLUTION OF A LINEAR SYSTEM WITH A TRIDIAGONAL MATRIX, WITH FORWARD AND BACK SUBSTITUTION; THE TRIANGULARLY DECOMPOSED FORM OF THE MATRIX,AS PRODUCED BY DECTRIPIV (SECTION 3.1.2.1.1.1.2.1), SHOULD BE GIVEN; ONE CALL OF DECTRIPIV FOLLOWED BY SEVERAL CALLS OF SOLTRIPIV MAY BE USED TO SOLVE SEVERAL LINEAR SYSTEMS HAVING THE SAME TRIDIAGONAL MATRIX, BUT DIFFERENT RIGHT-HAND SIDES.

EXAMPLE OF USE:

LET T BE THE MATRIX AS GIVEN IN THE EXAMPLE OF USE OF DECSOLTRI (THIS SECTION) AND LET B1 AND B2 BE THE SECOND AND THIRD COLUMN OF T, THEN THE SOLUTIONS OF THE LINEAR SYSTEMS TX = B1 AND TX = B2 ARE GIVEN BY THE SECOND AND THIRD UNIT VECTOR RESPECTIVELY: IN THE FOLLOWING PROGRAM THESE SYSTEMS ARE SOLVED AND THE ERRORS IN THE CALCULATED SOLUTIONS ARE PRINTED.

```
"BEGIN"
    "PROCEDURE" DECTRIPIV(L, D, U, N, A, AX, P); "CODE" 34426;
    "PROCEDURE" SOLTRIPIV(L, D, U, N, A, P, B); "CODE" 34427;
    "REAL" "PROCEDURE" VECVEC(L, U, S, A, B); "CODE" 34010;
    "INTEGER" I;
    "ARRAY" D, SUB, SUPER, AID, B1, B2[1:30], AUX[2:5];
    "BOOLEAN" "ARRAY" PIV[1:29];
    "FOR" I:= 1 "STEP" 1 "UNTIL" 30 "DO"
    "BEGIN" SUB[I]:= I * 2; SUPER[I]:= I; D[I]:= I + 10;
        B1[I]:= B2[I]:= 0
    "END"; B1[1]:= 1; B1[2]:= 12; B1[3]:= 4;
    B2[2]:= 2; B2[3]:= 13; B2[4]:= 6;
    AUX[2]:= "-14;
    DECTRIPIV(SUB, D, SUPER, 30, AID, AUX, PIV);
    SOLTRIPIV(SUB, D, SUPER, 30, AID, PIV, B1);
    SOLTRIPIV(SUB, D, SUPER, 30, AID, PIV, B2);
    OUTPUT(71, "("/","("AUX[3] AND AUX[5]:")",2(/,N)")",
    AUX[3], AUX[5]);
    B1[2]:= B1[2] - 1; B2[3]:= B2[3] - 1;
    OUTPUT(71, "("//"("ERROR IN B1: ")",N,"("ERROR IN B2: ")",N")",
    SQRT(VECVEC(1, 30, 0, B1, B1)), SQRT(VECVEC(1, 30, 0, B2, B2)))
"END"
```

RESULTS:

```
AUX[3] AND AUX[5]:
+3.0000000000000"+001
+1.2400000000000"+002

ERROR IN B1: +0.0000000000000"+000
ERROR IN B2: +0.0000000000000"+000
```

SUBSECTION: DECSOLTRIPIV.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DECSOLTRIPIV(SUB, DIAG, SUPER, N, AUX, B);
    "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AUX, B;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    SUB:    <ARRAY IDENTIFIER>;
            "ARRAY" SUB[1: N - 1];
            ENTRY:  THE  SUBDIAGONAL  OF  THE  GIVEN  MATRIX  T,  SAY:
                    T[I + 1, I]  SHOULD  BE  GIVEN  IN  SUB[I], I = 1,
                    ..., N - 1;
            EXIT:   THE ELEMENTS OF SUB WILL BE OVERWRITTEN;
    DIAG:   <ARRAY IDENTIFIER>;
            "ARRAY" DIAG[1: N];
            ENTRY:  THE DIAGONAL OF T;
            EXIT:   THE ELEMENTS OF DIAG WILL BE OVERWRITTEN;
    SUPER:  <ARRAY IDENTIFIER>;
            "ARRAY" SUPER[1: N - 1];
            ENTRY:  THE SUPERDIAGONAL OF T: T[I, I + 1] SHOULD BE GIVEN
                    IN SUPER[I], I = 1, ..., N - 1;
            EXIT:   THE ELEMENTS OF SUPER WILL BE OVERWRITTEN;
    N:      <ARITHMETICAL EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[2:5];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE:  A REASONABLE CHOICE FOR THIS
                    VALUE IS AN ESTIMATE OF  THE RELATIVE PRECISION  OF
                    THE  MATRIX ELEMENTS,  HOWEVER, IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            EXIT:
            AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
            AUX[5]: IF AUX[3] = N, THEN AUX[5] WILL EQUAL THE INFINITY-
                    NORM OF THE MATRIX  (SEE SECTION  3.1.2.1.1.1.2.1.,
                    SUBSECTION DECTRIPIV);
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1 : N];
            ENTRY:  THE RIGHT-HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   IF  AUX[3] = N,  THEN  THE SOLUTION  OF  THE LINEAR
                    SYSTEM WILL BE OVERWRITTEN ON B, ELSE B WILL REMAIN
                    UNALTERED.
```

PROCEDURES USED: NONE.

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: ONE AUXILIARY ARRAY OF TYPE BOOLEAN AND ORDER N IS DECLARED IN DECSOLTRIPIV;

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

ONE CALL OF DECSOLTRIPIV IS EQUIVALENT WITH CALLING CONSECUTIVELY DECTRIPIV (SECTION 3.1.2.1.1.1.2.1) AND SOLTRIPIV (THIS SECTION); HOWEVER, DECSOLTRIPIV DOES NOT MAKE USE OF DECTRIPIV AND SOLTRIPIV, TO SAVE MEMORY SPACE AND TIME; THIS IS ONLY TRUE IN THE CASE THAT LINEAR SYSTEMS WITH DIFFERENT MATRICES HAVE TO BE SOLVED; IF AUX[3] < N THEN DECSOLTRIPIV IS TERMINATED PREMATURELY (SEE DECTRIPIV IN SECTION 3.1.2.1.1.1.2.1).

EXAMPLE OF USE:

THE SAME LINEAR SYSTEM AS GIVEN IN THE EXAMPLE OF USE OF DECSOLTRI MAY BE SOLVED WITH DECSOLTRIPIV BY THE FOLLOWING PROGRAM:

```
"BEGIN"
    "PROCEDURE" DECSOLTRIPIV(L, D, U, N, A, B); "CODE" 34428;
    "REAL" "PROCEDURE" VECVEC(L, U, S, A, B); "CODE" 34010;
    "INTEGER" I;
    "ARRAY" D, SUB, SUPER, B[1:30], AUX[2:5];
    "FOR" I:= 1 "STEP" 1 "UNTIL" 30 "DO"
    "BEGIN" SUB[I]:= I * 2; SUPER[I]:= I; D[I]:= I + 10;
        B[I]:= 0
    "END"; B[1]:= 1; B[2]:= 12; B[3]:= 4;
    AUX[2]:= "-14;
    DECSOLTRIPIV(SUB, D, SUPER, 30, AUX, B);
    OUTPUT(71, "("/,"("AUX[3] AND AUX[5]:")",2(/,N)")",
    AUX[3], AUX[5]);
    B[2]:= B[2] - 1;
    OUTPUT(71, "("//"("ERROR IN THE SOLUTION: ")", N")",
    SQRT(VECVEC(1, 30, 0, B, B)))
"END"
```

RESULTS:

```
AUX[3] AND AUX[5]:
+3.0000000000000"+001
+1.2400000000000"+002

ERROR IN THE SOLUTION: +0.0000000000000"+000
```

SOURCE TEXTS:

```
"CODE" 34424;
  "PROCEDURE" SOLTRI(SUB, DIAG, SUPER, N, B);
  "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, B;
  "BEGIN" "INTEGER" I;
     "REAL" R;
     R:= B[1]:= B[1] / DIAG[1];
     "FOR" I:= 2 "STEP" 1 "UNTIL" N "DO"
     R:= B[I]:= (B[I] - SUB[I - 1] * R) / DIAG[I];
     "FOR" I:= N - 1 "STEP" -1 "UNTIL" 1 "DO"
     R:= B[I] := B[I] - SUPER[I] * R
  "END" SOLTRI;
        "EOP"


"CODE" 34425;
 "PROCEDURE" DECSOLTRI(SUB, DIAG, SUPER, N, AUX, B);
 "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AUX, B;
 "BEGIN" "PROCEDURE" DECTRI(SUB, DIAG, SUPER, N, AUX); "CODE" 34423;
         "PROCEDURE" SOLTRI( SUB, DIAG, SUPER, N, B);  "CODE" 34424;
         DECTRI(SUB, DIAG, SUPER, N, AUX); "IF" AUX[3]= N "THEN"
         SOLTRI(SUB, DIAG, SUPER, N, B)
 "END" DECSOLTRI;
        "EOP"


"CODE" 34427;
  "PROCEDURE" SOLTRIPIV(SUB, DIAG, SUPER, N, AID, PIV, B);
  "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AID, B;
  "BOOLEAN" "ARRAY" PIV;
  "BEGIN" "INTEGER" I, N1;
     "REAL" BI, BI1, R, S, T;
     N1:= N - 1;
     "FOR" I:= 1 "STEP" 1 "UNTIL" N1 "DO"
     "BEGIN" "IF" PIV[I] "THEN"
        "BEGIN" BI:= B[I+1]; BI1:= B[I] "END"
        "ELSE"
        "BEGIN" BI:= B[I]; BI1:= B[I+1] "END";
        R:= B[I]:= BI / DIAG[I];
        B[I+1]:= BI1 - SUB[I] * R
     "END";
     R:= B[N]:= B[N] / DIAG[N];
     T:= B[N1]:= B[N1] - SUPER[N1] * R;
     "FOR" I:= N - 2 "STEP" -1 "UNTIL" 1 "DO"
     "BEGIN" S:= R; R:= T; T:= B[I]:= B[I] - SUPER[I] * R -
        ("IF" PIV[I] "THEN" AID[I] * S "ELSE" 0)
     "END"
 "END" SOLTRIPIV;
        "EOP"
```

```
"CODE" 34428;
  "PROCEDURE" DECSOLTRIPIV(SUB, DIAG, SUPER, N, AUX, B);
  "VALUE" N; "INTEGER" N; "ARRAY" SUB, DIAG, SUPER, AUX, B;
  "BEGIN" "INTEGER" I, I1, N1, N2;
     "REAL" D, R, S, U, T, Q, V, W, NORM, NORM1, NORM2, TOL,
     BI, BI1, BI2;
     "BOOLEAN" "ARRAY" PIV[1:N];
     TOL:= AUX[2]; D:= DIAG[1]; R:= SUPER[1]; BI:= B[1];
     NORM:= NORM2:= ABS(D) + ABS(R); N2:= N - 2;
     "FOR" I:= 1 "STEP" 1 "UNTIL" N2 "DO"
     "BEGIN" I1:= I + 1; S:= SUB[I]; T:= DIAG[I1]; Q:= SUPER[I1];
        BI1:= B[I1];
        NORM1:= NORM2; NORM2:= ABS(S) + ABS(T) + ABS(Q);
        "IF" NORM2 > NORM "THEN" NORM:= NORM2;
        "IF" ABS(D) * NORM2 < ABS(S) * NORM1 "THEN"
        "BEGIN" "IF" ABS(S) <= TOL * NORM2 "THEN"
           "BEGIN" AUX[3]:= I - 1; AUX[5]:= S; "GOTO" EXIT "END";
           U:= SUPER[I]:= T / S; BI1:= B[I]:= BI1 / S;
           BI:= BI - BI1 * D; V:= SUB[I]:= Q / S;
           W:= SUPER[I1]:= -V * D; D:= DIAG[I1]:= R - U * D;
           R:= W; NORM2:= NORM1; PIV[I]:= "TRUE"
        "END" "ELSE"
        "BEGIN" "IF" ABS(D) <= TOL * NORM1 "THEN"
           "BEGIN" AUX[3]:= I - 1; AUX[5]:= D; "GOTO" EXIT "END";
           U:= SUPER[I]:= R / D; BI:= B[I]:= BI / D;
           BI:= BI1 - BI * S; D:= DIAG[I1]:= T - U * S;
           PIV[I]:= "FALSE"; R:= Q
        "END"
     "END";
     N1:= N - 1; S:= SUB[N1]; T:= DIAG[N]; NORM1:= NORM2; BI1:= B[N];
     NORM2:= ABS(S) + ABS(T); "IF" NORM2 > NORM "THEN" NORM:= NORM2;
     "IF" ABS(D) * NORM2 < ABS(S) * NORM1 "THEN"
     "BEGIN" "IF" ABS(S) <= TOL * NORM2 "THEN"
        "BEGIN" AUX[3]:= N2; AUX[5]:= S; "GOTO" EXIT "END";
        U:= SUPER[N1]:= T / S; BI1:= B[N1]:= BI1 / S;
        BI:= BI - BI1 * D; D:= R - U * D; NORM2:= NORM1
     "END" "ELSE"
     "BEGIN" "IF" ABS(D) <= TOL * NORM1 "THEN"
        "BEGIN" AUX[3]:= N2; AUX[5]:= D; "GOTO" EXIT "END";
        U:= SUPER[N1]:= R / D; BI:= B[N1]:= BI / D;
        BI:= BI1 - BI * S; D:= T - U * S
     "END";
     "IF" ABS(D) <= TOL * NORM2 "THEN"
     "BEGIN" AUX[3]:= N1; AUX[5]:= D; "GOTO" EXIT "END";
     AUX[3]:= N; AUX[5]:= NORM;
     BI1:= B[N]:= BI / D; BI:= B[N1]:= B[N1] - SUPER[N1] * BI1;
     "FOR" I:= N - 2 "STEP" -1 "UNTIL" 1 "DO"
     "BEGIN" BI2:= BI1; BI1:= BI;
         BI:= B[I]:= B[I] - SUPER[I] * BI1 -
         ("IF" PIV[I] "THEN" SUB[I] * BI2 "ELSE" 0)
     "END";
 EXIT:
 "END" DECSOLTRIPIV;
        "EOP"
```

AUTHOR : T.J. DEKKER.

CONTRIBUTOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 731001.

BRIEF DESCRIPTION :
THIS SECTION CONTAINS THE PROCEDURE CHLDECBND FOR THE CHOLESKY DECOMPOSITION OF A SYMMETRIC POSITIVE DEFINITE BAND MATRIX.

KEYWORDS :
LINEAR EQUATIONS,
CHOLESKY DECOMPOSITION,
SYMMETRIC POSITIVE DEFINITE MATRIX,
BAND MATRIX.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" CHLDECBND(A, N, W, AUX); "VALUE" N, W; "INTEGER" N, W;
"ARRAY" A, AUX;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

```
A   :  <ARRAY IDENTIFIER>;
       "ARRAY" A[1 : W * (N - 1) + N];
       ENTRY  : A CONTAINS COLUMNWISE (I.E. THE(I,J)-TH
       ELEMENT OF THE MATRIX IS A[(J-1)*W+I], J=1,...,N,
       I=MAX(1,J-W),...,J) THE UPPER-TRIANGULAR BAND
       ELEMENTS OF THE SYMMETRIC BAND MATRIX;
       EXIT   : THE BAND ELEMENTS OF THE CHOLESKY
       MATRIX, WHICH IS AN UPPER-TRIANGULAR BAND MATRIX WITH
       W SUPERDIAGONALS, ARE DELIVERED COLUMNWISE IN A;
N   :  <ARITHMETIC EXPRESSION>;
       ORDER OF THE BAND MATRIX;
W   :  <ARITHMETIC EXPRESSION>;
       NUMBER OF  SUPERDIAGONALS OF THE MATRIX;
AUX :  <ARRAY IDENTIFIER>;
       "ARRAY" AUX[2 : 3];
       ENTRY  : AUX[2] IS A RELATIVE TOLERANCE TO CONTROL THE
        CALCULATION OF THE DIAGONAL ELEMENTS OF THE
        CHOLESKY MATRIX (SEE METHOD AND PERFORMANCE);
       NORMAL EXIT  :
        AUX[3] = N;
       ABNORMAL EXIT :
        AUX[3] = K - 1, WHERE K IS THE INDEX OF THE DIAGONAL
        ELEMENT OF THE CHOLESKY MATRIX THAT CANNOT BE
        CALCULATED.
```

PROCEDURES USED :

VECVEC = CP34010.

RUNNING TIME :

(C1 * W + C2) * W * N;
THE CONSTANTS C1 AND C2 DEPEND UPON THE
ARITHMETIC OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

CHLDECBND PERFORMS THE CHOLESKY DECOMPOSITION OF A SYMMETRIC POSITIVE DEFINITE MATRIX, WHOSE NON-ZERO ELEMENTS ARE IN BAND FORM, AND WHOSE UPPER-TRIANGULAR BAND ELEMENTS ARE STORED COLUMNWISE IN IONAL ARRAY.
THE METHOD USED IS CHOLESKY'S SQUARE ROOT METHOD. IF THE GIVEN MATRIX IS POSITIVE DEFINITE, THEN THIS METHOD YIELDS AN UPPER-TRIANGULAR BAND MATRIX, THE CHOLESKY MATRIX. THE NUMBER OF NON-ZERO SUPERDIAGONALS OF THE GIVEN MATRIX AND ITS CHOLESKY MATRIX ARE EQUAL. THE PROCESS IS COMPLETED IN N STAGES, AT EACH STAGE PRODUCING A ROW OF THE CHOLESKY MATRIX. HOWEVER, THE PROCESS IS DISCONTINUED IF AT SOME STAGE, SAY K, THE K-TH DIAGONAL ELEMENT OF THE GIVEN MATRIX MINUS THE SUM OF SQUARES OF THE SUPERDIAGONAL ELEMENTS OF THE K-TH COLUMN OF THE CHOLESKY MATRIX (THE SQUARE ROOT OF THIS QUANTITY BEING THE K-TH DIAGONAL ELEMENT OF THE CHOLESKY MATRIX) IS EITHER NEGATIVE OR LESS THAN A GIVEN RELATIVE TOLERANCE (AUX[2]) TIMES THE MAXIMAL DIAGONAL ELEMENT THE GIVEN MATRIX. IN THIS CASE THE GIVEN MATRIX, POSSIBLY MODIFIED BY ROUNDING ERRORS, IS NOT POSITIVE DEFINITE. THIS IS INDICATED IN AUX[3], BY WHICH THE VALUE K - 1 IS DELIVERED. IF THE DECOMPOSITION IS CARRIED OUT FULLY, AUX[3] BECOMES N.
THE PROCEDURE DELIVERS THE BAND ELEMENTS OF THE CHOLESKY MATRIX.
SEE ALSO REF [1], SECTION 222.

REFERENCE :

[1] DEKKER, T.J. :
ALGOL 60 PROCEDURES IN NUMERICAL ALGEBRA, PART 1,
MC TRACT 22, 1968, MATHEMATISCH CENTRUM, AMSTERDAM.

EXAMPLE OF USE :

SEE EXAMPLE OF USE OF CHLSOLBND.

SOURCE TEXT(S) :

```
"CODE" 34330;
"PROCEDURE" CHLDECBND(A, N, W, AUX); "VALUE" N, W; "INTEGER" N, W;
"ARRAY" A, AUX;
"BEGIN" "INTEGER" J, K, JMAX, KK, KJ, W1, START;
        "REAL" R, EPS, MAX;
        "REAL""PROCEDURE" VECVEC(L, U, S, A, B); "CODE" 34010;
        MAX:= 0; KK:= - W; W1:= W + 1;
        "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" KK:= KK + W1; "IF" A[KK] > MAX "THEN" MAX:= A[KK]"END";
        JMAX:= W; W1:= W + 1; KK:= - W; EPS:= AUX[2] * MAX;
        "FOR" K:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN""IF" K + W > N "THEN" JMAX:= JMAX - 1; KK:= KK + W1;
            START:= KK - K + 1;
            R:= A[KK] - VECVEC("IF" K <= W1 "THEN" START "ELSE" KK - W,
            KK - 1, 0, A, A); "IF" R <= EPS "THEN"
            "BEGIN" AUX[3]:= K - 1; "GO TO" END "END";
            A[KK]:= R:= SQRT(R); KJ:= KK;
            "FOR" J:= 1 "STEP" 1 "UNTIL" JMAX "DO"
            "BEGIN" KJ:= KJ + W;
                A[KJ]:= (A[KJ] - VECVEC("IF" K + J <= W1 "THEN" START
                "ELSE" KK - W + J, KK - 1, KJ - KK, A, A)) / R
            "END"
        "END";
        AUX[3]:= N;
    END:
    "END" CHLDECBND;
        "EOP"
```

CONTRIBUTOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 731001.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS ONE PROCEDURE, CHLDETERMBND, FOR THE CALCULATION OF THE DETERMINANT OF A SYMMETRIC POSITIVE DEFINITE BAND MATRIX.

KEY WORDS :

DETERMINANT,
SYMMETRIC POSITIVE DEFINITE MATRIX,
BAND MATRIX.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"REAL""PROCEDURE" CHLDETERMBND(A, N, W); "VALUE" N, W;
"INTEGER" N, W; "ARRAY" A;
```

CHLDETERMBND DELIVERS THE DETERMINANT OF THE SYMMETRIC POSITIVE DEFINITE BAND MATRIX WHOSE CHOLESKY MATRIX IS STORED IN A.

THE MEANING OF THE FORMAL PARAMETERS IS :

A, N, W : SEE CALLING SEQUENCE OF CHLDECBND (SECTION 3.1.2.1.1.2.1.1.); THE CONTENTS OF A ARE AS PRODUCED BY CHLDECBND OR CHLDECSOLBND (SECTION 3.1.2.1.1.2.1.3.).

PROCEDURES USED : NONE.

RUNNING TIME : PROPORTIONAL TO N.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

CHLDETERMBND CAN BE CALLED AFTER CHLDECBND OR CHLDECSOLBND ONLY IF THE CHOLESKY DECOMPOSITION WAS SUCCESSFUL, I.E. IF AUX[3] = N.
THE FUNCTION VALUE OF CHLDETERMBND IS THE SQUARE OF THE DETERMINANT OF THE CHOLESKY MATRIX.
CHLDETERMBND SHOULD NOT BE CALLED WHEN OVERFLOW CAN BE EXPECTED.

EXAMPLE OF USE :

SEE EXAMPLES OF USE OF CHLSOLBND AND CHLDECSOLBND.

SOURCE TEXT(S) :

```
"CODE" 34331;
"REAL""PROCEDURE" CHLDETERMBND(A, N, W); "VALUE" N, W; "INTEGER" N,W;
"ARRAY" A;
"BEGIN""INTEGER" J, KK, W1; "REAL" P;
        W1:= W + 1; KK:= - W; P:= 1;
        "FOR" J:= 1 "STEP" 1 "UNTIL" N "DO"
        "BEGIN" KK:= KK + W1; P:= A[KK] * P "END";
        CHLDETERMBND:= P * P
    "END" CHLDETERMBND;
        "EOP"
```

AUTHOR : T.J. DEKKER.

CONTRIBUTOR : J. KOK.

INSTITUTE : MATHEMATICAL CENTRE.

RECEIVED : 731001.

BRIEF DESCRIPTION :

THIS SECTION CONTAINS TWO PROCEDURES.
A) CHLSOLBND, FOR THE SOLUTION OF ONE OR MORE SYSTEMS OF LINEAR EQUATIONS WITH THE SAME COEFFICIENT MATRIX, WHICH IS SYMMETRIC, POSITIVE DEFINITE AND IN BANDFORM, PROVIDED THAT THIS MATRIX HAS BEEN DECOMPOSED BY A CALL OF THE PROCEDURE CHLDECBND (SECTION 3.1.2.1.1.2.1.1.).
B) CHLDECSOLBND, FOR THE SOLUTION OF ONE SYSTEM OF LINEAR EQUATIONS BY CHOLESKY'S SQUARE ROOT METHOD, PROVIDED THAT THE SYMMETRIC POSITIVE DEFINITE COEFFICIENT MATRIX IS IN BAND FORM AND IS STORED COLUMNWISE IN A ONE-DIMENSIONAL ARRAY.

KEYWORDS :

LINEAR EQUATIONS,
CHOLESKY DECOMPOSITION,
SYMMETRIC POSITIVE DEFINITE MATRIX,
BAND MATRIX.

SUBSECTION : CHLSOLBND.

CALLING SEQUENCE :

THE HEADING OF THE PROCEDURE IS :

```
"PROCEDURE" CHLSOLBND(A, N, W, B); "VALUE" N, W; "INTEGER" N, W;
"ARRAY" A, B;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

```
A, N, W :   SEE CALLING SEQUENCE OF CHLDECBND,
            THE CONTENTS OF THE ARRAY A ARE AS PRODUCED BY
            CHLDECBND;
B   :       <ARRAY IDENTIFIER>;
            "ARRAY" B[1 : N];
            ENTRY : THE RIGHT HAND SIDE OF THE SYSTEM OF LINEAR
            EQUATIONS;
            EXIT  :  THE SOLUTION OF THE SYSTEM.
```

PROCEDURES USED :

VECVEC = CP34010,
SCAPRD1= CP34017.

RUNNING TIME:

(C3 * W + C4) * N;
THE CONSTANTS C3 AND C4 DEPEND UPON THE ARITHMETIC OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

THE PROCEDURE CHLSOLBND CALCULATES THE SOLUTION OF A SYSTEM OF LINEAR EQUATIONS, PROVIDED THAT THE COEFFICIENT MATRIX WAS DECOMPOSED BY A SUCCESSFUL CALL OF CHLDECBND (SECTION 3.1.2.1.1.2.1.1.).
THE SOLUTION OF THE LINEAR SYSTEM IS OBTAINED BY CARRYING OUT THE FORWARD AND BACK SUBSTITUTION WITH THE CHOLESKY MATRIX AND THE RIGHT HAND SIDE. THE LATTER IS OVERWRITTEN BY THE SOLUTION.
THE SOLUTIONS OF SEVERAL SYSTEMS WITH THE SAME COEFFICIENT MATRIX CAN BE OBTAINED BY SUCCESSIVE CALLS OF CHLSOLBND.

EXAMPLE OF USE :

THE FOLLOWING PROGRAM SOLVES THE SYSTEM OF SIMULTANEOUS EQUATIONS

```
  2 * X1  -   X2                              = 1
  -   X1 + 2 * X2  -   X3                     = 0
           -   X2 + 2 * X3  -   X4            = 0
                    -   X3 + 2 * X4  -   X5   = 0
                             -   X4 + 2 * X5  = 1
```

```
"BEGIN""COMMENT" 730829, TEST CHLDECBND, CHLSOLBND AND
   CHLDETERMBND;
   "PROCEDURE" CHLDECBND(A, N, W, AUX); "CODE" 34330;
   "PROCEDURE" CHLSOLBND(A, N, W, B); "CODE" 34332;
   "REAL""PROCEDURE" CHLDETERMBND(A, N, W); "CODE" 34331;
   "INTEGER" I;
   "ARRAY" SYMBAND[1 : 9], RIGHT[1 : 5], AUX[2 : 3];
   "FOR" I:= 1 "STEP" 1 "UNTIL" 9 "DO"
   SYMBAND[I]:= "IF" I // 2 * 2 < I "THEN" 2 "ELSE" - 1;
   RIGHT[1]:= RIGHT[5]:= 1;
   "FOR" I:= 2, 3, 4 "DO" RIGHT[I]:= 0; AUX[2]:= "- 12;
   CHLDECBND(SYMBAND, 5, 1, AUX);
   "IF" AUX[3] = 5 "THEN"
   "BEGIN" CHLSOLBND(SYMBAND, 5, 1, RIGHT);
      OUTPUT(61, "("5(+ZZ.4D2B), /"("DETERMINANT IS  ")" +.8D"+DD
      ")", (RIGHT[I], I:= 1 : 5), CHLDETERMBND(SYMBAND, 5, 1))
   "END"
"END".
```

THIS PROGRAM DELIVERS:

```
 +1.0000   +1.0000   +1.0000   +1.0000   +1.0000
DETERMINANT IS  +.60000000"+01
```

AUT SUBSECTION : CHLDECSOLBND.

CON

CALLING SEQUENCE :

INS

THE HEADING OF THE PROCEDURE IS :

REC

```
BRI "PROCEDURE" CHLDECSOLBND(A, N, W, AUX, B); "VALUE" N, W;
    "INTEGER" N, W; "ARRAY" A, AUX, B;
```

THE MEANING OF THE FORMAL PARAMETERS IS :

KEY

```
A, N, W, AUX    : SEE CHLDECBND;
B               : SEE CHLSOLBND.
```

PROCEDURES USED :

CAL

```
CHLDECBND = CP34330.
CHLSOLBND = CP34332.
```

RUNNING TIME:

(C1 * W + C5) * W * N;
THE CONSTANTS C1 AND C5 DEPEND UPON THE ELEMENTARY ARITHMETIC OF THE COMPUTER.

LANGUAGE : ALGOL 60.

METHOD AND PERFORMANCE :

CHLDECSOLBND SOLVES A SYSTEM OF LINEAR EQUATIONS BY CALLING CHLDECBND AND, PROVIDED THAT THE CHOLESKY DECOMPOSITION WAS SUCCESSFUL, CHLSOLBND. THE COEFFICIENT MATRIX OF THIS SYSTEM HAS TO BE A SYMMETRIC POSITIVE DEFINITE BAND MATRIX WHOSE UPPER-TRIANGULAR BAND ELEMENTS ARE STORED COLUMNWISE IN IONAL ARRAY.

EXAMPLE OF USE :

THE PROGRAM

```
"BEGIN""COMMENT" 730829, TEST CHLDECSOLBND AND CHLDETERMBND;
   "PROCEDURE" CHLDECSOLBND(A, N, W, AUX, B); "CODE" 34333;
   "REAL""PROCEDURE" CHLDETERMBND(A, N, W); "CODE" 34331;
   "INTEGER" I;
   "ARRAY" SYMBAND[1 : 9], RIGHT[1 : 5], AUX[2 : 3];
```

PROCEDURES USED: NONE.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE:

THE METHOD USED IN DECSYMTRI YIELDS A UNIT UPPER-BIDIAGONAL MATRIX U AND A DIAGONAL MATRIX D, SUCH THAT THE PRODUCT U'DU EQUALS THE GIVEN SYMMETRIC TRIDIAGONAL MATRIX; THE PROCESS IS TERMINATED IN THE K-TH STEP IF THE MODULUS OF THE K-TH DIAGONAL ELEMENT IS SMALLER THAN A CERTAIN SMALL VALUE, WHICH IS GIVEN BY AUX[2] MULTIPLIED BY THE 1-NORM OF THE K-TH ROW; IN THIS CASE AUX[3] WILL BE GIVEN THE VALUE K - 1 AND AUX[5] WILL BE GIVEN THE VALUE OF THE K-TH DIAGONAL ELEMENT.

EXAMPLE OF USE: SEE DECSOLSYMTRI (SECTION 3.1.2.1.1.2.2.3).

SOURCE TEXT:

```
"CODE" 34420;
        "PROCEDURE" DECSYMTRI(DIAG, CO, N, AUX); "VALUE" N; "INTEGER" N;
        "ARRAY" DIAG, CO, AUX;
        "BEGIN"  "INTEGER" I, N1;
           "REAL" D, R, S, U, TOL, NORM, NORMR;
           TOL:= AUX[2]; D:= DIAG[1]; R:= CO[1];
           NORM:= NORMR:= ABS(D) + ABS(R);
           "IF" ABS(D) <= NORMR * TOL "THEN"
           "BEGIN" AUX[3]:= 0; AUX[5]:= D; "GOTO" EXIT "END";
           U:= CO[1]:= R / D; N1:= N - 1;
           "FOR" I:= 2 "STEP" 1 "UNTIL" N1 "DO"
           "BEGIN" S:= R; R:= CO[I]; D:= DIAG[I];
              NORMR:= ABS(S) + ABS(D) + ABS(R);
              D:= DIAG[I]:= D - U * S;
              "IF" ABS(D) <= NORMR * TOL "THEN"
              "BEGIN" AUX[3]:= I - 1; AUX[5]:= D; "GOTO" EXIT "END";
              U:= CO[I]:= R / D; "IF" NORMR > NORM "THEN" NORM:= NORMR
           "END";
           D:= DIAG[N]; NORMR:= ABS(D) + ABS(R);
           D:= DIAG[N]:= D - U * R;
           "IF" ABS(D) <= NORMR * TOL "THEN"
           "BEGIN" AUX[3]:= N1; AUX[5]:= D; "GOTO" EXIT "END";
           "IF" NORMR > NORM "THEN" NORM:= NORMR;
           AUX[3]:= N; AUX[5]:= NORM;
        EXIT:
        "END" DECSYMTRI;
          "EOP"
```

N : CHLDECSOLBND.

EQUENCE :

EADING OF THE PROCEDURE IS :

```
EDURE" CHLDECSOLBND(A, N, W, AUX, B); "VALUE" N, W;
GER" N, W; "ARRAY" A, AUX, B;
```

ANING OF THE FORMAL PARAMETERS IS :

N, W, AUX : SEE CHLDECBND;
: SEE CHLSOLBND.

SYMMETRIC

USED :

(SECTION

BND = CP34330,
BND = CP34332.

UTION TO

ME:

W + C5) * W * N;
NSTANTS C1 AND C5 DEPEND UPON THE ELEMENTARY
ETIC OF THE COMPUTER.

: ALGOL 60.

PERFORMANCE :

SOLBND SOLVES A SYSTEM OF LINEAR EQUATIONS BY CALLING
BND AND, PROVIDED THAT THE CHOLESKY DECOMPOSITION WAS
SFUL, CHLSOLBND. THE COEFFICIENT MATRIX OF THIS SYSTEM HAS TO
YMMETRIC POSITIVE DEFINITE BAND MATRIX WHOSE UPPER-TRIANGULAR
LEMENTS ARE STORED COLUMNWISE IN IONAL ARRAY.

USE :

IGRAM

```
'"COMMENT" 730829, TEST CHLDECSOLBND AND CHLDETERMBND;
ICEDURE" CHLDECSOLBND(A, N, W, AUX, B); "CODE" 34333;
L""PROCEDURE" CHLDETERMBND(A, N, W); "CODE" 34331;
EGER" I;
AY" SYMBAND[1 : 9], RIGHT[1 : 5], AUX[2 : 3];
```

SUBSECTION: SOLSYMTRI.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" SOLSYMTRI(DIAG, CO, N, B);
    "VALUE" N; "INTEGER" N; "ARRAY" DIAG, CO, B; "CODE" 34421;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    DIAG:   <ARRAY IDENTIFIER>;
            "ARRAY" DIAG[1:N];
            ENTRY:  THE  DIAGONAL  MATRIX,  AS
            DELIVERED BY DECSYMTRI (SECTION 3.1.2.1.1.2.2.1);
    CO:     <ARRAY IDENTIFIER>;
            "ARRAY" CO[1: N - 1];
            ENTRY:  THE CODIAGONAL OF THE UNIT
            UPPER-BIDIAGONAL MATRIX AS DELIVERED BY DECSYMTRI;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   THE CALCULATED SOLUTION OF THE LINEAR SYSTEM.
```

PROCEDURES USED: NONE.

METHOD AND PERFORMANCE:

SOLSYMTRI CALCULATES THE SOLUTION OF A LINEAR SYSTEM WITH A SYMMETRIC TRIDIAGONAL MATRIX. WITH FORWARD AND BACK SUBSTITUTION: THE TRIANGULARLY DECOMPOSED FORM OF THE MATRIX, AS PRODUCED BY DECSYMTRI (SECTION 3.1.2.1.1.2.2.1), SHOULD BE GIVEN; ONE CALL OF DECSYMTRI FOLLOWED BY SEVERAL CALLS OF SOLSYMTRI MAY BE USED TO SOLVE SEVERAL LINEAR SYSTEMS HAVING THE SAME SYMMETRIC TRIDIAGONAL MATRIX, BUT DIFFERENT RIGHT HAND SIDES.

EXAMPLE OF USE: SEE DECSOLSYMTRI (THIS SECTION).

SUBSECTION: DECSOLSYMTRI.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DECSOLSYMTRI(DIAG, CO, N, AUX, B);
    "VALUE" N; "INTEGER" N; "ARRAY" DIAG, CO, AUX, B; "CODE" 34422;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    DIAG:    <ARRAY IDENTIFIER>;
             "ARRAY" DIAG[1: N];
             ENTRY:  THE DIAGONAL OF THE GIVEN MATRIX T, SAY;
             EXIT:   SUPPOSE U DENOTES THE UNIT UPPER-BIDIAGONAL MATRIX,
                     SUCH THAT  U'DU = T   FOR SOME  DIAGONAL MATRIX  D,
                     WHERE U' DENOTES THE TRANSPOSED MATRIX; THEN D[I,I]
                     WILL BE DELIVERED IN DIAG[I], I = 1, ..., AUX[3];
    CO:      <ARRAY IDENTIFIER>;
             "ARRAY" CO[1: N - 1];
             ENTRY:  THE CODIAGONAL OF T; T[I, I + 1] SHOULD BE GIVEN IN
                     CO[I], I = 1, ..., N - 1;
             EXIT:   U[I, I + 1] WILL BE DELIVERED IN CO[I], I = 1, ...,
                     AUX[3] - 1;
    N:       <ARITHMETIC EXPRESSION>;
             THE ORDER OF THE MATRIX;
    AUX:     <ARRAY IDENTIFIER>;
             "ARRAY" AUX[2:5];
             ENTRY:
             AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                     VALUE IS AN ESTIMATE OF  THE RELATIVE PRECISION  OF
                     THE  MATRIX ELEMENTS,  HOWEVER, IT  SHOULD  NOT  BE
                     CHOSEN SMALLER THAN THE MACHINE PRECISION;
             EXIT:
             AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
             AUX[5]: IF AUX[3] = N, THEN AUX[5] WILL EQUAL THE INFINITY-
                     NORM OF THE MATRIX,  ELSE  AUX[5]  IS SET  EQUAL TO
                     THE  VALUE  OF  THAT  ELEMENT   WHICH   CAUSES  THE
                     BREAKDOWN OF THE DECOMPOSITION;
    B:       <ARRAY IDENTIFIER>;
             "ARRAY" B[1:N];
             ENTRY:  THE RIGHT HAND SIDE OF THE LINEAR SYSTEM;
             EXIT:   IF  AUX[3] = N  THEN  THE  SOLUTION  OF  THE LINEAR
                     SYSTEM  IS  OVERWRITTEN  ON   B,  ELSE   B  REMAINS
                     UNALTERED.
```

PROCEDURES USED:

DECSYMTRI = CP34420,
SOLSYMTRI = CP34421.

METHOD AND PERFORMANCE:

DECSOLSYMTRI CALCULATES THE SOLUTION OF A LINEAR SYSTEM WITH A SYMMETRIC TRIDIAGONAL MATRIX; THE TRIANGULAR DECOMPOSITION IS DONE BY CALLING DECSYMTRI (SECTION 3.1.2.1.1.2.2.1) AND THE FORWARD AND BACK SUBSTITUTION BY CALLING SOLSYMTRI (THIS SECTION); IF AUX[3]<N, THEN THE EFFECT OF DECSOLSYMTRI IS MERELY THAT OF DECSYMTRI.

EXAMPLE OF USE:

LET T BE A SYMMETRIC TRIDIAGONAL MATRIX OF ORDER 100 WITH DIAGONAL ELEMENTS I (I = 1, ..., 100) AND CODIAGONAL ELEMENTS I * 2 (I = 1, ..., 99); LET THE RIGHT HAND SIDE B BE GIVEN BY THE SECOND COLUMN OF T; THEN THE SOLUTION OF THE LINEAR SYSTEM TX = B IS GIVEN BY THE SECOND UNIT VECTOR; BY THE FOLLOWING PROGRAM WE MAY SOLVE THIS SYSTEM AND PRINT THE ERROR IN THE CALCULATED SOLUTION.

```
"BEGIN"
    "PROCEDURE" DECSOLSYMTRI(D, C, N, A, B); "CODE" 34422;
    "REAL" "PROCEDURE" VECVEC(L, U, S, A, B); "CODE" 34010;
    "INTEGER" I; "ARRAY" D, CO, B[1:100], AUX[2:5];
    "FOR" I:= 1 "STEP" 1 "UNTIL" 100 "DO"
    "BEGIN" D[I]:= I; CO[I]:= I * 2; B[I]:= 0 "END";
    B[1]:= B[2]:= 2; B[3]:= 4;
    AUX[2]:= "-14;
    DECSOLSYMTRI(D, CO, 100, AUX, B);
    B[2]:= B[2] - 1;
    OUTPUT(71, "("/,"("     AUX[3] AND AUX[5]:")",2(/4B,N))",
    AUX[3], AUX[5]);
    OUTPUT(71, "("//,"("     ERROR IN THE SOLUTION:")",N,/")",
    SQRT(VECVEC(1, 100, 0, B, B)))
"END"
```

RESULTS:

```
AUX[3] AND AUX[5]:
+1.0000000000000"+002
+4.9300000000000"+002

ERROR IN THE SOLUTION:+0.0000000000000"+000
```

SUBSECTION: DECSOLSYMTRI.

CALLING SEQUENCE:

```
    THE HEADING OF THIS PROCEDURE IS:
    "PROCEDURE" DECSOLSYMTRI(DIAG, CO, N, AUX, B);
    "VALUE" N; "INTEGER" N; "ARRAY" DIAG, CO, AUX, B; "CODE" 34422;

    THE MEANING OF THE FORMAL PARAMETERS IS:
    DIAG:   <ARRAY IDENTIFIER>;
            "ARRAY" DIAG[1: N];
            ENTRY:  THE DIAGONAL OF THE GIVEN MATRIX T, SAY;
            EXIT:   SUPPOSE U DENOTES THE UNIT UPPER-BIDIAGONAL MATRIX,
                    SUCH THAT  U'DU = T   FOR SOME  DIAGONAL MATRIX  D,
                    WHERE U' DENOTES THE TRANSPOSED MATRIX; THEN D[I,I]
                    WILL BE DELIVERED IN DIAG[I], I = 1, ..., AUX[3];
    CO:     <ARRAY IDENTIFIER>;
            "ARRAY" CO[1: N - 1];
            ENTRY:  THE CODIAGONAL OF T; T[I, I + 1] SHOULD BE GIVEN IN
                    CO[I], I = 1, ..., N - 1;
            EXIT:   U[I, I + 1] WILL BE DELIVERED IN CO[I], I = 1, ...,
                    AUX[3] - 1;
    N:      <ARITHMETIC EXPRESSION>;
            THE ORDER OF THE MATRIX;
    AUX:    <ARRAY IDENTIFIER>;
            "ARRAY" AUX[2:5];
            ENTRY:
            AUX[2]: A RELATIVE TOLERANCE;  A REASONABLE CHOICE FOR THIS
                    VALUE IS AN ESTIMATE OF  THE RELATIVE PRECISION  OF
                    THE  MATRIX ELEMENTS,  HOWEVER, IT  SHOULD  NOT  BE
                    CHOSEN SMALLER THAN THE MACHINE PRECISION;
            EXIT:
            AUX[3]: THE NUMBER OF ELIMINATION STEPS PERFORMED;
            AUX[5]: IF AUX[3] = N, THEN AUX[5] WILL EQUAL THE INFINITY-
                    NORM OF THE MATRIX,  ELSE  AUX[5]  IS SET  EQUAL TO
                    THE   VALUE   OF   THAT  ELEMENT   WHICH   CAUSES  THE
                    BREAKDOWN OF THE DECOMPOSITION;
    B:      <ARRAY IDENTIFIER>;
            "ARRAY" B[1:N];
            ENTRY:  THE RIGHT HAND SIDE OF THE LINEAR SYSTEM;
            EXIT:   IF  AUX[3] = N  THEN  THE  SOLUTION  OF  THE LINEAR
                    SYSTEM  IS  OVERWRITTEN  ON   B,  ELSE   B  REMAINS
                    UNALTERED.
```

PROCEDURES USED:

DECSYMTRI = CP34420,
SOLSYMTRI = CP34421.

METHOD AND PERFORMANCE:

DECSOLSYMTRI CALCULATES THE SOLUTION OF A LINEAR SYSTEM WITH A SYMMETRIC TRIDIAGONAL MATRIX; THE TRIANGULAR DECOMPOSITION IS DONE BY CALLING DECSYMTRI (SECTION 3.1.2.1.1.2.2.1) AND THE FORWARD AND BACK SUBSTITUTION BY CALLING SOLSYMTRI (THIS SECTION); IF AUX[3]<N, THEN THE EFFECT OF DECSOLSYMTRI IS MERELY THAT OF DECSYMTRI.

EXAMPLE OF USE:

LET T BE A SYMMETRIC TRIDIAGONAL MATRIX OF ORDER 100 WITH DIAGONAL ELEMENTS I (I = 1, ..., 100) AND CODIAGONAL ELEMENTS I * 2 (I = 1, ..., 99); LET THE RIGHT HAND SIDE B BE GIVEN BY THE SECOND COLUMN OF T; THEN THE SOLUTION OF THE LINEAR SYSTEM TX = B IS GIVEN BY THE SECOND UNIT VECTOR; BY THE FOLLOWING PROGRAM WE MAY SOLVE THIS SYSTEM AND PRINT THE ERROR IN THE CALCULATED SOLUTION.

```
"BEGIN"
    "PROCEDURE" DECSOLSYMTRI(D, C, N, A, B); "CODE" 34422;
    "REAL" "PROCEDURE" VECVEC(L, U, S, A, B); "CODE" 34010;
    "INTEGER" I; "ARRAY" D, CO, B[1:100], AUX[2:5];
    "FOR" I:= 1 "STEP" 1 "UNTIL" 100 "DO"
    "BEGIN" D[I]:= I; CO[I]:= I * 2; B[I]:= 0 "END";
    B[1]:= B[2]:= 2; B[3]:= 4;
    AUX[2]:= "-14;
    DECSOLSYMTRI(D, CO, 100, AUX, B);
    B[2]:= B[2] - 1;
    OUTPUT(71, "("/,"("    AUX[3] AND AUX[5]:")",2(/4B,N)")",
    AUX[3], AUX[5]);
    OUTPUT(71, "("//,"("    ERROR IN THE SOLUTION:")",N,/")",
    SQRT(VECVEC(1, 100, 0, B, B)))
"END"
```

RESULTS:

```
AUX[3] AND AUX[5]:
+1.0000000000000"+002
+4.9300000000000"+002

ERROR IN THE SOLUTION:+0.0000000000000"+000
```

SOURCE TEXTS:

```
"CODE" 34421;
      "PROCEDURE" SOLSYMTRI(DIAG, CO, N, B); "VALUE" N; "INTEGER" N;
      "ARRAY" DIAG, CO, B;
      "BEGIN" "INTEGER" I;
         "REAL" R, S;
         R:= B[1]; B[1]:= R / DIAG[1];
         "FOR" I:= 2 "STEP" 1 "UNTIL" N "DO"
         "BEGIN" R:= B[I] - CO[I-1] * R; B[I]:= R / DIAG[I] "END";
         S:= B[N];
         "FOR" I:= N - 1 "STEP" -1 "UNTIL" 1 "DO"
         S:= B[I]:= B[I] - CO[I] * S
      "END" SOLSYMTRI;
        "EOP"


"CODE" 34422;
      "PROCEDURE" DECSOLSYMTRI(DIAG, CO, N, AUX, B); "VALUE" N;
      "INTEGER" N; "ARRAY" DIAG, CO, AUX, B;
      "BEGIN" "PROCEDURE" DECSYMTRI(DIAG, CO, N, AUX); "CODE" 34420;
              "PROCEDURE" SOLSYMTRI(DIAG, CO, N, B);   "CODE" 34421;
              DECSYMTRI(DIAG, CO, N, AUX); "IF" AUX[3] = N "THEN"
              SOLSYMTRI(DIAG, CO, N, B)
      "END" DECSOLSYMTRI;
        "EOP"
```

AUTHOR: P.W.HEMKER.

INSTITUTE: MATHEMATICAL CENTRE.

RECEIVED: 730615.

BRIEF DESCRIPTION:

CONJ GRAD SOLVES A LINEAR SYSTEM OF EQUATIONS BY THE METHOD OF CONJUGATE GRADIENTS. THE SYSTEM HAS TO BE SYMMETRIC AND POSITIVE DEFINITE.

CALLING SEQUENCE:

THE HEADING OF THE PROCEDURE READS:

```
"PROCEDURE" CONJ GRAD (MATVEC, X, R, L, N, GO ON, ITERATE, NORM2);
"VALUE" L, N; "BOOLEAN" GO ON; "INTEGER" L, N, ITERATE;
"REAL" NORM2; "ARRAY" X, R; "PROCEDURE" MATVEC;
```

THE MEANING OF THE FORMAL PARAMETERS IS:

MATVEC: <PROCEDURE IDENTIFIER>;
THE HEADING OF THIS PROCEDURE READS:
"PROCEDURE" MATVEC(P, Q); "ARRAY" P, Q;
THIS PROCEDURE DEFINES THE MATRIX A (THE COEFFICIENT MATRIX OF THE SYSTEM) AS FOLLOWS :
AT EACH CALL MATVEC DELIVERS IN Q THE MATRIX-VECTOR PRODUCT AP; P AND Q ARE ONE - DIMENSIONAL ARRAYS:
"ARRAY" P,Q[L:N];

X: <ARRAY IDENTIFIER>;
"ARRAY" X[L:N];
ENTRY: AN INITIAL APPROXIMATION TO THE SOLUTION X;
EXIT: THE SOLUTION;

R: <ARRAY IDENTIFIER>;
"ARRAY" R[L:N];
ENTRY: THE RIGHT-HAND SIDE OF THE SYSTEM;
EXIT: THE RESIDUE B - AX, COMPUTED RECURSIVELY;

L,N: <ARITHMETIC EXPRESSION>;
L AND N ARE RESPECTIVELY THE LOWER AND UPPER BOUND OF THE ARRAYS X,R,P,Q;

GO ON: <BOOLEAN EXPRESSION>;
GO ON INDICATES THE CONTINUATION OF THE PROCESS.
THIS EXPRESSION MAY DEPEND ON THE JENSEN PARAMETERS ITERATE AND NORM2. WITH THIS BOOLEAN EXPRESSION THE USER CONTROLS THE CONTINUATION OF THE PROCESS. IF GO ON:= "FALSE" THE ITERATION PROCESS IS STOPPED.

ITERATE: <IDENTIFIER>;
DELIVERS THE NUMBER OF ITERATION STEPS ALREADY PERFORMED;

NORM2: <IDENTIFIER>;
DELIVERS THE SQUARED EUCLIDIC NORM OF THE RESIDUE,COMPUTED RECURSIVELY

PROCEDURES USED:

VECVEC = CP34010 ,
ELMVEC = CP34020 .

REQUIRED CENTRAL MEMORY:

EXECUTION FIELD LENGTH: 7 + 2 * (N - L + 1).

RUNNING TIME:

THE RUNNING TIME IS PROPORTIONAL TO THE NUMBER OF ITERATION STEPS PERFORMED. EACH ITERATION STEP REQUIRES ONE EVALUATION OF THE PROCEDURE MATVEC, THE EVALUATION OF TWO SCALAR - VECTOR - PRODUCTS AND ONE VECTOR - VECTOR - PRODUCT.

LANGUAGE: ALGOL 60.

METHOD AND PERFORMANCE: SEE REF[1].

REFERENCES:

[1].J.K.REID.
ON THE METHOD OF CONJUGATE GRADIENTS FOR THE SOLUTION OF LARGE SPARSE SYSTEMS OF LINEAR EQUATIONS.
IN:LARGE SPARSE SETS OF LINEAR EQUATIONS (J.K.REID ED)1971.

EXAMPLE OF USE:

```
"BEGIN"
    "PROCEDURE" CONJ GRAD(A,X,P,L,N,GO ON,IT,NO);"CODE" 34220;
    "ARRAY" X,B[0:12];
    "INTEGER" IT,I;
    "REAL" NO;
    "PROCEDURE" A(X,B);
    "ARRAY" X,B;
    "BEGIN" B[0]:=2*X[0]-X[1];
        "FOR" I:=1 "STEP" 1 "UNTIL" 11 "DO"
        B[I]:=-X[I-1]+2*X[I]-X[I+1];
        B[12]:=2*X[12]-X[11]
    "END" A;
    "FOR" I:=0 "STEP" 1 "UNTIL" 12 "DO" X[I]:=B[I]:=0;
    B[0]:=1;B[12]:=4;
    CONJ GRAD(A,X,B,0,12,IT<20 "AND" NO>"-10,IT,NO);
    OUTPUT(61,"(""("IT= ")",B3D5B,"("NO= ")",N,//,10B,
    "("X")",20B,"("R")",//")",IT,NO);
    "FOR" I:=0 "STEP" 1 "UNTIL" 12 "DO"
    OUTPUT(61,"("N,5B,N,/")",X[I],B[I])
"END"
```

```
DELIVERS:
IT= 013      NO= +3.3424581859911"-027

              X                          R

+1.2142857142857"+000      -7.1054273576010"-015
+1.4285714285715"+000      +1.5151278924296"-014
+1.6428571428572"+000      -1.3184703260130"-014
+1.8571428571429"+000      +1.6718441615946"-014
+2.0714285714286"+000      -1.5514524667596"-014
+2.2857142857144"+000      +2.2130179956186"-014
+2.5000000000001"+000      -2.2524167805437"-014
+2.7142857142858"+000      +2.0834049529361"-014
+2.9285714285715"+000      -1.8674557504802"-014
+3.1428571428572"+000      +1.9163204503355"-014
+3.3571428571429"+000      -1.2366043539824"-014
+3.5714285714286"+000      +8.2548347242718"-015
+3.7857142857143"+000      +4.4408920985006"-016
```

SOURCE TEXT(S):

```
"CODE" 34220;
    "PROCEDURE" CONJ GRAD( MATVEC, X, R, L, N, GO ON, ITERATE, NORM2);
    "VALUE" L, N; "PROCEDURE" MATVEC; "ARRAY" X, R; "BOOLEAN" GO ON;
    "INTEGER" L, N, ITERATE; "REAL" NORM2;
    "BEGIN" "ARRAY" P, AP[ L: N];
        "INTEGER" I;
        "REAL"  A, B, PRR, RRP;
        "REAL" "PROCEDURE" VECVEC( A, B, C, D, E); "CODE" 34010;
        "PROCEDURE" ELMVEC( A, B, C, D, E, F); "CODE" 34020;
        "FOR" ITERATE:= 0, ITERATE + 1 "WHILE" GO ON "DO"
        "BEGIN" "IF" ITERATE = 0 "THEN"
            "BEGIN" MATVEC( X, P);
                "FOR" I:= L "STEP" 1 "UNTIL" N "DO"
                P[ I]:= R[ I]:= R[ I] - P[ I];
                PRR:= VECVEC( L, N, 0, R, R)
            "END" "ELSE"
            "BEGIN" B:= RRP / PRR; PRR:= RRP;
                "FOR" I:= L "STEP" 1 "UNTIL" N "DO"
                P[ I]:= R[ I] + B * P[ I]
            "END";
            MATVEC( P, AP);
            A:= PRR / VECVEC( L, N, 0, P, AP);
            ELMVEC( L, N, 0, X, P, A);
            ELMVEC( L, N, 0, R, AP, -A);
            NORM2:= RRP:= VECVEC( L, N, 0, R, R)
        "END"
    "END" CONJ GRAD;
        "EOP"
```

UITGAVEN IN DE SERIE MC SYLLABUS

Onderstaande uitgaven zijn verkrijgbaar bij het Mathematisch Centrum, 2e Boerhaavestraat 49 te Amsterdam-1005, tel. 020-947272.

MCS 1.1 F. GÖBEL & J. VAN DE LUNE, *Leergang Besliskunde, deel 1: Wiskundige basiskennis*, 1965. ISBN 90 6196 014 2.

MCS 1.2 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 2: Kansberekening*, 1965. ISBN 90 6196 015 0.

MCS 1.3 J. HEMELRIJK & J. KRIENS, *Leergang Besliskunde, deel 3: Statistiek*, 1966. ISBN 90 6196 016 9.

MCS 1.4 G. DE LEVE & W. MOLENAAR, *Leergang Besliskunde, deel 4: Markovketens en wachttijden*, 1966. ISBN 90 6196 017 7.

MCS 1.5 J. KRIENS & G. DE LEVE, *Leergang Besliskunde, deel 5: Inleiding tot de mathematische besliskunde*, 1966. ISBN 90 6196 018 5.

MCS 1.6a B. DORHOUT & J. KRIENS, *Leergang Besliskunde, deel 6a: Wiskundige programmering 1*, 1968. ISBN 90 6196 032 0.

MCS 1.6b B. DORHOUT, J. KRIENS & J.TH. VAN LIESHOUT, *Leergang Besliskunde, deel 6b: Wiskundige programmering 2*, 1977. ISBN 90 6196 150 5.

MCS 1.7a G. DE LEVE, *Leergang Besliskunde, deel 7a: Dynamische programmering 1*, 1968. ISBN 90 6196 033 9.

MCS 1.7b G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7b: Dynamische programmering 2*, 1970. ISBN 90 6196 055 X.

MCS 1.7c G. DE LEVE & H.C. TIJMS, *Leergang Besliskunde, deel 7c: Dynamische programmering 3*, 1971. ISBN 90 6196 066 5.

MCS 1.8 J. KRIENS, F. GÖBEL & W. MOLENAAR, *Leergang Besliskunde, deel 8: Minimaxmethode, netwerkplanning, simulatie*, 1968. ISBN 90 6196 034 7.

MCS 2.1 G.J.R. FÖRCH, P.J. VAN DER HOUWEN & R.P. VAN DE RIET, *Colloquium Stabiliteit van differentieschema's, deel 1*, 1967. ISBN 90 6196 023 1.

MCS 2.2 L. DEKKER, T.J. DEKKER, P.J. VAN DER HOUWEN & M.N. SPIJKER, *Colloquium Stabiliteit van differentieschema's, deel 2*, 1968. ISBN 90 6196 035 5.

MCS 3.1 H.A. LAUWERIER, *Randwaardeproblemen, deel 1*, 1967. ISBN 90 6196 024 X.

MCS 3.2 H.A. LAUWERIER, *Randwaardeproblemen, deel 2*, 1968. ISBN 90 6196 036 3.

MCS 3.3 H.A. LAUWERIER, *Randwaardeproblemen, deel 3*, 1968. ISBN 90 6196 043 6.

MCS 4 H.A. LAUWERIER, *Representaties van groepen*, 1968. ISBN 90 6196 037 1.

MCS 5 J.H. VAN LINT, J.J. SEIDEL & P.C. BAAYEN, *Colloquium Discrete wiskunde*, 1968.
ISBN 90 6196 044 4.

MCS 6 K.K. KOKSMA, *Cursus ALGOL 60*, 1969. ISBN 90 6196 045 2.

MCS 7.1 *Colloquium Moderne rekenmachines, deel 1*, 1969. ISBN 90 6196 046 0.

MCS 7.2 *Colloquium Moderne rekenmachines, deel 2*, 1969. ISBN 90 6196 047 9.

MCS 8 H. BAVINCK & J. GRASMAN, *Relaxatietrillingen*, 1969.
ISBN 90 6196 056 8.

MCS 9.1 T.M.T. COOLEN, G.J.R. FÖRCH, E.M. DE JAGER & H.G.J. PIJLS, *Elliptische differentiaalvergelijkingen, deel 1*, 1970.
ISBN 90 6196 048 7.

MCS 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.
ISBN 90 6196 049 5.

MCS 10 J. FABIUS & W.R. VAN ZWET, *Grondbegrippen van de waarschijnlijkheidsrekening*, 1970. ISBN 90 6196 057 6.

MCS 11 H. BART, M.A. KAASHOEK, H.G.J. PIJLS, W.J. DE SCHIPPER & J. DE VRIES, *Colloquium Halfalgebra's en positieve operatoren*, 1971.
ISBN 90 6196 067 3.

MCS 12 T.J. DEKKER, *Numerieke algebra*, 1971. ISBN 90 6196 068 1.

MCS 13 F.E.J. KRUSEMAN ARETZ, *Programmeren voor rekenautomaten; De MC ALGOL 60 vertaler voor de EL X8*, 1971. ISBN 90 6196 069 X.

MCS 14 H. BAVINCK, W. GAUTSCHI & G.M. WILLEMS, *Colloquium Approximatietheorie*, 1971. ISBN 90 6196 070 3.

MCS 15.1 T.J. DEKKER, P.W. HEMKER & P.J. VAN DER HOUWEN, *Colloquium Stijve differentiaalvergelijkingen, deel 1*, 1972. ISBN 90 6196 078 9.

MCS 15.2 P.A. BEENTJES, K. DEKKER, H.C. HEMKER, S.P.N. VAN KAMPEN & G.M. WILLEMS, *Colloquium Stijve differentiaalvergelijkingen, deel 2*, 1973. ISBN 90 6196 079 7.

MCS 15.3 P.A. BEENTJES, K. DEKKER, P.W. HEMKER & M. VAN VELDHUIZEN, *Colloquium Stijve differentiaalvergelijkingen, deel 3*, 1975.
ISBN 90 6196 118 1.

MCS 16.1 L. GEURTS, *Cursus Programmeren, deel 1: De elementen van het programmeren*, 1973. ISBN 90 6196 080 0.

MCS 16.2 L. GEURTS, *Cursus Programmeren, deel 2: De programmeertaal ALGOL 60*, 1973. ISBN 90 6196 087 8.

MCS 17.1 P.S. STOBBE, *Lineaire algebra, deel 1*, 1974. ISBN 90 6196 090 8.

MCS 17.2 P.S. STOBBE, *Lineaire algebra, deel 2*, 1974. ISBN 90 6196 091 6.

MCS 17.3 N.M. TEMME, *Lineaire algebra, deel 3*, 1976. ISBN 90 6196 123 8.

MCS 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*, 1974. ISBN 90 6196 092 4.

MCS 19 A. HORDIJK, R. POTHARST & J.Th. RUNNENBURG, *Optimaal stoppen van Markovketens*, 1974. ISBN 90 6196 093 2.

MCS 20 T.M.T. COOLEN, P.W. HEMKER, P.J. VAN DER HOUWEN & E. SLAGT, *ALGOL 60 procedures voor begin- en randwaardeproblemen*, 1976. ISBN 90 6196 094 0.

MCS 21 J.W. DE BAKKER (red.), *Colloquium Programmacorrectheid*, 1975. ISBN 90 6196 103 3.

MCS 22 R. HELMERS, F.H. RUYMGAART, M.C.A. VAN ZUYLEN & J. OOSTERHOFF, *Asymptotische methoden in de toetsingstheorie; Toepassingen van naburigheid*, 1976. ISBN 90 6196 104 1.

MCS 23.1 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 1*, 1976. ISBN 90 6196 105 X.

MCS 23.2 J.W. DE ROEVER (red.), *Colloquium Onderwerpen uit de biomathematica, deel 2*, 1976. ISBN 90 6196 115 7.

MCS 24.1 P.J. VAN DER HOUWEN, *Numerieke integratie van differentiaalvergelijkingen, deel 1: Eenstapsmethoden*, 1974. ISBN 90 6196 106 8.

MCS 25 *Colloquium Structuur van programmeertalen*, 1976. ISBN 90 6196 116 5.

MCS 26.1 N.M. TEMME (ed.), *Nonlinear analysis, volume 1*, 1976. ISBN 90 6196 117 3.

MCS 26.2 N.M. TEMME (ed.), *Nonlinear analysis, volume 2*, 1976. ISBN 90 6196 121 1.

MCS 27 M. BAKKER, P.W. HEMKER, P.J. VAN DER HOUWEN, S.J. POLAK & M. VAN VELDHUIZEN, *Colloquium Discretiseringsmethoden*, 1976. ISBN 90 6196 124 6.

MCS 28 O. DIEKMANN, N.M. TEMME (EDS), *Nonlinear Diffusion Problems*, 1976. ISBN 90 6196 126 2.

MCS 29.1 J.C.P. BUS (red.), *Colloquium Numerieke programmatuur, deel 1A, deel 1B*, 1976. ISBN 90 6196 128 9.

MCS 29.2 H.J.J. TE RIELE (red.), *Colloquium Numerieke programmatuur, deel 2*, 1976. ISBN 90 6196 144 0

* MCS 30 P. GROENEBOOM, R. HELMERS, J. OOSTERHOFF & R. POTHARST, *Efficiency begrippen in de statistiek*, . ISBN 90 6196 149 1.

MCS 31 J.H. VAN LINT (red.), *Inleiding in de coderingstheorie*, 1976. ISBN 90 6196 136 X.

MCS 32 L. GEURTS (red.), *Colloquium Bedrijfssystemen*, 1976. ISBN 90 6196 137 8.

MCS 33 P.J. VAN DER HOUWEN, *Differentieschema's voor de berekening van waterstanden in zeeën en rivieren*, 1977. ISBN 90 6196 138 6.

MCS 34 J. HEMELRIJK, *Oriënterende cursus mathematische statistiek*, ISBN 90 6196 139 4.

MCS 35 P.J.W. TEN HAGEN (red.), *Colloquium Computer Graphics*, 1977. ISBN 90 6196 142 4.

MCS 36 J.M. AARTS, J. DE VRIES, *Colloquium Topologische Dynamische Systemen*, 1977. ISBN 90 6196 143 2.

MCS 37 J.C. van Vliet (red.), *Colloquium Capita Datastructuren*, 1978. ISBN 90 6196 159 9.

MCS 38.1 T.H. KOORNWINDER (ED.), *Representations of locally compact groups with applications*, 1979. ISBN 90 6196 161 0.

MCS 38.2 T.H. KOORNWINDER (ED.), *Representations of locally compact groups with applications*, 1979. ISBN 90 6196 181 5.

MCS 39 O.J. VRIEZE & G.L. Wanrooij, *Colloquium Stochastische Spelen*, 1978. ISBN 90 6196 167 X.

MCS 40 J. VAN TIEL, *Convexe Analyse*, 1979. ISBN 90 6196 187 4.

MCS 41 H.J.J. TE RIELE (ED.), *Colloquium Numerical Treatment of Integral Equations*, 1979. ISBN 90 6196 189 0.

MCS 42 J.C. VAN VLIET (RED.), *Colloquium Capita Implementatie van Programmeertalen*, 1980. ISBN 90 6196 191 2.

MCS 43 A.M. COHEN & H.A. WILBRINK, *Eindige groepen (Een inleidende cursus)*, 1980. ISBN 90 6196 203 X

MCS 44 J.G. VERWER (ED.), *Numerical solution of partial differential equations*, 1980. ISBN 90 6196 205 6.

MCS 45 P. KLINT (red.), *Colloquium hogere programmeertalen en computerarchitectuur*, 1980. ISBN 90 6196 206 4.

MCS 46.1 P.M.G. APERS (RED.), *Colloquium Databankorganisatie*, 1981. ISBN 90 6196 212 9.

MCS 47.1 P.W. HEMKER (ED.), *NUMAL numerical procedures in ALGOL 60*, Part I: *General information and indices*, 1981. ISBN 90 6196 217 X.

MCS 47.2 P.W. HEMKER (ED.), *NUMAL numerical procedures in ALGOL 60*, Part II: *Elementary procedures, algebraic evaluation*, 1981, ISBN 90 6196 217 X.

MCS 47.3 P.W. HEMKER (ED.), *NUMAL numerical procedures in ALGOL 60*, Part III: *Linear algebra, part I*, 1981. ISBN 90 6196 217 X.

MCS 47.4 P.W. HEMKER (ED.), *NUMAL numerical procedures in ALGOL 60*, Part IV: *Linear algebra, part II*, 1981. ISBN 90 6196 217 X.

MCS 47.5 P.W. HEMKER (ED.), *NUMAL numerical procedures in ALGOL 60*, Part V: *Analytical evaluations, analytical problems, part I*, 1981. ISBN 90 6196 217 X.

MCS 47.6 P.W. HEMKER (ED.), *NUMAL numerical procedures in ALGOL 60*, Part VI: *Analytical problems, part II*, 1981. ISBN 90 6196 217 X.

MCS 47.7 P.W. HEMKER (ED.), *NUMAL numerical procedures in ALGOL 60*, Part VII: *Special functions and constants, interpolation and approximation*, 1981. ISBN 90 6196 217 X.

De met een * gemerkte uitgaven moeten nog verschijnen.