NUMVEC FORTRAN Library manual

Chapter: Simultaneous Linear Equations

W. Hoffmann

Department of Mathematics, University of Amsterdam
Roetersstraat 15, 1018 WB Amsterdam, The Netherlands

W.M. Lioen

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

This document describes a set of NUMVEC FORTRAN Library routines, dealing with the unique solution of real linear systems. Presently, only highly optimized non-portable implementations for the CYBER 200 series computer systems are included in the Library.

Note: The implementations are available in FORTRAN 200 (the CYBER 200 series FORTRAN, a superset of standard FORTRAN including vector extensions).

Chapter ∗∗∗

Simultaneous Linear Equations

1. Scope of the Chapter

This chapter is concerned with the solution of the matrix equation $AX = B$, where B may be a single vector or a matrix of multiple right hand sides.

2. Background to the problem

A set of linear equations may be written in the form

$$Ax = b$$

where the known matrix A is of shape $(m \times n)$, the known right hand vector b has m components, and the required solution vector x has n components. There may also be p vectors b_i, $i = 1, 2, \cdots, p$ on the right hand side and the equation may then be written as

$$AX = B$$

the required matrix X having as its p columns the solutions of $Ax_i = b_i$, $i = 1, 2, \cdots, p$.

In the following we assume A to be a real square non singular matrix (i.e. $\text{rank}(A) = m = n$), so $Ax = b$ has a unique solution.

In the Gaussian elimination case, the computation starts with the LDU decomposition $A = PLDUQ^{-1}$, where L and U are lower and upper triangular matrices, respectively, D is a diagonal matrix and P and Q are row and column permutation matrices, respectively, chosen so as to ensure that the decomposition is numerically stable. The solution is then obtained by solving in succession the simpler equations

$$LDz = P^{-1}b$$

$$Uy = z$$

$$x = Qy$$

the first by forward substitution, the second by back substitution and the third simply by permutation of the elements of the solution vector.

The application of complete pivoting (both row and column interchanges), which yields a stable algorithm, has the disadvantage of expensive code. In 1971 Businger published an idea for calculating, with little extra cost, an upperbound for the growth factor during Gaussian elimination with partial pivoting (see [1]). If this upperbound for the growth factor becomes too large, the pivoting strategy can be switched to complete pivoting, so the efficiency of partial pivoting is combined with the stability of complete pivoting. This so-called mixed pivoting strategy provides the user with a reasonable upperbound for the growth factor, so that an estimate of the error matrix can be made (see [4]).

In the Gauß-Jordan elimination case, the computation starts with the diagonalization of the system $AX = B$ by elementary transformations, using a partial pivoting scheme.
The p columns x of the solution matrix X are found by dividing by the diagonal of the diagonalized matrix and permuting the elements of the solution vector.

Due to rounding errors the computed solution x_0, say, will always be an approximation to the true solution x both for Gaussian and Gauß-Jordan elimination. This approximation can be satisfactory, agreeing with x to several figures, but if the problem is ill-conditioned then x and x_0 may have few or even no figures in common.

A detailed description on the algorithmic choices concerning the normalization of D and the pivoting strategy can be found in [3].

2.1. References

[1]   Businger, P.A., Monitoring the numerical stability of Gaussian elimination. In: Numerische Mathematik 16, 1971, pp. 360-361.

[2]   Golub, G.H. and Van Loan, C.F., Matrix computations, North Oxford Academic, 1983.

[3]   Hoffmann, W., Solving linear systems on a vector computer, to appear.

[4]   Stewart, G.W., Introduction to matrix computations, Academic Press, 1973.

3. Recommendations on Choice and Use of Available Routines

Since in exceptional cases partial pivoting may yield useless results, even for well-conditioned matrices, the user is advised to use CCRMCF. If it is known beforehand, that partial pivoting will suffice, one can use CCRPCF.

GAUJOR should never be used just because of its capability of dealing with multiple right hand sides: if the problem is large enough the user should use a factorization routine (LURMMP or LURMPC) followed by a call to SLDUCP or SLDUPC, respectively, to solve the equations for each right hand side.
Only for 'small' problems (up to order about 50 for the CYBER 205) the user might want to use GAUJOR which, despite its somewhat greater complexity compared to Gaussian elimination, performs rather well due to the fact that all vectors remain of full length during every stage of the algorithm.

The Black Box routines CCRMCF and CCRPCF both simply call two general purpose routines and are provided for convenience.

Warning: No particular paging scheme is implemented, the matrices *A* and *C* have to fit in the working set in order to avoid thrashing (mortal paging).

Chapter Contents - ***

Routine   Purpose
name

Black Box Routines
These routines solve the matrix equation $AX = B$

CCRMCF   CCRMCF calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, by an LDU factorization method with mixed pivoting.

CCRPCF   CCRPCF calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, by an LDU factorization method with partial pivoting.

GAUJOR   GAUJOR calculates the approximate solution of a set of real linear equations with multiple right hand sides, $AX = B$, by Gauß-Jordan elimination with partial pivoting.

General Purpose Routines
These routines solve the matrix equation $AX = B$

CPIND   CPIND searches the element of maximum modulus (complete pivot) in the $(n - j + 1) \times (n - j + 1)$ submatrix of the $n \times n$ matrix A with the upper left corner of the submatrix corresponding with A_{jj}.

LURMMP   LURMMP decomposes a real matrix into one diagonal and two triangular matrices by an LDU factorization method with mixed pivoting.

LURMPC   LURMPC decomposes a real matrix into one diagonal and two triangular matrices by an LDU factorization method with partial pivoting.

SLDUCP   SLDUCP calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, where A has been decomposed using LURMMP.

SLDUPC   SLDUPC calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, where A has been decomposed using LURMPC.

<div align="center">**CCRMCF - Routine Document**</div>

1. Purpose

CCRMCF calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, by an LDU factorization method with mixed pivoting (see [2]).

2. Specification

```
      SUBROUTINE CCRMCF(A, IA, B, N, C, CPIV, RPIV,
   +                    EPS, GRWLIM, MAXNRM, UPBGRW, IFAIL)
C     INTEGER IA, N, CPIV(N), RPIV(N), IFAIL
C     REAL A(IA,N), B(N), C(N), EPS, GRWLIM, MAXNRM, UPBGRW
```

3. Description

Given a set of linear equations, $Ax = b$, the routine first decomposes A using an LDU factorization with mixed pivoting: $P^{-1}AQ = LDU$, where P and Q are permutation matrices, L is lower triangular, D is diagonal and U is upper triangular. The diagonals are normalized according to: $L_{ii} = U_{ii} = D_{ii}^{-1}$

Partial pivoting by column interchanges will be used as long as the calculated upper bound for the growth (see [1]) is less than a critical value, that equals GRWLIM $\times$ N $\times$ A$_\infty$. In practice, the upper bound for the growth rarely exceeds the critical value if we choose GRWLIM properly (see [1] and [4]), hence we will usually take advantage of the greater speed of partial pivoting (maximum search order N $- k + 1$ in the k-th step), while in doubtful cases numerical difficulties will be recognized and the process will switch to complete pivoting (maximum search order $(N - k + 1)^2$ in the k-th step). The process will also switch to complete pivoting if the modulus of the pivot obtained with partial pivoting is less than EPS.

The choice GRWLIM $< N^{-1}$ will result in complete pivoting only, while partial pivoting will be used in every step if we choose GRWLIM $> 2^{(N-1)} \times N^{-1}$.

Usually, GRWLIM $= 8$ will give good results (see [4]).

The approximate solution x is found by forward substitution and backward substitution in $LDz = P^{-1}b$, $Uy = z$ and $x = Qy$, where b is the right hand side.

4. References

[1] Businger, P.A., Monitoring the numerical stability of Gaussian elimination. In: Numerische Mathematik 16, 1971, pp. 360-361.

[2] Hoffmann, W., Gaussian elimination algorithms on a vector computer. Report 85-10, University of Amsterdam, Department of Mathematics, 1985.

[3] Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[4] Wilkinson, J.H., Rounding errors in algebraic processing. Prentice Hall, 1963, Chapter 3, Section 16.

[5] Wilkinson, J.H. and Reinsch, C., Handbook for Automatic Computation. Volume II, Linear Algebra. Springer-Verlag, 1971, pp. 93-110.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** $\geq$ N.
   Before entry, A should contain the elements of the real matrix.
   On successful exit, it will contain the LDU factorization with the diagonals of L and U understood.

IA - INTEGER.
   On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA $\geq$ N).

Unchanged on exit.

B - REAL array of DIMENSION at least (N).
   Before entry, B should contain the elements of the right hand side.
   Unchanged on exit, but see Section 11.

N - INTEGER.
   On entry, N specifies the order of matrix A.
   Unchanged on exit.

C - REAL array of DIMENSION at least (N).
   On successful exit, C will contain the solution vector.

CPIV - INTEGER array of DIMENSION at least (N).
RPIV - INTEGER array of DIMENSION at least (N).
   Used as working space.
   (On successful exit, CPIV and RPIV will contain the pivotal column indices and the pivotal row indices respectively.)

EPS - REAL.
   On entry, EPS must specify a small positive value used as threshold for singularity tests (e.g. the machine precision: X02AAF).
   Unchanged on exit.

GRWLIM - REAL.
   On entry, GRWLIM must specify a value which is used for controlling the pivoting strategy. Usually GRWLIM = 8 will give good results. See also Section 3.
   Unchanged on exit.

MAXNRM - REAL.
   On exit, MAXNRM contains the maximum norm A_∞ of the matrix argument.

UPBGRW - REAL.
   On exit, UPBGRW contains an upper bound for the growth factor. Usually, however, this will be a crude overestimate.

IFAIL - INTEGER.
   Before entry, IFAIL must be assigned a value. For users not familiar with this parameter (described in Chapter P01 of [3]) the recommended value is 0.
   Unless the routine detects an error (see Section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

IFAIL = 1
   Failure in LURMMP, the matrix *A* is singular, possibly due to rounding errors.

7. Auxiliary routines

This routine calls the NUMVEC Library routines CPIND, LURMMP, SLDUCP and P01AAF.

8. Timing

The time taken is approximately proportional to N^3.
T(ime) in seconds for various N on a CYBER 205 (1-pipe):

N	25	50	100	200	400

T (partial pivoting throughout)	0.0016	0.0057	0.0243	0.1181	0.6779
T (complete pivoting from the beginning)	0.0031	0.0122	0.0550	0.2171	1.3704

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed error analysis see [4], page 107.

11. Further comments

If the routine is called with the same name for parameters B and C then the solution vector will overwrite the right hand side.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions. Special call statements are used where appropriate (e.g. the pivot search is done with sign control).
The routine uses at most $2N + \max\{2N, \left\lceil \frac{1}{64}\min\{65535, IA \times N\} \right\rceil\}$ words dynamic storage.

12. Keywords

Approximate Solution of Linear Equations.
LDU Factorization.
Real Matrix.
Single Right Hand Side.

13. Example

To solve the set of linear equations $Ax = b$ where

$$A = \begin{bmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} -359 \\ 281 \\ 85 \end{bmatrix}$$

13.1. Program text

```
C
C        CCRMCF EXAMPLE PROGRAM TEXT
C        NUMVEC 1986
C        MARK 1
C
         PROGRAMME
         REAL A(4,4), B(6), C(6), WKS1(18), WKS2(18),
     +     EPS, GRWLIM, MAXNRM, UPBGRW
         INTEGER NIN, NOUT, I, N, J, IA, IFAIL
         DATA NIN /5/, NOUT /6/
         OPEN(UNIT=NIN, FILE=´INPUT´)
         OPEN(UNIT=NOUT, FILE=´OUTPUT´)
         READ (NIN,99999) (WKS1(I),I=1,7)
         WRITE (NOUT,99997) (WKS1(I),I=1,6)
```

```
              N = 3
              READ (NIN,99998) ((A(I,J),J=1,N),I=1,N), (B(I),I=1,N)
              IA = 4
              EPS = X02AAF(XXXX)
              GRWLIM = 8.
              IFAIL = 1
              CALL CCRMCF(A,IA,B,N,C,WKS1,WKS2,EPS,GRWLIM,MAXNRM,UPBGRW,IFAIL)
              WRITE (NOUT,99996) MAXNRM
              IF (IFAIL.NE.0) THEN
                WRITE (NOUT,99995) IFAIL
                 STOP
              ENDIF
              WRITE (NOUT,99994) (C(I),I=1,N)
              WRITE (NOUT,99993) UPBGRW
              STOP
        99999 FORMAT (6A4, 1A3)
        99998 FORMAT (3F5.0)
        99997 FORMAT (4(1X/), 1X, 5A4, 1A3, ´RESULTS´/1X)
        99996 FORMAT (´0MAXIMUM NORM OF MATRIX: ´, F5.0)
        99995 FORMAT (´0ERROR IN CCRMCF IFAIL = ´, I2)
        99994 FORMAT (´0SOLUTIONS´/(1X, F4.1))
        99993 FORMAT (´0UPPER BOUND FOR THE GROWTH FACTOR: ´, F7.4)
              END
```

13.2. Program data

```
CCRMCF EXAMPLE PROGRAM DATA
    33   16   72
   -24  -10  -57
    -8   -4  -17
  -359  281   85
```

13.3. Program results

```
CCRMCF EXAMPLE PROGRAM RESULTS


MAXIMUM NORM OF MATRIX:   72.

SOLUTIONS
 1.0
-2.0
-5.0


UPPER BOUND FOR THE GROWTH FACTOR:  2.0370
```

<div align="center">

CCRPCF - Routine Document

</div>

1. Purpose

CCRPCF calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, by an LDU factorization method with partial pivoting (see [1]).

2. Specification

```
        SUBROUTINE CCRPCF(A, IA, B, N, C, PIV, EPS, IFAIL)
C       INTEGER IA, N, PIV(N), IFAIL
C       REAL A(IA,N), B(N), C(N), EPS
```

3. Description

Given a set of linear equations, $Ax = b$, the routine first decomposes A using an LDU factorization with partial pivoting by column interchanges: $AP = LDU$, where P is a permutation matrix, L is lower triangular, D is diagonal and U is upper triangular. The diagonals are normalized according to: $L_{ii} = U_{ii} = D_{ii}^{-1}$. The approximate solution x is found by forward substitution and backward substitution in $LDz = b$, $Uy = z$ and $x = Py$, where b is the right hand side.

4. References

[1]   Hoffmann, W., Gaussian elimination algorithms on a vector computer. Report 85-10, University of Amsterdam, Department of Mathematics, 1985.

[2]   Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[3]   Wilkinson, J.H. and Reinsch, C., Handbook for Automatic Computation. Volume II, Linear Algebra. Springer-Verlag, 1971, pp. 93-110.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** ≥ N.
    Before entry, A should contain the elements of the real matrix.
    On successful exit, it will contain the LDU factorization with the diagonals of L and U understood.

IA - INTEGER.
    On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA ≥ N).
    Unchanged on exit.

B - REAL array of DIMENSION at least (N).
    Before entry, B should contain the elements of the right hand side.
    Unchanged on exit, but see Section 11.

N - INTEGER.
    On entry, N specifies the order of matrix A.
    Unchanged on exit.

C - REAL array of DIMENSION at least (N).
    On successful exit, C will contain the solution vector.

PIV - INTEGER array of DIMENSION at least (N).
    Used as working space.
    (On successful exit, PIV will contain the pivotal column indices.)

EPS - REAL.
    On entry, EPS must specify a small positive value used as threshold for singularity tests (e.g. the

machine precision: X02AAF).
Unchanged on exit.

IFAIL - INTEGER.
Before entry, IFAIL must be assigned a value. For users not familiar with this parameter (described in Chapter P01 of [2]) the recommended value is 0.
Unless the routine detects an error (see Section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

IFAIL = 1
Failure in LURMPC, the matrix A is singular, possibly due to rounding errors.

7. Auxiliary routines

This routine calls the NUMVEC Library routines LURMPC, SLDUPC and P01AAF.

8. Timing

The time taken is approximately proportional to N^3.
T(ime) in seconds for various N on a CYBER 205 (1-pipe):

n	25	50	100	200	400
T	0.0014	0.0051	0.0232	0.1154	0.6705

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed error analysis see [3], page 107.

11. Further comments

If the routine is called with the same name for parameters B and C then the solution vector will overwrite the right hand side.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions. Special call statements are used where appropriate (e.g. the pivotal column search is done with sign control).
The routine explicitly uses N words dynamic storage.

12. Keywords

Approximate Solution of Linear Equations.
LDU Factorization.
Real Matrix.
Single Right Hand Side.

13. Example

To solve the set of linear equations $Ax = b$ where

$$A = \begin{bmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{bmatrix} \qquad \text{and} \quad b = \begin{bmatrix} -359 \\ 281 \\ 85 \end{bmatrix}$$

13.1. Program text

```
C
C         CCRPCF EXAMPLE PROGRAM TEXT
C         NUMVEC 1986
C         MARK 1
C
          PROGRAMME
          REAL A(4,4), B(6), C(6), WKS(18), EPS
          INTEGER NIN, NOUT, I, N, J, IA, IFAIL
          DATA NIN /5/, NOUT /6/
          OPEN(UNIT=NIN, FILE='INPUT')
          OPEN(UNIT=NOUT, FILE='OUTPUT')
          READ (NIN,99999) (WKS(I),I=1,7)
          WRITE (NOUT,99997) (WKS(I),I=1,6)
          N = 3
          READ (NIN,99998) ((A(I,J),J=1,N),I=1,N), (B(I),I=1,N)
          IA = 4
          EPS = X02AAF(XXXX)
          IFAIL = 1
          CALL CCRPCF(A, IA, B, N, C, WKS, EPS, IFAIL)
          IF (IFAIL.NE.0) THEN
             WRITE (NOUT,99996) IFAIL
             STOP
          END IF
          WRITE (NOUT,99995) (C(I),I=1,N)
          STOP
99999     FORMAT (6A4, 1A3)
99998     FORMAT (3F5.0)
99997     FORMAT (4(1X/), 1X, 5A4, 1A3, 'RESULTS'/1X)
99996     FORMAT ('0ERROR IN CCRPCF IFAIL = ', I2)
99995     FORMAT ('0SOLUTIONS'/(1X, F4.1))
          END
```

13.2. Program data

```
CCRPCF EXAMPLE PROGRAM DATA
    33   16    72
   -24  -10   -57
    -8   -4   -17
  -359  281    85
```

13.3. Program results

```
CCRPCF EXAMPLE PROGRAM RESULTS
```

```
SOLUTIONS
 1.0
-2.0
-5.0
```

CPIND - Routine Document

1. Purpose

CPIND searches the element of maximum modulus (complete pivot) in the $(n - j + 1) \times (n - j + 1)$ submatrix of the $n \times n$ matrix A with the upper left corner of the submatrix corresponding with A_{jj}.

2. Specification

```
        SUBROUTINE CPIND(A, IA, N, J, CIND, RIND, ACR, IFAIL)
C       INTEGER IA, N, J, CIND, RIND, IFAIL
C       REAL A(IA,N), ACR
```

3. Description

CPIND searches the (n-j+1)*(n-j+1) submatrix starting in the (j,j)-th element of array A for the element of maximum modulus (complete pivot). The most efficient of the two following alternatives is chosen depending on an estimate of the timings: the first version works on vectors of maximal length: a bit-vector is used to skip those elements of A that do not belong to the submatrix; the second version first searches every column of the submatrix for the element of maximum modulus, and afterwards searches these elements for their maximum.

4. References

None.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** ≥ N.
Before entry, A should contain the elements of the real matrix.
Unchanged on exit.

IA - INTEGER.
On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA ≥ N).
Unchanged on exit.

N - INTEGER.
On entry, N specifies the order of matrix A.
Unchanged on exit.

J - INTEGER.
On entry, J specifies the upper left corner of the submatrix: A(J,J).
Unchanged on exit.

CIND - INTEGER.
RIND - INTEGER.
On exit, CIND and RIND will contain the column and row indices of the complete pivot element.

ACR - REAL
On exit, ACR will contain the value of the complete pivot element.

IFAIL - INTEGER.
Unused.
IFAIL contains 0 on exit.

6. Error indicators and warnings

None.

7. Auxiliary routines

None.

8. Timing

The time taken is approximately proportional to $(N-J+1)^2$.

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

Not applicable.

11. Further comments

None.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions. Special call statements are used where appropriate.

The routine uses at most $\max\{2(N-J+1), \left\lceil \frac{1}{64} \min\{65535, IA \times (N-J+1)\} \right\rceil\}$ words dynamic storage.

12. Keywords

Real Matrix.
Complete Pivoting.

13. Example

To search the element of maximum modulus in the 2×2 submatrix with the upper left corner corresponding with A_{22}, where

$$A = \begin{vmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{vmatrix}$$

13.1. Program text

```
C
C       CPIND  EXAMPLE PROGRAM TEXT
C       NUMVEC 1986
C       MARK 1
C
        PROGRAMME
        REAL A(4,4), WKS1(18), ACR
        INTEGER NIN, NOUT, I, N, J, IA, IFAIL, CIND, RIND
        DATA NIN /5/, NOUT /6/
        OPEN(UNIT=NIN, FILE=´INPUT´)
```

```
            OPEN(UNIT=NOUT, FILE='OUTPUT')
            READ (NIN,99999) (WKS1(I),I=1,7)
            WRITE (NOUT,99997) (WKS1(I),I=1,6)
            N = 3
            READ (NIN,99998) ((A(I,J),J=1,N),I=1,N)
            IA = 4
            J = 2
            IFAIL = 1
            CALL CPIND(A, IA, N, J, CIND, RIND, ACR, IFAIL)
            IF (IFAIL.NE.0) THEN
              WRITE (NOUT,99996) IFAIL
              STOP
            ENDIF
            WRITE (NOUT,99995) CIND, RIND, ACR
            STOP
99999   FORMAT (6A4, 1A3)
99998   FORMAT (3F5.0)
99997   FORMAT (4(1X/), 1X, 5A4, 1A3, 'RESULTS'/1X)
99996   FORMAT ('0ERROR IN CPIND IFAIL = ', I2)
99995   FORMAT ('0A(', I2, ',', I2, ') = ', F5.0)
            END
```

13.2. Program data

```
CPIND   EXAMPLE PROGRAM DATA
   33   16   72
  -24  -10  -57
   -8   -4  -17
```

13.3. Program results

```
CPIND   EXAMPLE PROGRAM RESULTS


A( 2, 3) =  -57.
```

GAUJOR - Routine Document

1. Purpose

GAUJOR calculates the approximate solution of a set of real linear equations with multiple right hand sides, $AX = B$, by Gauß-Jordan elimination with partial pivoting (see [1]).

2. Specification

        SUBROUTINE GAUJOR(A, IA, B, IB, N, M, C, IC, DIAG, PIV, EPS,IFAIL)
C   INTEGER IA, N, IB, IC, M, PIV(N), IFAIL
C   REAL A(IA,N), B(IB,M), C(IC,M), DIAG(N), EPS

3. Description

Given a set of linear equations, $AX = B$, the routine first diagonalizes A using elementary transformations with partial pivoting by rows. The partial pivoting is implemented without actually interchanging the rows.
The columns x of the solution X are found by dividing by the diagonal and permuting the rows of X.

4. References

[1]   Hoffmann, W., Solving linear systems on a vector computer, to appear.

[2]   Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[3]   Wilkinson, J.H. and Reinsch, C., Handbook for Automatic Computation. Volume II, Linear Algebra. Springer-Verlag, 1971, pp. 93-110.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** ≥ N.
    Before entry, A should contain the elements of the real matrix.
    The contents of A are altered on exit.

IA - INTEGER.
    On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA ≥ N).
    Unchanged on exit.

B - REAL array of DIMENSION (IB,**p**) where **p** ≥ M.
    Before entry, B should contain the elements of the M right hand sides stored in columns.
    Unchanged on exit, but see Section 11.

IB - INTEGER.
    On entry, IB specifies the first dimension of array B as declared in the calling (sub)program (IB ≥ N).
    Unchanged on exit.

N - INTEGER.
    On entry, N specifies the order of matrix A.
    Unchanged on exit.

M - INTEGER.
    On entry, M specifies the number of right hand sides.
    Unchanged on exit.

C - REAL array of DIMENSION (IC,**p**) where **p** ≥ M.
    On successful exit, C will contain the M solution vectors.

IC - INTEGER.

   On entry, IC specifies the first dimension of array C as declared in the calling (sub)program ($IC \geq N$).
   Unchanged on exit.

DIAG - INTEGER array of DIMENSION at least (N).

   Used as working space.
   (On successful exit, DIAG will contain the non permuted elements of the diagonalized matrix A.)

PIV - INTEGER array of DIMENSION at least (N).

   Used as working space.
   (On successful exit, PIV will contain the pivotal row indices.)

EPS - REAL.

   On entry, EPS must specify a small positive value used as threshold for singularity tests (e.g. the
   machine precision: X02AAF).
   Unchanged on exit.

IFAIL - INTEGER.

   Before entry, IFAIL must be assigned a value. For users not familiar with this parameter (described in
   Chapter P01 of [2]) the recommended value is 0.
   Unless the routine detects an error (see Section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

IFAIL = 1

   The matrix *A* is singular, possibly due to rounding errors.

7. Auxiliary routines

This routine calls the NUMVEC Library routine P01AAF.

8. Timing

The time taken is approximately proportional to N^3.
T(ime) in seconds for various N on a CYBER 205 (1-pipe):

n	25	50	100	200	400
T	0.0012	0.0051	0.0249	0.1369	0.8649

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed
error analysis see [3], page 107.

11. Further comments

If the routine is called with the same name for parameters B and C then the solution vectors will overwrite
the right hand sides.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions. Special call statements are used where appropriate (e.g. the pivotal row search is done with sign control).
The routine explicitly uses N words dynamic storage.

12. Keywords

Approximate Solution of Linear Equations.
Gauß-Jordan elimination.
Multiple Right Hand Sides.
Real Matrix.

13. Example

To solve the set of linear equations $AX = B$ where

$$A = \begin{bmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -359 \\ 281 \\ 85 \end{bmatrix}$$

13.1. Program text

```
C
C          GAUJOR EXAMPLE PROGRAM TEXT
C          NUMVEC 1986
C          MARK 1
C
           PROGRAMME
           REAL A(4,4), B(6), C(6), WKS1(18), WKS2(18), EPS
           INTEGER NIN, NOUT, I, N, J, IA, IFAIL
           DATA NIN /5/, NOUT /6/
           OPEN(UNIT=NIN, FILE=´INPUT´)
           OPEN(UNIT=NOUT, FILE=´OUTPUT´)
           READ (NIN,99999) (WKS1(I),I=1,7)
           WRITE (NOUT,99997) (WKS1(I),I=1,6)
           N = 3
           M = 1
           READ (NIN,99998) ((A(I,J),J=1,N),I=1,N), (B(I),I=1,N)
           IA = 4
           IB = 6
           IC = 6
           EPS = X02AAF(XXXX)
           IFAIL = 1
           CALL GAUJOR(A, IA, B, IB, N, M, C, IC, WKS1, WKS2, EPS, IFAIL)
           IF (IFAIL.NE.0) THEN
              WRITE (NOUT,99996) IFAIL
              STOP
           END IF
           WRITE (NOUT,99995) (C(I),I=1,N)
           STOP
99999      FORMAT (6A4, 1A3)
99998      FORMAT (3F5.0)
99997      FORMAT (4(1X/), 1X, 5A4, 1A3, ´RESULTS´/1X)
```

```
99996   FORMAT ('0ERROR IN GAUJOR IFAIL = ', I2)
99995   FORMAT ('0SOLUTIONS '/(1X, F4.1))
        END
```

13.2. Program data

```
GAUJOR EXAMPLE PROGRAM DATA
   33   16   72
  -24  -10  -57
   -8   -4  -17
 -359  281   85
```

13.3. Program results

```
GAUJOR EXAMPLE PROGRAM RESULTS


SOLUTIONS
 1.0
-2.0
-5.0
```

LURMMP - Routine Document

1. Purpose

LURMMP decomposes a real matrix into one diagonal and two triangular matrices by an LDU factorization method with mixed pivoting (see [1]).

2. Specification

```
        SUBROUTINE LURMMP(A, IA, N, CPIV, RPIV, EPS, GRWLIM, MAXNRM, UPBGRW, IFAIL)
C       INTEGER IA, N, CPIV(N), RPIV(N), IFAIL
C       REAL A(IA,N), EPS, GRWLIM, MAXNRM, UPBGRW
```

3. Description

The routine decomposes A using an LDU factorization with mixed pivoting: $P^{-1}AQ = LDU$, where P and Q are permutation matrices, L is lower triangular, D is diagonal and U is upper triangular. The diagonals are normalized according to: $L_{ii} = U_{ii} = D_{ii}^{-1}$

Partial pivoting by column interchanges will be used as long as the calculated upper bound for the growth (see [1]) is less than a critical value, that equals $\text{GRWLIM} \times \text{N} \times A_\infty$. In practice, the upper bound for the growth rarely exceeds the critical value if we choose GRWLIM properly (see [1] and [4]), hence we will usually take advantage of the greater speed of partial pivoting (maximum search order $\text{N} - k + 1$ in the k-th step), while in doubtful cases numerical difficulties will be recognized and the process will switch to complete pivoting (maximum search order $(\text{N} - k + 1)^2$ in the k-th step). The process will also switch to complete pivoting if the modulus of the pivot obtained with partial pivoting is less than EPS.

The choice $\text{GRWLIM} < \text{N}^{-1}$ will result in complete pivoting only, while partial pivoting will be used in every step if we choose $\text{GRWLIM} > 2^{(\text{N}-1)} \times \text{N}^{-1}$.

Usually, $\text{GRWLIM} = 8$ will give good results (see [4]).

4. References

[1] Businger, P.A., Monitoring the numerical stability of Gaussian elimination. In: Numerische Mathematik 16, 1971, pp. 360-361.

[2] Hoffmann, W., Gaussian elimination algorithms on a vector computer. Report 85-10, University of Amsterdam, Department of Mathematics, 1985.

[3] Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[4] Wilkinson, J.H., Rounding errors in algebraic processing. Prentice Hall, 1963, Chapter 3, Section 16.

[5] Wilkinson, J.H. and Reinsch, C., Handbook for Automatic Computation. Volume II, Linear Algebra. Springer-Verlag, 1971, pp. 93-110.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** ≥ N.
     Before entry, A should contain the elements of the real matrix.
     On successful exit, it will contain the LDU factorization with the diagonals of L and U understood.

IA - INTEGER.
     On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA ≥ N).
     Unchanged on exit.

N - INTEGER.
     On entry, N specifies the order of matrix A.

Unchanged on exit.

CPIV - INTEGER array of DIMENSION at least (N).
RPIV - INTEGER array of DIMENSION at least (N).
On successful exit, CPIV and RPIV will contain the pivotal column indices and the pivotal row indices respectively.

EPS - REAL.
On entry, EPS must specify a small positive value used as threshold for singularity tests (e.g. the machine precision: X02AAF).
Unchanged on exit.

GRWLIM - REAL.
On entry, GRWLIM must specify a value which is used for controlling the pivoting strategy. Usually GRWLIM = 8 will give good results. See also Section 3.
Unchanged on exit.

MAXNRM - REAL.
On exit, MAXNRM contains the maximum norm A_∞ of the matrix argument.

UPBGRW - REAL.
On exit, UPBGRW contains an upper bound for the growth factor. Usually, however, this will be a crude overestimate.

IFAIL - INTEGER.
Before entry, IFAIL must be assigned a value. For users not familiar with this parameter (described in Chapter P01 of [3]) the recommended value is 0.
Unless the routine detects an error (see Section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

IFAIL = 1
The matrix *A* is singular, possibly due to rounding errors.

7. Auxiliary routines

This routine calls the NUMVEC Library routines CPIND and P01AAF.

8. Timing

The time taken is approximately proportional to N^3.

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed error analysis see [4], page 107.

11. Further comments

None.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions. Special call statements are used where appropriate (e.g. the pivot search is done with sign control).

The routine uses at most $2N + \max\{2N, \left\lceil \frac{1}{64} \min\{65535, IA \times N\} \right\rceil\}$ words dynamic storage.

12. Keywords

LDU Factorization.
Real Matrix.

13. Example

See SLDUCP.

LURMPC - Routine Document

1. Purpose

LURMPC decomposes a real matrix into one diagonal and two triangular matrices by an LDU factorization method with partial pivoting (see [1]).

2. Specification

```
        SUBROUTINE LURMPC(A, IA, N, PIV, EPS, IFAIL)
C       INTEGER IA, N, PIV(N), IFAIL
C       REAL A(IA,N), EPS
```

3. Description

The routine decomposes A using an LDU factorization with partial pivoting by column interchanges: $AP = LDU$, where P is a permutation matrix, L is lower triangular, D is diagonal and U is upper triangular. The diagonals are normalized according to: $L_{ii} = U_{ii} = D_{ii}^{-1}$.

4. References

[1] Hoffmann, W., Gaussian elimination algorithms on a vector computer. Report 85-10, University of Amsterdam, Department of Mathematics, 1985.

[2] Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[3] Wilkinson, J.H. and Reinsch, C., Handbook for Automatic Computation. Volume II, Linear Algebra. Springer-Verlag, 1971, pp. 93-110.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** ≥ N.
Before entry, A should contain the elements of the real matrix.
On successful exit, it will contain the LDU factorization with the diagonals of L and U understood.

IA - INTEGER.
On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA ≥ N).
Unchanged on exit.

N - INTEGER.
On entry, N specifies the order of matrix A.
Unchanged on exit.

PIV - INTEGER array of DIMENSION at least (N).
On successful exit, PIV will contain the pivotal column indices.

EPS - REAL.
On entry, EPS must specify a small positive value used as threshold for singularity tests (e.g. the machine precision: X02AAF).
Unchanged on exit.

IFAIL - INTEGER.
Before entry, IFAIL must be assigned a value. For users not familiar with this parameter (described in Chapter P01 of [2]) the recommended value is 0.
Unless the routine detects an error (see Section 6), IFAIL contains 0 on exit.

6. Error indicators and warnings

Errors detected by the routine:-

IFAIL = 1
  The matrix *A* is singular, possibly due to rounding errors.

7. Auxiliary routines

This routine calls the NUMVEC Library routine P01AAF.

8. Timing

The time taken is approximately proportional to N^3.

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed error analysis see [3], page 107.

11. Further comments

None.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions. Special call statements are used where appropriate (e.g. the pivotal column search is done with sign control).
The routine explicitly uses N words dynamic storage.

12. Keywords

LDU Factorization.
Real Matrix.

13. Example

See SLDUPC.

<div align="center">

SLDUCP - Routine Document

</div>

1. Purpose

SLDUCP calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, where A has been decomposed using LURMMP (see [2]).

2. Specification

        SUBROUTINE SLDUCP(A, IA, B, N, C, CPIV, RPIV, IFAIL)
C   INTEGER IA, N, CPIV(N), RPIV(N), IFAIL
C   REAL A(IA,N), B(N), C(N)

3. Description

The routine solves $Ax = b$ where A is real and b is the right hand side. The routine must be preceded by a call to LURMMP which calculates a permutation of the LDU factorization: $P^{-1}AQ = LDU$.
The approximate solution x is found by forward substitution and backward substitution in $LDz = P^{-1}b$, $Uy = z$ and $x = Qy$, where b is the right hand side.

4. References

[1] Businger, P.A., Monitoring the numerical stability of Gaussian elimination. In: Numerische Mathematik 16, 1971, pp. 360-361.

[2] Hoffmann, W., Gaussian elimination algorithms on a vector computer. Report 85-10, University of Amsterdam, Department of Mathematics, 1985.

[3] Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[4] Wilkinson, J.H., Rounding errors in algebraic processing. Prentice Hall, 1963, Chapter 3, Section 16.

[5] Wilkinson, J.H. and Reinsch, C., Handbook for Automatic Computation. Volume II, Linear Algebra. Springer-Verlag, 1971, pp. 93-110.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** $\geq$ N.
    Before entry, A must contain the LDU decomposition, as given by LURMMP.
    Unchanged on exit.

IA - INTEGER.
    On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA $\geq$ N).
    Unchanged on exit.

B - REAL array of DIMENSION at least (N).
    Before entry, B should contain the elements of the right hand side.
    Unchanged on exit, but see Section 11.

N - INTEGER.
    On entry, N specifies the order of matrix A.
    Unchanged on exit.

C - REAL array of DIMENSION at least (N).
    On successful exit, C will contain the solution vector.

CPIV - INTEGER array of DIMENSION at least (N).
RPIV - INTEGER array of DIMENSION at least (N).
  Before entry CPIV and RPIV must contain the details of the column and row interchanges in the LDU factorization as given by LURMMP.
  Unchanged on exit.

IFAIL - INTEGER.
  Unused.
  IFAIL contains 0 on exit.

6. Error indicators and warnings

None.

7. Auxiliary routines

None.

8. Timing

The time taken is approximately proportional to N^2.

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed error analysis see [4], page 107.

11. Further comments

If the routine is called with the same name for parameters B and C then the solution vector will overwrite the right hand side.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions.
The routine explicitly uses N words dynamic storage.

12. Keywords

Approximate Solution of Linear Equations.
LDU Factorization.
Real Matrix.
Single Right Hand Side.

13. Example

To solve the set of linear equations $Ax = b$ where

$$A = \begin{bmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} -359 \\ 281 \\ 85 \end{bmatrix}$$

13.1. Program text

```
C
C       SLDUCP EXAMPLE PROGRAM TEXT
C       NUMVEC 1986
C       MARK 1
C
        PROGRAMME
        REAL A(4,4), B(6), C(6), WKS1(18), WKS2(18),
     +     EPS, GRWLIM, MAXNRM, UPBGRW
        INTEGER NIN, NOUT, I, N, J, IA, IFAIL
        DATA NIN /5/, NOUT /6/
        OPEN(UNIT=NIN, FILE=´INPUT´)
        OPEN(UNIT=NOUT, FILE=´OUTPUT´)
        READ (NIN,99999) (WKS1(I),I=1,7)
        WRITE (NOUT,99997) (WKS1(I),I=1,6)
        N = 3
        READ (NIN,99998) ((A(I,J),J=1,N),I=1,N), (B(I),I=1,N)
        IA = 4
        EPS = X02AAF(XXXX)
        GRWLIM = 8.
        IFAIL = 1
        CALL LURMMP(A,IA,N,WKS1,WKS2,EPS,GRWLIM,MAXNRM,UPBGRW,IFAIL)
        WRITE (NOUT,99996) MAXNRM
        IF (IFAIL.NE.0) THEN
          WRITE (NOUT,99995) IFAIL
           STOP
        ENDIF
        CALL SLDUCP(A, IA, B, N, C, WKS1, WKS2, IFAIL)
        WRITE (NOUT,99994) (C(I),I=1,N)
        WRITE (NOUT,99993) UPBGRW
        STOP
99999   FORMAT (6A4, 1A3)
99998   FORMAT (3F5.0)
99997   FORMAT (4(1X/), 1X, 5A4, 1A3, ´RESULTS´/1X)
99996   FORMAT (´0MAXIMUM NORM OF MATRIX: ´, F5.0)
99995   FORMAT (´0ERROR IN SLDUCP IFAIL = ´, I2)
99994   FORMAT (´0SOLUTIONS´/(1X, F4.1))
99993   FORMAT (´0UPPER BOUND FOR THE GROWTH FACTOR: ´, F7.4)
        END
```

13.2. Program data

```
SLDUCP EXAMPLE PROGRAM DATA
   33    16    72
  -24   -10   -57
   -8    -4   -17
 -359   281    85
```

13.3. Program results

```
SLDUCP EXAMPLE PROGRAM RESULTS


MAXIMUM NORM OF MATRIX:    72.

SOLUTIONS
 1.0
-2.0
-5.0

UPPER BOUND FOR THE GROWTH FACTOR:  2.0370
```

SLDUPC - Routine Document

1. Purpose

SLDUPC calculates the approximate solution of a set of real linear equations with a single right hand side, $Ax = b$, where A has been decomposed using LURMPC (see [1]).

2. Specification

```
        SUBROUTINE SLDUPC(A, IA, B, N, C, PIV, IFAIL)
C       INTEGER IA, N, PIV(N), IFAIL
C       REAL A(IA,N), B(N), C(N)
```

3. Description

The routine solves $Ax = b$ where A is real and b is the right hand side. The routine must be preceded by a call to LURMPC which calculates a column permutation of the LDU factorization: $AP = LDU$.
The approximate solution x is found by forward substitution and backward substitution in $LDz = b$, $Uy = z$ and $x = Py$, where b is the right hand side.

4. References

[1]   Hoffmann, W., Gaussian elimination algorithms on a vector computer. Report 85-10, University of Amsterdam, Department of Mathematics, 1985.

[2]   Numerical Algorithms Group, NAG FORTRAN library manual - mark 11, 1984.

[3]   Wilkinson, J.H. and Reinsch, C., Handbook for Automatic Computation. Volume II, Linear Algebra. Springer-Verlag, 1971, pp. 93-110.

5. Parameters

A - REAL array of DIMENSION (IA,**p**) where **p** $\geq$ N.
   Before entry, A must contain the LDU decomposition, as given by LURMPC
   Unchanged on exit.

IA - INTEGER.
   On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA $\geq$ N).
   Unchanged on exit.

B - REAL array of DIMENSION at least (N).
   Before entry, B should contain the elements of the right hand side.
   Unchanged on exit, but see Section 11.

N - INTEGER.
   On entry, N specifies the order of matrix A.
   Unchanged on exit.

C - REAL array of DIMENSION at least (N).
   On successful exit, C will contain the solution vector.

PIV - INTEGER array of DIMENSION at least (N).
   Before entry, PIV must contain details of the column interchanges in the LDU factorization as given by LURMPC.
   Unchanged on exit.

IFAIL - INTEGER.
   Unused.

IFAIL contains 0 on exit.

6. Error indicators and warnings

None.

7. Auxiliary routines

None.

8. Timing

The time taken is approximately proportional to N^2.

9. Storage

There are no internally declared arrays, but see Section 11.1.

10. Accuracy

The accuracy of the computed solution depends on the conditioning of the original matrix. For a detailed error analysis see [3], page 107.

11. Further comments

If the routine is called with the same name for parameters B and C then the solution vector will overwrite the right hand side.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions.
The routine explicitly uses N words dynamic storage.

12. Keywords

Approximate Solution of Linear Equations.
LDU Factorization.
Real Matrix.
Single Right Hand Side.

13. Example

To solve the set of linear equations $Ax = b$ where

$$A = \begin{bmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} -359 \\ 281 \\ 85 \end{bmatrix}$$

13.1. Program text

```
C
C       SLDUPC EXAMPLE PROGRAM TEXT
C       NUMVEC 1986
C       MARK 1
C
        PROGRAMME
```

```
      REAL A(4,4), B(6), C(6), WKS(18), EPS
      INTEGER NIN, NOUT, I, N, J, IA, IFAIL
      DATA NIN /5/, NOUT /6/
      OPEN(UNIT=NIN, FILE=´INPUT´)
      OPEN(UNIT=NOUT, FILE=´OUTPUT´)
      READ (NIN,99999) (WKS(I),I=1,7)
      WRITE (NOUT,99997) (WKS(I),I=1,6)
      N = 3
      READ (NIN,99998) ((A(I,J),J=1,N),I=1,N), (B(I),I=1,N)
      IA = 4
      EPS = X02AAF(XXXX)
      IFAIL = 1
      CALL LURMPC(A, IA, N, WKS, EPS, IFAIL)
      IF (IFAIL.NE.0) THEN
        WRITE (NOUT,99996) IFAIL
        STOP
      END IF
      CALL SLDUPC(A, IA, B, N, C, WKS, IFAIL)
      WRITE (NOUT,99995) (C(I),I=1,N)
      STOP
99999 FORMAT (6A4, 1A3)
99998 FORMAT (3F5.0)
99997 FORMAT (4(1X/), 1X, 5A4, 1A3, ´RESULTS´/1X)
99996 FORMAT (´0ERROR IN SLDUPC IFAIL = ´, I2)
99995 FORMAT (´0SOLUTIONS´/(1X, F4.1))
      END
```

13.2. Program data

```
SLDUPC EXAMPLE PROGRAM DATA
   33   16   72
  -24  -10  -57
   -8   -4  -17
 -359  281   85
```

13.3. Program results

```
SLDUPC EXAMPLE PROGRAM RESULTS


SOLUTIONS
 1.0
-2.0
-5.0
```