Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

W. Hoffmann

NUMVEC FORTRAN Library manual
Chapter: Simultaneous linear equations
Update # 1

Department of Numerical Mathematics   Report NM-R8712   May

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

NUMVEC FORTRAN Library manual

Chapter: Simultaneous Linear Equations

Update #1

W. Hoffmann

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

This document describes two NUMVEC FORTRAN Library routines, INVGJ and GJPCF.
INVGJ calculates the approximate inverse of a real square matrix by Gauß-Jordan elimination with partial pivoting using column interchanges.
GJPCF 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 using column interchanges. GJPCF replaces routine GAUJOR, documented in Report NM-R8614, May 1986.

1980 Mathematics subject classification (1985 revision): 65V05, 65F05, 15A06
1982 CR Categories: 5.14.

Key Words & Phrases: Gauß-Jordan elimination, linear equations, software.

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

Report NM-R8712
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

GJPCF - NUMVEC FORTRAN Library Routine Document

1. Purpose

GJPCF 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 using column interchanges (see [1]).

2. Specification

```
      SUBROUTINE GJPCF(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 firstly diagonalizes A using elementary transformations with partial pivoting by column-interchanges. The columns of X are found by dividing by the diagonal and permuting the rows of X.
The columns x of the solution X are found by dividing by the diagonal and permuting the rows of X.

4. References

[1] Dekker, T.J. and Hoffmann, W., Rehabilitation of the Gauss-Jordan algorithm; Report 86-28, Math Inst., Univ of Amsterdam, 1986.

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

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 $\geqslant$ 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 $\geqslant$ N).
Unchanged on exit.

DIAG - INTEGER array of DIMENSION at least (N).
Used as working space.
(On successful exit, DIAG will contain the 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 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) times a quantity that reflects the order of magnitude of the matrix, like a norm of the matrix).
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 = k
The matrix A is singular, possibly due to rounding errors; in the k-th stage of the elimination process the pivot was less than EPS.

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	0.0013	0.0053	0.0252	0.1394	0.8739

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 [1].

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
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

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

13.1. Program text

```
C
C      GJPCF EXAMPLE PROGRAM TEXT
C      NUMVEC 1987
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 GJPCF(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 GJPCF IFAIL = ', I2)
99995 FORMAT ('0SOLUTIONS'/(1X, F4.1))
      END
```

13.2. Program data

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

13.3. Program results

```
GJPCF EXAMPLE PROGRAM RESULTS


SOLUTIONS
 1.0
-2.0
-5.0
```

INVGJ - NUMVEC FORTRAN Library Routine Document

1. Purpose

INVGJ calculates the approximate inverse of a square real matrix 'in situ' by Gauß-Jordan elimination with partial pivoting using column interchanges (see [1]).

2. Specification

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

3. Description

The inverse of the given real matrix A is constructed by accumulating the Gauß-Jordan transformations that diagonalize A. Partial pivoting by column interchanges is used throughout.
The result is divided by the calculated diagonal matrix and by the inverse of the permutation matrix, that corresponds with the column interchanges.

4. References

[1] Dekker, T.J. and Hoffmann, W., Rehabilitation of the Gauss-Jordan algorithm; Report 86-28, Math Inst., Univ of Amsterdam, 1986.

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

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, A contains the inverse matrix.

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.

DIAG - INTEGER array of DIMENSION at least (N).
Used as working space.
(On successful exit, DIAG will contain the 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 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) times a quantity that reflects the order of magnitude of the matrix, like a norm of the matrix).

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 $= k$

The matrix A is singular, possibly due to rounding errors; in the k-th stage of the elimination process the pivot was less than EPS.

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	0.0023	0.0098	0.0484	0.2719	1.7230

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 [1].

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 explicitly uses N words dynamic storage.

12. Keywords

Inversion
Gauß-Jordan elimination.
Real Matrix.

13. Example

To find the inverse of the 3 × 3 matrix:

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

13.1. Program text

```
C
C      INVGJ EXAMPLE PROGRAM TEXT
C      NUMVEC 1987
C      MARK 1
C
       PROGRAMME
       REAL A(5,5), WKS1(7), WKS2(7), 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
       READ (NIN,99998) ((A(I,J),J=1,N),I=1,N)
       IA = 5
       EPS = X02AAF(XXXX)
       IFAIL = 1
       CALL INVGJ (A, IA, N, WKS1, WKS2, EPS, IFAIL)
       IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99996) IFAIL
         STOP
       END IF
       WRITE (NOUT,99995) ((A(I,J),J=1,N),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 INVGJ IFAIL = ', I2)
99995  FORMAT ('0INVERSE'/(1X, 3F10.4))
       END
```

13.2. Program data

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

13.3. Program results

```
INVGJ  EXAMPLE PROGRAM RESULTS


INVERSE
   -9.6667   -2.6667  -32.0000
    8.0000    2.5000   25.5000
    2.6667    0.6667    9.0000
```