Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

P.P.M. de Rijk

NUMVEC FORTRAN library manual
Chapter: Simultaneous linear equations
Routine: SVDTJP and LSQMNS

Department of Numerical Mathematics Report NM-R8719 November

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

Routine: SVDTJP and LSQMNS

P.P.M. de Rijk

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

This document describes two NUMVEC FORTRAN Library routines.
SVDTJP computes the Singular Value Decomposition of a real rectangular matrix A, using a one-sided Jacobi algorithm.
LSQMNS finds the minimum norm least-squares solution of a system $Ax \approx b$, where the real rectangular matrix A has been decomposed into its Singular Value Decomposition using SVDTJP.

1980 Mathematics subject classification (1985 revision): 65F15, 65F20, 65F25, 15A18.
1982 CR Categories: 5.14.

Keywords & Phrases: least squares problems, one-sided Jacobi algorithm, rank deficiency, Singular Value Decomposition, vector computing.

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

Report NM-R8719
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

NUMVEC
is a library of NUMerical software for VECtor and parallel computers in FORTRAN.
The documentation conforms as much as possible to that of the NAG - library. A Subsection: *11.1 Vectorization information* mentions, a.o., whether the code is standard FORTRAN 77. If not, information is given about special vector-syntax used and about specitic machine(s) to which the code is aimed.
The source code described can be obtained by writing to the NUMVEC Library Manager at the CWI.

SVDTJP - NUMVEC FORTRAN Library Routine Document

1. Purpose

SVDTJP computes the singular value decomposition of a real rectangular matrix A.

2. Specification

```
      SUBROUTINE SVDTJP(M,N,A,LDA,JOB,NRANK,SVAL,SVEC,LDSVEC,WORK,IFAIL)
C     INTEGER    M, N, LDA, JOB, NRANK, LDSVEC, IFAIL
C     REAL       A(LDA,N), SVAL(N), SVEC(LDSVEC,N), WORK(N)
```

3. Description

Let A be a real $m \times n$ matrix and $p = min(m,n)$. A may be factorized by the singular value decomposition (SVD) as $A = U\Sigma V^T$, where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix and Σ is an $m \times n$ diagonal matrix with diagonal elements σ_i, $i=1,2, \cdots ,p$, such that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_p \geqslant 0$. The columns of U are the left singular vectors, the columns of V the right singular vectors and the diagonal elements of Σ the singular values of A. Let $r = \text{rank}(A)$, then $r \leqslant p$ and $\sigma_{r+1} = \cdots = \sigma_p = 0$, so $A = U_r \Sigma_r V_r^T$, where U_r is the $m \times r$ matrix formed by the first r columns of U, V_r is the $n \times r$ matrix formed by the first r columns of V and Σ_r is the square r^{th} order upper left part of Σ.
This routine returns r, U_r, Σ and optionally V, using a one-sided threshold Jacobi algorithm with pivoting (see [1]).

4. References

[1] P. P. M. de Rijk, A one-sided Jacobi algorithm for computing the singular value decompostion on a vector computer. Report 86-21, University of Amsterdam, Department of Mathematics, 1986.

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

5. Parameters

M - INTEGER.
On entry, M must specify the number of rows of matrix A.
Unchanged on exit.

N - INTEGER.
On entry, N must specify the number of columns of matrix A.
Unchanged on exit.

A - REAL array of DIMENSION (LDA, s), where s ⩾ N.
Before entry, the leading M × N part of A must contain the matrix to be decomposed.
Unchanged on exit, unless the routine is called with the same array supplied for both A and SVEC.

LDA - INTEGER.
On entry, LDA must specify the leading dimension of array A as declared in the calling (sub)program (LDA ⩾ M).
Unchanged on exit.

JOB - INTEGER.

On entry, JOB must be zero if the right singular vectors are required. If, on entry, JOB is not equal to zero, the right singular vectors are not calculated.
Unchanged on exit.

NRANK - INTEGER.

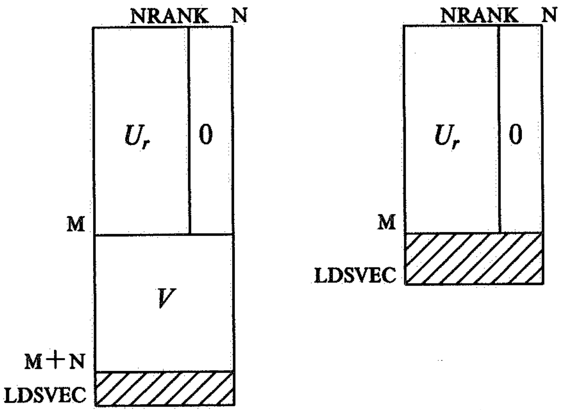
On successful exit, NRANK returns the numerical rank of matrix A.

SVAL - REAL array of DIMENSION at least (N).

On successful exit, SVAL contains the singular values of A arranged in descending order.

SVEC - REAL array of DIMENSION (LDA, t), where t $\geqslant$ N.

On successfull exit, the leading M $\times$ N part of SVEC (i.e. the first M rows) contains the first N left singular vectors (stored columnwise), of which the first NRANK columns are orthonormal and the subsequent N$-$NRANK columns are zero vectors. In addition, if JOB = 0, the adjacent N rows (i.e. the rows M+1 to M+N) contain the N right singular vectors (stored columnwise); otherwise the rest of SVEC is not referenced. Thus, SVEC looks as follows:



SVEC
(JOB = 0)

SVEC
(JOB $\neq$ 0)

LDSVEC - INTEGER.

On entry, LDSVEC must specify the leading dimension of array SVEC as declared in the calling (sub)program (if JOB = 0, then LDSVEC $\geqslant$ M+N, else LDSVEC $\geqslant$ M).
Unchanged on exit.

WORK - REAL array of DIMENSION at least (N).

Used as workspace.

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 next section), IFAIL contains zero on exit.

6. Error Indicators and Warnings

Errors detected by the routine:-

IFAIL = 1

M<1 or N<1 or M>LDA or
{(JOB = 0) and (M+N>LDSVEC)} or {(JOB $\neq$ 0) and (M>LDSVEC)}

IFAIL = 2
The Jacobi algorithm has failed to converge in 30 sweeps.

7. Auxilary Routines

This routine calls the NUMVEC Library routines X02AAF and P01AAF.

8. Timing

If the routine is called with JOB = 0, the time taken is approximately proportional to $(3M+2N)N^2/2$.
If the routine is called with JOB $\neq$ 0, the time taken is approximately proportional to $3MN^2/2$.

9. Storage

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

10. Accuracy

The calculated SVD exactly satisfies

$$U\Sigma V^T = A+E,$$

where

$$\|E\|_2 \leqslant c \times \mu \times \|A\|_2,$$

μ being the machine accuracy and c a modest function of M and N.

11. Further Comments

In case of M<N, it is recommended to call the routine with the transposed matrix A^T, interchanging the roles of U and V. This is more economical in storage requirements for SVEC (see Section 5) and requires less time (see Section 8).

When required, the complete $m \times m$ matrix U can be calculated by factorizing U_r into

$$U_r = QR,$$

where Q is an $m \times m$ orthogonal matrix and R is an $r \times r$ upper triangular matrix; since the columns of U_r are orthonormal, R is the identity matrix and the first r columns of Q are equal to the first r columns of U_r. This factorization can, for instance, be calculated by using the NAG Library routine F01QAF (see [2]).

11.1. Vectorization Information

The routine is written in FORTRAN200, making use of its vector syntax extensions.
The routine uses at most M+N words dynamic storage.

Since the transformations used to generate the matrices U and V are identical, matrix V is placed adjacent to U in SVEC. In doing so, every two individual vector operations on the columns of U and V respectively, now become one vector operation on a column of SVEC, thereby reducing the total sum of vector startup times.

12. Keywords

Real Rectangular Matrix, Rank, Singular Value Decomposition, One-sided Jacobi Algorithm.

13. Example

See LSQMNS.

LSQMNS - NUMVEC FORTRAN Library Routine Document

1. Purpose

LSQMNS finds the minimum norm least-squares solution of a system $Ax \approx b$, where the real $m \times n$ matrix A has been decomposed into its singular value decomposition using SVDTJP (see [2]) and b is a real m element vector.

2. Specification

```
        SUBROUTINE LSQMNS(M, N, NRANK, SVAL, SVEC, LDSVEC, B, X, IFAIL)
C       INTEGER    M, N, NRANK, LDSVEC, IFAIL
C       REAL       SVAL(N), SVEC(LDSVEC,N), B(M), X(N)
```

3. Description

The minimum norm least-squares solution of the problem $Ax \approx b$ is the vector x of minimum (Euclidean) norm which minimizes the norm of the residual vector $b - Ax$.
The real $m \times n$ matrix A with rank r may be factorized by the (reduced) singular value decomposition (SVD; see[2]) as $A = U_r \Sigma_r V_r^T$, where U_r is an $m \times r$ matrix having orthonormal columns, V_r is an $n \times r$ matrix having orthonormal columns and Σ_r is an $r \times r$ diagonal matrix with diagonal elements σ_i, $i = 1,2, \cdots ,r$, such that $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$. The columns of U_r are the left singular vectors, the columns of V_r the right singular vectors and the diagonal elements of Σ_r the singular values of A.
Given the SVD of A, the minimum norm least-squares solution is given by

$$x = \sum_{j=1}^{k} (u_j^T b \,/\, \sigma_j) v_j,$$

where u_j and v_j are the j^{th} left and right singular vector respectively and $k \leqslant r$ is such that the singular values $\sigma_{k+1}, \ldots, \sigma_r$ are negligible, but σ_k is not negligible, relative to the data errors in A; k is called the effective rank of A.
This routine must be preceded by a call to SVDTJP (with parameter JOB set to zero) which calculates the SVD of A.

4. References

[1] C. L. Lawson and R. J. Hanson, Solving least squares problems. Prentice Hall, New Jersey, 1974.

[2] P. P. M. de Rijk, A one-sided Jacobi algorithm for computing the singular value decompostion on a vector computer. Report 86-21, University of Amsterdam, Department of Mathematics, 1986.

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

5. Parameters

M - INTEGER.
On entry, M must specify the number of rows of the original matrix A.
Unchanged on exit.

N - INTEGER.
On entry, N must specify the number of columns of the original matrix A.

Unchanged on exit.

NRANK - INTEGER.

On entry, NRANK must specify the effective rank of matrix A. For a choice on NRANK, see [1]. NRANK must be less than or equal to the numerical rank as returned by SVDTJP (In problems where the matrix elements are known exactly or are contaminated only by roundoff error, NRANK can be set to the numerical rank as returned by SVDTJP).
Unchanged on exit.

SVAL - REAL array of DIMENSION at least (NRANK).

Before entry, SVAL must contain the first NRANK singular values of matrix A, as given by SVDTJP.
Unchanged on exit.

SVEC - REAL array of DIMENSION(LDA,t), where t $\geqslant$ NRANK.

Before entry, SVEC must contain the first NRANK (left and right) singular vectors of matrix A, as given by SVDTJP.
Unchanged on exit.

LDSVEC - INTEGER.

On entry, LDSVEC must specify the leading dimension of array SVEC as declared in the calling (sub)program (LDSVEC $\geqslant$ M+N).
Unchanged on exit.

B - REAL array of DIMENSION at least (M).

Before entry, B must contain the M element vector b.
Unchanged on exit.

X - REAL array of DIMENSION at least (N).

On successful exit, the first N elements of X contain the minimum norm least-squares solution vector x.

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 next section), IFAIL contains zero on exit.

6. Error

Errors detected by the routine:-

IFAIL = 1

M<1 or N<1 or M+N>LDSVEC.

IFAIL = 2

The numerical rank of matrix A is less than NRANK.

7. Auxilary Routines

This routine calls the NUMVEC Library routine P01AAF.

8. Timing

The time taken is approximately proportional to (M+N) × NRANK.

9. Storage

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

10. Accuracy

The accuracy of the computed solution depends on the condition of the original matrix. Furthermore, the solution x depends on the choice of NRANK. For a detailed error analysis see [1], especially pp. 50, 95 and 196-198.

11. Further Comments

11.1. Vectorization Information

The routine is written in FORTRAN200, making use of its vector syntax extensions. Special call statements are called where appropiate.
The routine explicitly uses NRANK words dynamic storage.

12. Keywords

Least-Squares Problems, Minimum Norm Least-Squares Solution, Rank Deficiency, Singular Value Decomposition.

13. Example

To obtain the minimal least squares solution for $Ax \approx b$ where

$$A = \begin{pmatrix} 0.05 & 0.05 & 0.25 & -0.25 \\ 0.25 & 0.25 & 0.05 & -0.05 \\ 0.35 & 0.35 & 1.75 & -1.75 \\ 1.75 & 1.75 & 0.35 & -0.35 \\ 0.30 & -0.30 & 0.30 & 0.30 \\ 0.40 & -0.40 & 0.40 & 0.40 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

First the matrix A is factorized into its singular value decomposition (SVD) using the NUMVEC Library routine SVDTJP.

13.1. Program text

```
C
C      LSQMNS EXAMPLE PROGRAM TEXT
C      NUMVEC 1987
C      MARK 1
C
       PROGRAMME
C      .. LOCAL SCALARS ..
       INTEGER I, IFAIL, J, JOB, LDA, LDSVEC, M, N, NIN, NOUT
       CHARACTER*22 TITLE
C      .. LOCAL ARRAYS ..
       REAL A(10,6), B(10), SVEC(10,10), SVAL(6), WORK(28), X(10)
C      .. SUBROUTINE REFERENCES ..
C      SVDTJP, LSQMNS
C      ..
```

```
      DATA NIN /5/, NOUT /6/
      OPEN (UNIT=NIN,FILE='INPUT')
      OPEN (UNIT=NOUT,FILE='OUTPUT')

      READ  (NIN,'(A)') TITLE
      WRITE (NOUT,'(1X, A, '' RESULTS''/1X)') TITLE

      LDA    = 10
      IFAIL  = 0
      JOB    = 0
      M      = 6
      N      = 4
      LDSVEC = M + N

      READ  (NIN,'(6F6.2)') ((A(J,I),J=1,M),I=1,N)
      READ  (NIN,'(6F6.2)') (B(I),I=1,M)

      WRITE (NOUT,'('' MATRIX A''/,(1X,4F9.3))') ((A(I,J),J=1,N),I=1,M)
      WRITE (NOUT,'(''0VECTOR B''/,(1X,6F9.3))') (B(I),I=1,M)

      CALL SVDTJP(M,N,A,LDA,JOB,NRANK,SVAL,SVEC,LDSVEC,WORK,IFAIL)

      IF (IFAIL .EQ. 0) THEN
        WRITE (NOUT,'(''0MATRIX UR'')')
        DO 10 I=1,M
          WRITE (NOUT,'(1X,4F9.3)') (SVEC(I,J),J=1,NRANK)
   10   CONTINUE
        WRITE (NOUT,'(''0SINGULAR VALUES''/,(1X,4F9.3))')
     +        (SVAL(I),I=1,N)
        WRITE (NOUT,'(''0MATRIX V''/,(1X,4F9.3))')
     +        ((SVEC(I,J),J=1,N),I=M+1,M+N)

        CALL LSQMNS(M,N,NRANK,SVAL,SVEC,LDSVEC,B,X,IFAIL)

        IF (IFAIL .EQ. 0) THEN
          WRITE (NOUT,'(''0SOLUTION VECTOR''/,1X,4F9.3)')
     +          (X(I),I=1,N)
        END IF
      END IF

      END
```

13.2. Program data

```
LSQMNS EXAMPLE PROGRAM DATA
  0.05  0.25  0.35  1.75  0.30  0.40
  0.05  0.25  0.35  1.75 -0.30 -0.40
  0.25  0.05  1.75  0.35  0.30  0.40
 -0.25 -0.05 -1.75 -0.35  0.30  0.40
   1.0   2.0   3.0   4.0   5.0   6.0
```

13.3. Program results

```
LSQMNS EXAMPLE PROGRAM RESULTS

MATRIX A
     0.050     0.050     0.250    -0.250
     0.250     0.250     0.050    -0.050
     0.350     0.350     1.750    -1.750
     1.750     1.750     0.350    -0.350
     0.300    -0.300     0.300     0.300
     0.400    -0.400     0.400     0.400

VECTOR B
     1.000     2.000     3.000     4.000     5.000     6.000

MATRIX UR
    -0.100    -0.100     0.000
    -0.100     0.100     0.000
    -0.700    -0.700     0.000
    -0.700     0.700     0.000
     0.000     0.000     0.600
     0.000     0.000     0.800

SINGULAR VALUES
     3.000     2.000     1.000     0.000

MATRIX V
    -0.500     0.500     0.500    -0.500
    -0.500     0.500    -0.500     0.500
    -0.500    -0.500     0.500     0.500
     0.500     0.500     0.500     0.500

SOLUTION VECTOR
     4.967    -2.833     4.567     3.233
```