

Algorithm 32

Partial singular value decomposition algorithm *

Sabine VAN HUFFEL

ESAT Laboratory, Katholieke Universiteit Leuven, Kardinaal Mercierlaan 94, B-3001 Heverlee, Belgium

Received 7 November 1988

Revised 10 April 1990

0. Information

A copy of the Fortran 77 subroutine can be obtained by writing to the author.

ABSTRACT :

The Partial Singular Value Decomposition (PSVD) subroutine computes a basis of the left and/or right singular subspace of a matrix corresponding to its smallest singular values. The dimension of the desired subspace may be fixed or depend on a given upper bound for those smallest singular values.

As only a basis of the desired singular subspace is needed, the classical Singular Value Decomposition (SVD) algorithm is modified in three ways. First, the Householder transformations of the bidiagonalization are only applied to the base vectors of the desired singular subspace. Second, the bidiagonal is only partially diagonalized and third, the convergence rate of the iterative diagonalization is improved by an appropriate choice between QR and QL iteration steps. Depending on the gap, the desired numerical accuracy and the dimension of the desired subspace, PSVD can be three times faster than the classical SVD algorithm while the same accuracy can be maintained. The new algorithm can be successfully used in total least squares applications, in the computation of the null space of a matrix and in solving (non)homogeneous linear equations.

* This work was supported by the Belgian NFWO (National Fund of Scientific Research).

KEYWORDS : singular value decomposition, linear algebra,
singular value decomposition, least squares, (non)homogeneous
equations null space.

XX

SUBROUTINE : PSVD

1 PURPOSE:

The subroutine PSVD computes in an efficient and reliable way a basis for the left and/or right singular subspace of a matrix corresponding to its smallest singular values. The dimension of the desired subspace may be given or may depend on a given upper bound for those smallest singular values.

2 SPECIFICATION:

```
SUBROUTINE PSVD(A, LDA, M, N, RANK, THETA, U, LDU, V, LDV, Q, INUL,
                WRK, TOL1, TOL2, MODE, IERR, IWARN)
INTEGER LDA, M, N, RANK, LDU, LDV, MODE, IERR, IWARN
DOUBLE PRECISION THETA, TOL1, TOL2
DOUBLE PRECISION A(LDA,N), U(LDU,M), V(LDV,N), Q(min(M,N)+min(M+1,N)),
                WRK(*)
LOGICAL INUL(max(M,N))
```

3 ARGUMENT LIST:

3.1 ARGUMENTS IN

- A** - DOUBLE PRECISION array of DIMENSION (LDA,N).
The leading $M \times N$ part of this array contains the matrix A for which the basis of a desired singular subspace must be computed.
NOTE that A is destroyed by PSVD.
- LDA** - INTEGER.
LDA is the leading dimension of the array A ($LDA \geq M$).
- M** - INTEGER.
M is the number of rows of the matrix A.
- N** - INTEGER.
N is the number of columns of the matrix A.
- RANK** - INTEGER.
Specifies on entry whether the rank is given or is to be computed.
i) $RANK < 0$: the rank is to be computed (see THETA).
ii) $RANK \geq 0$: specifies the given rank.
NOTE that RANK may be overwritten if A has multiple singular values.

THETA - DOUBLE PRECISION.

On entry, there are two possibilities (depending on RANK):

- i) RANK < 0: THETA specifies an upper bound on the smallest singular values of A corresponding to the singular subspace to be computed. THETA >= 0. THETA allows to compute the rank of A.
- ii) RANK >= 0: THETA is an initial estimate for computing an upper bound on the $\min(M,N) - \text{RANK}$ smallest singular values of A.
If not available, assign a negative value (< 0) to THETA.
NOTE that THETA is overwritten.

3.2 ARGUMENTS OUT**RANK - INTEGER.**

If not specified by the user, then RANK is computed by the routine.

If specified by the user, then the specified RANK is changed by the routine if the RANK-th and the (RANK+1)-th singular value of A are considered to be equal.

THETA - DOUBLE PRECISION.

If RANK >= 0, then THETA specifies the computed upper bound such that precisely RANK singular values of A are greater than THETA > 0.01.

U - DOUBLE PRECISION array of DIMENSION (LDU,M).

The leading $M \times M$ part of this array (where $S = \min(M,N)$ or $S = N$, depending on MODE) contains the $S - \text{RANK}$ M-dimensional base vectors of the desired left singular subspace of A corresponding to its singular values $\leq \text{THETA}$.

These vectors are stored in those columns of U whose index i corresponds with $\text{INUL}(i) = \text{.TRUE.}$

If $10 \leq \text{MODE} < 20$, then $S = M$.

If $\text{MODE} \geq 20$, then $S = \min(M,N)$.

If $\text{MODE} \leq 9$, then U is not referenced.

U may not be identified with A in the subroutine call.

LDU - INTEGER.

LDU is the leading dimension of the array U ($\text{LDU} \geq M$).

V - DOUBLE PRECISION array of DIMENSION (LDV,N).

The leading $N \times S$ part of this array (where $S = \min(M,N)$ or $S = N$, depending on MODE) contains the $S - \text{RANK}$ N-dimensional base vectors of the desired right singular subspace of A corresponding to its singular values $\leq \text{THETA}$.

These vectors are stored in those columns of V whose index i corresponds with $\text{INUL}(i) = \text{.TRUE.}$

If $\text{MODE} \bmod 10 = 1$, then $S = N$, else $S = \min(M,N)$.

If $\text{MODE} \bmod 10 = 0$, then V is not referenced.

V may not be identified with A in the subroutine call.

LDV - INTEGER.

LDV is the leading dimension of the array *v* ($LDV \geq N$).

Q - DOUBLE PRECISION array of DIMENSION (min(M,N)+min(M+1,N)).
Returns the partially diagonalized bidiagonal computed from *A*, at the moment that the desired singular subspace has been found.
The first $p = \min(M,N)$ entries of *Q* contain the diagonal elements $q(1), \dots, q(p)$ and the entries $Q(p+2), \dots, Q(p+s)$ (with $s = \min(M+1,N)$) contain the superdiagonal elements $e(2), \dots, e(s)$. $Q(p+1) = 0$.

INUL - LOGICAL array of DIMENSION (max(M,N)).

The indices of the elements of *INUL* with value *.TRUE.* indicate the columns in *U* and/or *V* containing the base vectors of the desired left and/or right singular subspace of *A*, if computed. They also equal the indices of the diagonal entries of the subbidiagonals in *Q*, corresponding to the computed singular subspaces.

3.3 WORK SPACE

WRK - DOUBLE PRECISION array of DIMENSION (t).

If $M < 5 \times N/3$ or if no basis of a left singular subspace is requested (i.e. if *MODE* < 10),
then *t* equals $M + N$,
else *t* equals $M + N + N \times (N+1)/2$ in order to provide $N \times (N+1)/2$ extra storage locations.

3.4 TOLERANCES

TOL1 - DOUBLE PRECISION.

This parameter defines the multiplicity of singular values by considering all singular values within an interval of length *TOL1* as coinciding. *TOL1* is used in checking how many singular values are $\leq \text{THETA}$. Also in computing an appropriate upper bound *THETA* by a bisection method, *TOL1* is used as stop criterion defining the minimal subinterval length.
According to the numerical properties of the SVD, *TOL1* must be $\geq \|A\| \times \text{EPS}$ where $\|A\|$ denotes the L2-norm and *EPS* is the machine precision.

TOL2 - DOUBLE PRECISION.

Working precision for the computation of the desired singular subspaces of *A*. This parameter specifies that elements of matrices used in the computation, which are $\leq \text{TOL2}$ in absolute value, are considered to be zero.

3.5 MODE PARAMETER

MODE - INTEGER.

MODE controls the computation of the desired singular subspace. It has the decimal expansion AB with the following meaning:

- A = 0, do not compute the left singular subspace.
- A = 1, return the M - RANK base vectors of the desired left singular subspace in U.
- A >= 2, return the first min(M,N) - RANK base vectors of the desired left singular subspace in U.
- B = 0, do not compute the right singular subspace.
- B = 1, return the N - RANK base vectors of the desired right singular subspace in V.
- B >= 2, return the first min(M,N) - RANK base vectors of the desired right singular subspace in V.

3.6 ERROR INDICATORS

IERR - INTEGER.

On return, IERR contains 0 unless the routine has failed.

IWARN - INTEGER.

On return, IWARN contains 0 unless RANK has been lowered by the routine.

4 ERROR INDICATORS and WARNINGS:

Errors detected by the routine.

- IERR = 0:** successful completion.
- 1: number M of rows of array A smaller than 1.
 - 2: number N of columns of array A smaller than 1.
 - 3: leading dimension LDA of array A smaller than M.
 - 4: leading dimension LDU of array U smaller than M.
 - 5: leading dimension LDV of array V smaller than N.
 - 6: rank of matrix A (RANK) larger than min(M,N).
 - 7: the parameters RANK and THETA are both negative (< 0).
 - 8: tolerance TOL1 is negative.
 - 9: tolerance TOL2 is negative.
 - 10: maximum number of QR/QL iteration steps (50) exceeded.
 - 11: parameter MODE out of range.

Warnings given by the routine.

IWARN = 0: no warnings.

- 1: the rank of matrix A, specified by the user, has been lowered because a singular value of multiplicity > 1 has been found.

5 EXTERNAL SUBROUTINES and FUNCTIONS:

DCOPY from BLAS [5];
 DQRDC from LINPACK [6];
 BIDIAG, INIT, CANCEL, QRQL, RESTOR.

6 METHOD DESCRIPTION:

PSVD is an efficient method (see [1]), computing the singular subspace of a matrix corresponding to its smallest singular values. It differs from the classical SVD algorithm [3] at three points, which results in high efficiency. First, the Householder transformations of the bidiagonalization need only to be applied on the base vectors of the desired singular subspaces. Second, the bidiagonal needs only to be partially diagonalized. Third, the convergence rate of the iterative diagonalization can be improved by an appropriate choice between QL and QR iterations. Depending on the gap, the desired numerical accuracy and the dimension of the desired singular subspace, PSVD can be three times faster than the classical SVD algorithm.

The PSVD algorithm [1-2] for an M by N matrix A proceeds as follows:

Step 1: Bidiagonalization phase

1.a): If $M \geq 5N/3$, transform A into upper triangular form R.

1.b): Transform A (or R) into bidiagonal form:

$$J = \begin{pmatrix} q(1) & e(2) & 0 & \dots & 0 \\ 0 & q(2) & e(3) & & \\ ! & . & & & \\ ! & . & & & e(N)! \\ ! & 0 & & \dots & q(N)! \end{pmatrix}$$

if $M \geq N$, or

$$J = \begin{pmatrix} q(1) & e(2) & 0 & \dots & 0 & 0 \\ 0 & q(2) & e(3) & & . & . \\ ! & . & & & . & . \\ ! & . & & & e(M) & . \\ ! & 0 & & \dots & q(M) & e(M+1)! \end{pmatrix}$$

if $M < N$, using Householder transformations.

1.c): If U is requested, initialize U with the identity matrix.
 If V is requested, initialize V with the identity matrix.

1.d): If $M < N$, then cancel $e(M+1)$, and reduce the bidiagonal to $M \times M$. Accumulate the Givens rotations in V (if V is desired).

Step 2: Partial diagonalization phase

If the upper bound THETA is not given, then compute THETA such that precisely $p = \text{RANK}$ singular values ($p = \min(M, N)$) of the bidiagonal are $\leq \text{THETA}$, using a bisection method [4].

Diagonalize the given bidiagonal J partially, using either QL iterations (if the upper left diagonal element of the considered subbidiagonal $>$ the lower right diagonal element) or QR iterations, such that J is splitted into unreduced subbidiagonals whose singular values are either all larger than THETA or all less than or equal to THETA.

Accumulate the Givens rotations in U and/or V (if desired).

Step 3: Back transformation phase

3.a): Apply the Householder transformations of step 1.b) onto the columns of U and/or V associated with the subbidiagonals with all singular values $\leq \text{THETA}$, (if U and/or V is desired).

3.b): If $M \geq 5N/3$ and U is desired, then apply the Householder transformations of step 1.a) onto each computed column of U in step 3.a).

NOTE. If $M > N$ (resp., $M < N$), then the base vectors of the orthogonal complement of the column (resp., row) space of the M by N matrix A can also be computed if desired (see MODE) by applying step 3 onto the last $M - N$ (resp., $N - M$) columns of U (resp., V).

7 REFERENCES:

- [1] S. Van Huffel, J. Vandewalle and A. Haegemans, An efficient and reliable algorithm for computing the singular subspace of a matrix associated with its smallest singular values. *J. Comput. and Applied Math.*, 19 (1987), 313 - 330.
- [2] S. Van Huffel, Analysis of the total least squares problem and its use in parameter estimation. Doctoral dissertation, Dept. of Electr. Eng., K.U.Leuven, June 1987.
- [3] T.F.Chan, An improved algorithm for computing the singular value decomposition. *ACM Trans. Math. Software*, 8 (1982), 72-93.
- [4] S. Van Huffel and J. Vandewalle, The Partial Total Least Squares Algorithm. *J. Comput. and Applied Math.*, 21 (1988), 333 - 341.
- [5] C.L. Lawson, R.J. Hanson, F.T. Krogh and O.R. Kincaid, Basic Linear Algebra Subprograms for FORTRAN Usage. *ACM Trans. Math. Software*, 5 (1979), 308-323.
- [6] J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, LINPACK User's Guide. SIAM, Philadelphia (1979).

8 NUMERICAL ASPECTS:

Using PSVD a large reduction in computation time can be gained in total least squares applications (cf [2 - 4]), in the computation of

the null space of a matrix and in solving (non)homogeneous linear equations.

9 EXAMPLE:

Consider the 6×4 matrix A :

A =

0.80010002D+00	0.39985167D+00	0.60005390D+00	0.89999446D+00
0.29996484D+00	0.69990689D+00	0.39997269D+00	0.82997570D+00
0.49994235D+00	0.60003167D+00	0.20012361D+00	0.79011189D+00
0.90013643D+00	0.20016919D+00	0.79995025D+00	0.85002662D+00
0.39998539D+00	0.80006338D+00	0.49985474D+00	0.99016399D+00
0.20002274D+00	0.90007114D+00	0.70009777D+00	0.10299439D+01

and set TOL1 = 1.0D-08, TOL2 = 1.0D-10, MODE = 11, RANK = -1 and
THETA = 1.0D-03.

Then, A has computed rank 3 and PSVD computes the following bases for the singular subspaces of A associated with its singular values $\leq 1.00 \cdot 10^{-3}$:

Basis of the computed LEFT singular subspace :

the 1-st base vector $U(x, 4) =$

0.269797D+00	0.153118D+00	-0.536944D+00	-0.186820D+00
0.642075D+00	-0.410236D+00		

the 2-th base vector $U(x, 5) =$

-0.578307D+00	-0.456351D+00	0.180389D+00	0.336878D+00
6.552879D+00	-0.748493D-01		

the 3-th base vector $U(x, 6) =$

```
0.484175D+00  -0.742503D+00   0.646079D-01  -0.334913D+00
0.115913D+00   0.290665D+00
```

Basis of the computed RIGHT singular subspace :

the 1-st base vector $V(x, 4) =$

-0.355483D+00 -0.568663D+00 -0.212821D+00 0.710606D+00

XX