ALGORITHM 581
An Improved Algorithm for Computing the Singular Value Decomposition [F1]

TONY F. CHAN
Department of Computer Science, Yale University

Categories and Subject Descriptors D.3.2 **[Programming Languages]** Language Classifications—*FORTRAN*, G 1.3 **[Numerical Analysis]** Numerical Linear Algebra—*pseudoinverses*, G 1 6 **[Numerical Analysis]**. Optimizations—*least squares methods*

General Terms: Algorithms

Additional Key Words and Phrases singular value decomposition

DESCRIPTION

The set of FORTRAN subroutines given here is an implementation of the algorithm [1] for computing the Singular Value Decomposition (SVD) of a general m by n rectangular matrix A defined as

$$A = UWV^{\mathrm{T}},$$

where U is an $m \times \min(m,n)$ matrix containing the left singular vectors, W is a diagonal matrix of size $\min(m, n)$ containing the singular values, and V is an $n \times \min(m, n)$ matrix containing the right singular vectors. Note that m is allowed to be greater than or less than n. For ease of presentation, we assume m to be greater than or equal to n in the following discussion.

The algorithm is an improvement of the Golub–Reinsch algorithm [4], which is implemented in subroutines SVD and MINFIT in EISPACK [3] and in subroutine SSVDC in LINPACK [2]. It should be more efficient than the Golub–Reinsch algorithm when m is approximately larger than $2n$, as is the case in many least squares applications.

The algorithm has a hybrid nature. When m is about equal to n, the Golub–Reinsch algorithm is employed. When the ratio m/n is larger than a threshold value, which is determined by detailed operation counts [1], the improved algorithm is used.

The improved algorithm first computes the QR factorization of A using Householder transformations, and then uses the Golub–Reinsch algorithm on R. A further improvement over the Golub–Reinsch algorithm is when the left singular

Received 7 October 1976; revised 25 January 1982, accepted 29 January 1982
This work was supported by NSF Grant DCR 75-13497 and NASA Ames Contract NCA 2-OR745-520. The computing time was provided by the Stanford Linear Accelerator Center (SLAC)
Author's address: Department of Computer Science, Yale University, 10 Hillhouse Avenue, P O. Box 2158 Yale Station, New Haven CT 06520.

vectors are to be accumulated and saved. Here, instead of accumulating the Givens transformations (in the second phase of the algorithm where the singular values of the bidiagonal matrix are computed) on the $m \times n$ matrix containing the left singular vectors, we accumulate them on a temporary $n \times n$ matrix. This requires a small overhead in storage of an $n \times n$ matrix (small compared with $m \times n$) but offers big savings in time.

An additional feature of the new algorithm is that it can accumulate all the left orthogonal transformations on a number of given vectors, which can then be used in computing least squares solutions. In this fashion, it is similar to the EISPACK routine MINFIT.

There are three main routines in the package:

HYBSVD: This is the main routine which implements the hybrid algorithm.
MGNSVD: This performs the same thing as HYBSVD, except that it assumes m .ge. n.
GRSVD: This is a slightly modified version of routine SVD in EISPACK which implements the Golub-Reinsch algorithm.

Besides, there are two utility routines:

SSWAP: BLAS routine for swapping two vectors.
SRELPR: Routine for computing the machine relative precision.

These five routines must be used together. They have been tested extensively on the IBM 370/168, 360/91 at the Stanford Linear Accelerator Center, and on the DEC 2060 in the Computer Science Department at Yale. They produce results that agree (up to machine precision) with those produced by SVD, MINFIT, and SSVDC. They have been verified by PFORT verifier [5] for portability.

REFERENCES

1. CHAN, T.F. An improved algorithm for computing the Singular Value Decomposition *ACM Trans, Math. Softw. 8*, 1 (Mar. 1982), 72–83.
2. DONGARRA, J J., BUNCH, J.R., MOLER, C.B., AND STEWART, G.W. *LINPACK User's Guide.* SIAM, Philadelphia, 1979.
3. GARBOW, B.S., ET AL. *Matrix Eigensystem Routines—EISPACK Guide Extension*, Lecture Notes in Computer Science Series, No. 51. Springer-Verlag, New York, 1977.
4. GOLUB, G.H., AND REINSCH, C Singular Value Decomposition and least squares solutions. In *Handbook for Automatic Computation*, II, *Linear Algebra*, J.H. Wilkinson and C. Reinsch (Eds.), Springer-Verlag, New York, 1971.
5. RYDER, B.G. The PFORT verifier In *Softw. Pract. Exper. 4* (1974) 359–377.

ALGORITHM

[A part of the listing is printed here. The complete listing is available from the ACM Algorithms Distribution Service (see page 91 for order form).]

```
      SUBROUTINE HYBSVD(NA, NU, NV, NZ, NB, M, N, A, W, MATU, U, MATV,
     * V, Z, B, IRHS, IERR, RV1)
      INTEGER NA, NU, NV, NZ, M, N, IRHS, IERR, MIN0
      REAL A(NA,1), W(1), U(NU,1), V(NV,1), Z(NZ,1), B(NB,IRHS), RV1(1)
      LOGICAL MATU, MATV
C
C     THIS ROUTINE IS A MODIFICATION OF THE GOLUB-REINSCH PROCEDURE (1)
C                                                 T
C     FOR COMPUTING THE SINGULAR VALUE DECOMPOSITION A = UWV  OF A
C     REAL M BY N RECTANGULAR MATRIX. U IS M BY MIN(M,N) CONTAINING
C     THE LEFT SINGULAR VECTORS, W IS A MIN(M,N) BY MIN(M,N) DIAGONAL
C     MATRIX CONTAINING THE SINGULAR VALUES, AND V IS N BY MIN(M,N)
C     CONTAINING THE RIGHT SINGULAR VECTORS.
C
C     THE ALGORITHM IMPLEMENTED IN THIS
C     ROUTINE HAS A HYBRID NATURE.  WHEN M IS APPROXIMATELY EQUAL TO N,
C     THE GOLUB-REINSCH ALGORITHM IS USED, BUT WHEN EITHER OF THE RATIOS
C     M/N OR N/M IS GREATER THAN ABOUT 2,
C     A MODIFIED VERSION OF THE GOLUB-REINSCH
C     ALGORITHM IS USED.  THIS MODIFIED ALGORITHM FIRST TRANSFORMS A
C                                                             T
C     INTO UPPER TRIANGULAR FORM BY HOUSEHOLDER TRANSFORMATIONS L
C     AND THEN USES THE GOLUB-REINSCH ALGORITHM TO FIND THE SINGULAR
C     VALUE DECOMPOSITION OF THE RESULTING UPPER TRIANGULAR MATRIX R.
C     WHEN U IS NEEDED EXPLICITLY IN THE CASE M.GE.N (OR V IN THE CASE
C     M.LT.N), AN EXTRA ARRAY Z (OF SIZE AT LEAST
C     MIN(M,N)**2) IS NEEDED, BUT OTHERWISE Z IS NOT REFERENCED
C     AND NO EXTRA STORAGE IS REQUIRED.  THIS HYBRID METHOD
C     SHOULD BE MORE EFFICIENT THAN THE GOLUB-REINSCH ALGORITHM WHEN
C     M/N OR N/M IS LARGE.  FOR DETAILS, SEE (2).
C
C     WHEN M .GE. N,
C     HYBSVD CAN ALSO BE USED TO COMPUTE THE MINIMAL LENGTH LEAST
C     SQUARES SOLUTION TO THE OVERDETERMINED LINEAR SYSTEM A*X=B.
C     IF M .LT. N (I.E. FOR UNDERDETERMINED SYSTEMS), THE RHS B
C     IS NOT PROCESSED.
C
C     NOTICE THAT THE SINGULAR VALUE DECOMPOSITION OF A MATRIX
C     IS UNIQUE ONLY UP TO THE SIGN OF THE CORRESPONDING COLUMNS
C     OF U AND V.
C
C     THIS ROUTINE HAS BEEN CHECKED BY THE PFORT VERIFIER (3) FOR
C     ADHERENCE TO A LARGE, CAREFULLY DEFINED, PORTABLE SUBSET OF
C     AMERICAN NATIONAL STANDARD FORTRAN CALLED PFORT.
C
C     REFERENCES:
C
C     (1) GOLUB,G.H. AND REINSCH,C. (1970) 'SINGULAR VALUE
C         DECOMPOSITION AND LEAST SQUARES SOLUTIONS,'
C         NUMER. MATH. 14,403-420, 1970.
C
C     (2) CHAN,T.F. (1982) 'AN IMPROVED ALGORITHM FOR COMPUTING
C         THE SINGULAR VALUE DECOMPOSITION,' ACM TOMS, VOL.8,
C         NO. 1, MARCH, 1982.
C
C     (3) RYDER,B.G. (1974) 'THE PFORT VERIFIER,' SOFTWARE -
C         PRACTICE AND EXPERIENCE, VOL.4, 359-377, 1974.
C
C     ON INPUT:
C
```

```
C        NA MUST BE SET TO THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C          ARRAY PARAMETER A AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT.  NOTE THAT NA MUST BE AT LEAST
C          AS LARGE AS M.
C
C        NU MUST BE SET TO THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C          ARRAY U AS DECLARED IN THE CALLING PROGRAM DIMENSION
C          STATEMENT. NU MUST BE AT LEAST AS LARGE AS M.
C
C        NV MUST BE SET TO THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C          ARRAY PARAMETER V AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT. NV MUST BE AT LEAST AS LARGE AS N.
C
C        NZ MUST BE SET TO THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C          ARRAY PARAMETER Z AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT.  NOTE THAT NZ MUST BE AT LEAST
C          AS LARGE AS MIN(M,N).
C
C        NB MUST BE SET TO THE ROW DIMENSION OF THE TWO-DIMENSIONAL
C          ARRAY PARAMETER B AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT. NB MUST BE AT LEAST AS LARGE AS M.
C
C        M IS THE NUMBER OF ROWS OF A (AND U).
C
C        N IS THE NUMBER OF COLUMNS OF A (AND NUMBER OF ROWS OF V).
C
C        A CONTAINS THE RECTANGULAR INPUT MATRIX TO BE DECOMPOSED.
C
C        B CONTAINS THE IRHS RIGHT-HAND-SIDES OF THE OVERDETERMINED
C         LINEAR SYSTEM A*X=B. IF IRHS .GT. 0 AND M .GE. N,
C         THEN ON OUTPUT, THE FIRST N COMPONENTS OF THESE IRHS COLUMNS
C                      T
C         WILL CONTAIN U B. THUS, TO COMPUTE THE MINIMAL LENGTH LEAST
C                                             +
C         SQUARES SOLUTION, ONE MUST COMPUTE V*W  TIMES THE COLUMNS OF
C                 +                       +
C         B, WHERE W  IS A DIAGONAL MATRIX, W (I)=0 IF W(I) IS
C         NEGLIGIBLE, OTHERWISE IS 1/W(I). IF IRHS=0 OR M.LT.N,
C         B IS NOT REFERENCED.
C
C        IRHS IS THE NUMBER OF RIGHT-HAND-SIDES OF THE OVERDETERMINED
C         SYSTEM A*X=B. IRHS SHOULD BE SET TO ZERO IF ONLY THE SINGULAR
C         VALUE DECOMPOSITION OF A IS DESIRED.
C
C        MATU SHOULD BE SET TO .TRUE. IF THE U MATRIX IN THE
C          DECOMPOSITION IS DESIRED, AND TO .FALSE. OTHERWISE.
C
C        MATV SHOULD BE SET TO .TRUE. IF THE V MATRIX IN THE
C          DECOMPOSITION IS DESIRED, AND TO .FALSE. OTHERWISE.
C
C        WHEN HYBSVD IS USED TO COMPUTE THE MINIMAL LENGTH LEAST
C        SQUARES SOLUTION TO AN OVERDETERMINED SYSTEM, MATU SHOULD
C        BE SET TO .FALSE. , AND MATV SHOULD BE SET TO .TRUE.
C
C     ON OUTPUT:
C
C        A IS UNALTERED (UNLESS OVERWRITTEN BY U OR V).
C
C        W CONTAINS THE (NON-NEGATIVE) SINGULAR VALUES OF A (THE
C          DIAGONAL ELEMENTS OF W).  THEY ARE SORTED IN DESCENDING
```

```
C          ORDER.  IF AN ERROR EXIT IS MADE, THE SINGULAR VALUES
C          SHOULD BE CORRECT AND SORTED FOR INDICES IERR+1,...,MIN(M,N).
C
C        U CONTAINS THE MATRIX U (ORTHOGONAL COLUMN VECTORS) OF THE
C          DECOMPOSITION IF MATU HAS BEEN SET TO .TRUE.  IF MATU IS
C          FALSE, THEN U IS EITHER USED AS A TEMPORARY STORAGE (IF
C          M .GE. N) OR NOT REFERENCED (IF M .LT. N).
C          U MAY COINCIDE WITH A IN THE CALLING SEQUENCE.
C          IF AN ERROR EXIT IS MADE, THE COLUMNS OF U CORRESPONDING
C          TO INDICES OF CORRECT SINGULAR VALUES SHOULD BE CORRECT.
C
C        V CONTAINS THE MATRIX V (ORTHOGONAL) OF THE DECOMPOSITION IF
C          MATV HAS BEEN SET TO .TRUE.  IF MATV IS
C          FALSE, THEN V IS EITHER USED AS A TEMPORARY STORAGE (IF
C          M .LT. N) OR NOT REFERENCED (IF M .GE. N).
C          IF M .GE. N, V MAY ALSO COINCIDE WITH A.  IF AN ERROR
C          EXIT IS MADE, THE COLUMNS OF V CORRESPONDING TO INDICES OF
C          CORRECT SINGULAR VALUES SHOULD BE CORRECT.
C
C        Z CONTAINS THE MATRIX X IN THE SINGULAR VALUE DECOMPOSITION
C                 T
C          OF R=XSY,  IF THE MODIFIED ALGORITHM IS USED. IF THE
C          GOLUB-REINSCH PROCEDURE IS USED, THEN IT IS NOT REFERENCED.
C          IF MATU HAS BEEN SET TO .FALSE. IN THE CASE M.GE.N (OR
C          MATV SET TO .FALSE. IN THE CASE M.LT.N), THEN Z IS NOT
C          REFERENCED AND NO EXTRA STORAGE IS REQUIRED.
C
C        IERR IS SET TO
C          ZERO        FOR NORMAL RETURN,
C          K           IF THE K-TH SINGULAR VALUE HAS NOT BEEN
C                      DETERMINED AFTER 30 ITERATIONS.
C          -1          IF IRHS .LT. 0 .
C          -2          IF M .LT. 1 .OR. N .LT. 1
C          -3          IF NA .LT. M .OR. NU .LT. M .OR. NB .LT. M.
C          -4          IF NV .LT. N .
C          -5          IF NZ .LT. MIN(M,N).
C
C        RV1 IS A TEMPORARY STORAGE ARRAY OF LENGTH AT LEAST MIN(M,N).
C
C     PROGRAMMED BY : TONY CHAN
C                     BOX 2158, YALE STATION,
C                     COMPUTER SCIENCE DEPT, YALE UNIV.,
C                     NEW HAVEN, CT 06520.
C     LAST MODIFIED : JANUARY, 1982.
C
C     HYBSVD USES THE FOLLOWING FUNCTIONS AND SUBROUTINES.
C       INTERNAL  GRSVD, MGNSVD, SRELPR
C       FORTRAN   MIN0,ABS,SQRT,FLOAT,SIGN,AMAX1
C       BLAS      SSWAP
C
C     ----------------------------------------------------------------
```