

4.3 Singular Value Decomposition and Analysis

A. Purpose

Any $M \times N$ matrix, A , has a Singular Value Decomposition (SVD) of the form

$$A = USV^t$$

where U is an $M \times M$ orthogonal matrix, V is an $N \times N$ orthogonal matrix, and S is an $M \times N$ matrix having non-negative elements on the diagonal and zeros elsewhere. It is customary to arrange that the diagonal elements of S , called the singular values of A , are in decreasing order. This is done in the software described here.

The SVD can be useful in analyzing a linear least-squares problem or other matrix problems, particularly when there is reason to believe that the model is ill-conditioned. The SVD also has a role as a component in many specialized algorithms of linear algebra and statistics.

Here we describe three subroutines to facilitate use of the SVD:

(1) Let a matrix, A , and a matrix or vector, B , be given. Denote the SVD of A by $A = USV^t$. Subroutine SSVDRS computes the SVD of a matrix, A , and returns S , V , and the product, U^tB . If one needs to obtain the matrix U explicitly, call SSVDRS with B set to be the M^{th} order identity matrix.

(2) Subroutine SSVA uses SSVDRS and produces a report of quantities useful for the singular value analysis of a least-squares problem, $Ax \simeq b$.

(3) Subroutine SCOV3 can be used following SSVDRS to compute a covariance matrix.

B. Usage

B.1 Singular Value Decomposition with Computation of U^tB

Compute the Singular Value Decomposition, USV^t , of a matrix, A , and optionally return the product, U^tB , where B is another given matrix.

B.1.a Program Prototype, Single Precision

INTEGER LDA, M, N, LDB, NB

REAL A(LDA, $\geq N$), SING($\geq N$), D($\geq N$),
WORK($\geq 2 \times N$), B(LDB, $\geq NB$) or B($\geq M$)

Assign values to A(.), LDA, M, N, B(), LDB, and NB.

```
CALL SSVDRS (A, LDA, M, N, B,  
            LDB, NB, SING, WORK)
```

© 1997 Calif. Inst. of Technology, 2015 Math à la Carte, Inc.

Computed quantities are returned in A(.), B(), and SING().

B.1.b Argument Definitions

A(.) [inout] On entry, contains an $M \times N$ matrix, A . On return contains the $N \times N$ matrix, V , such that $A = USV^t$.

LDA [in] First dimensioning parameter for A(.). Require $LDA \geq \max(M, N)$.

M,N [in] Number of rows and columns, respectively, in the given matrix, A . Either $M > N$ or $M \leq N$ is permitted. Require $M > 0$ and $N > 0$.

B() [inout] On entry, contains an M -vector, b , or an $M \times NB$ matrix, B . On return, contains the M -vector, $g = U^tb$, or the $M \times NB$ matrix, $G = U^tB$, where U satisfies $USV^t = A$. Note that if B is the M^{th} order identity matrix on entry, then on return the array B(.) will contain U^t .

LDB [in] First dimensioning parameter for the array B(.). Require $LDB \geq \max(M, N)$ when $NB \geq 1$ and $LDB \geq 1$ when $NB = 0$.

NB [in] Number of columns in the input matrix, B . Require $NB \geq 0$. If $NB = 1$, the array B() may be singly or doubly subscripted. If $NB > 1$, the array B(.) must be doubly subscripted. If $NB = 0$, the array B() will not be referenced.

SING() [out] On return, contains the singular values of A , in descending order, in locations indexed 1 through N . If $M < N$, SING($M+1$) through SING(N) will be set to zero.

WORK() [scratch] Work space of length at least $2 \times N$.

B.2 Singular Value Analysis

Computes quantities useful for the singular value analysis of a least-squares problem, $Ax \simeq b$. Optionally produces a report, with options to select the full report or only parts of it, to select the output unit for the report, and to specify the display width available for the report.

B.2.a Program Prototype, Single Precision

INTEGER LDA, M, N, MDATA, ISCALE,
KPVEC(4)

REAL A(LDA, $\geq N$), B($\geq M$), SING($\geq N$), D($\geq N$),
WORK($\geq 2 \times N$)

CHARACTER NAMES($\geq N$) * (lennam)
[lennam ≥ 1]

Assign values to A(,), LDA, M, N, MDATA, B(,), KPVEC(,), NAMES(,), ISCALE, and optionally to D().

**CALL SSSVA (A, LDA, M, N, MDATA,
B, SING, KPVEC, NAMES,
ISCALE, D, WORK)**

Computed quantities are returned in A(,), B(,), SING(,), and optionally in D(). A report is produced if selected by KPVEC(,).

B.2.b Argument Definitions

A(,) [inout] On entry, contains the $M \times N$ matrix, A , of the least-squares problem to be analyzed. This could be a matrix obtained by preliminary orthogonal transformations applied to the actual problem matrix which may have had more rows (See MDATA below.) On return, contains an N^{th} order matrix in which the j^{th} column is the j^{th} candidate solution for the least-squares problem, *i.e.* the solution computed as though singular values $j+1$ through N were zero.

LDA [in] First dimensioning parameter for A(,). Require $LDA \geq \max(M, N)$.

M,N [in] Number of rows and columns, respectively, in the matrix, A . Either $M > N$ or $M \leq N$ is permitted. Require $M > 0$ and $N > 0$.

MDATA [in] Number of rows in actual least-squares problem. Generally $MDATA \geq M$. MDATA is used only in computing statistics for the report and is not used as a loop count or array dimension.

B(,) [inout] On entry, contains the right-side M -vector, \mathbf{b} , of the least-squares problem. On return, contains the M -vector, $\mathbf{g} = U^t \mathbf{b}$, where U comes from the singular value decomposition of A .

SING(,) [out] On return, contains the singular values of A , in descending order, in locations indexed 1 through N . If $M < N$, $SING(M+1)$ through $SING(N)$ will be set to zero.

KPVEC(,) [in] Option array controlling report generation. If $KPVEC(1) = 0$, default settings will be used, producing the full report, sending it to the standard system output unit, formatted with a maximum line length of 79. If $KPVEC(1) = 1$, the contents of $KPVEC(I)$, $I = 2, \dots, 4$, set options for the report, as follows:

KPVEC(2) The decimal representation of $KPVEC(2)$ must be at most 6 digits, each being 0 or 1. The decimal digits will be interpreted as independent on/off flags for the 6 possible

blocks of the report. Examples: 101010 selects the 1st, 3rd, and 5th blocks, 111111 selects all blocks, 0 suppresses the whole report, etc. The default value is 111111. The six blocks are:

1. Header, with size and scaling option parameters.
2. V -matrix.
3. Singular values and related quantities.
4. Listing of $YNORM$ and $RNORM$ and their logarithms.
5. Levenberg-Marquardt analysis.
6. Candidate solutions.

KPVEC(3) Define $UNIT = KPVEC(3)$. If $UNIT \geq 0$, $UNIT$ will be used as the output unit number. If $UNIT = -1$, output will be written to the "*" output unit, *i.e.*, the standard system output unit. The default value is -1 . The calling program unit is responsible for opening and/or closing the selected output unit if the host system requires these actions.

KPVEC(4) Determines the width of blocks 2, 3, and 6 of the output report. Define $WIDTH = KPVEC(4)$. The default value is 79. Each output line will have a leading blank for Fortran "carriage control" with line widths as follows: Output blocks 1, 4, and 5 always have 63, 66, and 66 character positions respectively. Output blocks 2 and 6 will generally have at most $WIDTH$ character positions. One output line will contain a row number, a name from $NAMES(,)$, and from one to eight floating point numbers. The space allocated for a name will be that needed for the longest name in $NAMES(,)$, which may be less than the declared length of elements of $NAMES(,)$. The line length will only exceed $WIDTH$ if this is necessary to accommodate the row number and name plus one floating-point number. Output block 3 will have 69 character positions if $WIDTH < 95$ and otherwise will have 95 character positions.

NAMES(,) [in] $NAMES(j)$, for $j = 1, \dots, N$, may contain a name for the j^{th} component of the solution vector. If $NAMES(1)$ contains only blank characters, it will be assumed that no names have been provided, and this subroutine will not access the $NAMES(,)$ array beyond the first element.

ISCALE [in] Set by the user to 1, 2, or 3 to select the column scaling option.

- 1 The subroutine will use identity scaling and ignore the D(,) array.

- 2 The subroutine will scale nonzero columns of A to have unit Euclidean length, and will store reciprocal lengths of the original nonzero columns in $D()$.
- 3 User supplies column scaling factors in $D()$. The subroutine will multiply column j by $D(j)$, and remove the scaling from the solution at the end.

D() [ignored or out or in] Usage of $D()$ depends on $ISCALE$ as described above. When used, its length must be at least N .

WORK() [scratch] Scratch work space.

B.3 Computation of the Covariance Matrix

Subroutine, $SCOV3$, can be used to compute the covariance matrix for the solution vector of a least-squares problem, $Ax \simeq \mathbf{b}$, following use of $SSVDRS$ in cases in which $M > N$ and all N singular values of A are nonzero.

B.3.a Program Prototype, Single Precision

INTEGER LDA, N, IERR
REAL A(LDA, $\geq N$), SING($\geq N$), VAR,
WORK($\geq N$)

On entry, the arguments, $A()$, LDA, N, and $SING()$ should contain the same values as on return from a previous call to $SSVDRS$.

**CALL SCOV3(A, LDA, N, SING,
 VAR, WORK, IERR)**

Computed quantities are returned in $A()$ and IERR.

B.3.b Argument Definitions

A() [inout] On entry, contains the $N \times N$ matrix, V , computed by a previous call to $SSVDRS$. (Note that the quantities left in the array $A()$ by $SSVA$ are not appropriate for use as input to $SCOV3$.) On return contains the $N \times N$ symmetric covariance matrix, C , for the solution vector of the least-squares problem, $Ax \simeq \mathbf{b}$.

LDA [in] First dimensioning parameter for $A()$. Must be the same value used when $SSVDRS$ was called to compute the V matrix.

N [in] Number of rows and columns of the matrix, V , contained in the array, $A()$. Must be the same value used when $SSVDRS$ was called to compute the V matrix.

SING() [in] Contains the singular values of A in locations indexed 1 through N . Must all be nonzero.

VAR [in] User-supplied estimate of the variance of error in the vector, \mathbf{b} , of the least-squares problem.

WORK() [scratch] Work space of length at least N .

IERR [out] Set to 0 if all of the N given singular values are nonzero. Otherwise IERR will be set to the index of the first zero element of the array, $SING()$. In this latter case the covariance matrix cannot be computed and the contents of $A()$ on return will be meaningless.

B.4 Modifications for Double Precision

For double precision usage change all **REAL** type statements to **DOUBLE PRECISION**, and change the subroutine names from $SSVA$, $SSVDRS$, and $SCOV3$ to $DSVA$, $DSVDRS$, and $DCOV3$, respectively.

C. Examples and Remarks

The program, $DRDSVA$, illustrates the use of $DSVA$ to compute and report quantities for the singular value analysis of a 15×5 least-squares problem. This is an artificially constructed problem that was used as an example in [1]. The output of this example is shown in $ODDSVA$.

C.1 Large problems

If $M \gg N$ and storage limitations make it awkward or impossible to allocate $M \times N$ locations for the array, $A()$, one can use sequential accumulation of the rows of data to produce a smaller matrix to which $SSVDRS$ or $SSVA$ can then be applied. When the ratio M/N is sufficiently large this approach will be faster than direct application of $SSVDRS$ to the original matrix. See Chapter 4.4 for sequential accumulation.

C.2 The pseudoinverse of A

The pseudoinverse of A , conventionally denoted by A^+ , can be computed using results produced by $SSVDRS$. Set $B = I$, the M^{th} order identity matrix. Then call $SSVDRS$ obtaining S , V , and U^t . Since S is diagonal, its pseudoinverse, S^+ , is its transpose with nonzero elements replaced by their reciprocals. Depending on the application, it may be appropriate to treat nonzero singular values that are smaller than some problem-related threshold as though they were zero. After defining S^+ , one can compute $A^+ = VS^+U^t$.

C.3 Solution of the least-squares problem, $Ax \simeq \mathbf{b}$

From A and \mathbf{b} , $SSVDRS$ can compute V , S , and $\mathbf{g} = U^t\mathbf{b}$. The user can then determine S^+ from S as described in the preceding paragraph and compute the solution vector as $\mathbf{x} = VS^+\mathbf{g}$.

D. Functional Description

D.1 Method in SSVDRS

Subroutine SSVDRS computes the singular value decomposition by an algorithm due to G. Golub and W. Kahan, as described in [1]. This method uses approximately $2N$ Householder orthogonal transformations to reduce the given A matrix to a bidiagonal matrix and then calls SQRBD to apply a specialized version of the QR eigenvalue algorithm to the bidiagonal matrix to complete the reduction to the diagonal S matrix. The QR algorithm generally requires about $2N$ iterations to reach convergence to machine accuracy. The product of the orthogonal transformations is accumulated to produce the V matrix and the product, $G = U^t B$.

Results are permuted so the singular values are in decreasing order. The largest singular value will be accurate to nearly the machine accuracy. Other singular values will have about the same absolute accuracy as the largest one. The matrices V and U will be orthogonal to nearly machine accuracy.

This implementation gives special treatment to any column of A that is exactly zero. If $M > N$ each such column will give rise to an exactly zero singular value in the SING() array. This is convenient if one wishes to remove a variable from a problem by just zeroing its column in A .

D.2 Method in SSVA

When the matrix of a least-squares problem is ill-conditioned it will frequently be true that the vector \mathbf{x} that minimizes the residual norm $\|A\mathbf{x} - \mathbf{b}\|$ is undesirably large, *i.e.*, $\|\mathbf{x}\|$ is large. It is also generally true that there will be other vectors, \mathbf{x} , that are significantly smaller in norm than the exact solution vector, with only a small increase in the residual norm. One of these vectors may be preferable to the true solution as an operational solution for the problem.

There are various ways to define either a discrete or a continuous family of candidate solutions for a least-squares problem that range from the true solution through \mathbf{x} 's having smaller norms but allowing larger residual norms. This subroutine gives information on two such families of candidate solutions.

To simplify the discussion we assume $M > N$; however the subroutine also handles $M \leq N$. A useful discrete family of candidate solutions is obtained by defining $\mathbf{x}^{(i)}$ to be the solution obtained when all singular values following the i^{th} are set to zero. Then $\mathbf{x}^{(N)}$ is the true solution, $\mathbf{x}^{(0)}$ is the zero vector, and the intermediate candidate solution vectors satisfy the monotonicity conditions, $\|\mathbf{x}^{(i)}\| \leq \|\mathbf{x}^{(i+1)}\|$ and $\|A\mathbf{x}^{(i)} - \mathbf{b}\| \geq \|A\mathbf{x}^{(i+1)} - \mathbf{b}\|$.

A useful continuous family of candidate solutions is defined by letting \mathbf{x}^λ , for $\lambda \geq 0$, be the vector that minimizes $\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda^2\|\mathbf{x}\|^2$. This is equivalent to asking for the solution of the augmented least-squares problem:

$$\begin{bmatrix} A \\ \lambda I \end{bmatrix} \mathbf{x}^\lambda \simeq \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

where I denotes the N^{th} order identity matrix. If $\lambda < \mu$ then $\|\mathbf{x}^\lambda\| \geq \|\mathbf{x}^\mu\|$ and $\|A\mathbf{x}^\lambda - \mathbf{b}\| \leq \|A\mathbf{x}^\mu - \mathbf{b}\|$. This family of candidate solutions is discussed in the literature under various names, such as ridge regression, damped least-squares, and Levenberg-Marquardt stabilization.

Since the units in which variables are expressed is arbitrary and the value of $\|\mathbf{x}\|$, and thus the family of candidate solutions, depends on the choice of units, this subroutine gives the user the ability, via the parameters ISCALE and D(), to scale the columns of A and thus the components of \mathbf{x} .

Given A and \mathbf{b} defining a least-squares problem, $A\mathbf{x} \simeq \mathbf{b}$, SSVA performs the following steps, doing the indicated printing only if selected by the settings of KPVEC().

- 1a. As specified by the user's setting of ISCALE and D(), introduce a nonsingular diagonal scaling matrix, D , reformulating the problem as $(AD)(D^{-1}\mathbf{x}) \simeq \mathbf{b}$. Let $\mathbf{y} = D^{-1}\mathbf{x}$ so the problem can be written as $(AD)\mathbf{y} \simeq \mathbf{b}$.
- 1b. Use SSVDRS to compute the singular value decomposition of AD , obtaining U , S , and V satisfying $AD = USV^t$, and $\mathbf{g} = U^t\mathbf{b}$. Let s_i denote the i^{th} diagonal element of S , *i.e.* the i^{th} singular value of AD . The scaled solution vector, \mathbf{y} , is given by $\mathbf{y} = VS^+U^t\mathbf{b}$, and thus may be computed in two steps as $\mathbf{p} = S^+\mathbf{g}$ and $\mathbf{y} = V\mathbf{p}$. Specifically the components of the N -vector \mathbf{p} are computed as $p_i = g_i/s_i$ if $s_i \neq 0$ and $p_i = 0$ if $s_i = 0$.
2. Print the V matrix.
3. For $i = 1, \dots, N$, print $s_i, p_i, 1/s_i, g_i$, and g_i^2 . Define $\rho_j^2 = \|A\mathbf{x}^{(j)} - \mathbf{b}\|^2$. This quantity is computed as

$$\rho_j^2 = \sum_{i=j+1}^N g_i^2$$

and is printed with the heading "Cumulative Sum of Squares."

An estimate of the variance of the errors in the data vector \mathbf{b} , under the assumption that the singular values following the j^{th} are zero, is given by $\sqrt{\rho_j^2/(M-j)}$. This quantity is printed with the heading "Scaled Sqrt of Cum. S.S." for Scaled Square Root of the Cumulative Sum of Squares.

4. The quantities $\|\mathbf{y}^{(j)}\|$ are computed (using the p_i 's) and printed along with the corresponding values of ρ_j , with the headings YNORM and RNORM.
5. A range of values of the Levenberg-Marquardt parameter, λ , and associated $\|\mathbf{y}^\lambda\|$ and $\|AD\mathbf{y}^\lambda - \mathbf{b}\|$ are computed and printed. These quantities are computed from formulas involving s_i 's, p_i 's, and g_i 's.
6. The candidate solutions, $\mathbf{x}^{(j)} = D\mathbf{y}^{(j)}$, are computed and printed.

SSVA calls SPRTSV to print the V matrix and the candidate solutions.

D.3 Method for SCOV3

If the variance of the error in the data vector, \mathbf{b} , is VAR, the covariance matrix, C , for the solution vector, \mathbf{x} , is conventionally defined as $C = \text{VAR} \times (A^t A)^{-1}$. Using the singular value decomposition, $A = USV^t$, one can write $C = \text{VAR} \times (V S^t S V^t)^{-1} = \text{VAR} \times V S^+ (V S^+)^t$. SCOV3 computes C using this latter formula.

References

1. Charles L. Lawson and Richard J. Hanson, **Solving Least-Squares Problems**, Prentice-Hall, Englewood Cliffs, N. J. (1974) 340 pages.

E. Error Procedures and Restrictions

In SSVA, if $M \leq 0$ or $N \leq 0$ an immediate return will be made.

SSVDRS will issue an error message, set SING(1) = -1.0, and do an immediate return if any of the following conditions are noted.

$$M < 1, N < 1, NB < 0, LDA < \max(M, N),$$

$$LDB < M \text{ when } NB > 0, \text{ or } LDB < 1 \text{ when } NB = 0.$$

SCOV3 requires N nonzero singular values. If this is satisfied it will set IERR = 0. Otherwise it will set IERR to

the index of the first zero singular value and the results returned in A(.) will be meaningless.

The subroutine SCOV3 is intended for use following SSVDRS. It cannot be used following SSVA since SSVA does not leave the V matrix in the A(.) array on return.

F. Supporting Information

The source language is ANSI Fortran 77.

These subroutines were adapted from [1] for use with Fortran 77 by C. L. Lawson and S. Y. Chiu, May 1986, June 1987. Altered March 1989 by Lawson to introduce the vector KPVEC() to provide more report options.

Entry	Required Files
DCOV3	DCOPY, DCOV3, DDOT, DSCAL, ERFIN, ERMSG, IERM1, IERV1
DSVA	AMACH, DAXPY, DCOPY, DDOT, DHTCC, DHTGEN, DNRM2, DPRTSV, DQRBD, DROT, DROTG, DSVA, DSVDRS, DSWAP, ERFIN, ERMOR, ERMSG, IERV1
DSVDRS	AMACH, DAXPY, DCOPY, DDOT, DHTCC, DHTGEN, DNRM2, DQRBD, DROT, DROTG, DSVDRS, DSWAP, ERFIN, ERMOR, ERMSG, IERV1
SCOV3	ERFIN, ERMSG, IERM1, IERV1, SCOPY, SCOV3, SDOT, SSCAL
SSVA	AMACH, ERFIN, ERMOR, ERMSG, IERV1, SAXPY, SCOPY, SDOT, SHTCC, SHTGEN, SNRM2, SPRTSV, SQRBD, SROT, SROTG, SSVA, SSVDRS, SSWAP
SSVDRS	AMACH, ERFIN, ERMOR, ERMSG, IERV1, SAXPY, SCOPY, SDOT, SHTCC, SHTGEN, SNRM2, SQRBD, SROT, SROTG, SSVDRS, SSWAP

DRDSVA

```

program DRDSVA
c>> 1996-06-27 DRDSVA   Krogh   Special code for C conversion.
c>> 1996-05-28 DRDSVA   Krogh   Removed implicit statement.
c>> 1994-10-19 DRDSVA   Krogh   Changes to use M77CON
c>> 1989-03-07 DRDSVA   CLL
c      Demo driver for DSVA, Singular Value Analysis.
c
c      Mar 1987, C. L. Lawson & S. Y. Chiu, JPL. This sample problem
c      taken from the book, Solving Least Squares Problems,
c      Prentice-Hall, 1974, by C. L. Lawson and R. J. Hanson.
c
c-----D replaces "?:": DR?SVA, ?SVA
c
integer I, J, KPVEC(4), M, MDATA, MMAX, N, NMAX, NMAX2

```

```

parameter(MMAX = 15, NMAX = 5, NMAX2 = NMAX*2)
double precision A(MMAX,NMAX), B(MMAX), SING(NMAX)
double precision D(NMAX), WORK(NMAX2)
character*6 NAMES(NMAX)

c
  data ((A(I,J),J=1,5),I=1,15) /
*- .13405547D0, -.20162827D0, -.16930778D0, -.18971990D0, -.17387234D0,
*- .10379475D0, -.15766336D0, -.13346256D0, -.14848550D0, -.13597690D0,
* -.08779597D0, -.12883867D0, -.10683007D0, -.12011796D0, -.10932972D0,
* .02058554D0, .00335331D0, -.01641270D0, .00078606D0, .00271659D0,
* -.03248093D0, -.01876799D0, .00410639D0, -.01405894D0, -.01384391D0,
* .05967662D0, .06667714D0, .04352153D0, .05740438D0, .05024962D0,
* .06712457D0, .07352437D0, .04489770D0, .06471862D0, .05876455D0,
* .08687186D0, .09368296D0, .05672327D0, .08141043D0, .07302320D0,
* .02149662D0, .06222662D0, .07213486D0, .06200069D0, .05570931D0,
* .06687407D0, .10344506D0, .09153849D0, .09508223D0, .08393667D0,
* .15879069D0, .18088339D0, .11540692D0, .16160727D0, .14796479D0,
* .17642887D0, .20361830D0, .13057860D0, .18385729D0, .17005549D0,
* .11414080D0, .17259611D0, .14816471D0, .16007466D0, .14374096D0,
* .07846038D0, .14669563D0, .14365800D0, .14003842D0, .12571177D0,
* .10803175D0, .16994623D0, .14971519D0, .15885312D0, .14301547D0 /

c
  data (B(I),I=1,15) /
*- .4361D0, -.3437D0, -.2657D0, -.0392D0, .0193D0, .0747D0,
* .0935D0, .1079D0, .1930D0, .2058D0, .2606D0, .3142D0,
* .3529D0, .3615D0, .3647D0 /

c
  data NAMES /'Earth ', 'Water ', 'Air ', 'Fire ', 'Time '/
  data KPVEC / 1, 111111, -1, 76/

c
c
c
M = MMAX
N = NMAX
MDATA = M
print '( ' DRDSVA.. Demonstrate Singular Value Analysis ' /
* ' Listing of input matrix, A, and vector, B, follows..' )'
c++ Code for .C. is active
print '(/(5F12.7,F13.4) )',
* ((A(I,J),J=1,N),B(I),I=1,M)
c++ Code for .C. is inactive
c%% for (i = 1; i <= m; i++){
c%% printf( "\n" );
c%% for (j = 1; j <= n; j++){
c%% printf( "%12.7f", a[j - 1][i - 1] );
c%% printf( "%13.4f", B[i] );}
c%% fprintf( stdout, " \n\n\n \n" );
c++ End
print '(1X//1X) '

c
call DSVA(A, MMAX, M, N, MDATA, B, SING, KPVEC, NAMES, 1, D, WORK)

c
stop
end

```

ODDSVA

DRDSVA.. Demonstrate Singular Value Analysis
Listing of input matrix, A, and vector, B, follows..

-0.1340555	-0.2016283	-0.1693078	-0.1897199	-0.1738723	-0.4361
-0.1037948	-0.1576634	-0.1334626	-0.1484855	-0.1359769	-0.3437
-0.0877960	-0.1288387	-0.1068301	-0.1201180	-0.1093297	-0.2657
0.0205855	0.0033533	-0.0164127	0.0007861	0.0027166	-0.0392
-0.0324809	-0.0187680	0.0041064	-0.0140589	-0.0138439	0.0193
0.0596766	0.0666771	0.0435215	0.0574044	0.0502496	0.0747
0.0671246	0.0735244	0.0448977	0.0647186	0.0587645	0.0935
0.0868719	0.0936830	0.0567233	0.0814104	0.0730232	0.1079
0.0214966	0.0622266	0.0721349	0.0620007	0.0557093	0.1930
0.0668741	0.1034451	0.0915385	0.0950822	0.0839367	0.2058
0.1587907	0.1808834	0.1154069	0.1616073	0.1479648	0.2606
0.1764289	0.2036183	0.1305786	0.1838573	0.1700555	0.3142
0.1141408	0.1725961	0.1481647	0.1600747	0.1437410	0.3529
0.0784604	0.1466956	0.1436580	0.1400384	0.1257118	0.3615
0.1080317	0.1699462	0.1497152	0.1588531	0.1430155	0.3647

Singular Value Analysis of the least squares problem, $A * X = B$,
scaled as $(A * D) * Y = B$.

M = 15, N = 5, MDATA = 15

Scaling option No. 1. D is the identity matrix.

V-Matrix of the Singular Value Decomposition of A*D.
(Elements of V scaled up by a factor of 10**4)

	COL 1	COL 2	COL 3	COL 4
1 Earth	3742.	-7526.	3382.	1981.
2 Water	5196.	-636.	2301.	-6349.
3 Air	4123.	6510.	4741.	1067.
4 Fire	4796.	689.	-2493.	6877.
5 Time	4359.	302.	-7388.	-2707.

	COL 5
1 Earth	3741.
2 Water	-5195.
3 Air	4123.
4 Fire	-4797.
5 Time	4359.

INDEX	SING. VAL.	P COEF	RECIPROCAL SING. VAL.	G COEF	SCALED SQRT of CUM.S.S.
0					0.2635
1	1.000	0.9998	1.000	0.9998	0.5452E-01
2	0.1000	2.000	10.00	0.2000	0.1111E-01
3	0.1000E-01	-4.005	100.0	-0.4005E-01	0.4055E-04
4	0.9997E-05	-1.776	0.1000E+06	-0.1776E-04	0.4201E-04
5	0.9717E-07	-192.7	0.1029E+08	-0.1872E-04	0.4366E-04

INDEX	SING. VAL.	G COEF	G**2	CUMULATIVE SUM of SQRS	SCALED SQRT of CUM.S.S.
-------	------------	--------	------	---------------------------	----------------------------

0				1.041	0.2635
1	1.000	0.9998	0.9996	0.4162E-01	0.5452E-01
2	0.1000	0.2000	0.4001E-01	0.1604E-02	0.1111E-01
3	0.1000E-01	-0.4005E-01	0.1604E-02	0.1973E-07	0.4055E-04
4	0.9997E-05	-0.1776E-04	0.3153E-09	0.1941E-07	0.4201E-04
5	0.9717E-07	-0.1872E-04	0.3505E-09	0.1906E-07	0.4366E-04

INDEX	YNORM	RNORM	LOG10 YNORM	LOG10 RNORM
0	0.000E+00	0.102E+01	-1000.000	0.009
1	0.100E+01	0.204E+00	-0.000	-0.690
2	0.224E+01	0.400E-01	0.350	-1.397
3	0.459E+01	0.140E-03	0.662	-3.852
4	0.492E+01	0.139E-03	0.692	-3.856
5	0.193E+03	0.138E-03	2.285	-3.860

Norms of solution and residual vectors for a range of values of the Levenberg-Marquardt parameter, LAMBDA.

LAMBDA	YNORM	RNORM	LOG10 LAMBDA	LOG10 YNORM	LOG10 RNORM
0.100E+02	0.990E-02	0.101E+01	1.000	-2.004	0.005
0.354E+01	0.738E-01	0.948E+00	0.549	-1.132	-0.023
0.126E+01	0.388E+00	0.644E+00	0.099	-0.411	-0.191
0.445E+00	0.840E+00	0.255E+00	-0.352	-0.076	-0.593
0.158E+00	0.113E+01	0.150E+00	-0.802	0.054	-0.824
0.558E-01	0.183E+01	0.614E-01	-1.253	0.262	-1.212
0.198E-01	0.232E+01	0.328E-01	-1.704	0.365	-1.484
0.701E-02	0.349E+01	0.132E-01	-2.154	0.543	-1.878
0.248E-02	0.438E+01	0.233E-02	-2.605	0.642	-2.632
0.880E-03	0.456E+01	0.339E-03	-3.056	0.659	-3.470
0.312E-03	0.458E+01	0.146E-03	-3.506	0.661	-3.836
0.110E-03	0.459E+01	0.141E-03	-3.957	0.661	-3.852
0.391E-04	0.459E+01	0.140E-03	-4.407	0.662	-3.853
0.139E-04	0.463E+01	0.140E-03	-4.858	0.665	-3.854
0.491E-05	0.481E+01	0.139E-03	-5.309	0.682	-3.856
0.174E-05	0.494E+01	0.139E-03	-5.759	0.693	-3.856
0.617E-06	0.678E+01	0.139E-03	-6.210	0.831	-3.856
0.218E-06	0.322E+02	0.139E-03	-6.661	1.508	-3.857
0.774E-07	0.118E+03	0.138E-03	-7.111	2.072	-3.859
0.274E-07	0.179E+03	0.138E-03	-7.562	2.252	-3.860
0.972E-08	0.191E+03	0.138E-03	-8.012	2.281	-3.860

Sequence of candidate solutions, X

	SOLN 1	SOLN 2	SOLN 3	SOLN 4
1 Earth	0.374096	-1.13139	-2.48573	-2.83753
2 Water	0.519519	0.392268	-0.529133	0.598666
3 Air	0.412234	1.71454	-0.184141	-0.373618
4 Fire	0.479494	0.617368	1.61568	0.394121
5 Time	0.435809	0.496254	3.45479	3.93558
	SOLN 5			
1 Earth	-74.9158			
2 Water	100.682			
3 Air	-79.8044			
4 Fire	92.8170			
5 Time	-80.0529			