Warning: Subroutine EB10 performs functions which are adequately treated by routines in other standard subroutine libraries (for example, LAPACK). The use of this routine is not recommended, and it may be removed from future releases of this library.

1 SUMMARY

Computes the **singular value decomposition of a real matrix** $\mathbf{X}=\{x_{ij}\}_{n\times p}$. This decomposition takes the form $\mathbf{X}=\mathbf{USV}^T$, where **U** and **V** are orthogonal matrices and the matrix $\mathbf{S}=\{s_{ij}\}_{n\times p}$ is zero except for the diagonal elements $s_{ii}=\sigma_i$, i=1,2,...,min$\{n,p\}$, which are the singular values of **X**.

The computation of **U** and **V** is optional and there is an option to compute $\mathbf{U}^T\mathbf{B}$ where **B** is specified by the user.

The subroutine uses a variant of the algorithm described in Golub and Reinsch, 'Singular Value Decomposition and Least Squares Solutions', Numer. Math., Vol. 14, 1970.

ATTRIBUTES — Version: 1.0.0. **Types:** EB10A; EB10AD. **Calls:** _DOT, _ROTG, _ROT, _NRM2. **Original date:** March 1978. **Origin:** G.W.Stewart, Univ. of Maryland (for LINPACK); modified for Harwell by D.M.Gay, MIT.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence

The single precision version

```
      CALL EB10A(X,LDX,N,P,S,LS,U,LDU,NCU,V,
     +           LDV,NCV,B,LDB,NCB,E,LE,IFAIL)
```

The double precision version

```
      CALL EB10AD(X,LDX,N,P,S,LS,U,LDU,NCU,V,
     +           LDV,NCV,B,LDB,NCB,E,LE,IFAIL)
```

X is a two-dimensional REAL (DOUBLE PRECISION in the D version) array dimensioned (LDX,P), in whose first n rows and p columns the user must store the matrix to be decomposed. These rows and columns are overwritten by the subroutine.

LDX is an INTEGER variable which must be set by the user to the first dimension of the array X. The subroutine does not alter this parameter. **Restriction:** LDX $\geq n$.

N is an INTEGER variable which must be set by the user to n the number of rows in the matrix **X**. The subroutine does not alter this parameter. **Restriction:** $n\geq 1$.

P is an INTEGER variable which must be set by the user to p the number of columns in the matrix **X**. The subroutine does not alter this parameter. **Restriction:** $p\geq 1$.

S is a REAL (DOUBLE PRECISION in the D version) array of length LS $\geq \min(n+1,p)$. When the subroutine returns, S contains the singular values in descending order, i.e. S(i)=σ_i for $1\leq i\leq \min(n,p)$, with $\sigma_i\geq\sigma_{i+1}\geq 0$ (and with S(N+1)=0 if $n<p$).

LS is an INTEGER variable set by the user to the length of the array S. The subroutine does not alter this parameter. **Restriction:** LS $\geq \min(n+1,p)$.

U is a two-dimensional REAL (DOUBLE PRECISION in the D version) array dimensioned (LDU,NCU). If NCU $\geq n$, then the subroutine stores the entire matrix **U** of left singular vectors in U. If $p \leq$ NCU $< n$, then the subroutine stores the first NCU left singular vectors in U. Otherwise, i.e. if NCU $< \min(n,p)$, U is not referenced. If LDU $\leq$ LDX, then U may share storage with X.

LDU is an INTEGER variable which must be set by the user to the first dimension of the array U. The subroutine does not alter this parameter. **Restriction:** LDU $\geq n$.

NCU is an INTEGER variable which must be set by the user to the number of columns in **U**. It controls the number (if any) of left singular vectors returned in U — see the description of U above. The subroutine does not alter this parameter.

V is a two-dimensional REAL (DOUBLE PRECISION in the D version) array dimensioned (LDV,NCV). If NCV $\geq p$, then the subroutine stores the entire matrix **V** of right singular vectors in V. If $n+1 \leq$ NCV $< p$, then the subroutine stores the first NCV right singular vectors in V. Otherwise, i.e. if NCV $< \min(n+1,p)$, V is not referenced. If LDV $\leq$ LDX, then V may share storage with X.

LDV is an INTEGER variable which must be set by the user to the first dimension of the array V. The subroutine does not alter this parameter. **Restriction:** LDV $\geq p$.

NCV is an INTEGER variable which must be set by the user to the number of columns in **V**. It controls the number (if any) of right singular vectors returned in V — see the description of V above. The subroutine does not alter this parameter.

B is a two-dimensional REAL (DOUBLE PRECISION in the D version) array dimensioned (LDB,NCB) in whose first n rows and first NCB columns the user must store the NCB column vectors **b** to which the transformation **U** is to be applied; the subroutine returns $\mathbf{U}^T\mathbf{b}$ in B. If NCB ≤ 0, then B is not referenced.

LDB is an INTEGER variable which must be set by the user to the first dimension of the array B. The subroutine does not alter this parameter. **Restriction:** LDB $\geq n$.

NCB is an INTEGER variable which must be set by the user to the number of columns in B. The subroutine does not alter this parameter.

E is a REAL (DOUBLE PRECISION in the D version) array of length LE $\geq n+\max(n,p)$ which the subroutine uses for temporary storage.

LE is an INTEGER variable which must be set by the user to the length of the array E. The subroutine does not alter this parameter. **Restriction:** LE $\geq n+\max(n,p)$.

IFAIL is an INTEGER variable which is always set by the subroutine to indicate success or failure. On exit from the subroutine IFAIL will have one of the following values:

- 0 successful entry
- –1 parameter restriction violated — parameter values passed are echoed in the error message.
- >0 the QR algorithm failed with IFAIL singular values remaining to be determined. This should not occur. Please report this if it happens.

2.2 The common area and diagnostic messages .

The subroutine uses a common area which the user may also reference. To do this the calling program should include a COMMON statement of the form

The single precision version

```
COMMON/EB10B/ LP
```

The double precision version

```
COMMON/EB10BD/ LP
```

`LP` is an `INTEGER` variable which specifies the Fortran unit number to be used for the error messages. The default value is 6 (line printer). To suppress the printing of error messages set `LP` to zero.

3 GENERAL INFORMATION

Use of common: The subroutine uses the common area `EB10B/BD`, see §2.2.

Workspace: None.

Other routines called directly: the following subroutines are called: `_DOT`, `_ROTG`, `_ROT` and `_NRM2`.

Input/output: in the event of errors diagnostic messages are printed, see §2.2.

Restrictions: $n > 0$, $\texttt{LDX} \geq n$, $p > 0$, $\texttt{LS} \geq \min(n+1,p)$, $\texttt{LDU} \geq n$, $\texttt{LDV} \geq p$, $\texttt{LDB} \geq n$ and $\texttt{LE} \geq n+\max(n,p)$.

4 METHOD

The algorithm used is a variant of the one described bu Golub and Reinsch (1970). First **X** is reduced to bi-diagonal form via Householder transformations, then the implicitly shifted QR algorithm is applied to find the singular values. These are sorted into descending order, and the columns of **U** and **V** (if requested) are permuted accordingly. Care is taken to do operations columnwise. The algorithm is described in more detail in chapter 10 of Dongarra et al (1977).

References

Dongarra, J.J., Bunch, J.R., Moler, C.B. and Stewart, G.W. (1977). "LINPACK Working Note #9, Preliminary LINPACK User's Guide", Technical Memorandum No. 313, Applied Mathematics Division, Argonne National Laboratory, Argonne, Illinois 60439, U.S.A.

Golub, G.H. and Reinsch, C. (1970). "Singular Value Decomposition and Least Squares Solutions", Numer. Math. **14**, pp.403-420. Also: contribution I/10 of "Handbook for Automatic Computation", Vol. II, Linear Algebra (J.H. Wilkinson & C. Reinsch, Editors), Springer-Verlag, Berlin and New York, 1971.