

## 1 SUMMARY

To calculate the generalized inverse of a given  $m \times n$  rectangular matrix  $\mathbf{A}$ . One difficulty is deciding on the rank of  $\mathbf{A}$ , so the user can specify it, or provide an estimate of the accuracy of the elements of  $\mathbf{A}$ , in order that the rank can be estimated automatically.

**ATTRIBUTES** — **Version:** 1.0.0. **Types:** MB10A, MB10AD **Calls:** SDOT/DDOT. **Original date:** June 1970. **Origin:** M.J.D. Powell, Harwell.

## 2 HOW TO USE THE PACKAGE

### 2.1 The argument list and calling sequence

*The single precision version*

```
CALL MB10A(M,N,A,IA,AINV,IV,TOL,IRANK,W)
```

*The double precision version*

```
CALL MB10AD(M,N,A,IA,AINV,IV,TOL,IRANK,W)
```

$M$  is an INTEGER giving the first dimension of the matrix  $\mathbf{A}$ .

$N$  is an INTEGER giving the second dimension of the matrix  $\mathbf{A}$ .

$A$  is a REAL (DOUBLE PRECISION in the D version) two dimensional array, whose first dimension is  $IA$ , set by the user to the initial matrix  $\mathbf{A}$ .

$IA$  is an INTEGER giving the first dimension of the array  $A$ .

**Restriction**  $IA \geq M$ .

$AINV$  is a REAL (DOUBLE PRECISION in the D version) two dimensional array, whose first dimension is  $IV$ , which will be set by the routine to the transpose of the generalized inverse.

$IV$  is an INTEGER giving the first dimension of the array  $AINV$ .

**Restriction**  $IV \geq M$ .

$TOL$  is a REAL (DOUBLE PRECISION in the D version) variable set by the user to the required tolerance (see below).

$IRANK$  is an INTEGER set by the user to control the way in which  $TOL$  is used. If and only if the value of  $IRANK$  is positive,  $TOL$  serves as an upper bound on the rank of the calculated generalized inverse. Its value will change during the execution of the routine, when it is set to the rank of  $AINV$ .

$W$  is a one-dimensional REAL (DOUBLE PRECISION in the D version) array of dimension  $2N$ .

### 2.2 The choice of IRANK

In practice the elements of  $\mathbf{A}$  will include some error, so if just these elements are specified it is often not feasible to estimate the rank of  $\mathbf{A}$ . Moreover, if the estimated rank is too great, then consequently the calculated elements of  $AINV$  can be much too large. Therefore we have introduced a single parameter  $TOL$  to permit the user to indicate the error in the matrix elements.

Our aim is that the rank of  $\mathbf{A}$  should be equal to the least rank of all  $m \times n$  matrices  $\mathbf{B}$  that are close to  $\mathbf{A}$ , in the sense that they satisfy the condition

$$\|b_i - a_i\|_2^2 \leq TOL, \quad i=1,2,\dots,n, \quad (1)$$

where  $a_i$  and  $b_i$  are the  $i$ th columns of  $\mathbf{A}$  and  $\mathbf{B}$  respectively. But this aim is difficult to achieve, so instead we apply an iterative process to identify the dominant columns of  $\mathbf{A}$ . Specifically we first calculate  $(n-1)$  vectors by projecting the largest column of  $\mathbf{A}$  out of the remaining columns. Then we find the largest of these  $(n-1)$  vectors and we project it out of the rest of the  $(n-1)$  vectors to obtain  $(n-2)$  vectors. The process continues with the lengths of all the residual vectors are less than the square root of TOL, and IRANK is set to the number of times the process has to be applied. The IRANK vectors that are projected lead to IRANK dominant columns of  $\mathbf{A}$ , and we define the matrix  $\mathbf{B}$  by replacing the non-dominant columns of  $\mathbf{A}$  by their projections in the space spanned by the dominant columns. Clearly  $\mathbf{B}$  satisfies condition (1), and its rank tends to be rather small, so we set the elements of AINV to those of  $\mathbf{B}^+$ , the generalized inverse of  $\mathbf{B}$ .

The fact that TOL is a single parameter is not very restrictive, because if the user wishes the accuracy to be dependent on the column (or row) number, he can instead scale the matrix  $\mathbf{A}$  by a diagonal matrix. Indeed if the generalized inverse of  $\mathbf{D}_1 \mathbf{P} \mathbf{D}_2$  is calculated by MB10A, where  $\mathbf{D}_1$  and  $\mathbf{D}_2$  are non-singular square matrices, then the generalized inverse of  $\mathbf{P}$  is equal to

$$\mathbf{P}^+ = \mathbf{D}_2 (\mathbf{D}_1 \mathbf{P} \mathbf{D}_2)^+ \mathbf{D}_1.$$

### 3 GENERAL INFORMATION

**Use of common:** None.

**Workspace:** None.

**Input/output:** None.

**Restrictions:** None.

### 4 METHOD

Further details of the method will be published by M.J.D. Powell.