1 SUMMARY

To **‘‘downdate’’ a triangular factorization of a positive definite symmetric matrix,** which means to revise the factorization when a symmetric matrix of rank one is subtracted from a given matrix in factored form. In order to allow for the pivoting operations that can re-order rows and columns when a Cholesky factorization is done, the calculation of the subroutine is as follows.

Given an $m \times n$ upper triangular matrix **U**, where $n \geq m$, given an $n \times n$ permutation matrix **P** , and given an n component vector **z**, to calculate an upper triangular matrix $\mathbf{U}^+$ and a permutation matrix $\mathbf{P}^+$ that satisfy the equation

$$(\mathbf{U}^+\mathbf{P}^+)^T(\mathbf{U}^+\mathbf{P}^+) = (\mathbf{UP})^T(\mathbf{UP}) - \mathbf{zz}^T.$$

There is no error return if the matrix on the right hand side of this equation has a negative eigenvalue, because this situation occurs frequently due to computer rounding errors. Instead the equation is satisfied as closely as possible using only of order mn computer operations. The user has control over the number of rows of $\mathbf{U}^+$.

ATTRIBUTES — Version: 1.0.0. **Types:** `MC27A, MC27AD`. **Original date:** January 1981 **Origin:** M.J.D.Powell, Cambridge University. **Licence:** A third-party licence for this package is available without charge.

2 HOW TO USE THE PACKAGE

2.1 Argument list and calling sequence

The single precision version

```
CALL MC27A(U,M,N,Z,IPOINT,NEWM,W)
```

The double precision version

```
CALL MC27AD(U,M,N,Z,IPOINT,NEWM,W)
```

`U` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least equal to the number of components in the upper triangular part of **U**, i.e. $\frac{1}{2}m(2n+1-m)$. The user must set this array to contain the elements of the matrix $\mathbf{U}=\{u_{ij}\}_{m\times n}$ in the order $u_{1,1}, u_{1,2},.., u_{1,n}, u_{2,2}, u_{2,3},.., u_{2,n},.., u_{m,m}, u_{m,m+1},.., u_{m,n}$. On a normal return this array holds the elements of $\mathbf{U}^+$ in the same order.

`M` is an `INTEGER` variable which must be set initially by the user to m, the number of rows of **U**. Its final value is the number of rows of $\mathbf{U}^+$, but an error return occurs with `M` set to zero if any of the restrictions listed in §3 are violated. **Restriction:** $1 \leq m \leq n$.

`N` is an `INTEGER` variable which must be set by the user to n, the number of columns of **U**. Because $\mathbf{U}^+$ has the same number of columns, the value of `N` is not altered by the subroutine. **Restriction:** $n > 0$.

`Z` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least n which must be set by the user to contain the components of the vector **z**, mentioned in §1. Its elements are not altered by the subroutine.

`IPOINT` is an `INTEGER` array of length at least n which must be set by the user to contain a permutation of the first n integers, in order to define the matrix **P** . For `I=1,N, IPOINT(I)` is the column number of the non-zero element in the I-th row of **P** . Therefore, if one forms the matrix **UP** by rearranging the columns of **U** then the `IPOINT(I)-th` column of **UP** is the I-th column of **U**. Any call of the subroutine may make further exchanges of columns, and then the elements of `IPOINT` are reordered so that their final values define the permutation matrix $\mathbf{P}^+$. Thus the output from the subroutine `MC27` is suitable for a further downdating calculation.

`NEWM` is an `INTEGER` variable which must be set by the user to the required number of rows of the calculated matrix $\mathbf{U}^+$. The actual number of rows is returned in `M,` and except on an error return, the number is the minimum of the initial value of M and NEWM, because it is not possible to add rows to **U**. When a decrease in the number of rows is requested then a pivoting procedure helps to make small the rows that are deleted. The value of NEWM is not altered by the subroutine.

W is a REAL (DOUBLE PRECISION in the D version) array of length at least $2n$ whose elements are used for working space.

3 GENERAL INFORMATION

Workspace: $2n$ integers provided by the user.

Use of common: none.

Other routines called directly: none.

Input/output: none.

Restrictions:

$1 \le m \le n$,
$n > 0$.

The diagonal elements of the upper triangular matrix that is set in the array U on entry to the subroutine must all be non-zero.

The elements of IPOINT must be a permutation of the first n integers.

Accuracy: In order to achieve good accuracy it is important that, in each row of the given matrix **U**, the modulus of the diagonal element should be one of the larger of the moduli of the elements. Further, in order that the calculated matrix $\mathbf{U}^+$ is suitable for achieving good accuracy in a subsequent downdating calculation, it is helpful if the moduli of the diagonal elements are in decreasing order, or at least do not increase greatly.

4 METHOD

If $(\mathbf{U}^T\mathbf{U} - \mathbf{z}\mathbf{z}^T)$ has no negative eigenvalues, then an algorithm that is described by Gill, Golub, Murray and Saunders (Maths. of Comp., **28** (1974), 505-535) is used to calculate a suitable matrix $\mathbf{U}^+$ that has as many rows as **U**. Otherwise a new technique that is proposed by Powell (to be published) is applied. The pivoting technique tends to arrange the rows of $\mathbf{U}^+$ in decreasing order of their lengths. Therefore if NEWM is less than M, the later rows are deleted, and some loss of accuracy is prevented by a projection technique that is also described by Powell.

The method includes several Givens rotations, which are all done in the way that requires the calculation of square roots.