1 SUMMARY

This subroutine performs the arithmetic operations corresponding to two steps of Gaussian elimination.

ATTRIBUTES — Version: 1.0.0. Types: MC31A; MC31AD. Remark: This is a double pivot version of MC32. Calls: None. Language: A Cray-2 assembler version is available and is more than twice as fast as the equivalent vectorized version Fortran code. Original date: August 1986. Origin: I. S. Duff, Harwell and A. K. Dave, CRAY Research (UK) Ltd.

2 HOW TO USE THE PACKAGE

Each call to MC31 is equivalent to the execution of the Fortran code:

```fortran
DO 20 J=1,LFRONT
   DO 10 I=1,KFRONT
      FA(I,J) = FA(I,J) + PVKOL(I,1)*QQ(1,J) + PVKOL(I,2)*QQ(2,J)
   10 CONTINUE
20 CONTINUE
```

2.1 Argument list

The single precision version

```fortran
CALL MC31A(FA,NFRONT,KFRONT,LFRONT,PVKOL,QQ)
```

The double precision version

```fortran
CALL MC31AD(FA,NFRONT,KFRONT,LFRONT,PVKOL,QQ)
```

FA is a REAL (DOUBLE PRECISION in the D version) array of two dimensions, with leading dimension NFRONT. The user must set the leading KFRONT by LFRONT subarray to the values of the matrix entries prior to the Gaussian elimination steps to be executed by the call to MC31A/AD. On exit, FA contains the values of the matrix entries after the Gaussian elimination step. Only the leading KFRONT by LFRONT subarray is modified by MC31A/AD.

NFRONT is an INTEGER variable which must be set on entry to the leading dimension of arrays FA and PVKOL. It is not altered by MC31A/AD.

KFRONT is an INTEGER variable which must be set on entry to the number of entries in the pivot column (excluding the pivot). It is not altered by MC31A/AD.

LFRONT is an INTEGER variable which must be set on entry to the number of entries in the pivot row (excluding the pivot). It is not altered by MC31A/AD.

PVKOL is a REAL (DOUBLE PRECISION in the D version) array of two dimensions, with leading dimension NFRONT. The user must set the first NFRONT by 2 elements of this to the negative of the subdiagonal entries in the two pivot columns. PVKOL is not altered by MC31A/AD.
QQ is a REAL (DOUBLE PRECISION in the D version) array of two dimensions, with leading dimension 2 and a second dimension of at least LFRONT. The user must set these to the superdiagonal entries in the pivot rows divided by the pivots. QQ is not altered by MC31A/AD.

2.3 Common

None.

2.4 Errors and diagnostic messages

None.

3 GENERAL INFORMATION

Workspace: There are no work arrays used by MC31A/AD.

Use of common: The subroutine does not use any common areas or auxiliary subroutines.

Other routines called directly: None

Input/output: None

Portability: There is a CRAY-2 Assembler Language version available.

Restrictions: None.

4 METHOD

For the CRAY version, CRAY Assembler Language (CAL) is used to ensure that the vector PVKOL is kept in the vector registers throughout the sequence of inner loops. Thus there is only one memory fetch/store for each arithmetic operation and a maximum asymptotic rate of nearly 280 Megaflops is possible.

For systems of moderate size, n<50, a Gaussian elimination program based around this subroutine will typically run 1.5 times faster than the equivalent program based around the single pivot subroutine MC32A/AD, and for large systems it will typically run 1.75 times faster.

5 EXAMPLE OF USE

The following example shows the use of the subroutine MC31A to factorise a full matrix using Gaussian elimination without pivoting.

```fortran
PROGRAM MC31T
PARAMETER ( NMAX = 20 , NMAXP1 = NMAX + 1 )
REAL A(NMAX,NMAXP1), PVKOL(NMAX,2), QQ(2,NMAX), RHS1,
+ ZERO, EPSOLN, A11, A12, A21, A22, TMAX, ERRMAX
INTEGER I, J, K, N, NMK, NMKP1, POSITN
EXTERNAL FA01AS
REAL FA01AS
DATA ZERO/0.0E0/, EPSOLN/1.0E-09/
WRITE (6,99999)
N = NMAX
C GENERATE RANDOM MATRIX WITH
C RIGHT HAND SIDE AS COLUMN N+1.
DO 20 I=1,N
RHS1 = ZERO
DO 10 J=1,N
A(I,J) = FA01AS(-1)
```

All use is subject to licence.

http://www.hsl.rl.ac.uk/
RHS1 = RHS1 + A(I,J)*FLOAT(J)
10 CONTINUE
  A(I,N+1) = RHS1
20 CONTINUE

C EACH PASS THROUGH THIS LOOP PERFORMS TWO
C STEPS OF GAUSSIAN ELIMINATION.
DO 70 K=1,N,2
  NMK = N - K - 1
  C HOLD TWO BY TWO BLOCK.
  A11 = A(K,K)  
  A12 = A(K,K+1)/A11
  C STORE UPPER TRIANGLE READY FOR BACKSOLUTION.
  A(K,K+1) = A12
  C JUMP IF AT LAST ONE BY ONE PIVOT.
  IF (NMK.GE.0) THEN
  C SAVE REST OF TWO BY TWO BLOCK.
    A21 = -A(K+1,K)
    A22 = A(K+1,K+1) + A21*A12
  C CREATE QQ
  40 DO 50 I=1,NMK+1
    QQ(1,I) = A(K,K+I+1)/A11
    QQ(2,I) = (A(K+1,K+I+1)+A21*QQ(1,I))/A22
  50 CONTINUE
  C CREATE PVKOL
  DO 30 I=1,NMK
    PVKOL(I,1) = -A(K+I+1,K)  
    PVKOL(I,2) = -(A(K+I+1,K+1) + PVKOL(I,1)*A12)
  30 CONTINUE
  C PERFORM GAUSSIAN ELIMINATION
  IF (NMK .NE. 0) THEN
    CALL MC31A(A(K+2,K+2), NMAX, NMK, NMK+1, PVKOL, QQ)
  ENDIF
  70 CONTINUE
C PERFORM BACK SUBSTITUTION
DO 90 I=N-1,1,-1
  DO 80 J=I+1,N
    A(I,N+1) = A(I,N+1) - A(I,J)*A(J,N+1)
  80 CONTINUE
  90 CONTINUE
100 CONTINUE

C PRINT MAXIMUM ERROR AND THE ASSOCIATED COMPONENT
  ERRMAX = ZERO
  POSITN = 0
  DO 110 I=1,N
    TMAX = AMAX1( ABS(A(I,N+1)-FLOAT(I)), ERRMAX)
    IF (TMAX .NE. ERRMAX) THEN
      POSITN = I
      ERRMAX = TMAX
    ENDIF
  110 CONTINUE
WRITE(6,99998) ERRMAX , POSITN , A(POSITN,N+1)
  99999 FORMAT (" OUTPUT FROM TEST OF MC31A ")
This produces the following output:

```
OUTPUT FROM TEST OF MC31A
MAXIMUM ERROR = 0.65995E-10 ATTAINED AT COMPONENT 1
VALUE OF COMPONENT = 0.10000E+01
```