

```

SUBROUTINE ZM_INVERSE(A,N,B,DET)
USE FMVALS
USE FMZM
IMPLICIT NONE

! Return B as the inverse of the N x N matrix A, and DET as the determinant of A.

! A and B are type (zm) (complex) multiprecision arrays.

INTEGER :: N
TYPE (ZM) :: A(N,N), B(N,N), DET
TYPE (FM), SAVE :: TOL
TYPE (ZM), ALLOCATABLE :: A1(:,,:), A2(:,,:), B1(:), R1(:), X1(:)
INTEGER, ALLOCATABLE :: KSWAP(:)
INTEGER :: I, J, K, KWARN_SAVE, NDSAVE

CALL FM_ENTER_USER_ROUTINE
TOL = EPSILON(TO_FM(1))/MBASE/TO_FM(10)**10
N = SIZE(A,DIM=1)

ALLOCATE(A1(N,N),A2(N,N),B1(N),R1(N),X1(N),KSWAP(N),STAT=J)
IF (J /= 0) THEN
    WRITE (*,"(/' Error in ZM_INVERSE. Unable to allocate arrays with N = ',I8/)'") N
    STOP
ENDIF

!          Raise precision.

NDSAVE = NDIG
NDIG = 2*NDIG
KWARN_SAVE = KWARN
KWARN = 0

!          Copy A to A1 with higher precision.

110 CALL FM_EQU_R1(TOL,NDSAVE,NDIG)
DO I = 1, N
    DO J = 1, N
        CALL ZM_EQU(A(I,J),A1(I,J),NDSAVE,NDIG)
    ENDDO
ENDDO
A2 = A1

!          Factor A into L*U form.

CALL ZM_FACTOR_LU(A1,N,DET,KSWAP)
IF (DET == 0 .OR. IS_UNKNOWN(DET)) THEN
    IF (KWARN > 0) THEN
        WRITE (*,"(/' Error in ZM_INVERSE. The matrix is singular.'/)'")
    ENDIF
    IF (KWARN >= 2) STOP
    B = TO_ZM(' UNKNOWN ')
    GO TO 120
ENDIF

```

```

!           Solve for the inverse matrix one column at a time.

DO K = 1, N
  B1 = 0
  B1(K) = 1
  CALL ZM_SOLVE_LU(A1,N,B1,X1,KSWAP)

!           Do an iterative refinement.

R1 = MATMUL(A2,X1) - B1

CALL ZM_SOLVE_LU(A1,N,R1,B1,KSWAP)
X1 = X1 - B1

!           Check for accuracy at the user's precision.

IF (SQRT( ABS(DOT_PRODUCT( B1 , B1 )) ) > TOL) THEN
  NDIG = 2*NDIG
  GO TO 110
ENDIF

!           Round the result and store column K in the B matrix.

DO I = 1, N
  CALL ZM_EQU(X1(I),B(I,K),NDIG,NDSAVE)
ENDDO
ENDDO
120 CALL ZMEQU_R1(DET%MZM,NDIG,NDSAVE)

CALL FM_DEALLOCATE(A1)
CALL FM_DEALLOCATE(A2)
CALL FM_DEALLOCATE(B1)
CALL FM_DEALLOCATE(R1)
CALL FM_DEALLOCATE(X1)
DEALLOCATE(A1,A2,B1,R1,X1,KSWAP)

NDIG = NDSAVE
KWARN = KWARN_SAVE
CALL FM_EXIT_USER_ROUTINE
END SUBROUTINE ZM_INVERSE

```