

```

      SUBROUTINE DGETRF( M, N, A, LDA, IPIV, INFO )
*
*  -- LAPACK routine (version 3.2) --
*  -- LAPACK is a software package provided by Univ. of Tennessee,    --
*  -- Univ. of California Berkeley, Univ. of Colorado Denver and NAG Ltd.--
*  November 2006
*
*  .. Scalar Arguments ..
      INTEGER          INFO, LDA, M, N
*  ..
*  .. Array Arguments ..
      INTEGER          IPIV( * )
      DOUBLE PRECISION A( LDA, * )
*  ..
*
* Purpose
* =====
*
* DGETRF computes an LU factorization of a general M-by-N matrix A
* using partial pivoting with row interchanges.
*
* The factorization has the form
*   A = P * L * U
* where P is a permutation matrix, L is lower triangular with unit
* diagonal elements (lower trapezoidal if m > n), and U is upper
* triangular (upper trapezoidal if m < n).
*
* This is the right-looking Level 3 BLAS version of the algorithm.
*
* Arguments
* =====
*
* M      (input) INTEGER
*        The number of rows of the matrix A.  M >= 0.
*
* N      (input) INTEGER
*        The number of columns of the matrix A.  N >= 0.
*
* A      (input/output) DOUBLE PRECISION array, dimension (LDA,N)
*        On entry, the M-by-N matrix to be factored.
*        On exit, the factors L and U from the factorization
*        A = P*L*U; the unit diagonal elements of L are not stored.
*
* LDA    (input) INTEGER
*        The leading dimension of the array A.  LDA >= max(1,M).
*
* IPIV   (output) INTEGER array, dimension (min(M,N))
*        The pivot indices; for 1 <= i <= min(M,N), row i of the
*        matrix was interchanged with row IPIV(i).
*
* INFO   (output) INTEGER
*        = 0: successful exit
*        < 0: if INFO = -i, the i-th argument had an illegal value
*        > 0: if INFO = i, U(i,i) is exactly zero. The factorization
*              has been completed, but the factor U is exactly
*              singular, and division by zero will occur if it is used
*              to solve a system of equations.
*
* =====
*

```

```

*      .. Parameters ..
DOUBLE PRECISION    ONE
PARAMETER            ( ONE = 1.0D+0 )
*
*      ..
*      .. Local Scalars ..
INTEGER              I, IINFO, J, JB, NB
*
*      ..
*      .. External Subroutines ..
EXTERNAL              DGEMM, DGETF2, DLASWP, DTRSM, XERBLA
*
*      ..
*      .. External Functions ..
INTEGER              ILAENV
EXTERNAL              ILAENV
*
*      ..
*      .. Intrinsic Functions ..
INTRINSIC              MAX, MIN
*
*      ..
*      .. Executable Statements ..
*
Test the input parameters.
*
INFO = 0
IF( M.LT.0 ) THEN
    INFO = -1
ELSE IF( N.LT.0 ) THEN
    INFO = -2
ELSE IF( LDA.LT.MAX( 1, M ) ) THEN
    INFO = -4
END IF
IF( INFO.NE.0 ) THEN
    CALL XERBLA( 'DGETRF', -INFO )
    RETURN
END IF
*
Quick return if possible
*
IF( M.EQ.0 .OR. N.EQ.0 )
$   RETURN
*
Determine the block size for this environment.
*
NB = ILAENV( 1, 'DGETRF', ' ', M, N, -1, -1 )
IF( NB.LE.1 .OR. NB.GE.MIN( M, N ) ) THEN
*
    Use unblocked code.
*
    CALL DGETF2( M, N, A, LDA, IPIV, INFO )
ELSE
*
    Use blocked code.
*
    DO 20 J = 1, MIN( M, N ), NB
        JB = MIN( MIN( M, N )-J+1, NB )
*
        Factor diagonal and subdiagonal blocks and test for exact
        singularity.
*
        CALL DGETF2( M-J+1, JB, A( J, J ), LDA, IPIV( J ), IINFO )
*
        Adjust INFO and the pivot indices.
*

```

```

      IF( INFO.EQ.0 .AND. IINFO.GT.0 )
$       INFO = IINFO + J - 1
      DO 10 I = J, MIN( M, J+JB-1 )
        IPIV( I ) = J - 1 + IPIV( I )
10     CONTINUE
*
*       Apply interchanges to columns 1:J-1.
*
      CALL DLASWP( J-1, A, LDA, J, J+JB-1, IPIV, 1 )
*
      IF( J+JB.LE.N ) THEN
*
*       Apply interchanges to columns J+JB:N.
*
      CALL DLASWP( N-J-JB+1, A( 1, J+JB ), LDA, J, J+JB-1,
$       IPIV, 1 )
*
*       Compute block row of U.
*
      CALL DTRSM( 'Left', 'Lower', 'No transpose', 'Unit', JB,
$       N-J-JB+1, ONE, A( J, J ), LDA, A( J, J+JB ),
$       LDA )
      IF( J+JB.LE.M ) THEN
*
*       Update trailing submatrix.
*
      CALL DGEMM( 'No transpose', 'No transpose', M-J-JB+1,
$       N-J-JB+1, JB, -ONE, A( J+JB, J ), LDA,
$       A( J, J+JB ), LDA, ONE, A( J+JB, J+JB ),
$       LDA )
        END IF
      END IF
20    CONTINUE
      END IF
      RETURN
*
*       End of DGETRF
*
      END

```