NAG Toolbox

nag_lapack_dpstrf (f07kd)

1 Purpose

nag_lapack_dpstrf (f07kd) computes the Cholesky factorization with complete pivoting of a real symmetric positive semidefinite matrix.

2 Syntax

```
[a, piv, rank, info] = nag_lapack_dpstrf(uplo, a, 'n', n, 'tol', tol)
[a, piv, rank, info] = f07kd(uplo, a, 'n', n, 'tol', tol)
```

3 Description

nag_lapack_dpstrf (f07kd) forms the Cholesky factorization of a real symmetric positive semidefinite matrix A either as $P^{\mathrm{T}}AP = U^{\mathrm{T}}U$ if **uplo** $=$ 'U' or $P^{\mathrm{T}}AP = LL^{\mathrm{T}}$ if **uplo** $=$ 'L', where P is a permutation matrix, U is an upper triangular matrix and L is lower triangular.

This algorithm does not attempt to check that A is positive semidefinite.

4 References

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

Lucas C (2004) LAPACK-style codes for Level 2 and 3 pivoted Cholesky factorizations *LAPACK Working Note No. 161. Technical Report CS-04-522* Department of Computer Science, University of Tennessee, 107 Ayres Hall, Knoxville, TN 37996-1301, USA http://www.netlib.org/lapack/lawnspdf/lawn161.pdf

5 Parameters

5.1 Compulsory Input Parameters

1: **uplo** – CHARACTER(1)

Specifies whether the upper or lower triangular part of A is stored and how A is to be factorized.

uplo $=$ 'U'
The upper triangular part of A is stored and A is factorized as $U^{\mathrm{T}}U$, where U is upper triangular.

uplo $=$ 'L'
The lower triangular part of A is stored and A is factorized as LL^{T}, where L is lower triangular.

Constraint: **uplo** $=$ 'U' or 'L'.

2: **a**$(lda,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** must be at least $\max(1, \mathbf{n})$.

The second dimension of the array **a** must be at least $\max(1, \mathbf{n})$.

The n by n symmetric positive semidefinite matrix A.

If **uplo** $=$ 'U', the upper triangular part of a must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** $=$ 'L', the lower triangular part of a must be stored and the elements of the array above the diagonal are not referenced.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the array **a**.

n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

2: **tol** – REAL (KIND=nag_wp)

Default: -1

User defined tolerance. If $\mathbf{tol} < 0$, then $n \times \max_{k=1,n} |A_{kk}| \times$ ***machine precision*** will be used. The algorithm terminates at the rth step if the $(r+1)$th step pivot $< \mathbf{tol}$.

5.3 Output Parameters

1: $\mathbf{a}(lda,:)$ – REAL (KIND=nag_wp) array

The first dimension of the array **a** will be $\max(1, \mathbf{n})$.

The second dimension of the array **a** will be $\max(1, \mathbf{n})$.

If **uplo** $=$ 'U', the first **rank** rows of the upper triangle of A are overwritten with the nonzero elements of the Cholesky factor U, and the remaining rows of the triangle are destroyed.

If **uplo** $=$ 'L', the first **rank** columns of the lower triangle of A are overwritten with the nonzero elements of the Cholesky factor L, and the remaining columns of the triangle are destroyed.

2: $\mathbf{piv}(\mathbf{n})$ – INTEGER array

piv is such that the nonzero entries of P are $P(\mathbf{piv}(k), k) = 1$, for $k = 1, 2, \ldots, n$.

3: **rank** – INTEGER

The computed rank of A given by the number of steps the algorithm completed.

4: **info** – INTEGER

$\mathbf{info} = 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

$\mathbf{info} < 0$

If $\mathbf{info} = -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

$\mathbf{info} = 1$ (*warning*)

The matrix A is not positive definite. It is either positive semidefinite with computed rank as returned in **rank** and less than n, or it may be indefinite, see Section 9.

7 Accuracy

If **uplo** $=$ 'L' and **rank** $= r$, the computed Cholesky factor L and permutation matrix P satisfy the following upper bound

$$\frac{\|A - PLL^{\mathrm{T}}P^{\mathrm{T}}\|_2}{\|A\|_2} \leq 2rc(r)\epsilon\big(\|W\|_2 + 1\big)^2 + O\big(\epsilon^2\big),$$

where

$$W = L_{11}^{-1}L_{12}, \quad L = \begin{pmatrix} L_{11} & 0 \\ L_{12} & 0 \end{pmatrix}, \quad L_{11} \in \mathbb{R}^{r\times r},$$

$c(r)$ is a modest linear function of r, ϵ is ***machine precision***, and

$$\|W\|_2 \leq \sqrt{\frac{1}{3}(n-r)(4^r - 1)}.$$

So there is no guarantee of stability of the algorithm for large n and r, although $\|W\|_2$ is generally small in practice.

8 Further Comments

The total number of floating-point operations is approximately $nr^2 - 2/3r^3$, where r is the computed rank of A.

This algorithm does not attempt to check that A is positive semidefinite, and in particular the rank detection criterion in the algorithm is based on A being positive semidefinite. If there is doubt over semidefiniteness then you should use the indefinite factorization nag_lapack_dsytrf (f07md). See Lucas (2004) for further information.

The complex analogue of this function is nag_lapack_zpstrf (f07kr).

9 Example

This example computes the Cholesky factorization of the matrix A, where

$$A = \begin{pmatrix} 2.51 & 4.04 & 3.34 & 1.34 & 1.29 \\ 4.04 & 8.22 & 7.38 & 2.68 & 2.44 \\ 3.34 & 7.38 & 7.06 & 2.24 & 2.14 \\ 1.34 & 2.68 & 2.24 & 0.96 & 0.80 \\ 1.29 & 2.44 & 2.14 & 0.80 & 0.74 \end{pmatrix}.$$

9.1 Program Text

```
     function f07kd_example

fprintf('f07kd example results\n\n');

% Semidefinite matrix A
a = [2.51, 4.04, 3.34, 1.34, 1.29;
     4.04, 8.22, 7.38, 2.68, 2.44;
     3.34, 7.38, 7.06, 2.24, 2.14;
     1.34, 2.68, 2.24, 0.96, 0.80;
     1.29, 2.44, 2.14, 0.80, 0.74];

% Catch warnings about rank defficient matrix ifail=1
wstat = warning();
warning('OFF');

% Factorize a
uplo = 'l';
[afac, piv, rank, info] = f07kd( ...
                                 uplo, a);
```

```
if (info==0 || info==1)
  fprintf('Computed rank: %d\n\n', rank);
  % Zero out columns rank+1 onwards
  afac(:, rank+1:5) = 0;

  [ifail] = x04ca( ...
                   uplo, 'n', afac, 'Factor');

  fprintf('\n piv:\n   ');
  fprintf('%11d', piv);
  fprintf('\n');
else
  fprintf('\nf07kd returned with error, info = %d.\n',info);
end

warning(wstat);
```

9.2 Program Results

```
     f07kd example results

Computed rank: 3

 Factor
             1          2          3          4          5
 1      2.8671
 2      1.4091     0.7242
 3      2.5741    -0.3965     0.5262
 4      0.9348     0.0315    -0.2920     0.0000
 5      0.8510     0.1254    -0.0018     0.0000     0.0000

 piv:
             2          1          3          4          5
```