NAG Toolbox

nag_lapack_dsysv (f07ma)

1 Purpose

nag_lapack_dsysv (f07ma) computes the solution to a real system of linear equations

$$AX = B,$$

where A is an n by n symmetric matrix and X and B are n by r matrices.

2 Syntax

```
[a, ipiv, b, info] = nag_lapack_dsysv(uplo, a, b, 'n', n, 'nrhs_p', nrhs_p)
[a, ipiv, b, info] = f07ma(uplo, a, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

nag_lapack_dsysv (f07ma) uses the diagonal pivoting method to factor A as $A = UDU^{\mathrm{T}}$ if **uplo** = 'U' or $A = LDL^{\mathrm{T}}$ if **uplo** = 'L', where U (or L) is a product of permutation and unit upper (lower) triangular matrices, and D is symmetric and block diagonal with 1 by 1 and 2 by 2 diagonal blocks. The factored form of A is then used to solve the system of equations $AX = B$.

Note that, in general, different permutations (pivot sequences) and diagonal block structures are obtained for **uplo** = 'U' or 'L'

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

5.1 Compulsory Input Parameters

1: **uplo** – CHARACTER(1)

If **uplo** = 'U', the upper triangle of A is stored.

If **uplo** = 'L', the lower triangle of A is stored.

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 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.

3: **b**(ldb, :) – REAL (KIND=nag_wp) array

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

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

The n by r right-hand side matrix B.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the number of linear equations, i.e., the order of the matrix A.

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

2: **nrhs_p** – INTEGER

Default: the second dimension of the array **b**.

r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: $\mathbf{nrhs\_p} \geq 0$.

5.3 Output Parameters

1: **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 $\mathbf{info} = 0$, the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization $A = UDU^{\mathrm{T}}$ or $A = LDL^{\mathrm{T}}$ as computed by nag_lapack_dsytrf (f07md).

2: **ipiv**(:) – INTEGER array

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

Details of the interchanges and the block structure of D. More precisely,

if $\mathbf{ipiv}(i) = k > 0$, d_{ii} is a 1 by 1 pivot block and the ith row and column of A were interchanged with the kth row and column;

if **uplo** = 'U' and $\mathbf{ipiv}(i-1) = \mathbf{ipiv}(i) = -l < 0$, $\begin{pmatrix} d_{i-1,i-1} & \bar{d}_{i,i-1} \\ \bar{d}_{i,i-1} & d_{ii} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i-1)$th row and column of A were interchanged with the lth row and column;

if **uplo** = 'L' and $\mathbf{ipiv}(i) = \mathbf{ipiv}(i+1) = -m < 0$, $\begin{pmatrix} d_{ii} & d_{i+1,i} \\ d_{i+1,i} & d_{i+1,i+1} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i+1)$th row and column of A were interchanged with the mth row and column.

3: **b**(ldb, :) – REAL (KIND=nag_wp) array

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

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

If $\mathbf{info} = 0$, the n by r solution matrix X.

4: **info** – INTEGER

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

6 Error Indicators and Warnings

info < 0

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

info > 0 (*warning*)

Element $\langle$*value*$\rangle$ of the diagonal is exactly zero. The factorization has been completed, but the block diagonal matrix D is exactly singular, so the solution could not be computed.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \|A^{-1}\|_1\|A\|_1$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* (1999) for further details.

nag_lapack_dsysvx (f07mb) is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, nag_linsys_real_symm_solve (f04bh) solves $Ax = b$ and returns a forward error bound and condition estimate. nag_linsys_real_symm_solve (f04bh) calls nag_lapack_dsysv (f07ma) to solve the equations.

8 Further Comments

The total number of floating-point operations is approximately $\frac{1}{3}n^3 + 2n^2r$, where r is the number of right-hand sides.

The complex analogues of nag_lapack_dsysv (f07ma) are nag_lapack_zhesv (f07mn) for Hermitian matrices, and nag_lapack_zsysv (f07nn) for symmetric matrices.

9 Example

This example solves the equations

$$Ax = b,$$

where A is the symmetric matrix

$$A = \begin{pmatrix} -1.81 & 2.06 & 0.63 & -1.15 \\ 2.06 & 1.15 & 1.87 & 4.20 \\ 0.63 & 1.87 & -0.21 & 3.87 \\ -1.15 & 4.20 & 3.87 & 2.07 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0.96 \\ 6.07 \\ 8.38 \\ 9.50 \end{pmatrix}.$$

Details of the factorization of A are also output.

9.1 Program Text

```
function f07ma_example

fprintf('f07ma example results\n\n');

% Indefinite matrix A
uplo = 'Upper';
a = [-1.81, 2.06,  0.63, -1.15;
      0,    1.15,  1.87,  4.20;
      0,    0,    -0.21,  3.87;
      0,    0,     0,     2.07];

% RHS
b = [0.96;
     6.07;
     8.38;
     9.50];

% Solve
[af, ipiv, x, info] = f07ma( ...
                             uplo, a, b);

disp('Solution');
disp(x');

[ifail] = x04ca( ...
                 uplo, 'Non-unit', af, 'Details of factorization');

fprintf('\nPivot indices\n   ');
fprintf('%11d', ipiv);
fprintf('\n');
```

9.2 Program Results

```
f07ma example results

Solution
   -5.0000   -2.0000    1.0000    4.0000

 Details of factorization
             1          2          3          4
 1      0.4074     0.3031    -0.5960     0.6537
 2                -2.5907     0.8115     0.2230
 3                            1.1500     4.2000
 4                                       2.0700

Pivot indices
             1          2         -2         -2
```