NAG Toolbox

nag_linsys_real_posdef_vband_solve (f04mc)

1 Purpose

nag_linsys_real_posdef_vband_solve (f04mc) computes the approximate solution of a system of real linear equations with multiple right-hand sides, $AX = B$, where A is a symmetric positive definite variable-bandwidth matrix, which has previously been factorized by nag_matop_real_vband_posdef_fac (f01mc). Related systems may also be solved.

2 Syntax

```
[x, ifail] = nag_linsys_real_posdef_vband_solve(al, d, nrow, b, iselct, 'n', n,
'lal', lal, 'ir', ir)
```

```
[x, ifail] = f04mc(al, d, nrow, b, iselct, 'n', n, 'lal', lal, 'ir', ir)
```

3 Description

The normal use of this function is the solution of the systems $AX = B$, following a call of nag_matop_real_vband_posdef_fac (f01mc) to determine the Cholesky factorization $A = LDL^{\mathrm{T}}$ of the symmetric positive definite variable-bandwidth matrix A.

However, the function may be used to solve any one of the following systems of linear algebraic equations:

1. $LDL^{\mathrm{T}}X = B$ (usual system),
2. $LDX = B$ (lower triangular system),
3. $DL^{\mathrm{T}}X = B$ (upper triangular system),
4. $LL^{\mathrm{T}}X = B$
5. $LX = B$ (unit lower triangular system),
6. $L^{\mathrm{T}}X = B$ (unit upper triangular system).

L denotes a unit lower triangular variable-bandwidth matrix of order n, D a diagonal matrix of order n, and B a set of right-hand sides.

The matrix L is represented by the elements lying within its **envelope**, i.e., between the first nonzero of each row and the diagonal (see Section 10 for an example). The width $\mathbf{nrow}(i)$ of the ith row is the number of elements between the first nonzero element and the element on the diagonal inclusive.

4 References

Wilkinson J H and Reinsch C (1971) *Handbook for Automatic Computation II, Linear Algebra* Springer–Verlag

5 Parameters

5.1 Compulsory Input Parameters

1: **al**(**lal**) – REAL (KIND=nag_wp) array

The elements within the envelope of the lower triangular matrix L, taken in row by row order, as returned by nag_matop_real_vband_posdef_fac (f01mc). The unit diagonal elements of L must be stored explicitly.

2: **d**(:) – REAL (KIND=nag_wp) array

The dimension of the array **d** must be at least 1 if **iselct** ≥ 4, and at least **n** otherwise

The diagonal elements of the diagonal matrix D. **d** is not referenced if **iselct** ≥ 4.

3: **nrow**(**n**) – INTEGER array

nrow(i) must contain the width of row i of L, i.e., the number of elements between the first (leftmost) nonzero element and the element on the diagonal, inclusive.

Constraint: $1 \leq$ **nrow**(i) $\leq i$.

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

ldb, the first dimension of the array, must satisfy the constraint $ldb \geq$ **n**.

The n by r right-hand side matrix B. See also Section 9.

5: **iselct** – INTEGER

Must specify the type of system to be solved, as follows:

iselct $= 1$
Solve $LDL^{\mathrm{T}}X = B$.

iselct $= 2$
Solve $LDX = B$.

iselct $= 3$
Solve $DL^{\mathrm{T}}X = B$.

iselct $= 4$
Solve $LL^{\mathrm{T}}X = B$.

iselct $= 5$
Solve $LX = B$.

iselct $= 6$
Solve $L^{\mathrm{T}}X = B$.

Constraint: **iselct** $= 1$, 2, 3, 4, 5 or 6.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **nrow** and the first dimension of the array **b**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix L.

Constraint: **n** ≥ 1.

2: **lal** – INTEGER

Default: the dimension of the array **al**.

The dimension of the array **al**.

Constraint: **lal** $\geq$ **nrow**$(1) +$ **nrow**$(2) + \ldots +$ **nrow**(n).

3: **ir** – INTEGER

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

r, the number of right-hand sides.

Constraint: **ir** ≥ 1.

5.3 Output Parameters

1: **x**(ldx, **ir**) – REAL (KIND=nag_wp) array

The n by r solution matrix X. See also Section 9.

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **n** < 1,
or for some i, **nrow**$(i) < 1$ or **nrow**$(i) > i$,
or **lal** $<$ **nrow**$(1) +$ **nrow**$(2) + \cdots +$ **nrow**(**n**).

ifail $= 2$

On entry, **ir** < 1,
or $ldb <$ **n**,
or $ldx <$ **n**.

ifail $= 3$

On entry, **iselct** < 1,
or **iselct** > 6.

ifail $= 4$

The diagonal matrix D is singular, i.e., at least one of the elements of **d** is zero. This can only occur if **iselct** ≤ 3.

ifail $= 5$

At least one of the diagonal elements of L is not equal to unity.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The usual backward error analysis of the solution of triangular system applies: each computed solution vector is exact for slightly perturbed matrices L and D, as appropriate (see pages 25–27 and 54–55 of Wilkinson and Reinsch (1971)).

8 Further Comments

The time taken by nag_linsys_real_posdef_vband_solve (f04mc) is approximately proportional to pr, where $p =$ **nrow**$(1) +$ **nrow**$(2) + \cdots +$ **nrow**(n).

9 Example

This example solves the system of equations $AX = B$, where

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 & 5 & 0 \\ 2 & 5 & 3 & 0 & 14 & 0 \\ 0 & 3 & 13 & 0 & 18 & 0 \\ 0 & 0 & 0 & 16 & 8 & 24 \\ 5 & 14 & 18 & 8 & 55 & 17 \\ 0 & 0 & 0 & 24 & 17 & 77 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 6 & -10 \\ 15 & -21 \\ 11 & -3 \\ 0 & 24 \\ 51 & -39 \\ 46 & 67 \end{pmatrix}$$

Here A is symmetric and positive definite and must first be factorized by nag_matop_real_vband_posdef_fac (f01mc).

9.1 Program Text

```
function f04mc_example

fprintf('f04mc example results\n\n');

a = [1;
     2;     5;
            3;     13;
                           16;
     5;    14;     18;      8;     55;
                   24;     17;     77];
nrow = [nag_int(1);  2; 2; 1; 5; 3];

% Factorize
[L, D, ifail] = f01mc(a, nrow);

% Solve Ax = b
b = [  6.0 -10.0;
      15.0 -21.0;
      11.0  -3.0;
       0.0  24.0;
      51.0 -39.0;
      46.0  67.0];
iselct = nag_int(1);
[x, ifail] = f04mc(L, D, nrow, b, iselct);

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

9.2 Program Results

```
     f04mc example results

Solution
    -3     4
     2    -2
    -1     3
    -2     1
     1    -2
     1     1
```