NAG Toolbox

nag_matop_real_symm_posdef_inv (f01ab)

1 Purpose

nag_matop_real_symm_posdef_inv (f01ab) calculates the accurate inverse of a real symmetric positive definite matrix, using a Cholesky factorization and iterative refinement.

2 Syntax

```
[a, b, ifail] = nag_matop_real_symm_posdef_inv(a, 'n', n)
[a, b, ifail] = f01ab(a, 'n', n)
```

3 Description

To compute the inverse X of a real symmetric positive definite matrix A, nag_matop_real_symm_posdef_inv (f01ab) first computes a Cholesky factorization of A as $A = LL^{\mathrm{T}}$, where L is lower triangular. An approximation to X is found by computing L^{-1} and then the product $L^{-\mathrm{T}}L^{-1}$. The residual matrix $R = I - AX$ is calculated using ***additional precision***, and a correction D to X is found by solving $LL^{\mathrm{T}}D = R$. X is replaced by $X + D$, and this iterative refinement of the inverse is repeated until full machine accuracy has been obtained.

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

lda, the first dimension of the array, must satisfy the constraint $lda \geq \mathbf{n} + 1$.

The upper triangle of the n by n positive definite symmetric matrix A. The elements of the array below the diagonal need not be set.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 1$.

5.3 Output Parameters

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

The lower triangle of the inverse matrix X is stored in the elements of the array below the diagonal, in rows 2 to $n + 1$; x_{ij} is stored in $\mathbf{a}(i + 1, j)$ for $i \geq j$. The upper triangle of the original matrix is unchanged.

2: **b**$(ldb, \mathbf{n})$ – REAL (KIND=nag_wp) array

The lower triangle of the inverse matrix X, with x_{ij} stored in **b**(i, j), for $i \geq j$.

3: **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 $= 1$

The matrix A is not positive definite, possibly due to rounding errors.

ifail $= 2$

The refinement process fails to converge, i.e., the matrix A is ill-conditioned.

ifail $= 3$

n < 1, or $lda <$ **n** $+ 1$, or $ldb <$ **n**.

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 computed inverse should be correct to full machine accuracy. For a detailed error analysis see page 40 of Wilkinson and Reinsch (1971).

8 Further Comments

The time taken by nag_matop_real_symm_posdef_inv (f01ab) is approximately proportional to n^3.

9 Example

This example finds the inverse of the 4 by 4 matrix:

$$\begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix}.$$

9.1 Program Text

```
function f01ab_example

fprintf('f01ab example results\n\n');

a = [  5,  7,  6,  5;
       7, 10,  8,  7;
       6,  8, 10,  9;
```

```
       5,  7,  9, 10];

% add row for storing updates.
a = [a; 0 0 0 0];

[X, L, ifail] = f01ab(a);

matrix = 'Lower';
diag   = 'Non-unit';
xtitl  = 'Lower triangle of inverse:';
[ifail] = x04ca( ...
                matrix, diag, L, xtitl);
```

9.2 Program Results

```
     f01ab example results

 Lower triangle of inverse:
             1          2          3          4
 1     68.0000
 2    -41.0000    25.0000
 3    -17.0000    10.0000     5.0000
 4     10.0000    -6.0000    -3.0000     2.0000
```