NAG Toolbox

nag_linsys_real_square_solve_1rhs (f04at)

1 Purpose

nag_linsys_real_square_solve_1rhs (f04at) calculates the accurate solution of a set of real linear equations with a single right-hand side, using an LU factorization with partial pivoting, and iterative refinement.

2 Syntax

```
[c, aa, ifail] = nag_linsys_real_square_solve_1rhs(a, b, 'n', n)
[c, aa, ifail] = f04at(a, b, 'n', n)
```

3 Description

Given a set of real linear equations, $Ax = b$, the function first computes an LU factorization of A with partial pivoting, $PA = LU$, where P is a permutation matrix, L is lower triangular and U is unit upper triangular. An approximation to x is found by forward and backward substitution in $Ly = Pb$ and $Ux = y$. The residual vector $r = b - Ax$ is then calculated using ***additional precision***, and a correction d to x is found by solving $LUd = r$. x is replaced by $x + d$, and this iterative refinement of the solution is repeated until full machine accuracy is 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**$(lda, :)$ – 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 matrix A.

2: **b**$(:)$ – REAL (KIND=nag_wp) array

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

The right-hand side vector b.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the order of the matrix A.

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

5.3 Output Parameters

1: **c**(**n**) – REAL (KIND=nag_wp) array

The solution vector x.

2: **aa**($ldaa$, **n**) – REAL (KIND=nag_wp) array

The first dimension of the array **aa** will be max(1, **n**).

The second dimension of the array **aa** will be max(1, **n**).

The triangular factors L and U, except that the unit diagonal elements of U are not stored.

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 singular, possibly due to rounding errors.

ifail $= 2$

Iterative refinement fails to improve the solution, i.e., the matrix A is too ill-conditioned.

ifail $= 3$

On entry, **n** < 0,
or $ldaa < \max(1, \mathbf{n})$,
or $ldaa < \max(1, \mathbf{n})$.

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 solutions should be correct to full machine accuracy. For a detailed error analysis see page 107 of Wilkinson and Reinsch (1971).

8 Further Comments

The time taken by nag_linsys_real_square_solve_1rhs (f04at) is approximately proportional to n^3.

The function **must not** be called with the same name for arguments **b** and **c**.

9 Example

This example solves the set of linear equations $Ax = b$ where

$$A = \begin{pmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} -359 \\ 281 \\ 85 \end{pmatrix}.$$

9.1 Program Text

```
     function f04at_example

fprintf('f04at example results\n\n');

% Accurate solution to Ax = b, for general A
a = [  33,  16,  72;
      -24, -10, -57;
       -8,  -4, -17];
b = [-359;
      281;
       85];

[x, LU, ifail] = f04at(a, b);

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

9.2 Program Results

```
     f04at example results

Solution
     1
    -2
    -5
```