NAG Library Routine Document

F07ACF (DSGESV)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F07ACF (DSGESV) computes the solution to a real system of linear equations

$$AX = B,$$

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

2 Specification

```
SUBROUTINE F07ACF (N, NRHS, A, LDA, IPIV, B, LDB, X, LDX, WORK, SWORK,          &
                   ITER, INFO)

INTEGER            N, NRHS, LDA, IPIV(N), LDB, LDX, ITER, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*), X(LDX,*), WORK(N*NRHS)
REAL (KIND=nag_rp) SWORK(N*(N+NRHS))
```

The routine may be called by its LAPACK name ***dsgesv***.

3 Description

F07ACF (DSGESV) first attempts to factorize the matrix in single precision and use this factorization within an iterative refinement procedure to produce a solution with full double precision accuracy. If the approach fails the method switches to a double precision factorization and solve.

The iterative refinement process is stopped if

$$\text{ITER} > \textit{itermax},$$

where ITER is the number of iterations carried out thus far and *itermax* is the maximum number of iterations allowed, which is fixed at 30 iterations. The process is also stopped if for all right-hand sides we have

$$\|resid\| < \sqrt{\text{N}}\|x\|\|A\|\epsilon,$$

where $\|resid\|$ is the ∞-norm of the residual, $\|x\|$ is the ∞-norm of the solution, $\|A\|$ is the ∞-operator-norm of the matrix A and ϵ is the ***machine precision*** returned by X02AJF.

The iterative refinement strategy used by F07ACF (DSGESV) can be more efficient than the corresponding direct full precision algorithm. Since this strategy must perform iterative refinement on each right-hand side, any efficiency gains will reduce as the number of right-hand sides increases. Conversely, as the matrix size increases the cost of these iterative refinements become less significant relative to the cost of factorization. Thus, any efficiency gains will be greatest for a very small number of right-hand sides and for large matrix sizes. The cut-off values for the number of right-hand sides and matrix size, for which the iterative refinement strategy performs better, depends on the relative performance of the reduced and full precision factorization and back-substitution. For now, F07ACF (DSGESV) always attempts the iterative refinement strategy first; you are advised to compare the performance of F07ACF (DSGESV) with that of its full precision counterpart F07AAF (DGESV) to determine whether this strategy is worthwhile for your particular problem dimensions.

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

Buttari A, Dongarra J, Langou J, Langou J, Luszczek P and Kurzak J (2007) Mixed precision iterative refinement techniques for the solution of dense linear systems *International Journal of High Performance Computing Applications*

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

5 Parameters

1: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

2: NRHS – INTEGER *Input*

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

Constraint: $\mathrm{NRHS} \geq 0$.

3: A(LDA,$*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1,\mathrm{N})$.

On entry: the n by n coefficient matrix A.

On exit: if iterative refinement has been successfully used (i.e., if $\mathrm{INFO} = 0$ and $\mathrm{ITER} \geq 0$), then A is unchanged. If double precision factorization has been used (when $\mathrm{INFO} = 0$ and $\mathrm{ITER} < 0$), A contains the factors L and U from the factorization $A = PLU$; the unit diagonal elements of L are not stored.

4: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07ACF (DSGESV) is called.

Constraint: $\mathrm{LDA} \geq \max(1,\mathrm{N})$.

5: IPIV(N) – INTEGER array *Output*

On exit: if no constraints are violated, the pivot indices that define the permutation matrix P; at the ith step row i of the matrix was interchanged with row $\mathrm{IPIV}(i)$. $\mathrm{IPIV}(i) = i$ indicates a row interchange was not required. IPIV corresponds either to the single precision factorization (if $\mathrm{INFO} = 0$ and $\mathrm{ITER} \geq 0$) or to the double precision factorization (if $\mathrm{INFO} = 0$ and $\mathrm{ITER} < 0$).

6: B(LDB,$*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array B must be at least $\max(1,\mathrm{NRHS})$.

On entry: the n by r right-hand side matrix B.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07ACF (DSGESV) is called.

Constraint: $\mathrm{LDB} \geq \max(1,\mathrm{N})$.

8: X(LDX,$*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array X must be at least $\max(1, \mathrm{NRHS})$.

On exit: if INFO $= 0$, the n by r solution matrix X.

9: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07ACF (DSGESV) is called.

Constraint: LDX $\geq \max(1, \mathrm{N})$.

10: WORK(N $*$ NRHS) – REAL (KIND=nag_wp) array *Workspace*

11: SWORK(N $\times$ (N $+$ NRHS)) – REAL (KIND=nag_rp) array *Workspace*

Note: this array is utilized in the reduced precision computation, consequently its type nag_rp reflects this usage.

12: ITER – INTEGER *Output*

On exit: if ITER > 0, iterative refinement has been successfully used and ITER is the number of iterations carried out.

If ITER < 0, iterative refinement has failed for one of the reasons given below and double precision factorization has been carried out instead.

ITER $= -1$

Taking into account machine parameters, and the values of N and NRHS, it is not worth working in single precision.

ITER $= -2$

Overflow of an entry occurred when moving from double to single precision.

ITER $= -3$

An intermediate single precision factorization failed.

ITER $= -31$

The maximum permitted number of iterations was exceeded.

13: INFO – INTEGER *Output*

On exit: INFO $= 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

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

INFO > 0

If INFO $= i$, u_{ii} is exactly zero. The factorization has been completed, but the factor U is exactly singular, so the solution could not be computed.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies the 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 $\kappa(A) = \left\|A^{-1}\right\|_1 \|A\|_1$, 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.

8 Further Comments

The complex analogue of this routine is F07AQF (ZCGESV).

9 Example

This example solves the equations

$$Ax = b,$$

where A is the general matrix

$$A = \begin{pmatrix} 1.80 & 2.88 & 2.05 & -0.89 \\ 5.25 & -2.95 & -0.95 & -3.80 \\ 1.58 & -2.69 & -2.90 & -1.04 \\ -1.11 & -0.66 & -0.59 & 0.80 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 9.52 \\ 24.35 \\ 0.77 \\ -6.22 \end{pmatrix}.$$

9.1 Program Text

```
    Program f07acfe

!     F07ACF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dsgesv, nag_rp, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, info, iter, lda, ldb, ldx, n, r
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:,:), work(:), x(:,:)
      Real (Kind=nag_rp), Allocatable  :: swork(:)
      Integer, Allocatable             :: ipiv(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07ACF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, r
      lda = n
      ldb = n
      ldx = n
      Allocate (a(lda,n),b(n,r),work(n*r),x(ldx,r),swork(n*(n+r)),ipiv(n))

!     Read A and B from data file

      Read (nin,*)(a(i,1:n),i=1,n)
      Read (nin,*)(b(i,1:r),i=1,n)

!     Solve the equations Ax = b for x
!     The NAG name equivalent of dsgesv is f07acf
      Call dsgesv(n,r,a,lda,ipiv,b,ldb,x,ldx,work,swork,iter,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Solution'
        Write (nout,99999)(x(i,1:r),i=1,n)

!       Print pivot indices

        Write (nout,*)
        Write (nout,*) 'Pivot indices'
        Write (nout,99998) ipiv(1:n)

      Else
        Write (nout,99997) 'The (', info, ',', info, ')', &
          ' element of the factor U is zero'
      End If

99999 Format ((3X,7F11.4))
99998 Format ((3X,7I11))
99997 Format (1X,A,I3,A,I3,A,A)
    End Program f07acfe
```

9.2 Program Data

```
F07ACF Example Program Data

  4     1                     :Value of N, R

  1.80   2.88   2.05  -0.89
  5.25  -2.95  -0.95  -3.80
  1.58  -2.69  -2.90  -1.04
 -1.11  -0.66  -0.59   0.80   :End of matrix A

  9.52  24.35   0.77  -6.22   :End of vector b
```

9.3 Program Results

```
 F07ACF Example Program Results

 Solution
        1.0000    -1.0000     3.0000    -5.0000

 Pivot indices
             2          2          3          4
```