NAG Library Routine Document

F07APF (ZGESVX)

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

F07APF (ZGESVX) uses the LU factorization to compute the solution to a complex system of linear equations

$$AX = B \qquad \text{or} \qquad A^{\mathrm{T}}X = B \qquad \text{or} \qquad A^{\mathrm{H}}X = B,$$

where A is an n by n matrix and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Specification

```
SUBROUTINE F07APF (FACT, TRANS, N, NRHS, A, LDA, AF, LDAF, IPIV, EQUED, R,      &
                   C, B, LDB, X, LDX, RCOND, FERR, BERR, WORK, RWORK, INFO)

INTEGER                 N, NRHS, LDA, LDAF, IPIV(*), LDB, LDX, INFO
REAL (KIND=nag_wp)      R(*), C(*), RCOND, FERR(NRHS), BERR(NRHS),              &
                        RWORK(max(1,2*N))
COMPLEX (KIND=nag_wp)   A(LDA,*), AF(LDAF,*), B(LDB,*), X(LDX,*), WORK(2*N)
CHARACTER(1)            FACT, TRANS, EQUED
```

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

3 Description

F07APF (ZGESVX) performs the following steps:

1. **Equilibration**

   The linear system to be solved may be badly scaled. However, the system can be equilibrated as a first stage by setting FACT = 'E'. In this case, real scaling factors are computed and these factors then determine whether the system is to be equilibrated. Equilibrated forms of the systems $AX = B$, $A^{\mathrm{T}}X = B$ and $A^{\mathrm{H}}X = B$ are

   $$(D_R A D_C)\left(D_C^{-1}X\right) = D_R B,$$

   $$(D_R A D_C)^{\mathrm{T}}\left(D_R^{-1}X\right) = D_C B,$$

   and

   $$(D_R A D_C)^{\mathrm{H}}\left(D_R^{-1}X\right) = D_C B,$$

   respectively, where D_R and D_C are diagonal matrices, with positive diagonal elements, formed from the computed scaling factors.

   When equilibration is used, A will be overwritten by $D_R A D_C$ and B will be overwritten by $D_R B$ (or $D_C B$ when the solution of $A^{\mathrm{T}}X = B$ or $A^{\mathrm{H}}X = B$ is sought).

2. **Factorization**

   The matrix A, or its scaled form, is copied and factored using the LU decomposition

   $$A = PLU,$$

   where P is a permutation matrix, L is a unit lower triangular matrix, and U is upper triangular.

This stage can be by-passed when a factored matrix (with scaled matrices and scaling factors) are supplied; for example, as provided by a previous call to F07APF (ZGESVX) with the same matrix A.

3. **Condition Number Estimation**

   The LU factorization of A determines whether a solution to the linear system exists. If some diagonal element of U is zero, then U is exactly singular, no solution exists and the routine returns with a failure. Otherwise the factorized form of A is used to estimate the condition number of the matrix A. If the reciprocal of the condition number is less than ***machine precision*** then a warning code is returned on final exit.

4. **Solution**

   The (equilibrated) system is solved for X ($D_C^{-1}X$ or $D_R^{-1}X$) using the factored form of A (D_RAD_C).

5. **Iterative Refinement**

   Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for the computed solution.

6. **Construct Solution Matrix *X***

   If equilibration was used, the matrix X is premultiplied by D_C (if TRANS = 'N') or D_R (if TRANS = 'T' or 'C') so that it solves the original system before equilibration.

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

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

5 Parameters

1: FACT – CHARACTER(1) *Input*

On entry: specifies whether or not the factorized form of the matrix A is supplied on entry, and if not, whether the matrix A should be equilibrated before it is factorized.

FACT = 'F'
: AF and IPIV contain the factorized form of A. If EQUED $\neq$ 'N', the matrix A has been equilibrated with scaling factors given by R and C. A, AF and IPIV are not modified.

FACT = 'N'
: The matrix A will be copied to AF and factorized.

FACT = 'E'
: The matrix A will be equilibrated if necessary, then copied to AF and factorized.

Constraint: FACT = 'F', 'N' or 'E'.

2: TRANS – CHARACTER(1) *Input*

On entry: specifies the form of the system of equations.

TRANS = 'N'
: $AX = B$ (No transpose).

TRANS = 'T'
: $A^{\mathrm{T}}X = B$ (Transpose).

TRANS = 'C'
$A^{\mathrm{H}}X = B$ (Conjugate transpose).

Constraint: TRANS = 'N', 'T' or 'C'.

3: N – INTEGER *Input*

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

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

4: NRHS – INTEGER *Input*

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

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

5: A(LDA,$*$) – COMPLEX (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 matrix A.

If FACT = 'F' and EQUED $\neq$ 'N', A must have been equilibrated by the scaling factors in R and/or C.

On exit: if FACT = 'F' or 'N', or if FACT = 'E' and EQUED = 'N', A is not modified.

If FACT = 'E' or EQUED $\neq$ 'N', A is scaled as follows:

if EQUED = 'R', $A = D_R A$;

if EQUED = 'C', $A = A D_C$;

if EQUED = 'B', $A = D_R A D_C$.

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07APF (ZGESVX) is called.

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

7: AF(LDAF,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: if FACT = 'F', AF contains the factors L and U from the factorization $A = PLU$ as computed by F07ARF (ZGETRF). If EQUED $\neq$ 'N', AF is the factorized form of the equilibrated matrix A.

If FACT = 'N' or 'E', AF need not be set.

On exit: if FACT = 'N', AF returns the factors L and U from the factorization $A = PLU$ of the original matrix A.

If FACT = 'E', AF returns the factors L and U from the factorization $A = PLU$ of the equilibrated matrix A (see the description of A for the form of the equilibrated matrix).

If FACT = 'F', AF is unchanged from entry.

8: LDAF – INTEGER *Input*

On entry: the first dimension of the array AF as declared in the (sub)program from which F07APF (ZGESVX) is called.

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

9: IPIV($*$) – INTEGER array *Input/Output*

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

On entry: if FACT = 'F', IPIV contains the pivot indices from the factorization $A = PLU$ as computed by F07ARF (ZGETRF); at the ith step row i of the matrix was interchanged with row IPIV(i). IPIV(i) $= i$ indicates a row interchange was not required.

If FACT = 'N' or 'E', IPIV need not be set.

On exit: if FACT = 'N', IPIV contains the pivot indices from the factorization $A = PLU$ of the original matrix A.

If FACT = 'E', IPIV contains the pivot indices from the factorization $A = PLU$ of the equilibrated matrix A.

If FACT = 'F', IPIV is unchanged from entry.

10: EQUED – CHARACTER(1) *Input/Output*

On entry: if FACT = 'N' or 'E', EQUED need not be set.

If FACT = 'F', EQUED must specify the form of the equilibration that was performed as follows:

if EQUED = 'N', no equilibration;

if EQUED = 'R', row equilibration, i.e., A has been premultiplied by D_R;

if EQUED = 'C', column equilibration, i.e., A has been postmultiplied by D_C;

if EQUED = 'B', both row and column equilibration, i.e., A has been replaced by $D_R A D_C$.

On exit: if FACT = 'F', EQUED is unchanged from entry.

Otherwise, if no constraints are violated, EQUED specifies the form of equilibration that was performed as specified above.

Constraint: if FACT = 'F', EQUED = 'N', 'R', 'C' or 'B'.

11: R($*$) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: if FACT = 'N' or 'E', R need not be set.

If FACT = 'F' and EQUED = 'R' or 'B', R must contain the row scale factors for A, D_R; each element of R must be positive.

On exit: if FACT = 'F', R is unchanged from entry.

Otherwise, if no constraints are violated and EQUED = 'R' or 'B', R contains the row scale factors for A, D_R, such that A is multiplied on the left by D_R; each element of R is positive.

12: C($*$) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: if FACT = 'N' or 'E', C need not be set.

If FACT = 'F' or EQUED = 'C' or 'B', C must contain the column scale factors for A, D_C; each element of C must be positive.

On exit: if FACT = 'F', C is unchanged from entry.

Otherwise, if no constraints are violated and EQUED = 'C' or 'B', C contains the row scale factors for A, D_C; each element of C is positive.

13: B(LDB,$*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

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.

On exit: if EQUED = 'N', B is not modified.

If TRANS = 'N' and EQUED = 'R' or 'B', B is overwritten by $D_R B$.

If TRANS = 'T' or 'C' and EQUED = 'C' or 'B', B is overwritten by $D_C B$.

14: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07APF (ZGESVX) is called.

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

15: X(LDX,$*$) – COMPLEX (KIND=nag_wp) array *Output*

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

On exit: if INFO $= 0$ or N $+ 1$, the n by r solution matrix X to the original system of equations. Note that the arrays A and B are modified on exit if EQUED $\neq$ 'N', and the solution to the equilibrated system is $D_C^{-1}X$ if TRANS = 'N' and EQUED = 'C' or 'B', or $D_R^{-1}X$ if TRANS = 'T' or 'C' and EQUED = 'R' or 'B'.

16: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07APF (ZGESVX) is called.

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

17: RCOND – REAL (KIND=nag_wp) *Output*

On exit: if no constraints are violated, an estimate of the reciprocal condition number of the matrix A (after equilibration if that is performed), computed as RCOND $= 1.0/\left(\|A\|_1 \|A^{-1}\|_1\right)$.

18: FERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$ or N $+ 1$, an estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq \text{FERR}(j)$ where $\hat{x}_j$ is the jth column of the computed solution returned in the array X and x_j is the corresponding column of the exact solution X. The estimate is as reliable as the estimate for RCOND, and is almost always a slight overestimate of the true error.

19: BERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$ or N $+ 1$, an estimate of the component-wise relative backward error of each computed solution vector $\hat{x}_j$ (i.e., the smallest relative change in any element of A or B that makes $\hat{x}_j$ an exact solution).

20: WORK($2 \times \text{N}$) – COMPLEX (KIND=nag_wp) array *Workspace*

21: RWORK($\max(1, 2 \times \text{N})$) – REAL (KIND=nag_wp) array *Output*

On exit: RWORK(1) contains the reciprocal pivot growth factor $\|A\|/\|U\|$. The 'max absolute element' norm is used. If RWORK(1) is much less than 1, then the stability of the LU factorization of the (equilibrated) matrix A could be poor. This also means that the solution X, condition estimator RCOND, and forward error bound FERR could be unreliable. If factorization fails with INFO > 0 and INFO $\leq$ N, then RWORK(1) contains the reciprocal pivot growth factor for the leading INFO columns of A.

22: 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 and INFO $\leq$ N

If INFO $= i$, u_{ii} is exactly zero. The factorization has been completed, but the factor U is exactly singular, so the solution and error bounds could not be computed. RCOND $= 0.0$ is returned.

INFO $=$ N $+ 1$

The triangular matrix U is nonsingular, but RCOND is less than ***machine precision***, meaning that the matrix is singular to working precision. Nevertheless, the solution and error bounds are computed because there are a number of situations where the computed solution can be more accurate than the value of RCOND would suggest.

7 Accuracy

For each right-hand side vector b, the computed solution $\hat{x}$ is the exact solution of a perturbed system of equations $(A + E)\hat{x} = b$, where

$$|E| \leq c(n)\epsilon P|L||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. See Section 9.3 of Higham (2002) for further details.

If x is the true solution, then the computed solution $\hat{x}$ satisfies a forward error bound of the form

$$\frac{\|x - \hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq w_c \operatorname{cond}(A, \hat{x}, b)$$

where $\operatorname{cond}(A, \hat{x}, b) = \left\| |A^{-1}|(|A||\hat{x}| + |b|) \right\|_\infty / \|\hat{x}\|_\infty \leq \operatorname{cond}(A) = \left\| |A^{-1}||A| \right\|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the jth column of X, then w_c is returned in BERR(j) and a bound on $\|x - \hat{x}\|_\infty / \|\hat{x}\|_\infty$ is returned in FERR(j). See Section 4.4 of Anderson *et al.* (1999) for further details.

8 Further Comments

The factorization of A requires approximately $\frac{8}{3}n^3$ floating point operations.

Estimating the forward error involves solving a number of systems of linear equations of the form $Ax = b$ or $A^{\mathrm{T}}x = b$; the number is usually 4 or 5 and never more than 11. Each solution involves approximately $8n^2$ operations.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham (2002) for further details.

The real analogue of this routine is F07ABF (DGESVX).

9 Example

This example solves the equations

$$AX = B,$$

where A is the general matrix

$$A = \begin{pmatrix} -1.34 + \ 2.55i & 0.28 + 3.17i & -6.39 - \ 2.20i & 0.72 - \ 0.92i \\ -1.70 - 14.10i & 33.10 - 1.50i & -1.50 + 13.40i & 12.90 + 13.80i \\ -3.29 - \ 2.39i & -1.91 + 4.42i & -0.14 - \ 1.35i & 1.72 + \ 1.35i \\ 2.41 + \ 0.39i & -0.56 + 1.47i & -0.83 - \ 0.69i & -1.96 + \ 0.67i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 26.26 + 51.78i & 31.32 - \ 6.70i \\ 64.30 - 86.80i & 158.60 - 14.20i \\ -5.75 + 25.31i & -2.15 + 30.19i \\ 1.16 + \ 2.57i & -2.56 + \ 7.55i \end{pmatrix}.$$

Error estimates for the solutions, information on scaling, an estimate of the reciprocal of the condition number of the scaled matrix A and an estimate of the reciprocal of the pivot growth factor for the factorization of A are also output.

9.1 Program Text

```
    Program f07apfe

!     F07APF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zgesvx
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: rcond
      Integer                         :: i, ifail, info, lda, ldaf, ldb, ldx, &
                                         n, nrhs
      Character (1)                   :: equed
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), af(:,:), b(:,:), work(:),  &
                                            x(:,:)
      Real (Kind=nag_wp), Allocatable :: berr(:), c(:), ferr(:), r(:), rwork(:)
      Integer, Allocatable            :: ipiv(:)
      Character (1)                   :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07APF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      lda = n
      ldaf = n
      ldb = n
      ldx = n
      Allocate (a(lda,n),af(ldaf,n),b(ldb,nrhs),work(2*n),x(ldx,nrhs), &
        berr(nrhs),c(n),ferr(nrhs),r(n),rwork(2*n),ipiv(n))

!     Read A and B from data file

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

!     Solve the equations AX = B for X

!     The NAG name equivalent of zgesvx is f07apf
      Call zgesvx('Equilibrate','No transpose',n,nrhs,a,lda,af,ldaf,ipiv, &
        equed,r,c,b,ldb,x,ldx,rcond,ferr,berr,work,rwork,info)

      If ((info==0) .Or. (info==n+1)) Then
```

```
!       Print solution, error bounds, condition number, the form
!       of equilibration and the pivot growth factor

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('General',' ',n,nrhs,x,ldx,'Bracketed','F7.4', &
          'Solution(s)','Integer',rlabs,'Integer',clabs,80,0,ifail)

        Write (nout,*)
        Write (nout,*) 'Backward errors (machine-dependent)'
        Write (nout,99999) berr(1:nrhs)
        Write (nout,*)
        Write (nout,*) 'Estimated forward error bounds (machine-dependent)'
        Write (nout,99999) ferr(1:nrhs)
        Write (nout,*)
        If (equed=='N') Then
          Write (nout,*) 'A has not been equilibrated'
        Else If (equed=='R') Then
          Write (nout,*) 'A has been row scaled as diag(R)*A'
        Else If (equed=='C') Then
          Write (nout,*) 'A has been column scaled as A*diag(C)'
        Else If (equed=='B') Then
          Write (nout,*) &
            'A has been row and column scaled as diag(R)*A*diag(C)'
        End If
        Write (nout,*)
        Write (nout,*) 'Reciprocal condition number estimate of scaled matrix'
        Write (nout,99999) rcond
        Write (nout,*)
        Write (nout,*) 'Estimate of reciprocal pivot growth factor'
        Write (nout,99999) rwork(1)

        If (info==n+1) Then
          Write (nout,*)
          Write (nout,*) 'The matrix A is singular to working precision'
        End If
      Else
        Write (nout,99998) 'The (', info, ',', info, ')', &
          ' element of the factor U is zero'
      End If

99999 Format ((3X,1P,7E11.1))
99998 Format (1X,A,I3,A,I3,A,A)
    End Program f07apfe
```

9.2 Program Data

```
F07APF Example Program Data

   4              2                                   :Values of N and NRHS

 (-1.34,  2.55) (  0.28,  3.17) (-6.39,-2.20) ( 0.72,-0.92)
 (-1.70,-14.10) ( 33.10, -1.50) (-1.50,13.40) (12.90,13.80)
 (-3.29, -2.39) ( -1.91,  4.42) (-0.14,-1.35) ( 1.72, 1.35)
 ( 2.41,  0.39) ( -0.56,  1.47) (-0.83,-0.69) (-1.96, 0.67) :End of matrix A

 (26.26, 51.78) ( 31.32, -6.70)
 (64.30,-86.80) (158.60,-14.20)
 (-5.75, 25.31) ( -2.15, 30.19)
 ( 1.16,  2.57) ( -2.56,  7.55)                             :End of matrix B
```

9.3 Program Results

```
 F07APF Example Program Results

 Solution(s)
                   1                  2
 1  ( 1.0000, 1.0000) (-1.0000,-2.0000)
```

```
 2  ( 2.0000,-3.0000) ( 5.0000, 1.0000)
 3  (-4.0000,-5.0000) (-3.0000, 4.0000)
 4  ( 0.0000, 6.0000) ( 2.0000,-3.0000)

 Backward errors (machine-dependent)
       5.3E-17    4.8E-17

 Estimated forward error bounds (machine-dependent)
       5.8E-14    7.4E-14

 A has been row scaled as diag(R)*A

 Reciprocal condition number estimate of scaled matrix
       1.0E-02

 Estimate of reciprocal pivot growth factor
       8.3E-01
```