

NAG Fortran Library Routine Document

F12AGF

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

F12AGF is the main solver routine in a suite of routines consisting of F12AGF, F12ADF and F12AFF, that must be called following an initial call to F12AFF and following any calls to F12ADF.

F12AGF returns approximations to selected eigenvalues, and (optionally) the corresponding eigenvectors, of a standard or generalized eigenvalue problem defined by real banded nonsymmetric matrices. The banded matrix must be stored using the LAPACK storage format for real banded nonsymmetric matrices.

2 Specification

```

SUBROUTINE F12AGF (KL, KU, AB, LDAB, MB, LDMB, SIGMAR, SIGMAI, NCONV,
1                DR, DI, Z, LDZ, RESID, V, LDV, COMM, ICOMM, IFAIL)
    INTEGER      KL, KU, LDAB, LDMB, NCONV, LDZ, LDV, ICOMM(*), IFAIL
    double precision AB(LDAB,*), MB(LDMB,*), SIGMAR, SIGMAI, DR(*), DI(*),
1                Z(LDZ,*), RESID(*), V(LDV,*), COMM(*)

```

3 Description

The suite of routines is designed to calculate some of the eigenvalues, λ , (and optionally the corresponding eigenvectors, x) of a standard eigenvalue problem $Ax = \lambda x$, or of a generalized eigenvalue problem $Ax = \lambda Bx$ of order n , where n is large and the coefficient matrices A and B are banded, real and nonsymmetric.

Following a call to the initialization routine F12AFF, F12AGF returns the converged approximations to eigenvalues and (optionally) the corresponding approximate eigenvectors and/or an orthonormal basis for the associated approximate invariant subspace. The eigenvalues (and eigenvectors) are selected from those of a standard or generalized eigenvalue problem defined by real banded nonsymmetric matrices. There is negligible additional computational cost to obtain eigenvectors; an orthonormal basis is always computed, but there is an additional storage cost if both are requested.

The banded matrices A and B must be stored using the LAPACK storage format for banded nonsymmetric matrices; please refer to the f07 Chapter Introduction for details on this storage format.

F12AGF is based on the banded driver routines **dnbdr1** to **dnbdr6** from the ARPACK package, which uses the Implicitly Restarted Arnoldi iteration method. The method is described in Lehoucq and Sorensen (1996) and Lehoucq (2001) while its use within the ARPACK software is described in great detail in Lehoucq *et al.* (1998). An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices is provided in Lehoucq and Scott (1996). This suite of routines offers the same functionality as the ARPACK banded driver software for real nonsymmetric problems, but the interface design is quite different in order to make the option setting clearer to the user and to combine the different drivers into a general purpose routine.

F12AGF, is a general purpose forward communication routine that must be called following initialization by F12AFF. F12AGF uses options, set either by default or explicitly by calling F12ADF, to return the converged approximations to selected eigenvalues and (optionally):

- the corresponding approximate eigenvectors;
- an orthonormal basis for the associated approximate invariant subspace;
- both.

4 References

Lehoucq R B (2001) Implicitly Restarted Arnoldi Methods and Subspace Iteration *SIAM Journal on Matrix Analysis and Applications* **23** 551–562

Lehoucq R B and Scott J A (1996) An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices *Preprint MCS-P547-1195* Argonne National Laboratory

Lehoucq R B and Sorensen D C (1996) Deflation Techniques for an Implicitly Restarted Arnoldi Iteration *SIAM Journal on Matrix Analysis and Applications* **17** 789–821

Lehoucq R B, Sorensen D C and Yang C (1998) *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods* SIAM, Philadelphia

5 Parameters

- 1: KL – INTEGER *Input*
On entry: the number of subdiagonals of the matrices *A* and *B*.
Constraint: $KL \geq 0$.
- 2: KU – INTEGER *Input*
On entry: the number of superdiagonals of the matrices *A* and *B*.
Constraint: $KU \geq 0$.
- 3: AB(LDAB,*) – **double precision** array *Input*
Note: the second dimension of the array AB must be at least $\max(1, N)$.
On entry: AB must contain the matrix *A* in LAPACK banded storage format for nonsymmetric matrices.
- 4: LDAB – INTEGER *Input*
On entry: the first dimension of the array LDAB as declared in the (sub)program from which F12AGF is called.
Constraint: $LDAB \geq 2 \times KL + KU + 1$.
- 5: MB(LDMB,*) – **double precision** array *Input*
Note: the second dimension of the array MB must be at least $\max(1, N)$.
On entry: MB must contain the matrix *B* in LAPACK banded storage format for nonsymmetric matrices.
- 6: LDMB – INTEGER *Input*
On entry: the first dimension of the array LDMB as declared in the (sub)program from which F12AGF is called.
- 7: SIGMAR – **double precision** *Input*
On entry: if one of the **Shifted** modes have been selected then SIGMAR contains the real part of the shift used; otherwise SIGMAR is not referenced.
- 8: SIGMAI – **double precision** *Input*
On entry: if one of the **Shifted** modes have been selected then SIGMAI contains the imaginary part of the shift used; otherwise SIGMAI is not referenced.
- 9: NCONV – INTEGER *Output*
On exit: the number of converged eigenvalues.

- 10: DR(*) – **double precision** array *Output*
Note: the dimension of the array DR must be at least NEV.
On exit: the first NCONV locations of the array DR contain the real parts of the converged approximate eigenvalues.
- 11: DI(*) – **double precision** array *Output*
Note: the dimension of the array DI must be at least NEV.
On exit: the first NCONV locations of the array DI contain the imaginary parts of the converged approximate eigenvalues.
- 12: Z(LDZ,*) – **double precision** array *Output*
Note: the second dimension of the array Z must be at least NEV + 1 if the default option **Vectors** = Ritz has been selected and at least 1 if the option **Vectors** = None or Schur has been selected.
On exit: if the default option **Vectors** = Ritz has been selected then Z contains the final set of eigenvectors corresponding to the eigenvalues held in DR and DI. The complex eigenvector associated with the eigenvalue with positive imaginary part is stored in two consecutive columns. The first column holds the real part of the eigenvector and the second column holds the imaginary part. The eigenvector associated with the eigenvalue with negative imaginary part is simply the complex conjugate of the eigenvector associated with the positive imaginary part.
- 13: LDZ – INTEGER *Input*
On entry: the first dimension of the array Z as declared in the (sub)program from which F12AGF is called.
Constraints:
 if the default option **Vectors** = Ritz has been selected, $LDZ \geq N$;
 if the option **Vectors** = None or Schur has been selected, $LDZ \geq 1$.
- 14: RESID(*) – **double precision** array *Input/Output*
Note: the dimension of the array RESID must be at least N.
On entry: RESID need not be set unless the option **Initial Residual** has been set in a prior call to F12ADF in which case RESID should contain an initial residual vector.
On exit: RESID contains the final residual vector.
- 15: V(LDV,*) – **double precision** array *Output*
Note: the second dimension of the array V must be at least $\max(1, NCV)$.
On exit: if the option **Vectors** = Schur has been set or the option **Vectors** = Ritz has been set and a separate array Z has been passed then the first NCONV columns of V will contain approximate Schur vectors that span the desired invariant subspace.
- 16: LDV – INTEGER *Input*
On entry: the first dimension of the array V as declared in the (sub)program from which F12AGF is called.
- 17: COMM(*) – **double precision** array *Communication Array*
 COMM, on initial entry, must remain unchanged from a prior call to F12AFF or F12ADF.
- 18: ICOMM(*) – INTEGER array *Communication Array*
 ICOMM, on initial entry, must remain unchanged from the prior call to F12AFF or F12ADF.

19: IFAIL – INTEGER

Input/Output

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

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

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, $KL < 0$.

IFAIL = 2

On entry, $KU < 0$.

IFAIL = 3

On entry, $LDAB < 2 \times KL + KU + 1$.

IFAIL = 4

On entry, the option **shifted inverse imaginary** was selected, and SIGMAI = zero, but SIGMAI must be non-zero for this computational mode.

IFAIL = 5

iteration limit < 0.

IFAIL = 6

The options **Generalized** and **Regular** are incompatible.

IFAIL = 7

The option **Initial Residual** was selected but the starting vector held in RESID is zero.

IFAIL = 8

Either the initialization routine F12AFF has not been called prior to the first call of this routine or a communication array has become corrupted.

IFAIL = 9

On entry, $LDZ < \max(1, N)$ or $LDZ < 1$ when no vectors are required.

IFAIL = 10

On entry, the option **Vectors** = Select was selected, but this is not yet implemented.

IFAIL = 11

The number of eigenvalues found to sufficient accuracy is zero.

IFAIL = 12

Could not build an Arnoldi factorization. Consider changing NCV or NEV in the initialization routine (see Section 5 of the document for F12AAF for details of these parameters).

IFAIL = 13

Unexpected error in internal call to compute eigenvalues and corresponding error bounds of the current upper Hessenberg matrix. Please contact NAG.

IFAIL = 14

Unexpected error during calculation of a real Schur form: there was a failure to compute all the converged eigenvalues. Please contact NAG.

IFAIL = 15

Unexpected error: the computed Schur form could not be reordered by an internal call. Please contact NAG.

IFAIL = 16

Unexpected error in internal call while calculating eigenvectors. Please contact NAG.

IFAIL = 17

Failure during internal factorization of real banded matrix. Please contact NAG.

IFAIL = 18

Failure during internal solution of real banded system. Please contact NAG.

IFAIL = 19

Failure during internal factorization of complex banded matrix. Please contact NAG.

IFAIL = 20

Failure during internal solution of complex banded system. Please contact NAG.

IFAIL = 21

The maximum number of iterations has been reached. Some Ritz values may have converged; NCONV returns the number of converged values.

IFAIL = 22

No shifts could be applied during a cycle of the implicitly restarted Arnoldi iteration. One possibility is to increase the size of NCV relative to NEV (see Section 5 of the document for F12AFF for details of these parameters).

IFAIL = 23

Overflow occurred during transformation of Ritz values to those of the original problem.

IFAIL = 24

The routine was unable to dynamically allocate sufficient internal workspace. Please contact NAG.

IFAIL = 25

An unexpected error has occurred. Please contact NAG.

7 Accuracy

The relative accuracy of a Ritz value, λ , is considered acceptable if its Ritz estimate $\leq \textbf{Tolerance} \times |\lambda|$. The default **Tolerance** used is the *machine precision* given by X02AJF.

8 Further Comments

None.

9 Example

The example constructs the matrices A and B using LAPACK band storage format and solves $Ax = \lambda Bx$ in shifted imaginary mode using the complex shift σ .

9.1 Program Text

```
*      F12AGF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
      INTEGER          LICOMM, NIN, NOUT
      PARAMETER        (LICOMM=140,NIN=5,NOUT=6)
      INTEGER          MAXBDW, MAXN, MAXNCV, LDAB, LDMB, LDV
      PARAMETER        (MAXBDW=50,MAXN=1000,MAXNCV=50,LDAB=MAXBDW,
+                      LDMB=MAXBDW,LDV=MAXN)
      INTEGER          LCOMM
      PARAMETER        (LCOMM=60)
      DOUBLE PRECISION ONE, ZERO
      PARAMETER        (ONE=1.0D+0,ZERO=0.0D+0)
*      .. Local Scalars ..
      DOUBLE PRECISION H, RHO, SIGMAI, SIGMAR
      INTEGER          I, IDIAG, IFAIL, IFAIL1, ISUB, ISUP, J, KL, KU,
+                      LO, N, NCONV, NCV, NEV, NX
      LOGICAL          FIRST
*      .. Local Arrays ..
      DOUBLE PRECISION AB(LDAB,MAXN), AX(MAXN), COMM(LCOMM),
+                      D(MAXNCV,3), MB(LDAB,MAXN), MX(MAXN),
+                      RESID(MAXN), V(LDV,MAXNCV)
      INTEGER          ICOMM(LICOMM)
*      .. External Functions ..
      DOUBLE PRECISION DNRM2, F06BNF
      EXTERNAL          DNRM2, F06BNF
*      .. External Subroutines ..
      EXTERNAL          DAXPY, DGBMV, F06QHF, F12ADF, F12AFF, F12AGF,
+                      X04ABF, X04CAF
*      .. Intrinsic Functions ..
*
      INTRINSIC          DABS, DBLE
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F12AGF Example Program Results'
      WRITE (NOUT,*)
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) NX, NEV, NCV, SIGMAR, SIGMAI
      N = NX*NX
      IF (N.LT.1 .OR. N.GT.MAXN) THEN
         WRITE (NOUT,99999) 'N is out of range: N = ', N
      ELSE IF (NCV.GT.MAXNCV) THEN
         WRITE (NOUT,99999) 'NCV is out of range: NCV = ', NCV
      ELSE
         IFAIL = 0
*      Initialize communication arrays.
         CALL F12AFF(N,NEV,NCV,ICOMM,LICOMM,COMM,LCOMM,IFAIL)
*      Set the mode.
         CALL F12ADF('SHIFTED IMAGINARY',ICOMM,COMM,IFAIL)
*      Set problem type
         CALL F12ADF('GENERALIZED',ICOMM,COMM,IFAIL)
*
```

```

*      Construct the matrix A in banded form and store in AB.
*      Zero out AB and MB.
          CALL F06QHF('G',LDAB,N,ZERO,ZERO,AB,LDAB)
          CALL F06QHF('G',LDAB,N,ZERO,ZERO,MB,LDAB)
*      KU, KL are number of superdiagonals and subdiagonals within the
*      band of matrices A and M.
          KL = NX
          KU = NX
*      Main diagonal of A.
          IDIAG = KL + KU + 1
          DO 20 J = 1, N
              AB(IDIAG,J) = 4.0D+0
              MB(IDIAG,J) = 4.0D+0
20      CONTINUE
*      First subdiagonal and superdiagonal of A.
          ISUP = KL + KU
          ISUB = KL + KU + 2
          RHO = 100.0D0
          H = ONE/DBLE(NX+1)
          DO 60 I = 1, NX
              LO = (I-1)*NX
              DO 40 J = LO + 1, LO + NX - 1
                  AB(ISUB,J+1) = -ONE + 0.5D0*H*RHO
                  AB(ISUB,J) = -ONE - 0.5D0*H*RHO
40          CONTINUE
60      CONTINUE
*
          DO 80 J = 1, N - 1
              MB(ISUB,J+1) = ONE
              MB(ISUP,J) = ONE
80      CONTINUE
*      KL-th subdiagonal and KU-th super-diagonal.
          ISUP = KL + 1
          ISUB = 2*KL + KU + 1
          DO 120 I = 1, NX - 1
              LO = (I-1)*NX
              DO 100 J = LO + 1, LO + NX
                  AB(ISUP,NX+J) = -ONE
                  AB(ISUB,J) = -ONE
100         CONTINUE
120      CONTINUE
*
*      Find eigenvalues closest in value to SIGMA and corresponding
*      eigenvectors.
          IFAIL = -1
          CALL F12AGF(KL,KU,AB,LDAB,MB,LDMB,SIGMAR,SIGMAI,NCONV,D,D(1,2),
+              V,LDV,RESID,V,LDV,COMM,ICOMM,IFAIL)
          IF (IFAIL.EQ.0) THEN
*      Compute the residual norm ||A*x - lambda*x||.
              FIRST = .TRUE.
              DO 140 J = 1, NCONV
                  IF (D(J,2).EQ.ZERO) THEN
+                      CALL DGBMV('NoTranspose',N,N,KL,KU,ONE,AB(KL+1,1),
+                          LDAB,V(1,J),1,ZERO,AX,1)
+                      CALL DGBMV('NoTranspose',N,N,KL,KU,ONE,MB(KL+1,1),
+                          LDMB,V(1,J),1,ZERO,MX,1)
+                      CALL DAXPY(N,-D(J,1),MX,1,AX,1)
                      D(J,3) = DNRM2(N,AX,1)
                      D(J,3) = D(J,3)/DABS(D(J,1))
                  ELSE IF (FIRST) THEN
+                      CALL DGBMV('NOTRANSPOSE',N,N,KL,KU,ONE,AB(KL+1,1),
+                          LDAB,V(1,J),1,ZERO,AX,1)
+                      CALL DGBMV('NOTRANSPOSE',N,N,KL,KU,ONE,MB(KL+1,1),
+                          LDAB,V(1,J),1,ZERO,MX,1)
+                      CALL DAXPY(N,-D(J,1),MX,1,AX,1)
+                      CALL DGBMV('NOTRANSPOSE',N,N,KL,KU,ONE,MB(KL+1,1),
+                          LDMB,V(1,J+1),1,ZERO,MX,1)
+                      CALL DAXPY(N,D(J,2),MX,1,AX,1)
                      D(J,3) = DNRM2(N,AX,1)
+                      CALL DGBMV('NOTRANSPOSE',N,N,KL,KU,ONE,AB(KL+1,1),
+                          LDAB,V(1,J+1),1,ZERO,AX,1)

```

```

      CALL DGBMV('NOTRANSPOSE',N,N,KL,KU,ONE,MB(KL+1,1),
+          LDMB,V(1,J+1),1,ZERO,MX,1)
      CALL DAXPY(N,-D(J,1),MX,1,AX,1)
      CALL DGBMV('NOTRANSPOSE',N,N,KL,KU,ONE,MB(KL+1,1),
+          LDMB,V(1,J),1,ZERO,MX,1)
      CALL DAXPY(N,-D(J,2),MX,1,AX,1)
      D(J,3) = F06BNF(D(J,3),DNRM2(N,AX,1))
      D(J,3) = D(J,3)/F06BNF(D(J,1),D(J,2))
      D(J+1,3) = D(J,3)
      FIRST = .FALSE.
    ELSE
      FIRST = .TRUE.
    END IF
140    CONTINUE
      WRITE (NOUT,*)
      CALL X04ABF(1,NOUT)
      CALL X04CAF('G','N',NCONV,2,D,MAXNCV,
+          ' Ritz values and residuals',IFAIL1)
    ELSE
      WRITE (NOUT,99998) IFAIL
    END IF
  END IF
*
99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,' NAG Routine F12AGF Returned with IFAIL = ',I6)
END

```

9.2 Program Data

F12AGF Example Program Data

10 4 10 0.4 0.6 : Values for NX NEV NCV SIGMAR SIGMAI

9.3 Program Results

F12AGF Example Program Results

Ritz values and residuals

	1	2
1	-4.1710	-1.0613
2	-4.1710	1.0613
3	-6.8764	1.2147
4	-6.8764	-1.2147
