# NAG Library Routine Document

# G02AJF

**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

G02AJF computes the nearest correlation matrix, using element-wise weighting in the Frobenius norm and optionally with bounds on the eigenvalues, to a given square, input matrix.

## 2 Specification

```
SUBROUTINE G02AJF (G, LDG, N, ALPHA, H, LDH, ERRTOL, MAXIT, X, LDX,        &
                   ITER, NORM, IFAIL)

INTEGER            LDG, N, LDH, MAXIT, LDX, ITER, IFAIL
REAL (KIND=nag_wp) G(LDG,N), ALPHA, H(LDH,N), ERRTOL, X(LDX,N), NORM
```

## 3 Description

G02AJF finds the nearest correlation matrix, $X$, to an approximate correlation matrix, $G$, using element-wise weighting, this minimizes $\|H \circ (G - X)\|_F$, where $C = A \circ B$ denotes the matrix $C$ with elements $C_{ij} = A_{ij} \times B_{ij}$.

You can optionally specify a lower bound on the eigenvalues, $\alpha$, of the computed correlation matrix, forcing the matrix to be strictly positive definite, if $0 < \alpha < 1$.

Zero elements in $H$ should be used when you wish to put no emphasis on the corresponding element of $G$. The algorithm scales $H$ so that the maximum element is 1. It is this scaled matrix that is used in computing the norm above and for the stopping criteria described in Section 7.

Note that if the elements in $H$ vary by several orders of magnitude from one another the algorithm may fail to converge.

## 4 References

Borsdorf R and Higham N J (2010) A preconditioned (Newton) algorithm for the nearest correlation matrix *IMA Journal of Numerical Analysis* **30(1)** 94–107

Jiang K, Sun D and Toh K-C (To appear) An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP

Qi H and Sun D (2006) A quadratically convergent Newton method for computing the nearest correlation matrix *SIAM J. Matrix AnalAppl* **29(2)** 360–385

## 5 Parameters

1: G(LDG, N) – REAL (KIND=nag_wp) array *Input/Output*

*On entry*: $G$, the initial matrix.

*On exit*: $G$ is overwritten.

2: LDG – INTEGER *Input*

*On entry*: the first dimension of the array G as declared in the (sub)program from which G02AJF is called.

*Constraint*: LDG $\geq$ N.

3: N – INTEGER *Input*

*On entry*: the order of the matrix $G$.

*Constraint*: N $> 0$.

4: ALPHA – REAL (KIND=nag_wp) *Input*

*On entry*: the value of $\alpha$.

If ALPHA $< 0.0$, 0.0 is used.

*Constraint*: ALPHA $< 1.0$.

5: H(LDH, N) – REAL (KIND=nag_wp) array *Input/Output*

*On entry*: the matrix of weights $H$.

*On exit*: a symmetric matrix $\frac{1}{2}(H + H^{\mathrm{T}})$ with its diagonal elements set to zero and the remaining elements scaled so that the maximum element is 1.0.

*Constraint*: $H(i,j) \geq 0.0$, for all $i$ and $j = 1, 2, \ldots, n$, $i \neq j$.

6: LDH – INTEGER *Input*

*On entry*: the first dimension of the array H as declared in the (sub)program from which G02AJF is called.

*Constraint*: LDH $\geq$ N.

7: ERRTOL – REAL (KIND=nag_wp) *Input*

*On entry*: the termination tolerance for the iteration. If ERRTOL $\leq 0.0$ then N $\times \sqrt{\textbf{\textit{machine precision}}}$ is used. See Section 7 for further details.

8: MAXIT – INTEGER *Input*

*On entry*: specifies the maximum number of iterations to be used.

If MAXIT $\leq 0$, 200 is used.

9: X(LDX, N) – REAL (KIND=nag_wp) array *Output*

*On exit*: contains the nearest correlation matrix.

10: LDX – INTEGER *Input*

*On entry*: the first dimension of the array X as declared in the (sub)program from which G02AJF is called.

*Constraint*: LDX $\geq$ N.

11: ITER – INTEGER *Output*

*On exit*: the number of iterations taken.

12: NORM – REAL (KIND=nag_wp) *Output*

*On exit*: the value of $\|H \circ (G - X)\|_F$ after the final iteration.

13: IFAIL – INTEGER *Input/Output*

*On entry*: IFAIL must be set to 0, $-1$ or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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, if you are 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.**

*On exit*: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

## 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, N $= \langle value \rangle$.
Constraint: N $> 0$.

IFAIL $= 2$

On entry, LDG $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LDG $\geq$ N.

IFAIL $= 3$

On entry, LDH $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LDH $\geq$ N.

IFAIL $= 4$

On entry, LDX $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LDX $\geq$ N.

IFAIL $= 5$

On entry, ALPHA $= \langle value \rangle$.
Constraint: ALPHA $< 1.0$.

IFAIL $= 6$

On entry, one or more of the off-diagonal elements of $H$ were negative.

IFAIL $= 7$

Routine fails to converge in $\langle value \rangle$ iterations.
Increase MAXIT or check the call to the routine.

IFAIL $= 8$

Failure to solve intermediate eigenproblem. This should not occur. Please contact NAG with details of your call.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

## 7 Accuracy

The returned accuracy is controlled by ERRTOL and limited by ***machine precision***. If $e_i$ is the value of NORM at the $i$th iteration, that is

$$e_i = \|H \circ (G - X)\|_F,$$

where $H$ has been scaled as described above, then the algorithm terminates when:

$$\frac{|e_i - e_{i-1}|}{1 + \max(e_i, e_{i-1})} \leq \text{ERRTOL}.$$

## 8 Parallelism and Performance

G02AJF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02AJF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

## 9 Further Comments

Arrays are internally allocated by G02AJF. The total size of these arrays is $15 \times \text{N} + 5 \times \text{N} \times \text{N} + \max(2 \times \text{N} \times \text{N} + 6 \times \text{N} + 1, 120 + 9 \times \text{N})$ real elements and $5 \times \text{N} + 3$ integer elements. All allocated memory is freed before return of G02AJF.

## 10 Example

This example finds the nearest correlation matrix to:

$$G = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

weighted by:

$$H = \begin{pmatrix} 0.0 & 10.0 & 0.0 & 0.0 \\ 10.0 & 0.0 & 1.5 & 1.5 \\ 0.0 & 1.5 & 0.0 & 0.0 \\ 0.0 & 1.5 & 0.0 & 0.0 \end{pmatrix}$$

with minimum eigenvalue 0.04.

### 10.1 Program Text

```
    Program g02ajfe

!     G02AJF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
```

```
      Use nag_library, Only: dsyev, g02ajf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: alpha, errtol, norm
      Integer                          :: i, ifail, iter, ldg, ldh, ldx,       &
                                          lwork, maxit, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: eig(:), g(:,:), h(:,:), work(:),     &
                                          x(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'G02AJF Example Program Results'
      Write (nout,*)
      Flush (nout)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size and alpha
      Read (nin,*) n, alpha

      ldg = n
      ldh = n
      ldx = n
      lwork = 66*n
      Allocate (g(ldg,n),h(ldh,n),x(ldx,n),eig(n),work(lwork))

!     Read in the matrix G
      Read (nin,*)(g(i,1:n),i=1,n)

!     Read in the matrix H
      Read (nin,*)(h(i,1:n),i=1,n)

!     Use the defaults for ERRTOL and MAXIT
      errtol = 0.0E0_nag_wp
      maxit = 0

!     Calculate nearest correlation matrix
      ifail = 0

      Call g02ajf(g,ldg,n,alpha,h,ldh,errtol,maxit,x,ldx,iter,norm,ifail)

!     Display results
      ifail = 0
      Call x04caf('General',' ',n,n,h,ldh,'Returned H Matrix',ifail)
      Write (nout,*)

      ifail = 0
      Call x04caf('General',' ',n,n,x,ldx,'Nearest Correlation Matrix X', &
        ifail)
      Write (nout,*)

      Write (nout,99999) 'Number of iterations:', iter
      Write (nout,*)
      Write (nout,99998) 'Norm value:', norm
      Write (nout,*)
      Write (nout,99997) 'ALPHA: ', alpha

      ifail = 0
!     The NAG name equivalent of dsyev is f08faf
      Call dsyev('N','U',n,x,ldx,eig,work,lwork,ifail)
      Write (nout,*)
      Flush (nout)
      Call x04caf('General',' ',1,n,eig,1,'Eigenvalues of X',ifail)
```

```
99999 Format (1X,A,I11)
99998 Format (1X,A,F26.4)
99997 Format (1X,A,F30.4)

    End Program g02ajfe
```

## 10.2 Program Data

```
G02AJF Example Program Data
 4  0.04                      :: N, ALPHA
   2.0   -1.0    0.0    0.0
  -1.0    2.0   -1.0    0.0
   0.0   -1.0    2.0   -1.0
   0.0    0.0   -1.0    2.0  :: End of G
   0.0   10.0    0.0    0.0
  10.0    0.0    1.5    1.5
   0.0    1.5    0.0    0.0
   0.0    1.5    0.0    0.0  :: End of H
```

## 10.3 Program Results

```
 G02AJF Example Program Results

 Returned H Matrix
          1       2       3       4
 1   0.0000  1.0000  0.0000  0.0000
 2   1.0000  0.0000  0.1500  0.1500
 3   0.0000  0.1500  0.0000  0.0000
 4   0.0000  0.1500  0.0000  0.0000

 Nearest Correlation Matrix X
               1            2            3            4
 1    1.0000E+00  -9.2285E-01   7.7335E-01   2.5854E-03
 2   -9.2285E-01   1.0000E+00  -7.8433E-01  -8.4891E-07
 3    7.7335E-01  -7.8433E-01   1.0000E+00  -6.1477E-02
 4    2.5854E-03  -8.4891E-07  -6.1477E-02   1.0000E+00

 Number of iterations:         66

 Norm value:                    0.1183

 ALPHA:                         0.0400

 Eigenvalues of X
             1          2          3          4
 1      0.0769     0.2637     1.0031     2.6563
```