# NAG Library Routine Document

# G02ABF

**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

G02ABF computes the nearest correlation matrix, in the Frobenius norm or weighted Frobenius norm, and optionally with bounds on the eigenvalues, to a given square, input matrix.

## 2 Specification

```
SUBROUTINE G02ABF (G, LDG, N, OPT, ALPHA, W, ERRTOL, MAXITS, MAXIT, X,     &
                   LDX, ITER, FEVAL, NRMGRD, IFAIL)
INTEGER            LDG, N, MAXITS, MAXIT, LDX, ITER, FEVAL, IFAIL
REAL (KIND=nag_wp) G(LDG,N), ALPHA, W(N), ERRTOL, X(LDX,N), NRMGRD
CHARACTER(1)       OPT
```

## 3 Description

Finds the nearest correlation matrix $X$ by minimizing $\frac{1}{2}\|G - X\|^2$ where $G$ is an approximate correlation matrix.

The norm can either be the Frobenius norm or the weighted Frobenius norm $\frac{1}{2}\left\|W^{\frac{1}{2}}(G - X)W^{\frac{1}{2}}\right\|_F^2$.

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

Note that if the weights 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

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 Arguments

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 G02ABF is called.

*Constraint*: LDG $\geq$ N.

3: N – INTEGER *Input*

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

*Constraint*: $\mathrm{N} > 0$.

4: OPT – CHARACTER(1) *Input*

*On entry*: indicates the problem to be solved.

OPT = 'A'
The lower bound problem is solved.

OPT = 'W'
The weighted norm problem is solved.

OPT = 'B'
Both problems are solved.

*Constraint*: OPT = 'A', 'W' or 'B'.

5: ALPHA – REAL (KIND=nag_wp) *Input*

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

If OPT = 'W', ALPHA need not be set.

*Constraint*: $0.0 < \mathrm{ALPHA} < 1.0$.

6: W(N) – REAL (KIND=nag_wp) array *Input/Output*

*On entry*: the square roots of the diagonal elements of $W$, that is the diagonal of $W^{\frac{1}{2}}$.

If OPT = 'A', W is not referenced and need not be set.

*On exit*: if OPT = 'W' or 'B', the array is scaled so $0 < \mathrm{W}(i) \leq 1$, for $i = 1, 2, \ldots, n$.

*Constraint*: $\mathrm{W}(i) > 0.0$, for $i = 1, 2, \ldots, n$.

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

*On entry*: the termination tolerance for the Newton iteration. If $\mathrm{ERRTOL} \leq 0.0$ then $\mathrm{N} \times \sqrt{\textit{machine precision}}$ is used.

8: MAXITS – INTEGER *Input*

*On entry*: specifies the maximum number of iterations to be used by the iterative scheme to solve the linear algebraic equations at each Newton step.

If $\mathrm{MAXITS} \leq 0$, $2 \times \mathrm{N}$ is used.

9: MAXIT – INTEGER *Input*

*On entry*: specifies the maximum number of Newton iterations.

If $\mathrm{MAXIT} \leq 0$, 200 is used.

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

*On exit*: contains the nearest correlation matrix.

11: LDX – INTEGER *Input*

*On entry*: the first dimension of the array X as declared in the (sub)program from which G02ABF is called.

*Constraint*: $\mathrm{LDX} \geq \mathrm{N}$.

12: ITER – INTEGER *Output*

*On exit*: the number of Newton steps taken.

13: FEVAL – INTEGER *Output*

*On exit*: the number of function evaluations of the dual problem.

14: NRMGRD – REAL (KIND=nag_wp) *Output*

*On exit*: the norm of the gradient of the last Newton step.

15: IFAIL – INTEGER *Input/Output*

*On entry*: IFAIL must be set to 0, $-1$ or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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, all elements of W were not positive.

On entry, ALPHA $= \langle value \rangle$.
Constraint: $0.0 <$ ALPHA $< 1.0$.

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

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

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

On entry, OPT $\neq$ 'A', 'W' or 'B'.

IFAIL $= 2$

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

IFAIL $= 3$

The ***machine precision*** is limiting convergence. In this instance the returned value of X may be useful.

IFAIL $= 4$

An intermediate eigenproblem could not be solved. 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 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

## 7 Accuracy

The returned accuracy is controlled by ERRTOL and limited by ***machine precision***.

## 8 Parallelism and Performance

G02ABF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02ABF 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 G02ABF. The total size of these arrays is $12 \times \mathrm{N} + 3 \times \mathrm{N} \times \mathrm{N} + \max(2 \times \mathrm{N} \times \mathrm{N} + 6 \times \mathrm{N} + 1, 120 + 9 \times \mathrm{N})$ real elements and $5 \times \mathrm{N} + 3$ integer elements. All allocated memory is freed before return of G02ABF.

## 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 $W^{\frac{1}{2}} = \mathrm{diag}(100, 20, 20, 20)$ with minimum eigenvalue 0.02.

### 10.1 Program Text

```
    Program g02abfe

!     G02ABF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dsyev, g02abf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)                :: alpha, errtol, nrmgrd
      Integer                           :: feval, i, ifail, iter, ldg, ldx,     &
                                           lwork, maxit, maxits, n
      Character (1)                     :: opt
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: eig(:), g(:,:), w(:), work(:),       &
                                           x(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'G02ABF Example Program Results'
      Write (nout,*)
      Flush (nout)

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

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

      ldg = n
      ldx = n
      lwork = 66*n
      Allocate (g(ldg,n),w(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 vector W
      Read (nin,*) w(1:n)

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

!     Calculate nearest correlation matrix
      ifail = 0

      Call g02abf(g,ldg,n,opt,alpha,w,errtol,maxits,maxit,x,ldx,iter,feval,    &
        nrmgrd,ifail)

!     Display results
      ifail = 0
      Call x04caf('General',' ',n,n,x,ldx,'Nearest Correlation Matrix X',      &
        ifail)
      Write (nout,*)
      Write (nout,99999) 'Number of Newton steps taken:', iter
      Write (nout,99998) 'Number of function evaluations:', feval

      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,I9)
99997 Format (1X,A,F37.3)

    End Program g02abfe
```

## 10.2 Program Data

```
G02ABF Example Program Data
 4 'B' 0.02                      :: N, OPT, 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
 100.0   20.0   20.0   20.0  :: W
```

## 10.3 Program Results

```
 G02ABF Example Program Results

 Nearest Correlation Matrix X
          1       2       3       4
 1   1.0000 -0.9187  0.0257  0.0086
 2  -0.9187  1.0000 -0.3008  0.2270
 3   0.0257 -0.3008  1.0000 -0.8859
 4   0.0086  0.2270 -0.8859  1.0000

 Number of Newton steps taken:         5
 Number of function evaluations:       6

 ALPHA:                                0.020

 Eigenvalues of X
             1          2          3          4
 1      0.0392     0.1183     1.6515     2.1910
```