NAG Library Routine Document

G02APF

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

G02APF computes a correlation matrix, by using a positive definite **target** matrix derived from weighting the approximate input matrix, with an optional bound on the minimum eigenvalue.

2 Specification

```
SUBROUTINE G02APF (G, LDG, N, THETA, H, LDH, ERRTOL, EIGTOL, X, LDX,       &
                   ALPHA, ITER, EIGMIN, NORM, IFAIL)
INTEGER            LDG, N, LDH, LDX, ITER, IFAIL
REAL (KIND=nag_wp) G(LDG,N), THETA, H(LDH,N), ERRTOL, EIGTOL,              &
                   X(LDX,N), ALPHA, EIGMIN, NORM
```

3 Description

Starting from an approximate correlation matrix, G, G02APF finds a correlation matrix, X, which has the form

$$X = \alpha T + (1 - \alpha)G,$$

where $\alpha \in [0, 1]$ and $T = H \circ G$ is a target matrix. $C = A \circ B$ denotes the matrix C with elements $C_{ij} = A_{ij} \times B_{ij}$. H is a matrix of weights that defines the target matrix. The target matrix must be positive definite and thus have off-diagonal elements less than 1 in magnitude. A value of 1 in H essentially fixes an element in G so it is unchanged in X.

G02APF utilizes a shrinking method to find the minimum value of α such that X is positive definite with unit diagonal and with a smallest eigenvalue of at least $\theta \in [0, 1)$ times the smallest eigenvalue of the target matrix.

4 References

Higham N J, Strabić N and Šego V (2014) Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block *MIMS EPrint 2014.54* Manchester Institute for Mathematical Sciences, The University of Manchester, UK

5 Arguments

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

On entry: G, the initial matrix.

On exit: a symmetric matrix $\frac{1}{2}(G + G^{\mathrm{T}})$ with the diagonal elements set to 1.0.

2: LDG – INTEGER *Input*

On entry: the first dimension of the array G as declared in the (sub)program from which G02APF is called.

Constraint: LDG $\geq$ N.

3: N – INTEGER *Input*

On entry: the order of the matrix G.

Constraint: $N > 0$.

4: THETA – REAL (KIND=nag_wp) *Input*

On entry: the value of θ. If THETA < 0.0, 0.0 is used.

Constraint: THETA < 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 1.0.

6: LDH – INTEGER *Input*

On entry: the first dimension of the array H as declared in the (sub)program from which G02APF is called.

Constraint: LDH $\geq$ N.

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

On entry: the termination tolerance for the iteration.

If ERRTOL ≤ 0, $\sqrt{\textbf{\textit{machine precision}}}$ is used. See Section 7 for further details.

8: EIGTOL – REAL (KIND=nag_wp) *Input*

On entry: the tolerance used in determining the definiteness of the target matrix $T = H \circ G$.

If $\lambda_{\min}(T) > \mathrm{N} \times \lambda_{\max}(T) \times \mathrm{EIGTOL}$, where $\lambda_{\min}(T)$ and $\lambda_{\max}(T)$ denote the minimum and maximum eigenvalues of T respectively, T is positive definite.

If EIGTOL ≤ 0, ***machine precision*** is used.

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

On exit: contains the matrix X.

10: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G02APF is called.

Constraint: LDX $\geq$ N.

11: ALPHA – REAL (KIND=nag_wp) *Output*

On exit: the constant α used in the formation of X.

12: ITER – INTEGER *Output*

On exit: the number of iterations taken.

13: EIGMIN – REAL (KIND=nag_wp) *Output*

On exit: the smallest eigenvalue of the target matrix T.

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

On exit: the value of $\|G - X\|_F$ after the final iteration.

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, 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, THETA $= \langle value \rangle$.
Constraint: THETA < 1.0.

IFAIL $= 4$

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

IFAIL $= 5$

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

IFAIL $= 6$

The target matrix is not positive definite.

IFAIL $= 7$

Failure to solve intermediate eigenproblem. This should not occur. Please contact NAG.

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 algorithm uses a bisection method. It is terminated when the computed α is within ERRTOL of the minimum value.

Note: when θ is zero X is still positive definite, in that it can be successfully factorized with a call to F07FDF (DPOTRF).

The number of iterations taken for the bisection will be:

$$\left\lceil \log_2\left(\frac{1}{\mathrm{ERRTOL}}\right)\right\rceil.$$

8 Parallelism and Performance

G02APF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02APF 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 G02APF. The total size of these arrays does not exceed $2 \times n^2 + 3 \times n$ real elements. All allocated memory is freed before return of G02APF.

10 Example

This example finds the smallest α such that $\alpha(H \circ G) + (1 - \alpha)G$ is a correlation matrix. The 2 by 2 leading principal submatrix of the input is preserved, and the last 2 by 2 diagonal block is weighted to give some emphasis to the off diagonal elements.

$$G = \begin{pmatrix} 1.0000 & -0.0991 & 0.5665 & -0.5653 & -0.3441 \\ -0.0991 & 1.0000 & -0.4273 & 0.8474 & 0.4975 \\ 0.5665 & -0.4273 & 1.0000 & -0.1837 & -0.0585 \\ -0.5653 & 0.8474 & -0.1837 & 1.0000 & -0.2713 \\ -0.3441 & 0.4975 & -0.0585 & -0.2713 & 1.0000 \end{pmatrix}$$

and

$$H = \begin{pmatrix} 1.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 1.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.5000 \\ 0.0000 & 0.0000 & 0.0000 & 0.5000 & 1.0000 \end{pmatrix}.$$

10.1 Program Text

```
    Program g02apfe

!     G02APF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dsyev, g02apf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)              :: alpha, eigmin, eigtol, errtol, norm, &
                                         theta
      Integer                         :: i, ifail, iter, ldg, ldh, ldx,       &
                                         lwork, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: eig(:), g(:,:), h(:,:), work(:),     &
                                         x(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'G02APF Example Program Results'
      Write (nout,*)
      Flush (nout)

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

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

      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 EIGTOL and ERRTOL
      eigtol = -1.E0_nag_wp
      errtol = -1.E0_nag_wp

!     Calculate nearest correlation matrix using target matrix
      ifail = 0

      Call g02apf(g,ldg,n,theta,h,ldh,errtol,eigtol,x,ldx,alpha,iter,eigmin,   &
        norm,ifail)

      ifail = 0
!     Display the symmetrised input matrix
      Call x04caf('General',' ',n,n,g,ldg,'Symmetrised Input Matrix G',ifail)
      Write (nout,*)

!     Display results
      ifail = 0
      Call x04caf('General',' ',n,n,x,ldx,'Nearest Correlation Matrix X',     &
        ifail)
      Write (nout,*)
      Write (nout,99999) 'Number of iterations taken:', iter
      Write (nout,*)
      Write (nout,99998) 'ALPHA:                             ', alpha
      Write (nout,*)
      Write (nout,99998) 'Norm value:                        ', norm
      Write (nout,*)
```

```
      Write (nout,99998) 'THETA:                          ', theta
      Write (nout,*)
      Write (nout,99998) 'Smallest eigenvalue of target:', eigmin

      ifail = 0
!     Compute the eigenvalues of X
      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,I9)
99998 Format (1X,A,F11.4)

    End Program g02apfe
```

10.2 Program Data

```
G02APF Example Program Data
  5, 0.1                                   :: N, THETA
  1.0000 -0.0991  0.5665 -0.5653 -0.3441
 -0.0991  1.0000 -0.4273  0.8474  0.4975
  0.5665 -0.4273  1.0000 -0.1837 -0.0585
 -0.5653  0.8474 -0.1837  1.0000 -0.2713
 -0.3441  0.4975 -0.0585 -0.2713  1.0000 :: End of G
  1.0000  1.0000  0.0000  0.0000  0.0000
  1.0000  1.0000  0.0000  0.0000  0.0000
  0.0000  0.0000  1.0000  0.0000  0.0000
  0.0000  0.0000  0.0000  1.0000  0.5000
  0.0000  0.0000  0.0000  0.5000  1.0000 :: End of H
```

10.3 Program Results

```
 G02APF Example Program Results

 Symmetrised Input Matrix G
          1       2       3       4       5
 1   1.0000 -0.0991  0.5665 -0.5653 -0.3441
 2  -0.0991  1.0000 -0.4273  0.8474  0.4975
 3   0.5665 -0.4273  1.0000 -0.1837 -0.0585
 4  -0.5653  0.8474 -0.1837  1.0000 -0.2713
 5  -0.3441  0.4975 -0.0585 -0.2713  1.0000

 Nearest Correlation Matrix X
          1       2       3       4       5
 1   1.0000 -0.0991  0.3799 -0.3791 -0.2308
 2  -0.0991  1.0000 -0.2865  0.5683  0.3336
 3   0.3799 -0.2865  1.0000 -0.1232 -0.0392
 4  -0.3791  0.5683 -0.1232  1.0000 -0.2266
 5  -0.2308  0.3336 -0.0392 -0.2266  1.0000

 Number of iterations taken:       27

 ALPHA:                             0.3294

 Norm value:                        0.6526

 THETA:                             0.1000

 Smallest eigenvalue of target:     0.8643

 Eigenvalues of X
             1          2          3          4          5
 1      0.0864     0.7431     1.0044     1.2018     1.9642
```