NAG Toolbox

nag_correg_corrmat_nearest_bounded (g02ab)

1 Purpose

nag_correg_corrmat_nearest_bounded (g02ab) 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 Syntax

```
[g, w, x, iter, feval, nrmgrd, ifail] = nag_correg_corrmat_nearest_bounded(g, opt, alpha, w, 'n', n, 'errtol', errtol, 'maxits', maxits, 'maxit', maxit)
[g, w, x, iter, feval, nrmgrd, ifail] = g02ab(g, opt, alpha, w, 'n', n, 'errtol', errtol, 'maxits', maxits, 'maxit', maxit)
```

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, α, 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 Parameters

5.1 Compulsory Input Parameters

1: **g**(ldg, **n**) – REAL (KIND=nag_wp) array

ldg, the first dimension of the array, must satisfy the constraint $ldg \geq$ **n**.

G, the initial matrix.

2: **opt** – CHARACTER(1)

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'.

3: **alpha** – REAL (KIND=nag_wp)

The value of α.

If **opt** = 'W', **alpha** need not be set.

Constraint: $0.0 <$ **alpha** < 1.0.

4: **w**(**n**) – REAL (KIND=nag_wp) array

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.

Constraint: $\mathbf{w}(i) > 0.0$, for $i = 1, 2, \ldots, n$.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **w** and the first dimension of the array **g** and the second dimension of the array **g**. (An error is raised if these dimensions are not equal.)

The order of the matrix G.

Constraint: **n** > 0.

2: **errtol** – REAL (KIND=nag_wp)

Default: 0.0

The termination tolerance for the Newton iteration. If **errtol** ≤ 0.0 then **n** $\times \sqrt{\textit{machine precision}}$ is used.

3: **maxits** – INTEGER

Default: 0

Specifies the maximum number of iterations to be used by the iterative scheme to solve the linear algebraic equations at each Newton step.

If **maxits** ≤ 0, $2 \times$ **n** is used.

4: **maxit** – INTEGER

Default: 0

Specifies the maximum number of Newton iterations.

If **maxit** ≤ 0, 200 is used.

5.3 Output Parameters

1: **g**(ldg, **n**) – REAL (KIND=nag_wp) array

2: **w**(**n**) – REAL (KIND=nag_wp) array

If **opt** = 'W' or 'B', the array is scaled so $0 < \mathbf{w}(i) \leq 1$, for $i = 1, 2, \ldots, n$.

3: **x**(ldx, **n**) – REAL (KIND=nag_wp) array

Contains the nearest correlation matrix.

4: **iter** – INTEGER

The number of Newton steps taken.

5: **feval** – INTEGER

The number of function evaluations of the dual problem.

6: **nrmgrd** – REAL (KIND=nag_wp)

The norm of the gradient of the last Newton step.

7: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

Constraint: $0.0 <$ **alpha** < 1.0.

Constraint: $ldg \geq$ **n**.

Constraint: $ldx \geq$ **n**.

Constraint: **n** > 0.

On entry, all elements of **w** were not positive.

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 function.

ifail $= 3$ (*warning*)

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

ifail $= 4$ (*warning*)

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.

ifail $= -399$

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

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

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

8 Further Comments

Arrays are internally allocated by nag_correg_corrmat_nearest_bounded (g02ab). The total size of these arrays is $12 \times \mathbf{n} + 3 \times \mathbf{n} \times \mathbf{n} + \max(2 \times \mathbf{n} \times \mathbf{n} + 6 \times \mathbf{n} + 1, 120 + 9 \times \mathbf{n})$ double elements and $5 \times \mathbf{n} + 3$ integer elements. All allocated memory is freed before return of nag_correg_corrmat_nearest_bounded (g02ab).

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

9.1 Program Text

```
     function g02ab_example

fprintf('g02ab example results\n\n');

opt = 'b';
alpha = 0.02;
g = [2, -1,  0,  0;
    -1,  2, -1,  0;
     0, -1,  2, -1;
     0,  0, -1,  2];
w = [100, 20, 20, 20];

% Calculate nearest correlation matrix
[g, w, x, iter, feval, nrmgrd, ifail] = ...
  g02ab(g, opt, alpha, w);

fprintf('\n Nearest Correlation Matrix:\n');
disp(x);
fprintf('\n Number of Newton steps taken:   %d\n', iter);
fprintf(' Number of function evaluations: %d\n', feval);
fprintf('\n\n Alpha: %30.3f\n', alpha);

[x, eig, info] = f08fa('n', 'u', x);
fprintf('\n Eigenvalues of x:\n');
disp(transpose(eig));
```

9.2 Program Results

```
     g02ab example results

 Nearest Correlation Matrix:
    1.0000   -0.9187    0.0257    0.0086
   -0.9187    1.0000   -0.3008    0.2270
    0.0257   -0.3008    1.0000   -0.8859
    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:
    0.0392    0.1183    1.6515    2.1910
```