

# NAG Toolbox for Matlab

## g02aa

### 1 Purpose

g02aa computes the nearest correlation matrix, in the Frobenius norm, to a given square, input matrix.

### 2 Syntax

```
[g, x, iter, feval, nrmgrd, ifail] = g02aa(g, errtol, maxits, maxit, 'n', n)
```

### 3 Description

A correlation matrix may be characterized as a real square matrix that is symmetric, has a unit diagonal and is positive semi-definite.

g02aa applies an inexact Newton method to a dual formulation of the problem, as described by Qi and Sun (2006). It applies the improvements suggested by Borsdorf (2007).

### 4 References

Borsdorf R (2007) A Newton Algorithm for the Nearest Correlation Matrix. *M Sc Dissertation* School of Mathematics, University of Manchester

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)** – double array

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

*G*, the initial matrix.

2: **errtol** – double scalar

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

3: **maxits** – int32 scalar

**maxits** specifies the maximum number of iterations used for the iterative scheme used to solve the linear algebraic equations at each Newton step.

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

4: **maxit** – int32 scalar

Specifies the maximum number of Newton iterations.

If **maxit**  $\leq 0$ , 200 is used.

## 5.2 Optional Input Parameters

### 1: **n** – int32 scalar

*Default:* 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 size of the matrix *G*.

*Constraint:*  $n > 0$ .

## 5.3 Input Parameters Omitted from the MATLAB Interface

ldg, ldx

## 5.4 Output Parameters

### 1: **g(ldg,n)** – double array

A symmetric matrix  $\frac{1}{2}(G + G^T)$  with the diagonal set to *I*.

### 2: **x(ldx,n)** – double array

Contains the nearest correlation matrix.

### 3: **iter** – int32 scalar

The number of Newton steps taken.

### 4: **feval** – int32 scalar

The number of function evaluations of the dual problem.

### 5: **nrmgrd** – double scalar

The norm of the gradient of the last Newton step.

### 6: **ifail** – int32 scalar

ifail = 0 unless the function detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the function:

**ifail** = 1

On entry,  $n \leq 0$ ,  
or  $ldg < n$ ,  
or  $ldx < n$ .

**ifail** = 2

The routine fails to converge in **maxit** iterations. Increase **maxit** or check the call to the routine.

**ifail** = 3

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

**ifail** = 4

An internal eigenproblem could not be solved. This should not occur. Please contact NAG with details of your call.

**ifail** = -999

Internal memory allocation failed.

## 7 Accuracy

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

## 8 Further Comments

None.

## 9 Example

```

g = [2, -1, 0, 0;
     -1, 2, -1, 0;
      0, -1, 2, -1;
      0, 0, -1, 2];
errtol = 1e-07;
maxits = int32(200);
maxit = int32(10);
[gOut, x, iter, feval, nrmgrd, ifail] = g02aa(g, errtol, maxits, maxit);
if (ifail == 0)
    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);
    if (nrmgrd > errtol)
        fprintf(' Norm of gradient of last Newton step: %6.4f\n', nrmgrd);
    end
end

```

```

Nearest Correlation Matrix
  1.0000   -0.8084    0.1916    0.1068
 -0.8084    1.0000   -0.6562    0.1916
  0.1916   -0.6562    1.0000   -0.8084
  0.1068    0.1916   -0.8084    1.0000

```

```

Number of Newton steps taken:   3
Number of function evaluations: 4

```