

# NAG Library Routine Document

## G08CJF

**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

G08CJF calculates the Anderson–Darling goodness-of-fit test statistic and its probability for the case of standard uniformly distributed data.

### 2 Specification

```
SUBROUTINE G08CJF (N, ISSORT, Y, A2, P, IFAIL)
  INTEGER          N, IFAIL
  REAL (KIND=nag_wp) Y(N), A2, P
  LOGICAL          ISSORT
```

### 3 Description

Calculates the Anderson–Darling test statistic  $A^2$  (see G08CHF) and its upper tail probability by using the approximation method of Marsaglia and Marsaglia (2004) for the case of uniformly distributed data.

### 4 References

Anderson T W and Darling D A (1952) Asymptotic theory of certain ‘goodness-of-fit’ criteria based on stochastic processes *Annals of Mathematical Statistics* **23** 193–212

Marsaglia G and Marsaglia J (2004) Evaluating the Anderson–Darling distribution *J. Statist. Software* **9(2)**

### 5 Parameters

- 1: N – INTEGER *Input*  
*On entry:*  $n$ , the number of observations.  
*Constraint:*  $N > 1$ .
- 2: ISSORT – LOGICAL *Input*  
*On entry:* set ISSORT = .TRUE. if the observations are sorted in ascending order; otherwise the routine will sort the observations.
- 3: Y(N) – REAL (KIND=nag\_wp) array *Input/Output*  
*On entry:*  $y_i$ , for  $i = 1, 2, \dots, n$ , the  $n$  observations.  
*On exit:* if ISSORT = .FALSE., the data sorted in ascending order; otherwise the array is unchanged.  
*Constraint:* if ISSORT = .TRUE., the values must be sorted in ascending order. Each  $y_i$  must lie in the interval  $(0, 1)$ .
- 4: A2 – REAL (KIND=nag\_wp) *Output*  
*On exit:*  $A^2$ , the Anderson–Darling test statistic.

5: P – REAL (KIND=nag\_wp)

*Output*

*On exit:*  $p$ , the upper tail probability for  $A^2$ .

6: 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:  $N > 1$ .

IFAIL = 3

ISSORT = .TRUE. and the data in Y is not sorted in ascending order.

IFAIL = 9

The data in Y must lie in the interval (0, 1).

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

Probabilities greater than approximately 0.09 are accurate to five decimal places; lower value probabilities are accurate to six decimal places.

## 8 Parallelism and Performance

Not applicable.

## 9 Further Comments

None.

## 10 Example

This example calculates the  $A^2$  statistic and its  $p$ -value for uniform data obtained by transforming exponential variates.

### 10.1 Program Text

```

Program g08cjfe

!      G08CJF Example Program Text
!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
!      Use nag_library, Only: g08cjf, nag_wp
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)         :: a2, mu, p
!      Integer                    :: i, ifail, n
!      Logical                    :: issort
!      .. Local Arrays ..
!      Real (Kind=nag_wp), Allocatable :: x(:), y(:)
!      .. Intrinsic Procedures ..
!      Intrinsic                  :: exp
!      .. Executable Statements ..
!      Write (nout,*) 'G08CJF Example Program Results'
!      Write (nout,*)

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

!      Read number of observations and parameter value
!      Read (nin,*) n, mu

!      Memory allocation
!      Allocate (x(n),y(n))

!      Read observations
!      Read (nin,*)(x(i),i=1,n)

!      PIT
!      Do i = 1, n
!         y(i) = 1.0E0_nag_wp - exp(-x(i)/mu)
!      End Do

!      Let g08cjf sort the uniform variates
!      issort = .False.

!      Calculate A-squared and probability
!      ifail = 0
!      Call g08cjf(n,issort,y,a2,p,ifail)

!      Results
!      Write (nout,'(1X,A,E11.4)') &
!         'H0: data from exponential distribution with mean', mu
!      Write (nout,'(1X,A,1X,F8.4)') 'Test statistic, A-squared: ', a2
!      Write (nout,'(1X,A,1X,F8.4)') 'Upper tail probability: ', p

End Program g08cjfe

```

## 10.2 Program Data

G08CJF Example Program Data

```
26 1.65 :: n, mu
0.4782745 1.2858962 1.1163891 2.0410619 2.2648109 0.0833660 1.2527554
0.4031288 0.7808981 0.1977674 3.2539440 1.8113504 1.2279834 3.9178773
1.4494309 0.1358438 1.8061778 6.0441929 0.9671624 3.2035042 0.8067364
0.4179364 3.5351774 0.3975414 0.6120960 0.1332589 :: end of observations
```

## 10.3 Program Results

G08CJF Example Program Results

```
H0: data from exponential distribution with mean 0.1650E+01
Test statistic, A-squared:    0.1830
Upper tail probability:      0.9945
```

---