# NAG Library Routine Document

# G08ECF

**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

G08ECF performs the triplets test on a sequence of observations from the interval $[0, 1]$.

## 2    Specification

```
SUBROUTINE G08ECF (CL, N, X, MSIZE, NCOUNT, LDC, EX, CHI, DF, PROB, IFAIL)

INTEGER          N, MSIZE, NCOUNT(LDC,LDC,MSIZE), LDC, IFAIL
REAL (KIND=nag_wp) X(N), EX, CHI, DF, PROB
CHARACTER(1)     CL
```

## 3    Description

G08ECF computes the statistics for performing a triplets test which may be used to investigate deviations from randomness in a sequence, $x = \{x_i : i = 1, 2, \ldots, n\}$, of $[0, 1]$ observations.

An $m$ by $m$ matrix, $C$, of counts is formed as follows. The element $c_{jkl}$ of $C$ is the number of triplets $(x_i, x_{i+1}, x_{i+2})$ for $i = 1, 4, 7, \ldots, n - 2$, such that

$$\frac{j-1}{m} \le x_i < \frac{j}{m}$$

$$\frac{k-1}{m} \le x_{i+1} < \frac{k}{m}$$

$$\frac{l-1}{m} \le x_{i+2} < \frac{l}{m}.$$

Note that all triplets formed are non-overlapping and are thus independent under the assumption of randomness.

Under the assumption that the sequence is random, the expected number of triplets for each class (i.e., each element of the count matrix) is the same; that is, the triplets should be uniformly distributed over the unit cube $[0, 1]^3$. Thus the expected number of triplets for each class is just the total number of triplets, $\sum_{j,k,l=1}^{m} c_{jkl}$, divided by the number of classes, $m^3$.

The $\chi^2$ test statistic used to test the hypothesis of randomness is defined as

$$X^2 = \sum_{j,k,l=1}^{m} \frac{(c_{jkl} - e)^2}{e},$$

where $e = \sum_{j,k,l=1}^{m} c_{jkl}/m^3 = $ expected number of triplets in each class.

The use of the $\chi^2$-distribution as an approximation to the exact distribution of the test statistic, $X^2$, improves as the length of the sequence relative to $m$ increases and hence the expected value, $e$, increases.

G08ECF may be used in two different modes:

(i)   a single call to G08ECF which computes all test statistics after counting the triplets;

(ii)  multiple calls to G08ECF with the final test statistics only being computed in the last call.

The second mode is necessary if all the data do not fit into the memory. See parameter CL in Section 5 for details on how to invoke each mode.

## 4     References

Dagpunar J (1988) *Principles of Random Variate Generation* Oxford University Press

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

Morgan B J T (1984) *Elements of Simulation* Chapman and Hall

Ripley B D (1987) *Stochastic Simulation* Wiley

## 5     Parameters

1:    CL – CHARACTER(1)                                                                                 *Input*

*On entry*: indicates the type of call to G08ECF.

CL = 'S'
        This is the one and only call to G08ECF (single call mode). All data are to be input at once.
        All test statistics are computed after counting of the triplets is complete.

CL = 'F'
        This is the first call to the routine. All initializations are carried out and the counting of
        triplets begins. The final test statistics are not computed since further calls will be made to
        G08ECF.

CL = 'I'
        This is an intermediate call during which counts of the triplets are updated. The final test
        statistics are not computed since further calls will be made to G08ECF.

CL = 'L'
        This is the last call to G08ECF. The test statistics are computed after the final counting of
        the triplets is complete.

*Constraint*: CL = 'S', 'F', 'I' or 'L'.

2:    N – INTEGER                                                                                       *Input*

*On entry*: $n$, the number of observations.

*Constraints*:

        if CL = 'S', N $\geq$ 3;
        otherwise N $\geq$ 1.

3:    X(N) – REAL (KIND=nag_wp) array                                                                   *Input*

*On entry*: the sequence of observations.

*Constraint*: $0.0 \leq X(i) \leq 1.0$, for $i = 1, 2, \ldots, n$.

4:    MSIZE – INTEGER                                                                                   *Input*

*On entry*: $m$, the size of the count matrix to be formed.

MSIZE must not be changed between calls to G08ECF.

*Constraint*: MSIZE $\geq$ 2.

5:    NCOUNT(LDC,LDC,MSIZE) – INTEGER array                                                    *Input/Output*

*On entry*: if CL = 'S' or 'F', NCOUNT need not be set.

If CL = 'I' or 'L', NCOUNT must contain the values returned by the previous call to G08ECF.

*On exit*: is an MSIZE by MSIZE by MSIZE matrix containing the counts of the number of triplets, $c_{jkl}$, for $j = 1, 2, \ldots, m$, $k = 1, 2, \ldots, m$ and $l = 1, 2, \ldots, m$.

6:     LDC – INTEGER                                                                   *Input*

*On entry*: the first dimension of the array NCOUNT and the second dimension of the array NCOUNT as declared in the (sub)program from which G08ECF is called.

*Constraint*: LDC $\geq$ MSIZE.

7:     EX – REAL (KIND=nag_wp)                                                        *Output*

*On exit*: if CL = 'S' or 'L' (i.e., if it is a final exit) then EX contains the expected number of counts for each element of the count matrix.

Otherwise EX is not set.

8:     CHI – REAL (KIND=nag_wp)                                                       *Output*

*On exit*: if CL = 'S' or 'L' (i.e., if it is a final exit) then CHI contains the $\chi^2$ test statistic, $X^2$, for testing the null hypothesis of randomness.

Otherwise CHI is not set.

9:     DF – REAL (KIND=nag_wp)                                                        *Output*

*On exit*: if CL = 'S' or 'L' (i.e., if it is a final exit) then DF contains the degrees of freedom for the $\chi^2$ statistic.

Otherwise DF is not set.

10:    PROB – REAL (KIND=nag_wp)                                                      *Output*

*On exit*: if CL = 'S' or 'L' (i.e., if it is a final exit) then PROB contains the upper tail probability associated with the $\chi^2$ test statistic, i.e., the significance level.

Otherwise PROB is not set.

11:    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, because for this routine the values of the output parameters may be useful even if IFAIL $\neq$ 0 on exit, the recommended value is $-1$. **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).

**Note**: G08ECF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL = 1

On entry, CL ≠ 'S', 'F', 'I' or 'L'.

IFAIL = 2

On entry, N < 1,
or            CL = 'S' and N < 3.

IFAIL = 3

On entry, MSIZE ≤ 1.

IFAIL = 4

On entry, LDC < MSIZE.

IFAIL = 5

On entry, $X(i) < 0.0$,
or            $X(i) > 1.0$, for some $i = 1, 2, \ldots, n$.

IFAIL = 6

No triplets were found because less than 3 observations were provided in total.

IFAIL = 7

The expected value for the counts in each element of the count matrix is less than or equal to 5.0. This implies that the $\chi^2$-distribution may not be a very good approximation to the distribution of the test statistic.

## 7    Accuracy

The computations are believed to be stable. The computations of PROB given the values of CHI and DF will obtain a relative accuracy of five significant figures for most cases.

## 8    Further Comments

If the call to G08ECF is an initial call or intermediate call with further calls to follow then any unused observations are saved for use at the beginning of the new sequence provided in the following call. Clearly any observations left over from an only or final call to G08ECF are ignored.

The time taken by the routine increases with the number of observations $n$, and also depends to some extent whether the call to G08ECF is an only, first, intermediate or last call.

## 9    Example

The following program performs the triplets test on 500 pseudorandom numbers. G08ECF is called 5 times with 100 observations on each call. The triplets are tallied into a 2 by 2 by 2 matrix.

### 9.1    Program Text

```
    Program g08ecfe

!    G08ECF Example Program Text

!    Mark 24 Release. NAG Copyright 2012.

!      .. Use Statements ..
      Use nag_library, Only: g08ecf, nag_wp, x04eaf
```

```
!       .. Implicit None Statement ..
        Implicit None
!       .. Parameters ..
        Integer, Parameter                :: nin = 5, nout = 6
!       .. Local Scalars ..
        Real (Kind=nag_wp)                :: chi, df, ex, prob
        Integer                           :: i, ifail, ldc, msize, n, nsamp, pn
        Logical                           :: bapp
        Character (1)                     :: cl
        Character (80)                    :: title
!       .. Local Arrays ..
        Real (Kind=nag_wp), Allocatable   :: x(:)
        Integer, Allocatable              :: ncount(:,:,:)
!       .. Executable Statements ..
        Write (nout,*) 'G08ECF Example Program Results'
        Write (nout,*)

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

!       Read in number of samples
        Read (nin,*) nsamp, msize

        ldc = msize
        Allocate (ncount(ldc,ldc,msize),x(1))

        If (nsamp==1) Then
          cl = 'S'
        Else
          cl = 'F'
        End If

        pn = 0
        bapp = .False.
        Do i = 1, nsamp
!         Skip run heading in data file
          Read (nin,*)

!         Read in sample size
          Read (nin,*) n

          If (n>pn) Then
!           Reallocate X if required
            Deallocate (x)
            Allocate (x(n))
            pn = n
          End If

!         Read in the sample
          Read (nin,*) x(1:n)

!         Process the sample
          ifail = -1
          Call g08ecf(cl,n,x,msize,ncount,ldc,ex,chi,df,prob,ifail)
          If (ifail==7) Then
            bapp = .True.
          Else If (ifail/=0) Then
            Go To 100
          End If

!         Adjust CL for intermediate calls
          If (i<nsamp-1) Then
            cl = 'I'
          Else
            cl = 'L'
          End If

        End Do

!       Display results
        Write (nout,*) 'Count matrix'
```

```
      Do i = 1, msize
        Write (nout,*)
        Write (title,99999) 'I = ', i
        Flush (nout)
        ifail = 0
        Call x04eaf('General',' ',msize,msize,ncount(1,1,i),ldc,title,ifail)
      End Do
      Write (nout,*)
      Write (nout,99998) 'Expected value = ', ex
      Write (nout,99997) 'CHISQ          = ', chi
      Write (nout,99998) 'DF             = ', df
      Write (nout,99997) 'Prob           = ', prob
      If (bapp) Then
        Write (nout,*) ' ** Note : expected value <= 5.0'
        Write (nout,*) &
          '      the chi square approximation may not be very good.'
      End If

100   Continue

99999 Format (1X,A,I2)
99998 Format (1X,A,F8.2)
99997 Format (1X,A,F10.4)
      End Program g08ecfe
```

## 9.2   Program Data

```
G08ECF Example Program Data
5  2                 :: NSAMP,MSIZE
## Sample 1
100                  :: N
0.11389 0.84996 0.84821 0.18431 0.14104 0.03144 0.68013 0.13297 0.27696 0.86743
0.32674 0.87990 0.85580 0.47830 0.75318 0.93643 0.19396 0.31091 0.34956 0.94923
0.18940 0.24715 0.62503 0.50406 0.05686 0.26481 0.68746 0.80387 0.48184 0.25034
0.20141 0.35062 0.58591 0.93407 0.93848 0.98496 0.66180 0.35957 0.71122 0.35875
0.96504 0.60832 0.36569 0.73499 0.25223 0.88296 0.06659 0.78113 0.40016 0.31768
0.47655 0.15008 0.20608 0.62633 0.62737 0.16400 0.44104 0.56993 0.13178 0.50499
0.44176 0.44385 0.75372 0.82178 0.60227 0.98944 0.33133 0.81067 0.40798 0.71608
0.69306 0.22144 0.47942 0.65697 0.50881 0.25223 0.82373 0.50148 0.65246 0.53275
0.92935 0.13455 0.19901 0.78844 0.14006 0.50600 0.41069 0.49703 0.47858 0.02210
0.91444 0.10784 0.54642 0.63091 0.14419 0.80457 0.51336 0.71451 0.12564 0.88051
## Sample 2
100                  :: N
0.84976 0.63094 0.46109 0.80538 0.62387 0.90670 0.09969 0.67992 0.70503 0.09560
0.69991 0.37616 0.42030 0.23665 0.28771 0.24935 0.94950 0.12008 0.66217 0.20900
0.97026 0.98368 0.80206 0.43918 0.73232 0.03533 0.97995 0.06637 0.54726 0.48530
0.68865 0.94302 0.33718 0.61014 0.70127 0.36827 0.51335 0.24476 0.14203 0.02428
0.73691 0.22192 0.40374 0.85757 0.83335 0.73309 0.05563 0.17332 0.72253 0.43291
0.77476 0.35967 0.94242 0.61337 0.43513 0.80573 0.70630 0.83115 0.24622 0.45445
0.53595 0.31476 0.87968 0.75365 0.86291 0.34051 0.62232 0.16762 0.45506 0.15561
0.76615 0.77421 0.06035 0.72290 0.93712 0.83223 0.40044 0.96575 0.73176 0.27827
0.02174 0.75326 0.82876 0.64979 0.98038 0.61054 0.87742 0.95273 0.39091 0.42146
0.89020 0.08617 0.90953 0.00416 0.70915 0.21123 0.95342 0.19269 0.68252 0.27600
## Sample 3
100                  :: N
0.40629 0.96486 0.66026 0.07134 0.35492 0.34348 0.87164 0.59746 0.43724 0.26730
0.11840 0.04604 0.49037 0.99669 0.32784 0.34772 0.93599 0.95806 0.80635 0.18897
0.60061 0.83359 0.63026 0.14084 0.05323 0.70247 0.28532 0.09572 0.36153 0.50378
0.42679 0.71801 0.51010 0.72090 0.97537 0.29919 0.30059 0.23610 0.25668 0.07510
0.92481 0.65715 0.69686 0.27840 0.20555 0.64015 0.05725 0.25120 0.32288 0.22320
0.16582 0.71466 0.34030 0.55575 0.51468 0.18013 0.74670 0.21455 0.52649 0.47487
0.85805 0.24616 0.11459 0.38690 0.83475 0.83629 0.83754 0.18998 0.46715 0.24162
0.19488 0.03281 0.39291 0.37834 0.97169 0.65229 0.88913 0.53777 0.05780 0.20468
0.33788 0.10130 0.72771 0.31306 0.74279 0.26546 0.37941 0.04878 0.03061 0.52394
0.74104 0.97192 0.04550 0.81382 0.44430 0.32402 0.06791 0.73602 0.22640 0.67260
## Sample 4
100                  :: N
0.46016 0.95901 0.37581 0.45836 0.26220 0.30389 0.46845 0.52940 0.71121 0.89187
0.33346 0.81783 0.07194 0.01163 0.63324 0.69208 0.28685 0.02491 0.97931 0.53225
0.47009 0.12105 0.80291 0.21191 0.74158 0.78269 0.30493 0.06901 0.54152 0.88463
```

```
0.60358 0.81066 0.77771 0.74140 0.65465 0.32613 0.42757 0.36584 0.42506 0.39980
0.04686 0.79805 0.53593 0.15562 0.09924 0.68011 0.61072 0.88701 0.56239 0.64343
0.19223 0.07325 0.40971 0.85265 0.27507 0.88884 0.10551 0.62646 0.11055 0.91368
0.58845 0.68942 0.29994 0.30395 0.45696 0.88127 0.38773 0.12028 0.48981 0.28535
0.84174 0.46451 0.17140 0.90827 0.49424 0.29557 0.25788 0.76838 0.19073 0.26051
0.47442 0.03224 0.32034 0.97378 0.43992 0.13338 0.45850 0.02122 0.30482 0.49427
0.89839 0.01770 0.85679 0.90157 0.29537 0.15213 0.21464 0.37237 0.86199 0.60364
## Sample 5
100                 :: N
0.66793 0.00711 0.17970 0.98702 0.50449 0.88105 0.08259 0.77263 0.06050 0.73389
0.86517 0.76088 0.40239 0.50178 0.13811 0.63441 0.91949 0.48518 0.96923 0.08820
0.14556 0.28177 0.99598 0.46908 0.83279 0.26252 0.64987 0.20426 0.41060 0.76120
0.78022 0.44662 0.04918 0.36644 0.62337 0.16849 0.63846 0.41247 0.54464 0.05721
0.79852 0.23048 0.76139 0.22493 0.45640 0.07671 0.96152 0.50771 0.02376 0.49537
0.07095 0.86385 0.71385 0.35192 0.68827 0.49737 0.44847 0.26744 0.46983 0.44270
0.78845 0.72560 0.38886 0.45552 0.45917 0.64241 0.44654 0.42665 0.01122 0.76716
0.01727 0.33687 0.02836 0.48409 0.02777 0.63643 0.59711 0.02880 0.63758 0.56746
0.41342 0.40939 0.61578 0.89186 0.70151 0.38707 0.94021 0.17271 0.27477 0.04308
0.91821 0.97517 0.57249 0.14325 0.46058 0.26434 0.85927 0.77526 0.64717 0.08314
```

## 9.3   Program Results

```
G08ECF Example Program Results

 Count matrix

  I =   1
      1   2
1  22  25
2  18  17

  I =   2
      1   2
1  23  24
2  24  13

Expected value =     20.75
CHISQ          =      6.1446
DF             =      7.00
Prob           =      0.5230
```