# NAG C Library Function Document

# nag_triplets_test (g08ecc)

## 1 Purpose

nag_triplets_test (g08ecc) performs the triplets test on a sequence of observations from the interval [0,1].

## 2 Specification

```
#include <nag.h>
#include <nagg08.h>

void nag_triplets_test (Integer n, const double x[], Integer max_count,
          double *chi, double *df, double *p, NagError *fail)
```

## 3 Description

nag_triplets_test computes the statistics for performing a triplets test which may be used to investigate deviations from randomness in a sequence 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 ($\mathbf{x}(i)$, $\mathbf{x}(i+1)$, $\mathbf{x}(i+2)$), for $i = 1, 4, 7, \ldots, n-2$, such that

$$\begin{array}{lclcl} \frac{j-1}{m} & \leq & \mathrm{X}(i) & < & \frac{j}{m} \\ \frac{k-1}{m} & \leq & \mathrm{X}(i+1) & < & \frac{k}{m} \\ \frac{l-1}{m} & \leq & \mathrm{X}(i+2) & < & \frac{l}{m}. \end{array}$$

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, hence the expected value, $e$, increases.

## 4 Parameters

1: **n** – Integer *Input*

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

*Constraint:* $\mathbf{n} \geq 3$.

2: **x[n]** – const double *Input*

*On entry:* the sequence of observations.

*Constraint:* $0.0 \leq \mathbf{x}[i-1] \leq 1.0$, for $i = 1, 2, \ldots, n$.

3: **max_count** – Integer *Input*

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

*Constraint:* **max_count** $\geq 2$.

4: **chi** – double * *Output*

*On exit:* contains the $\chi^2$ test statistic, $X^2$, for testing the null hypothesis of randomness.

5: **df** – double * *Output*

*On exit:* contains the degrees of freedom for the $\chi^2$ statistic.

6: **prob** – double * *Output*

*On exit:* contains the upper tail probability associated with the $\chi^2$ test statistic, i.e., the significance level.

7: **fail** – NagError * *Input/Output*

The NAG error parameter (see the Essential Introduction).

## 5 Error Indicators and Warnings

**NE_INT_ARG_LT**

On entry, **n** must not be less than 3: **n** = <*value*>.

**NE_INT_ARG_LE**

On entry, **max_count** must not be less than or equal to 1: **max_count** = <*value*>.

**NE_REAL_ARRAY_CONS**

On entry, **x**[<*value*>] = <*value*>
Constraint: $0 <$ **x**$[i] < 1.0$, for $i = 0, 1, \ldots, n - 1$.

**NE_G08EC_TRIPLETS**

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

**NE_G08EC_CELL**

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 test statistic.

**NE_ALLOC_FAIL**

Memory allocation failed.

**NE_INTERNAL_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

## 6 Further Comments

The time taken by the routine increases with the number of observations, $n$.

### 6.1 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 5 significant figures for most cases.

### 6.2 References

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

Ripley B D (1987) *Stochastic Simulation* Wiley

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

## 7 See Also

None.

## 8 Example

The following program performs the pairs test on 10000 pseudo-random numbers from a uniform distribution $U(0,1)$ generated by nag_random_continuous_uniform (g05cac). nag_triplets_test is called with **m** set to 5.

### 8.1 Program Text

```
/* nag_triplets_test (g08ecc) Example Program.
 *
 * Copyright 2000 Numerical Algorithms Group.
 *
 * Mark 6, 2000.
 */


#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg05.h>
#include <nagg08.h>

int main (void)
{
  double chi, df, p, *x=0, enda, endb;
  Integer i, init, exit_status=0, max_count, n;
  NagError fail;

  INIT_FAIL(fail);
  Vprintf("g08ecc Example Program Results\n");

  init = 0;
  g05cbc(init);
  n = 10000;
  if (!(x = NAG_ALLOC(n, double)))
    {
      Vprintf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }
  enda = 0.0;
  endb = 1.0;
  for (i = 0; i < n; i++)
    x[i] = g05dac(enda, endb);
  max_count = 5;
  g08ecc(n, x, max_count, &chi, &df, &p, &fail);

  if (fail.code != NE_NOERROR && fail.code != NE_G08EC_CELL)
```

```
    {
      Vprintf("Error from g08ecc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  Vprintf("\n");
  Vprintf("%s%10.4f\n", "chisq          = ", chi);
  Vprintf("%s%8.2f\n", "df             = ", df);
  Vprintf("%s%10.4f\n", "prob           = ", p);
  if (fail.code == NE_G08EC_CELL)
    Vprintf("Error from g08ecc.\n%s\n", fail.message);
 END:
  if (x) NAG_FREE(x);
  return exit_status;
}
```

## 8.2 Program Data

None.

## 8.3 Program Results

```
g08ecc Example Program Results

chisq          =   135.0093
df             =   124.00
prob           =     0.2353
```