NAG C Library Function Document

nag_rngs_arma_time_series (g05pac)

1 Purpose

nag_rngs_arma_time_series (g05pac) generates a realisation of a univariate time series from an autoregressive moving average (ARMA) model. The realisation may be continued or a new realisation generated at subsequent calls to nag_rngs_arma_time_series (g05pac).

2 Specification

```
void nag_rngs_arma_time_series (Integer mode, double xmean, Integer p,
     const double phi[], Integer q, const double theta[], double avar, double *var,
     Integer n, double x[], Integer igen, Integer iseed[], double r[], NagError *fail)
```

3 Description

Let the vector x_t, denote a time series which is assumed to follow an autoregressive moving average (ARMA) model of the form:

$$\begin{aligned} x_t - \mu = \quad & \phi_1(x_{t-1}-\mu) + \phi_2(x_{t-2}-\mu) + \ldots + \phi_p(x_{t-p}-\mu) + \\ & \epsilon_t - \theta_1\epsilon_{t-1} - \theta_2\epsilon_{t-2} - \ldots - \theta_q\epsilon_{t-q} \end{aligned} \tag{1}$$

where ϵ_t, is a residual series of independent random perturbations assumed to be Normally distributed with zero mean and variance σ^2. The parameters $\{\phi_i\}$, for $i = 1, 2, \ldots, p$, are called the autoregressive (AR) parameters, and $\{\theta_j\}$, for $j = 1, 2, \ldots, q$, the moving average (MA) parameters. The parameters in the model are thus the p ϕ-values, the q θ-values, the mean μ and the residual variance σ^2.

nag_rngs_arma_time_series (g05pac) sets up a reference vector containing initial values corresponding to a stationary position using the method described in Tunnicliffe–Wilson (1979). The function can then return a realisation of $x_1, x_2, \ldots, x_n$. On a successful exit, the recent history is updated and saved in the reference vector **r** so that nag_rngs_arma_time_series (g05pac) may be called again to generate a realisation of $x_{n+1}, x_{n+2}, \ldots$,etc. See the description of the parameter **mode** in Section 5 for details.

One of the initialisation functions nag_rngs_init_repeatable (g05kbc) (for a repeatable sequence if computed sequentially) or nag_rngs_init_nonrepeatable (g05kcc) (for a non-repeatable sequence) must be called prior to the first call to nag_rngs_arma_time_series (g05pac).

4 References

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

Tunnicliffe–Wilson G (1979) Some efficient computational procedures for high order ARMA models *J. Statist. Comput. Simulation* **8** 301–309

5 Parameters

1: **mode** – Integer *Input*

On entry: a code for selecting the operation to be performed by the function:

mode = 0

Set up reference vector only.

mode = 1

Generate terms in the time series using reference vector set up in a prior call to nag_rngs_arma_time_series (g05pac).

mode $= 2$

Set up reference vector and generate terms in the time series.

Constraint: $0 \leq$ **mode** ≤ 2.

2: **xmean** – double *Input*

On entry: the mean of the time series.

3: **p** – Integer *Input*

On entry: the number of autoregressive coefficients supplied, p.

Constraint: **p** ≥ 0.

4: **phi**$[dim]$ – const double *Input*

Note: the dimension, dim, of the array **phi** must be at least $\max(1, \mathbf{p})$.

On entry: the autoregressive coefficients of the model, $\phi_1, \phi_2, \ldots, \phi_p$.

5: **q** – Integer *Input*

On entry: the number of moving average coefficients supplied, q.

Constraint: **q** ≥ 0.

6: **theta**$[dim]$ – const double *Input*

Note: the dimension, dim, of the array **theta** must be at least $\max(1, \mathbf{q})$.

On entry: the moving average coefficients of the model, $\theta_1, \theta_2, \ldots, \theta_q$.

7: **avar** – double *Input*

On entry: the variance of the normal perturbations, σ^2.

Constraint: **avar** ≥ 0.0.

8: **var** – double * *Output*

On exit: the proportion of the variance of a term in the series that is due to the moving-average (error) terms in the model. The smaller this is, the nearer is the model to non-stationarity.

9: **n** – Integer *Input*

On entry: the number of observations to be generated, n.

Constraint: **n** ≥ 0.

10: **x**$[dim]$ – double *Output*

Note: the dimension, dim, of the array **x** must be at least $\max(1, \mathbf{n})$.

On exit: contains the next n observations from the time series.

11: **igen** – Integer *Input*

On entry: must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialisation by a prior call to one of the functions nag_rngs_init_repeatable (g05kbc) or nag_rngs_init_nonrepeatable (g05kcc).

12: **iseed**$[4]$ – Integer *Input/Output*

On entry: contains values which define the current state of the selected generator.

On exit: contains updated values defining the new state of the selected generator.

13: **r**[*dim*] – double *Input/Output*

Note: the dimension, *dim*, of the array **r** must be at least $\mathbf{p} + \mathbf{q} + 5 + \max(\mathbf{p}, \mathbf{q} + 1)$.

On exit: the reference vector.

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

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

On entry, **p** = ⟨*value*⟩.
Constraint: $\mathbf{p} \geq 0$.

On entry, **q** = ⟨*value*⟩.
Constraint: $\mathbf{q} \geq 0$.

On entry, **mode** = ⟨*value*⟩.
Constraint: $0 \leq \mathbf{mode} > \leq$.

On entry, **n** = ⟨*value*⟩.
Constraint: $\mathbf{n} \geq 0$.

NE_REAL

On entry, **avar** = ⟨*value*⟩.
Constraint: $\mathbf{avar} \geq 0.0$.

NE_STATIONARY_AR

phi does not define a stationary autoregresive process.

NE_BAD_PARAM

On entry, parameter ⟨*value*⟩ had an illegal value.

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.

7 Accuracy

The errors in the initialisation process should be very much smaller than the error term; see Tunnicliffe–Wilson (1979).

8 Further Comments

The time taken by nag_rngs_arma_time_series (g05pac) is essentially of order $(\mathbf{p})^2$.

Note: nag_rngs_init_repeatable (g05kbc) and nag_rngs_init_nonrepeatable (g05kcc) must be used with care if this function is used as well. The reference vector, as mentioned before, contains a copy of the recent history of the series. This will not be altered properly by calls to any of the above functions. A call to nag_rngs_init_repeatable (g05kbc) or nag_rngs_init_nonrepeatable (g05kcc) should be followed by a call to nag_rngs_arma_time_series (g05pac) with **mode** = 0 to re-initialise the time series reference vector in use. To maintain repeatability with nag_rngs_init_repeatable (g05kbc), the calls to nag_rngs_arma_time_series (g05pac) should be performed in the same order and at the same point or points in the simulation every time nag_rngs_init_repeatable (g05kbc) is used. When the generator state is saved and restored using the parameters **igen** and **iseed**, the time series reference vector must be saved and restored as well.

The ARMA model for a time series can also be written as:

$$(x_n - E) = A_1(x_{n-1} - E) + \cdots + A_{NA}(x_{n-NA} - E) + B_1 a_n + \cdots + B_{NB} a_{n-NB+1}$$

where

x_n is the observed value of the time series at time n,

NA is the number of autoregressive parameters, A_i,

NB is the number of moving average parameters, B_i,

E is the mean of the time series,

and

a_t is a series of independent random Standard Normal perturbations.

This is related to the form given in Section 3 by:

$B_1^2 = \sigma^2$,

$B_{i+1} = -\theta_i \sigma = -\theta_i B_1, \quad i = 1, 2, \ldots, q$,

$NB = q + 1$,

$E = c$,

$A_i = \phi_i, \quad i = 1, 2, \ldots, p$,

$NA = p$.

9 Example

This example program calls nag_rngs_arma_time_series (g05pac) to set up the reference vector for an autoregressive model after initialisation by nag_rngs_init_repeatable (g05kbc). The model is given by

$$x_t = 0.4x_{t-1} + 0.2x_{t-2} + \epsilon_t$$

where ϵ_t is a series of independent random Normal perturbations with variance 1.0. nag_rngs_arma_time_series (g05pac) is then called generate a sample of ten observations, which are printed.

9.1 Program Text

```
/* nag_rngs_arma_time_series(g05pac) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

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

int main(void)
{
  /* Scalars */
  double   avar, var, xmean;
  Integer  i, igen, ip, iq, n, nr;
  Integer  exit_status=0;
  NagError fail;

  /* Arrays */
  double   *phi=0, *r=0, *theta=0, *x=0;
  Integer  iseed[4];

  INIT_FAIL(fail);
  Vprintf("g05pac Example Program Results\n\n");
```

```
  ip=2;
  iq=0;
  n=10;
  nr=ip+iq+5+ip;

  /* allocate memory */
  if ( !(phi = NAG_ALLOC(ip, double)) ||
       !(r = NAG_ALLOC(nr, double)) ||
       !(theta = NAG_ALLOC(1, double)) ||
       !(x = NAG_ALLOC(n, double)) )
    {
      Vprintf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  /* Set the ARMA model parameters */
  xmean = 0.0;
  phi[0] = 0.4;
  phi[1] = 0.2;
  avar = 1.0;

  /* Initialise the seed to a repeatable sequence */
  iseed[0] = 1762543;
  iseed[1] = 9324783;
  iseed[2] = 4234401;
  iseed[3] = 742355;
  /* igen identifies the stream. */
  igen = 1;
  g05kbc(&igen, iseed);

  /* Set up the reference vector */
  g05pac(0, xmean, ip, phi, iq, theta, avar, &var, n, x, igen,
         iseed, r, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from g05pac.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  /* Generate a sample of 10 observations */
  g05pac(1, xmean, ip, phi, iq, theta, avar, &var, n, x, igen,
         iseed, r, &fail);
  if (fail.code != NE_NOERROR)
    {
      Vprintf("Error from g05pac.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
  for (i = 0; i < n; ++i)
    {
      Vprintf("%12.4f\n", x[i]);
    }
 END:
  if (phi) NAG_FREE(phi);
  if (r) NAG_FREE(r);
  if (theta) NAG_FREE(theta);
  if (x) NAG_FREE(x);
  return exit_status;
}
```

9.2 Program Data

None.

9.3 Program Results

```
g05pac Example Program Results

     -1.0654
     -0.2828
     -2.0924
     -2.3304
     -2.5998
     -1.7143
     -2.4882
     -1.3882
     -2.2722
     -1.8806
```