NAG Library Routine Document

G05ZTF

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

G05ZTF produces realisations of a fractional Brownian motion, using the circulant embedding method. The square roots of the extended covariance matrix (or embedding matrix) need to be input, and can be calculated using G05ZMF or G05ZNF.

2 Specification

```
SUBROUTINE G05ZTF (NS, S, M, XMAX, H, LAM, RHO, STATE, Z, XX, IFAIL)

INTEGER             NS, S, M, STATE(*), IFAIL
REAL (KIND=nag_wp) XMAX, H, LAM(M), RHO, Z(NS+1,S), XX(NS+1)
```

3 Description

The routines G05ZMF or G05ZNF and G05ZTF are used to simulate a fractional Brownian motion process with Hurst parameter H over an interval $[0, x_{\text{max}}]$, using a set of equally spaced gridpoints. Fractional Brownian motion itself cannot be simulated directly using this method, since it is not a stationary Gaussian random field; however its increments can be simulated like a stationary Gaussian random field. The circulant embedding method is described in the documentation for G05ZMF or G05ZNF.

G05ZTF takes the square roots of the eigenvalues of the embedding matrix as returned by G05ZMF or G05ZNF, and its size M, as input and outputs S realisations of the fractional Brownian motion in Z.

One of the initialization routines G05KFF (for a repeatable sequence if computed sequentially) or G05KGF (for a non-repeatable sequence) must be called prior to the first call to G05ZTF.

4 References

Dietrich C R and Newsam G N (1997) Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix *SIAM J. Sci. Comput.* **18** 1088–1107

Schlather M (1999) Introduction to positive definite functions and to unconditional simulation of random fields *Technical Report ST 99–10* Lancaster University

Wood A T A and Chan G (1994) Simulation of stationary Gaussian processes in $[0,1]^d$ *Journal of Computational and Graphical Statistics* **3(4)** 409–432

5 Parameters

1: NS – INTEGER *Input*

On entry: the number of sample points (grid points) to be generated in realisations of the increments of the fractional Brownian motion. This must be the same value as supplied to G05ZMF or G05ZNF when calculating the eigenvalues of the embedding matrix.

Constraint: $\text{NS} \geq 1$.

2: S – INTEGER *Input*

On entry: the number of realisations of the fractional Brownian motion to simulate.

Constraint: $\text{S} \geq 1$.

3: M – INTEGER *Input*

On entry: the size of the embedding matrix, as returned by G05ZMF or G05ZNF.

Constraint: $\mathrm{M} \geq \max(1, 2(\mathrm{NS} - 1))$.

4: XMAX – REAL (KIND=nag_wp) *Input*

On entry: the upper bound for the interval over which the fractional Brownian motion is to be simulated, as returned by G05ZMF or G05ZNF.

Constraint: $\mathrm{XMAX} > 0.0$.

5: H – REAL (KIND=nag_wp) *Input*

On entry: the Hurst parameter for the fractional Brownian motion. This must be the same value as supplied to G05ZMF or G05ZNF when calculating the eigenvalues of the embedding matrix.

Constraint: $0.0 < \mathrm{H} < 1.0$.

6: LAM(M) – REAL (KIND=nag_wp) array *Input*

On entry: contains the square roots of the eigenvalues of the embedding matrix, as returned by G05ZNF.

Constraint: $\mathrm{LAM}(i) = 0$, $i = 1, 2, \ldots, \mathrm{M}$.

7: RHO – REAL (KIND=nag_wp) *Input*

On entry: indicates the scaling of the covariance matrix, as returned by G05ZMF or G05ZNF.

Constraint: $0.0 < \mathrm{RHO} \leq 1.0$.

8: STATE($*$) – INTEGER array *Communication Array*

Note: the actual argument supplied must be the array STATE supplied to the initialization routines G05KFF or G05KGF.

On entry: contains information on the selected base generator and its current state.

On exit: contains updated information on the state of the generator.

9: Z(NS + 1,S) – REAL (KIND=nag_wp) array *Output*

On exit: contains the realisations of the fractional Brownian motion. Each column of Z contains one realisation of the fractional Brownian motion, with $\mathrm{Z}(i,j)$, for $j = 1, 2, \ldots, \mathrm{S}$, corresponding to the gridpoint $\mathrm{XX}(i)$.

10: XX(NS + 1) – REAL (KIND=nag_wp) array *Output*

On exit: the gridpoints at which values of the fractional Brownian motion are output. The first gridpoint is always zero, and the subsequent NS gridpoints represent the equispaced steps towards the last gridpoint, XMAX. Note that in G05ZMF and G05ZNF, the returned NS sample points are the mid-points of the grid returned in XX here.

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, 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: $\mathrm{IFAIL} = 0$ 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, NS = ⟨*value*⟩.
Constraint: NS $\geq$ 1.

IFAIL = 2

On entry, S = ⟨*value*⟩.
Constraint: S $\geq$ 1.

IFAIL = 3

On entry, M = ⟨*value*⟩, and NS = ⟨*value*⟩.
Constraint: M $\geq \max(1, 2(\text{NS} - 1))$.

IFAIL = 4

On entry, XMAX = ⟨*value*⟩.
Constraint: XMAX > 0.0.

IFAIL = 5

On entry, H = ⟨*value*⟩.
Constraint: 0.0 < H < 1.0.

IFAIL = 6

On entry, at least one element of LAM was negative.
Constraint: all elements of LAM must be non-negative.

IFAIL = 7

On entry, RHO = ⟨*value*⟩.
Constraint: 0.0 < RHO $\leq$ 1.0.

IFAIL = 8

On entry, STATE vector has been corrupted or not initialized.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example calls G05ZTF to generate 5 realisations of a fractional Brownian motion over 10 steps from $x = 0.0$ to $x = 2.0$ using eigenvalues generated by G05ZNF with ICOV1 = 14.

9.1 Program Text

```
!   G05ZTF Example Program Text

!   Mark 24 Release. NAG Copyright 2012.

    Program g05ztfe

!     G05ZTF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g05znf, g05ztf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                   :: lenst = 17, nin = 5, nout = 6,   &
                                              npmax = 4
!     .. Local Scalars ..
      Real (Kind=nag_wp)                   :: h, rho, var, xmax, xmin
      Integer                              :: approx, icorr, icount, icov1,    &
                                              ifail, m, maxm, np, ns, pad, s
!     .. Local Arrays ..
      Real (Kind=nag_wp)                   :: eig(3), params(npmax)
      Real (Kind=nag_wp), Allocatable      :: lam(:), xx(:), yy(:), z(:,:)
      Integer                              :: state(lenst)
!     .. Executable Statements ..
      Write (nout,*) 'G05ZTF Example Program Results'
      Write (nout,*)

!     Set fixed problem specifications for simulating fractional Brownian
!     motion.
      icov1 = 14
      np = 2
      xmin = 0.0_nag_wp
      var = 1.0_nag_wp

!     Get other problem specifications from data file
      Call read_input_data(params,xmax,ns,maxm,icorr,pad,s)

      Allocate (lam(maxm),xx(ns))

!     Get square roots of the eigenvalues of the embedding matrix
      ifail = 0
      Call g05znf(ns,xmin,xmax,maxm,var,icov1,np,params,pad,icorr,lam,xx,m, &
        approx,rho,icount,eig,ifail)

      Call display_embedding_results(approx,m,rho,eig,icount)


!     Initialize state array
      Call initialize_state(state)

      Allocate (yy(ns+1),z(ns+1,s))

!     Compute s fractional Brownian Motion realisations.
      h = params(1)
      ifail = 0
      Call g05ztf(ns,s,m,xmax,h,lam,rho,state,z,yy,ifail)

      Call display_realizations(ns,s,yy,z)

    Contains
      Subroutine read_input_data(params,xmax,ns,maxm,icorr,pad,s)

!       .. Implicit None Statement ..
        Implicit None
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (Out)     :: xmax
        Integer, Intent (Out)                :: icorr, maxm, ns, pad, s
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out)     :: params(npmax)
```

```
!       .. Intrinsic Procedures ..
        Intrinsic                            :: real
!       .. Executable Statements ..
!       Skip heading in data file
        Read (nin,*)

!       Read in the Hurst parameter, H
        Read (nin,*) params(1)

!       Read in domain endpoint
        Read (nin,*) xmax

!       Read in number of sample points
        Read (nin,*) ns

        params(2) = xmax/(real(ns,kind=nag_wp))

!       Read in maximum size of embedding matrix
        Read (nin,*) maxm

!       Read in choice of scaling in case of approximation
        Read (nin,*) icorr

!       Read in choice of padding
        Read (nin,*) pad

!       Read in number of realization samples to be generated
        Read (nin,*) s

        Return

      End Subroutine read_input_data

      Subroutine display_embedding_results(approx,m,rho,eig,icount)

!       .. Implicit None Statement ..
        Implicit None
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: rho
        Integer, Intent (In)                 :: approx, icount, m
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: eig(3)
!       .. Executable Statements ..
!       Display size of embedding matrix
        Write (nout,*)
        Write (nout,99999) 'Size of embedding matrix = ', m

!       Display approximation information if approximation used
        Write (nout,*)
        If (approx==1) Then
          Write (nout,*) 'Approximation required'
          Write (nout,*)
          Write (nout,99998) 'RHO = ', rho
          Write (nout,99997) 'EIG = ', eig(1:3)
          Write (nout,99999) 'ICOUNT = ', icount
        Else
          Write (nout,*) 'Approximation not required'
        End If

        Return

99999   Format (1X,A,I7)
99998   Format (1X,A,F10.5)
99997   Format (1X,A,3(F10.5,1X))

      End Subroutine display_embedding_results

      Subroutine initialize_state(state)

!       .. Use Statements ..
        Use nag_library, Only: g05kff
```

```
!       .. Implicit None Statement ..
        Implicit None
!       .. Parameters ..
        Integer, Parameter                   :: genid = 1, inseed = 14965,     &
                                                lseed = 1, subid = 1
!       .. Array Arguments ..
        Integer, Intent (Out)                :: state(lenst)
!       .. Local Scalars ..
        Integer                              :: ifail, lstate
!       .. Local Arrays ..
        Integer                              :: seed(lseed)
!       .. Executable Statements ..
!       Initialize the generator to a repeatable sequence
        lstate = lenst
        seed(1) = inseed
        ifail = 0
        Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

      End Subroutine initialize_state

      Subroutine display_realizations(ns,s,yy,z)

!       .. Use Statements ..
        Use nag_library, Only: x04cbf
!       .. Implicit None Statement ..
        Implicit None
!       .. Parameters ..
        Integer, Parameter                   :: indent = 0, ncols = 80
        Character (1), Parameter             :: charlab = 'C', intlab = 'I',   &
                                                matrix = 'G', unit = 'n'
        Character (5), Parameter             :: form = 'F10.5'
!       .. Scalar Arguments ..
        Integer, Intent (In)                 :: ns, s
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: yy(ns+1), z(ns+1,s)
!       .. Local Scalars ..
        Integer                              :: i, ifail
        Character (61)                       :: title
!       .. Local Arrays ..
        Character (1)                        :: clabs(0)
        Character (6), Allocatable           :: rlabs(:)
!       .. Executable Statements ..
        Allocate (rlabs(ns+1))

!       Set row labels to mesh points (column label is realization number).
        Do i = 1, ns + 1
          Write (rlabs(i),99999) yy(i)
        End Do

!       Display random field results
        title = &
          'Fractional Brownian motion realisations (x coordinate first):'
        Write (nout,*)
        ifail = 0
        Call x04cbf(matrix,unit,ns+1,s,z,ns+1,form,title,charlab,rlabs,intlab, &
          clabs,ncols,indent,ifail)

99999   Format (F6.1)

      End Subroutine display_realizations

    End Program g05ztfe
```

9.2 Program Data

```
G05ZTF Example Program Data
    0.35       : h
    2          : xmax
   10          : ns
 2048          : maxm
    2          : icorr
    1          : pad
    5          : s
```

9.3 Program Results

```
 G05ZTF Example Program Results


 Size of embedding matrix =      32

 Approximation not required

 Fractional Brownian motion realisations (x coordinate first):
              1          2          3          4          5
 0.0    0.00000    0.00000    0.00000    0.00000    0.00000
 0.2   -0.52650   -0.16159   -0.96224   -0.40096    0.65803
 0.4   -1.81085   -0.85811   -1.43661    0.03947    0.99671
 0.6   -1.65690   -0.74802   -0.61733   -0.34685    0.05141
 0.8   -1.72240   -0.14958    0.14996    0.18134    0.26567
 1.0   -2.20349    0.46219    0.70982    0.66405    0.40706
 1.2   -2.38542    0.52085    0.36330    0.31831    0.81515
 1.4   -3.13939    0.68433    0.79826   -0.35408    1.12296
 1.6   -3.54602    0.64413    0.85751   -0.39303    1.14220
 1.8   -4.09082    1.67048    0.06038    0.30181    1.30350
 2.0   -2.97487    1.72275   -0.67253   -0.07439    1.57169
```