# NAG Library Routine Document

# G05ZRF

**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

G05ZRF performs the setup required in order to simulate stationary Gaussian random fields in two dimensions, for a preset variogram, using the *circulant embedding method*. Specifically, the eigenvalues of the extended covariance matrix (or embedding matrix) are calculated, and their square roots output, for use by G05ZSF, which simulates the random field.

## 2 Specification

```
SUBROUTINE G05ZRF (NS, XMIN, XMAX, YMIN, YMAX, MAXM, VAR, ICOV2, NORM, NP,       &
                   PARAMS, PAD, ICORR, LAM, XX, YY, M, APPROX, RHO, ICOUNT,      &
                   EIG, IFAIL)
INTEGER            NS(2), MAXM(2), ICOV2, NORM, NP, PAD, ICORR, M(2),             &
                   APPROX, ICOUNT, IFAIL
REAL (KIND=nag_wp) XMIN, XMAX, YMIN, YMAX, VAR, PARAMS(NP),                       &
                   LAM(MAXM(1)*MAXM(2)), XX(NS(1)), YY(NS(2)), RHO, EIG(3)
```

## 3 Description

A two-dimensional random field $Z(\mathbf{x})$ in $\mathbb{R}^2$ is a function which is random at every point $\mathbf{x} \in \mathbb{R}^2$, so $Z(\mathbf{x})$ is a random variable for each $\mathbf{x}$. The random field has a mean function $\mu(\mathbf{x}) = \mathbb{E}[Z(\mathbf{x})]$ and a symmetric positive semidefinite covariance function $C(\mathbf{x},\mathbf{y}) = \mathbb{E}[(Z(\mathbf{x}) - \mu(\mathbf{x}))(Z(\mathbf{y}) - \mu(\mathbf{y}))]$. $Z(\mathbf{x})$ is a Gaussian random field if for any choice of $n \in \mathbb{N}$ and $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^2$, the random vector $[Z(\mathbf{x}_1), \dots, Z(\mathbf{x}_n)]^{\mathrm{T}}$ follows a multivariate Gaussian distribution, which would have a mean vector $\tilde{\boldsymbol{\mu}}$ with entries $\tilde{\mu}_i = \mu(\mathbf{x}_i)$ and a covariance matrix $\tilde{C}$ with entries $\tilde{C}_{ij} = C(\mathbf{x}_i, \mathbf{x}_j)$. A Gaussian random field $Z(\mathbf{x})$ is stationary if $\mu(\mathbf{x})$ is constant for all $\mathbf{x} \in \mathbb{R}^2$ and $C(\mathbf{x},\mathbf{y}) = C(\mathbf{x}+\mathbf{a},\mathbf{y}+\mathbf{a})$ for all $\mathbf{x},\mathbf{y},\mathbf{a} \in \mathbb{R}^2$ and hence we can express the covariance function $C(\mathbf{x},\mathbf{y})$ as a function $\gamma$ of one variable: $C(\mathbf{x},\mathbf{y}) = \gamma(\mathbf{x}-\mathbf{y})$. $\gamma$ is known as a variogram (or more correctly, a semivariogram) and includes the multiplicative factor $\sigma^2$ representing the variance such that $\gamma(0) = \sigma^2$.

The routines G05ZRF and G05ZSF are used to simulate a two-dimensional stationary Gaussian random field, with mean function zero and variogram $\gamma(\mathbf{x})$, over a domain $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$, using an equally spaced set of $N_1 \times N_2$ gridpoints; $N_1$ gridpoints in the $x$-direction and $N_2$ gridpoints in the $y$-direction. The problem reduces to sampling a Gaussian random vector $\mathbf{X}$ of size $N_1 \times N_2$, with mean vector zero and a symmetric covariance matrix $A$, which is an $N_2$ by $N_2$ block Toeplitz matrix with Toeplitz blocks of size $N_1$ by $N_1$. Since $A$ is in general expensive to factorize, a technique known as the *circulant embedding method* is used. $A$ is embedded into a larger, symmetric matrix $B$, which is an $M_2$ by $M_2$ block circulant matrix with circulant blocks of size $M_1$ by $M_1$, where $M_1 \geq 2(N_1 - 1)$ and $M_2 \geq 2(N_2 - 1)$. $B$ can now be factorized as $B = W\Lambda W^* = R^*R$, where $W$ is the two-dimensional Fourier matrix ($W^*$ is the complex conjugate of $W$), $\Lambda$ is the diagonal matrix containing the eigenvalues of $B$ and $R = \Lambda^{\frac{1}{2}}W^*$. $B$ is known as the embedding matrix. The eigenvalues can be calculated by performing a discrete Fourier transform of the first row (or column) of $B$ and multiplying by $M_1 \times M_2$, and so only the first row (or column) of $B$ is needed – the whole matrix does not need to be formed.

As long as all of the values of $\Lambda$ are non-negative (i.e., $B$ is positive semidefinite), $B$ is a covariance matrix for a random vector $\mathbf{Y}$ which has $M_2$ blocks of size $M_1$. Two samples of $\mathbf{Y}$ can now be simulated from the real and imaginary parts of $R^*(\mathbf{U} + i\mathbf{V})$, where $\mathbf{U}$ and $\mathbf{V}$ have elements from the standard Normal distribution. Since $R^*(\mathbf{U} + i\mathbf{V}) = W\Lambda^{\frac{1}{2}}(\mathbf{U} + i\mathbf{V})$, this calculation can be done using a discrete Fourier transform of the vector $\Lambda^{\frac{1}{2}}(\mathbf{U} + i\mathbf{V})$. Two samples of the random vector $\mathbf{X}$ can now be recovered by taking the first $N_1$ elements of the first $N_2$ blocks of each sample of $\mathbf{Y}$ – because the original covariance matrix $A$ is embedded in $B$, $\mathbf{X}$ will have the correct distribution.

If $B$ is not positive semidefinite, larger embedding matrices $B$ can be tried; however if the size of the matrix would have to be larger than MAXM, an approximation procedure is used. We write $\Lambda = \Lambda_+ + \Lambda_-$, where $\Lambda_+$ and $\Lambda_-$ contain the non-negative and negative eigenvalues of $B$ respectively. Then $B$ is replaced by $\rho B_+$ where $B_+ = W\Lambda_+ W^*$ and $\rho \in (0,1]$ is a scaling factor. The error $\epsilon$ in approximating the distribution of the random field is given by

$$\epsilon = \sqrt{\frac{(1-\rho)^2 \operatorname{trace} \Lambda + \rho^2 \operatorname{trace} \Lambda_-}{M}}.$$

Three choices for $\rho$ are available, and are determined by the input parameter ICORR:

setting ICORR $= 0$ sets

$$\rho = \frac{\operatorname{trace} \Lambda}{\operatorname{trace} \Lambda_+},$$

setting ICORR $= 1$ sets

$$\rho = \sqrt{\frac{\operatorname{trace} \Lambda}{\operatorname{trace} \Lambda_+}},$$

setting ICORR $= 2$ sets $\rho = 1$.

G05ZRF finds a suitable positive semidefinite embedding matrix $B$ and outputs its sizes in the vector M and the square roots of its eigenvalues in LAM. If approximation is used, information regarding the accuracy of the approximation is output. Note that only the first row (or column) of $B$ is actually formed and stored.

## 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 (1997) Algorithm AS 312: An Algorithm for Simulating Stationary Gaussian Random Fields *Journal of the Royal Statistical Society, Series C (Applied Statistics) (Volume 46)* **1** 171–181

## 5 Parameters

1: NS(2) – INTEGER array *Input*

*On entry*: the number of sample points (gridpoints) to use in each direction, with NS(1) sample points in the $x$-direction, $N_1$ and NS(2) sample points in the $y$-direction, $N_2$. The total number of sample points on the grid is therefore NS(1) $\times$ NS(2).

*Constraints*:

NS(1) $\geq 1$;
NS(2) $\geq 1$.

2: XMIN – REAL (KIND=nag_wp) *Input*

*On entry*: the lower bound for the $x$-coordinate, for the region in which the random field is to be simulated.

*Constraint*: XMIN $<$ XMAX.

3: XMAX – REAL (KIND=nag_wp) *Input*

*On entry*: the upper bound for the $x$-coordinate, for the region in which the random field is to be simulated.

*Constraint*: XMIN < XMAX.

4: YMIN – REAL (KIND=nag_wp) *Input*

*On entry*: the lower bound for the $y$-coordinate, for the region in which the random field is to be simulated.

*Constraint*: YMIN < YMAX.

5: YMAX – REAL (KIND=nag_wp) *Input*

*On entry*: the upper bound for the $y$-coordinate, for the region in which the random field is to be simulated.

*Constraint*: YMIN < YMAX.

6: MAXM(2) – INTEGER array *Input*

*On entry*: determines the maximum size of the circulant matrix to use – a maximum of MAXM(1) elements in the $x$-direction, and a maximum of MAXM(2) elements in the $y$-direction. The maximum size of the circulant matrix is thus MAXM(1)$\times$MAXM(2).

*Constraint*: $\mathrm{MAXM}(i) \geq 2^k$, where $k$ is the smallest integer satisfying $2^k \geq 2(\mathrm{NS}(i) - 1)$, for $i = 1, 2$.

7: VAR – REAL (KIND=nag_wp) *Input*

*On entry*: the multiplicative factor $\sigma^2$ of the variogram $\gamma(\mathbf{x})$.

*Constraint*: VAR $\geq 0.0$.

8: ICOV2 – INTEGER *Input*

*On entry*: determines which of the preset variograms to use. The choices are given below. Note that $x' = \left\| \frac{x}{\ell_1}, \frac{y}{\ell_2} \right\|$, where $\ell_1$ and $\ell_2$ are correlation lengths in the $x$ and $y$ directions respectively and are parameters for most of the variograms, and $\sigma^2$ is the variance specified by VAR.

ICOV2 = 1

Symmetric stable variogram

$$\gamma(\mathbf{x}) = \sigma^2 \exp\left(-\left(x'\right)^\nu\right),$$

where

$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$,
$\nu = \mathrm{PARAMS}(3)$, $0 < \nu \leq 2$.

ICOV2 = 2

Cauchy variogram

$$\gamma(\mathbf{x}) = \sigma^2 \left(1 + \left(x'\right)^2\right)^{-\nu},$$

where

$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$,
$\nu = \mathrm{PARAMS}(3)$, $\nu > 0$.

ICOV2 $= 3$
Differential variogram with compact support

$$\gamma(\mathbf{x}) = \begin{cases} \sigma^2\left(1 + 8x' + 25(x')^2 + 32(x')^3\right)(1 - x')^8, & x' < 1, \\ 0, & x' \geq 1, \end{cases}$$

where

$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$.

ICOV2 $= 4$
Exponential variogram

$$\gamma(\mathbf{x}) = \sigma^2 \exp(-x'),$$

where

$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$.

ICOV2 $= 5$
Gaussian variogram

$$\gamma(\mathbf{x}) = \sigma^2 \exp\left(-(x')^2\right),$$

where

$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$.

ICOV2 $= 6$
Nugget variogram

$$\gamma(\mathbf{x}) = \begin{cases} \sigma^2, & \mathbf{x} = \mathbf{0}, \\ 0, & \mathbf{x} \neq \mathbf{0}. \end{cases}$$

No parameters need be set for this value of ICOV2.

ICOV2 $= 7$
Spherical variogram

$$\gamma(\mathbf{x}) = \begin{cases} \sigma^2\left(1 - 1.5x' + 0.5(x')^3\right), & x' < 1, \\ 0, & x' \geq 1, \end{cases}$$

where

$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$.

ICOV2 $= 8$
Bessel variogram

$$\gamma(\mathbf{x}) = \sigma^2 \frac{2^\nu \Gamma(\nu + 1) J_\nu(x')}{(x')^\nu},$$

where

$J_\nu(\cdot)$ is the Bessel function of the first kind,
$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$,
$\nu = \mathrm{PARAMS}(3)$, $\nu \geq 0$.

ICOV2 = 9

Hole effect variogram

$$\gamma(\mathbf{x}) = \sigma^2 \frac{\sin(x')}{x'},$$

where

$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$.

ICOV2 = 10

Whittle-Matérn variogram

$$\gamma(\mathbf{x}) = \sigma^2 \frac{2^{1-\nu}(x')^{\nu} K_{\nu}(x')}{\Gamma(\nu)},$$

where

$K_{\nu}(\cdot)$ is the modified Bessel function of the second kind,
$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$,
$\nu = \mathrm{PARAMS}(3)$, $\nu > 0$.

ICOV2 = 11

Continuously parameterised variogram with compact support

$$\gamma(\mathbf{x}) = \begin{cases} \sigma^2 \frac{2^{1-\nu}(x')^{\nu} K_{\nu}(x')}{\Gamma(\nu)} \left(1 + 8x'' + 25(x'')^2 + 32(x'')^3\right)(1 - x'')^8, & x'' < 1, \\ 0, & x'' \geq 1, \end{cases}$$

where

$x'' = \left\| \frac{x'}{\ell_1 s_1}, \frac{y'}{\ell_2 s_2} \right\|$,
$K_{\nu}(\cdot)$ is the modified Bessel function of the second kind,
$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$,
$s_1 = \mathrm{PARAMS}(3)$, $s_1 > 0$,
$s_2 = \mathrm{PARAMS}(4)$, $s_2 > 0$,
$\nu = \mathrm{PARAMS}(5)$, $\nu > 0$.

ICOV2 = 12

Generalized hyperbolic distribution variogram

$$\gamma(\mathbf{x}) = \sigma^2 \frac{\left(\delta^2 + (x')^2\right)^{\frac{\lambda}{2}}}{\delta^{\lambda} K_{\lambda}(\kappa\delta)} K_{\lambda}\left(\kappa\left(\delta^2 + (x')^2\right)^{\frac{1}{2}}\right),$$

where

$K_{\lambda}(\cdot)$ is the modified Bessel function of the second kind,
$\ell_1 = \mathrm{PARAMS}(1)$, $\ell_1 > 0$,
$\ell_2 = \mathrm{PARAMS}(2)$, $\ell_2 > 0$,
$\lambda = \mathrm{PARAMS}(3)$, no constraint on $\lambda$,
$\delta = \mathrm{PARAMS}(4)$, $\delta > 0$,
$\kappa = \mathrm{PARAMS}(5)$, $\kappa > 0$.

9: NORM – INTEGER *Input*

*On entry*: determines which norm to use when calculating the variogram.

NORM = 1

The 1-norm is used, i.e., $\|x, y\| = |x| + |y|$.

NORM = 2

The 2-norm (Euclidean norm) is used, i.e., $\|x,y\| = \sqrt{x^2 + y^2}$.

*Suggested value*: NORM $= 2$.

*Constraint*: NORM $= 1$ or 2.

10: NP – INTEGER *Input*

*On entry*: the number of parameters to be set. Different covariance functions need a different number of parameters.

ICOV2 $= 6$
: NP must be set to 0.

ICOV2 $= 3$, 4, 5, 7 or 9
: NP must be set to 2.

ICOV2 $= 1$, 2, 8 or 10
: NP must be set to 3.

ICOV2 $= 11$ or 12
: NP must be set to 5.

11: PARAMS(NP) – REAL (KIND=nag_wp) array *Input*

*On entry*: the parameters for the variogram as detailed in the description of ICOV2.

*Constraint*: see ICOV2 for a description of the individual parameter constraints.

12: PAD – INTEGER *Input*

*On entry*: determines whether the embedding matrix is padded with zeros, or padded with values of the variogram. The choice of padding may affect how big the embedding matrix must be in order to be positive semidefinite.

PAD $= 0$
: The embedding matrix is padded with zeros.

PAD $= 1$
: The embedding matrix is padded with values of the variogram.

*Suggested value*: PAD $= 1$.

*Constraint*: PAD $= 0$ or 1.

13: ICORR – INTEGER *Input*

*On entry*: determines which approximation to implement if required, as described in Section 3.

*Suggested value*: ICORR $= 0$.

*Constraint*: ICORR $= 0$, 1 or 2.

14: LAM(MAXM(1) $\times$ MAXM(2)) – REAL (KIND=nag_wp) array *Output*

*On exit*: contains the square roots of the eigenvalues of the embedding matrix.

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

*On exit*: the gridpoints of the $x$-coordinates at which values of the random field will be output.

16: YY(NS(2)) – REAL (KIND=nag_wp) array *Output*

*On exit*: the gridpoints of the $y$-coordinates at which values of the random field will be output.

17: M(2) – INTEGER array *Output*

*On exit*: M(1) contains $M_1$, the size of the circulant blocks and M(2) contains $M_2$, the number of blocks, resulting in a final square matrix of size $M_1 \times M_2$.

18: APPROX – INTEGER *Output*

*On exit*: indicates whether approximation was used.

APPROX $= 0$
No approximation was used.

APPROX $= 1$
Approximation was used.

19: RHO – REAL (KIND=nag_wp) *Output*

*On exit*: indicates the scaling of the covariance matrix. RHO $= 1$ unless approximation was used with ICORR $= 0$ or 1.

20: ICOUNT – INTEGER *Output*

*On exit*: indicates the number of negative eigenvalues in the embedding matrix which have had to be set to zero.

21: EIG(3) – REAL (KIND=nag_wp) array *Output*

*On exit*: indicates information about the negative eigenvalues in the embedding matrix which have had to be set to zero. EIG(1) contains the smallest eigenvalue, EIG(2) contains the sum of the squares of the negative eigenvalues, and EIG(3) contains the sum of the absolute values of the negative eigenvalues.

22: 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, NS $= [\langle value\rangle, \langle value\rangle]$.
Constraint: NS(1) $\geq 1$, NS(2) $\geq 1$.

IFAIL $= 2$

On entry, XMIN $= \langle value\rangle$ and XMAX $= \langle value\rangle$.
Constraint: XMIN $<$ XMAX.

IFAIL $= 4$

On entry, YMIN $= \langle$*value*$\rangle$ and YMAX $= \langle$*value*$\rangle$.
Constraint: YMIN $<$ YMAX.

IFAIL $= 6$

On entry, MAXM $= [\langle$*value*$\rangle, \langle$*value*$\rangle]$.
Constraint: the calculated minimum value for MAXM are $[\langle$*value*$\rangle, \langle$*value*$\rangle]$.

Where the minimum calculated value of MAXM$(i)$ is given by $2^k$, where $k$ is the smallest integer satisfying $2^k \geq 2(\mathrm{NS}(i) - 1)$.

IFAIL $= 7$

On entry, VAR $= \langle$*value*$\rangle$.
Constraint: VAR $\geq 0.0$.

IFAIL $= 8$

On entry, ICOV2 $= \langle$*value*$\rangle$.
Constraint: ICOV2 $\geq 1$ and ICOV2 $\leq 12$.

IFAIL $= 9$

On entry, NORM $= \langle$*value*$\rangle$.
Constraint: NORM $= 1$ or 2.

IFAIL $= 10$

On entry, NP $= \langle$*value*$\rangle$.
Constraint: for ICOV2 $= \langle$*value*$\rangle$, NP $= \langle$*value*$\rangle$.

IFAIL $= 11$

On entry, PARAMS$(\langle$*value*$\rangle) = \langle$*value*$\rangle$.
Constraint: dependent on ICOV2, see documentation.

IFAIL $= 12$

On entry, PAD $= \langle$*value*$\rangle$.
Constraint: PAD $= 0$ or 1.

IFAIL $= 13$

On entry, ICORR $= \langle$*value*$\rangle$.
Constraint: ICORR $= 0$, 1 or 2.

## 7 Accuracy

Not applicable.

## 8 Further Comments

None.

## 9 Example

This example calls G05ZRF to calculate the eigenvalues of the embedding matrix for 25 sample points on a 5 by 5 grid of a two-dimensional random field characterized by the symmetric stable variogram (ICOV2 $= 1$).

## 9.1 Program Text

```
!   G05ZRF Example Program Text

!   Mark 24 Release. NAG Copyright 2012.

    Program g05zrfe

!     G05ZRF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g05zrf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                   :: nin = 5, nout = 6, npmax = 4
!     .. Local Scalars ..
      Real (Kind=nag_wp)                   :: rho, var, xmax, xmin, ymax, ymin
      Integer                              :: approx, icorr, icount, icov2,    &
                                              ifail, norm, np, pad
!     .. Local Arrays ..
      Real (Kind=nag_wp)                   :: eig(3), params(npmax)
      Real (Kind=nag_wp), Allocatable      :: lam(:), xx(:), yy(:)
      Integer                              :: m(2), maxm(2), ns(2)
!     .. Executable Statements ..
      Write (nout,*) 'G05ZRF Example Program Results'
      Write (nout,*)

!     Get problem specifications from data file
      Call read_input_data(icov2,np,params,norm,var,xmin,xmax,ymin,ymax,ns, &
        maxm,icorr,pad)

      Allocate (lam(maxm(1)*maxm(2)),xx(ns(1)),yy(ns(2)))

!     Get square roots of the eigenvalues of the embedding matrix
      ifail = 0
      Call g05zrf(ns,xmin,xmax,ymin,ymax,maxm,var,icov2,norm,np,params,pad, &
        icorr,lam,xx,yy,m,approx,rho,icount,eig,ifail)

!     Output results
      Call display_results(approx,m,rho,eig,icount,lam)

    Contains
      Subroutine read_input_data(icov2,np,params,norm,var,xmin,xmax,ymin,ymax, &
        ns,maxm,icorr,pad)

!       .. Implicit None Statement ..
        Implicit None
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (Out)     :: var, xmax, xmin, ymax, ymin
        Integer, Intent (Out)                :: icorr, icov2, norm, np, pad
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out)     :: params(npmax)
        Integer, Intent (Out)                :: maxm(2), ns(2)
!       .. Executable Statements ..
!       Skip heading in data file
        Read (nin,*)

!       Read in covariance function number
        Read (nin,*) icov2

!       Read in number of parameters
        Read (nin,*) np

!       Read in parameters
        If (np>0) Then
          Read (nin,*) params(1:np)
        End If

!       Read in choice of norm to use
        Read (nin,*) norm
```

```
!       Read in variance of random field
        Read (nin,*) var

!       Read in domain endpoints
        Read (nin,*) xmin, xmax
        Read (nin,*) ymin, ymax

!       Read in number of sample points
        Read (nin,*) ns(1:2)

!       Read in maximum size of embedding matrix
        Read (nin,*) maxm(1:2)

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

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

        Return

      End Subroutine read_input_data

      Subroutine display_results(approx,m,rho,eig,icount,lam)

!       .. Implicit None Statement ..
        Implicit None
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: rho
        Integer, Intent (In)                 :: approx, icount
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: eig(3)
        Integer, Intent (In)                 :: m(2)
        Real (Kind=nag_wp), Intent (In)      :: lam(m(1),m(2))
!       .. Local Scalars ..
        Integer                              :: i
!       .. Executable Statements ..
!       Display size of embedding matrix
        Write (nout,*)
        Write (nout,99999) 'Size of embedding matrix = ', m(1)*m(2)

!       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

!       Display square roots of the eigenvalues of the embedding matrix
        Write (nout,*)
        Write (nout,*) 'Square roots of eigenvalues of embedding matrix:'
        Write (nout,*)
        Do i = 1, m(1)
          Write (nout,99996) lam(i,1:m(2))
        End Do

        Return

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

```
99996   Format (1X,8F8.4)

      End Subroutine display_results

    End Program g05zrfe
```

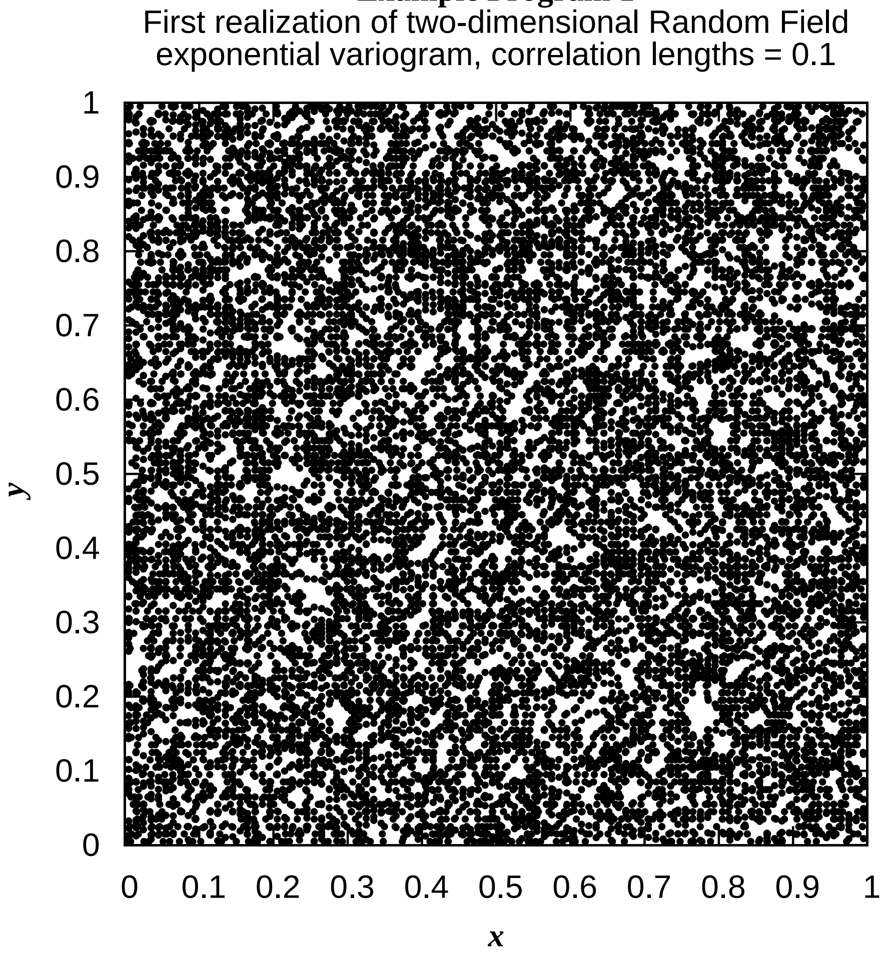
## 9.2 Program Data

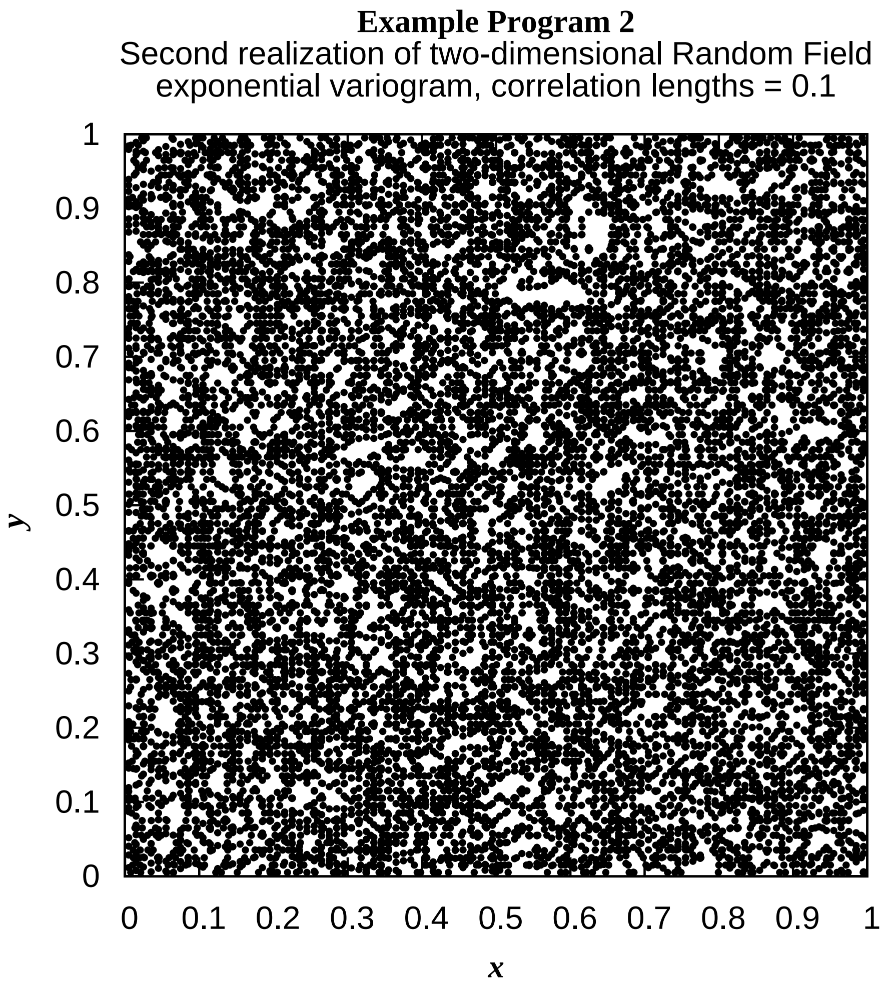
```
G05ZRF Example Program Data
  1               : icov2  (icov2=1, symmetric stable)
  3               : np     (icov2=1, 3 parameters)
  0.1   0.15  1.2 : params (icov2=1, l1, l2  and nu)
  2               : norm
  0.5             : var
 -1.0   1.0       : xmin, xmax
 -0.5   0.5       : ymin, ymax
  5     5         : ns
 64    64         : maxm
  2               : icorr
  1               : pad
```

## 9.3 Program Results

```
 G05ZRF Example Program Results


 Size of embedding matrix =      64

 Approximation not required

 Square roots of eigenvalues of embedding matrix:

   0.8966  0.8234  0.6810  0.5757  0.5391  0.5757  0.6810  0.8234
   0.8940  0.8217  0.6804  0.5756  0.5391  0.5756  0.6804  0.8217
   0.8877  0.8175  0.6792  0.5754  0.5391  0.5754  0.6792  0.8175
   0.8813  0.8133  0.6780  0.5751  0.5390  0.5751  0.6780  0.8133
   0.8787  0.8116  0.6774  0.5750  0.5390  0.5750  0.6774  0.8116
   0.8813  0.8133  0.6780  0.5751  0.5390  0.5751  0.6780  0.8133
   0.8877  0.8175  0.6792  0.5754  0.5391  0.5754  0.6792  0.8175
   0.8940  0.8217  0.6804  0.5756  0.5391  0.5756  0.6804  0.8217
```

**Example Program 1**




**Example Program 2**
Second realization of two-dimensional Random Field
exponential variogram, correlation lengths = 0.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
y
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x

---