WE-NEED

WatEr NEEDs, availability, quality and sustainability

Deliverable Number:	D1.3
Work package number:	WP1
Deliverable title	Software to generate unconditional and conditional sub-Gaussian fields
Type	Report
Dissemination Level	Public
Lead participant	Politecnico di Milano
Contributing scientists and other personnel	Martina Siena, Alberto Guadagnini, Monica Riva
Scheduled delivery date	20 October 2017
Actual / forecast delivery date	20 October 2017

Deliverable summary

As detailed in the Deliverable 1.2, a Generalized Sub-Gaussian (GSG) random field, Y(x), is defined as the product of a single- or multi-scale Gaussian random field, G(x), and a non-negative subordinator, U(x), independent of G. This model has been shown to be associated with the remarkable feature of characterizing within a unique theoretical framework the scaling properties of the sample probability density functions of increments of Y at various lags and the non-Gaussian nature of the distribution of Y. We developed a software, GSG_GEN, for the generation of multidimensional random realizations of statistically isotropic or anisotropic (i) unconditional or conditional G fields or (ii) unconditional GSG fields when U(x) has a log-normal distribution. This document provides a manual for the use of GSG_GEN. The software is implemented in FORTRAN and it takes advantage of MPICH2 (Argonne National Laboratory) for parallel computing and of MATLAB libraries (Mathworks Inc.) for a visual inspection of the results. We are currently working on the development of a graphical user interface of GSG_GEN to make the software readily available for practitioners and stakeholders.

D1.3

Software to generate unconditional and conditional sub-Gaussian fields

Contents

1. Introduction

In the following, we describe the content of the input files of the program **GSG_GEN**, together with the basic theoretical definitions of the covariance function models that can be used to generate realizations of Gaussian (G) and sub-Gaussian (Y) random fields. Section 2 illustrates the content of the two input files "**inputdata.dat**" and "**logconductivity.dat**". Sections 3 and 4 describe how the quantities contained in these input files are used by the program in order to define (a) the spatial metric (we recall that anisotropic covariance functions are handled by working in a non-Euclidean space) and (b) the covariance functions.

2. Input files

2.1 "inputdata.dat"

Line 3: mm, nn, nz (3I10)
number of elements along directions x, y and z

Line 6: dx, dy, dz (3F10.0)
element size along x, y and z

Line 9: iNMeasLocG (I10)
number of measurements of G (see Section 2.2)

Line 12: iCovarianceType (I10)
Covariance function of G

Line 15: dCovDistance0, dNugget, dMatern (3F10.0)
see Sections 3 and 4

Line 18: dIntegralScaleX, dIntegralScaleY, dIntegralScaleZ, dLambda_lower (4F10.0)
see Sections 3 and 4

Line 21: ang1, ang2, ang3 (3F10.0)
see Sections 3 and 4

Line 25: NMC, iSeed (I10, I20)
number of Monte Carlo (MC) realizations, seed value

Line 28: iOptY, dAlpha (I10, F10.0)
iOptY>0 option for generating sub-Gaussian fields.

iOptY = 0 option for generating Gaussian fields.

dAlpha = parameter α to generate U as $\ln N\left(0,(2-\alpha)^2\right)$.

Line 32: iOutOpt(1), iOutOpt(2), iOutOpt(3), iOutOpt(4) (4I10)

Option for printing output files containing sample mean field, sample covariance matrix, sample variance field and collection of realizations. Set iOutOpt(*i*)=1 for formatted files, iOutOpt(*i*)=2 for unformatted files and iOutOpt(*i*)=3 for Matlab (i.e., "*.mat") files. If iOutOpt(*i*)=0, then the corresponding file is not produced. The correspondence between element number and coordinates of the element centre are in the output file "Grid.out".

2.2 "logconductivity.dat"

The format of this file is the same as the one used within the GSLIB package (Deutsch and Journel, 1992).

If iNMeasLocG > 0 (i.e., conditional generation of *G*):

Line 8 + *i*, $i = 1,\ldots,\text{iNMeasLocG}$: measurement number, coordinates of measurement point along *x*, *y* and *z* directions, measurement value, variance of measurement error.

The sample average of the measurement values is equal to the ensemble mean of *G*.

If iNMeasLocG = 0 (i.e., unconditional generation): measurement value must be set equal to zero

3. Definition of the space metric

The space metric adopted by the program is the same as the one used within the GSLIB package (Deutsch and Journel, 1992). The covariance between two values of *G* at points **x** and **y**, $C(\mathbf{x},\mathbf{y})$, is defined as a function of a scalar lag, *h* (i.e., $C(\mathbf{x},\mathbf{y}) = C(h)$). The latter is defined as

$$h = \left((\mathbf{r}')^T \mathbf{r}'\right)^{0.5} \tag{3.1}$$

where

$$\mathbf{r}' = \mathbf{RAr} \tag{3.2}$$

In (3.2), $\mathbf{r}$ is the separation vector between points $\mathbf{x}$ and $\mathbf{y}$ (i.e., $\mathbf{r} = \|\mathbf{x}-\mathbf{y}\|$, $\|\bullet\|$ being the Euclidean norm), $\mathbf{R}$ is a rotation matrix and $\mathbf{A}$ is a transformation matrix that allows taking into account the anisotropy ratios. $\mathbf{R}$ depends on: θ_1 (azimuth, angle between *y*-axis and vertical direction, North,- clockwise is positive); θ_2 (dip angle); θ_3 (angle in the third degree of freedom) and is defined as

$$\mathbf{R} = \mathbf{R}_3\mathbf{R}_2\mathbf{R}_1 \tag{3.3}$$

where $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ are given by

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & \sin\theta_3 & \cos\theta_3 \end{bmatrix} \tag{3.4}$$

$$\mathbf{R}_2 = \begin{bmatrix} \cos(-\theta_2) & 0 & \sin(-\theta_2) \\ 0 & 1 & 0 \\ -\sin(-\theta_2) & 0 & \cos(-\theta_2) \end{bmatrix} \tag{3.5}$$

$$\mathbf{R}_3 = \begin{bmatrix} \cos(\pi/2-\theta_1) & -\sin(\pi/2-\theta_1) & 0 \\ \sin(\pi/2-\theta_1) & \cos(\pi/2-\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

$\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_1/a_2 & 0 \\ 0 & 0 & a_1/a_3 \end{bmatrix} \tag{3.7}$$

where a_1, a_2 and a_3 are the range values along directions *y*, *x*, and *z*, respectively (in the original system of coordinates, before applying the axis rotation). Quantities θ_1, θ_2, θ_3, a_1/a_2 and a_1/a_3 appearing in (3.4 – 3.7) are defined in “inputdata.dat”, being: $\theta_1,\theta_2,\theta_3$ respectively indexed by ang1, ang2, ang3; a_1/a_2 by dIntegralScaleX / dIntegralScaleY and a_1/a_3 by dIntegralScaleX / dIntegralScaleZ.

4. Definition of the covariance functions

This section lists the covariance functions implemented in the program, together with the corresponding functional forms and parameterizations. A description of the link between parameters and input variables defined in Section 2 is also provided. Whenever an input variable is not used (e.g., the input variable **dMatern** defined in Section 2 is not used when describing the exponential covariance function in Section 4.1) the value associated with this input variable is not used by the program and therefore its value does not affect the output of a simulation.

4.1 Exponential (iCovarianceType=1)

$$C(h) = \left(\sigma_Y^2 - c_0\right)\exp\left(-h/a_1\right) \qquad h > 0 \qquad (4.1)$$

$$C(h) = \sigma_Y^2 \qquad h = 0 \qquad (4.2)$$

where:

$\left(\sigma_Y^2 - c_0\right)$: dCovDistance0

σ_Y^2: dCovDistance0 + dNugget

a_1: dIntegralScaleX

4.2 Gaussian (iCovarianceType=4)

$$C(h) = \left(\sigma_Y^2 - c_0\right)\exp\left(-h^2/{a_1}^2\right) \qquad h > 0 \qquad (4.3)$$

$$C(h) = \sigma_Y^2 \qquad h = 0 \qquad (4.4)$$

where:

$\left(\sigma_Y^2 - c_0\right)$: dCovDistance0

σ_Y^2: dCovDistance0 + dNugget

a_1: dIntegralScaleX

4.3 Matèrn (iCovarianceType=5)

$$C(h) = \left(\sigma_Y^2 - c_0\right)\frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\frac{h}{a_1}\right)^{\nu} K_{\nu}\left(\frac{h}{a_1}\right) \qquad h > 0 \qquad (4.5)$$

$$C(h) = \sigma_Y^2 \qquad h = 0 \qquad (4.6)$$

where:

$\left(\sigma_Y^2 - c_0\right)$: dCovDistance0

σ_Y^2: dCovDistance0 + dNugget

a_1: dIntegralScaleX

ν: dMatern

<u>Special cases for Matèrn covariance function:</u>

$$C(h) = \left(\sigma_Y^2 - c_0\right)\exp(-h/a) \qquad h > 0\,;\ \nu = 0.5 \qquad (4.7)$$

$$C(h) = \left(\sigma_Y^2 - c_0\right)(1 + h/a)\exp(-h/a) \qquad h > 0\,;\ \nu = 1.5 \qquad (4.8)$$

$$C(h) = \left(\sigma_Y^2 - c_0\right)\left(\frac{1}{3}\left(\frac{h}{a}\right)^2 + \frac{h}{a} + 1\right)\exp(-h/a) \qquad h > 0\,;\ \nu = 2.5 \qquad (4.9)$$

$$C(h) = \left(\sigma_Y^2 - c_0\right)\exp\left(-h^2/a^2\right) \qquad h > 0\,;\ \nu \to +\infty \qquad (4.10)$$

4.4 Truncated Power Variogram (TPV; Di Federico and Neuman, 1997): iCovarianceType=6 or 7 for exponential or Gaussian modes respectively

$$C(h) = c_0 + \sigma_Y^2(\lambda_u) - \sigma_Y^2(\lambda_l) \qquad h = 0 \qquad (4.11)$$

$$C(h) = \gamma_G^2(h, \lambda_u) - \gamma_G^2(h, \lambda_l) \qquad h > 0 \qquad (4.12)$$

$$\gamma_G^2(h, \lambda_m) = \sigma_Y^2(\lambda_m)\rho(h/\lambda_m) \qquad m = l, u \qquad (4.13)$$

$$\sigma_Y^2(\lambda_m) = A\lambda_m^{\ 2H}/2H \qquad (4.14)$$

<u>Exponential modes:</u>

$$\rho(h/\lambda_m) = \exp(-h/\lambda_m) - (h/\lambda_m)^{2H}\,\Gamma(1-2H, h/\lambda_m) \qquad (4.15)$$

<u>Gaussian modes:</u>

$$\rho(h/\lambda_m) = \exp\left(-\pi h^2/4\lambda_m^{\ 2}\right) - \left(\pi h^2/4\lambda_m^{\ 2}\right)^H \Gamma\left(1-H, \pi h^2/4\lambda_m^{\ 2}\right) \qquad (4.16)$$

<u>where:</u>

A: dCovDistance0; $2H$: dMatern; λ_u: dIntegralScaleX; λ_l: dLambda_lower; c_0: dNugget.

5 Example of application

As an example, we report in Figures 5.1 and 5.2 the input files used to generate one unconditional realization of a statistically-anisotropic GSG field, *Y*. The covariance structure of the Gaussian field, *G*, is set according to a TPV model with exponential modes. The output files resulting from running GSG_GEN by setting *inputdata.dat* as in Figure 5.1 are:

- *Debug.out*, collecting information on input parameters and on the simulation outcomes;
- *Grid.out*, listing the *x*, *y* and *z*-coordinates of each node of the simulation grid;
- *G_Fields.mat*, collecting all the generated random realizations of *G*;
- *Y_Fields_alpha_1.50.mat*, collecting all the generated random realizations of *Y*;
- *Mean_G.out*, listing, for each grid node, the mean of *G* obtained over all realizations;
- *Var_G.out* listing, for each grid node, the variance of *G* obtained over all realizations.
- *Mean_Y_alpha_1.50.out*, listing, for each grid node, the mean of *Y* obtained over all realizations;
- *Var_Y_alpha_1.50.out* listing, for each grid node, the variance of *Y* obtained over all realizations.

Figure 5.3 depicts the resulting realization, which can be obtained by running the MATLAB code “Post_processing.m”. The directory “Example.zip”, provided as supplementary material, collects all programs (GSG_GEN.exe and Post_processing.m) input and output files used in this example.

```
% MESH DEFINITION (Number of elements along x, y and z directions)
%         *          *          *
        300        200          1
% CELL SIZE DEFINITION (Size of elements along x, y and z directions)
%         *          *          *
       0.01       0.01       0.01
% NUMBER OF MEASUREMENTS OF G
%         *
          0
% DEFINITION OF THE COVARIANCE MODEL
% First, the covariance type : 1: Exponential; 3: spherical; 4: Gaussian; 5: Matérn; 6: TPV (exponential modes); 7: TPV (Gaussian modes).
          6
% [Covariance at distance 0], [nugget], [Matérn parameter (if cov type = 5); 2H (if cov type = 6 or 7)]
%         *          *          *
          1       0.00      0.666
% Correlation lengths: [a1], [a2], [a3], [a1_lower (if cov type = 6 or 7)]
%         *          *          *          *
          1       0.25       1.00       0.01
% Anisotropy angles theta_1, theta_2, theta_3
%         *          *          *
       30.0        0.0        0.0
% GENERATION OPTIONS
%       NMC                  seed
%         *                     *
          1                177781
% Y generation scheme (0: no; >0: yes), Alpha
%         *          *
          1        1.5
% Ouput files (0: no file; 1: '.out', formatted; 2: '.out', unformatted; 3: '.mat' file)
%    Mean_Y     Cov_Y      Var_Y    Y_real
%         *         *          *         *
          1         0          1         3
```

Figure 5.1 Example of *inputdata.dat* file.

```
Log Conductivity
6
Measurement number
x_Y
y_Y
z_Y
Val_Y
Var_Y
     1      0.0000      0.0000      0.0000      0.0000      0.0000
```

Figure 5.2 Example of *logconductivity.dat* file.

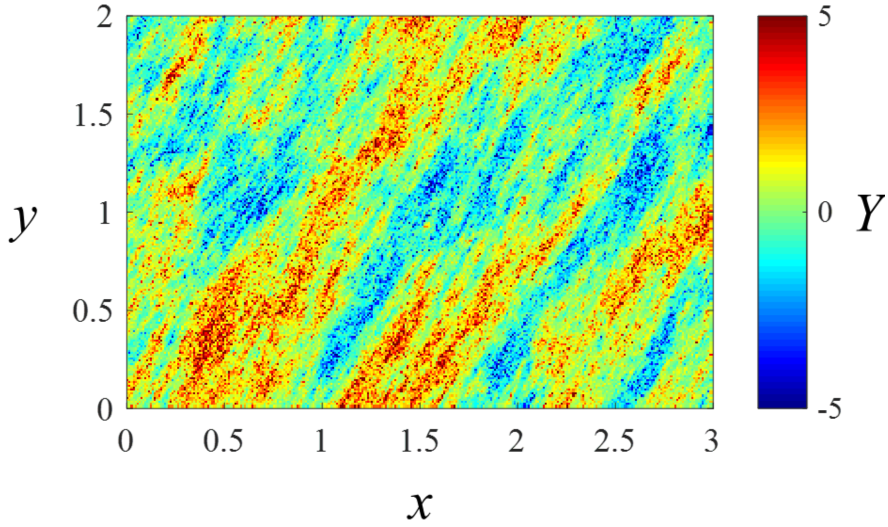

Figure 5.3 Realization of *Y* field obtained with the above input files.

References

Deutsch C.V., Journel A.G. GSLIB, 1992. Geostatistical software library and user's guide. New York, Oxford, Oxford university press. doi:10.1016/0098-3004(94)90041-8.

Di Federico V., S.P. Neuman, 1997 Scaling of random fields by means of truncated power variograms and associated spectra, *Water Resour. Res.*, 33(5), 1075-1085, 1997.

Guadagnini, A., M. Riva, M. Siena, 2017, Report on new sub-Gaussian model, Deliverable D1.2 – WE-NEED.