Multi-scale random field simulation program

1.15. 2010 (Updated 12.22.2010)

Andrew Seifried, Stanford University

Introduction

This is a supporting document for the series of Matlab scripts used to perform sequential simulation of spatially correlated random fields at multiple scales. The scripts were used to simulate samples in "Characterization of random fields and their impact on the mechanics of geosystems at multiple scales" by Chen et al (in review). This document describes the scripts and how to use them.

The goal of the program is to simulate a multi-scale two-dimensional random field from a user-specified distribution. The process is described in detail in the Chen et. al. paper, but here are a few general points:

- The field is first simulated in standard Normal space, then it is transformed to the target distribution by normal-score mapping. The target distribution should be defined at the finest scale.
- The coarse-scale field is simulated first, then some elements are refined to the fine scale. While only two scales are utilized here, the main script file is set up more generally to allow future expansion to more scales. Refinement may be controlled by several options:
    - Using an input file specifying coordinates of elements to subdivide.
    - Checking the difference between simulated values of adjacent elements, refining those with differences above a defined threshold.
    - Checking the variance among groups of neighboring simulated coarse-scale elements, refining those where the variance of the group is above a defined threshold.
- Spatial correlation is defined by one of three variogram models (spherical, Gaussian, or exponential) as indicated by a flag (see variables below).
- To speed computation, the number of previously generated elements to correlate to is limited to the closest (by normalized distance) n_max (see discussion below where the variable n_max is defined). The required matrix inversions quickly become time consuming as n_max grows.
- Also to speed computation, a surface of X and Y distance versus correlation is precalculated for all cases (intra- and inter-scale), and correlation is interpolated as needed.

Matlab files

The following files are provided as part of this software program. A brief description of the function of each file is provided here.

main_nmax_multiscale.m	Main script. This is where the input is entered/retrieved, where the remaining functions below are called, and where output files are generated.
get_indices.m	Returns the element index number, local i (row) and j (column) indices for the two-dimensional field, as well as and global x and y (spatial location) values according to figure 1.
get_rho_pre.m	Precalculates correlation versus distance (within and between scales) to save computational expense. See figure 2 for how the three cases differ in the two-scale example.
get_stats.m	Determines conditional mean and standard deviation based on previously simulated elements.
variogram.m	Determines correlation between points by calculating normalized distance and interpolating from the precalculated surfaces. variogram.m can handle individual points or arrays of coordinates.
get_var_avg.m	Determines the variance of a coarse-scale element.
transform.m	Transforms the standard normal field to the target distribution. The user defines the target distribution within this function.
subdivide.m	Reads user-created text file (subdivide.txt, see below) to flag elements to be subdivided. Also contains framework to check for large differences in simulated values of adjacent elements or large variances of groups of simulated neighboring coarse-scale elements points to flag them for subdivision.
write_out.m	Creates output text files according to the format described below.
displaypatch.m	Displays a graphical representation of the simulated sample.

Input text file

The following input text file provides additional information required by the program. It may be modified by the user as needed.

subdivide.txt — X,Y coordinates of the center of coarse-scale elements which are predetermined to be refined to the finer scale are input here by the user. The text file should have the following format:

Title	Title/description of the file
x location *y location*	Column arrays of x and y locations of the center of each element to be refined from the coarse scale to the fine scale.

Output text files

The following two text files are produced when the program is executed.

scale01_output.txt — Key values for variables are set here by the user and read by the program.

Local index # *x location* *y location* *RV realized value* *scale flag* *subdivide flag*

All listed variables in this file are column arrays where each row refers to a common element.

Local index #	Defined according to numbering scheme in figure 1
x location	x location of the center of the element in (x,y) plane
y location:	y location of the center of the element in (x,y) plane
RV realized value	simulated value from target distribution.
Scale flag	1 for coarse scale, 2 for fine scale. These should all be 1 in this file.
Subdivide flag	1 to refine to finer scale, 0 to remain untouched.

scale02_output.txt — X,Y coordinates of the center of coarse-scale elements which are predetermined to be refined to the finer scale are input here by the user.

Local index # *x loc* *y loc* *RV realized val* *scale flag* *subdiv. flag* *Associated upscaled element*

Same format as scale01_output.txt with one additional column. All listed variables in this file are column arrays, and each row refers to a common element.

Associated upscaled element — Local index number of the element in the previous scale which was refined.

Soft input variables

The following variables are specified by the user. They are specified inside the Matlab script "main_nmax_multiscale.m", and can be modified there by the user.

m	Number of elements in each column of coarse grid.
n	Number of elements in each row of coarse grid.
ds	"division size". Coarse-scale elements are subdivided into ds x ds finer elements. ds MUST be even!
d_base	Units of dist. comprising 1 side of coarsest element.
theta	Angle of principal axis (+ deg. CCW from horizontal).
a	This value dictates how quickly correlation decays with distance along the principal axis.
b	This value dictates how quickly correlation decays with distance along the axis perpendicular to the principal axis (the y axis if theta = 0).
n_max	Maximum number of previously generated elements to correlate to. Reduce this to speed up computation, but too small a number may introduce additional error by not including elements with strong correlation. A general rule of how large to make n_max is difficult to determine because a) parameters a and b can change the strength of correlation with distance and therefore the number of "significant" elements, and b) correlation is upheld somewhat indirectly to elements not included in n_max by correlating to other mutual elements.
variogramFlag	Determines which variogram model to use: variogramFlag = 1 → exponential model variogramFlag = 2 → Gaussian model variogramFlag = 3 → spherical model

cutoff	Cutoff value used as critical value in chosen subdivision method.
cutoffFlag	Determines which subdivision method to use: cutoffFlag = 0 → subdivision performed only on elements defined un file "subdivide.txt"

**"subdivide.txt." is also used to in addition to the checks performed in the following three methods **

cutoffFlag = 1 → subdivision based on the maximum difference with the value of any neighboring element.
cutoffFlag = 2 → subdivision based on the variance of neighboring elements.
cutoffFlag = 3 → subdivision based on the variance of neighboring elements. Includes more elements than 2.

Hard input variables

The following other variables are used in the Matlab program. The user should not change the values of these variables.

mu	Mean of standard normal RV at finest scale (scale 2 in two-scale example).
sigma	Standard deviation of standard normal RV at finest scale.
scale_factor	Initial scale factor is always 1 (simulation begins in the coarse scale).

Other variables

The following other variables have values that are defined within the operation of the program.

indices	Keeps track of the location of each element locally. indices(1,i) is index (local) of element I according to figure 1. The numbering scheme begins at the lower left corner of the field and increases left-to-right, then moves up to the next row above, etc.. indices(2,i) is i location (local) in field of an element. indices(3,i) is j location (local) in field of an element. indices(4,i) is x location (global) in field of an element. indices(5,i) is y location (global) in grid of an element. The indices are randomized, and the first column (i = 1) contains information for the next point to simulate. After it is

simulated, this column is deleted, and again (:,1) is the next point to simulate.

USED_MATRIX	stores information only regarding current refined state (as elements are refined, the coarse-scale simulated values are removed from this variable). USED_MATRIX is used for correlation of new elements to previously generated elements, and deleting the coarse-scale values avoids "double-counting" elements representing the same area. USED_MATRIX(:,1) is index of point (local). USED_MATRIX(:,2) is i location (local). USED_MATRIX(:,3) is j location (local). USED_MATRIX(:,4) is x location (global). USED_MATRIX(:,5) is y location (global). USED_MATRIX(:,6) is realization of RV at this point. USED_MATRIX(:,7) is scale factor. USED_MATRIX(:,8) is flag to determine whether to subdivide. USED_MATRIX(:,9) is the associated element index at previous scale.
HISTORY	Stores information of all previously simulated elements. Its format is the same as USED_MATRIX, but here the coarse-scale values are not removed as those elements are refined. This enables the user to verify marginal distributions at each scale and that the refinement process is working properly.
RHO	stores calculated values of correlation versus distance. (:,:,1) is a matrix of possible x distances. (:,:,2) is a matrix of possible y distances. (:,:,3) is a matrix of correlation between elements in scale 2 (fine). (:,:,4) is a matrix of correlation between element in scale 1 and another in scale 2. (:,:,5) is a matrix of correlation between elements in scale 1 (coarse). For a given row and column index (i,j), correlation at x distance (i,j,1) and y distance (i,j,2) is equal to (i,j,[3,4,or 5]).
Z	A holding matrix for the current random field (m x n for coarse scale, or ds x ds for fine scale) being generated.
sigma_avg	The standard deviation of a coarse-scale element.

How to use the program

The user specifies in advance both the target distribution and which coarse scale elements will be refined to the fine scale. The transformation is performed within the file transformation.m, and is initially set up by the authors to follow a lognormal distribution for the purposes of Chen et. al.. This paper should be referenced for a discussion of issues related to performing the transformation at different scales.

The file subdivide.txt should be created as described above and included in the same working directory as the other program files. If no (x,y) pairs corresponding to the center of any coarse-scale elements are listed within this file, no refinement of any coarse-scale elements will occur. If the value of d_base is set to one, the center of bottom left coarse element is (0.5,0.5). If d_base is set to 0.5, then the bottom left element is located at (0.25,0.25), etc..

The parameter d_base and all other necessary variables are defined at the top of the main Matlab file main_nmax_multiscale.m under the heading “Soft input variables.” “Soft” implies that the user may change the values to suit their puposes, while the “hard” variables below should not be altered.

The program is run from the main file, main_nmax_multiscale.m, after all variables have been defined and subdivide.txt has been created. The variables are initially set to create a simulated sample similar to figure 1. The user may play around with theta, a, and b to see how the correlation structure can change to produce more or less banded samples, or they may change subdivide.txt to change which coarse-scale elements are refined, or they may comment out the transformation command to see the effect of normal-score mapping between the standard Normal and target distributions.

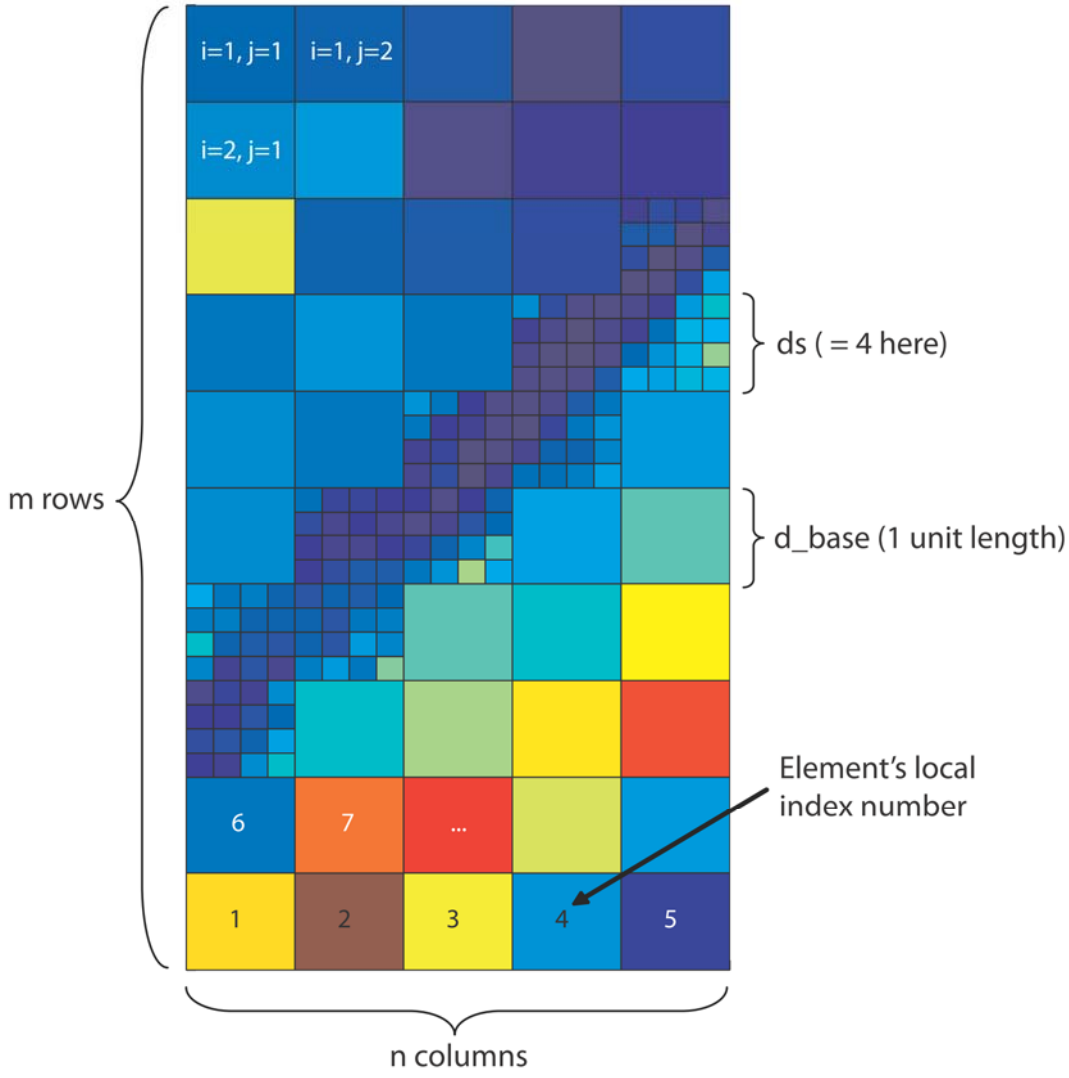

Figure 1: visual representation of variables and indices.

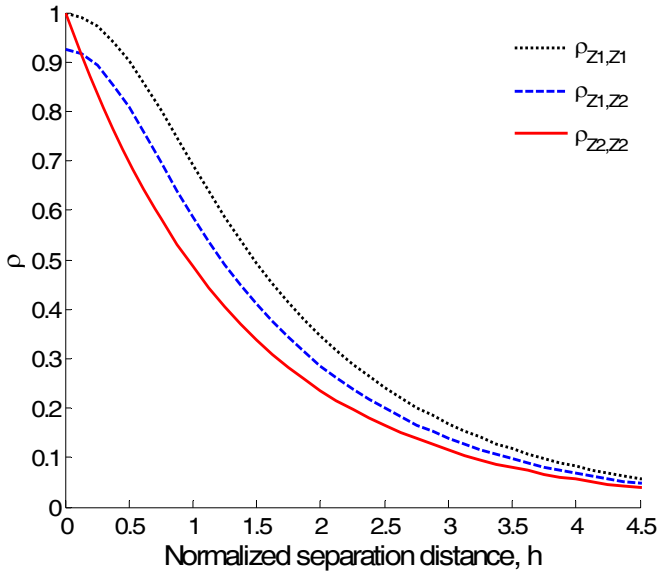

Figure 2: correlation structure (using the exponential variogram)
Z1 denotes coarse scale, Z2 denotes fine scale