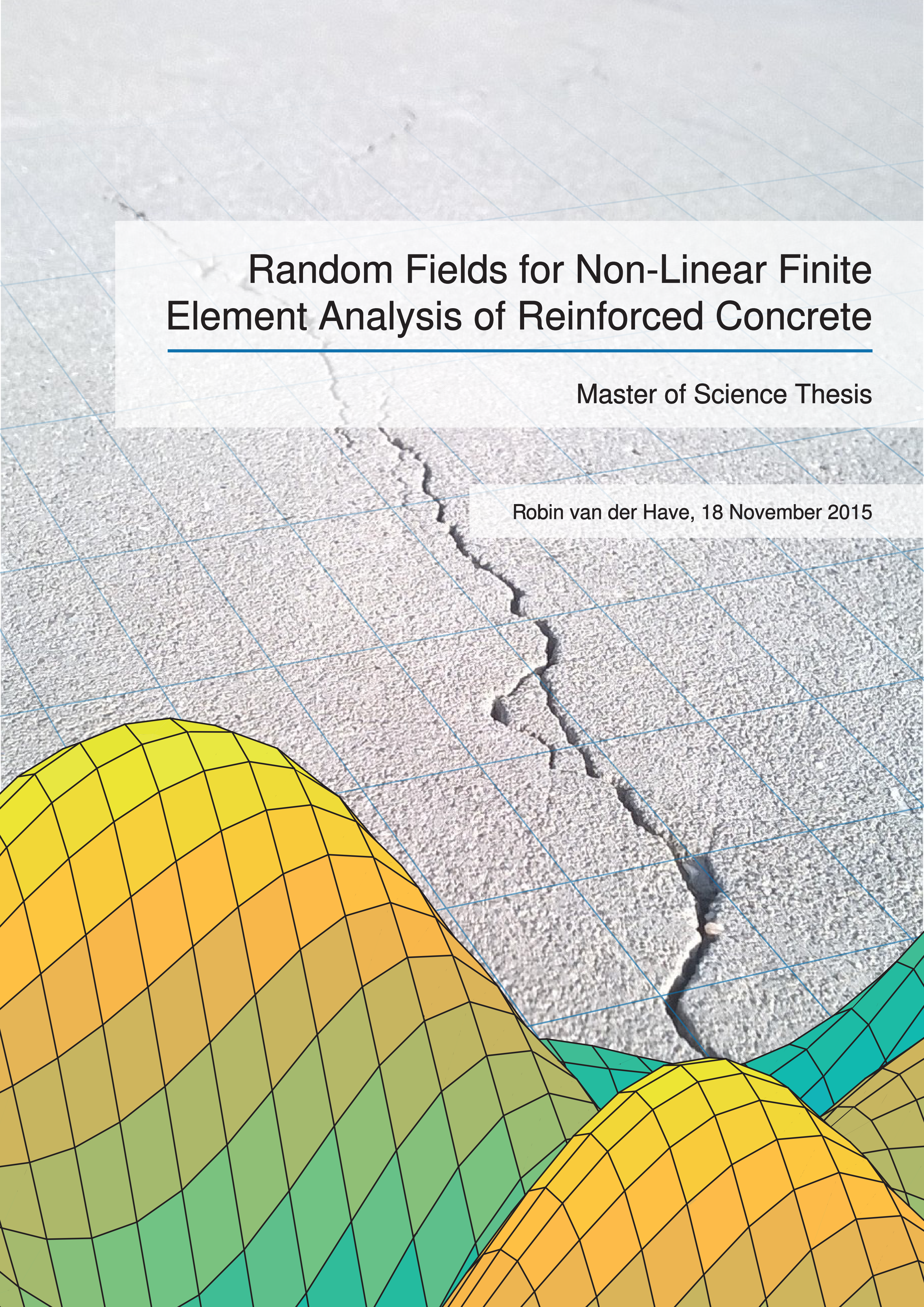
Random Fields for Non-Linear Finite Element Analysis of Reinforced Concrete

Master of Science Thesis

Robin van der Have, 18 November 2015



Random Fields for Non-Linear Finite Element Analysis of Reinforced Concrete

Master of Science Thesis

Robin van der Have
18 November 2015

Delft University of Technology

TNO DIANA BV

Thesis for the degree of Master of Science in Civil Engineering

Faculty of Civil Engineering and Geosciences
Structural Engineering, Structural Mechanics

Delft University of Technology, the Netherlands

Author: Robin Christiaan van der Have
Student number: 1514598
E-mail: r.c.vanderhave@student.tudelft.nl

Thesis committee:
Dr. Ir. M.A.N. Hendriks (TU Delft, Structural Mechanics) - Chairman
Prof. Dr. M.A. Hicks (TU Delft, Soil Mechanics)
Ir. A. Roy (TU Delft, Structural Mechanics)
Dr. Ir. G.M.A. Schreppers (TNO Diana BV)

Abstract

With more advanced methods and increasing computational power, the simulation of reinforced concrete in a Finite Element Analysis (FEA) has become more and more realistic. In a non-linear analysis of reinforced concrete, cracking behaviour and the maximum load-capacity can be determined. Such analyses sometimes suffer from unstable behaviour, especially when large parts of the structure crack at the same time. It is expected that spatially varying concrete material properties will affect crack initialization, crack patterns and the stability of the analysis. In this report, the use of spatial variability in the material properties of concrete in a Finite Element Analysis (FEA) was investigated.

To incorporate spatial variation in the Finite Element method (FEM), discretized random fields are used which are assigned to elements or integration points in the Finite Element (FE) model. In this research the following methods are implemented and have been assessed on their performance:

- Covariance Matrix Decomposition method (CMD)
- Fast Fourier Transform method (FFT)
- Local Avarage Subdivision method (LAS)
- Expansion Optimal Linear Estimation method (EOLE)

To be appropriate for the implementation in a general purpose FEM program the method has to be *efficient* with respect to computation time, *accurate* in representing the statistical characteristics of concrete and *easy to implement* in the program.

In a literature review, a large variation was found in the used values for the statistical characteristics which are involved in the modelling of the spatial variation of concrete properties. In the assessment of the random field generators this range of values was used as input. From literature and the assessment it was found that the CMD method is easy to implement and is the most accurate in representing the statistical characteristics of concrete. With respect to efficiency, the method performs poorly when the number of nodes increases. This is the case for random fields in multiple dimensions and/or for random fields with a small correlation length. The FFT method is slightly less accurate but performs very well with respect to efficiency when the number of nodes increase. The derivation of the one sided Spectral Density Function (SDF), which is needed in the FFT method, is however quite difficult. The threshold value in the correlation function and the distribution type have the largest influence on the accuracy of the random field. If the threshold value increases, and a log-normal distribution type with a high coefficient of variation (COV) is selected, the accuracy of the different methods decreases. The FFT method is slightly more accurate in representing the statistical characteristics in that case. With a large correlation length and a threshold value, the values in the random field are strongly correlated. It was found that in such a case the assumption of ergodicity does not hold any more.

In the general purpose FEM program DIANA, a random field application has been developed. The guidelines in the JCSS Probabilistic Model Code are followed and implemented as a material model in the program. This material model is used in an example to assess the model code and the influence of spatially varying material properties on a non-linear FEA. A concrete floor submitted to a shrinkage load was analysed using the JCSS material model and some variations on this material model. Unfortunately, none of the analyses reached the convergence norm in all the load steps where cracking occurs. The results are therefore not reliable since the true equilibrium path may not be followed. However, they do give insight in the influences of spatially varying material properties on a non-linear FEA. The analysis resulted in non-symmetric cracking patterns, more gradual grow in the total number of cracks and crack initialization on the weakest point in the structure. In future studies the observations from this research can be used in a probabilistic analysis where the uncertainty in the material properties, which vary in space, can be taken into account which will yield a more accurate estimate of the reliability of a structure.

Preface

In this document the results of the master's project, carried out for the company TNO DIANA BV, are reported. This master's thesis is part of the master's program Structural Mechanics at Delft University of Technology. Structural Mechanics is one of the specialization of the master track Structural Engineering.

In February 2015 the question was raised by TNO DIANA BV how spatial distributed properties can be defined in the DIANA finite element program. The theory, comparative studies between different methods and an example with a random field in a non-linear finite element analysis, carried out with the application which was developed during this research, can be found in this report. All together it gives a thoroughly response to the raised question. The project outline, provided by TNO DIANA BV, can be found in appendix A.

I would like to thank my supervisors for their support during my research. I am very pleased with the guidance Dr. Max Hendriks gave me during the project and the perspective to keep in mind the bigger picture. I want to thank Professor Michael Hicks for his help with in-depth questions about random field generation and his valuable suggestions. I want to thank Ir. Anindya Roy for his support with the mathematical problems and for the sharing of his experience he had during his master's project. I want to thank Dr. Gerd-Jan Schreppers for his help with the development of the random field application in DIANA. I am also very grateful for the opportunity he gave me to work at TNO DIANA BV and cooperate with very talented people. During my stay I have gained valuable experience and a good understanding of computer programming and Finite Element Analyses.

Finally I want to thank family and friends for their mental support during the project. Especially, I want to express my sincere thanks to my wife for her gentle touch during the frustrating moments and her encouragement to shine.

Robin van der Have
Delft, November 18, 2015

Contents

Nomenclature

Notations

$\bar{\cdot}$	Mean / expected value
$\hat{\cdot}$	Approximated value
$.^T$	Transpose of matrix or vector
$\lvert\cdot\rvert$	Absolute value / size
$\Delta\cdot$	Difference/ (lag) distance
$\sim$	denotes "has the probability distribution of"

Greek symbols

$\alpha(t,\tau)$	Time dependent factor used in concrete analysis
β_d	Ratio between the permanent and total load
$\gamma(\cdot)$	Variance function
$\delta(\cdot)$	Dirac delta function
δ_{ij}	Kronecker delta
ε	Strain
ε_u	Ultimate strain
θ	Single outcome in sample space
Θ	Sample space
λ	Eigenvalue
$\mathbf{\Lambda}$	Matrix containing the eigenvalues on the diagonal
$\tilde{\mathbf{\Lambda}}$	Matrix containing the square root of the eigenvalues on the diagonal
μ	Mean
μ_H	Mean value of random field
μ_m	Mean value of all the mean values of the generated random fields
μ_s	Standard deviation in all the mean values of the generated random fields
ν	Poisson's ratio
$\xi,\ \eta$	Spatial variable / natural coordinates
ρ	Correlation coefficient
$\rho(\cdot)$	Correlation function
$\rho_T(\cdot)$	Transformed correlation function, according to Nataf transformation
σ	Standard deviation / stress
σ_m	Mean value of all the standard deviations of the generated random fields
σ_s	Deviation in all the standard deviations of of the generated random fields
σ_T	Transformed standard deviation, according to Nataf transformation
σ_X	Standard deviation of random variable X
Σ	Transformed standard deviation of the basic compressive strength

$\phi(\cdot)$	Eigenfunction / creep coefficient
Φ_k	Random phase angles
χ	Independent zero mean, unit variance, normally distributed random variable
χ_c	Correlated standard normal random variable
ω	Angular frequency / spectral coordinate
Ω	Domain of the random field
Ω_e	Domain of an element
Ω_i	Domain associated to an integration point

Latin symbols

a	Coefficient in the the local average subdivision algorithm to determine the best linear estimate for the mean
A_k, B_k, C_k	Random amplitudes in Fourier transform methods
$B(\cdot)$	Covariance function
$B_D(\cdot)$	Covariance function of two local averages
c_1	Threshold value in correlation function
$C_{err,m}$	Mean value of the absolute error in the correlation structure
$C_{err,s}$	Deviation in the absolute error of the correlation structure
(c^i)	Standard deviation of the white noise process in i$^{\text{th}}$ stage of local average subdivision algorithm
$\mathrm{Cov}[\,\cdot\,,\,\cdot\,]$	Covariance operator
D	Domain over which a random field is averaged
D^i	Cell size in the i$^{\text{th}}$ stage of local average subdivision algorithm
$\mathrm{E}[\,\cdot\,]$	Expectation operator
E_c	Young's modulus concrete
$\mathrm{erf}(\cdot)$	Error function
$\mathcal{F}$	Sample space with a well-defined probability
$\mathbf{f}$	right hand side vector / force vector
$\mathbf{f}_e$	Element right hand side vector / element force vector
$f(\cdot)$	Weighting function in the moving average method/ basis function in series expansion methods
f_c	Compressive strength concrete
f_{ct}	Tensile strength concrete
f_{c0}	Basic compressive strength concrete
f_y	Yield strength reinforcement steel
$f_X(\cdot)$	Probability density function of random variable X
$F_X(\cdot)$	Cumulative distribution function of random variable X
$G(\omega)$	One sided spectral density function
$H(x,\theta)$	Random Field
$H_D(\cdot)$	Moving local average of the random field
$i, j, k, ..$	Primary indexes
K	Total number of frequency points in spectral domain
$\mathbf{K}$	Stiffness matrix
$\mathbf{K}_e$	Element stiffness matrix
k_1, k_2	The number of cells in stage 0 of te local average subdivision algorithm for both directions
$\mathbf{L}$	Decomposed covariance matrix
L_c	Correlation length / scale of fluctuation

L_{RF}	Element size of random field mesh
M	Transformed mean of the basic compressive strength
N	Total number of nodes in the random field mesh
N_x, N_y, N_z	Number of nodes in the x, y and z direction of the random field mesh
$N(\mu, \sigma^2)$	To indicate that a variable is normal distributed
$P[\,\cdot\,]$	Likelihood of the occurrence of an event / Probability measure
$\mathbf{Q}$	Matrix containing the eigenvectors
$\mathbf{R}$	Correlation matrix
$S(\omega)$	Two sided spectral density function of random field
$\mathbf{u}$	Displacement vector
$\mathrm{Var}[\,\cdot\,]$	Variance operator
W	Discrete zero mean and unit variance white noise process
x, y, z	Cartesian coordinates
x_c	Coordinate of the midpoint of an element
X	Random variable
$z_c(x_i)$	Correlated random variable associated with the i^{th} node in the random field mesh
$Z_c^i(x_{2j})$	Correlated random variable in i^{th} stage of local average subdivision algorithm

Abbreviations

BFGS	Broyden–Fletcher–Goldfarb–Shanno (algorithm)
CEB	Comité Européen du Béton / European Committee for Concrete
CMD	Covariance Matrix Decomposition
CDF	Cumulative Distribution Function
COV	Coefficient of variation
CPU	Central Processing Unit
DFT	Discrete Fourier Transform
DIANA	DIsplacement ANAlysis, product name of general purpose finite element program
DOF	Degree of Freedom
Exp	Exponential (correlation function)
FCM	Finite Cell Method
FE	Finite Element
FEA	FInite Element Analysis
FEM	Finite Element Method
FERM	Finite Element Reliability Method
FIB	Fédération Internationale du Béton / International Federation for Structural Concrete
FFT	Fast Fourier Transform
FORM	First Order Reliability Method
EOLE	Expansion Optimal Linear Estimation
IP	Integration point
JCSS	Joint Commission on Structural Safety
KL	Karhunen-Loève
LAS	Local Average Subdivision
LHS	Left-hand Side
MA	Moving Average
MP	Midpoint (method)
MSC	Monte Carlo Simulation
OLE	Optimal Linear Estimation

OSE	Orthogonal Series Expansion
PC	Polynomial Chaos
PDF	Probability Density Function
RF	Random Field
RHS	Right-Hand Side
RV	Random Variable
SDF	Spectral Density Function
SExp	Squared Exponential (correlation function)
SF	Shape Function (method)
SFEM	Stochastic Finite Element Method
SLS	Serviceability Limit State
SORM	Second Order Reliability Method
SSFEM	Spectral Stochastic Finite Element Method
TBM	Turning Bands Method
ULS	Ultimate Limit State

Introduction

1

1.1 Numerical analysis and structural reliability

Engineering firms often make use of computer programs to design new structures and to reassess existing structures. In these programs an idealized reality is modelled to determine the behaviour of structures. Such a model requires input variables (which describe the loading, geometry and material properties), response variables (displacements, strains and stresses) and the relationships between these variables. Since the 19th century a lot of research is done to describe these relationships. For example, structural models (bars, beams, plates, shells, ...) and constitutive laws (elasticity, plasticity, damage models, ...) were developed and improved [1].

With the development of computer science, more and more numerical methods have been developed to solve the boundary value problems of mechanical systems. The last decades a spectacular growth of the computational power of computers led to a great development of these methods. The Finite Element Method (FEM) is currently the most used method. The behaviour of structures can be described very well with this method.

In FEM the input variables are often described by deterministic values. However, in reality these variables have a random nature. For example, the execution quality of concrete is not known beforehand. Another example is the uncertainty of the wind loading on a structure during its life time. To deal with this variability, partial safety factors are applied on the loading parameters and material properties. These partial safety factors are based on the statistical properties of the input variables. Although these factors are based on a probabilistic framework, they give no insight in the reliability of the structure. Beside that, this approach may lead to conservative results. In case the engineer wants to optimize a structure, insight in the structural reliability is desirable. Especially in the case of a reassessment, it could be very valuable if it turns out that a structure meets its requirements and does not have to be replaced. To gain more insight into the reliability of a structure, probabilistic design methods are combined with FEM. This led to a collection of methods which are denoted by the collective term, the Stochastic Finite Element Method (SFEM) [2].

1.2 Spatial variabilty and random fields

To capture the spatial variability of the input variables, random fields are used. A realization of a random field can be generated with different methods and results in a spatial function. The statistical characteristics of the input variable, such as the mean, variance and covariance structure, can be obtained via experiments. An appropriate method to generate a random field represents these statistical characteristics accurately and is efficient in terms of computation time. With the use of random fields the uncertainty of properties varying in space can be taken into account in a FEM program, yielding a more precise estimate of the reliability. Random field can be beneficial for a non-linear Finite Element Analysis (FEA) with respect to numerical stability. In a non-linear analysis, phenomena as plasticity and cracking are taken into account. This leads to a change in the response of the model during loading, i.e. the stiffness of the model changes in every load step. The true equilibrium path is sought for by applying an iteration scheme to find static equilibrium in every load step. If multiple failure mechanisms can occur, for example in case of symmetry or a zone with constant strain in the model, it can be very hard to find a static equilibrium since the model fails in multiple failure mechanisms at once. In reality however, failing starts with a single mechanism in such a case due to inhomogeneous material properties. When this variation in space is taken into account with a random field, the model may show more realistic material behaviour and failure mechanisms. With the use of a random field, crack initialization will occur more locally (weakest point) which may lead to more numerical stable non-linear analyses.

1.3 Aim and scope of the study

In literature many examples, especially for soil structures, can be found where random fields are applied in the analysis of a structure. It is very convenient to do so for soil, since the variability in space for the material properties is large and has a significant influence on the failure behaviour. Another material for which it may be interesting to apply a random field is reinforced concrete. Concrete is composite material which is build up out of aggregates which are 'glued' to each other with hardened cement. Since the aggregates differ in shape and size, and are not distributed equally during the production of concrete, the material properties vary in space. The concrete is reinforced with steel to improve the bearing capacity. The (spatial) variation in the steel properties is however much lower. It is therefore questionable if it is relevant to take the spatial variability of the concrete material properties into account in a analysis of reinforced concrete. In literature, a limited number of examples can be found where a random field is applied in the analysis of a reinforced concrete structure. In this study the focus is therefore on random field generation for reinforced concrete models. It is investigated which method to generate a random field is most efficient and accurate in representing the statistical characteristics of concrete. Next, the influences of spatially varying material properties on a non-linear concrete cracking analysis is explored. Both subjects are investigated in order to answer the following research question:

"*To what extent are the available random field generators suitable to represent the statistical characteristics of concrete in a general purpose program, and what is the influence of spatially varying concrete material properies on a non-linear FEA?*"

Different methods to generate a random field are described and applied in literature. Which method is the most accurate and efficient may differ for every concrete property since it depends on the following input variables:

- The type of probabilistic density function
- Coefficient of variation
- Type of correlation function
- Size of the model/random field

Even though comparisons of different methods are available in literature, one cannot apply them for concrete, as the statistical characteristics of concrete are different from the used characteristics in these comparisons.

The most promising methods are implemented in the general FEM program DIANA for a non-linear FEA of a reinforced concrete model. The random field application is integrated in the work flow of the program. It may function as the input for a reliability analysis. Besides that, it may also be of use with respect to crack initialization and numerically stability of a non-linear FEA.

In this research the focus will be on finding a random field generator which is efficient and describes the statistical characteristics of concrete the most accurate. This random field generator will be used in the non-linear analyses where spatially varying material properties are included in the analysis. Only a couple of analyses with spatially varying material properties are carried out to investigate the differences in the behaviour of the model, compared to the case where no variation is applied in the properties of the model. This will give useful insights in what practical issues can be encountered in the work flow and the results of such an analysis. These have to be explored first, before carrying out a full probabilistic analysis where the total number of non-linear FEA is very large. Selecting the appropriate settings for every single non-linear analysis in such a case is very important with respect to efficiency and numerical stability. For example the selected element size or the iteration scheme are factors which have a large influence on those aspects. The findings of this research will be a good starting point for studies which aim to do a full probabilistic analysis with spatially varying concrete material properties.

1.4 Thesis Structure

In this report the raised research question will be answered step by step. First, the theoretical background, required to understand the different concepts and methods within this field of research, is described in chapter 2. Different methods to generate a random field will be explained. Next to the theoretical background, a review of literature on random fields generation for reinforced concrete structures is presented in this chapter. For some of the statistical characteristics, that describe the spatial variation of concrete parameters, a large variation was found in the used values. In chapter 3 an assessment of the different methods to generate a random field is carried out. Implementation issues of the selected methods for this assessment are considered. The different parameters involved in this comparison are varied, based on the range of values found during the literature review. This will give a basis to select the most appropriate method for the generation of a random field for the different concrete properties. In chapter 4 one of the probabilistic model codes is chosen for the implementation in DIANA. This model code gives guidelines for the different statistical characteristics involved in the generation of a random field for reinforced concrete. An extra comparison is carried out to check if the findings hold in general. Examples of random fields for the different material parameters, which described in the selected model code, will be given at the end of the chapter. In chapter 5 an example of the implemented functionality in DIANA will be presented to investigate the influence of a random field on a non-linear FEA. Finally the work in this report will be discussed and an answer will be formulated in the last chapter of this report.

Theory 2

In this first chapter a basis is formed for this study. All the theory which is needed tot understand the different concepts and methods is explained. Also a notation is introduced which is used throughout the remainder of this report. For a more complete overview reference is made to [1, 3–6]. First, basic statistics and random variables will be explained. Second, the concept of random fields and the available random field generators will be explained. Thereafter, different approaches are considered to take into account spatial variation in a reliability analysis of reinforced concrete structures. In the last section the structure of a general purpose FEM program a will be given.

2.1 Random variables and basic statistics

To be able to define random variables and random fields, first, the mathematical construct to model a real world process should be defined as shown hereafter. The possible outcomes of an experiment are called *events*, and the collection of all possible outcomes of an experiment is called the *sample space* $[\Theta]$. The collection of outcomes in this sample space which have a well-defined probability is called the $\sigma-$algebra, denoted by $\mathcal{F}$. The *probability measure* of an event is denoted by P. The mathematical construct for modelling an experiment is called the *probability space* $(\Theta, \mathcal{F}, P)$. The likelihood of the occurrence of event A is denoted by $P[A]$ and satisfies the following three axioms:

1. For any event A,

$$P[A] \geq 0 \tag{2.1}$$

2. The probability associated with the entire sample space (or a certain event) is:

$$P[\Theta] = 1 \tag{2.2}$$

3. For any sequence of disjoint events, A_1, A_2, ...,A_m:

$$P[A_1 \cup A_2 \cup ... \cup A_m] = P[A_1] + P[A_2] + ... + P[A_m] \tag{2.3}$$

From these three fundamental assumptions, all the other rules for probability calculations are defined.

2.1.1 Random variables

A *random variable*, denoted by X, is a means to map from a *sample space* Θ to real numbers. To underline the random nature of X, the dependency on the outcomes may be added in some cases as in $X(\theta)$. An example of a random variable is whether a stranger is left- or right-handed. This could be written as, $X(left) = 0$ and $X(right) = 1$. Than $X > 0$ means the person is right-handed.

Or it could be written as $x = X(Right)$, where the lower case, x, refers to the known outcome of the experiment. One can say that x is a realisation of the random variable X.

A distinguish can be made between *continuous* and *discrete* random variables. In this study only continuous random variables are of interest. An example of a continuous random variable is the tensile strength of a steel bar. As for many engineering applications, the sample space is already expressed in real numbers. In that case direct mapping $(X = f_y(N/mm^2))$ can be applied. It is a continuous random variable because the variable could attain any value between 0 and ∞.

2.1.2 Probabilistic distributions

The *cumulative distribution function* (CDF) of X is defined by:

$$F_X(x) = P[X \leq x] = \int_{-\infty}^{x} f_X(y)dy \tag{2.4}$$

and has the following properties:

- $0 \leq F_X(x) \leq 1$
- $\lim_{x\to\infty} F_X(x) = 1$
- $\lim_{x\to-\infty} F_X(x) = 0$
- $F_X(x)$ is non-decreasing
- $F_X(x)$ is right continuous

Since $P[X = x]$ is infinitesimally small for a continuous random variable, one can not speak of a probability. Therefore, it is defined as a density, which has to be multiplied by a length to obtain a probability. The *probability density function* (PDF) is defined by:

$$f_X(x) = \frac{dF_X(x)}{dx} \tag{2.5a}$$

$$f_X(x)dx = P[x < X \leq x + dx] \tag{2.5b}$$

and has the following properties:

- $f_X(x) \geq 0$
- $\int_{-\infty}^{\infty} f_X(x)dx = 1$

2.1.3 Random vectors

In many problems more than one random variable is considered. For example the mean temperature in Rotterdam and in Amsterdam at a certain day. The observation is than expressed as a random vector as:

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{2.6}$$

where, according to the example, X_1 is the temperature in Rotterdam and X_2 the temperature in Amsterdam. A random vector can have any length. Also different capital letters can be used to denote different random variables.

2.1.4 Characteristics of random variables

A random variable is fully defined by its PDF. In this section, useful characteristics of a random variable are derived from the PDF.

Mean The first moment of the PDF is a measure for the *mean* of a random variable. It gives information about where the PDF is centred.

$$\mu = \frac{\int_{-\infty}^{\infty} x f_X(x) dx}{\int_{-\infty}^{\infty} f_X(x) dx} = \int_{-\infty}^{\infty} x f_X(x) dx \tag{2.7}$$

Variance The second central moment of the PDF is called the *variance* of the random variable. The variance gives information about degree of scatter.

$$\sigma^2 = \text{Var}[X] = \int_{-\infty}^{\infty} (x-\mu)^2 f_X(x) dx \tag{2.8}$$

The positive square root of the variance is called the *standard deviation* and is denoted by σ. It has the same units as X and μ and is another measure for the degree of scatter.

$$\sigma = \sqrt{\text{Var}[X]} \tag{2.9}$$

The dimensionless *coefficient of variation*, COV, is defined as $\frac{\sigma}{\mu}$. For the evaluation of the spread of a random variable around its mean, this normalised value gives us more insight.

Covariance The linear dependence between two random variables can be determined with the *covariance*, and is defined by:

$$\text{Cov}[X,Y] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\mu_X)(y-\mu_Y) f_{XY}(x,y) dx\, dy \tag{2.10}$$

where f_{XY} denotes the *joint probability density function* of the random variables X and Y. The variance can actually be seen as a special case of the covariance, namely $\text{Cov}[X,X] = \text{Var}[X]$. If the variable X has no affect on the value Y, the variables are called *independent*. The covariance is equal to zero in that case. The converse is not necessarily true, i.e. if $\text{Cov}[X,Y] = 0$, X and Y are not necessarily independent. It can can only be said that they are *uncorrelated*. The normalized quantity of the covariance is called the *correlation coefficient* and is defined by:

$$\rho_{XY} = \frac{\text{Cov}[X,Y]}{\sigma_X \sigma_Y} \qquad -1 \leq \rho \leq 1 \tag{2.11}$$

Logically, this value will again be equal to zero when X and Y are uncorrelated. A value of 1 indicates that the random variables are fully correlated and -1 that they are fully negatively correlated.

Expected value The *expected value* of $g(X)$, which can be any function of X, is defined by:

$$\mathrm{E}[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx \tag{2.12}$$

where $\mathrm{E}[\cdot]$ is the expectation operator. The mean, variance and covariance can be expressed using this operator as follows:

$$\mu = \int_{-\infty}^{\infty} x f_X(x) dx = \mathrm{E}[X] \tag{2.13}$$

$$\sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2 f_X(x) dx = \mathrm{E}[(X-\mu)^2] = \mathrm{E}[X^2] - \mathrm{E}^2[X] \tag{2.14}$$

$$\begin{aligned} \mathrm{Cov}[X,Y] &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\mu_X)(y-\mu_Y) f_{XY}(x,y) dx\, dy = E[(X-\mu_X)(Y-\mu_Y)] \\ &= \mathrm{E}[XY - \mu_Y X - \mu_X Y + \mu_X\mu_Y] = \mathrm{E}[XY] - \mathrm{E}[X]\mathrm{E}[Y] \end{aligned} \tag{2.15}$$

Following the rules of integration, other calculation rules for the determination of the expectation can be derived, like:

$$\mathrm{E}[g(X) \pm h(Y)] = E[g(X)] \pm \mathrm{E}[h(Y)] \tag{2.16}$$

$$\mathrm{E}[aX \pm bY \pm c] = a\,\mathrm{E}[X] \pm b\,\mathrm{E}[Y] \pm c \tag{2.17}$$

Following the same reasoning for the variance, its calculation rules reads:

$$\mathrm{Var}[aX \pm bY \pm c] = a^2\,\mathrm{Var}[X] + 2ab\,\mathrm{Cov}[X,Y] + b^2\,\mathrm{Var}(Y) \tag{2.18}$$

And for the covariance the following calculation rules hold:

$$\mathrm{Cov}[X, y \pm Z] = \mathrm{Cov}[X,Y] \pm \mathrm{Cov}[X,Z] \tag{2.19}$$

$$\mathrm{Cov}[aX \pm b, cY \pm d] = ac\,\mathrm{Cov}[X,Y] \tag{2.20}$$

2.1.5 Common distributions

Uniform distribution The simplest distribution is the *uniform distribution* and is defined by:

$$f_X(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \le x \le b \\ 0 & \text{otherwise.} \end{cases} \tag{2.21}$$

Its characteristics are:

$$\mu = \mathrm{E}[X] = \int_a^b \frac{x}{b-a}\, dx = \frac{1}{2}(a+b) \tag{2.22a}$$

$$\sigma^2 = \mathrm{Var}[X] = \int_a^b \frac{x^2}{b-a}\, dx - E^2[X] = \frac{1}{12}(b-a)^2 \tag{2.22b}$$

Normal distribution One of the most commonly used types is the *normal distribution* or *Gauss-distribution*. It is defined by:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left(-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right) \tag{2.23a}$$

$$F_X(x) = 1/2 + 1/2 \operatorname{erf}\left(\frac{x-\mu_X}{\sigma\sqrt{2}}\right) \tag{2.23b}$$

where erf is the *error function* and is defined as:

$$\operatorname{erf}(x) = \frac{2\left(\int_0^x e^{-t^2}dt\right)}{\sqrt{\pi}} \tag{2.24}$$

The PDF is symmetric about the mean μ and its maximum is at μ. When a random variable is normal distributed, it is denoted as $X \sim N(\mu, \sigma^2)$. The *standard normal distribution* has a mean equal to zero and a standard deviation equal to one. A Gaussian random variable X with mean μ and standard deviation σ can be obtained by:

$$X = \sigma Y + \mu \tag{2.25}$$

where Y is a standard normally distributed random variable.

Log-normal distribution If a variable X has a log-normal distribution, than in the expression $Y = ln(X)$, the variable Y is normal distributed. Log-normal random variables only have outcomes greater than zero. To acquire a log-normal distributed random variable, the following transformation is carried out:

$$X = \exp(\mu_Y + \sigma_Y Y) \tag{2.26}$$

where Y is a standard normal variable. The required parameters can be expressed as:

$$\sigma_Y^2 = \ln\left(1 + \frac{\sigma_X^2}{\mu_X^2}\right), \qquad \mu_Y = \ln(\mu_X) - \frac{1}{2}\sigma_Y^2 \tag{2.27}$$

where μ_X and σ_X are respectively the mean and standard deviation of the log-normal random variable X.

2.1.6 Central limit theorem

According to the *central limit theorem*, the sum of a large number of independent random variables having arbitrary distributions result in a random variable which is (almost) normally distributed. As a consequence of the central limit theorem, the sum of two normally distributed random variables is also normally distributed. Analogously tot the central limit theorem for the sum of large number of independent random variables, the product of a large number of independent random variables is (almost) log-normally distributed. In practise the normal and log-normal distribution are often used since a lot of properties are sums or products of random variables. An example of such a property is the mean compressive strength of a number of concrete samples, which will be normally distributed. For estimating maxima and minima of a set of random variables, other so called *asymptotic extreme value distributions* are used.

2.1.7 Random number generators

In probabilistic analyses often use is made of a random number generator. In this section it is explained how Gaussian distributed random numbers can be obtained. Also the requirements for a ’good’ random number generator will be given.

To generate Gaussian distributed random numbers, use is made of uniform distributed random numbers. As for others distributions (eg. exponential, weibull and poisson), Gaussian random numbers can be obtained by a transformation of uniformly distributed random numbers. This is convenient to do because uniformly distributed random numbers can be generated very well. This arises from the fact that every number in the range has an equal likelihood to be drawn from the distribution.

For the transformation the inverse PDF is used for most of the distributions. However, for the Gaussian distribution no closed-form of the inverse PDF exists. The transformation method of *Box and Muller* [7] provides a good alternative, which is exact and efficient, and is given by:

$$x_1 = \sqrt{-2ln(u_1)}\text{cos}(2\pi u_2) \tag{2.28a}$$

$$x_2 = \sqrt{-2ln(u_1)}\text{sin}(2\pi u_2) \tag{2.28b}$$

where u_1 and u_2 are uniformly distributed random numbers between zero and one and x_1 and x_2 are two independent standard Gaussian distributed random numbers.

The most common methods to generate uniformly distributed random numbers are the so called *arithmetic generators*. These consecutive methods generate numbers based on one or multiple preceding values following a fixed algorithm. These type of generators are not truly random since the random numbers are derived from previous numbers in a deterministic way. They are therefore often called *pseudorandom number generators*. Such a method has to posses the following properties to be appropriate [5]:

1. the generated numbers should appear to be independent and uniformly distributed
2. the code should be fast and not require large amounts of storage
3. have the ability to reproduce a given stream of random numbers exactly
4. should have a very long period.

When a random number generator is used for the generation of a random field it has to satisfy these properties in order to achieve no distortion in the field.

2.2 Random fields

Many processes around us show variation in time and space. If, for example a measurement of the wind speed on a certain location is considered, the signal has a random character. On another location, close to this measurement, the wind speed will be different from, and correlated to this signal. When this variability is captured into a model we come to the concept of *random fields*. According to Vanmarcke [6], a distinction can be made between uncertainty about the properties of a *random medium*, which is independent of time, and a *space-time* process, where properties at different points in space change randomly with time. In this study only the first type of random fields will be considered. An example of such a random field can be found in figure 2.1

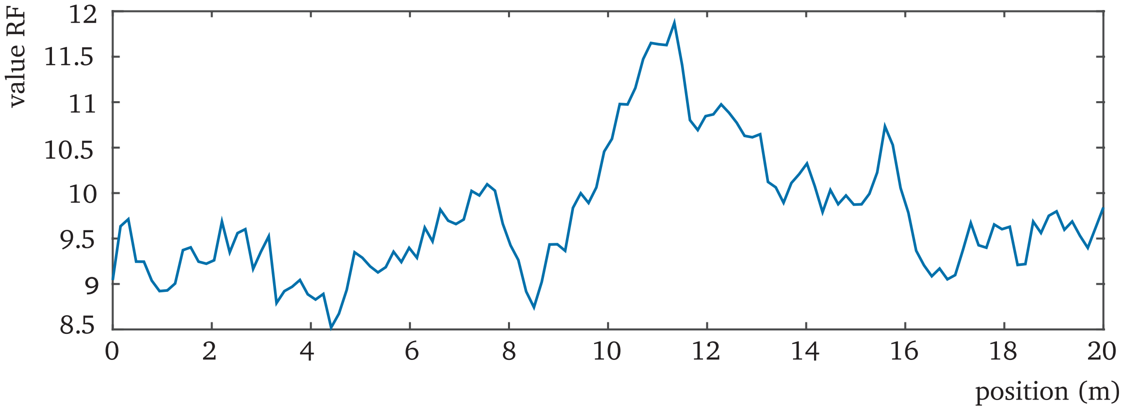


Fig. 2.1.: Example of a random field.

"A *random field* can be defined as a curve in $\mathcal{L}^2(\Theta, \mathcal{F}, P)$, that is a collection of random variables indexed by a continuous parameter $\mathbf{x} \in \Omega$, where Ω is an open set of $\mathbb{R}^d$ describing the system geometry [1]." In this definition, $\mathcal{L}^2$ denotes the inner product space of real random variables with finite second moments $(E[X^2] < \infty)$. $\mathcal{L}^2(\Theta, \mathcal{F}, P)$ belongs to the Hilbert space and is complete, which implies that the techniques of calculus can be used in this space [8].

A random field $H(\mathbf{x}, \theta)$ is thus a continuous function in space, consisting of infinite small parts which are associated with a random variable. The random field is completely defined by the set of infinite many random variables and its joint distribution function. However, this joint distribution function is very complex since there are infinite many random variables. If the random field $H(\mathbf{x}, \theta)$ is considered at a fixed location, $\mathbf{x}$, it is a random variable and is called a *sample*. For a fixed outcome, θ, of all the possible outcomes in the sample space, $H(\mathbf{x}, \theta)$ is a deterministic function of $\mathbf{x}$ and is called a realisation of the field. A random field can also be denoted with $H(\mathbf{x})$ in short.

In the above definition only one variable is considered. Such a random field is called a *univariate* random field. When more than one variable is considered, for example the tensile strength and Young's modulus of concrete, the random variable is replaced by a random vector, $\mathbf{H}(x)$. Such a field is called a *multivariate* random field. The random field can be defined in *one* dimension $H(x, \theta)$, where x is a spatial variable or in *multiple* dimensions $H(\mathbf{x}, \theta)$ where $\mathbf{x}$ is a vector, containing spatial variables.

Covariance function In reality it is likely that two points in a system which are close to each other have approximately the same value. Contrariwise, two points which are separated by a large distance are not correlated with each other. In a random field this interdependence can be described with the *Covariance function*. This function describes the covariance between two points (x_1 and x_2) having a certain distance to each other and is defined by:

$$B(x_1, x_2) = \text{Cov}[X_1, X_2] = E[X_1 X_2] - E[X_1]E[X_2] \tag{2.29}$$

where $X_1 = H(x_1)$ and $X_2 = H(x_2)$.

The *correlation function* is related to the covariance function and is defined by:

$$\rho(x_1, x_2) = \frac{B(x_1, x_2)}{\sigma(x_1)\sigma(x_2)} \tag{2.30}$$

Gaussian and non-Gaussian random fields A random field is *Gaussian* if the random variables $(H(x_1), (H(x_2), \ldots, (H(x_n)$ are all normally distributed. When the mean and covariance function of a Gaussian random field are known the random field is completely defined. Conversely, if the random variables are non-Gaussian, the random field is non-Gaussian. If a non-linear transformation is possible between the non-Gaussian random variable and the normally distributed random variable, the non-Gaussian random field can be obtained by a non-linear transformation of a Gaussian random field. Not only the random variables, but also the covariance structure of the field is transformed non-linear. For log-normally distributed random fields a transformation of the correlation function exists. The log-normal distributed random fields are very important in the modelling of engineering problems due to its non-negative domain. This transformation will be explained in more detail in section 2.3.1.

Statistical homogeneity When the joint PDF is independent of the spatial position, the random field is called *homogeneous*. This implies that the mean, variance and higher order central moments of this distribution are constant in space. The covariance function (and correlation function) only depend on the relative distance between two points in the field, i.e. $B(x_1, x_2) = B(\Delta x)$, where Δx is called the *lag distance*. When a random field is called weakly homogeneous it is only required that the mean is independent of the location in the field and the covariance function only depends on the lag distance. In this study it is assumed that all the random fields are weakly homogeneous and it is just referred to as 'homogeneous'. As a consequence of statistical homogeneity the covariance function is even, positive definite and bounded. Its maximum value is equal to the variance of the random field. The variance can be obtained at the origin of the covariance function, i.e. $B_X(0) = \sigma_X^2$.

Ergodicity A random field can be ergodic with respect to any statistical property. Ergodicity can be explained best by an example. For this example two different mean values are defined first. The mean value of a single random field can be determined with:

$$\bar{H} = \frac{1}{\Omega} \int_{\Omega} H(\mathbf{x}) dx \tag{2.31}$$

where Ω is the domain of the random field.

The sample mean for a certain position, $\mathbf{x}_1$, in the random field can be determined with:

$$\bar{X_1} = \frac{1}{N} \sum^{N} H(\mathbf{x}_1) \tag{2.32}$$

where N is the number of realisations of the random field. If Ω and N are significantly large and the mean value of a single random field is equal to the sample mean, the random field is ergodic with respect to the mean. As a consequence of ergodicity, all the information about the joint PDF of a random field can be obtained form a single realization of the random field.

2.2.1 Spectral representation of random fields

In some cases it could be interesting to express the second-order properties of a random field in the frequency domain. The second order properties can be expressed with the *Spectral Density Function* (SDF), which is comparable with the covariance function in the spatial domain. The SDF will be derived in the following paragraph for a one dimensional random field. In higher dimension the SDF can easily be found by extending the SDF for one dimension.

According to Priestley [9], a homogeneous random field, with $\rho(\Delta x)$ continuous at $\Delta x = 0$, can be expressed as a sum of sinusoids with mutually independent amplitudes and phase angles as:

$$H(x) = \mu_H + \sum_{k=-K}^{K} C_k \cos(\omega_k x + \Phi_k) \tag{2.33}$$

where μ_H is the mean value of the random field, C_k are the random amplitudes, ω_k are the frequencies which are equal to $\Delta\omega(2k-1)/2$ and Φ_k are the phase angles which are uniformly distributed over the interval $[0, 2\pi]$. Now the variance of the k^{th} component of the summation is examined by squaring this component and taking the expectation:

$$\sigma_k^2 = \mathrm{E}[(C_k \cos(\omega_k x + \Phi_k))^2] = \mathrm{E}[C_k^2]\mathrm{E}[\cos^2(\omega_k x + \Phi_k)] = \frac{1}{2}\mathrm{E}[C_k^2] \tag{2.34}$$

where $\mathrm{E}[\cos^2(\omega_k t + \Phi_k)] = \frac{1}{2}$ since Φ_k is uniformly distributed over the interval $[0, 2\pi]$. Every C_k is coupled to a ω_k. So, the variance of a the k^{th} component is a function of ω_k. To express this relation, the *spectral mass function* is introduced and is defined by:

$$S(\omega_k)\Delta\omega = \frac{1}{2}\mathrm{E}[C_k^2] \tag{2.35}$$

The variance of the random field can be expressed as the summation of all the variances of the components of the summation as:

$$\sigma_H^2 = \sum_{k=-K}^{K} \sigma_k^2 = \sum_{k=-K}^{K} \frac{1}{2}\mathrm{E}[C_k^2] \tag{2.36}$$

Combining equation 2.35 and equation 2.36 and letting K approach to infinity, the variance of the random field can be expressed as:

$$\sigma_H^2 = \lim_{K\to\infty} \sum_{k=-K}^{K} S(\omega_k)\Delta\omega = \int_{-\infty}^{\infty} S(\omega)d\omega \tag{2.37}$$

where $S(\omega)$ is called the *two sided* SDF of H(x) since it is defined for negative and positive frequencies. The summation sign can be replaced by an integral sign since $\Delta\omega \to 0$ when $k \to \infty$. In figure 2.2 the two sided spectral density function is shown. According equation 2.37, the variance of the random field is equal to area under the SDF.

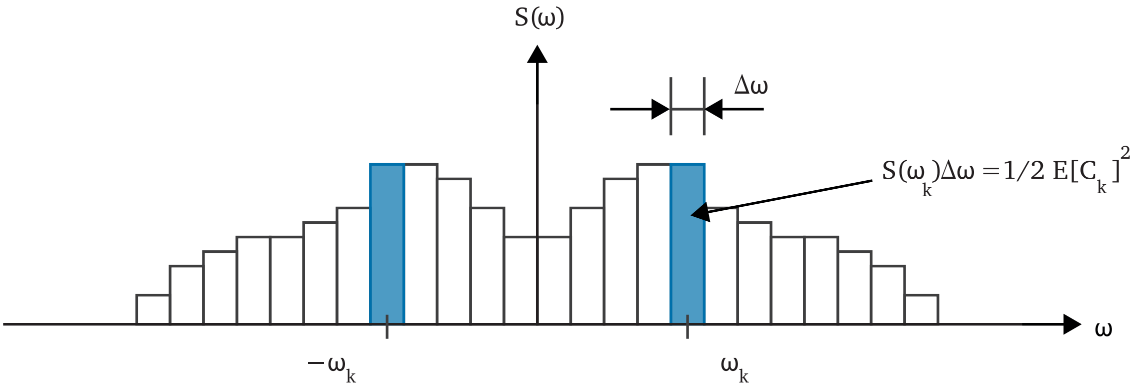


Fig. 2.2.: Two sided spectral density function [5].

Wiener-Khinchine Relations The *wiener-Khinchine relations* give the relation between the SDF and the covariance function. To come to this relation the covariance function for a homogeneous field is rewritten using equation 2.29.

$$\begin{aligned} B(\Delta x) &= \text{Cov}[H(0)H(\Delta x)] = E[H(0)H(\Delta x)] - E[H(0)]E[H(\Delta x)] \\ &= E[(\mu_H + \sum_{k=-K}^{K} C_k \cos(\Phi_k))(\mu_H + \sum_{j=-K}^{K} C_j \cos(\omega_j \Delta x + \Phi_j))] - \\ & \quad E[\mu_H + \sum_{k=-K}^{K} C_k \cos(\Phi_k)]E[\mu_H + \sum_{j=-K}^{K} C_j \cos(\omega_j \Delta x + \Phi_j)] \end{aligned} \tag{2.38}$$

Rearranging terms result in:

$$\begin{aligned} B(\Delta x) &= \sum_{k=-K}^{K} \sum_{j=-K}^{K} E[C_k C_j \cos(\Phi_k) \cos(\omega_j \Delta x + \Phi_j)] + \mu_H^2 \\ &- \mu_H^2 - \sum_{k=-K}^{K} \sum_{j=-K}^{K} E[C_k \cos(\Phi_k)]E[C_j \cos(\omega_j \Delta x + \Phi_j)] \end{aligned} \tag{2.39}$$

The expectation of cos(Φ) is equal to zero since Φ is uniformly distributed over the interval $[0, 2\pi]$. Now Using the fact that the random amplitudes C_k and phase angles Φ_k are independent result in:

$$\begin{aligned} B(\Delta x) &= \sum_{k=-K}^{K} E[C_k^2 \cos(\Phi_k) \cos(\omega_k \Delta x + \Phi_k)] \\ &= \sum_{k=-K}^{K} E[C_k^2]E[\frac{1}{2}(\cos(\omega_k \Delta x + 2\Phi_k) + \cos(\omega_k \Delta x))] \\ &= \sum_{k=-K}^{K} \frac{1}{2}E[C_k^2]\cos(\omega_k \Delta x) \\ &= \sum_{k=-K}^{K} S(\omega_k)\Delta\omega \cos(\omega_k \Delta x) \end{aligned} \tag{2.40}$$

Again the summation sign can be replaced by an integral sign letting $\Delta\omega \to 0$. This leads to the Wiener-Khinchine relations which relates $B(\Delta x)$ to $S(\omega)$ as follows:

$$B(\Delta x) = \int_{-\infty}^{\infty} S(\omega)\cos(\omega \Delta x) d\Delta x \tag{2.41}$$

$$S(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} B(\Delta x)\cos(\omega \Delta x) d\Delta x \tag{2.42}$$

From equation 2.42 and cos$(-\omega\Delta x)$ = cos$(\omega\Delta x)$ it follows that if the covariance function is symmetric the two sided SDF is also symmetric. The SDF can therefore be expressed as a *one sided* SDF as:

$$G(\omega) = 2S(\omega) \quad \text{for } \omega \geq 0 \tag{2.43}$$

The Wiener-Khinchine relations in terms of $G(\omega)$ then become:

$$B(\Delta x) = \int_0^\infty G(\omega)\text{cos}(\omega\Delta x)d\Delta x \tag{2.44}$$

$$G(\omega) = \frac{2}{\pi}\int_0^\infty B(\Delta x)\text{cos}(\omega\Delta x)d\Delta x \tag{2.45}$$

As a last remark, it can be shown that equation 2.41 and 2.42 are equivalent to the Fourier transform pair in equation 2.46 and 2.47. Expressing exp$(i\omega\Delta x)$ as cos$(\omega\Delta x) + i$ sin$(\omega\Delta x)$, the imaginary parts will cancel out for functions which are even and real. Which holds for both $B(\Delta x)$ and $S(\omega)$.

$$B(\Delta x) = \int_{-\infty}^\infty S(\omega)\text{exp}(i\omega\Delta x)d\Delta x \tag{2.46}$$

$$S(\omega) = \frac{1}{2\pi}\int_{-\infty}^\infty B(\Delta x)\text{exp}(-i\omega\Delta x)d\Delta x \tag{2.47}$$

2.2.2 Local averages of random fields

Basically all the data around us is observed with a certain resolution. Therefore all engineering properties are in some way measured as a local average. For example, in a compressive test of a concrete block, the compressive strength is determined by dividing the failure load by the cross-section of the block. In this case, failure is not a function of the strength of a single point in the block but a function of an average bond strength along the failure surface [5].

It is therefore very relevant to determine statistical properties as a spatial average over a certain domain. In this section, the formulas will be given for the general case of an n-dimensional random field. The formulas, for example for a 1D random field, can be easily found by leaving out the higher dimension terms and integrals.

For a n-dimensional random field a local average over the domain D of size $|D| = D_1 D_2 \dots D_n$ and centred at $\mathbf{x}$ is defined as:

$$H_D(\mathbf{x}) = \frac{1}{|D|}\int_{x_n - D_n/2}^{x_n + D_n/2} \cdots \int_{x_1 - D_1/2}^{x_1 + D_1/2} H(\xi_1, \dots, \xi_n)d\xi_1 \dots d\xi_n \tag{2.48}$$

where ξ is a spatial variable which is bounded by the domain over which the random field is averaged.

$H_D(\mathbf{x})$ is sometimes also referred to as the moving local average of the random field. The mean is not effected by the local averaging of the random field. This can be shown by taking the expectation of equation 2.48:

$$\text{E}[H_D(\mathbf{x})] = \text{E}\left[\frac{1}{|D|}\int_D H(\boldsymbol{\xi})d\boldsymbol{\xi}\right] = \frac{1}{|D|}\int_D \text{E}[H(\boldsymbol{\xi})]d\boldsymbol{\xi} = \text{E}[H(\mathbf{x})] \tag{2.49}$$

The variance and the covariance are however are effected by locally averaging the random field. First the covariance between two local averages is derived.

Covariance of two local averages The covariance between two local averages can be derived by considering two averaging domains of size $|D_a|$ and $|D_b|$, centred at the points $\mathbf{x}_a$ and $\mathbf{x}_b$ and taking expectations.

$$\begin{aligned} \mathrm{E}[H_{D_a} H_{D_b}] &= \mathrm{E}\left[\frac{1}{|D_a|}\int_{D_a} H(\boldsymbol{\xi})d\boldsymbol{\xi}\frac{1}{|D_b|}\int_{D_b} H(\boldsymbol{\eta})d\boldsymbol{\eta}\right] \\ &= \frac{1}{|D_a|\,|D_b|}\int_{D_a}\int_{D_b} \mathrm{E}[H(\boldsymbol{\xi})H(\boldsymbol{\eta})]d\boldsymbol{\xi}d\boldsymbol{\eta} \end{aligned} \tag{2.50}$$

According to equation 2.29, the covariance function for two local averages can be written as:

$$B_{D_a D_b}(\mathbf{x}_a - \mathbf{x}_b) = \frac{1}{|D_a|\,|D_b|}\int_{D_a}\int_{D_b} \mathrm{E}[H(\boldsymbol{\xi})H(\boldsymbol{\eta})]\,d\boldsymbol{\xi}d\boldsymbol{\eta} - \mathrm{E}^2[H(\mathbf{x})] \tag{2.51}$$

For a homogeneous zero mean random field the covariance of two local averages, having the same domain of size $|D|$ and separated by a distance equal to $k_n D_n$ in every direction, can be written as follows:

$$\begin{aligned} B_D(\mathbf{kD}) &= \frac{1}{|D|^2}\int_0^D \int_{nD}^{(n+1)D} \mathrm{E}[H(\boldsymbol{\xi})H(\boldsymbol{\eta})]\,d\boldsymbol{\xi}d\boldsymbol{\eta} \\ &= \frac{1}{|D|^2}\int_0^{D_n}\int_{k_n D_n}^{(k_n+1)D_n} \cdots \int_0^{D_1}\int_{k_1 D_1}^{(k_1+1)D_1} B(\xi_1-\eta_1,\ldots,\xi_n-\eta_n)d\xi_1 d\eta_1 \ldots d\xi_n d\eta_n \end{aligned} \tag{2.52}$$

Where k_n is a positive real number.

To simplify this equation the following change of variables is applied:

$$\xi_n = \Delta y_n + \Delta x_n \tag{2.53}$$

$$\eta_n = \Delta y_n \tag{2.54}$$

Thereafter the 2n-fold integral is reduced to a n-fold integral by integrating over Δy_n for every n. This leads to the following expression for the covariance function of two local averages having the same domain size.

$$B_D(\mathbf{kD}) = \frac{1}{|D|^2} \sum_{j_1=1}^{2} \dots \sum_{j_n=1}^{2} \int_{A_{j_n,n}} \dots \int_{A_{j_1,1}} (C_{j_1,1}) \dots (C_{j_n,n}) B(\Delta x_1, \dots, \Delta x_2) d\Delta x_1 \dots d\Delta x_n \tag{2.55a}$$

where

$$\begin{aligned} \int_{A_{1,n}} &= \int_{(k_n-1)D_n}^{k_n D_n} \\ \int_{A_{2,n}} &= \int_{k_n D_n}^{(k_n+1)D_n} \end{aligned} \tag{2.55b}$$

$$\begin{aligned} C_{1,n} &= (k_n+1)D_n - \Delta x_n \\ C_{2,n} &= \Delta x_n - (k_n-1)D_n \end{aligned} \tag{2.55c}$$

and Δx_1 till Δx_n are the lag distances in different directions which are bounded by the size of the domain. The derivation of equation 2.55 can be found in appendix C.

Variance function The variance of a local average is reduced with increasing domain size $|D|$. It can be expressed as:

$$\text{Var}[H_D(\mathbf{x})] = \gamma(|D|)\sigma^2 \tag{2.56}$$

where $\gamma(|D|)$ is by definition the *variance function* of $H_D(\mathbf{x})$, which expresses the reduction of the point variance σ^2 of the random field under local averaging. So after this operation the obtained field is more flatten out.

To find the expression for the variance function a special case of the covariance between two local averages is considered. The variance of the moving local average can be obtained at the origin of the covariance function, i.e. $\text{Var}[H_D(\mathbf{x})] = B_D(0)$. This can be achieved by letting k_n be equal to zero in every direction, which is the same as overlapping the two domains.

For a *homogeneous zero mean* random field the variance of the moving local average can be written as:

$$B_D(0) = \frac{1}{|D|^2} \sum_{j_1=1}^{2} \dots \sum_{j_n=1}^{2} \int_{A_{j_n,n}} \dots \int_{A_{j_1,1}} (C_{j_1,1}) \dots (C_{j_n,n}) B(\Delta x_1, \dots, \Delta x_2) d\Delta x_1 \dots d\Delta x_n n \tag{2.57a}$$

where

$$\begin{aligned} \int_{A_{1,n}} &= \int_{-D_n}^{0} \\ \int_{A_{2,n}} &= \int_{0}^{D_n} \end{aligned} \tag{2.57b}$$

$$\begin{aligned} C_{1,n} &= D_n - \Delta x_n \\ C_{2,n} &= \Delta x_n - D_n \end{aligned} \tag{2.57c}$$

Making use of the fact that the correlation function is an even function, i.e. $\rho(-\Delta \mathbf{x}) = \rho(\Delta \mathbf{x})$ The 2^n n-fold integrals can be taken together. According to equation 2.56, the variance function can thus be written as:

$$\gamma(D_1, \dots, D_n) = \frac{2^n}{|D|^2} \int_0^{D_n} \dots \int_0^{D_1} (D_1 - \Delta x_1) \dots (D_n - \Delta x_n) \rho(\Delta x_1, \dots, \Delta x_n) d\Delta x_1 \dots d\Delta x_n \tag{2.58}$$

Like the correlation function the variance function is also an even function and is bounded $(0 \leq \gamma(D_1, \dots, D_n) \leq 1)$. Considering the limits of the domain size D leads to the following findings [10]:

$$\begin{aligned} &\text{As } D \to 0 \\ &H_D(\mathbf{x}) \to H(\mathbf{x}) \qquad \sigma_D^2 \to \sigma^2 \qquad \gamma(D \to 0) \to 1 \end{aligned} \tag{2.59}$$

$$\begin{aligned} &As| : D \to \infty \\ &H_D(\mathbf{x}) \to \bar{H}(\mathbf{x}) \qquad \sigma_D^2 \to 0 \qquad \gamma(D \to \infty) \to 0 \end{aligned} \tag{2.60}$$

Covariance function in terms of variance function The covariance between two local averages can also be expressed in terms of the variance function. In n-dimensions the formula, derived by Vanmarcke [6], is given by:

$$B_{D_\alpha D_\beta}(\boldsymbol{\Delta x}) \frac{\sigma^2}{2^n |D_\alpha| |D_\beta|} \sum_{j_1=0}^{3} \dots \sum_{j_n=0}^{3} (-1)^{j_1} \dots (-1)^{j_n} |D_{j_1 j_2 \dots j_n}|^2 \gamma(D_{1j_1}, \dots, D_{nj_n}) \tag{2.61}$$

where $|D_{jkl}| = D_{1j} D_{2k} D_{3l}$. The distances used in this formula can be found in figure 2.3.

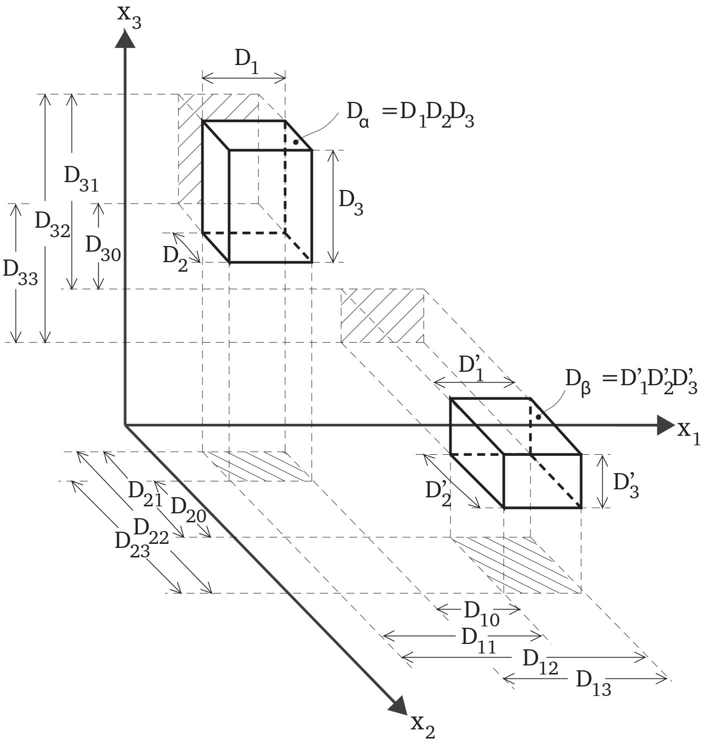


Fig. 2.3.: Distances characterizing the relative location of the volumes D_α and D_β in the three-dimensional parameter space [6].

2.2.3 Correlation length

The *correlation length* or *scale of fluctuation* is a measure for the variability in the random field. In literature it is often denoted by θ. In this report however it will be denoted by L_c since the θ symbol is already used to indicate a possible outcome of the sample space. The correlation length is defined by Vanmarcke [6] as $L_c = \lim\limits_{D\to\infty} D\gamma(D)$. Using equation 2.58 the correlation length can be written as:

$$L_c = \int_{-\infty}^{\infty} \rho(\Delta x)\, d\Delta x = 2\int_{0}^{\infty} \rho(\Delta x)\, d\Delta x \tag{2.62}$$

The correlation length is thus equal to the area under the correlation function. One necessary condition for L_c to exist , i.e. to be finite, is given by:

$$\lim_{D\to\infty} \frac{1}{D}\int_0^D \Delta x \rho(\Delta x) d\Delta x = 0 \tag{2.63}$$

Using equation 2.45, the correlation length can also be written in terms of the one sided SDF as:

$$L_c = \frac{\pi G(0)}{\sigma^2} \tag{2.64}$$

In practice it could be very hard to determine the correlation length. The scatter in the data of experiments can be to high and/or the correlation length depends mostly on multiple factors. Examples of such factors for reinforced concrete are the shape of the mould and the used aggregate sizes.

The influence of the correlation length in a random field visualized in figure 2.4. It shows two random fields for the compression strength in a concrete beam of 5 m long with a different correlation length. In the left picture a correlation length of 50 cm is used and in the right picture a correlation length of 200 cm is used.

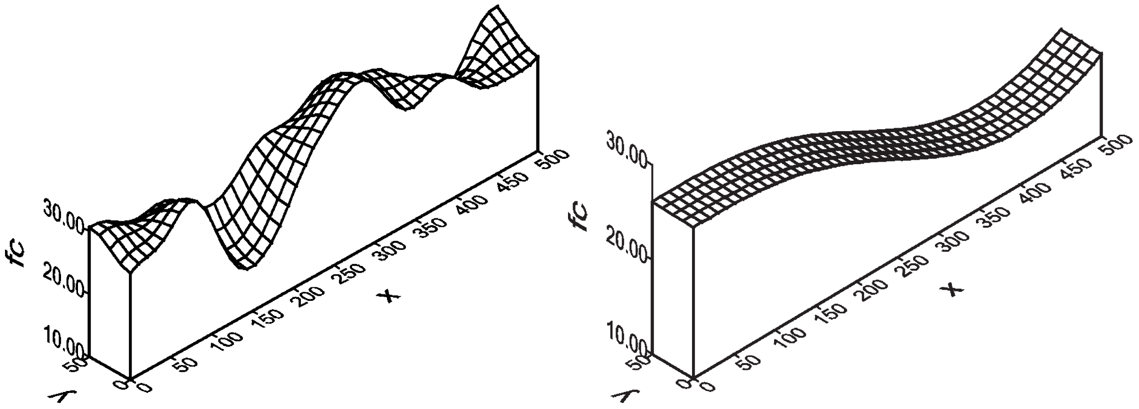
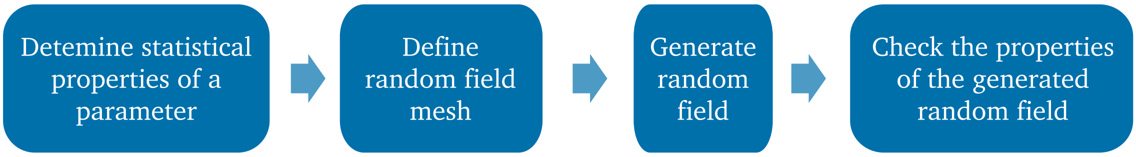


Fig. 2.4.: Random fields for the compressive strength in a concrete beam of 500 cm long. Left L_c=50 cm, right L_c=200 cm [11].

2.3 Generation of random fields

To generate a random field several steps have to be taken. This process is visualised in figure 2.5. In the first step statistical properties like the distribution type, mean, variance and covariance of a parameter are determined. For most of the parameters the statistical properties can be found in databases which are based on experimental data. The next step is to define a random field mesh. This can be the same mesh as the finite element mesh, but most of the time a separate mesh is more desirable. On locations where the stress gradient is high the spatial variability does not have to be high and vice versa. Next to that, also a refinement of the finite element mesh can be sometimes desirable. In such case it is more practical that the random field mesh can be left unchanged. The size of the elements of the random field mesh depends on the type of correlation function and the correlation length. According to Sudret [1] the element size of the RF-mesh (L_{RF}) should be taken between $L_c/10$ and $L_c/5$ for the exponential correlation function and should be taken between $L_c/4$ and $L_c/2$ in other cases. After the random field mesh is defined, a random field can be generated. Several methods are available to generate a random field. Which method is the most efficient and accurate to generate a random field differs for every parameter and case. It depends for example on the type of correlation function and the size of the random field.


Detemine statistical properties of a parameter

Define random field mesh

Generate random field

Check the properties of the generated random field

Fig. 2.5.: Process random field generation.

After the random field is generated a check should be performed to see if the field is accurate enough. An evaluation of the generated field can be done to see if the characteristics like the mean, variance and covariance are represented well by the field, i.e. if they correspond to the input values of the random field generator. If the required accuracy is not met the random field mesh has to be refined or another method to generate a random field has to be selected.

2.3.1 Nataf transformation

All the random field generators generate a Gaussian random field. However, for concrete properties a non-Gaussian field, having a log-normal distribution, is desirable. To acquire such a random field, a standard normal distributed field has to be transformed. The values in the random field can be transformed according to equation 2.26. Where the mean and the standard deviation are transformed according to equation 2.27. Also the correlation function has to be transformed. This has to be done in such a way that after the transformation, the obtained random field has a correlation structure which corresponds with the untransformed correlation function $\rho(\Delta x)$. The transformation of the correlation function is given by:

$$\rho_T(\Delta x) = \frac{\exp(\sigma_T^2 \rho(\Delta x)) - 1}{\exp(\sigma_T^2) - 1} \tag{2.65}$$

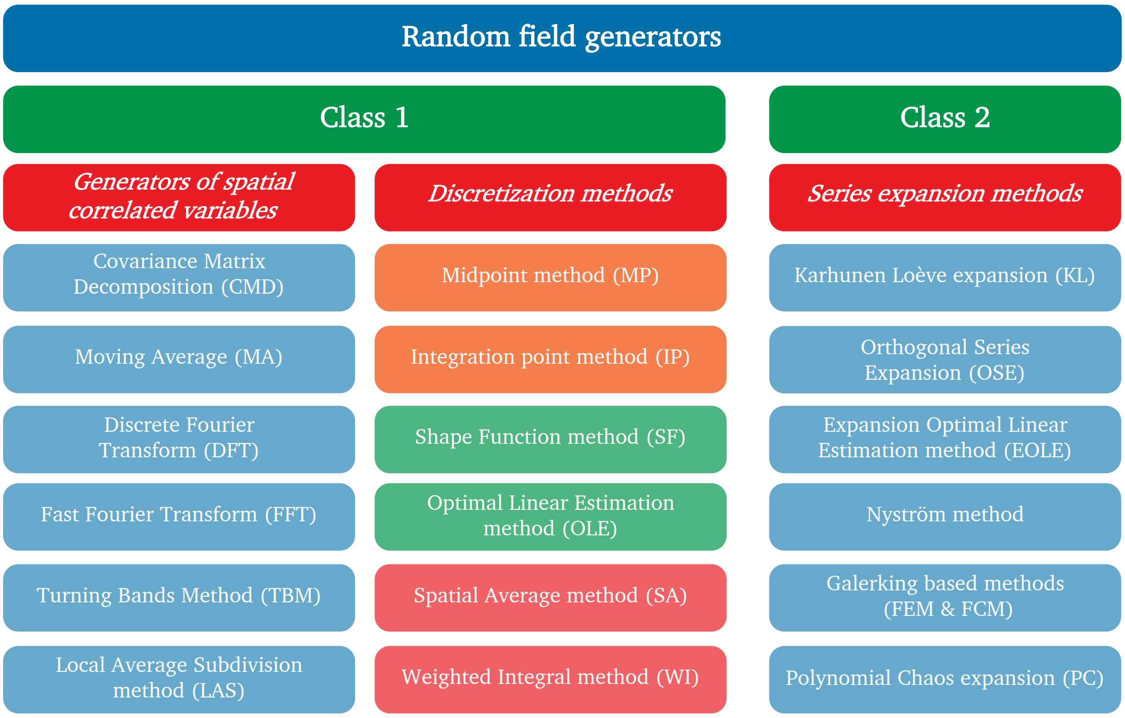
Where $\rho_T(\Delta x)$ is the transformed correlation function, $\rho(\Delta x)$ is the untransformed correlation function and σ_T is the transformed standard deviation. This transformation is called the *Nataf transformation* [12]. The derivation of equation 2.65 is given in appendix B

The variance function and SDF of the transformed covariance function can not be derived exact. The variance function will be approximated numerically. For the SDF it will be assumed that the transformed SDF is approximately the same as the untransformed SDF. This is a good assumption when the mean value is at least twice as big as the standard deviation of the log-normal random field.

2.3.2 Classification of random field generators

In literature a number of methods are available to generate a random field. These methods can be divided into two classes. An overview of these methods and classification is shown in figure 2.6. First the two different classes will be introduced shortly whereafter all the different methods will be explained in more detail.

Class 1 generators In the first class, *generators of spatially correlated random variables* are combined with a *discretization method*. For every node in the random field mesh a random variable is evaluated which is correlated to the other nodes in the random field. These spatially correlated random variables have to be allocated to an element or integration point in the FE model. This can be done with different discretization methods. These discretization methods can be divided into three categories. The first category are the point discretization methods (orange blocks) which result in a piecewise constant random field. The second category are the point discretization methods (green blocks) which result in a continuous random field. In these methods use is made of interpolation function between the spatially correlated random variables. The third category are the so-called average discretization methods (red blocks). Not all combinations are possible between the generators of spatially correlated random variables (1st column) and discretization methods (2nd column) as will be shown at the end of section 2.5.



Random field generators		
Class 1		Class 2
Generators of spatial correlated variables	*Discretization methods*	*Series expansion methods*
Covariance Matrix Decomposition (CMD)	Midpoint method (MP)	Karhunen Loève expansion (KL)
Moving Average (MA)	Integration point method (IP)	Orthogonal Series Expansion (OSE)
Discrete Fourier Transform (DFT)	Shape Function method (SF)	Expansion Optimal Linear Estimation method (EOLE)
Fast Fourier Transform (FFT)	Optimal Linear Estimation method (OLE)	Nyström method
Turning Bands Method (TBM)	Spatial Average method (SA)	Galerking based methods (FEM & FCM)
Local Average Subdivision method (LAS)	Weighted Integral method (WI)	Polynomial Chaos expansion (PC)

Fig. 2.6.: Classification random field generators.

Class 2 generators The second class of random field generators are the so-called *series expansion methods*. The random field is represented by a sum of functions which are multiplied by a random variable. These methods result in a continuous random field. It can be reasoned that the Discrete and Fast Fourier Transform methods belong to class 2 since these methods yield a series expansion of sinusoids to generate the random field. However the generated field is not continuous, it is only determined on the nodes of the random field. Therefore it is classified as a class 1 method.

In the FE model the continuous random fields have to be numerically integrated. This comes down to a summation of the evaluated values of the continuous functions at the integration points multiplied by the weight of the corresponding integration point. Which integration scheme is the most accurate is questionable, since the continuous function does not have to be an nth-order polynomial.

If an uncorrelated random field is desired these methods do not have to be applied. In that case a random number generator belonging to the distribution type of the parameter has to be used. The random numbers can be generated for every element or every integration point in the FE model.

In the section 2.4 the generators of spatially correlated random variables will be explained. In section 2.5 the different discretization methods will be explained. However, the weighted integral method will not be considered in this report. In section 2.6 the class 2 methods will be explained briefly.

2.4 Class 1 generators

In this section the different methods to generate spatially correlated random variables are explained. These correlated random variables are collected in a vector as follows:

In 1D
$$\mathbf{z_c}(x) = z_c(x_i) = \begin{bmatrix} z_c(x_1) \\ \vdots \\ z_c(x_n) \end{bmatrix} \tag{2.66a}$$

In 2D and 3D
$$\mathbf{z_c}(\mathbf{x}) = z_c(\mathbf{x_i}) = \begin{bmatrix} z_c(\mathbf{x_1}) \\ \vdots \\ z_c(\mathbf{x_n}) \end{bmatrix} \tag{2.66b}$$

Where N is equal to the number of nodes in the random field mesh and $\mathbf{x_i}$ gives the nodal location of the i$^{\text{th}}$ node. The correlated random variables $z_c(\mathbf{x_i})$ are only evaluated on the locations of the nodes. When the generator is coupled with a discretization method a random field is acquired which can be implemented in a FE model. In 2D the correlated random variables can also be expressed in a matrix to match its 2D character.

$$\mathbf{Z_c}(\mathbf{x}) = Z_c(x_i, y_j) = \begin{bmatrix} Z_c(x_1, y_1) & Z_c(x_1, y_2) & \dots & Z_c(x_1, y_m) \\ Z_c(x_2, y_1) & Z_c(x_2, y_2) & \dots & Z_c(x_2, y_m) \\ \vdots & \vdots & \ddots & \vdots \\ Z_c(x_n, y_1) & Z_c(x_n, y_2) & \dots & Z_c(x_n, y_m) \end{bmatrix} \tag{2.67}$$

Where N is equal to the number of nodes in x direction and M is equal to the number of nodes in the y direction in the random field mesh. In 3D it becomes a three-dimensional matrix.

All the methods in this section describe how spatially correlated random variables, which are standard normal distributed, can be generated. These random variables ($\mathbf{X}$) can be transformed to normally distributed random variables having a variance equal to σ^2 and a mean equal to μ ($\mathbf{Y}$) as follows:

$$\mathbf{Y} = \mu + \sigma\mathbf{X} \tag{2.68}$$

To obtain spatially correlated random variables which are log-normally distributed, the Nataf transformation, described in section 2.3.1, can be applied.

The information in this section is based on the following references [4, 5, 13, 14]

2.4.1 Covariance Matrix Decomposition method (CMD)

With the *Covariance Matrix Decomposition* method (CMD) a set of correlated random variables can be generated as follows:

$$\mathbf{z_c}(\mathbf{x}) = \mathbf{L}\boldsymbol{\chi} \tag{2.69}$$

where:

$$\begin{aligned} \mathbf{z_c}(\mathbf{x}) &= \text{vector containing spatially correlated random variables} \\ \boldsymbol{\chi} &= \text{vector of independent zero mean, unit variance, normally distributed} \\ &\quad\ \text{random variables} \\ \mathbf{L} &= \text{decomposed correlation matrix} \end{aligned}$$

For a set of N random variables which are collected in vector $\mathbf{y}$, the correlation matrix is defined as:

$$R_{ij} = \frac{\text{Cov}(y_i, y_j)}{\sqrt{\text{Var}(y_i)\text{Var}(y_j)}} = \begin{bmatrix} 1 & \rho(y_1, y_2) & \dots & \rho(y_1, y_n) \\ & 1 & & \rho(y_2, y_n) \\ & & \ddots & \vdots \\ symmetric & & & 1 \end{bmatrix} \tag{2.70}$$

This matrix has to be decomposed such that the multiplication of $\mathbf{L}\boldsymbol{\chi}$ results in a vector ($\boldsymbol{\chi_c}$) of normally distributed random variables having a zero mean, a unit variance and a specific correlation. Using equation 2.29 it can be shown that the correlation matrix of the vector $\mathbf{x_c}$ can be decomposed into two matrices as follows:

$$\begin{aligned} \mathbf{R} &= \text{Cov}[\boldsymbol{\chi_c}, \boldsymbol{\chi_c}] = E[\boldsymbol{\chi_c}\boldsymbol{\chi_c}^T] - 0 \cdot 0 = E[\boldsymbol{\chi_c}\boldsymbol{\chi_c}^T] = E[\mathbf{L}\boldsymbol{\chi}\mathbf{L}\boldsymbol{\chi}^T] = \mathbf{L}E[\boldsymbol{\chi}\boldsymbol{\chi}^T]\mathbf{L}^T \\ &= \mathbf{L}\mathbf{I}\mathbf{L}^T = \mathbf{L}\mathbf{L}^T \end{aligned} \tag{2.71}$$

Where matrix $\mathbf{L}$ can be taken out of the expectation since the elements only consist constants.

The correlation matrix is symmetric and positive definite. It can therefore be decomposed using *Cholesky decomposition* and *eigendecomposition* to obtain a real valued matrix $\mathbf{L}$. With Cholesky decomposition the matrix is decomposed in an upper triangular matrix and a symmetric lower triangular matrix like $\mathbf{R} = \mathbf{L}\mathbf{L}^T$.

The eigendecomposition of a square matrix entails the following factorization:

$$\mathbf{R} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q} \tag{2.72}$$

Where $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues (λ_i) of matrix $\mathbf{R}$ on the diagonal and matrix $\mathbf{Q}$ contains the associated eigenvectors. The matrix $\mathbf{L}$ is obtained as follows:

$$\mathbf{R} = \mathbf{Q}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{\Lambda}}\mathbf{Q} = \mathbf{L}\mathbf{L}^T \rightarrow \mathbf{L} = \mathbf{Q}\tilde{\boldsymbol{\Lambda}} \tag{2.73}$$

Where $\tilde{\boldsymbol{\Lambda}} = \text{diag}(\sqrt{\boldsymbol{\lambda}})$

The values in the correlation matrix (equation 2.70) can be evaluated using a correlation functions. In the next chapter, two common used correlation functions are given in equation 3.1 and 3.2. The coordinates of the nodes associated with the two random variables are the arguments for this correlation function.

The computation becomes very computational intensive as N increases, i.e. the matrices become very large. Also the method is more prone to round off errors when N increases [14]. So this method is less suited for random fields with many nodes.

2.4.2 Moving Average method (MA)

With the *Moving Average* method (MA) the correlated random variables are constructed by averaging an underlying white noise process. In 1D the values of the random field can be calculated with:

$$z_c(x_i) = \sum_{k=1}^{N} f(x_k - x_i) W_k \tag{2.74}$$

Where W_k is the discrete white noise process, which has a zero mean and unit variance and f is a weighting function which satisfies:

$$\sigma^2 \rho(\Delta x) = \int_{-\infty}^{\infty} f(x) f(x - \Delta x) dx \tag{2.75}$$

This method is computationally very slow and it is difficult to compute f for an arbitrary correlation function [14].

2.4.3 Discrete Fourier Transform method (DFT)

In the *Discrete Fourier Transform* method (DFT) the random field is represented as a sum of sinusoids. In order to obtain the values for the correlated random variables the following function is evaluated at every grid point of the random field mesh:

$$z_c(x_i) = \sum_{k=1}^{N} A_k \text{cos}(\omega_k x_i) + B_k \text{sin}(\omega_k x_i) \tag{2.76}$$

The A_k and B_k coefficients are mutually independent and normally distributed random variables with zero mean and variances equal to:

$$E[A_1^2] = E[B_1^2] = \frac{1}{2} G(\omega_0) \Delta\omega \tag{2.77}$$

$$E[A_k^2] = E[B_k^2] = G(\omega_k) \Delta\omega, \qquad \text{for } k = 2, 3, \ldots, N \tag{2.78}$$

Where $G(\omega)$ is the one sided spectral density function. This function is discretized into N parts. An example of this function and the determination of the variances of the coefficients A_k and B_k can be found in figure 2.7. In equation 2.76 only the real part of the discrete Fourier transform is given since the random field only consist of real values. The DFT method is computationally very slow, especially in 2D and 3D [14].

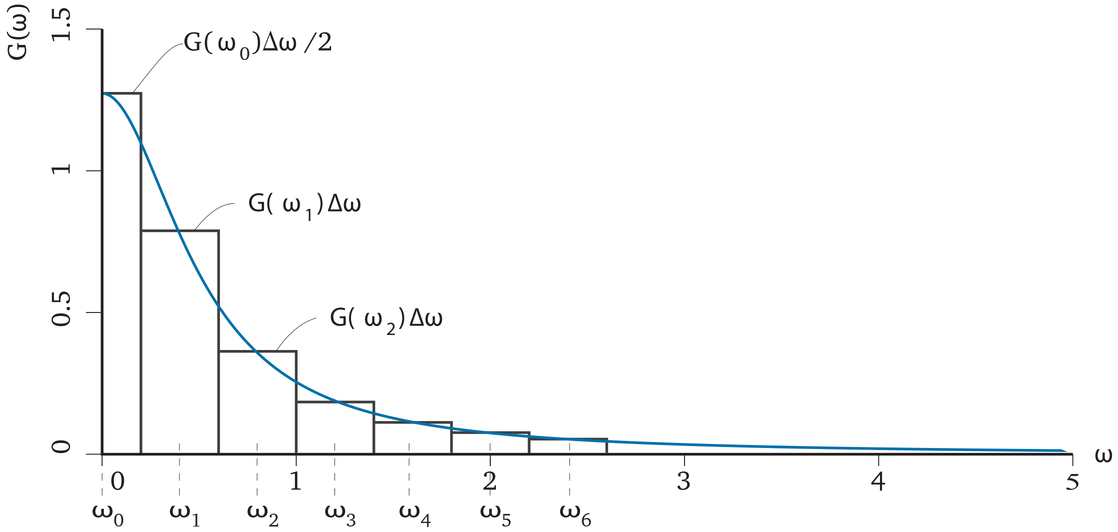


Fig. 2.7.: Spectral density function which is discretized into 7 parts. The area under the graph per part gives the variance for the coefficients [13].

2.4.4 Fast Fourier Transform method (FFT)

As the name indicates, the *Fast Fourier Transform* method (FFT) can calculate the Discrete Fourier Transform in equation 2.76 more efficient. For this, the number of spatial points is taken the same as the number of frequency points ($N = K$) and they are discretized equispaced according to:

$$x_j = (j-1)\Delta x = \frac{(j-1)D}{N-1}, \qquad \text{for } j = 1, 2, \ldots, N \tag{2.79}$$

$$\omega_k = (k-1)\Delta\omega = \frac{2\pi(k-1)(N-1)}{ND}, \qquad \text{for } k = 1, 2, \ldots, N \tag{2.80}$$

where D is the size of the random field mesh and N is equal to the number of nodes in the random field mesh. In this method N has to be equal to a power of 2, i.e. 2^m.

Based on this discretization and using the inverse Fourier relationships, the variances of the coefficients A_k and B_k can be found [4], which results in:

$$\text{Var}[A_k] = \begin{cases} \frac{1}{2}G(\omega_k)\Delta\omega & \text{if } k = 1 \\ \frac{1}{4}[G(\omega_k) + G(\omega_{N-k+2})]\Delta\omega & \text{if } k = 2, \ldots, \frac{N}{2} \\ G(\omega_k)\Delta\omega & \text{if } k = 1 + \frac{N}{2} \end{cases} \tag{2.81}$$

$$\text{Var}[B_k] = \begin{cases} 0 & \text{if } k = 1 \text{ or } k = 1 + N/2 \\ \frac{1}{4}[G(\omega_k) + G(\omega_{N-k+2})]\Delta\omega & \text{if } k = 2, \ldots, \frac{N}{2} \end{cases} \tag{2.82}$$

The coefficients have the following symmetries [5]:

$$A_k = \frac{1}{N}\sum_{j=1}^{N} z_c(x_j)\cos(2\pi\frac{(j-1)(k-1)}{N}) = A_{N-k+2} \tag{2.83a}$$

$$B_k = \frac{1}{N}\sum_{j=1}^{N} z_c(x_j)\sin(2\pi\frac{(j-1)(k-1)}{N}) = -B_{N-k+2} \tag{2.83b}$$

This arises from the equispaced discretization and the real valued values of $z_c(x_j)$. Therefore only half of the coefficients have to be generated. The values of the spatially correlated random variables in 1D can then be calculated by applying a FFT algorithm on the following Fourier transform:

$$z_c(x_j) = \sum_{k=1}^{N} [A_k - iB_k] \exp\left(i2\pi(j-1)(k-1)/N\right) \tag{2.84}$$

FFT method in 2D According to [4], the discrete Fourier transform in 2D and its characteristics are given by:

$$\begin{aligned} Z_c(x_i, y_j) &= \sum_{l=1}^{N_x} \sum_{m=1}^{N_y} A_{lm} \cos\left(\frac{2\pi(l-1)(i-1)}{N_x} + \frac{2\pi(m-1)(j-1)}{N_y}\right) \\ &+ B_{lm} \sin\left(\frac{2\pi(l-1)(i-1)}{N_x} + \frac{2\pi(m-1)(j-1)}{N_y}\right) \end{aligned} \tag{2.85}$$

Where N_x and N_y are the number of nodes in the random field in x and y direction respectively. Using the inverse Fourier transform relationships will result in the following planar symmetries for the Fourier coefficients:

$$\begin{aligned} A_{N_x-l+2, N_y-m+2} &= A_{l,m}, & B_{N_x-l+2, N_y-m+2} &= -B_{l,m} \\ A_{N_x-l+2, m} &= A_{l, N_y-m+2}, & B_{N_x-l+2, m} &= -B_{l, N_y-m+2} \end{aligned} \tag{2.86}$$

for $l, m = 2, 3, \ldots, 1 + \frac{N_\alpha}{2}$, where N_α is either N_x or N_y. Where it is convenient to choose the direction with the highest number of nodes. Also the following line symmetries can be found:

$$\begin{aligned} A_{n, N_y-m+2} &= A_{n,m}, & B_{n, N_y-m+2} &= -B_{n,m} \\ A_{N_x-l+2, n} &= A_{l,n}, & B_{N_x-l+2, n} &= -B_{l,n} \end{aligned} \tag{2.87}$$

which hold for the same range of l and m as for the planar symmetries and n equal to 1 or $1 + \frac{N_\alpha}{2}$. The graphical impression of the symmetries is shown in figure 2.8.

Only the coefficients in the grey areas and on the four half lines along $n = 1$ or $1 + \frac{N_\alpha}{2}$ have to be generated. The mean value of these coefficients is again equal to zero. The variances can be determined as follows:

$$\mathrm{E}[A_{lm}^2] = \Delta\omega_x \Delta\omega_y G^d(\omega_l, \omega_m) \qquad \text{for } l = 1, 1 + \frac{N_x}{2} \text{ and } m = 1, 1 + \frac{N_y}{2} \tag{2.88}$$

$$\mathrm{E}[B_{lm}^2] = 0 \qquad \text{for } l = 1, 1 + \frac{N_x}{2} \text{ and } m = 1, 1 + \frac{N_y}{2} \tag{2.89}$$

and

$$\mathrm{E}[A_{lm}^2] = \mathrm{E}[B_{lm}^2] = \begin{cases} \frac{1}{4}\Delta\omega_x\Delta\omega_y(G^d(\omega_l,\omega_m) + G^d(\omega_l, \omega_{N_y-m+2})) & \text{for } l = 1, 1 + \frac{N_x}{2} \\ \frac{1}{4}\Delta\omega_x\Delta\omega_y(G^d(\omega_l,\omega_m) + G^d(\omega_{N_x-l+2}, \omega_m)) & \text{for } m = 1, 1 + \frac{N_y}{2} \\ \frac{1}{8}\Delta\omega_x\Delta\omega_y(G^d(\omega_l,\omega_m) + G^d(\omega_l, \omega_{N_y-m+2}) & \\ +G^d(\omega_{N_x-l+2}, \omega_m) + G^d(\omega_{N_x-l+2}, \omega_{N_y-m+2})) & \text{otherwise} \end{cases} \tag{2.90}$$

where

$$G^d(\omega_l, \omega_m) = \frac{G(\omega_l, \omega_m)}{2^d} \tag{2.91}$$

where d is the number of components of (ω_l, ω_m) equal to zero. The values of the spatially correlated random variables in 2D can then be calculated by applying a FFT algorithm on the discrete Fourier transform in equation 2.85.

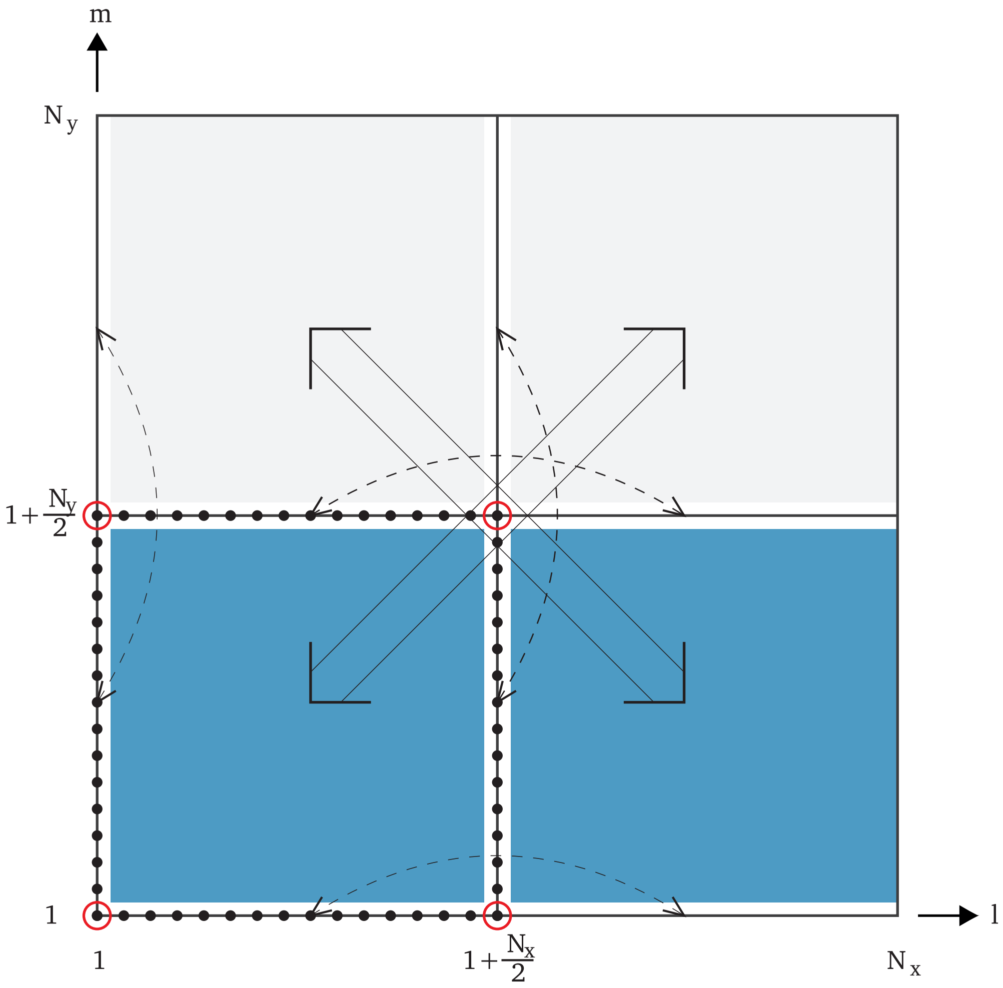


Fig. 2.8.: Graphical impression of the planar and line symmetries in the variances of the Fourier coefficients [4].

Performance FFT method According to *Fenton* [4], the Fast Fourier Transform is computationally more efficient than the preceding methods. However, to acquire accurate results, the method has to be applied carefully. As shown in [4], the covariance function of a real FFT process is symmetric about the midpoint of the field regardless of the desired target correlation structure. To bypass this problem, the random field size has to be enlarged at least with L_c in every dimension. After the generation of spatially correlated random variables the variables outside the original domain are left out. Within the original domain the variables will approximately have the desired target covariance structure. Another issue of the FFT method has to do with frequency discretization. This discretization is coupled to the space discretization. The frequency discretization has to be sufficiently fine, in order to approximate the area under the one-sided SDF appropriately. Therefore, the random field mesh has to be determined carefully to be sure the frequency discretization is fine enough to omit erroneous results.

2.4.5 Turning Bands Method (TBM)

The *Turning Bands Method* (TBM) can only generate spatially correlated random variables for 2D and 3D random fields. They can be generated from a series of 1D processes. First a number of lines with a random direction are defined in the domain. For every line a random process is simulated with the FFT method. The values at the grid points of the random field mesh are determined by the sum of orthogonal projections of the 1D processes to that point divided by the square root of the number of lines. In figure 2.9 the turning bands method with two lines is visualised. Also the formula to determine the values of the random field is given.

Initially, this method is proposed by Matheron [15]. The performance of this method is considered by Fenton in [4]. It is stated that if 16 lines are used in 2D, the method is as efficient as the FFT method in 2D. However, more lines have to be used to acquire no distortion in the field. It is mentioned that 64 lines are sufficient. Yet, the method becomes 3-5 times slower than the FFT method. Another issue of this method is the selection of the correlation function or the SDF for the 1D process. This function has to be chosen such that correlated random variables, for the random field in 2D or 3D, have the desired target correlation structure. To find this correlation function or SDF is quite hard in general. If the number of lines is increased, the method becomes very accurate. The efficiency although decreases as the number of lines increases.

$$Z_c(x_i) = \frac{1}{\sqrt{L}} \sum_{k=1}^{L} H_k(\mathbf{x}_i \cdot \mathbf{u}_k)$$

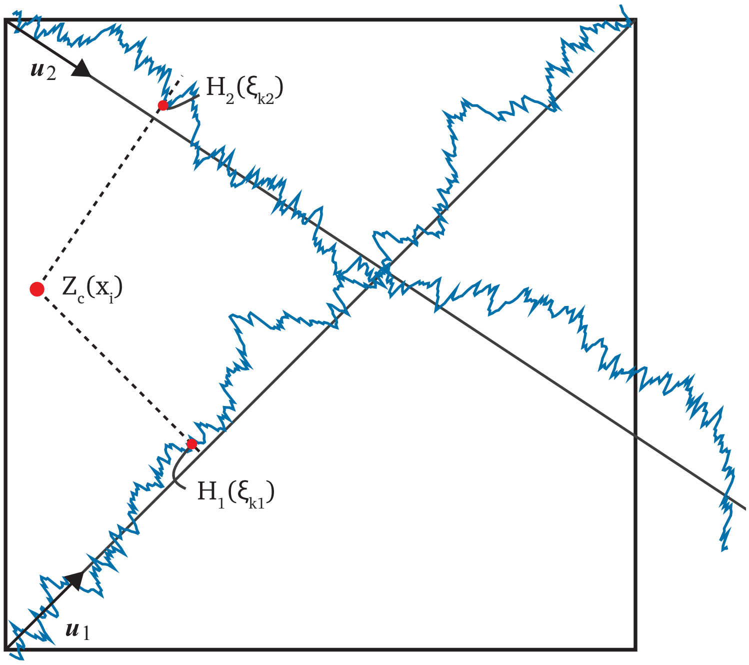


Fig. 2.9.: Visualisation of the turning bands method [4].

2.4.6 Local Average Subdivision method (LAS)

The *Local Average Subdivision* method (LAS) is a fast and accurate method to generate discontinuous random fields which are locally averaged. In this method the spatially correlated random variables are automatically discretized with the *spatial average* method to obtain the random field. The variance of the random variables is reduced based on the element size of the random field mesh. As explained in section 2.2.2, this reduction can be determined with the variance function. In section

2.5.5, the spatial average discretization method will be explained in more detail. In this section the method as proposed by Fenton in [4, 14] will be explained.

The LAS algorithm starts with a very coarse random field mesh. With this mesh, a discontinuous random field is generated using the CMD method combined with the spatial average discretization method. As mentioned in section 2.4.1, the CMD method performs well for fields with a low number of nodes. Thereafter the random field mesh is repeatedly divided till the desired coarseness is obtained. This top down approach is visualised schematically in figure 2.10, where the starting random field mesh has only one element. In this method not the nodes but the centre of the elements are taken as reference point to determine the covariance between the different random variables. In the following text the word *cells* will be used which refers to the elements of the random field mesh. First the LAS method in 1D will be explained, next the 2D algorithm will be explained and thereafter the performance of the method will be discussed.

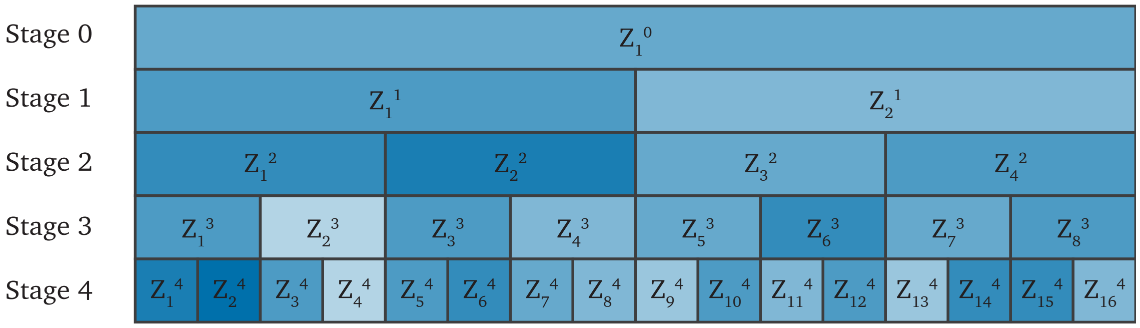

Stage 0
Stage 1
Stage 2
Stage 3
Stage 4
Z_1^0
Z_1^1 Z_2^1
Z_1^2 Z_2^2 Z_3^2 Z_4^2
Z_1^3 Z_2^3 Z_3^3 Z_4^3 Z_5^3 Z_6^3 Z_7^3 Z_8^3
Z_1^4 Z_2^4 Z_3^4 Z_4^4 Z_5^4 Z_6^4 Z_7^4 Z_8^4 Z_9^4 Z_{10}^4 Z_{11}^4 Z_{12}^4 Z_{13}^4 Z_{14}^4 Z_{15}^4 Z_{16}^4

Fig. 2.10.: Top-down approach of the local average subdivision process[5].

LAS method in 1D As mentioned above, the algorithm starts with the generation of a random field with a coarse random field mesh, thereafter the cells are divided repeatedly. The values of the cells which are divided in stage *i* are indicated with $Z_c^i(x_{2j-1})$ and $Z_c^i(x_{2j})$, where j is the cell number of the parent cell in previous stage. This numbering is visualised in figure 2.11. The values of the newly created cells are normally distributed and should fulfil the following requirements:

1. They posses the correct variance. The variance is reduced according to the variance function (equation 2.58), where the new cell size is taken as domain size.
2. They are properly correlated with each another according to equation 2.55.
3. The average value is equal to the parent cell value, i.e. $\frac{1}{2}(Z_c^i(x_{2j-1}) + Z_c^i(x_{2j})) = Z_c^{i-1}(x_j)$.
4. They are properly correlated with the neighbouring cells of the parent cell, $Z_c^{i-1}(x_{j-1})$ and $Z_c^{i-1}(x_{j+1})$.

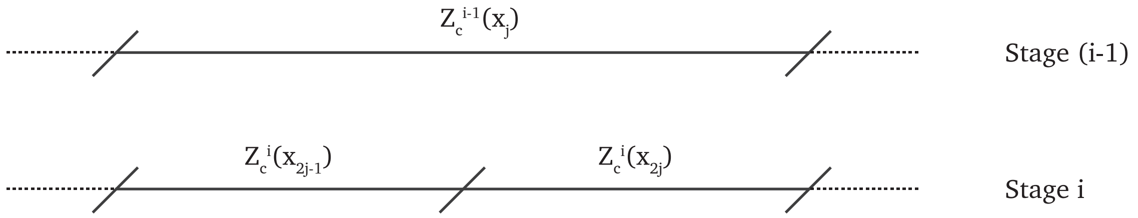


Fig. 2.11.: Numbering cells LAS in 1D.

Because the value of each cell is correlated with its neighbours, in the end the random field will have the desired target covariance structure. The value of the right cell of one of the divided cells in stage *i* is determined by estimating its mean and adding a zero mean discrete white noise $c^i W_j^i$ having variance $(c^i)^2$ which results in:

$$Z_c^i(x_{2j}) = M_{2j}^i + c^i W_j^i \tag{2.92}$$

Where M_{2j}^i is the best linear estimate for the mean, which can be determined by a linear combination of the values of the neighbouring cells in the previous stage as follows:

$$M_{2j}^i = \sum_{k=j-n}^{j+n} a_{k-j}^{i-1} Z_c^{i-1}(x_k) \tag{2.93}$$

The number of neighbouring cells *n* can be chosen equal to 1 or 2. The value of its accompanying left cell ($Z_c^i(x_{2j-1})$) can be determined using requirement three which results in:

$$Z_c^i(x_{2j-1}) = 2Z_c^{i-1}(x_j) - Z_c^i(x_{2j}) \tag{2.94}$$

The c and a coefficients for every stage can be found by following the four requirements mentioned above. To find the values for the a coefficients, equation 2.92 is multiplied by $Z_c^{i-1}(x_m)$, with $m = j-n, \dots, j+n$. Taking expectations and using the fact that W_j^i is uncorrelated with the stage *i-1* values result in:

$$\mathrm{E}[Z_c^i(x_{2j}) Z_c^{i-1}(x_m)] = \sum_{k=j-n}^{j+n} a_{k-j}^{i-1} \mathrm{E}[Z_c^{i-1}(x_k) Z_c^{i-1}(x_m)] \tag{2.95}$$

The c coefficients can be determined by squaring equation 2.92, taking expectations and employing the result of equation 2.95, which results in:

$$(c^i)^2 = \mathrm{E}[Z_c^i(x_{2j})^2] - \sum_{k=j-n}^{j+n} a_{k-j}^{i-1} \mathrm{E}[Z_c^i(x_{2j}) Z_c^{i-1}(x_k)] \tag{2.96}$$

The cross-stage covariances ($\mathrm{E}[Z_c^i(x_{2j}) Z_c^{i-1}(x_m)]$) in equation 2.95 and 2.96 can be determined with:

$$\mathrm{E}[Z_c^i(x_{2j}) Z_c^{i-1}(x_m)] = \frac{1}{2}\mathrm{E}[Z_c^i(x_{2j}) Z_c^i(x_{2m-1})] + \frac{1}{2}\mathrm{E}[Z_c^i(x_{2j}) Z_c^{i-1}(x_{2m})] \tag{2.97}$$

Where use is made of the third requirement, which demands upwards averaging.

The covariances in the equations above can be determined using equation 2.55 or 2.61. In these equations the domain size of a cell in stage i is determined with:

$$D^i = \frac{D}{k_1 2^i} \tag{2.98}$$

Where D is the domain of the random field and k_1 is the number of cells of the random field in stage 0.

At the boundaries of the random field, the algorithm may require values from cells of a previous stage which lie outside the domain of the random field. Fenton [4] handles with this problem by assuming that what happens outside the domain is uncorrelated with what happens within the domain. This may lead to an error, but it is considered to be insignificant.

LAS method in 2D In two dimensions the rectangular cells are repeatedly divided into four equal sized cells. The division of one cell and the numbering is visualised in figure 2.12. The values of the cells which are divided in stage *i* are indicated with $Z_c^i(x_{2j-1}, y_{2k-1})$, $Z_c^i(x_{2j-1}, y_{2k})$, $Z_c^i(x_{2j}, y_{2k-1})$ and $Z_c^i(x_{2j}, y_{2k})$, where j and k are the cell numbers of the parent cell in previous stage. The same requirements as in the 1D algorithm are used to come to the following expression of the new cell values:

$$Z_1^i = Z_c^i(x_{2j-1}, y_{2k-1}) = c_{11}^i W_{1jk}^i + \sum_{l=1}^{n_{xy}} a_{l1}^{i-1} Z_c^{i-1}(x_{m(l)}, y_{n(l)}) \tag{2.99a}$$

$$Z_2^i = Z_c^i(x_{2j-1}, y_{2k}) = c_{21}^i W_{1jk}^i + c_{22}^i W_{2jk}^i + \sum_{l=1}^{n_{xy}} a_{l2}^{i-1} Z_c^{i-1}(x_{m(l)}, y_{n(l)}) \tag{2.99b}$$

$$Z_3^i = Z_c^i(x_{2j}, y_{2k-1}) = c_{31}^i W_{1jk}^i + c_{32}^i W_{2jk}^i + c_{33}^i W_{2jk}^i + \sum_{l=1}^{n_{xy}} a_{l3}^{i-1} Z_c^{i-1}(x_{m(l)}, y_{n(l)}) \tag{2.99c}$$

$$Z_4^i = Z_c^i(x_{2j}, y_{2k}) = 4Z_c^{i-1}(x_j, y_k) - Z_c^i(x_{2j-1}, y_{2k-1}) - Z_c^i(x_{2j-1}, y_{2k}) - Z_c^i(x_{2j}, y_{2k-1}) \tag{2.99d}$$

Where $m(l)$ and $n(l)$ are functions which indicate the neighbouring cells in the previous stage which are shown in orange in figure 2.12. The a coefficients can be determined with:

$$\mathrm{E}[Z_c^i(x_{2j}, y_{2k}) Z_c^{i-1}(x_{m(p)}, y_{n(p)})] = \sum_{l=1}^{n_{xy}} a_{l1}^{i-1} \mathrm{E}[Z_c^{i-1}(x_{m(l)}, y_{n(l)}) Z_c^{i-1}(x_{m(p)}, y_{n(p)})] \tag{2.100a}$$

$$\mathrm{E}[Z_c^i(x_{2j}, y_{2k-1}) Z_c^{i-1}(x_{m(p)}, y_{n(p)})] = \sum_{l=1}^{n_{xy}} a_{l2}^{i-1} \mathrm{E}[Z_c^{i-1}(x_{m(l)}, y_{n(l)}) Z_c^{i-1}(x_{m(p)}, y_{n(p)})] \tag{2.100b}$$

$$\mathrm{E}[Z_c^i(x_{2j-1}, y_{2k}) Z_c^{i-1}(x_{m(p)}, y_{n(p)})] = \sum_{l=1}^{n_{xy}} a_{l3}^{i-1} \mathrm{E}[Z_c^{i-1}(x_{m(l)}, y_{n(l)}) Z_c^{i-1}(x_{m(p)}, y_{n(p)})] \tag{2.100c}$$

Where p varies from 1 to n_{xy}, which results in a covariance matrix at the RHS of the formulas in equation 2.100. To find the c coefficients, it is assumed that they can be found in the lower triangular matrix $\mathbf{c}^i$, according to:

$$\mathbf{c}^i(\mathbf{c}^i)^T = \mathbf{R} \tag{2.101}$$

Where $\mathbf{R}$ is a symmetric matrix and is given by:

$$R_{rs} = \mathrm{E}[Z_r^i Z_s^i] - \sum_{l=1}^{n_{xy}} a_{lr}^{i-1} \mathrm{E}[Z_c^{i-1}(x_{m(l)}, y_{n(l)}) Z_s^i] \qquad \text{for } r, s = 1, 2, 3 \tag{2.102}$$

In this equation the index notation, which is given at the left hand side of equation 2.99, is used. In green the same index notation can be found in figure 2.12.

The cross-stage covariances in the formulas above can be determined by considering the formula which ensures upwards averaging, i.e.

$$Z_c^{i-1}(x_m, y_n) = \frac{1}{4} Z_c^i(x_{2m}, y_{2n}) + Z_c^i(x_{2m-1}, y_{2n}) + Z_c^i(x_{2m}, y_{2n-1}) + Z_c^i(x_{2m-1}, y_{2n-1}) \tag{2.103}$$

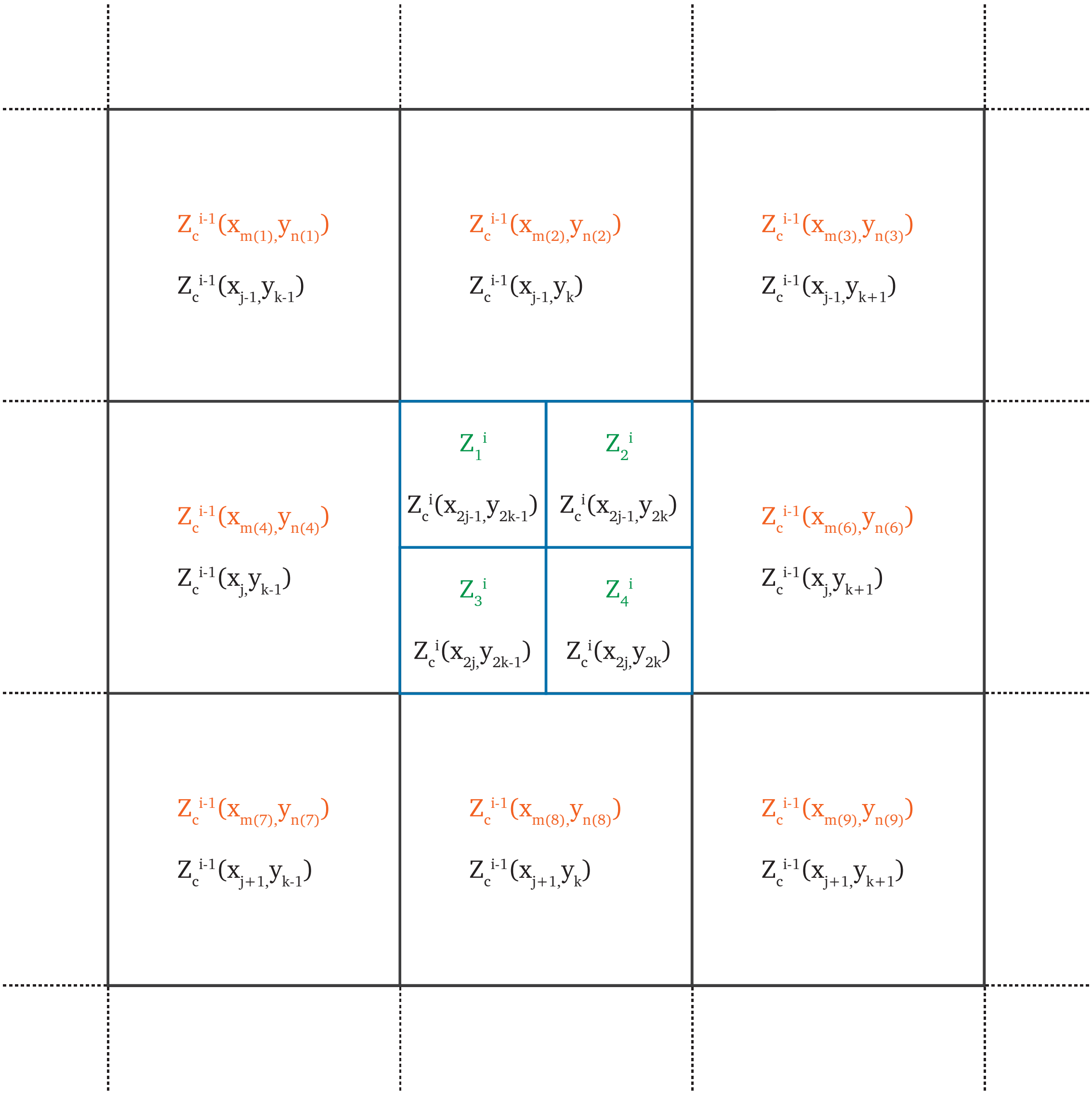

$Z_c^{i-1}(x_{m(1)},y_{n(1)})$
$Z_c^{i-1}(x_{j-1},y_{k-1})$
$Z_c^{i-1}(x_{m(2)},y_{n(2)})$
$Z_c^{i-1}(x_{j-1},y_{k})$
$Z_c^{i-1}(x_{m(3)},y_{n(3)})$
$Z_c^{i-1}(x_{j-1},y_{k+1})$
$Z_c^{i-1}(x_{m(4)},y_{n(4)})$
$Z_c^{i-1}(x_{j},y_{k-1})$
Z_1^i
$Z_c^i(x_{2j-1},y_{2k-1})$
Z_2^i
$Z_c^i(x_{2j-1},y_{2k})$
Z_3^i
$Z_c^i(x_{2j},y_{2k-1})$
Z_4^i
$Z_c^i(x_{2j},y_{2k})$
$Z_c^{i-1}(x_{m(6)},y_{n(6)})$
$Z_c^{i-1}(x_{j},y_{k+1})$
$Z_c^{i-1}(x_{m(7)},y_{n(7)})$
$Z_c^{i-1}(x_{j+1},y_{k-1})$
$Z_c^{i-1}(x_{m(8)},y_{n(8)})$
$Z_c^{i-1}(x_{j+1},y_{k})$
$Z_c^{i-1}(x_{m(9)},y_{n(9)})$
$Z_c^{i-1}(x_{j+1},y_{k+1})$

Fig. 2.12.: Numbering of cells for the LAS method in 2D where only the centre parent cell is divided in four rectangular new cells which are indicated with blue.

Multiplying by $Z_c^i(x_{2j}, y_{2k})$ and taking expectations gives:

$$\begin{aligned}
\mathrm{E}[Z_c^i(x_{2j}, y_{2k}) Z_c^{i-1}(x_m, y_n)] &= \frac{1}{4}\mathrm{E}[Z_c^i(x_{2j}, y_{2k}) Z_c^i(x_{2m}, y_{2n})] \ldots \\
\ldots &+ \frac{1}{4}\mathrm{E}[Z_c^i(x_{2j}, y_{2k}) Z_c^i(x_{2m-1}, y_{2n})] \ldots \\
\ldots &+ \frac{1}{4}\mathrm{E}[Z_c^i(x_{2j}, y_{2k}) Z_c^i(x_{2m}, y_{2n-1})] \ldots \\
\ldots &+ \frac{1}{4}\mathrm{E}[Z_c^i(x_{2j}, y_{2k}) Z_c^i(x_{2m-1}, y_{2n-1})] \qquad (2.104)
\end{aligned}$$

Performance LAS method According to *Fenton* [4], in most cases the LAS method is slightly less or as efficient as the FFT method. Especially, if the symmetric covariance structure is amended in the FFT method. With increasing field size and in multiple dimensions the method then even surpass the efficiency of the FFT method. For homogeneous random fields, the calculation of the a and c coefficients only have to be done once for every stage because they are independent of the position of the cell in the field. To deal with the boundary conditions a distinction has to made between interior, corner and side cells. The a and c coefficients will be different for the cells at the boundary of the random field since it is assumed that what happens inside the random field domain is uncorrelated with the values outside the domain.

The values of the random field with the LAS method are automatically averaged locally, which results in a lower variance in the values of the random field. However, if the random field mesh becomes fine enough it is virtually indistinguishable from the limiting continuous random field. The point variance of the random field is approached since the reduction of the variance is negligible small. Thus the method can be used to approximate continuous random fields as well.

2.5 Discretization methods for class 1 generators

After the spatially correlated variables are generated they have to be assigned to the elements or integration points of the finite elements in some way. To do so, different discretization methods are available. The first four discretization methods are the so called point discretization methods and are visualised in 1D and 2D in figure 2.13

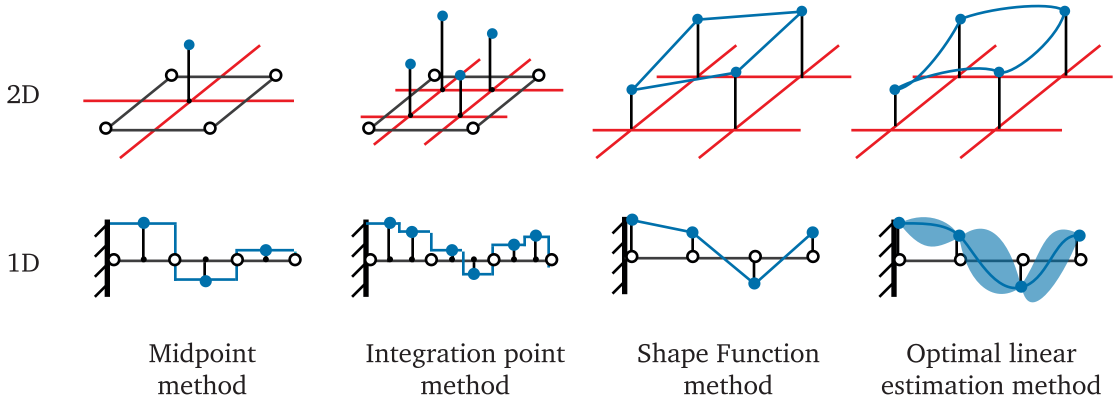


Fig. 2.13.: Visualization of the discretization methods in 1d and 2D.

2.5.1 Midpoint method

In the midpoint method, the nodes of the random field mesh coincide with the midpoints of the finite elements. In this method the random field mesh is not independent of the finite element mesh. After applying the midpoint method, a piecewise constant random field is created which is discontinuous at the finite element boundaries. In the FEM program every element is assigned a spatially correlated value of the discontinuous random field.

The discretized random field $\hat{H}(\mathbf{x})$ can be expressed as:

$$\hat{H}(\mathbf{x}) = z_c(\mathbf{x}_c), \qquad x \in \Omega_e \tag{2.105}$$

Where $\mathbf{x}_c$ is the coordinate of the midpoint of an element and Ω_e is the domain of the element.

If the random field mesh does not coincide with the midpoints of the finite elements, mapping techniques can be used to obtain a value. The value for a certain finite element can then be determined by taking for example the value of the nearest grid point of the random field to the centre of the finite element.

2.5.2 Integration point method

In the integration point method, the grid points of the random field coincide with the integration points of the finite elements. Every integration point is provided with its own value. The random field mesh is more refined than the finite element mesh. The discontinuities are not only localized at the element boundaries but also within the finite element. Again mapping techniques can be used if the grid points of the random field mesh do not coincide with the integration points.

The discretized random field can be expressed as:

$$\hat{H}(\mathbf{x}) = z_c(\mathbf{x}_i), \qquad \mathbf{x} \in \Omega_i \tag{2.106}$$

Where $\mathbf{x}_i$ is the coordinate of an integration point of an element and Ω_i is the domain of that integration point.

2.5.3 Shape function method

In the shape function method, shape functions are used to interpolate between the grid points of the random field mesh. These shape functions may be the same as used for the description of the displacement field, but may also be arbitrarily chosen. This method results in a continuous random field.

The discretized random field can be expressed as:

$$\hat{H}(\mathbf{x}) = \sum_{i=1}^{q} N_i(\mathbf{x}) z_c(\mathbf{x}_i), \qquad \mathbf{x} \in \Omega_e \tag{2.107}$$

Where q is the number of nodes of the random field element e, $\mathbf{x}_i$ is the coordinate of the i^{th} node of that element and N_i are the shape functions.

2.5.4 The Optimal Linear Estimation (OLE) method

In this method the functions to interpolate between the grid points of the random field mesh are such that the variance of the approximation error for each point inside the field is minimized. It is a special case of regression on linear functionals and result in a kind of optimal shape functions. The result is a continuous random field.

The discretized random field can be expressed as:

$$\hat{H}(\mathbf{x}) = a(\mathbf{x}) + \sum_{i=1}^{q} b_i(\mathbf{x}) z_c(\mathbf{x}_i) \tag{2.108}$$

Where q is the number of nodal points involved in the approximation and $a(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ are given by:

$$\begin{aligned} a(\mathbf{x}) &= \mu(x) - \mathbf{b}^T(\mathbf{x})\boldsymbol{\mu} && (2.109) \\ \mathbf{b}(\mathbf{x}) &= \mathbf{R}_{z_c z_c}^{-1} \mathbf{R}_{H(x)z_c} && (2.110) \end{aligned}$$

Where $\mathbf{R}_{z_c z_c}$ is the correlation matrix as defined in 2.70 and $\mathbf{R}_{H(x)z_c}$ is a vector of length q containing the covariances between a node and any location in the field, which are thus functions of the coordinate $\mathbf{x}$. The derivation of equation 2.108 is given in [16].

The optimized shape functions can now be defined as:

$$N_i^{OLE} = (\mathbf{R}_{z_c z_c}^{-1} \mathbf{R}_{H(x)z_c})_i = \sum_{j=1}^{q} (\mathbf{R}_{z_c z_c}^{-1})_{ij} \sigma(\mathbf{x}) \sigma(\mathbf{x}_j) \rho(\mathbf{x}, \mathbf{x}_j) \tag{2.111}$$

2.5.5 Spatial average method

The spatial average method belongs to the second class of discretization methods, namely the *average discretization methods*. The method is based on the idea that in the stochastic finite element method the weighted average, which is normally found by numerical integration of a continuous random field, is already unified in the discretization method. Like measurement of material properties, which are carried out on a certain volume, the values of the random field are also local averages. Based on local average theory in section 2.2.2, the point variance is reduced with the variance function (2.58) and the covariance function between two local averages (2.55) is used.

The discretized random field can be expressed as:

$$\hat{H}(\mathbf{x}) = \frac{\int_{\Omega_e} H(\mathbf{x}) d\Omega_e}{|\Omega_e|} \equiv \bar{z}_c(\mathbf{x}_c), \qquad x \in \Omega_e \tag{2.112}$$

Where $\bar{z}_c(\mathbf{x}_c)$ are spatially correlated random variables according to the local average theory, $\mathbf{x}_c$ is the coordinate of the midpoint of an element and Ω_e is the domain of the element.

Matthies et al. [17] mention that this method is not applicable for non-rectangular elements in the random field mesh. For such domains the covariance function and variance function are hard to be found. However, if the random field mesh and FE mesh do not coincide the weighted average can be taken to determine the values for a certain element or integration point. In this way this problem can be omitted since the random field elements stay rectangular. This approach is visualised in figure 2.14. In both cases the weighted averages can be taken of the cells which are overlapped by the blue surface.

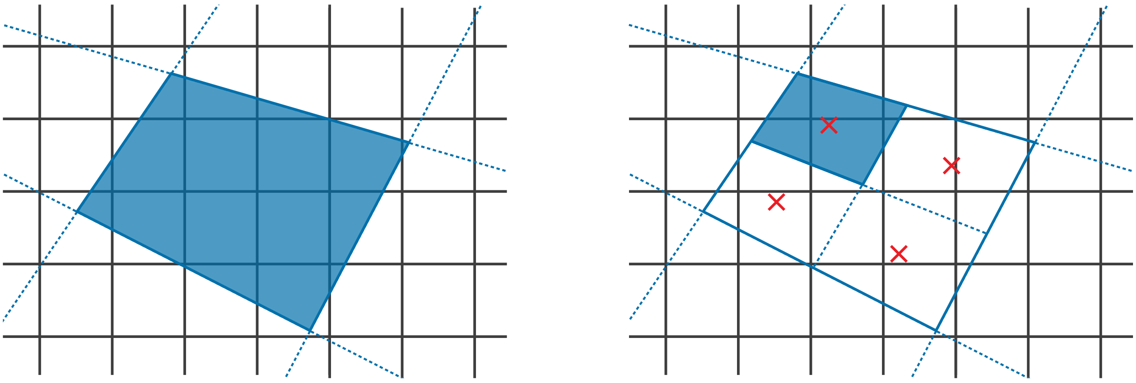


Fig. 2.14.: Weighted average for one element or an integration point.

An advantage of this method is that low order finite elements can be used in the FE model. In a FEM program a continuous random field is numerically integrated. For rough random fields this requires a very fine FE mesh or high order elements.

2.5.6 Comparison discretization methods

In [16] the different methods are compared for a 1D and 2D zero mean and unit variance random fields. The following three different correlation functions are considered in this comparison:

$$\rho(\Delta x) = \exp\left(-\frac{\Delta x}{L_c}\right) \quad \text{Exponential correlation function} \tag{2.113}$$

$$\rho(\Delta x) = \exp\left(-\frac{\Delta x^2}{L_c^2}\right) \quad \text{Squared Exponential correlation function} \tag{2.114}$$

$$\rho(\Delta x) = \frac{\sin\left(\frac{2.2\Delta x}{L_c}\right)}{\frac{2.2\Delta x}{L_c}} \quad \text{Cardinal Sine correlation function} \tag{2.115}$$

In this report only the results for the Exponential (Exp) and Squared Exponential (SExp) correlation function are considered which are both used often in engineering problems. In this comparison the relative error in the variance of the random field is used to quantify the accuracy of the methods as:

$$E(\Omega_e) = \sup_{x\in\Omega_e} \frac{\mathrm{Var}[H(\mathbf{x}) - \hat{H}(\mathbf{x})]}{\mathrm{Var}[H(\mathbf{x})]} \tag{2.116}$$

The results of the comparison are shown in figure 2.15. In this comparison linear shape functions are selected in the SF method. The following remarks can be made with regard to these graphs. For the Exp correlation function the error remains large, even for a small element size. For both correlation functions the SF and OLE method are the most accurate. OLE is even slightly more accurate than SF for the SExp correlation function.

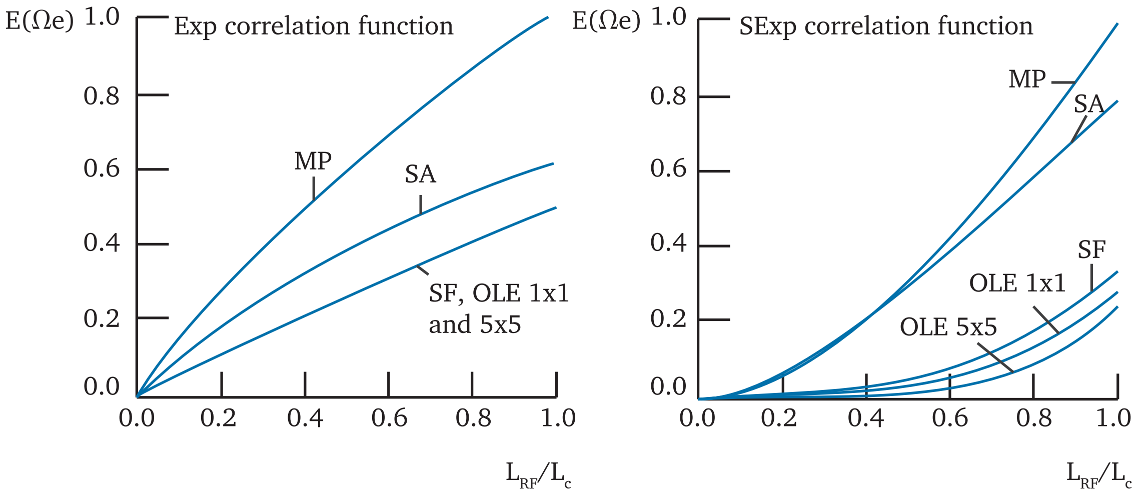


Fig. 2.15.: Comparison of errors for MP, SA, SF and OLE methods for varying element size[16].

2.5.7 Combining generators with a discretization method

In figure 2.6, an overview was given of the different generators of spatially correlated random variables and discretization methods. In this section the combinations which are not possible are enlightened.

The OLE method entails a eigendecomposition of the covariance matrix to find the optimised shape functions. It is therefore convenient to combine the OLE method with the CMD method using eigendecomposition as decomposition method. Other combinations are possible but are less advantageous.

The Spatial average method can be combined with the CMD method quite easily. The covariance matrix is then composed using the covariance function between two local averages. With the LAS method, the random variables are automatically discretized with the spatial average method. For the other methods the SDF function of the covariance function between two local averages has to be found, which is quite hard in general. *Jha and Ching* [18] show an approach to combine the spatial average discretization method with the FFT method.

2.6 Class 2 Generators

The second class of random field generators entail the series expansion methods. The random field is not a combination of discretized sections any more. Instead, the random field is described by a continuous function which holds for the whole domain of the random field. As for other series representation methods, like the Fourier series expansion, such a function is build up by a set of deterministic functions which are each multiplied by a constant. In this case, these constants are random variables $c_i(\theta)$ which all have their own variance based on the contribution of the basis functions f_i to the series expansion. The series expansion for random fields can be expressed as:

$$H(\mathbf{x}, \theta) = \sum_{i=1}^{\infty} c_i(\theta) f_i(\mathbf{x}) \qquad (2.117)$$

The summation of the complete set of basis function is now truncated to find an expression for the random field, which gives an approximation $\hat{H}(\mathbf{x})$ of a continuous random field $H(\mathbf{x})$. This can be seen as the discretization of the random field and leads to the following expression:

$$\hat{H}(\mathbf{x},\theta) = \sum_{i=1}^{N} c_i(\theta) f_i(\mathbf{x}) \tag{2.118}$$

It is very advantageous to sort the constants and basis functions such that the first terms have the largest contribution to the expansion. As remark, it is interesting to see that, in some sense, space and randomness of the random field are separated in these equations [1].

2.6.1 Karhunen-Loève (KL) expansion

A widely used series expansion method is the *Karhunen-Loève (KL)* expansion. The KL expansion entails a spectral decomposition of the covariance function. The discretized expression of the KL expansion is given by:

$$\hat{H}(\mathbf{x},\theta) = \mu(\mathbf{x}) + \sum_{i=1}^{N} \sqrt{\lambda_i}\phi_i(\mathbf{x})\chi_i(\theta) \tag{2.119}$$

Where $\mu(\mathbf{x})$ is the mean and $\chi_i(\theta)$ are uncorrelated N(0,1) distributed random variables. λ_i and $\phi_i(\mathbf{x})$ are the eigenvalues and eigenfunctions of the following eigenvalue problem:

$$\int_\Omega B(\mathbf{x}_1, \mathbf{x}_2)\phi_k(\mathbf{x}_2)dx_2 = \lambda_i \phi_i(\mathbf{x}_1) \tag{2.120}$$

This eigenvalue problem is called the *Fredholm integral equation of the second kind* where the covariance function is used as kernel. A valid covariance function is a bounded, symmetric and positive semi-definite kernel [6]. The eigenfunctions are continuous and orthonormal to each other, i.e.

$$\int_\Omega \phi_i(\mathbf{x})\phi_j(\mathbf{x}) = \delta_{ij} \tag{2.121}$$

Where δ_{ij} is the Kronecker delta. The derivation of equation 2.119 can be found in appendix D. In this derivation it can be seen that the KL expansion is a optimal representation of the random field in the sense that the mean square error is minimized. The KL expansion can be viewed as a decomposition of the random field along the orthogonal random basis vectors with $\phi_i(\mathbf{x})$ being the projections along the random basis vectors.

2.6.2 Numerical methods to solve the KL expansion

The eigenvalue problem in equation 2.120 can be solved analytically for only a few covariance functions and geometries of the random field. In the case an analytical solution can not be found, a numerical method to solve the KL expansion have to be selected.

The following methods approximate the eigenvalues en eigenfunctions to find a solution for the KL expansion:

- *Orthogonal Series Expansion (OSE)* - the basis functions for the series expansion are chosen arbitrarily.
- *Expansion Optimal Linear Estimation (EOLE) method* - eigenvectors and eigenvalues of the covariance matrix are used to find the approximated eigenfunctions and eigenvalues for the KL expansion
- *Nyström method* - the integral (2.120) is approximated by a numerical integration scheme. For a specific selection of the parameters this method is equivalent to the EOLE method [19]
- *Galerking based methods* - entails the FEM and Finite Cell method (FCM) and are practical in problems with complex geometries.

The EOLE method will be explained in the next section. For more detailed information about the other methods reference is made to [19]. In a comparison between the preceding expansion methods, it depends on the used correlation function and geometry of the domain which method is the most accurate [1, 19].

2.6.3 Expansion Linear Optimal Estimation method

In the EOLE method, the eigenfunctions and eigenvalues for the KL expansion are approximated with the eigenvalues and eigenvectors of the covariance matrix. The approximated eigenfunctions $\hat{\phi}_i(x)$ are determined as follows:

$$\hat{\phi}_i(x) = \frac{1}{\hat{\lambda}_i} \sum_{j=1}^{N} Q_{ij} B(x_j, x) \tag{2.122}$$

Where λ_i are the eigenvalues and $\mathbf{Q}$ is a matrix containing the eigenvectors of the covariance matrix. $B(x_i, x)$ is the covariance function with x_i the nodal coordinate of one of the nodes in the random field mesh and x a continuous spatial variable. The KL expansion can then be approximated as follows:

$$\hat{H}(\mathbf{x}, \theta) = \mu(\mathbf{x}) + \sum_{i=1}^{M} \frac{\chi_i}{\sqrt{\hat{\lambda}_i}} \sum_{j=1}^{N} Q_{ij} B(x_j, x) \tag{2.123}$$

Where $\boldsymbol{\chi}$ is a vector containing independent zero mean, unit variance, normally distributed random variables. More detailed information can be found in [19].

2.6.4 Series expansion for Non-Gaussian random fields

In general, the generation of non-Gaussian random fields with the KL expansion is quite hard. The Polynomial Chaos (PC) expansion may be of use for generating non-Gaussian fields. It is another widely used method to represent the random field. It can be shown that the KL-expansion is a special case of polynomial chaos expansion.

If a non-linear transformation, like the Nataf transformation, is possible, KL expansion is possible. However, it can not be confirmed that the transformed field inherits the optimal representation of the random field [19].

2.7 Random field generation for reinforced concrete structures

In this section the generation of random fields, specific for reinforced structures, is considered. First, a short introduction to reinforced concrete is given. Next, the different parameters which have a stochastic character are considered. In the end, a review of literature where random fields are used to model reinforced concrete structures is presented.

2.7.1 Introduction to reinforced concrete

Concrete is a composite material which is build up out of aggregate, cement, water and often some additives. When the cement is in contact with the water the cement starts to harden which binds the aggregates together. If the different components are mixed well and with the right proportions, a material is fabricated which can resist high compressive stresses. The tensile strength is only 10% of the compressive strength. To counteract the low tensile strength of concrete in structures, concrete is combined with reinforcement steel.

During the hardening process the concrete shrinks due to emission of water from the pores and the ongoing hydration process. If the shrinkage is restrained at the boundaries, for example the interface with the sub-base, piles and columns, cracks will occur in the concrete. This mechanism is often neglected in the design phase and can have undesired consequences. Cracks can precipitate degradation of the concrete and are perceived as aesthetically unappealing.

In the failure modes of concrete, as for other materials, a distinction is made between the Ultimate Limit State (ULS) and the Serviceability Limit State (SLS). In the ULS the failure load of the construction is determined using partial safety factors. A distinction can be made in failure due to bending, shear, tension or crushing of the concrete in the compression zone. In the SLS the focus is on the limitation of the crack width in the concrete. No partial safety factors are applied in such calculations.

Due to the inhomogeneity in concrete, the material properties have a high variation. Especially the tensile strength of concrete has a high COV of 0.3. The high variation results in relative high safety factors for concrete material properties. The variation in the material properties of reinforcement steel is very low.

2.7.2 Modelling the variability of reinforced concrete

If the different parameters, which are involved in an analysis of a reinforced concrete structure, are considered, three different categories can be identified. In table 2.1 an overview of the different parameters with a random character is given.

Depending on the material model chosen in the non-linear analysis, some of the parameters are linked to other parameters and can therefore not be chosen independently. Another point of concern is that some material parameters, like the Poisson's ratio, have limited boundaries. The mean value and standard deviation of the material parameters depend on the concrete or steel class. The associated distribution varies for every parameter. Most common distribution types are the normal, truncated normal and log-normal distribution type.

Tab. 2.1.: Examples of parameters in concrete with a uncertain character.

Parameter type	Parameter
Geometry	Dimensions structural element, e.g. thickness and position of reinforcement bar
Material	Young's modulus Poisson ratio Tensile strength Compression Ultimate strain Fracture energy Hardening modulus Angle of internal friction Cohesion coefficient Yield stress steel
Loading	Pre-stressing

In the CEB-FIB 2010 model code, deterministic relations between the different parameters are given. They can be used to determine the mean values of a material parameter, based on another material parameter. The only guideline which gives probabilistic relations between the different parameters is the Probabilistic Model Code of the Joint Committee on Structural Safety (JCSS) [20]. In this code it is advised to take into account the spatial variability for the material properties of concrete only. Other parameters as the cover of the reinforcement or yield stress of the steel should be taken into account in a reliability analysis with one single random variable.

The correlation function used in this code entails a threshold value, which is different from all the other approaches found in literature. It is stated that in one batch of concrete the material properties are correlated with a minimum of 0.5. The used correlation function is a variant of the SExp correlation function and is given by:

$$\rho(\Delta x) = c1 + (1 - c1)e^{-(\frac{\Delta x}{L_c})^2} \tag{2.124}$$

with the correlation length equal to 5 meters and $c1$ equal to 0.5. For two different jobs, the values are uncorrelated according the JCSS model code. The correlation length of 5 m is quite large in comparison to other approaches in literature, which are given in the next section. This correlation function should be used only for structural elements. For smaller elements, where local failure mechanisms are studied, this correlation function is not appropriate. For very large structures the influence of the spatial variability is negligible. The variation should then be taken into account with a random variable for the whole structure or parts of the structure.

2.7.3 Review of random field generation for reinforced concrete structures

In literature several efforts have been made to carry out a non-linear probabilistic analysis for reinforced concrete structures. In this section an overview of the different approaches to take into account the spatial variation is given.

In [21, 22] the reliability of concrete bridges is determined by just simply assigning random variables to several material, load and geometrical properties which are correlated to each other. In this case the spatial distribution of the properties is not taken into account. According to *Milton* and *Zhang* [23, 24], this approach is however only appropriate when the dimensions of the construction are smaller than the correlation length. It is shown that when the dimensions of the construction are larger than the correlation length, the spatial variation has a significant influence on the structural reliability. In [25], *Lee and Mosalam* observed that neglecting the spatial variability of concrete properties can lead to overestimation of the variation in the load-capacity of a concrete structure. An interesting contribution is made by *De Vasconcellos*. In [11] he concludes that the impact on the reliability due to the spatial variation of concrete properties depends on the failure mechanism of the concrete structure. For a concrete structure where ductile rupture of the reinforcement steel was the failure mechanism, it had little influence on the reliability. In that case the variation in the yield strength of the steel had the most impact on the reliability. For a concrete structure where crushing of the concrete was the failure mechanism, the variability in the concrete properties did have an impact on the reliability. In both cases it was shown that the variation in geometrical properties had a small influence on the reliability. In case of the serviceability limit state, where the deflections are governing, the variation in concrete properties did have an impact on the reliability. Also it was noticed that the variation in deflection is larger when the structure is in the cracked stage than in the non-cracked stage.

An overview of the different methods to model the spatial variation of the concrete properties with a random field is given in table 2.2. If these methods are compared, some similarities and differences can be pointed out. First of all, most scholars apply the Covariance Matrix Decomposition (CMD) method. Next, the method is only applied on a small scale structure as for example a column or a beam. The CEB 90 model code is often used to relate other concrete properties to the compressive strength of the concrete. The correlation length varies between 0.5 and 10 m and the coefficient of variation varies between 0.1 and 0.3. The squared exponential covariance function is mostly used without the threshold value proposed in the probabilistic model code of JCSS. In all the articles the reliability is determined with the Monte Carlo Simulation (MCS) method. In most cases, MCS is combined with a sampling technique to reduce the calculation time. *Matthies* [17] uses more advanced techniques, but he only performs linear analyses. No specific information about the computational performance of different methods is given by the authors. Some authors mention that other random field generators have to be applied for larger structures and for models in 3D.

Beside the articles mentioned in table 2.2, where the applied method is described and applied on an (academic) example, there are a couple of articles where software is used to determine the reliability of a concrete structure. In [26–32] use is made of the Non-linear FEM program ATENA which is combined with the probabilistic module FREET. The combination of this software is called SARA. In this program the CMD is used for random field generation and MCS is applied with Latin Hypercube Sampling and Simulated Annealing. Specific information about correlation lengths or the used correlation function is not given. If other random field generators are used is also not mentioned.

Other examples of probabilistic software, which can be used in combination with deterministic FEM software, are COSSAN-X/OpenCOSSAN [33], NESSUS [34] and CalREL/FERUM/OpenSees [35].

Tab. 2.2.: Comparison random field generation methods for reinforced concrete structures.

Ref.	Construction	Scale	Method	Variable	Distribution	COV	CF	$\mathbf{L_c}$	Dim.	Remarks
[11]	Simply supported beam & a column	Small	FFT	E_c f_c f_{ct}	Normal Normal Noraml	0.15 0.15 0.2	SExp SExp SExp	$\sim$ $\sim$ $\sim$	2D 2D 2D	Geometric and steel properties are considered as RV's and L_c is assigned three values (h, $4 \cdot h$ and $10m$) to check its affect on the reliability.
[23]	Eccentric loaded column	Small	CMD+MP	E_c f_c f_{ct}	Normal (trunc.) Normal (trunc.) Normal (trunc.)	0.1 0.1 0.2	SExp SExp SExp	$6 \cdot b$ $6 \cdot b$ $6 \cdot b$	1D 1D 1D	RF-mesh and FE-mesh coincide, correlation length depends on width of column and steel propetries and loads are considered as RV's.
[36]	Blast load on column	Small	CMD+MP	Cover f_c	Normal (trunc.) log-normal	0.12 0.12	SExp SExp	1.0 m 0.5 m	1D 1D	RF-mesh and FE-mesh coincide, correlation lengths are based on experimental data and steel propetries are considered as RV's.
[37]	Seismic action on gravity dam	Large	CMD+MP	f_c	Normal (trunc.)	0.2	SExp	-	2D	RF-mesh and FE-mesh coincide, other concrete properties are obtained via deterministic relations with f_c, and L_c is not mentioned.
[38]	Simply supported bending beam	Small	OSE/EOLE	f_c f_{ct} E_c G_f α	Log-normal Log-normal Log-normal Log-normal Log-normal	0.100 0.211 0.105 0.341 0.050	Exp Exp Exp Exp Exp	0.6 0.6 0.6 0.6 0.6	1D 1D 1D 1D 1D	The RF's of f_c, f_{ct} and E_c are correlated, geometry properties and the load are taken as RV's, FE-mesh is in 3D, values of RF's are assigned to MP of FE.
				E_c	Log-normal	0.1	Exp	-	2D	FORM is applied in stead of MCS.
[39]	Three point bending beam	Small	CMD+IP	f_{ct} E_c G_f	Log-normal log-normal Log-normal	0.2 0.2 0.2	Exp Exp Exp	0.6 0.6 0.6	2D 2D 2D	Eigendecomposition is applied, only the eigenvectors belonging to the largest eigenvalues are taken into account. In the second case, two values for COV and L_c are applied and compared with an experiment.[1]
	Simply supported beam	Small	CMD+IP	f_{ct} E_c G_f	Log-normal Log-normal Log-normal	0.3/0.25 0.3/0.25 0.3/0.25	Exp Exp Exp	0.5/2.1 0.5/2.1 0.5/2.1	2D 2D 2D	

[1] In [40] this approach is improved with regard to computational efficiency by applying Artificial Neural Networks

2.8 FEM and methods for probabilistic FEM

In this section some of the methods where a probabilistic design method is combined with the Finite Element Method (FEM) are briefly considered. They belong to a collection of methods which are denoted by the collective term, the Stochastic Finite Element Method (SFEM). First a short introduction to FEM is given to refresh the memory.

The FEM is a numerical method to find approximate solutions to boundary value problems for elliptic partial differential equations (PDEs). These PDEs describe the behaviour of physical phenomena. Such a PDE can be written in its 'weak form' which can be found with variational methods. This has the effect of reducing the order of the derivatives appearing in the equation, and leads to a from which is convenient for a numerical solution. The domain of the problem is subdivided into smaller parts which are called finite elements. Shape functions are then used along which the approximated solution can be projected. For solid mechanics problems the displacements are usually the fundamental unknowns which are approximated by minimizing the error in the displacement field. This approach is a called the *Galerkin method* which gives a solution which is optimal in terms of the strain energy. Based on the solution for the displacement field other properties as the strains, stresses and internal forces can be determined. To solve the displacement field a system of equations has to be solved which can be expressed as:

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{2.125}$$

Where matrix $\mathbf{K}$ is commonly known as the stiffness matrix, vector $\mathbf{f}$ as the right hand side (RHS) vector and vector $\mathbf{u}$ as the displacement vector, which can be found by solving the linear system of equations. This entails the inversion of the stiffness matrix which is quite an expansive operation. Different solvers are available to do so. For non-linear problems the true failure path can be found with iterative solvers. The stiffness matrix changes during such an analysis which makes a non-linear analysis quite expansive in terms of computation time.

2.8.1 Monte Carlo Simulation (MCS)

The Monte Carlo Simulation (MCS) method is the simplest method to perform a SFEM analysis. This method involves the repeated simulation of a stochastic process to determine the probability of a certain outcome. The probability of occurrence can be determined by observing the number of times the simulated outcome is equal to the considered outcome. By the law of large numbers, the outcome will converge to the expected value. When FEM is combined with MCS, the reliability of a structure can be determined by the following procedure. As input for the MCS a number of random variables or random fields are generated. With every random variable or random field a FEM analysis is carried out. A selection is made in the output, for example the deflection of the tip of a cantilever beam, and statistical operations are carried out to determine the characteristics of the distribution of the selected variable. If enough runs are carried out, the MCS is the most accurate method. The MCS procedure is visualised in figure 2.16.

The problem with MCS is that the computation time for computation intensive problems becomes very large since the stiffness matrix for the FEM analysis changes for every run. Sampling methods, where a different sampling distribution than the original distribution is chosen, can be used to reduce the number of needed runs to acquire a certain accuracy. During the simulation a large fraction of realisations will be obtained in the failure domain [40]. Examples of sampling methods

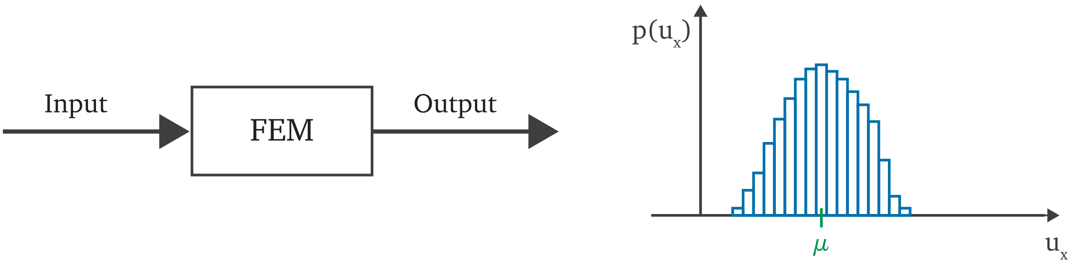
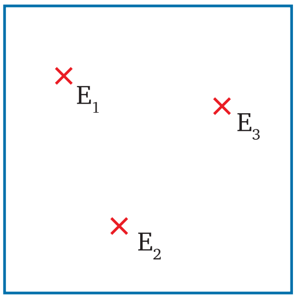


Fig. 2.16.: MCS procedure visualised where the FEM program functions as a black box. On the RHS an example of the analysis of the response of the system.

are Latin Hypercube Sampling, importance sampling and the response surface method. When a sampling method is applied it is often referred to as improved Monte Carlo simulations. Another improvement can be made by taking a 'mean' stiffness matrix and correct for the deviation in the load vector [37]. Figure 2.17 illustrates the difference between the use of a random variable and a random field in the Monte Carlo simulation. When random variables are used the property has a constant value in the model while random fields result in variability within the model. All the random field generation methods mentioned earlier can be applied in the MCS method.



Random variable

Run 1: E_1=E_2=E_3=a
Run 2: E_1=E_2=E_3=b
...
Run n: E_1=E_2=E_3=c

With a, b and c as random variables

Random field

Run 1: E_1=a_1, E_2=a_2, E_3=a_3
Run 1: E_1=b_1, E_2=b_2, E_3=b_3
...
Run 1: E_1=c_1, E_2=c_2, E_3=c_3

With **a**, **b** and **c** being vectors containing correlated random variables

Fig. 2.17.: Comparison between the use of random variables and random fields in MCS.

The MCS method is statistically consistent, it is therefore often used as a reference solution to test other methods to determine the reliability.

2.8.2 Other probabilistic FEM methods

For computational intensive problems it is advantageous to use a more efficient method which can determine the reliability of a structure. In this section some of the methods are briefly described.

Perturbation method In the perturbation method a Taylor series expansion of the stochastic finite element matrix, the loading vector and the displacement vector is made around the mean of the random variables involved. A statistical evaluation can be made of the displacement vector which results in an expression for the approximation of statistical properties as the mean and covariance [41]. The accuracy of the approximation increases when a higher order of the Taylor expansion is used. The increase in the accuracy is however small compared to the increase in computational time. The method works quite well for problems with a small COV (<0.2). This can be explained by the fact Taylor expansion is made around the mean. The tail of the response is therefore hard to approximate [2]. In literature some examples can be found where the perturbation method is applied. In [1] it is combined with random fields generated with a class 1 generator and discretized as piecewise constant random fields. In [42] an example can be found where they are discretized with the shape function method.

Finite Element Reliability Method (FERM) The method couples the classical reliability methods, like FORM and SORM, with the FEM. The probabilities of failure of the different components of a system are determined by approximating the limit state function belonging to the failure of that component. The needed information like the design point and the gradient can be determined by an algorithm combined with FEM. In literature some examples can be found where the reliability method is applied. In [43] it is combined with random fields generated with a class 1 generator and discretized as piecewise constant random fields. In [44] an example can be found where it is combined with a series expansion method.

Spectral Stochastic Finite Element Method (SSFEM) In this method the random field is expressed using the KL or PC expansion. This expression is used in the definition of the stiffness matrix. For example, $E(\mathbf{x}, \theta)$ can be the expanded random field for the Young's modulus in the stiffness matrix. The solution of the system, which are the displacements, can now be expanded using one of the expansion methods. The displacements can be expressed as a function of the set of used random variables. Sudret [1] states that SSFEM is limited to linear problems only. Material non-linearity or geometrical non-linearity cannot be dealt with by SSFEM in its latest state of development. The SSFEM is always combined with class 2 generator methods.

2.8.3 SFEM for non-linear FEM

The FE analyses of concrete structures are almost always non-linear. It is therefore convenient to consider which of the SFEM methods can be used in a non-linear analysis. In literature only examples of the MCS method combined with a non-linear analysis can be found. The other methods become to involved to determine the reliability of the structure. No examples where those methods are combined with a non-linear analysis can be found in literature till now. Although the MCS method is not very efficient, it seems the only possible method for non-linear analyses. For improvements in computation time a proper sampling technique should be selected to reduce the number of samples in the MCS analysis.

2.9 Structure of general purpose FEM program

In this section the work flow of a general purpose FEM program will be considered, where use is made of reference [45]. Thereafter it is indicated where random fields could possibly be implemented in the work flow of such a program.

2.9.1 Work flow FEM program

The structure of a FEM program can be divided into three steps, namely: preprocessing, analysis and post-processing.

Preprocessing The first step in the FE analysis is known as *preprocessing*. This step involves generating a FE mesh. For larger models this it is convenient to use a mesh generation program. For a FE mesh the element type, the coordinates of the nodes, the boundary conditions and element connectivity have to be specified. For every element the material data, like the Young's modulus, the Poissons's ratio and the density of the material, have to be specified. All the data of this step is stored and can be used as input for the FE analysis.

Analysis This phase starts with reading the input data from the preprocessing stage. Next, the element stiffness matrix $\mathbf{K}_e$ is formed for every element and added into the global stiffness matrix $\mathbf{K}$. The element RHS vector $\mathbf{f}_e$ of each element is formed. If a Dirichlet (displacement) boundary conditions is applied on a degree of freedom the corresponding element in the element RHS is modified. Thereafter all the element RHS vectors are added into the global RHS vector $\mathbf{f}$. Now the Neumann (loading) boundary condition are applied by adding them into the RHS vector. Thereafter, the main part of this stage can be carried out in which the system of equations, given in equation 2.125, is solved. The best method to solve this system of equation depends on the nature of the problem and its size.

Post processing The solution to the system of equations is now processed to obtain other interesting quantities as the displacement field, strains, stresses and reaction forces. For static problems the internal force vector should be zero where no Dirichlet boundary conditions are applied. This means that the body is in equilibrium.

2.9.2 Non-linear FE analysis

The work flow has to be modified when a non-linear analysis is carried out. A FEA involves non-linearities when for example the material behaviour is described by non-linear relations (e.g. crushing and cracking). Large displacements can also introduce non-linearities in the FE analysis. In that case the configuration is changed such that equilibrium equations must be rewritten with respect to the deformed structural geometry. Also the load directions and magnitudes may change.

To find a solution which is in equilibrium the loads or displacements are increased with small increments. In every step equilibrium is sought for by an iterative algorithm. In figure 2.18 an example of such an algorithm is given in case the load is increased. The unbalance can be determined in terms of displacements, force or energy. If the convergence criteria is met the next load step is executed.

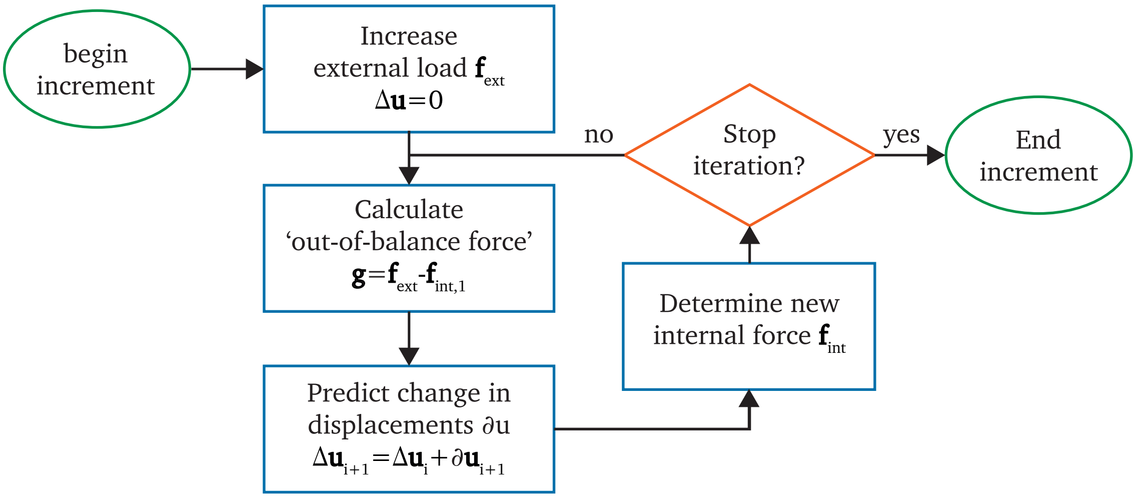

begin increment
Increase external load $\mathbf{f}_{ext}$ $\Delta\mathbf{u}=0$
no
Stop iteration?
yes
End increment
Calculate 'out-of-balance force' $\mathbf{g}=\mathbf{f}_{ext}-\mathbf{f}_{int,1}$
Determine new internal force $\mathbf{f}_{int}$
Predict change in displacements ∂u $\Delta\mathbf{u}_{i+1}=\Delta\mathbf{u}_i+\partial\mathbf{u}_{i+1}$

Fig. 2.18.: Solution procedure Incremental-Iterative methods [46].

2.9.3 Implementation of Random fields in FEM

In the work flow of a general purpose FEM program the material data can be specified in the preprocessing stage. The random fields for the different parameters should be generated before the analysis phase. The values in the random field can be assigned to the model on element level or integration point level. These values have to be stored on the database during the whole analysis. In the different steps during the analysis, for example when the stiffness matrix is assembled, the values per element or integration point can be retrieved from the database. An example of a linear FEM code of a bar in tension, where a random field for the Young's modulus can be defined, can be found in appendix E.

3 Assessment of random field generators

In this chapter the performance of the different methods to generate a random field are considered. To do so, some of the methods mentioned in chapter 2 are modelled to asses their performance. In the first section the set-up for the comparison is described. In the next section information is given on how the selected methods are implemented. Thereafter the results of the comparison will be given. The conclusions of the comparison will be drawn in the last section of this chapter. After this assessment the performance of the different methods will be more clear. This will give a basis for the selection of the most accurate and efficient random field generator for the modelling of concrete. Besides that the influence of the different parameters on the accuracy and efficiency will be described.

3.1 Set-up for comparison of random field generators

In this section the set-up for the comparison of the random field generators will be considered. Based on findings in literature and on the current structure of DIANA, some of the methods to generate a random field will be selected. Next, the parameters which are considered in the comparison are given. In the last part of this section the criteria for the comparison to asses the methods on accuracy and efficiency are given.

3.1.1 Selection of methods for comparison

There are different methods to generate a random field. An overview of the methods to generate a random field is given in figure 2.6 and the description of the different methods is given in section 2.4 till section 2.6. First the structure of the DIANA program will be considered to asses the boundary conditions of the program for the generation of a random field.

Structure of DIANA program for random fields In the preprocessing part of the DIANA program a function exists which can specify the spatial variation of properties in a FE model. This is used in problems where a priori the relation with a spatial coordinate is known. An example is the increase of the stiffness of soil with increasing depth. For this function in DIANA an orthogonal grid is defined. This can be done in 1D, 2D and 3D. Every point in the grid can be assigned a value. In the FEM analysis the material properties for every integration point can be determined by interpolating between the grid points.

This description matches perfectly with the approach where a class 1 generator is used in combination with the Shape Function (SF) method. It is therefore very advantageous to select a class 1 generator which is discretized with the SF method. The structure of DIANA can of course be changed, but this effort has to result in an increase in performance when another method is chosen.

Selection of methods First, a selection is made in the class 1 methods. Based on the findings in literature, which are reported in section 2.4, the CMD, FFT and LAS method are selected as most promising generators of spatially correlated random variables. The CMD method is easy to implement and straight forward, which makes it an attractive option. It is stated that in higher dimensions and for a large number of nodes this method becomes slow and prone to round of errors. It is therefore of interest to check to what extend this method is workable in a FEA. The FFT method and LAS method promise to be very efficient in higher dimensions and for a large number of nodes. The methods, however, are harder to implement and may be more complex for the user to understand. In terms of accuracy it is not clear which one of these methods will perform the best. The MA and DFT method where not selected because they are less efficient than the FFT and LAS method and also hard to implement. The TBM method is not selected because it is not clear a priori how many lines have to be selected to acquire accurate results and the derivation of the correlation function for the 1D process is difficult.

As discretization method the SF method will be selected. In figure 2.15 it is shown that only the OLE method is slightly more accurate when the *Squared Exponential correlation function* is used. The performance of other discretization methods than the SF method will not be assessed in this report.

As class 2 method the EOLE method is selected to generate random fields. This method is straight forward to implement and efficient in obtaining a random field approximation. It is however less efficient in the evaluation of a realization of the field than other methods [19]. No comparison is made in literature yet between class 1 and class 2 methods to asses them on efficiency and accuracy. In advance, it can therefore not be stated if the application of a class 2 methods outweighs the efforts of changing the structure of the DIANA program for the implementation of a random field. It is therefore very interesting to explore the differences between class 1 and 2 methods.

3.1.2 Selection of parameters for comparison

In this section the different parameters, which may have a fixed or varying value, will be described. It is chosen to compare the methods in 1D and 2D only. In most cases one or two of the dimensions are smaller than the correlation length in concrete structures. In such a case the number of dimensions for the random field can be smaller than the number of dimensions in which the FE model is modelled. A slender concrete beam can for example be modelled with a 2D FE model and with a 1D random field. In 1D 2000 fields and in 2D 200 fields will be generated for the comparison.

The values for the statistical characteristics of concrete are determined based on the review of literature in chapter 2. In this review, the values used for the correlation length varies from 0.5 till 5 m in most cases. In a single case it was equal to 10 m, but this was not assumed to be realistic. Therefore 0.5 m, 2.5 m and 5 m are selected as correlation lengths.

For the determination of the domain size and number of nodes the following considerations are taken into account:

- For the exponential correlation function the element size of the RF-mesh L_{RF} should be taken between $L_c/10$ and $L_c/5$
- For other correlation functions the element size needs to be taken between $L_c/4$ and $L_c/2$.
- For some of the methods it is required that the number of nodes needs to be a power of 2.

The domain size is set equal to 17.5 m in 1D and 17.5 m x 17.5 m in 2D. For a correlation length of 0.5 m the number of nodes is chosen equal to 64, 128 and 256 in each direction. For a correlation length of 2.5 m the number of nodes is chosen equal to 16, 32 and 64 in each direction. For a correlation length of 5 m the number of nodes is chosen equal to 8, 16 and 32 nodes in each direction. An overview of the different correlation lengths, number of nodes, element size of the RF-mesh and the ratio between the correlation length and the element size of the RF-mesh is given in table 3.1.

Tab. 3.1.: Number of nodes and corresponding element size of random field mesh for the different correlation lengths.

Correlation length	Number of nodes	Element size RF-Mesh	L_c/L_{RF}
5 m	8	2.5 m	2
	16	1.17 m	4.3
	32	0.5645 m	8.9
2.5 m	16	1.17 m	2.1
	32	0.5645 m	4.4
	64	0.2778 m	9
0.5 m	64	0.2778 m	1.8
	128	0.1378 m	3.6
	256	0.0686 m	7.3

With all the methods standard normal random fields are generated which are thereafter transformed to the desired distribution having a certain value for the mean and standard deviation. In the review of random field generation for concrete structures both the normal and log-normal distribution type are used. It is therefore chosen to consider both distribution types in the comparison. When a normal distribution is chosen the mean is set equal to 0 and the standard deviation is set equal to 1, which corresponds to a standard normal distributed random field. With the Nataf transformation, which is described in section 2.3.1, a random field with a log-normal distribution is acquired. It is expected that if the standard normal random field can be generated accurately the log-normal random field will also be accurate. This is checked by generating log-normal random fields with both mean and standard deviation equal to 1. A COV of 1 is higher than the COV for concrete material properties which varies from 0.1 - 0.3. For a COV of 1, the transformation to a log-normally distributed random field is stronger than for a lower COV. The differences between the use of a normal and log-normal distribution type will be more clear in this way.

In literature both the the *Exponential* (Exp) and *Squared Exponential* (SExp) correlation function are used often for engineering practices. For concrete structures the SExp correlation function is used more often. In some rare cases the Exp correlation function is used. This type of correlation function is more suitable for the modelling of the spatial variability of soil properties.

In a comparative study between generators of random fields [47] the Exp correlation function is used. It is then assumed that for other correlation functions the same holds for the performance of the different generators. However, in [1] it is shown that the accuracy of a generator does depend on the used correlation function. It is therefore chosen to select both correlation functions for the comparison to check the differences. Furthermore a threshold value (c_1) for the correlation function of 0.5 is suggested by the Joint Commission of Structural Safety (JCSS). In the comparative studies, mentioned above, this threshold value is not considered, i.e. $c_1 = 0$. In the comparative study of this research a value of 0 and 0.5 is used. The correlation functions, used in the assessment, are given below.

$$\text{(Exp)} \qquad \rho(\Delta x) = c_1 + (1 - c_1)\text{exp}\left(-\frac{\Delta x}{L_c}\right) \tag{3.1}$$

$$\text{(SExp)} \qquad \rho(\Delta x) = c_1 + (1 - c_1)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right) \tag{3.2}$$

where $\Delta x = |x_1 - x_2|$.

3.1.3 Selection of criteria for comparison

All the methods will be assessed on efficiency and accuracy. With respect to efficiency a distinction is made between the *initialization time* and the *realization time*. For the generation of a random field, first, some quantities have to be determined which can then be used in the generation of every single field. For example in the CMD method a correlation matrix has to be assembled and decomposed before a random field can be generated. This decomposed matrix can then be used for the generation of every random field. The time to determine the quantities which have to be determined once is denoted with the initialization time. The time to generate 2000 random fields in 1D and 200 random fields in 2D with this data is denoted with the realization time. Both the initialization and realization time are measured in CPU-time. The method which takes the least time will, logically, perform the best with respect to efficiency.

With respect to accuracy a distinction is made in the representation of the mean, standard deviation and correlation structure of the field. For every field the mean value and the standard deviation is determined. Of all the mean values of the different fields the mean value, μ_m, and the standard deviation in the mean values, μ_s, is determined. For the standard deviation the same is done and is denoted with σ_m and σ_s. For the mean values, μ_m and σ_m, the absolute error with the target value is determined. A smaller error indicates that the method is accurate. The deviations μ_s and σ_s have to be as small as possible for an accurate method.

The correlation structure will be estimated over all the generated random fields. For all the lag distances equal to a multiple of the element size of the random field mesh the correlation coefficient is determined. This value is compared with the exact value, which is determined with the exact correlation function. The mean value of the absolute differences is called the mean correlation error, $C_{err,m}$. The deviation in these values is called the deviation in the correlation error, $C_{err,s}$. A low value for $C_{err,m}$ and $C_{err,s}$ will indicate that the method is accurate. A higher value for $C_{err,s}$ indicates that on certain locations the correlation deviates strongly from desired correlation structure.

The Matlab code in which the statistical properties of the random fields are determined to asses the accuracy of the method can be found in appendix F

3.2 Implementation of random field generator methods

In this section, information is given on how the different methods are implemented and some implementation issues are considered. For the implementation of the different methods, the variance function and one sided SDF have to be known. These will be derived in the next subsection. Also the characteristics of the used correlation functions are studied to give a better understanding of the results of the comparison.

3.2.1 Characteristics of the used correlation functions

In this study the Exponential (Exp) and Squared Exponential (SExp) correlation function are used. These functions are given in equation 3.1 and equation 3.2. In this section the characteristics of these functions will be considered

First, the differences between the Exp and SExp correlation function are considered. To do so, two fields which are generated with the different correlation functions are examined. In figure 3.1 two realizations of standard normal random fields, having a domain size of 17.5 m x 17.5 m and a correlation length of 5 m, are shown. At LHS the field which is generated with the Exp correlation function is shown and at the RHS the field which is generated with the SExp correlation function is shown. The field, generated with the Exp correlation function, shows a very rough pattern in comparison with the random field which is generated with the SExp correlation function. This can be explained by examining the graph at the LHS in figure 3.2 where both correlation functions are plotted. It can be observed that the correlation decreases faster for the Exp correlation function with increasing lag distance than for the SExp correlation function. This can also be concluded when the formulas of both correlation functions are examined. For the SExp correlation function the values in the random field are stronger correlated when the lag distance is smaller than the correlation length and less strong correlated when the lag distance is larger than the correlation length. This results in a very smooth field as can be observed in figure 3.1.

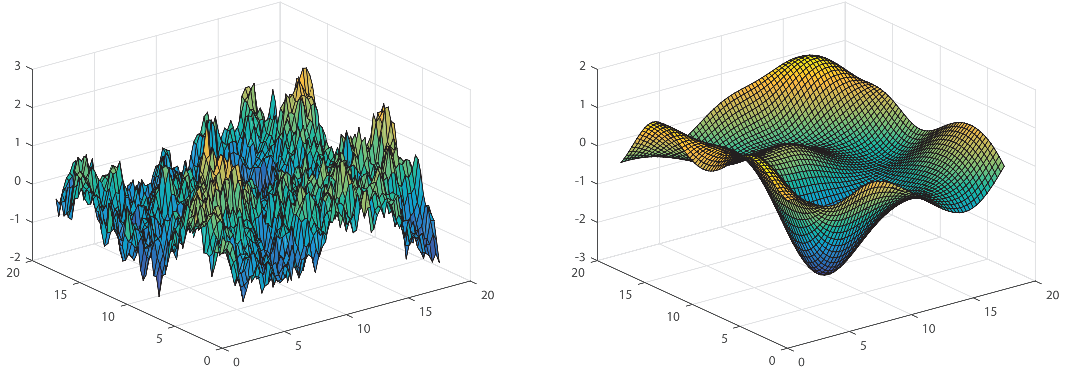


Fig. 3.1.: Two standard normal random fields with a correlation length of 5 m. Left: Exponential correlation function, right: Squared Exponential correlation function.

The influence of the correlation length (L_c) can be clarified by observing figure 3.2. At the LHS the correlation functions are plotted with a correlation length of 5 m and at the RHS with a correlation length of 1 m. The graph is horizontally scaled to the y-axis when the correlation length decreases. The values are correlated less strong when the correlation length decreases which will lead to a rougher surface of the field.

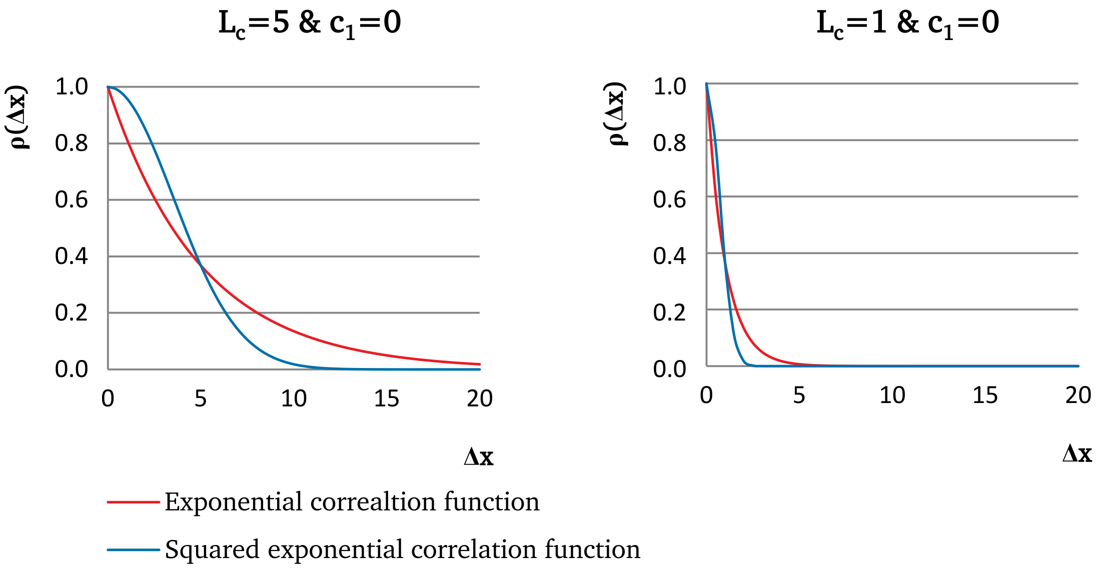


Fig. 3.2.: Effect of the correlation length on the shape of the correlation function.

The influence of the threshold value (c_1) for the minimum correlation in the random field can be clarified by observing figure 3.3. At the LHS the correlation functions are plotted with a threshold value of zero and at the RHS with a value of 0.5. It can be seen that the correlation decreases less slow with increases lag distance. Also it can be observed that when the lag distance increases the correlation will be equal to the threshold value. If the threshold value increases, the field will be more smooth since values are stronger correlated with each other.

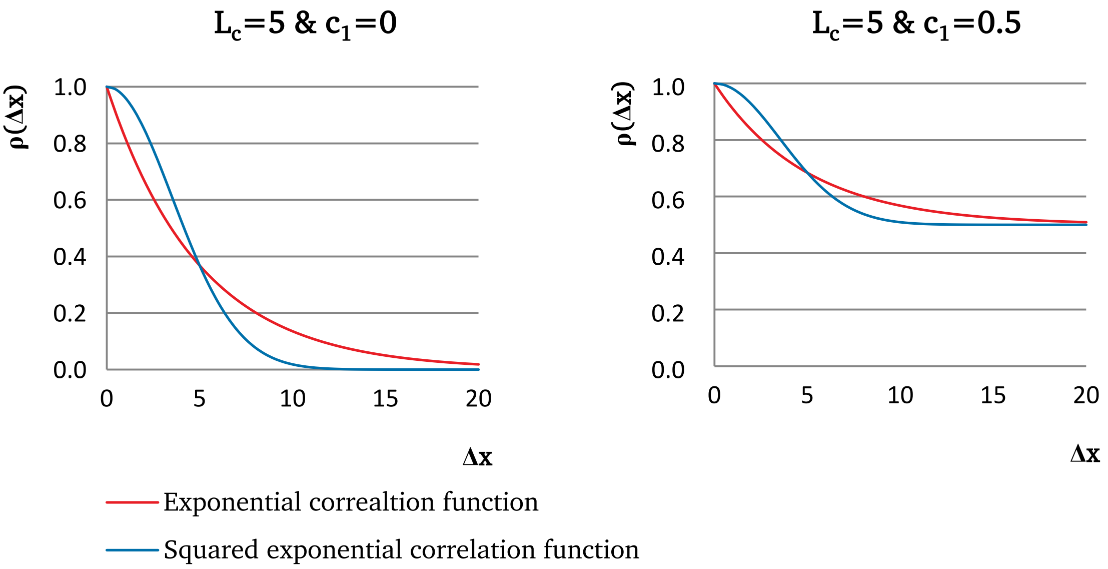


Fig. 3.3.: Effect of threshold value on the shape of the correlation function.

Now the one sided spectral density functions and variance functions will be given for the 1D and 2D case. These will be used in the algorithms of the different random field generators which will be explained in the next sections.

In 1D the one sided SDF and the variance function for both correlation functions are given by:

$$\text{(Exp)} \qquad G(\omega) = 2\sigma^2 c_1 \delta(\omega) + \frac{2(1-c_1)\sigma^2 L_c}{\pi(L_c^2\omega^2+1)} \tag{3.3}$$

$$\text{(SExp)} \qquad G(\omega) = 2\sigma^2 c_1 \delta(\omega) + \frac{(1-c_1)\sigma^2 L_c}{\sqrt{\pi}} \exp\left(-\frac{\omega^2 L_c^2}{4}\right) \tag{3.4}$$

$$\text{(Exp)} \qquad \gamma(D) = c_1 + \frac{2(1-c_1)}{D^2}\left(DL_c + L_c^2 \exp(-\frac{D}{L_c}) - L_c^2\right) \tag{3.5}$$

$$\text{(SExp)} \qquad \gamma(D) = c_1 + \frac{(1-c_1)}{D^2}\left(L_c^2(\exp(-\frac{D^2}{L_c^2}) - 1) + D\sqrt{\pi}L_c erf(\frac{D}{L_c})\right) \tag{3.6}$$

In those functions the $\delta(\omega)$ is the Dirac delta function and $erf(D/L_c)$ is the error function.

In 2D both correlation functions are as follows:

$$\text{(Exp)} \qquad \rho(\Delta x_1, \Delta x_2) = c_1 + (1-c_1)\exp\left(-\sqrt{(\frac{\Delta x_1}{L_{c,1}})^2 + (\frac{\Delta x_2}{L_{c,2}})^2}\right) \tag{3.7}$$

$$\text{(SExp)} \qquad \rho(\Delta x_1, \Delta x_2) = c_1 + (1-c_1)\exp\left[-\left((\frac{\Delta x_1}{L_{c,1}})^2 + (\frac{\Delta x_2}{L_{c,2}})^2\right)\right] \tag{3.8}$$

The one sided spectral density functions in 2D are given by:

$$\text{(Exp)} \qquad G(\omega_1, \omega_2) = 4\sigma^2 c_1 \delta(\omega_1)\delta(\omega_2) + \frac{2\sigma^2(1-c_1)L_{c,1}L_{c,2}}{\pi(1+\omega_1^2 L_{c,1}^2 + \omega_2^2 L_{c,2}^2)^{\frac{3}{2}}} \tag{3.9}$$

$$\text{(SExp)} \qquad G(\omega_1, \omega_2) = 4\sigma^2 c_1 \delta(\omega_1)\delta(\omega_2) + \frac{\sigma^2(1-c_1)L_{c,1}L_{c,2}}{\pi} \exp\left[-\left(\frac{\omega_1^2 L_{c,1}^2}{4} + \frac{\omega_2^2 L_{c,2}^2}{4}\right)\right] \tag{3.10}$$

The variance functions in 2D are not derived exact. The derivations of the given one sided SDF's and variance functions can be found in appendix G.

Covariance of two local avarages The different methods to generate spatially correlated random variables can be combined with the SA discretization method. For the LAS method it is even automatically combined with this discretization method. The reduced variance can be determined with the variance functions which are given in this section. To determine the covariance of two local averages equation 2.61 is used where again the variance function has to be applied. The covariance of two local averages can also be computed using gauss quadrature. In the case the variance function is not known, which is the case in 2D, the covariance has to be determined applying gauss quadrature. The Matlab Code to determine the covariance of two local averages using gauss quadrature can be found in appendix H for 1D and 2D.

The implementation of the different methods and some implementation issues will now be considered.

3.2.2 Covariance Matrix Decomposition method

The theory of the CMD method is explained in section 2.4.1. The Matlab code for the CMD method can be found in appendix I.1 for 1D and in appendix I.2 for 2D.

The first step in the algorithm is the assembly of the *correlation matrix*. This matrix has size $N_x \cdot N_y \cdot N_z$. So in case a random field is generated in multiple dimensions the size of the matrix increases strongly when the number of nodes increases. Thereafter the correlation matrix is decomposed with one of the selected decomposition methods. In the last step, the decomposed matrix is multiplied with a vector, containing normally distributed random variables, having a zero mean and a standard deviation equal to the desired standard deviation of the field. If the CMD method is combined with the SA discretization method, the correlation matrix is filled with covariances according to the local average theory, which are determined with gauss quadrature.

Decomposition issues For the decomposition of the correlation matrix, three methods can be selected to do so. These are the Cholesky decomposition, modified Cholesky decomposition and modified eigendecomposition. The modified decomposition methods are created to avoid numerical problems which were encountered during the comparison.

It was observed that when the SExp correlation function is selected and the number of nodes is relatively high in comparison with the correlation length, the correlation matrix becomes non-positive definite, i.e. not all eigenvalues of the matrix are positive any more. In such a case Cholesky decomposition is not possible any more since this method requires positive eigenvalues.

The eigendecomposition results in an imaginary field since the square root of the eigenvalues is taken to form the decomposed matrix. In the case that some of the eigenvalues are negative, it will result in complex numbers. In theory the correlation matrix is positively definite. Therefore, such a problem should arise from numerical issues. In the previous section, it was shown that the derivative of the SExp correlation function is zero at its origin. For nodes which are close to each other, the correlation is therefore very close to one. The numbers on the diagonal of the correlation matrix are therefore approximately the same as the numbers on first off-diagonal. This results in a correlation matrix which is (numerically) linear dependent. The matrix becomes nearly singular and has eigenvalues which are slightly less then zero.

In the Cholesky decomposition algorithm, the diagonal terms are composed by taking the square root of the corresponding diagonal term in the correlation matrix, which is subtracted by the squared values of the corresponding row of the decomposed matrix like:

$$L_{kk} = \sqrt{R_{kk} - \sum_{j=1}^{k-1} L_{kj}^2} \tag{3.11}$$

where k is equal to the row number of the computed diagonal element.

When the matrix becomes non-positive definite, the argument of the square root becomes negative. In the modified Cholesky decomposition algorithm, the argument is set equal to zero when this value is smaller then a tolerance value, which can be chosen slightly larger than zero, to acquire stable results. In this way, the algorithm can find an 'approximated' Cholesky decomposition. The Matlab code with the modified Cholesky decomposition can be found in appendix J.

In the modified eigendecomposition algorithm, the absolute value of the eigenvalues is taken to form the decomposed matrix like:

```
L=vec*sqrt(abs(lambda)
```

Where *vec* is the matrix containing the eigenvectors and *lambda* is a matrix containing the eigenvalues on the diagonal of the matrix.

Nataf transformation To generate random fields having a log-normal distribution, the Nataf transformation is applied. Before the decomposition is carried out the standard deviation, the mean and the correlation matrix are modified in the code as follows:

```
std_norm=sqrt(log(1+(std_RF/mean_RF)^2));
mean_norm=log(mean_RF)-0.5*std_norm^2;
Cor_M=(exp(Cor_M*std_norm^2)-1)/(exp(std_norm^2)-1);
```

The modified correlation matrix is then decomposed. With the decomposed correlation matrix and the modified standard deviation and modified mean, the random fields are computed as follows:

```
RF(:,j)=exp(mean_norm+L*RV(:,j)*std_norm);    % Generate random field
```

Where j corresponds to the j^{th} field of the total number of fields which are generated.

3.2.3 Fast Fourier Transform method

The theory of the FFT method is explained in section 2.4.4. The Matlab code for the FFT method can be found in appendix I.3 for 1D and in appendix I.4 for 2D.

The algorithm starts with the discretization of the spatial and spectral domain as specified in the theory section. The one sided SDF is then evaluated on the spectral coordinates. The one sided SDF's, which where given in section 3.2.1, are used without the part with the Dirac delta function. This would otherwise result in an infinite value for $G(\omega = 0)$, which results in an infinite value for the variance of the A_1 coefficient, which is not correct. For the right contribution of this part to the variance of the A_1 coefficient, it is important to keep in mind how the coefficients are determined. As stated in the theory section, they can be found by following the inverse Fourier relationships. In [4], it is stated that instead of taking the evaluated values of the one sided SDF on the spectral coordinates for the determination of the variances of the coefficients it is even more precise to use the following expression:

$$G(\omega_k) = \int_{\omega_k - \frac{1}{2}\Delta\omega}^{\omega_k + \frac{1}{2}\Delta\omega} G(\omega) d\omega \tag{3.12}$$

Which is noting else but the area under the one sided SDF. For the part with the dirac delta function, this can be evaluated in 1D as follows:

$$G(\omega_k) = \int_{\omega_k - \frac{1}{2}\Delta\omega}^{\omega_k + \frac{1}{2}\Delta\omega} 2\sigma^2 c_1 \delta(\omega) d\omega = 2\sigma^2 c_1 \tag{3.13}$$

According to equation 2.81, the contribution in 1D to the variance of A_1 is equal to $\sigma^2 c_1$. In 2D the same procedure can be followed and leads to the same contribution for the variance of coefficient A_{11}. Both contributions can be found back in the matlab codes.

The rest of the variances can be determined as reported in the theory section. After the first half of the A and B coefficients are generated, the symmetry conditions are used to find the other half of the coefficients. Thereafter, an inverse FFT algorithm is used to transform the coefficients to the spatial domain. This algorithm can be found in appendix K. The real part of the Fourier transform, i.e. the A_k coefficients, will be a zero-mean random field, having the desired standard deviation and correlation structure. To acquire the random field with the desired mean value, the mean value is added to the A_k coefficients.

Symmetric correlation structure In the statistical evaluation of the fields, a symmetric correlation structure was observed. This drawback of the method was already mentioned in the theory section. To deal with this problem, the domain of the random field is taken twice as big. After the whole field is generated, only half of the random field is taken in 1D and a quarter of the field is taken in 2D. This decreases the efficiency of the method but strongly increases the accuracy of the method since the random fields will have the desired correlation structure.

Some improvement in efficiency can be obtained by increasing the domain of the field in each direction only with L_c or twice L_c. The domain for which the values are created is smaller in that case. Another option is to take the values of the unused parts. The domain, in that case, stays twice as big in every direction. In 1D the realisation time will be twice as small and in 2D even four times as small.

Nataf transformation For the FFT method, only the mean and the standard deviation are transformed. This is done in the same manner as for the CMD method. The correlation function is not transformed since the exact expression for SDF could not be derived. This gives rise to an error in the correlation structure of the random field. For a small coefficient Of variation (COV) this error is however very small since the transformed correlation function is approximately the same as the untransformed correlation function, as can be seen in figure 3.4. For the material parameters of concrete, the COV is equal or smaller than 0.3, so it can be concluded that no big error is made when the untransformed correlation function is used.

3.2.4 Local Average Subdivision method

The theory of the LAS method is explained in section 2.4.6. The Matlab code for the LAS method can be found in appendix I.5 for 1D and in appendix I.6 for 2D.

The LAS algorithm start with the assembly of the correlation matrix for the initial random field of stage 0. Thereafter the matrices and vectors containing the a and c coefficients are allocated for every stage. The neighbourhood size in 1D can be chosen equal to 3 or 5. In 2D only a neighbourhood size of 3 is possible. All these values are determined as specified in the theory section where all the covariances are determined according to the local average theory. In 1D, both the exact variance function as gauss quadrature are used to determine the covariances. In 2D only gauss quadrature is used. The values for stage zero are generated as performed in the CMD method, whereby the covariances in the correlation matrix are determined according to the local average theory. Thereafter, the cells are divided multiple times till the desired coarseness is reached. In each stage, the a and c coefficients of that stage are used in the determination of the new values of the field to acquire the desired statistical characteristics. In the end, the desired mean value is added to the generated values to acquire the right mean value.

boundary conditions Taking into account the boundary conditions is one of the most involved parts of the LAS method. For cells which lay at boundary of the domain, the algorithm requires values which lay outside the domain. It is assumed that the values inside the domain are uncorrelated with the values outside the domain. In 1D the values outside the domain are simply set equal to 0 and it is assumed that it has no effect on the a and c coefficients. In 2D a more sophisticated approach is followed. Again it is assumed that the values inside the domain are uncorrelated with what happens outside the domain. Here a distinction is made in the a and c coefficients for the determination of the side values, corner values and interior values. The side and corner coefficients

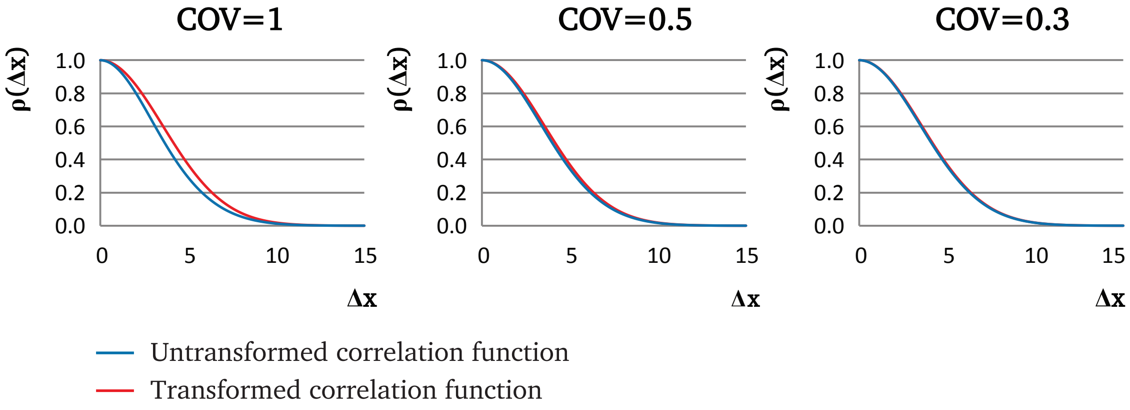


Fig. 3.4.: Influence of the coefficient of variation on the Nataf transformation of the covariance function.

are determined by taking the best linear estimation with only the cells which lie inside the domain. The a and c coefficients for the interior values are determined in the regular way.

Combining LAS with the SF disretization method To combine the LAS method with the SF discretization method, the centre of the cell is taken as nodal coordinates for the random field mesh. The generated field is made a half cell larger at every side in order to match the domain size with the coordinates of the random field mesh. The generated field has to be fine enough in order to obtain te correct variance in the field. In such a case the reduction of the variance is negligible.

Nataf transformation If a log-normal field is required, only gauss quadrature can be used to obtain the right transformed covariances of two local averages. The transformed covariances of two local averages are obtained by using the expression in equation 2.65 in the gauss quadrature algorithm. The mean value and the standard deviation are transformed in the same way as done in the CMD method. In the end, the values of the obtained field is placed in the exponent to obtain the field which is log-normally distributed.

3.2.5 Expansion Optimal Linear Estimation method

The EOLE method is one of the numerical methods to solve the KL expansion which is explained in section 2.6.1 and 2.6.3. The Matlab code for the EOLE method can be found in appendix I.7 for 1D and in appendix I.8 for 2D.

The algorithm starts with the assembly of the covariance matrix. In contrast with the other methods, where the *correlation* matrix was assembled, here the covariance function has to be used instead of the correlation function. The covariance matrix is decomposed with the modified eigendecomposition algorithm, described in CMD method section. Thereafter, the eigenvalues and corresponding eigenvectors are sorted from the largest eigenvalue to the smallest eigenvalue. Only the M largest eigenvalues and corresponding eigenvectors are then used to determine the M approximated eigenfunctions. The approximated eigenfunctions and eigenvalues are then used to generate the continuous random field. This field is then evaluated on the coordinates which correspond to the locations of the gauss points in the FE model. The evaluated values are used for the determination of the accuracy of the method.

3.3 Results of comparative study

In this section, the results of the comparative study will be presented. First, the performance with respect to efficiency will be considered. Thereafter, the performance with respect to accuracy will be considered. All the data used in this comparison can be found in appendix M. This data is obtained by running the Matlab-files which are described in the previous section where implementation issues where considered. It is assumed that the results in DIANA will be the same since they are programmed in the same way. This assumption is checked by comparing the performance for some cases in DIANA with the data obtained with Matlab. With respect to accuracy, comparable results where obtained with DIANA which gives some ground for the assumption. With regard to efficiency some significant differences were observed. These will be described in section 3.3.1.

As mentioned, the CMD method combined with Cholesky decomposition could not decompose the covariance matrix in some cases since the matrix became non-positive definite. This was the case in both 1D and 2D for L_c is equal to 5 and the number of nodes equal to 32, L_c is equal to 2.5 and the number of nodes equal to 64 and L_c is equal to 0.5 and the number of nodes equal to 256. Next to these cases, in 2D also L_c is equal to 5 and the number of nodes equal to 16, L_c is equal to 2.5 and the number of nodes equal to 32 and L_c is equal to 0.5 and the number of nodes equal to 128 gave no solution to the decomposition. For these cases no results are obtained.

In 2D, no results for the CMD method with 256 nodes in both directions could be obtained since the correlation matrix of 65536 x 65336 became to large to store.

Only the results obtained with gauss integration are presented in this section. If the variance function is used to determine the covariances in the LAS method, the method is slower than when exact integration is used. The variance function has to be evaluated multiple times in equation 2.61 to determine the covariance of two local averages which takes more time. When the variance function is used in the LAS method, comparable results are obtained with respect to accuracy.

The results of the EOLE method have been left out in this comparison since the performance of this method is very poor. Especially the realization time of this method is very large. The evaluation of the continuous function on coordinates of the integration points takes a lot time. On every coordinate the contribution of every mode has to be determined. A more efficient way to evaluate such a function on the coordinates of the integration points may be found in future studies.

3.3.1 Efficiency performance random field generators

Initialization time In 1D the initialization time of all methods is very small. For all cases the CPU-time never exceeds a value of 0.25 s. The initialization time for the FFT method is slightly larger than for the other methods. There are no significant differences in the initialization time when the correlation type, threshold value or distribution type is changed. In 2D the initialization time for the FFT and LAS method is very small in all cases. For the CMD methods the initialization time increases very strong for fields having more than 64 nodes in each direction. The decomposition time of the correlation matrix increases strongly. In figure 3.5, the initialization time of the different methods as function of the number of nodes is plotted. In this figure the line of the FFT method overlaps the line of the LAS method. Again there are no significant differences when the correlation type, threshold value or distribution type is changed.

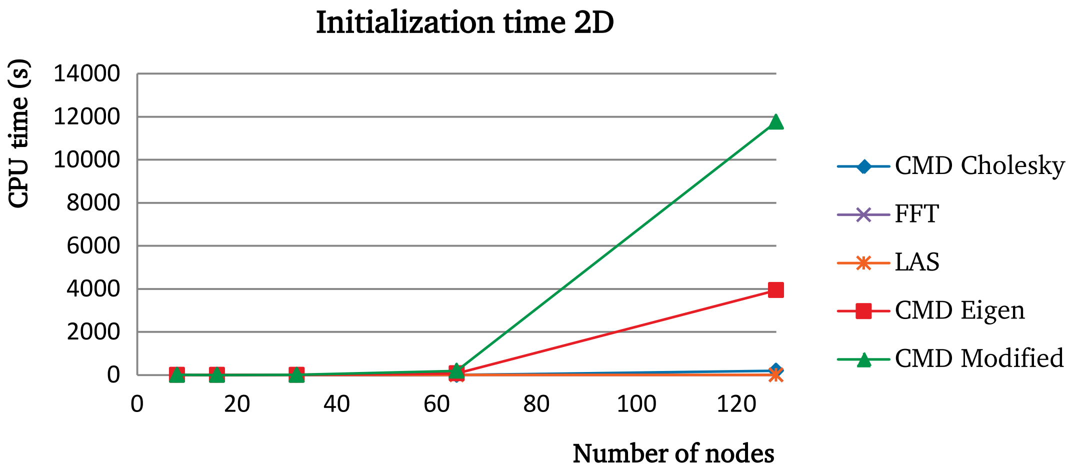


Fig. 3.5.: Initialization time of different methods in CPU-time for the 2D case. FFT and LAS overlap each other.

In DIANA the initialization time for CMD with modified Cholesky decomposition is about 7 times as short as in Matlab. The initialization time for LAS and FFT are negligible small. For the CMD method with eigendecomposition the initialization time is approximately the same in case 8, 16 or 32 nodes are used in each direction. When more nodes are used the initialization time strongly increases. This can be explained by the fact that the used algorithm in DIANA to perform an eigendecomposition differs from the one applied in Matlab. Some improvement in the initialization time in DIANA for the eigendecomposition could be achieved by applying a more efficient algorithm.

Realization time In 1D the realization time is the smallest for the CMD method. The realisation time for the LAS method is slightly larger and for the FFT method the realisation time is the largest. If the FFT method is applied without making the domain twice as big, to omit the symmetric correlation structure, the method is as fast as the LAS method. In figure 3.6 the realization time of 2000 fields in 1D are plotted as function of the number of nodes. A linear relation can be found for all the methods between the number of nodes and the realization time. In figure 3.7 the realization time of 200 fields in 2D is plotted as function of the number of nodes in each direction of the field. Here the relation seems to be quadratic. Between the total number of nodes in the random field and the realization time again a linear relation is found. The CMD method is again the fastest method in terms of CPU-time and the FFT method is the slowest method with respect to realization time. Both in 1D as in 2D, no significant differences can be found if the correlation type, correlation length, threshold value or distribution type is changed.

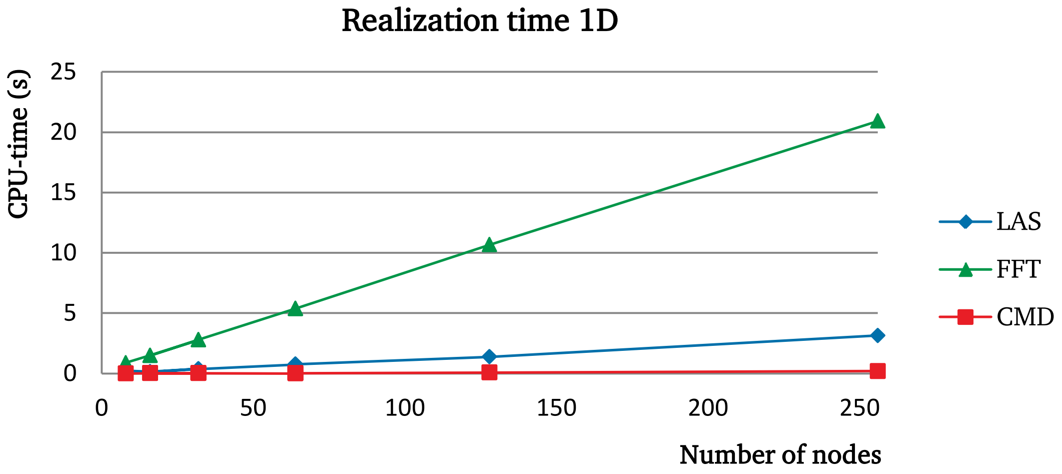


Fig. 3.6.: Realization time of different methods for the 1D case in CPU-time. All the CMD methods are represented by CMD.

In Diana also a linear relation was found between the total number of nodes and the realization time. The realization time for the CMD method is about 4 times larger in Matlab. The realization time for the FFT and LAS method is about 4 times smaller in Matlab.

Overall performance efficiency If the initialization time and realization time are considered both, the CMD method is the fastest in 1D. In 2D the CMD method is the fastest for fields having less then 64 nodes in each direction. If the fields have more than 64 nodes in each direction the LAS methods it the fastest method.

The time to determine the statistical properties of the random field is also considered. This time increases when the number of nodes increases. No significant differences can be found between the different methods, which is logical since the same procedure is followed for every method after the random field is generated to determine the statistical properties.

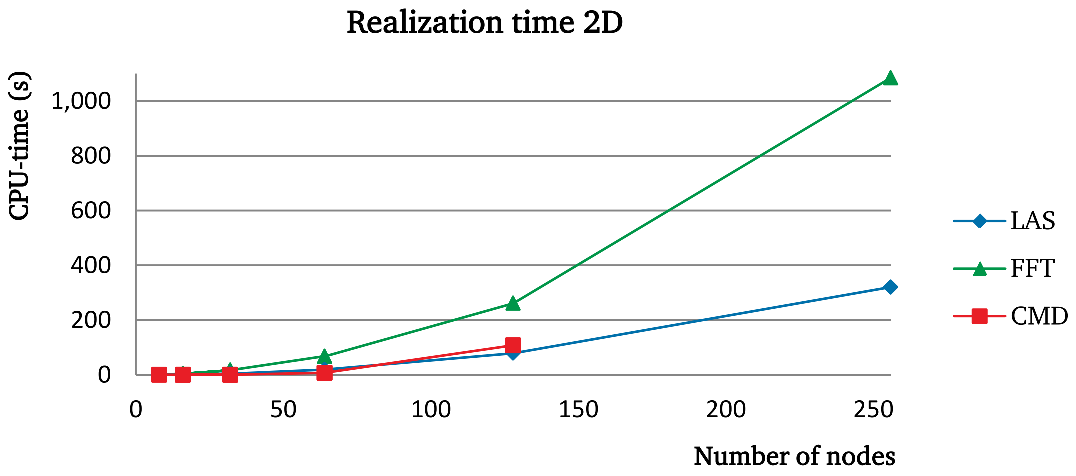


Fig. 3.7.: Realization time of different methods for the 2D case in CPU-time. All the CMD methods are represented by CMD.

3.3.2 Accuracy performance random field generators

Before the accuracy of the random field generators is assessed, it is checked if the random number generators fulfil the requirements of section 2.1.7. To do so, a sample of $2 \cdot 10^5$ generated standard normal random numbers generated in both Matlab and DIANA are assessed on their accuracy. In both cases the samples have the desired mean value and standard deviation. A Z-test is performed to see if the numbers are normally distributed. In both cases, a very small deviation is observed at the tails of the distribution. The CPU-time to generate the both samples is very small (i.e. around 0.03 seconds). Both random number generators can reproduce a given stream of random numbers exactly by initializing a seed value and both methods have a very long period. Despite the small deviations in the tails, it can therefore be concluded that both random number generators are accurate and fulfil the requirements of section 2.1.7. The results of the Z-test for both methods can be found in appendix L.

Representation of the mean value For the representation of the mean, a distinction is made between the mean value of all the mean values of the generated fields (μ_m) and the standard deviation of all the mean values of the generated fields (μ_s). First, the mean value of all the mean values (μ_m) is considered.

When the normal distribution is selected as distribution type all the methods represent the mean very well. The absolute error in μ_m is approximately equal to 0.01 in 1D and 0.02 in 2D. The number of nodes, the correlation type or method seems to have no influence on this value. The threshold value does have an influence. Both in 1D and 2D, this value is on average twice as high in case the threshold value (c_1) is set equal to 0.5. The maximum absolute error is equal to 0.05 in 1D and 0.15 in 2D.

When the log-normal distribution type is selected, the absolute error in μ_m is larger in 1D when the LAS method with a neighbourhood size of 5 is selected. The other methods perform the same as in the case the normal distribution type is selected.

For all the methods where the absolute error μ_m was small, the values for μ_s were approximately the same. The values for μ_s averaged over all the method where the absolute error in μ_m was small are given in table 3.2. It can be seen that the standard deviations of all the mean values (μ_s) are quite high, especially when the threshold value is equal to 0.5 and the correlation length is large.

Tab. 3.2.: Averaged value μ_s for different values of the correlation length, threshold value and distribution type given for the 1D and 2D case.

	1D				**2D**			
	c_1=0		c_1=0.5		c_1=0		c_1=0.5	
$\mathbf{L_c}$	normal	log-normal	normal	log-normal	normal	log-normal	normal	log-normal
5	0.64	0.57	0.84	0.75	0.43	0.36	0.77	0.65
2.5	0.49	0.43	0.79	0.69	0.26	0.21	0.73	0.60
0.5	0.23	0.20	0.73	0.61	0.06	0.05	0.70	0.59

In 1D, the CMD with the modified Cholesky decomposition gives deviating results in some of the cases where regular Cholesky decomposition could not obtain a solution. The decomposed correlation matrix is not approximated well by the modified Cholesky decomposition. The deviation in the mean values is larger in those cases. When the tolerance value for the argument of the square root in the modified Cholesky algorithm is changed this error can be made smaller. When the log-normal distribution was selected in 1D, the LAS method with a neighbourhood of 5 gives higher values for μ_s.

Overall the LAS method with a neighbourhood size of 5 and a log-normal distributed random field gave higher values for the errors in μ_m and μ_s. The rest of the methods give approximately the same results. The errors in the representation of the mean are relative small which makes those methods appropriate for representing the mean value. The deviation in the mean values is quite large. Increasing the number of samples does not give a change in this value. As will be shown in the next section, the distribution in the field will be changed when the variables are stronger correlated. This is the case when a correlation function is used with a threshold value equal to 0.5 and a large correlation length. The deviation in the mean values is slightly smaller when a SExp correlation function is used then when a Exp correlation function is used.

Representation of the standard deviation In the assessment of the random field generators on the representation of the standard deviation a distinction is made in the mean value of all the standard deviations (σ_m) and deviation in the all the standard deviations (σ_s).

When the results are observed for σ_m it can be seen that, for all the methods, the standard deviation reduces when the correlation length increases and a threshold value of 0.5 is selected. Exactly in the same way the deviation in the mean (μ_s) values increases, the mean values of all the standard deviations of each field decreases. This can be seen in table 3.3, where the values of σ_m are averaged over all the methods. For uncorrelated variables, the mean value of all the standard deviations should be equal to 1. In the case of random fields the random variables are however correlated with each other. The desired value is therefore not known a priori. Because all the method have approximately the same values for σ_m, it is assumed that the methods which have values close to the values in table 3.3 are accurate with respect to the representation of the standard deviation. The LAS method slightly deviates from these values since the variance is further reduced by the variance function. If the domain size is small compared to the correlation length this deviation is almost negligible. The values are then approximately the same as in table 3.3. In 1D, the LAS method with a neighbourhood size of 5 the values for σ_m are all larger than the values in the table. When the CMD method is combined with modified Cholesky decomposition the value for σ_m is higher for the same cases as where the mean value was represented poorly.

Tab. 3.3.: Averaged value σ_m for different values of the correlation length, threshold value and distribution type given for the 1D and 2D case.

	1D				2D			
	c_1=0		c_1=0.5		c_1=0		c_1=0.5	
L_c	normal	log-normal	normal	log-normal	normal	log-normal	normal	log-normal
5	0.71	0.62	0.52	0.46	0.87	0.81	0.62	0.58
2.5	0.83	0.74	0.59	0.55	0.95	0.91	0.67	0.64
0.5	0.96	0.90	0.68	0.64	0.98	0.96	0.69	0.67

When the deviation in standard deviations (σ_s) are considered, again a clear pattern in the results can be found if the normal distribution type is selected. The averaged values are given in table 3.4. It can be seen that the value for σ_s increases when the correlation length increases. When the SExp correlation function is chosen, the values are larger then when the Exp correlation function is chosen. If a threshold value of 0.5 is applied the values are smaller. The values for LAS with a neighbourhood size of 5 deviates slightly from the values given in the table.

Tab. 3.4.: Averaged value σ_s for different values of the correlation length, threshold value and correlation function given for the 1D and 2D random fields which are normally distributed.

	1D				2D			
	c_1=0		c_1=0.5		c_1=0		c_1=0.5	
L_c	Exp	SExp	Exp	SExp	Exp	SExp	Exp	SExp
5	0.23	0.30	0.16	0.22	0.14	0.18	0.09	0.13
2.5	0.21	0.26	0.15	0.18	0.10	0.11	0.07	0.08
0.5	0.12	0.13	0.08	0.09	0.03	0.02	0.02	0.02

When the log-normal distribution type is selected, the pattern in the results is less clear, especially when a threshold value of 0.5 is selected. In that case, most of the values are close to 0.4 both in 1D and 2D. Again the values for σ_s are slightly larger when the SExp correlation is chosen in stead of the Exp correlation function.

In general, the LAS method with a neighbourhood size of 5 gives deviating results, which indicates that this method represents the standard deviation less accurate. The LAS method in general under represent the variance because it is reduced by the variance function. This is correct if this method is combined with the SA discretization method. If it is however combined with the SF discretization method a small domain size have be selected to represent the point variance of the random field correct.

Representation of the correlation structure The correlation structure of the generated random fields is compared with the exact correlation function. For all the lag distances equal to a multiple of the element size of the random field mesh the correlation coefficient is determined. The absolute error with the exact correlation function is then determined. The mean values of all these errors is called the mean correlation error ($C_{err,m}$) and the deviation in these values is called the deviation in the correlation error ($C_{err,s}$).

When the mean correlation error is considered it can be seen that the error depends significantly on the used method. The values are therefore averaged per method and are given in table 3.5 till 3.7. The mean correlation error is a lot smaller in case the normal distribution is chosen instead of the log-normal distribution. If the normal distribution type is selected, the CMD method combined with

Cholesky or eigendecomposition gave the smallest error in both 1D and 2D. When the log-normal distribution is selected, the FFT method gives the smallest error both in 1D and 2D. Especially when the a threshold value of 0.5 is selected, this error is a lot smaller then for the other methods. Despite of the untransformed correlation function, the FFT method seems to represent the correlation structure very well when a strong transformation to the log-normal distribution is performed. The mean correlation error of LAS with a neighbourhood size of 3 is in between the errors of the CMD and FFT method for both 1D and 2D. When a neighbourhood size of 5 is selected the error is the largest.

Tab. 3.5.: Averaged value of the mean correlation error for every method (part 1).

		1D							
		CMD						*FFT*	
		Cholesky		Eigen		Mod. Chol.			
	c_1	Exp	SExp	Exp	SExp	Exp	SExp	Exp	SExp
Normal	**0**	0.0083	0.0078	0.0074	0.0073	0.0103	0.0398	0.0095	0.0111
	0.5	0.0055	0.0071	0.0071	0.0085	0.2331	0.2575	0.0126	0.0107
Log-normal	**0**	0.0479	0.0302	0.0487	0.0311	0.0618	0.0620	0.0215	0.0162
	0.5	0.1523	0.1503	0.1578	0.1590	0.4450	0.4433	0.0735	0.0758

Tab. 3.6.: Averaged value of the mean correlation error for every method (part 2).

		1D				**2D**			
		LAS				*CMD*			
		nbh=3		nbh=5		Cholesky		Eigen	
	c_1	Exp	SExp	Exp	SExp	Exp	SExp	Exp	SExp
Normal	**0**	0.0210	0.0176	0.0239	0.0283	0.0101	0.0134	0.0095	0.0087
	0.5	0.0251	0.0159	0.0717	0.0935	0.0212	0.0083	0.0191	0.0292
Log-normal	**0**	0.0357	0.0316	0.0931	0.0766	0.0509	0.0286	0.0609	0.0367
	0.5	0.1415	0.1597	0.3171	0.3607	0.1673	0.1635	0.1458	0.1454

Tab. 3.7.: Averaged value of the mean correlation error for every method (part 3).

		2D					
		CMD		*FFT*		*LAS*	
		Mod. Chol.					
	c_1	Exp	SExp	Exp	SExp	Exp	SExp
Normal	**0**	0.0156	0.0682	0.0114	0.0099	0.0218	0.0167
	0.5	0.1958	0.2966	0.0293	0.0146	0.0309	0.0205
Log-normal	**0**	0.0926	0.0901	0.0226	0.0254	0.0291	0.0336
	0.5	0.3174	0.3837	0.0298	0.0270	0.1081	0.1555

The deviation in the correlation error is strongly linked to the mean correlation error. When the mean error is small, the deviation in the values is also small and visa versa. When the normal distribution type is selected, these values are slightly smaller as the mean values. In case the log-normal distribution is selected, these value are slightly higher than the mean values. This means that in case the log-normal distribution is selected, the correlation structure shows a more rough pattern then when the normal distribution is selected.

To illustrate the absolute error in the correlation structure, the correlation structures for some cases are plotted. In figure 3.8 the correlation structure, estimated over 2000 normal distributed 1D random fields, is shown. The squared exponential correlation function is used with a correlation length of 5 m and no threshold value. In figure 3.9 the correlation structure, estimated over 2000 log-normal distributed 1D random fields, is shown. The squared exponential correlation function is used with a correlation length of 5 m and a threshold value of 0.5 m. In figure 3.10 the correlation structure, estimated over 200 normal distributed 2D random fields, is shown. The exponential correlation function is used with a correlation length of 2.5m and a threshold value of 0.5 m. The same observations can be made with regard to the absolute error in the correlation structure as done in the previous paragraphs. When no threshold value is applied, the LAS method with neighbourhood size of 5 gives the largest deviation in the correlation structure. When the log-normal distribution is selected the FFT methods represents the correlation structure the most accurate.

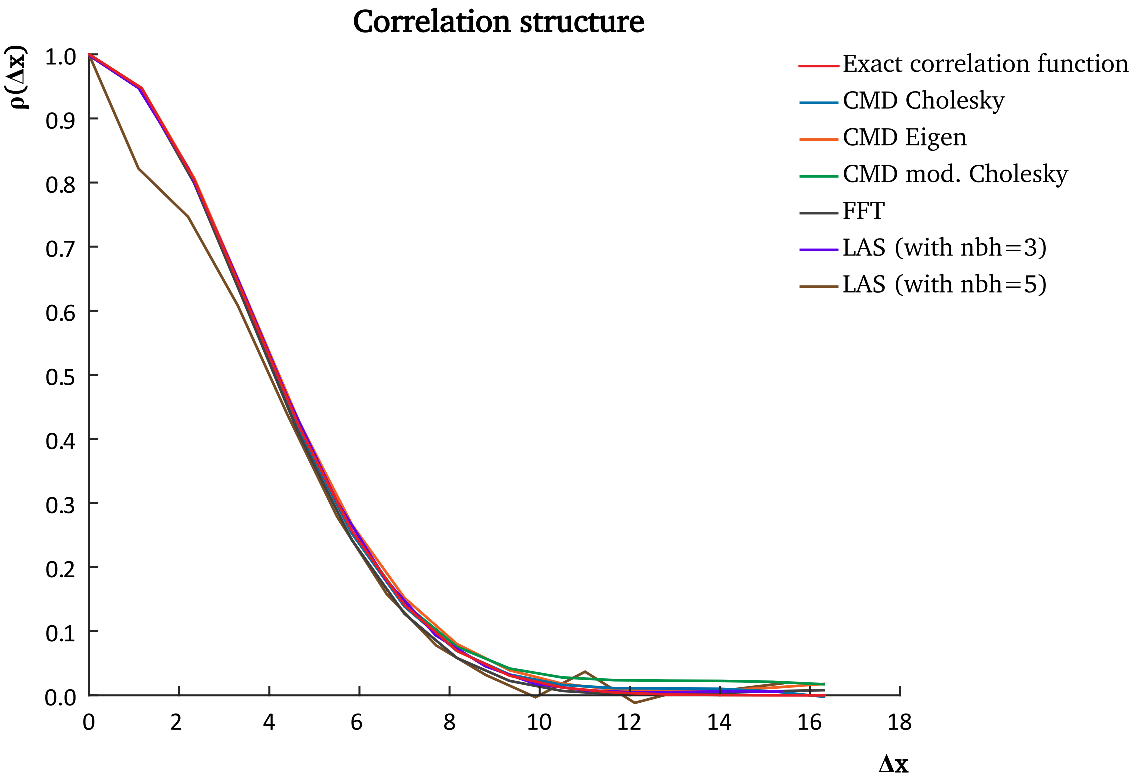


Fig. 3.8.: Correlation structure of different methods in 1D estimated over 2000 realizations for normal distributed random fields generated with a squared exponential correlation function with no threshold value.

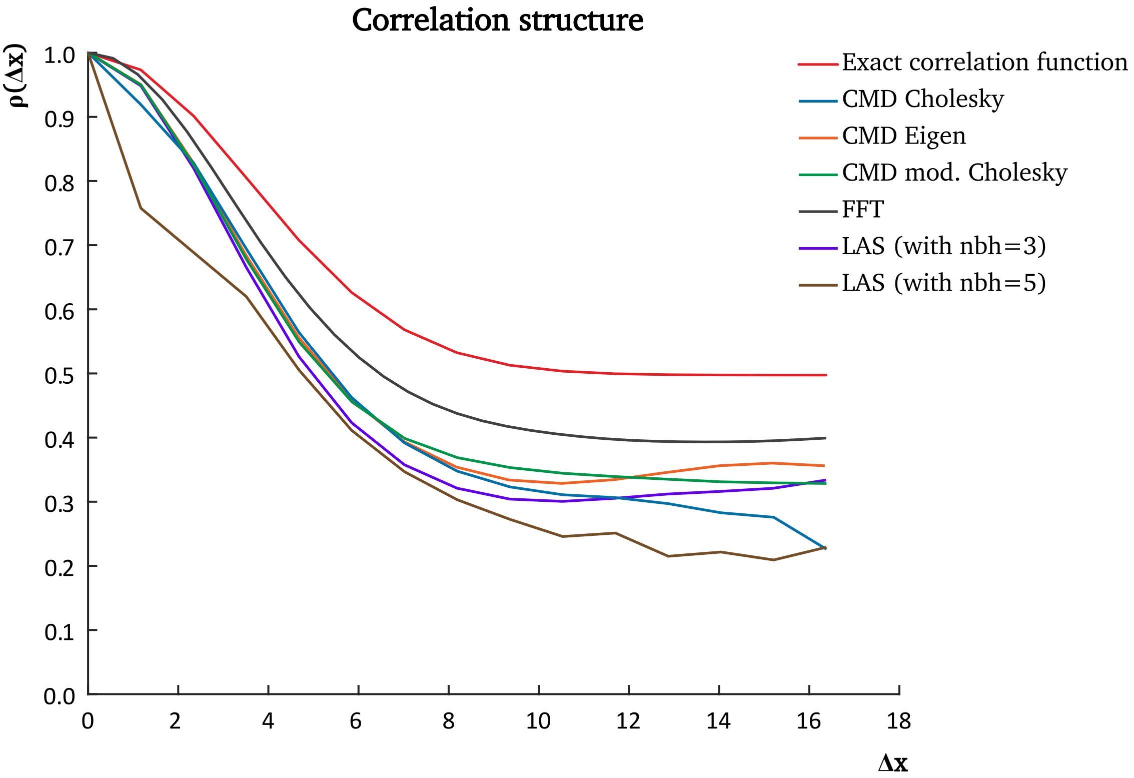


Fig. 3.9.: Correlation structure of different methods in 1D estimated over 2000 realizations for log-normal distributed random fields generated with a squared exponential correlation function with a threshold value of 0.5.

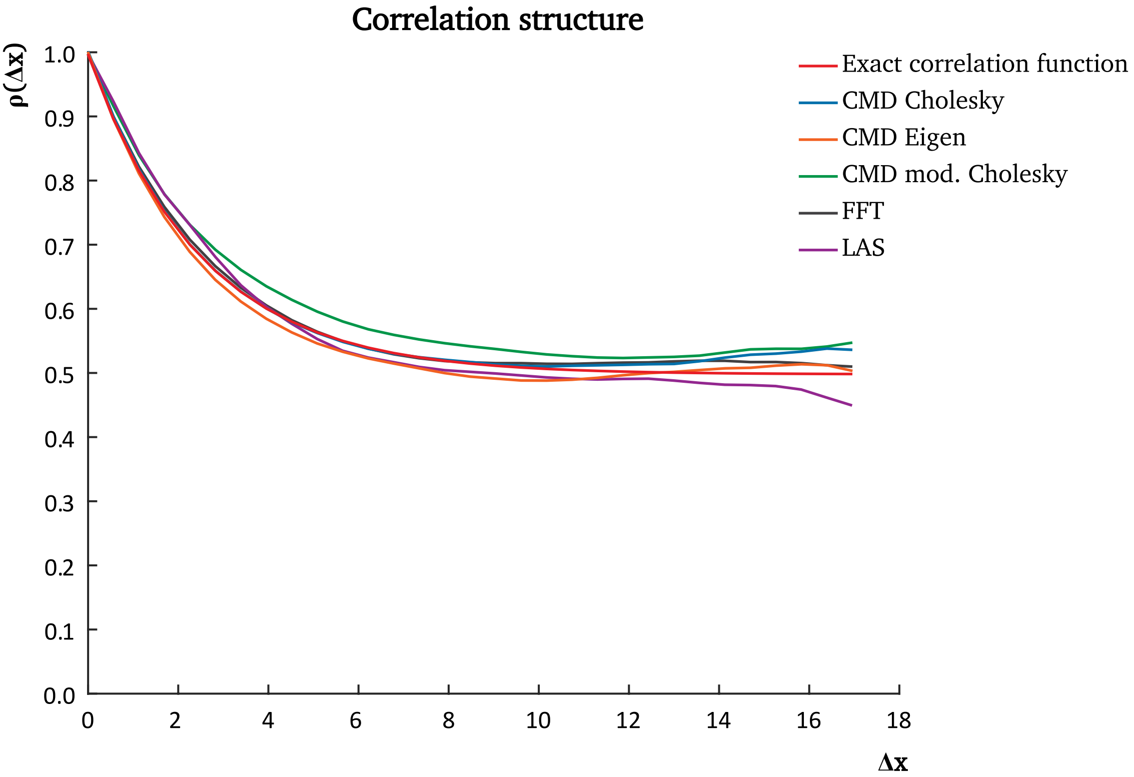


Fig. 3.10.: Correlation structure of different methods in 2D estimated over 200 realizations for normal distributed random fields generated with a exponential correlation function with a threshold value of 0.5.

3.3.3 Strong correlation and loss of ergodicity

It was observed that for random fields with a correlation function having a large correlation length and a threshold value of 0.5, the variance in the field decreases and the deviation in the mean values increases. In this section this behaviour is studied in more detail. To do so, several histograms are considered which are shown in figure 3.11 and 3.12. At the LHS of those figures, the histograms are plotted with values of one single random field. At the RHS of the figures, the histograms are plotted with the values on a single location in the field for all the 1024 generated random fields. The 2D normal distributed random fields have a SExp correlation function and a domain size of 17.5 x 17.5 m. The number of nodes in each direction is equal to 32, which results in 1024 nodes per field. In figure 3.11 the correlation length was equal to 0.5 m and in figure 3.12 the correlation length was equal to 5 m. The threshold value ranges from zero to 0.5. The following observations can be made for the histograms at the RHS of the figures:

- The mean of the distribution is equal to zero.
- The width of the distribution stays approximately the same.
- The shape of the distribution looks like a normal distributed sample.

For the histograms at the LHS of the figure the following observation can be made:

- The mean deviates from zero and the width of the distribution is smaller.
- The shape of the distribution deviates from the shape of a normal distribution.
- Both effects are observed stronger when the correlation length increases and when the threshold value is equal to 0.5 as can be seen in figure 3.12.

From this observations it can be concluded that ergodicity is lost when the values in the field are strongly correlated with each other. This is the case when the correlation length is large in comparison with the domain size and when a threshold values is applied. Normally, it is assumed that random fields are ergodic, however, for random fields where the values are strongly correlated this assumption is not valid. The joint PDF of the random field can not be determined by a single realization of the random field any more. Unfortunately, no explicit relation is found between the used correlation function and the target value for the standard deviation in the field and deviation in the mean values. It can only be stated that those values are in some way related to the area under the correlation function. A larger area results in a smaller deviation in the field and a larger deviation in the mean values of the random fields. When random fields are generated for the modelling of spatial variations in concrete this effect has to be considered.

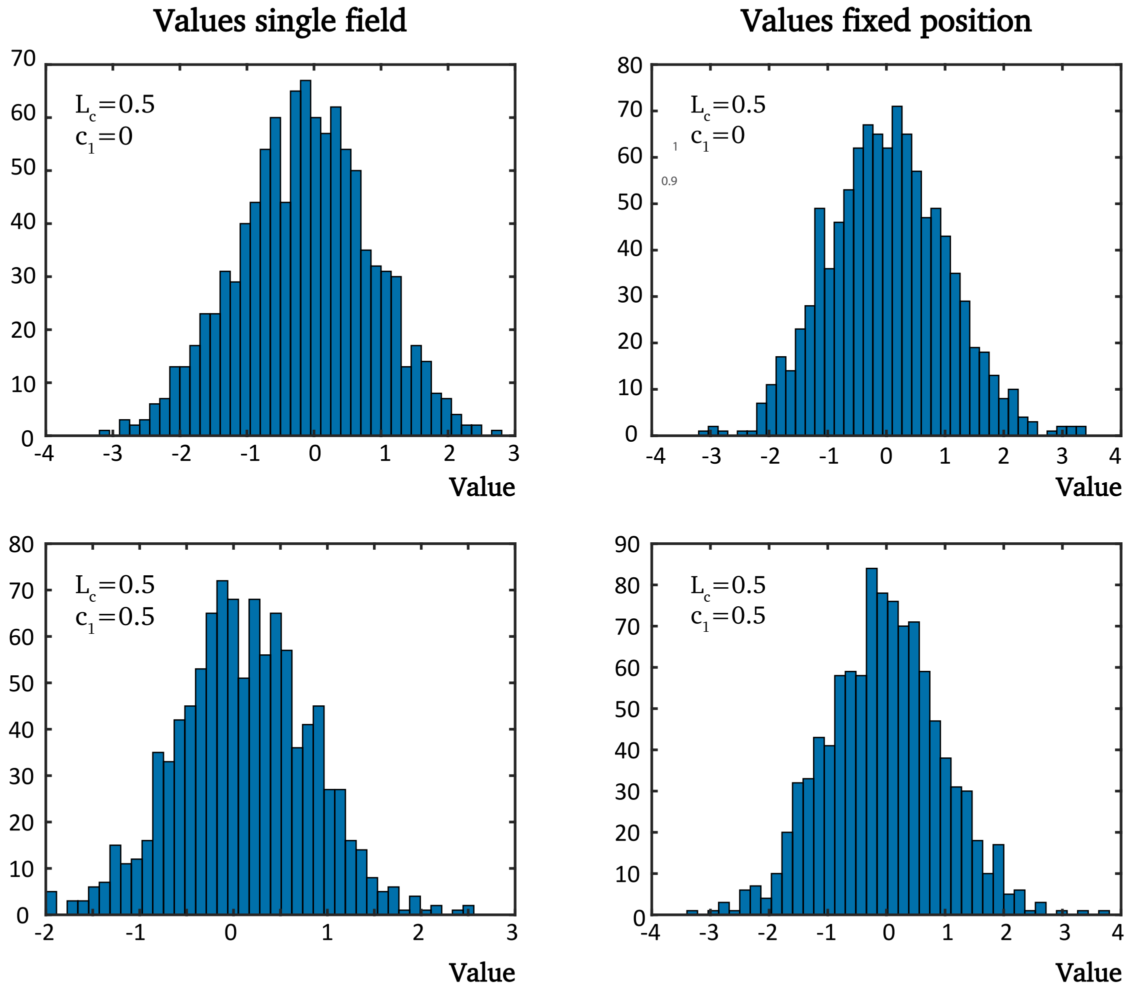


Fig. 3.11.: Histograms with values of the random field for L_c equal to 0.5 m and different values for c_1. Left: values for one field, right: values for 1024 field at one point.

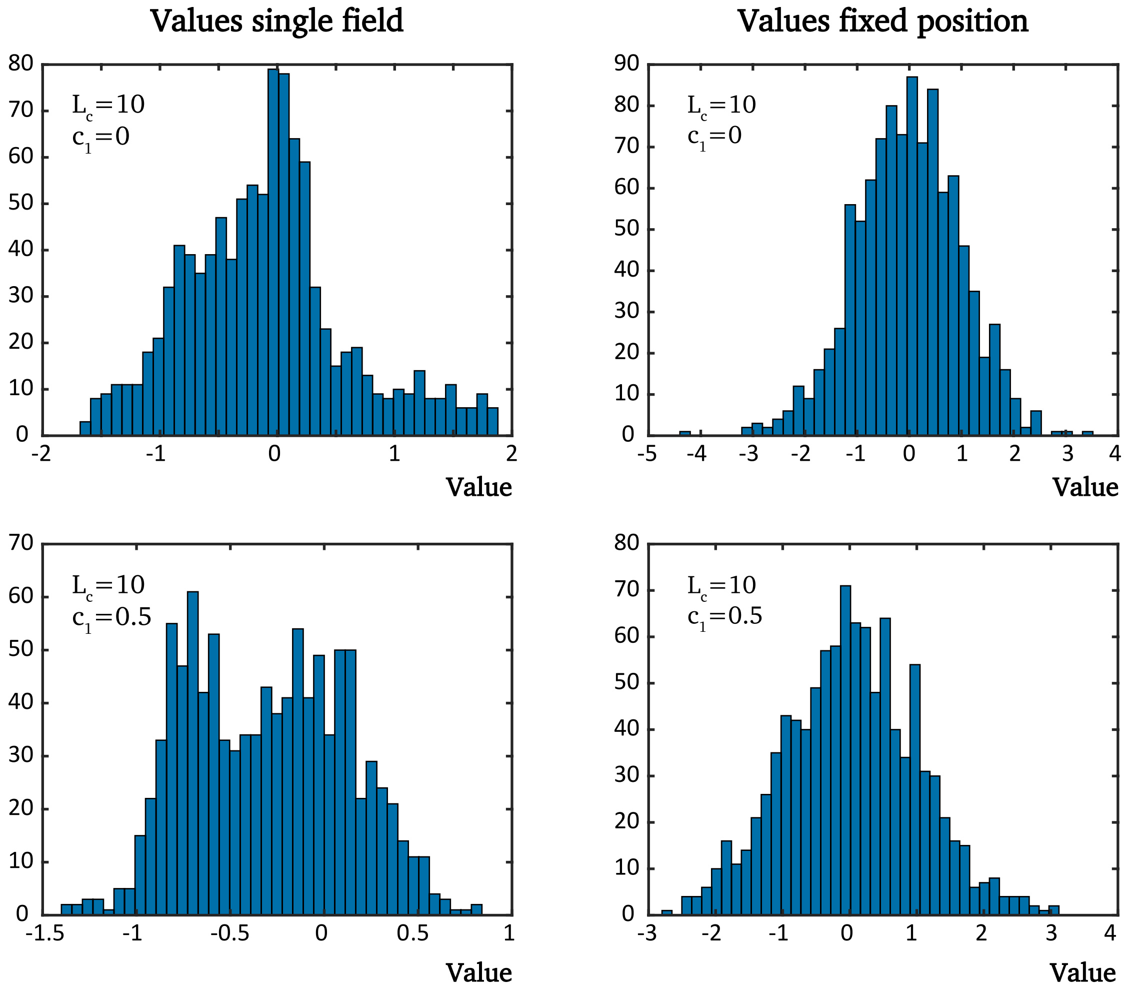


Fig. 3.12.: Histograms with values of the random field for L_c equal to 10 m and different values for c_1. Left: values for one field, right: values for 1024 field at one point.

3.4 Conclusion

In this chapter, the influence of the different parameters on the statistical properties of a random field is studied. Also the performance of the different generators is assessed. In this section these aspects are both summarized to come to a conclusion. This conclusion can then be used in the next chapter where a method is chosen based on the selected correlation function, distribution type and other statistical characteristics for the modelling of spatial variability of concrete material properties. First, the influence of the different parameters on the statistics of a random field is considered.

3.4.1 Influence of different parameters on the statistics of random fields

The number of nodes in each direction is varied based on the desired ratio between the correlation length and the element size of the random field mesh. This indeed led to accurate results for the random fields for both the Exp and SExp correlation function. Both the initialization time and realization time increases when the number of nodes increases. A linear relation is found between the number of nodes and the realization time for all methods.

Between the Exp and SExp correlation function no significant differences where found with respect to efficiency. With respect to accuracy the differences were very small. The deviation in the mean values were slightly smaller and the deviation in the standard deviations were slightly larger when a SExp correlation function is chosen in stead of the Exp correlation function. In general it can be concluded that the correlation function type has no significant influence on the efficiency and accuracy of the random fields. It has an effect on the pattern of the random fields. The nodes separated by a lag smaller than L_c are stronger correlated and the nodes separated by a lag larger than L_c are less strong correlated when the SExp correlation function is chosen in stead of the Exp correlation function. The fields will therefore have a more smooth surface when a SExp correlation function is chosen. With regard to implementation issues some problems were encountered with the decomposition of the covariance matrix when the SExp correlation function is selected, especially when the ratio between L_c and L_{RF} was relatively high.

The correlation length has no significant influence on the accuracy of the different methods. For a larger correlation length the mean correlation error is slightly larger. With respect to ergodicity the correlation length has a significant influence. The deviation in the mean values increases and the mean standard deviation decreases when the correlation length is larger. It can unfortunately not be stated if the obtained values for the deviation in the mean and the mean values of the standard deviations are accurate since there is no explicit relation known.

The use of a threshold value for the correlation function has a negative influence on the accuracy of the random field. In general, the defined errors became somewhat larger, especially when a threshold value was used in combination with the log-normal distribution type. The use of a threshold value results in a stronger correlation between the different random variables in the random field which again leads to a loss of ergodicity.

A very strong transformation from a Gaussian random field to a log-normal distributed random field with a COV of 1.0 was carried out to study the influence on the accuracy. It can be concluded that the distribution type has a significant influence on the accuracy of the random fields. The error in the mean of all the mean values increased. The error in the mean value of the standard deviations became slightly smaller in most cases. With respect to the representation of the correlation structure the largest differences where observed. The errors became much larger when a log-normal distribution type was selected. In the case of concrete parameters, the COV is much smaller, which gives a less strong transformation. The errors will be more close to the errors found when a normal distribution type was selected.

3.4.2 Performance different random field generators

In this section the performance of the different methods is discussed. From the assessment it can be concluded that the LAS with a neighbourhood size of 5 is inappropriate for the generation of random fields. The accuracy of the method is lower than for the other methods, which can be attributed to the poor determination of the values at the boundary of the field. When a SExp correlation function is selected, the CMD method with Cholesky decomposition is inappropriate because the covariance matrix can not be decomposed in most cases. If the modified Cholesky decomposition method is used instead, it gave accurate results for most of these cases. However, for some cases the results where not accurate. The decomposed matrix is not approximated very well, which leads to larger errors. For the CMD with eigendecomposition, FFT and LAS with a neighbourhood size of 3, no large errors were observed. It depends on the selected parameters which method is the most appropriate to select.

First, the case that a normal distribution is selected as distribution type is considered. In 1D the CMD method with eigendecomposition is the best method to choose. It is the most efficient method, gives solutions for both the Exp and SExp correlation function and gives the most accurate result for the representation of the mean value and correlation structure. In 2D the CMD method with eigendecomposition represents the mean the most accurate. The correlation structure is represented by the best by the FFT method and CMD method with eigendecomposition. With respect to efficiency the LAS method performs the best. Till 64 nodes in each direction the CMD method with eigendecomposition is as fast as the LAS method. For models which have less than 64 nodes in both directions it is therefore concluded that the CMD method with eigendecomposition is the best method. If the number of nodes increases the FFT method can be selected best. Although the method is less efficient than the LAS method it is slightly more accurate then LAS which is considered as more important for non-linear analysis of concrete models. In comparison with the runtime of such models this difference in efficiency can be neglected.

If the log-normal distribution is selected, the error in the correlation structure is quite large for most of the methods, especially when the threshold value is equal to 0.5. The FFT method represents the correlation structure the most accurate in 1D and 2D. The representation of the mean value by the FFT method is comparable with the other methods. The FFT method should therefore be selected for random fields with a strong transformation to a log-normal distributed random field.

For random fields with a log-normal distribution and a COV between 0.1 and 0.3, what is the case for concrete material properties, the error in the correlation structure will be closer to the error when the normal distribution is selected. Based on the results in this chapter one unambiguous conclusion which method performs the best with respect to accuracy is therefore hard to draw for such a case. With respect to efficiency the FFT method performs much better than the CMD method combined with eigendecomposition when the total number of nodes is large.

4 Random field generation for reinforced concrete

In the previous chapter, the different methods to generate a random field were assessed. In this assessment a wide range of values, based on findings in literature (see section 2.7), was used for the statistical characteristics which are involved in the modelling of the spatial variability of concrete material properties. In this chapter specific values for these statistical characteristics are selected for the implementation in DIANA. Based on the findings in the previous chapter, a selection is made of random field generators which perform the best with the selected statistical characteristics. Between the two most appropriate methods, a more specific comparison is made to identify the best method. In this comparison, the selected statistical characteristics are used and the size and shape of the domain of the random field is varied. After this comparison the most appropriate method is selected. At the end of this chapter, plots of random fields for the different material properties of concrete are shown to give an impression on how they vary in space. In the next chapter, the implemented approach to model the spatial variability of concrete properties is tested in a small example. In the case poor results are obtained, other statistical characteristics can be selected to follow the same selection procedure, starting with the findings of the assessment in the previous chapters.

4.1 Selection of statistical characteristics of concrete

In the different approaches to determine the reliability of concrete structures, a large variation was found in the statistical characteristics which are involved in the modelling of the spatial variation of concrete properties. An overview of the used values can be found in table 2.2. In some cases, the values where determined by fitting the results of the FE analysis with experimental results [39]. In other cases, a sensitivity analysis was carried out with varying values for the different statistical characteristics, to determine their influence on the reliability of a concrete structure [11]. For the deterministic relations between the different material properties, often the CEB-FIB 1990 model code is used. In general it seems hard to determine the statistical characteristics with regard to spatial variations directly from experimental tests. For example, the correlation length varies from 0.5 to 5 m. Only in [36] reference is made to experimental data where a correlation length of 0.5 m is found. In the other approaches no reference to experimental data or test is made to support the chosen value.

The Probabilistic Model Code of the Joint Committee on Structural Safety (JCSS) gives a thorough guideline for the modelling of the spatial variation in the material properties of concrete. In literature however, no example was found where the JCSS model code was applied in a FE analysis of a concrete structure. It is therefore a very interesting approach to follow. Beside the lack of an application, the relatively high correlation length and the use of a threshold value in the advised correlation function make it also worthwhile to investigate this guideline. A large correlation length is beneficial with respect to computation time. When a correlation length of 5 m is applied, the random field mesh can be more coarse. As shown in the previous chapter the initialization and realization time, increases with increasing number of nodes in the random field mesh. In all the other examples found in literature a smaller correlation length is applied. The JCSS approach is thus different from other approaches which makes it interesting to explore the use of this guideline. The JCSS model code is therefore chosen to follow for the implementation in DIANA to model the spatial variation of concrete material properties. No information is given for the *spatial* variation of the loading or geometric properties in the JCSS model code. Examples of geometric properties are the thickness of a plate or the cover for the reinforcement. Additionally, spatial variations of the properties of reinforcement steel are not taken into account. All these variations are taken into account by using a random variable instead of a random field in the JCSS model code. In this study, only variations in material properties of concrete will therefore be considered.

4.1.1 Model code of Joint Committee on Structural Safety

In this section the probabilistic model code of the Joint Committee on Structural Safety (JCSS) for concrete properties will be explained. This specification for the material properties of concrete can be found in the third part of the JCSS probabilistic model code [20] which contains the resistance models.

According to this guideline, one single log-normal random field has to be generated to which the material properties are related. This random field represents the basic compression strength of concrete (f_{c0}) which is based on standard test specimens (cylinder of 300 mm height and 150 m diameter), tested according to standard conditions and at a standard age of 28 days. On a particular point *i* in a given structure *j* it is expressed as:

$$f_{c0,ij} = exp(U_{ij}\Sigma_j + M_j) \tag{4.1}$$

where U_{ij} is the collection of spatially correlated standard normal random variables which represent the variability within one structure. These values are correlated according to the SExp correlation function using a correlation length of 5 m and a threshold value of 0.5 which result in the following correlation function:

$$\rho(\Delta x) = 0.5 + 0.5 \cdot \exp\left(-\left(\frac{\Delta x}{5}\right)^2\right) \tag{4.2}$$

Σ_j and M_j are the transformed standard deviation and mean value of the basic concrete compression strength. If no data is available a priori, the values in table 4.1 can be used. In this table also the untransformed values for the mean and the standard deviation of the log-normal distributed basic

Tab. 4.1.: Mean and standard deviation of the basic compressive strength.

Concrete type	Grade	M	Σ	μ_{ln}	σ_{ln}
Ready Mixed	C15	3.40	0.192	30.52	5.90
Ready Mixed	C25	3.65	0.164	39.00	6.45
Ready Mixed	C35	3.85	0.123	47.35	5.86
Ready Mixed	C45	3.98	0.096	53.76	5.17
Pre-cast elements	C25	3.80	0.123	45.04	5.57
Pre-cast elements	C35	3.95	0.110	52.25	5.74
Pre-cast elements	C45	4.08	0.090	59.39	5.38
Pre-cast elements	C55	4.15	0.065	63.57	4.11

compressive strength are given. All the values in the table are sorted per concrete type and grade. If prior data is available for job j, the values for Σ_j and M_j can be determined based on the available information using a student distribution for f_{c0}.

The probabilistic relation of the concrete compressive strength (f_c) with the basic compression strength (f_{c0}) is given by:

$$f_{c,ij} = \alpha(t,\tau)(f_{c0,ij})^{\lambda}Y_{1,j} \tag{4.3}$$

Where:

$\alpha(t,\tau)$ is a deterministic function which takes into account the concrete age at the loading time *t* [in days] and the duration of loading τ [in days]. For most cases $\alpha(t,\tau)$ is equal to 0.8. This value is therefore used for the implementation in DIANA.

λ is a factor taking into account the systematic variation of in-situ compressive strength, and strength of standard test. In most cases it is sufficient to take λ as a deterministic variable and equal to 0.96.

$Y_{1,j}$ is a log-normal variable representing additional variations due to the special placing, curing and hardening conditions of in-situ concrete at job *j*.

Other material properties like the concrete tensile strength (f_{ct}), the modulus of elasticity (E_c) and the ultimate compression strain (ε_u) are related with the concrete compressive strength as follows:

$$\begin{aligned}
f_{ct,ij} &= 0.3f_{c,ij}^{2/3}Y_{2,j} && (4.4)\\
E_{c,ij} &= 10.5f_{c,ij}^{1/3}Y_{3,j}(1+\beta_d\phi(t,\tau))^{-1} && (4.5)\\
\varepsilon_{u,ij} &= 6\cdot 10^{-3}f_{c,ij}^{-1/6}Y_{4,j}(1+\beta_d\phi(t,\tau)) && (4.6)
\end{aligned}$$

Where:

β_d is the ratio of the permanent load to the total load and depends on the type of the structure. In DIANA a value of 0.7 is take for this value.

$\phi(t,\tau)$ is the creep coefficient which is set equal to 1.0 in the implementation in DIANA since it is assumed that creep does not take place. For short term loading this is a good assumption.

$Y_{2,j}$, $Y_{3,j}$ and $Y_{4,j}$ are log-normal distributed random variables which mainly reflect variations due to factors not accounted for by concrete compressive strength. Examples are the gravel type and size, chemical composition of cement and other ingredients and climactic conditions. The mean values and coefficient of variation (COV) of those variables and of the random variable $Y_{1,j}$ are given in table 4.2. Again these values may also be determined from direct measurements. In the relations of the concrete material parameters, the variables $Y_{1,j}$ to $Y_{4,j}$ may also be taken as a spatially varying random fields. However, no correlation coefficients between the different variables and no correlation functions for the different variables are given in the JCSS model code.

Tab. 4.2.: Mean and COV for the different random variables used in the relations for the material properties of concrete.

Variable	μ	COV	Related to:
$Y_{1,j}$	1.0	0.06	Compression
$Y_{2,j}$	1.0	0.30	Tension
$Y_{3,j}$	1.0	0.15	E-modulus
$Y_{4,j}$	1.0	0.15	Ultimate strain

Based on the values of the random field for the basic compressive strength, the values for all other material parameters for concrete can be determined. The schematic stress-strain relationship for concrete is shown in figure 4.1. The material properties are shown in blue in this scheme, which vary for every position in a model, when the JCSS model code is applied. The use of JCSS in a FEM analysis results thus in a different stress-strain relationship for all the gauss points.

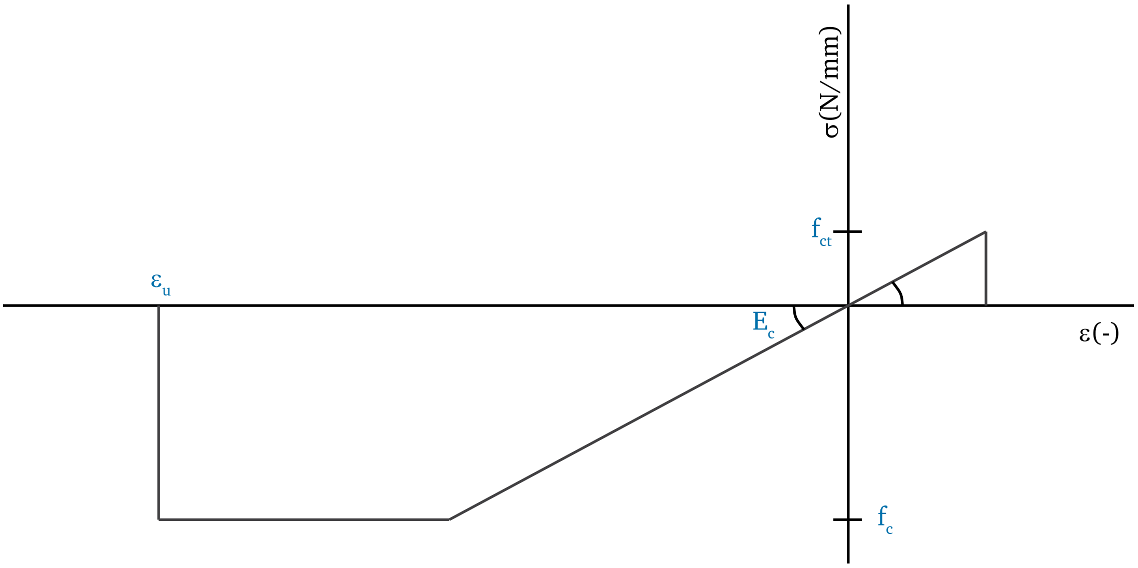


Fig. 4.1.: Schematic stress-strain relationship according to JCSS where f_c, f_{ct}, E_c and ε_u vary for every integration point.

4.2 Selection of a random field generator

In this section, the best method to generate a random field with the described properties in the JCSS model code will be selected. This will be done based on the findings in chapter 3 and on an additional comparison. In this comparison different domain sizes and shapes are used and the coarseness for the random field mesh is varied. This comparison will give a more solid ground for concluding which method is the best, and it will give an indication for the desired coarseness of the random field. First, the findings of chapter 3 with respect to the selected statistical characteristics will be discussed.

4.2.1 Findings from the assessment of random field generators

In the JCSS model code, a log-normal random field with a squared exponential correlation function having a correlation length of 5 m and a threshold value equal to 0.5 is advised. The COV varies from 0.6 to 0.19 for the different concrete grades given in table 4.1. For the C15 ready mixed concrete type the highest COV is found, which is equal to 0.19. Based on the findings of the assessment between the different random field generators, it cannot be concluded which of the methods is the most accurate. Both the CMD method combined with eigendecomposition as the FFT method could be a good alternative. It can however be stated that for random fields with a large number of nodes the FFT outperforms the CMD method combined with eigendecomposition. In the next section the results of an additional comparison are presented.

4.2.2 Additional assessment of random field generators

To investigate which of the methods is the most accurate for the specific statistical characteristics of the JCSS model code, an extra comparison is made. The suggested correlation function in equation 4.2 is used and the C15 ready mixed concrete type is selected for this assessment. Only the 2D case is considered since no significant differences were found between the results for 1D and for 2D. In 2D, the domain size and shape is varied to investigate if the conclusions can be drawn in general. The size of the square is varied from 5 m to 80 m and the elongated rectangle had a size of 5 m x 40 m. For both cases the number of nodes is varied to study the desired coarseness of the random field. This comparison is carried out in DIANA. In table 4.3 an overview of different domain sizes and the used number of nodes is given.

Tab. 4.3.: Set-up for the additional comparison made in DIANA.

Shape of domain	Size of domain	Number of nodes in x and y direction
Square	5 m x 5 m	2x2, 4x4 and 8x8
	10 m x 10 m	2x2, 4x4 8x8 and 16x16
	20 m x 20 m	2x2, 4x4 8x8, 16x16 and 32x32
	40 m x 40 m	2x2, 4x4 8x8, 16x16, 32x32 and 64x64
	80 m x 80 m	2x2, 4x4 8x8, 16x16, 32x32, 64x64 and 128x128
Elongated rectangle	5 m x 40 m	2x16, 2x32 4x16 and 4x32

Next to the FFT method and the CMD method combined with eigendecomposition the results for the LAS method and CMD method combined with modified Cholesky decomposition are given. This gives some reference for the performance of the FFT and CMD method combined with eigendecomposition. The different errors with respect to accuracy will now be considered.

Representation of the mean value In figure 4.2 the absolute errors in the mean value of all the square random fields are plotted against the number of nodes in x-direction. It can be seen that the deviation in the values decreases when the number of nodes increases. No specific relation is found between the absolute error in the mean and the ratio between the correlation length and the element size of the random field. In this figure no significant differences between the different methods can be observed. If the averaged value of the error in the mean value is checked for every method, also no significant differences are observed. The averaged values are shown in table 4.4.

Tab. 4.4.: Averaged value of absolute error in the value per method.

CMD Eigen	FFT	LAS	CMD Mod. Chol
0.257	0.255	0.251	0.273

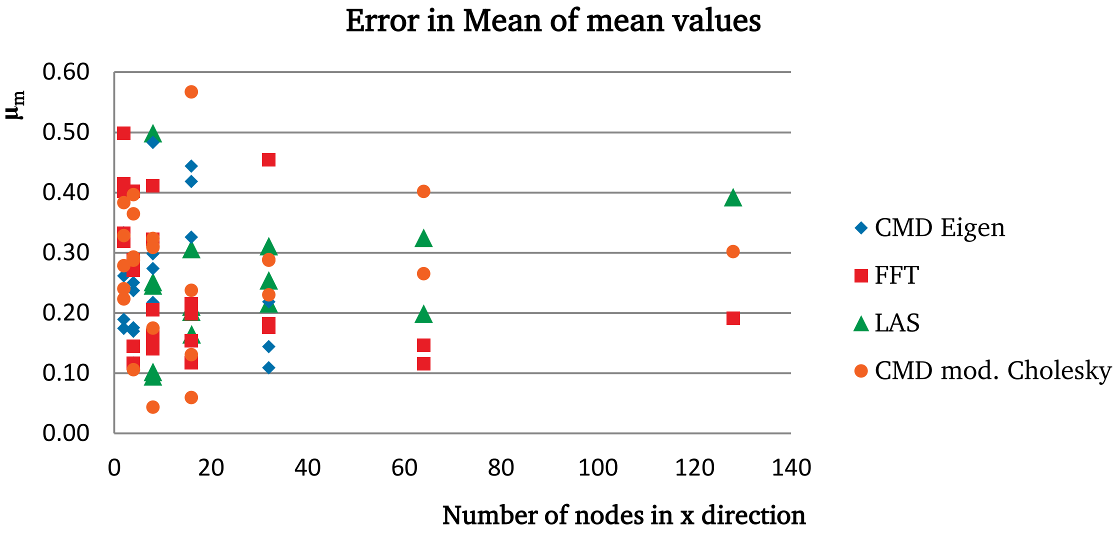


Fig. 4.2.: The absolute error in the mean of the mean values of the random field estimated over 200 fields.

Representation of the standard deviation If the representation of the standard deviations is considered, it can be observed that it is significantly related to the ratio between the correlation length and the element size of the random field mesh. The plot with mean values of the standard deviation is shown in figure 4.3 and the plot with the deviation in the standard deviation is shown in figure 4.4, both for the square shaped random field of 80 m x 80 m. In both plots, it can be observed that the value converges to the target value when the ratio between the correlation length and the element size of the random field increases, i.e. the random field mesh becomes more refined. It can be observed that the CMD method converges faster than the FFT method and the FFT method converges faster then the LAS method to the target value. The target value of the mean value of the standard deviation is smaller than 5.9, which is the standard deviation of the C15 concrete grade. This can be explained by the loss of ergodicity due to the strong correlation in the random field. The ratio between the correlation length and the element size of the random field mesh should be larger than 2 according to these plots to acquire accurate results.

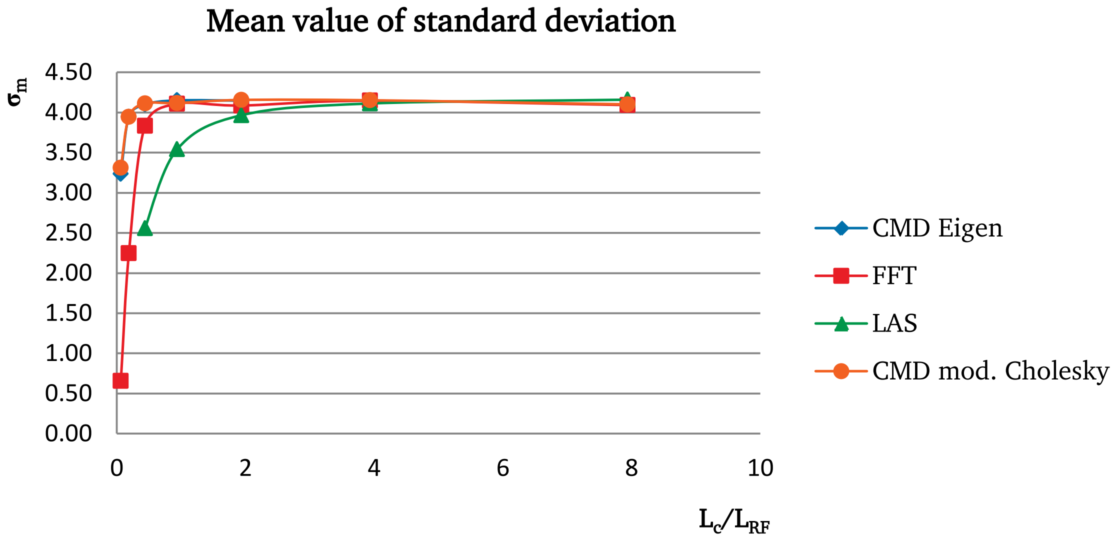


Fig. 4.3.: The mean value of the standard deviation of the random field estimated over 200 fields.

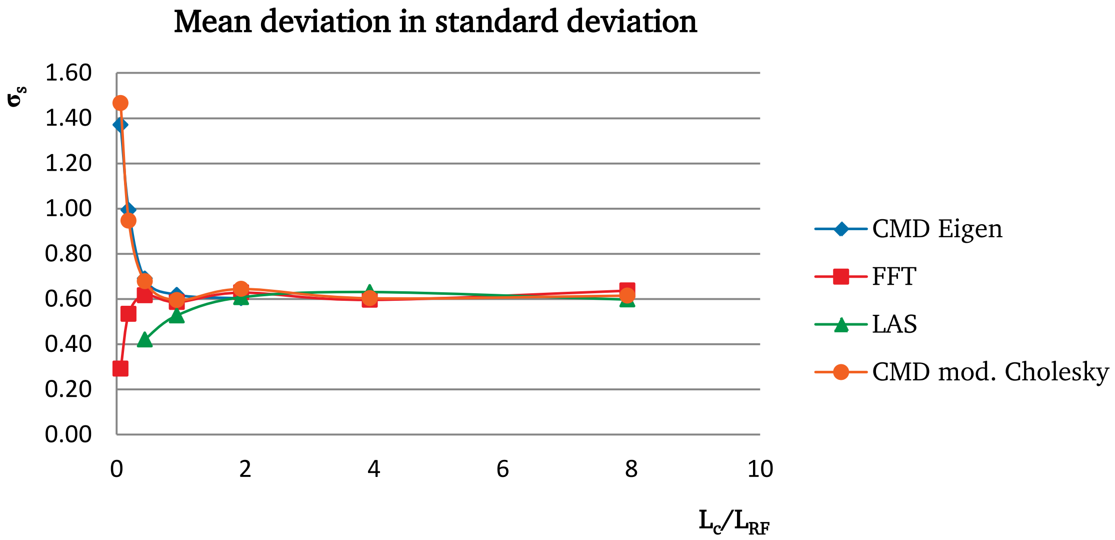


Fig. 4.4.: The deviation in the standard deviation of the random field estimated over 200 fields.

Representation of the correlation structure If the mean value of the absolute error in the correlation structure is considered, no significant relation with the ratio between the correlation length and the element size of the random field mesh can be observed. In figure 4.5 the mean value of the absolute error in the correlation structure is plotted against this ratio. However, it is observed that for a small ratio the values are somewhat higher. If the different methods are compared, it can be observed that the error for the LAS method is higher than for the other methods. The error for the other methods is comparable. The averaged values for the cases that the ratio is between 1.5 and 4 is given in table 4.5 for the different methods. It can be seen that the value for the CMD method combined with eigendecomposition is slightly lower than the value for the FFT method and the CMD method combined with modified Cholesky decomposition. The LAS method has the largest error in the correlation structure.

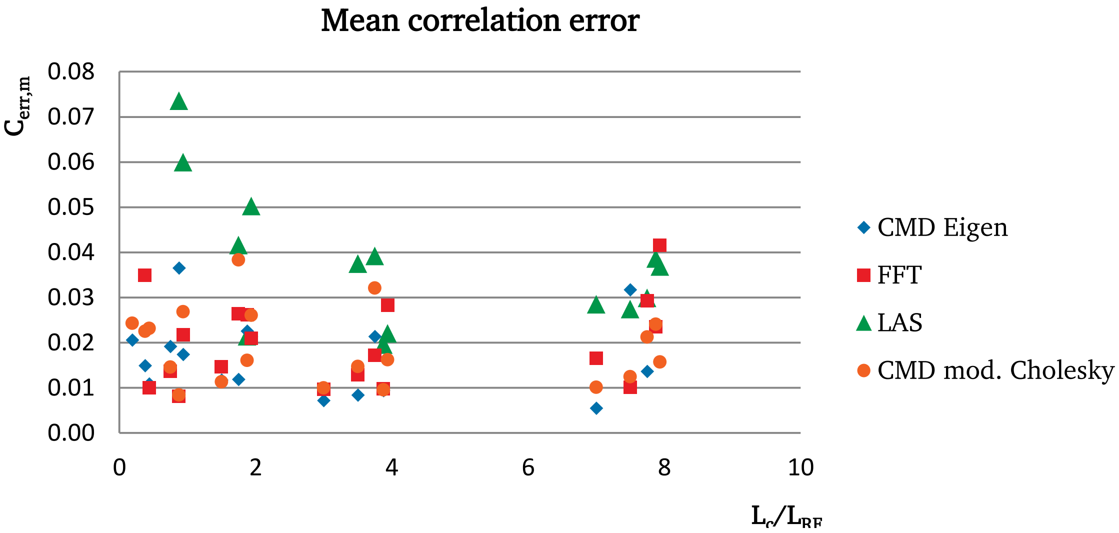


Fig. 4.5.: The mean absolute error in the correlation structure of the random field estimated over 200 fields.

Tab. 4.5.: Averaged mean error in the correlation structure per method for L_c/L_{RF} between 1.5 and 4.

CMD Eigen	FFT	LAS	CMD Mod. Chol
0.014	0.018	0.033	0.019

4.2.3 Overall performance and selection

From the additional comparison it can be concluded that the CMD method combined with eigendecomposition is the most accurate method when the suggested statistical characteristics from the JCSS model code are used. For other concrete types and grades, the COV is even lower which will probably lead to even smaller errors in the representation of the statistical characteristics. The FFT method and the CMD method combined with modified Cholesky decomposition are slightly less accurate and the LAS method is the least accurate method. If the efficiency of the different methods is also taken into account the FFT method outperforms the CMD method. Especially in DIANA, the initialization time is very large for the eigendecomposition. Above a total number of 1032 nodes the computation time strongly increases which lead to impractical situations. An iterative solver has been applied, in order to try to decrease the computation time for the eigendecomposition, without any success. An overview of the total runtime for the CMD combined with eigendecomposition and the FFT method for different number of nodes is given in table 4.6.

Tab. 4.6.: Total runtime in seconds for the generation of 200 fields in 2D.

Number of nodes	CMD Eigen	FFT
4	0.04 s	0.06 s
16	0.06 s	0.13 s
64	0.15 s	0.36 s
256	0.81 s	1.20 s
1024	57.1 s	16.9 s
4096	7384 s	66.7 s

It can be seen that the runtime for the FFT method stay quite small with increasing number of nodes. Although the FFT method is slightly less accurate it is therefore selected for the implementation in DIANA in the JCSS random field application.

4.2.4 Optimal element size

Based on the comparison in this chapter, it can be concluded that the element size of the random field mesh should smaller than half the correlation length. If the results from [16], which were shown in figure 2.15, are taken into account it can be concluded that this ratio is also sufficient when the shape function discretization method is selected. A higher ratio leads to even more accurate results. Please note that in figure 2.15 the inverse ratio is shown on the x-axis. When this inverse ratio decreases, the relative error in the variance decreases. Based on this ratio the number of nodes can be specified for every domain size by ensuring that the ratio is always higher than 2. If the size of the domain is smaller than 5 m in one direction the dimension of the random should be lowered one order. If in all directions the sizes of the domain are smaller than 5 m, a random variable should be applied in stead of a random field since the variation in the field is negligible small.

Tab. 4.7.: Required number of nodes specified for every domain size.

Domain size in meters			Number of nodes
5	-	7.5	4
7.5	-	17.5	8
17.5	-	37.5	16
37.5	-	77.5	32
77.5	-	157.5	64
157.5	-	317.5	128
317.5	-	637.5	256
637.5	-	1277.5	512
1277.5	-	2557.5	1024
2557.5	-	5117.5	2048

4.3 Work flow & in and output of DIANA

In DIANA the JCSS model code is implemented as a material model. In this section the work flow in DIANA for the user is explained. Also some implementation issues will be discussed. Finally, plots for different material properties will be shown, generated with the JCSS material model.

The user can select the *JCSS Probabilistic Model Code* material model for one of the material definitions, which can be assigned to element sets in the program. A total strain rotating crack model with linear softening in tension and a constant curve in compression is selected. The random ultimate compression strain is left out of the implementation. The user can select the concrete type and class which automatically defines the average and standard deviation of the basic compressive strength according to table 4.1. Other material parameters as the Poisson's ratio, the fracture energy and the density can also be defined by the user. The different random field generators, the number of nodes in each direction, the distribution type and correlation function can be specified by the user. Also a constant field can be selected which is appropriate for small models or models where spatial variability is not important.

It is the authors opinion that the number of nodes should be coupled to the size of the element set to which the material definition is assigned, according to table 4.7. If two dimensions are larger than $0.5 \cdot L_c$ a 2D random field should be generated. If only one of the dimensions is larger than $0.5 \cdot L_c$ a 1D random field should be generated. If all dimensions are smaller than $0.5 \cdot L_c$ a constant field depending on a random variable should be generated. In 1D the CMD method with eigendecomposition and in 2D the FFT method should be selected automatically.

In an analysis for every node in the random field mesh, a correlated random variable is evaluated for the basic compressive strength, based on the selected input. For every integration point a value is determined for the basic compressive strength by interpolating between the values of the nodes. The other material properties are then determined for every integration point with equations 4.3 till 4.5.

Examples of the spatial function for the Young's modulus, compression strength and tensile strength in a concrete plate of 25 x 25 m can be found in figure 4.6 till 4.8. It can be seen that the different fields show the same pattern. This is due to the fact that in the JCSS probabilistic model the values are coupled with one single relation (equation 4.3-4.5). The random variable in these relations can be changed in a random field, i.e. for all the material properties spatially correlated random variable are generated for every node in the random field mesh.

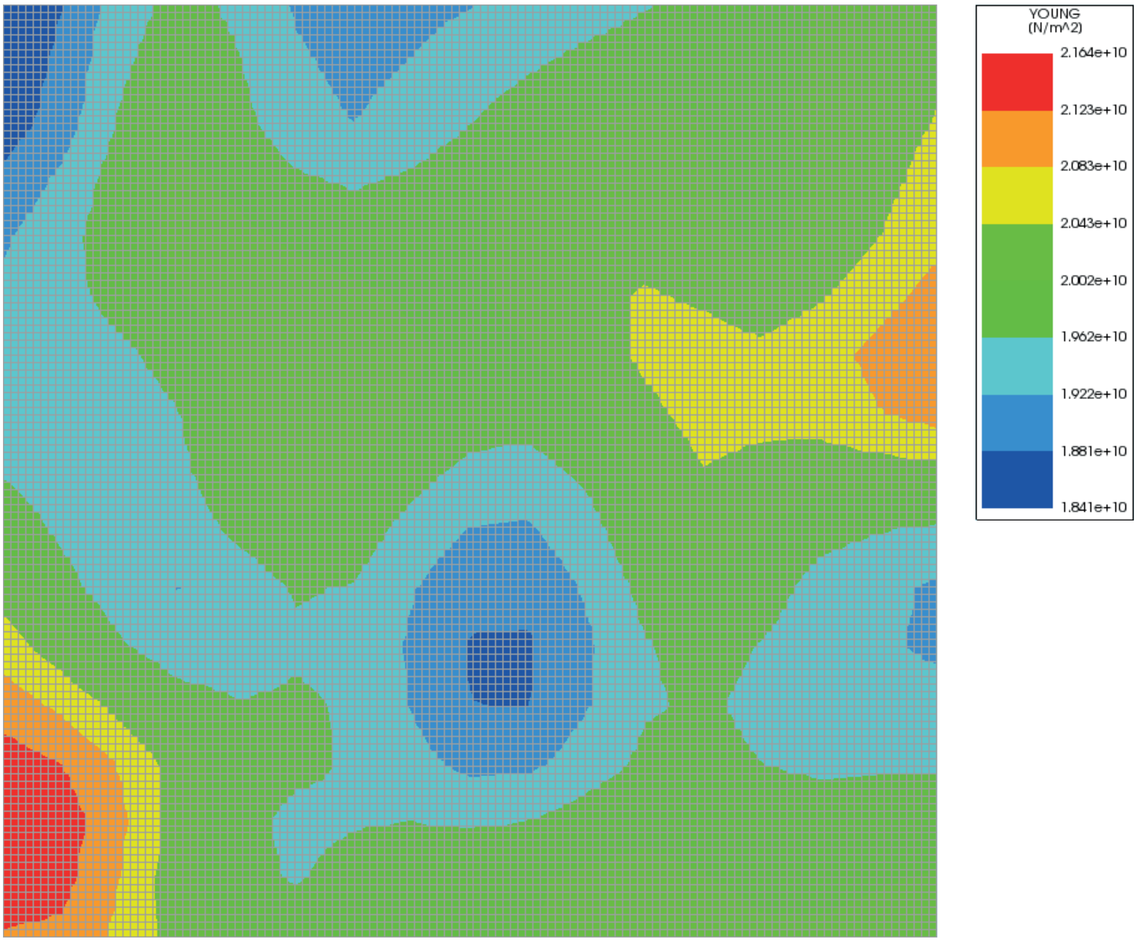


Fig. 4.6.: Example of a spatially varying function for the Young's modulus of concrete.

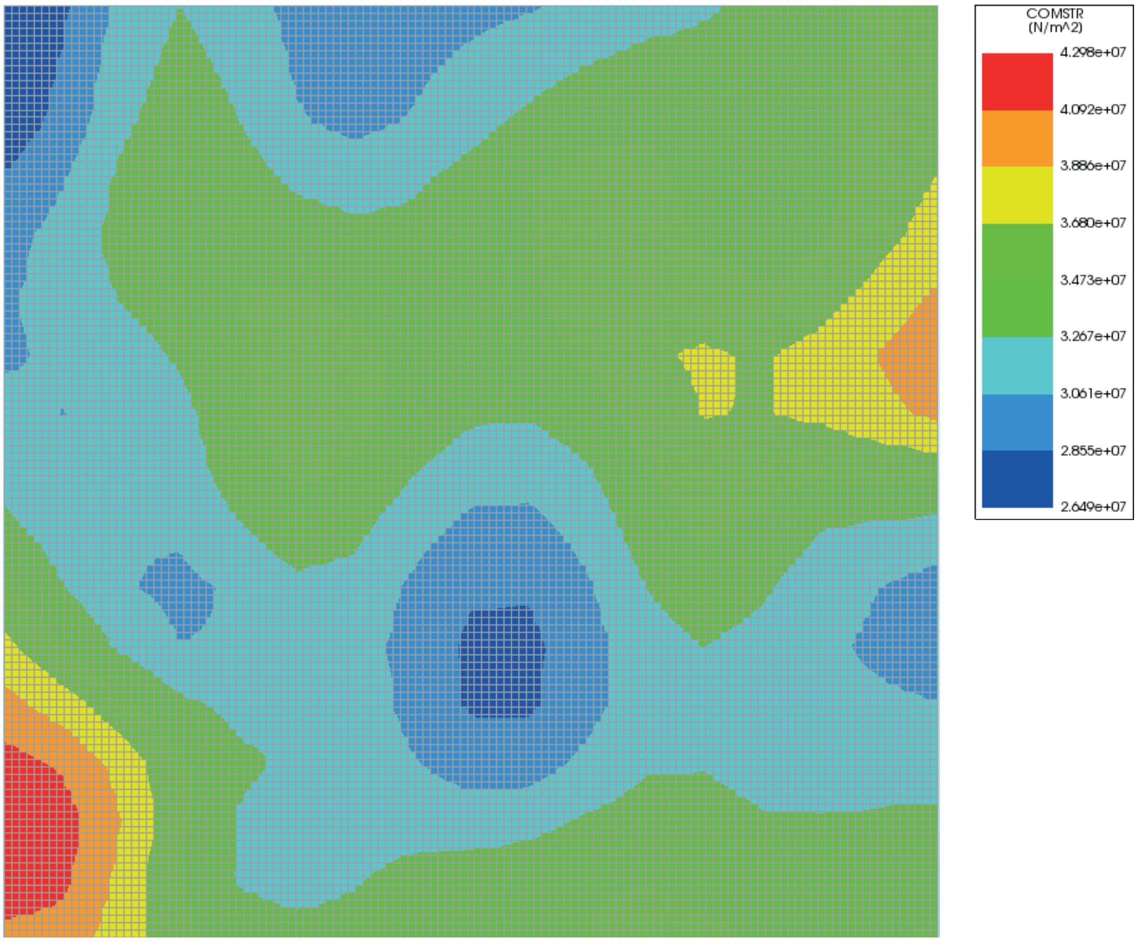


Fig. 4.7.: Example of a spatially varying function for the compressive strength of concrete.

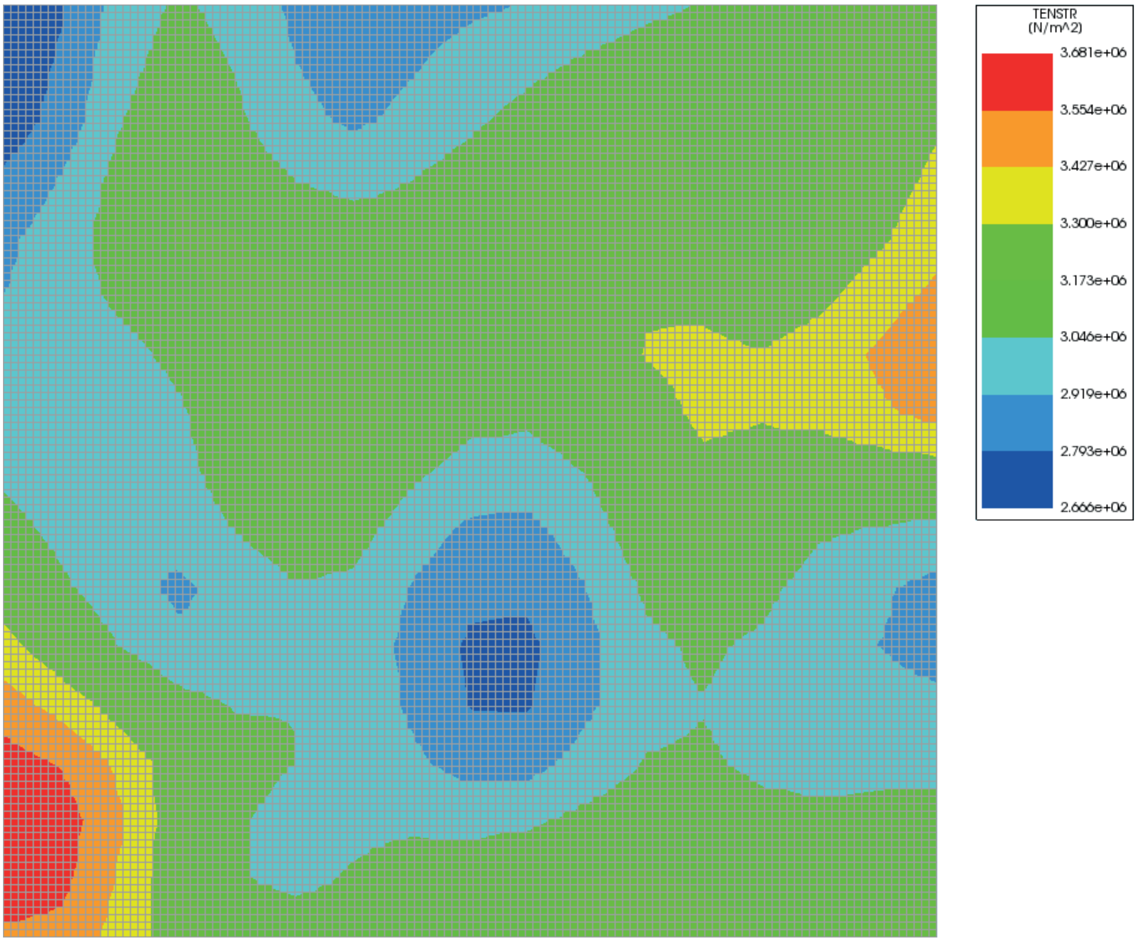


Fig. 4.8.: Example of a spatially varying function for the tensile strength of concrete.

5 Spatially varying material properties in a concrete floor

In most industry halls a large concrete floor is poured. Due to restrained shrinkage, often cracks arise in the floor over time. Shrinkage in a floor can be restrained by its sub-base, piles, columns and perimeter strips. In the example of this chapter, only the restraint of the sub-base will be considered. This problem is typically a non-linear FEM problem where equilibrium is hard to find, since cracking occurs at once over the whole length of the plate, and in both directions. In the load steps in which these cracks arise, highly non-linear behaviour occurs, which is hard to model in a FEA. The analysis will be carried out with and without spatially varying material properties. It is investigated if the analysis becomes more stable when a random field is applied in the material definition. Also the crack initialization and patterns will be compared. It would be very advantageous if the analysis would be numerical stable. In such a case the design of the floor can be made using a FEM program. The thickness and reinforcement ratio can be determined in a economical way and damage due to cracks in the floor can be prevented.

At the end of this chapter the impact on a non-linear analysis of spatially varying properties will be more clear. Also the JCSS Probabilistic Model Code will be evaluated and some recommendation will be given. First a description of the model is given.

5.1 Model description

The reinforced concrete floor is modelled in 3D. The mechanical scheme of the reinforced concrete floor on a linear elastic Winkler bedding submitted to a shrinkage load is given in figure 5.1. The deformation of the floor is restrained by the sub-base which induces stresses in the floor. When the stresses exceed the maximum tensile strength of concrete, cracks will arise.

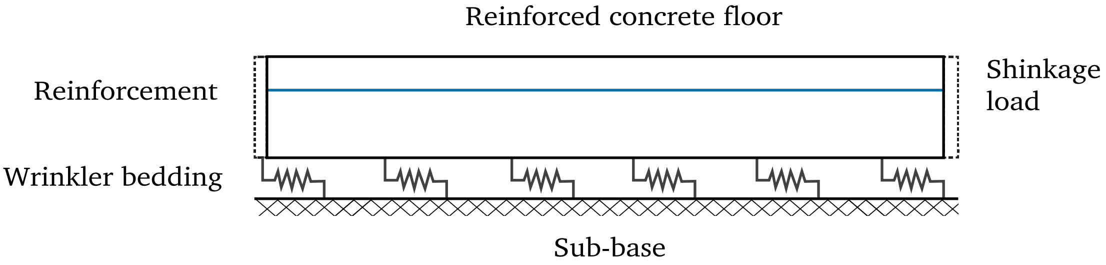


Fig. 5.1.: Mechanical scheme of concrete floor on a linear elastic bedding with reinforcement at 0.05 m from the top.

To model the linear elastic bedding, interface elements (Q24IF) are placed between the supports and the bottom nodes of the floor. The configuration of the floor with the embedded reinforcement, the interface elements and supports is shown in figure 5.2. The nodes below the interface elements are supported in all translation directions. The shear stiffness of the interface element is equal to $10^8 N/m^3$ and in the normal direction it is equal to $10^{14} N/m^3$. Reinforcement is placed in both directions, at 0.05 m from the top of the plate. The orthogonal grid is smeared out and has a equivalent thickness of 0.0008 m. This corresponds to a reinforcement ratio of 0.4%.

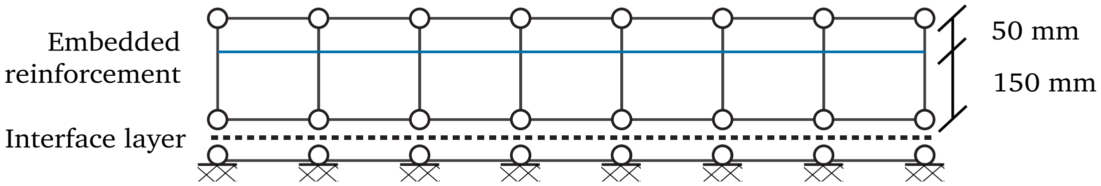


Fig. 5.2.: Configuration of the model with embedded reinforcement, interface elements and the nodes at the sub-base which are supported in all translation directions.

The size of the floor is 25 x 25 x 0.2 m. It consist out of 15625 eight-noded isoparametric solid brick elements (HX24L) of 0.2 x 0.2 x 0.2 m. In figure 5.3 the 3D model is shown. A shrinkage load of 0.4 ‰ is applied on the model. All the brick elements are thus submitted to a prescribed strain of -0.0004.

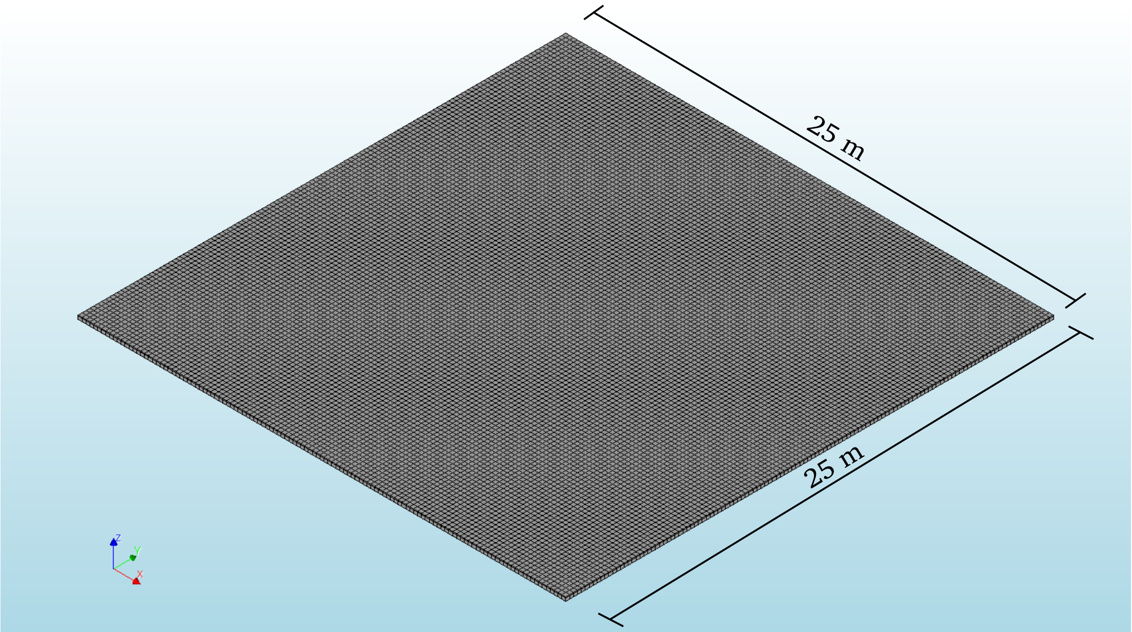


Fig. 5.3.: 3D FE model of a concrete floor on a linear elastic bedding with reinforcement at 0.05 m from the top.

The material properties of concrete with spatially varying properties and concrete with constant properties are given in table 5.1. For the material with spatial variation, the model code of JCSS is followed. The input parameters are given in this table and give all the information required to generate the basic random field. All material parameters required for the material crack model of concrete can be derived from this field for every integration point, as described in chapter 4. For the reinforcement steel, Von misses plasticity is selected as material model with no hardening and also no bond-slip is taken into account. The Young's modulus is equal to $2 \cdot 10^5$ MPa and the yield strength is equal to 500 MPa.

Tab. 5.1.: Material properties of concrete.

	Concrete random field *spatial varying properties*		**Concrete homogeneous** *no variation in properties*
Concrete Class	C35	Concrete Class	C35
Concrete type	Ready mixed	Concrete type	Ready mixed
μ of f_{c0}[1]	47.35 MPa	f_c	32.46 MPa
σ of f_{c0}[1]	5.86 MPa	f_{ct}	3.05 MPa
Distribution type	log-normal	E_c	$1.97 \cdot 10^4$ MPa
RF method	FFT	Poisson ratio	0.15
N_x & N_y	16	Fracture energy	100 N/m
Correlation function	SExp		
Threshold value	0.5		
Correlation length	5 m		
Poisson ratio[2]	0.15		
Fracture energy[2]	100 N/m		

[1] The fields for f_c, f_{ct} and E_c can be derived from the basic compressive strength (f_{c0}) field according to equations 4.3 till 4.5. The variables $Y_{1,j}$, $Y_{2,j}$, $Y_{3,j}$ are kept equal to 1 in every realisation of the random field.

[2] These property are constant in the model where spatially varying materials are used.

Analysis settings In the non-linear FEA, the Secant-Newton BFGS iteration scheme is used in combination with a line-search technique. As convergence criteria a displacement norm and force norm have been selected with a tolerance value of 0.01. If at least one of both criteria is met, the analysis will continue to the next load step. If, after the maximum number of iterations, none of the criteria is met, the analysis is forced to continue at the following load step. In this situation, unreliable results are obtained since the true equilibrium path is not followed. It is however interesting to consider the results because the relative out of balance force and relative displacement variation will give insight in the numerical stability of the analysis. Additionally, it will provide insight in where the crack initialization starts and the crack patterns can be compared. The step sizes and maximum iterations for the analyses are given in table 5.2

Tab. 5.2.: Step sizes and maximum number iterations in iteration scheme.

Step size	**Maximum itterations**
1 x 0.4, 4 x 0.01, 50 x 0.001	5
50 x 0.01	20

5.2 Results

In this section the results for the different models will be given, with respect to numerical stability, crack initialization and crack patterns. For the models that include spatially varying material properties, three analyses are carried out to estimate the deviation in the results. It is important to keep in mind that in none of the analyses the convergence norm is reached in the load steps in which cracking occurs. The true equilibrium path may not be followed, which makes the results unreliable. The results of the analysis will be described in the next sections to give some insight in the influence of spatially varying material properties on the FEA.

5.2.1 Crack initialization

To show how the cracks develop during the analyses, the first principal crack strain is plotted for all the analyses, starting in the load steps where the first cracks arises. In the plots of the subsequent load steps, each with a step size of 0.001, the area with crack strains expands in the concrete floor. These plots are shown in figure 5.4 and 5.5. The load step number and maximum crack strain are shown in the lower right corner of every plot.

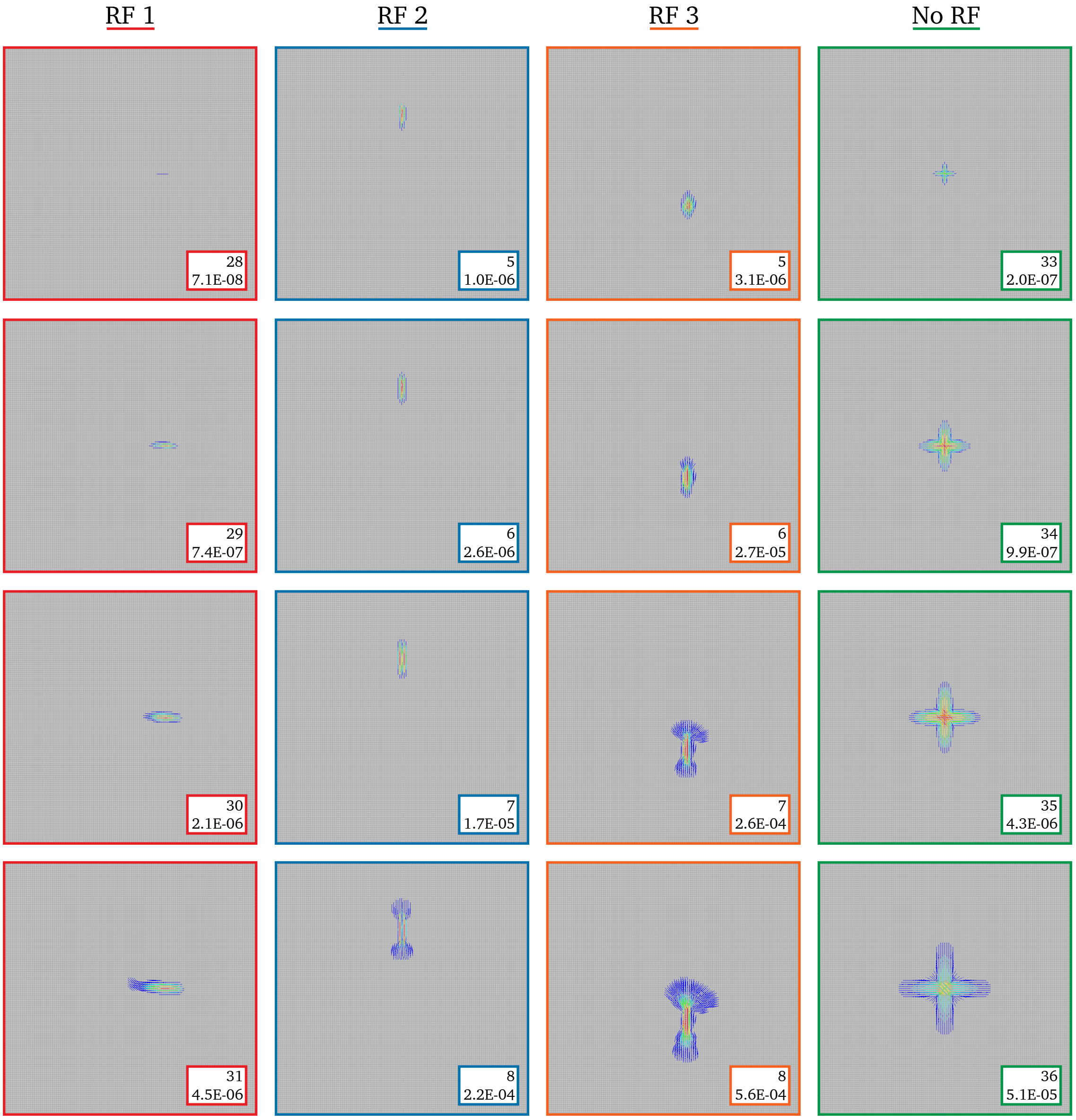


Fig. 5.4.: First principal cracks strains for the loadsteps subsequent to the loadstep where cracking starts. In the lower right corner the load step and maximum crack strain is given which correspond to the red lines in the plots.

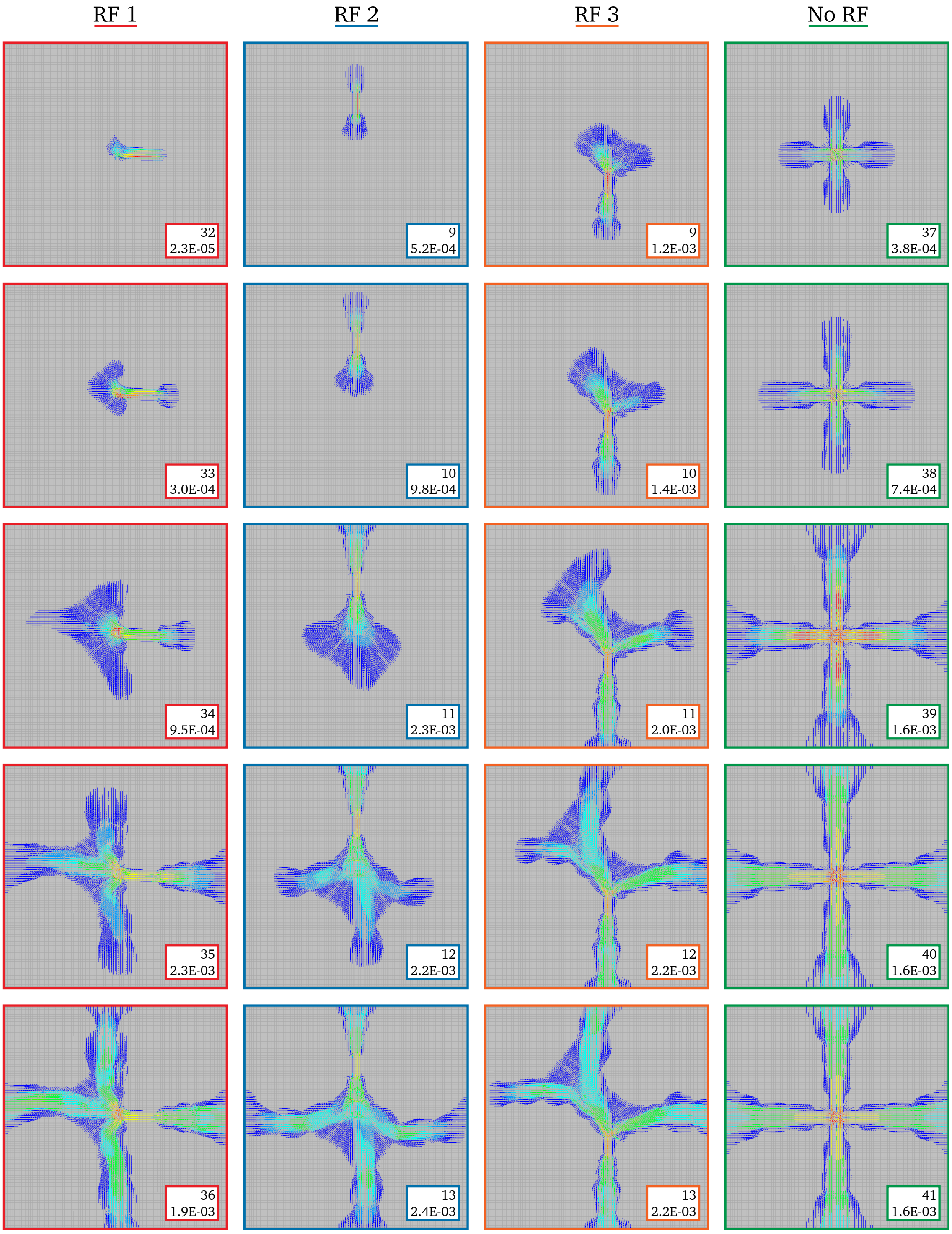


Fig. 5.5.: First principal cracks strains for the subsequent load steps of the last load step in figure 5.4. In the lower right corner the load step and maximum crack strain is given which correspond to the red lines in the plots.

Considering the plots of the crack strains (figure 5.4-5.5) a few observations can be made. It can be seen in every case that the cracks initialize at a single location and develop from bottom to top and from left to right. For the plate without spatial variation, this crack is developed in a lower number of load steps. The load step number where crack initialization takes place is different for every analysis when a random field is applied. The maximum crack strain is somewhat higher in the case spatially varying properties are applied. The (maximum) crack strain increases strongly in the load steps where cracking starts. When the crack has developed from bottom to top and from left to right, the crack strains stay approximately the same. In case no spatial variation is applied the crack starts perfectly in the middle of the floor. The highest stresses occur in the middle of the plate, since the floor is restrained at the bottom by the linear elastic bedding.

To give a better understanding of this behaviour, the shear stress in the interface elements is plotted along the plate for different load factors. This plot is given in figure 5.6. When the plate is uncracked, it can be seen that at both sides of the floor, the shear stresses are the highest. When the concrete floor shrinks, all the nodes have a displacement to the middle of the floor. This displacement is restrained by the bedding which lead to tensile stresses in the floor. These tensile stresses are build up from the side to the middle of the floor, where the tensile stresses are the highest. When the tensile stresses exceed the tensile strength of the concrete a crack arise. On that location a jump in the shear stress can be seen in the graphs. After the floor is cracked in four parts, the same phenomena starts again with a quarter of the plate and secondary cracks arise.

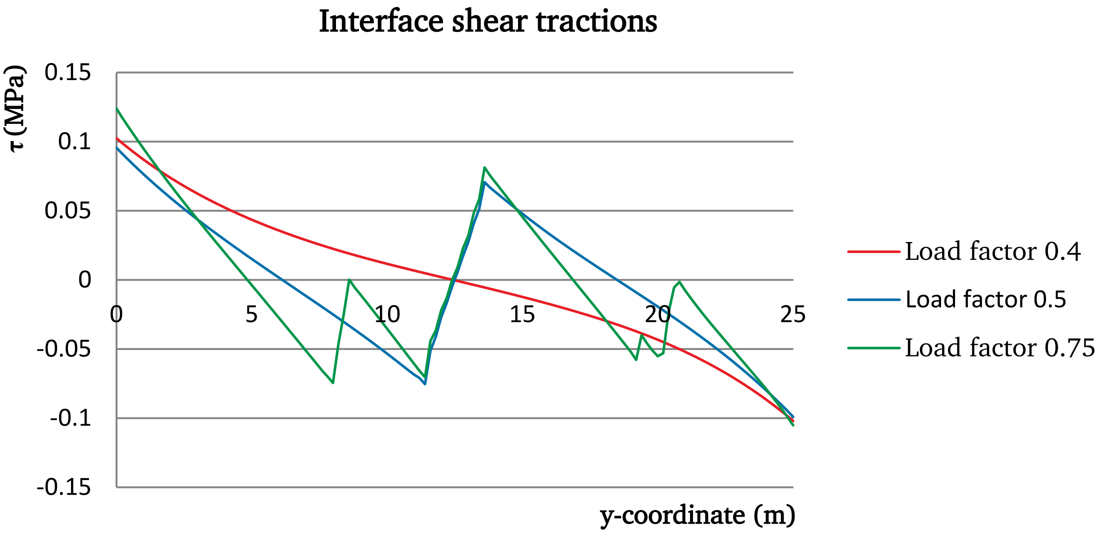


Fig. 5.6.: Shear tractions along the plate at x=20 m for the first analysis of the concrete floor with spatially varying material properties.

For the analysis where a random field is applied, cracking starts somewhere close to the middle. If the tensile strength in the plate, see figure 5.7, is compared to the starting plot in figure 5.4, it can be observed that cracking starts at a weaker point in the structure.

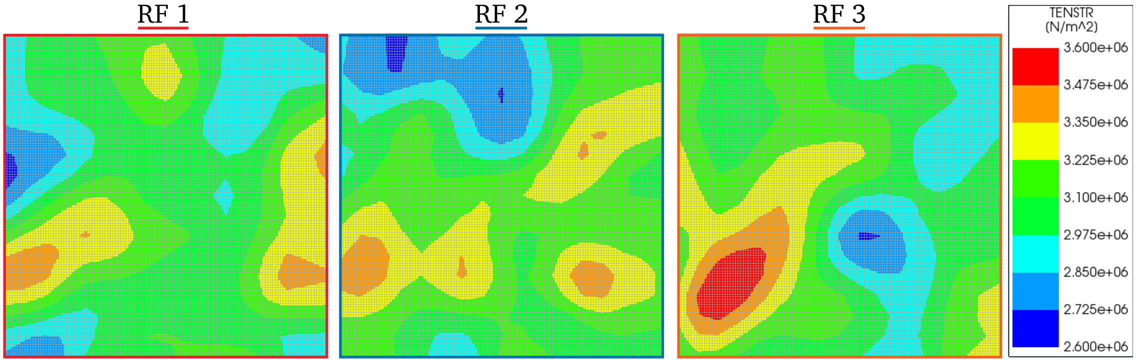


Fig. 5.7.: Tensile strength in the concrete floors for the cases where a random field is applied in the material definition.

In figure 5.8 the number of cracks, as function of the load factor, can be seen. It can be observed that for the analyses with spatial variation in the material properties, the total number of cracks is lower than for the analysis without spatial variation. It can also be seen that around load step 0.43 a large amount of crack occur in all cases. The cracks in the floor develop over the whole length in that step. It can be seen that for the analyses with a spatial variation in the material properties, the cracks occur more gradual than for the one with no spatial variation.

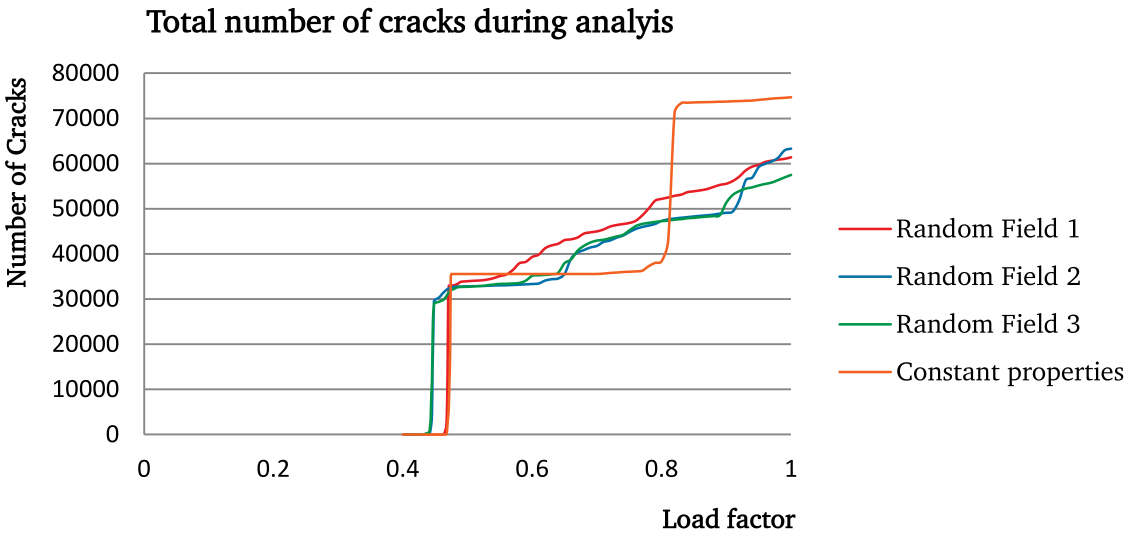


Fig. 5.8.: Number of cracks during the analysis as function of the load factor.

5.2.2 Crack patterns

The patterns in the cracks for the cases where spatially varying material properties are applied, differ somewhat from the case where constant material properties are used. Here, the crack patterns are not perfect symmetric any more. The crack strains for the load factors which correspond to a submitted strain of 0.2, 0.3 and 0.4 ‰ are shown in figure 5.9. Again, it can be observed that the crack patterns are not symmetric, as in the analysis with no spatial variation in the material properties. When spatially varying material properties are applied, it can be observed that the secondary cracks, which develop in both radial direction and tangential direction, grow more gradual. This corresponds with figure 5.8, where it can be seen that the secondary cracks for the case where no spatial variation is applied, develop in just a few load steps.

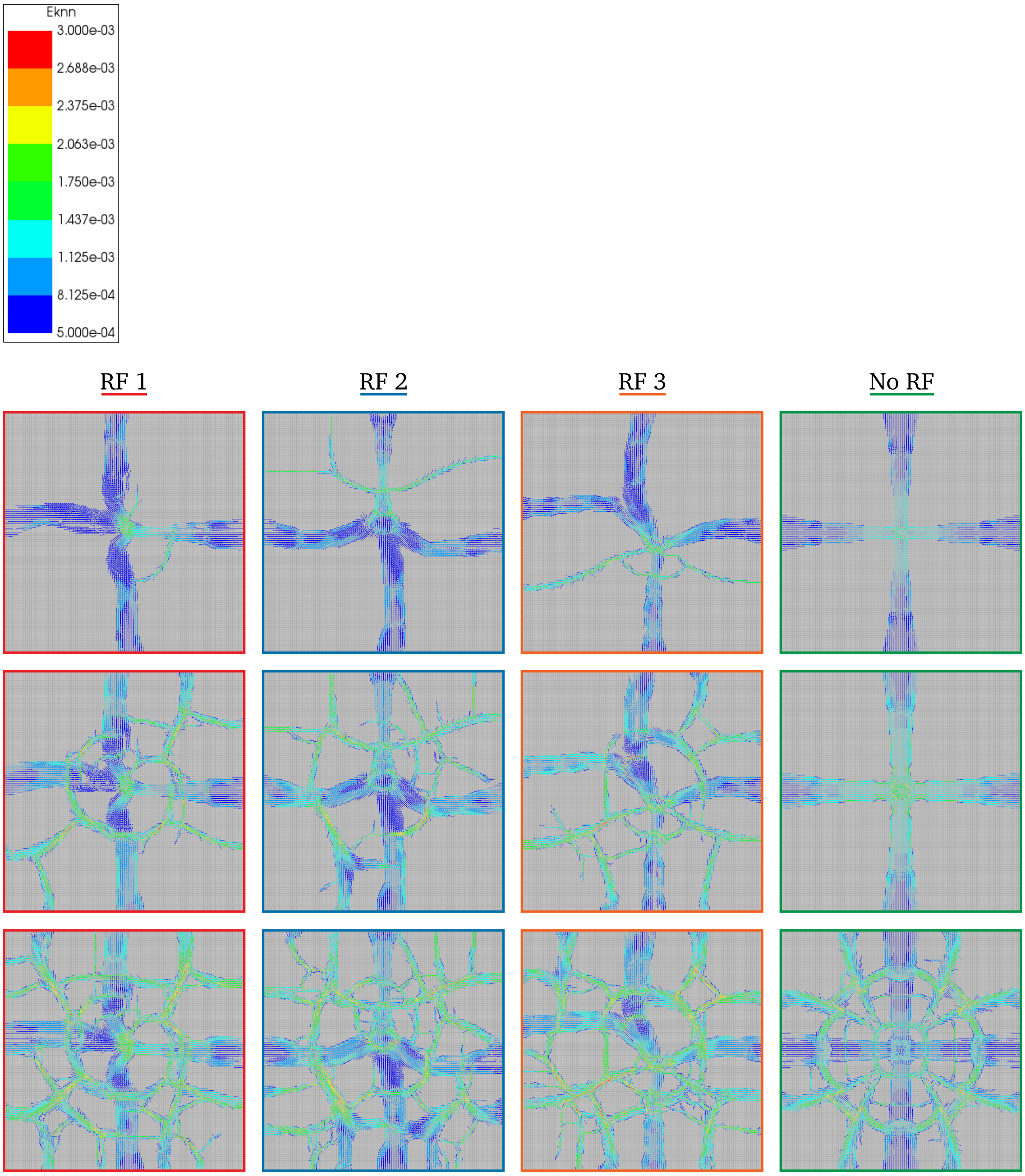


Fig. 5.9.: First principal crack strain corresponding to a submitted strain of 0.2‰ (top), 0.3‰ (middle) and 0.4‰ (bottom). The crack strains give a good impression of the crack patterns in the concrete.

5.2.3 Numerical stability

In both cases, with and without spatial variability, the analysis diverges from its equilibrium path in the load steps where the crack develops from the middle of the floor to the sides. To give some insight in why the analyses do not converge, the relative strain is plotted in figure 5.10. The relative strain is determined by dividing the difference in the displacement of two nodes by the distance between those nodes. A large jump in the relative strain can be observed at the load factor where the crack develops over the length of the plate. This large jump is observed in all the analyses and shows the nature of the problem. A large strain energy is build up before cracking, which leads to an explosive growth of the cracked area, after the initialization of the crack.

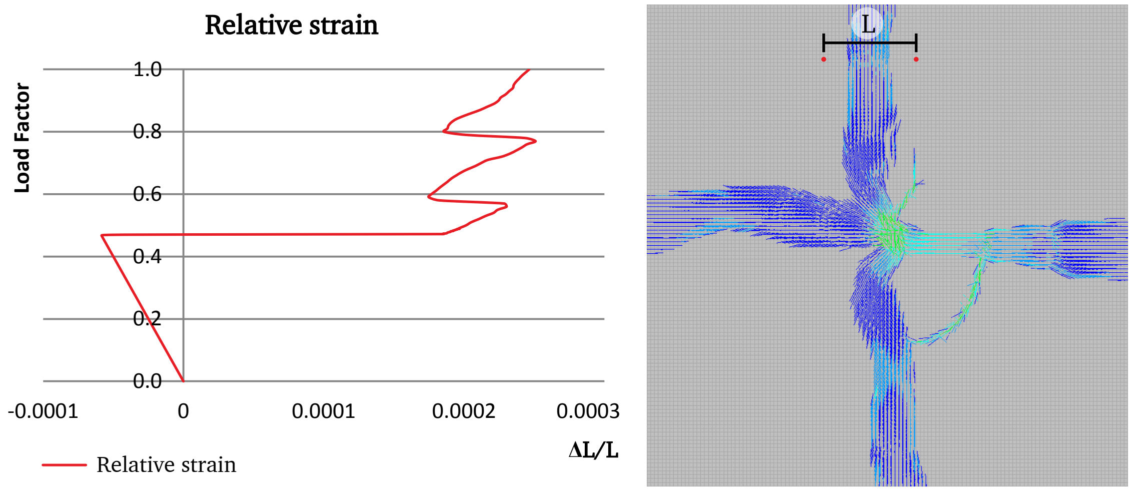


Fig. 5.10.: Relative crack strain as function of the load factor. Relative strain is determined by dividing the difference in the displacement of two points by the distance between those points.

In all the analyses the convergence norm is reached in the first two or three load steps after crack initialization takes place. Thereafter strong deviations from the convergence norm are observed. The maximum number of iterations is therefore limited to 5 in order to limit the divergence from the true equilibrium path. Since all the analyses diverge from the true equilibrium path, it is hard to say something strict about the numerical stability of the analysis. Though, it can be observed that the number of cracks grows more gradual when spatially varying material properties are applied, especially when secondary cracks are developing. Also, the maximum relative out of balance force, see table 5.3, is somewhat larger in the case where there is no spatial variation in the material properties. Both observations are indications that with spatially varying material properties the analysis is somewhat more stable than without spatially varying material properties.

Tab. 5.3.: Maximum relative out of balance force.

RF 1	RF 2	RF 3	No RF
1.17	0.90	1.00	1.15

5.3 Variations on the JCSS Probabilistic Model Code

In order to get a better picture of what the influence of spatially varying material properties is on a non-linear FEA, a few additional analyses are carried out. Up till now the analyses did not reached the convergence norm in the load steps after the first cracks occur. Additionally, the cracks do not localize, the band of cracks spreads out over several elements. The variations on the JCSS Probabilistic model code may not be realistic but are all aiming on stimulating crack localization, which may lead to a more stable analysis. In this section only the results which differ from the previous analysis will be presented.

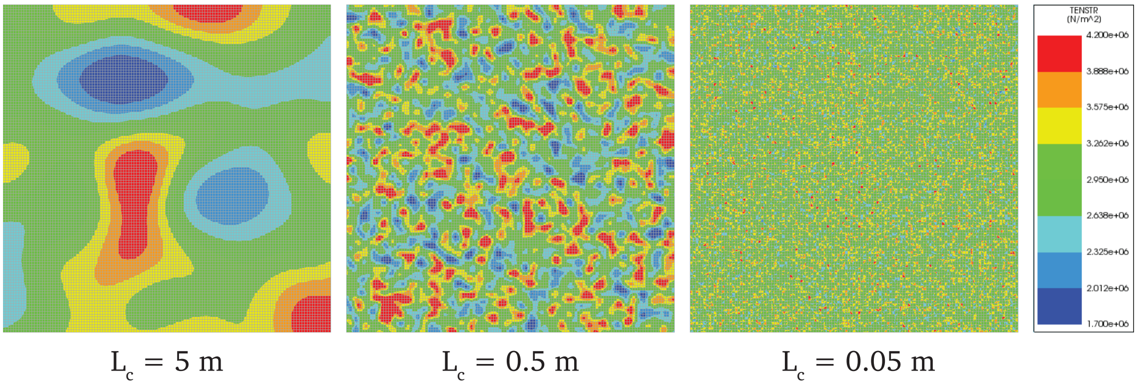
5.3.1 Increasing the variation in the field

In the JCSS Probabilistic Model Code a quite large correlation length is used, and the COV of 0.12 is quite low in comparison with the used values found in the review of literature. Additionally, a threshold value is used which is not used in other approaches. In this section the analyses are presented where the variation in the values of the field are increased by lowering the correlation length, increasing the COV and not applying a threshold value. The used input parameters are given in table 5.4. The used values will result in fields for the material parameters with a more coarse surface and with larger fluctuations. The extreme case, where a correlation length of 0.05 m is used, corresponds to the case where the values in the random field are uncorrelated. Every integration point is assigned a value which is uncorrelated with other values. Due to the increase in the variation in the fields, the load factor where cracking arises is lower than in the previous case. The used sizes for the load steps in the analyses are also given in table 5.4. Again, none of the analyses reached the convergence norm in all the load steps.

Tab. 5.4.: Used L_c and COV in additional analyses.

Analysis number	L_c	COV	Load steps
Analysis 1	0.5 m	0.3	1x 0.30, 1x 0.01, 50 x 0.001 and 64 x 0.01
Analysis 2	0.5 m	0.5	1x 0.27, 1x 0.01, 50 x 0.001 and 67 x 0.01
Analysis 3	0.05 m	0.3	1x 0.30, 1x 0.01, 50 x 0.001 and 64 x 0.01
Analysis 4	0.05 m	0.5	1x 0.30, 1x 0.01, 50 x 0.001 and 64 x 0.01

In figure 5.11, the plots of the tensile strength for different correlation lengths is given. For all cases, the COV is equal to 0.3 and no threshold value is used in the correlation function. It can be seen that the fluctuation is much more gradual when the correlation length is higher. This has a significant effect on the analysis as will be shown.



L_c = 5 m

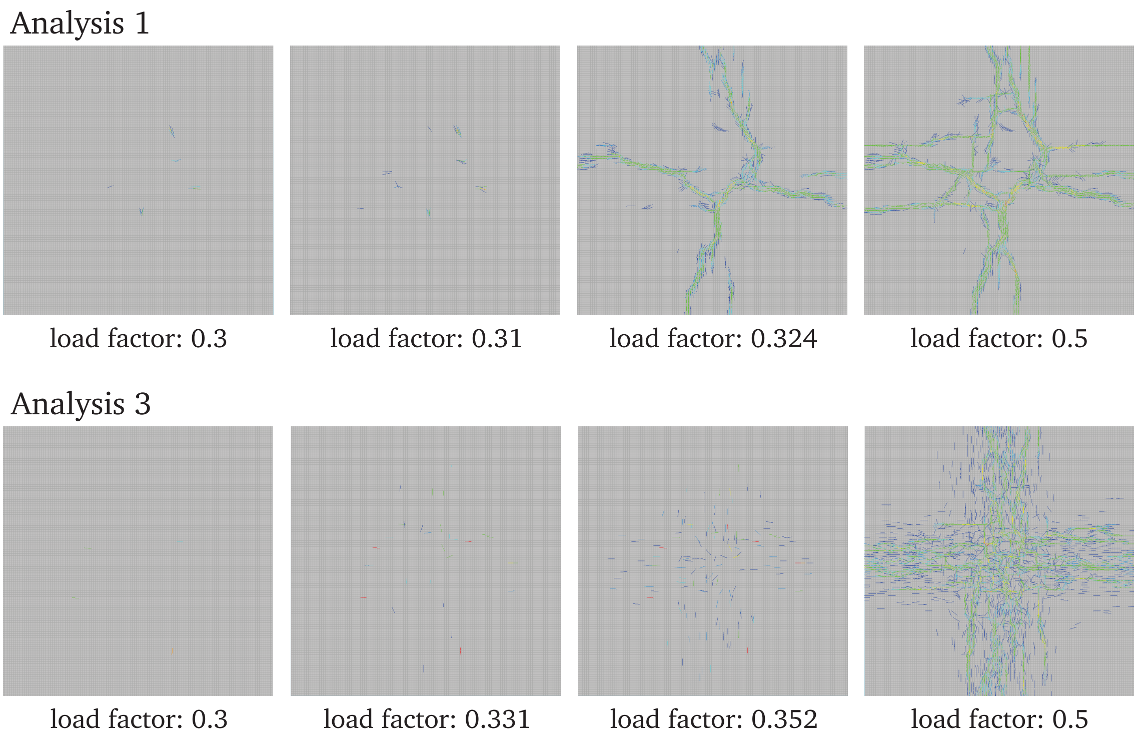
L_c = 0.5 m

L_c = 0.05 m

Fig. 5.11.: Tensile strength in concrete floor with different values for correlation length. COV is equal to 0.3 and c_1 is equal to zero.

In figure 5.12 the crack strains are plotted for analysis 1 and analysis 3 for several load steps. It can be observed that the cracks arise on more locations as the correlation length decreases. The cracks develop on different locations and grow to each other. As in the previous example, again a cross pattern arises. In the case where a correlation length of 0.5 m is used, the crack band is somewhat smaller than in the case of a correlation length of 5 m. This could be an indication for a more stable analysis. If the correlation length decreases further, the number of integration points which are in the cracked state is much larger than for the other cases.

When the COV is increased to 0.5, the crack patterns are comparable to the analysis where a COV of 0.3 is used, and are therefore not shown. On average, the load factor where cracking starts is lower in the analyses with a COV of 0.5. The crack band is approximately the same. From these observations it can be concluded that especially the correlation length has a significant influence on the crack patterns.



Analysis 1

load factor: 0.3 load factor: 0.31 load factor: 0.324 load factor: 0.5

Analysis 3

load factor: 0.3 load factor: 0.331 load factor: 0.352 load factor: 0.5

Fig. 5.12.: Crack strains for analysis 1 and analysis 3 for several load steps.

5.3.2 Homogeneous Young's modulus

In chapter 4, it was stated that in the JCSS Probabilistic Model Code, the different material parameters are fully correlated. This can also be seen in the figures 4.6 till 4.8. In these figures, the maxima and minima are on the same locations in the field for every material parameter. Using the expression in the JCSS Probabilistic Model Code, the strain at which cracking occurs can be expressed as follows:

$$\varepsilon_{crack} = \frac{f_{ct}}{E_c} = \frac{0.3 f_c^{2/3} Y_2}{10.5 f_c^{1/3} Y_3 (1 + \beta_d \phi(t,\tau))^{-1}} = \frac{0.3}{10.5} f_c^{1/3} (1 + \beta_d \phi(t,\tau)) \tag{5.1}$$

The variables Y_2 and Y_3 are set equal to 1 in all analyses for every realization of a random field. The tensile strain at which cracking occurs is thus a function of the compression strength only. This function is plotted in figure 5.13. It can be observed that for the concrete class C35 the tensile strain, at which cracking occurs, varies only little. This can be one of the reasons that, although spatially varying material properties are used, the cracks do not localize.

To increase the variation in the tensile strain at which cracking occurs, the analysis is carried out with a constant Young's modulus. This again led to divergence from the true equilibrium path. The primary cracks again occur in just a few load steps. The crack patterns where comparable with the case where the Young's modulus is varied. It also gave no other development of the total number of cracks in the model.

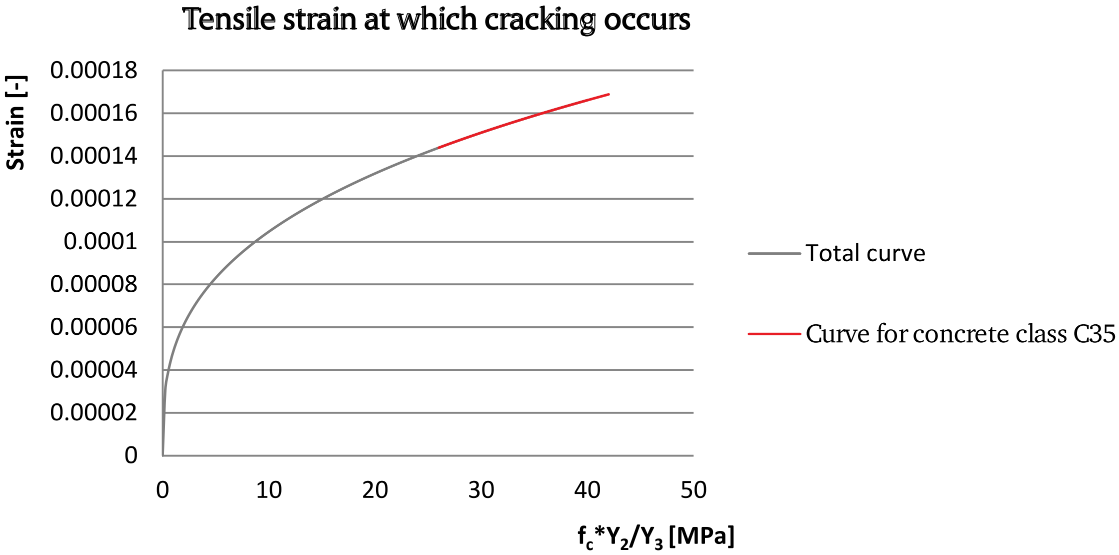


Fig. 5.13.: The strain at which cracking occurs as function of the compression strength.

5.4 Evaluation

In this chapter a concrete floor which is submitted to shrinkage is considered. The crack patterns, occurring in the analyses, are as expected. Based on examples in reality, it was expected that the floor would crack in four parts due to restrained shrinkage. Based on the presented results it can not be stated if the FEA becomes more stable when a random field is applied, since in all cases the analysis diverges from the true equilibrium path. The iteration scheme and step sizes have been varied to find a converging solution without any success. After the first cracks arise, the crack develops in both directions over the full length of the floor in a few load steps. This explosive growth of cracks is hard to model with the FEM. Arc length control would be a good method to improve the numerical stability. Unfortunately, it can not be applied since the strain load is defined such that it is not included in the external load vector. This leads to a division by zero in the arc length control method.

The Young's modulus is kept constant in one of the analysis to increase the variation in the tensile strain at which cracking occurs. This gave no improvement in the numerical stability of the analysis. From this observation, it can be reasoned that if instead of random variables, random fields would be used for the variables Y_1 to Y_3, such a modification will give no improvement in the numerical stability of the analysis. The cracking occurs very explosive, which is not different when spatially varying material properties are used. For other highly non-linear problems it could be that the use of a random field would improve the numerical stability of the analysis.

With regard to the load step selection some remarks can be made. In case spatially varying material properties are applied, it is not known a priori at which load factor non-linear behaviour starts occurring. This leads to a more practical problem. The zone over which smaller load steps are taken, have to be larger in order to be sure that the transition of linear to non-linear behaviour is in one of the smaller load steps. This problem may be omitted by doing first a linear analysis to estimate the load factor where non-linear behaviour starts occurring and thereafter doing a non-linear analysis using the same random field for the model. It is the authors opinion that more models should be examined using a random field to explore other practical issues which may not be advantageous when a MCS or other reliability analysis is carried out.

The material model which have been developed according to the JCSS Probabilistic model Code can also be improved in order to stimulate localization of cracks. In the model code, nothing is mentioned about the fracture energy. This value is therefore kept as a constant in the analysis. In figure 5.14, the tension-softening stress-strain diagram is given, which corresponds to the material model according JCSS. In the left diagram, the relation between the fracture energy G_f and other parameters is given. The parameter h is the crack band width over which the crack is smeared out. In the right diagram, two softening curves for two different integration points are given, which each have a different Young's modulus and tensile strength. They have the same value for the fracture energy which results in a different slope for the softening part of the diagram. During an analysis, the integration points with a higher tensile strength will have a stiffer response, when they are in the cracked stage, than the integration points with a lower tensile strength. It could be the case that in a load step the integration point with the lowest tensile strength will be in the cracked state because the maximum tensile stress is exceeded. In a next step, the stress in another integration point, close to the first integration point which is in the cracked state, may also exceed the tensile strength and will be in the cracked state. Due to the stiffer response of the integration point with a higher tensile strength, it will attract more of the deformation energy. This will lead to a wider area of integration points which are in the cracked state.

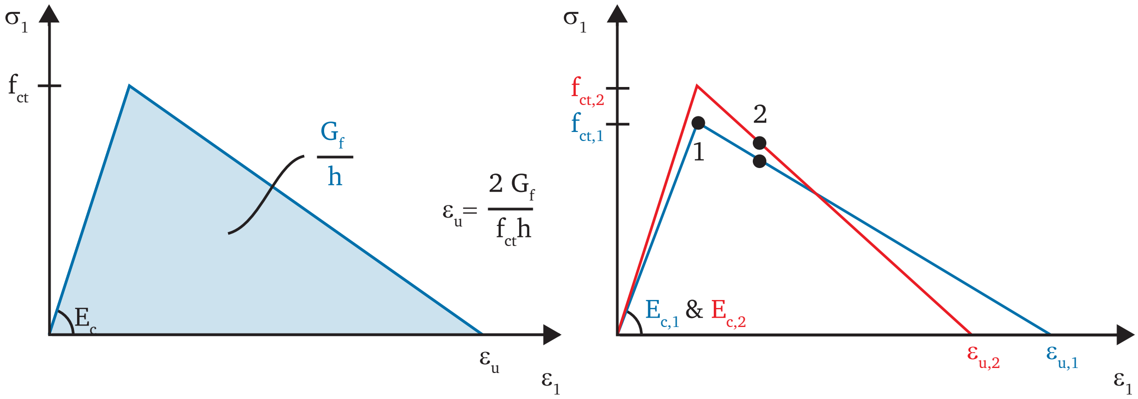


Fig. 5.14.: Left: Tension-softening stress-strain diagram with indication of different parameters. Right: Two softening curves with differnt values for E_c and f_{ct} and the same value for G_f.

From a practical point of view, it would also be better to vary the fracture energy based on the tensile strength and the Young's modulus. In the analysis with a COV of 0.5, the program gave an error that the tensile strength had to be reduced. The softening curve would other wise be in the direction of a strain which is lower than the strain belonging to the tensile strain at which cracking occurs. For a high tensile strain, the fracture energy should also be higher to avoid this problem.

Next to the fracture energy, another remark can be made with regard to the JCSS Probabilistic Model Code. The average value for the Young's modulus is around 20 GPa for a concrete class of C35. This is value is quite low in comparison to the value used in the Eurocode for that concrete class. The value in the JCSS Probabilistic Model Code is equal to the value which is used for cracked concrete. It is therefore not clear if this code can be applied for every problem with reinforced concrete.

5.5 Conclusion

A reinforce concrete floor was modelled on a linear bedding with and without spatially varying concrete material properties. Crack patterns are as expected, but can not be relied on since the analyses did not converge in the load steps after the first cracks occurred in the floor. Based on the results of the different analyses, it can not be stated if the application of a random field in a non-linear FEA would be beneficial for the numerical stability. The more gradual grow in the total number of cracks and the somewhat higher convergence norm may be indications that this is the case.

From the plot of the relative strain around a crack, it can be concluded that a large strain energy is build up before cracking occurs. This results in an explosive growth of cracks after the first crack arises. With and without spatially varying material properties this is the case. The cracks did not localise in a single element but where spread out over several elements. To stimulate localization, which may lead to more numerical stable results, several variations on the JCSS Probabilistic Model Code have been applied in an analysis. The correlation length and COV have been varied to increase the variation in the field. With a correlation length of 0.5 m the cracks localised somewhat more than in the other cases. The analysis did however still not converge to the convergence norm. When the correlation length is decreased further, the cracks arise on a large number of locations in the floor. This leads to a very broad area with cracks which is not beneficial for the analysis with respect to numerical stability. The increase of the COV did not had an influence on the pattern in the cracks. Only the load factor from which cracking occurs became lower. By taking the Young's modulus constant, it was explored if it would help if the different material parameters would not be fully correlated to each other. This gave no improvement with regard to the numerical stability of the analysis.

Discussion 6

In this report the research question, raised in the introduction, is tried to be answered with literature study, comparative studies and the examination of an example. In this section the results in this research is placed in perspective and is discussed.

Initially, the focus was put on the probabilistic aspect of using a random field in a non-linear FEA. During the research it turned out that possible problems with numerical stability of the analysis should be explored first before doing a full probabilistic analysis. Finally, the research resulted in an application in the general purpose FEM program DIANA, where spatial variation can be assigned to any material property. The application is integrated in the work flow of the program. In literature and practise only examples of external modules can be found which can be combined with existing FEM software. The research gives an approach which can be followed by other scholars and companies to find out which method is the most appropriate to incorporate spatial variability in a FEA. The same approach could be followed for other unexplored materials such as masonry or timber. Most scholars assume that random fields, used in their research, are ergodic with respect to the mean and standard deviation. In this research it is showed that this assumption is only valid in a limited number of cases.

6.1 Assesment of random field generators

The assessment, carried out in chapter 3, gives a good insight in the influence of the different statistical characteristics on the accuracy and the efficiency of a random field generator. It also gives insight in what method performs best, when specific statistical characteristics for the concrete material properties are selected. When a log-normal distribution with a COV between 0.1 and 0.3 is selected, the assessment gives no clear conclusion on what method to use. In order to deal with this problem, an extra comparison could be made with a log-normal distribution type and a COV between 0.1 and 0.3. In chapter 4 this extra comparison is carried out only for the correlation function which is suggested in the JCSS Probabilistic Model Code to see which method is the most appropriate.

In the assessment the mean and standard deviation are considered per random field. It would also be very interesting to consider those values per node in the random field mesh, estimated over a large number of random fields. In this case, it can be checked if the method is accurate in the representation of the mean and the standard deviation. It could be the case that in this comparison one of the methods would outperform the others. This comparison is already carried out with values for the statistical characteristics which are used for soil material properties [14].

The full potential of class 2 methods is not yet considered. In order to do this, another set-up in the definition of the spatial function should be defined in the DIANA program. When these methods are assessed it could turn out that one of them outperforms the CMD or FFT method. At a glance, these methods seem quite hard to implement in a general purpose program.

The focus of the assessment of random field generators in this research is put on the accuracy of the random field generators. For a MCS reliability assessment this is very important, since the number of samples can be lower with an accurate random field generator. For a non-linear crack analysis where it is not clear where the crack starts, for example the concrete floor, a random field can be favourable for the numerical stability. In such a case the accuracy of the random field is of less importance.

6.2 Probabilistic Model Code JCSS

In this research the Probabilistic Model Code of the JCSS was selected to implemented as a material model in the FEM program DIANA. Its applicability is assessed by applying it in a non-linear FE example, namely a concrete floor submitted to a shrinkage load. The model code is selected because it is the only guideline which gives an integral overview on how to take into account the spatial variability in concrete material properties. Also no other examples where the spatial variability of concrete parameters is taken into account according to the JCSS Probabilistic Model Code were found in literature. The used correlation functions entails a relative large correlation length and a threshold value of 0.5. Both are different from most of the other approaches found in literature. It is not clear on what experimental data the used correlation function or other parameters are based. The suggested values in the JCSS model code are therefore open for discussion in future research. A large correlation length is nonetheless advantageous for the efficiency of the random field methods since the random field mesh can be more coarse, which means less random variables are needed to acquire an accurate random field. More experiments should be carried out to find the appropriate values for the statistical characteristics, which are involved in the modelling of spatial variation in concrete material properties. In the review of literature a large variation was found in these values. The determination of the correct values could be very hard in practise, especially for the correlation function this is difficult. First of all a lot of experiments have to be carried out to have enough data on which the correlation function can be based. Next to that, it could be the case that the correlation length depends on several factors, of which each has its own scale of fluctuation. For example the fluctuation can be influenced by the way the concrete is poured, the shape of the mould and the used aggregate sizes. In that case the correlation function would be dependent on several factors and would thus be different for every problem considered.

Due to the high correlation length and the use of a threshold value, the variance in a single field reduces significantly. It is questionable if this reduction of the variance is significantly large in reality too. If this is not the case this could be an indication that the advised correlation function in the model code is not correct. A smaller correlation length or/and no threshold value should be used in that case.

The JCSS model code is implemented with the variables $Y_{1,j}$, $Y_{2,j}$, $Y_{3,j}$ and $Y_{4,j}$ being a random variable. It should be more appropriate to apply a random field for those variables. In that case, the different material parameters are not fully correlated any more, which would be more realistic. Also, the fracture energy should variate in the model to improve the material model in DIANA. However, in the model code no suggestions are given for the fracture energy. The fracture energy could be made dependent on the tensile strength and Young's modulus, or a random field can be generated for the fracture energy which is correlated to the other material parameters. This may stimulate crack localization, which may increase the numerical stability of the analysis.

The average value for the Young's modulus in the JCSS model code is equal to the one of cracked concrete, which is lower than the Young's modulus of uncracked concrete. It is therefore questionable for what kind of analysis the model code is applicable.

6.3 Non-linear analysis and random fields

The concrete floor which is submitted to a shrinkage load is modelled on a linear bedding. In reality the connection between the floor and the sub-base will fail at a certain shear stress. It would therefore be more appropriate to model the concrete floor on a non-linear bedding with a τ_{max} at which the shear stress in the interface element is limited to the maximum friction shear stress. This will result in more realistic behaviour of the model. This will enlarge the area over which the maximum tensile stress is build up in the centre of the plate, and cracks will occur when the floor is submitted to a higher shrinkage. However, because of practical considerations it is decided to model floor with a linear bedding. By keeping the bedding linear it was intended to let the analysis converge to the convergence norm in the load steps where cracks arise. Another reason is that the floor should be larger to get similar crack patterns which would increase the CPU-time of the analysis strongly.

The model is modelled with 3D linear brick elements. It would also be sufficient to model the problem with membrane elements, since no significant 3D effect where observed in the results of the analyses. To increase the numerical stability of the analysis, membrane elements with a quadratic interpolation and Gauss integration scheme could be selected. The variability in the material properties could be captured better in that case.

Based on this single example no conclusion can be drawn in general. More applications and cases should be considered to explore the difficulties which are faced when a random field is applied in combination with a non-linear analysis of reinforced concrete. The observation with regard to this aspect are however valuable for further research.

Conclusion and recommendations 7

7.1 General research findings

Concerning the literature review on random field generation for reinforced structures, which was reported in 2.7.3, the following conclusions can be drawn:

1. A wide variety is found in literature for the statistical characteristics, which are involved in the modelling of spatial variability. Both the Exp and SExp correlation function are suggested with correlation lengths varying from 0.5 m to 5 m. As distribution type the normal, truncated normal and log-normal distribution are used and the COV varies from 0.1 to 0.3. For the deterministic relations between the different material properties, to determine the mean value, often the CEB-FIB 90 model code is used. The wide variety in the statistical characteristics can be mainly attributed to the lack of experimental data on which the appropriate values can be based.
2. The JCSS Probabilistic Model Code gives a complete guideline for the modelling of spatial variability in the material properties of concrete. In this guideline, a threshold value for the correlation function is introduced to take into account the fact that within one structural element the properties are correlated with a minimum value, since it is poured with one batch of concrete.
3. The number of dimensions in which the random field should be generated, should be equal to the number of dimensions of the structure which are larger than L_c. In the case the dimension is larger than L_c, taking into account the spatial variability has a significant influence on the structural reliability [23, 24].
4. Neglecting the spatial variability in a concrete structure can lead to an overestimation of the variation in the load capacity of a concrete structure [25].
5. In a reliability analysis of reinforced concrete, it is only relevant to take into account the spatial variation in concrete properties in case of an ULS analysis where crushing of the concrete is governing in the failure mechanisms and in a SLS analysis [11].

In the assessment of the random field generators in chapter 3, the influence of the input parameters on the accuracy and efficiency of the generated random field is determined. For the different statistical characteristics the following conclusions can be drawn:

1. The threshold value in the correlation function and the distribution type both have a significant influence on the accuracy of the random field. If the threshold value increases the defined errors became larger. For a log-normal distributed random field the errors where also larger. Especially the error in the correlation structure becomes larger in both cases.
2. Loss of ergodicity is observed for strongly correlated random fields. This is the case when a correlation function with a large correlation length or/and a threshold value is applied. Ergodicity can not be assumed any longer in such cases. The deviation in the values of the random field decreases, and the deviation in the mean values increases. No direct relation was found between the used correlation function and the change in these statistical properties.

3. No significant differences where found with respect to efficiency and accuracy when a Exp or SExp correlation function was applied. The pattern of the random field is however different. For the SExp correlation function the fields have a much smoother surface.
4. The initialization time and realization time increase with increasing number of nodes in the random field mesh. A linear relation is found between the number of nodes and the realization time for all the methods.

In this study the random field generators are categorized into two classes. The first class entails generators which evaluate correlated random variables on the nodes in the random field mesh. The correlated random variables are thereafter assigned to the elements or integration points with one of the discretization methods. Class 2 entail the series expansion methods where a number of continuous functions is multiplied with a random variable and summed to acquire a continuous random field. A realization of a continuous random field can be numerically integrated to acquire the values for the integration points. In literature the methods in the second class promise to be accurate and efficient. However, the preliminary results of the EOLE method, which is a class 2 method, were very poor with respect to efficiency. Also the program structure in DIANA is very well suited for class 1 methods. It was therefore chosen to focus on class 1 methods in the assessment of the random field generators. With regard to the performance of the class 1 random field generators the following conclusions can be drawn:

1. The CMD method is very accurate in the representation of the mean and the correlation structure of the random field. The decomposition methods are modified to avoid problems with a (numerical) non-positive definite correlation matrix. For the generation of random fields with more than 1000 nodes the method becomes very poor with respect to efficiency. In case a log-normal distributed random field with a COV of 1 is generated, the method is less accurate.
2. The FFT method is a very efficient method. It is also very accurate in the representation of the mean and the correlation structure in the case of a normal and log-normal distributed random field. It is slightly less accurate than the CMD method when the normal distribution type is selected. With respect to efficiency it is slightly less efficient than the LAS method.
3. The LAS method is the most efficient method. With respect to accuracy the method gave more accurate results when a neighbourhood size of 3 was selected, instead of 5. This has to do with the poor treatment of the boundary conditions in the 1D LAS algorithm. The method is less accurate than the other assessed methods in this study.
4. The MA, DFT and TBM were not selected for the assessment. With respect to efficiency the first two methods are said to perform very poor in comparison with the FFT and LAS method. The TBM method is competitive in efficiency with the FFT and LAS method. For this method it is however not clear a priori how many lines have to be selected to acquire accurate results. It is also hard to determine the correlation function for the 1D process for every line in the algorithm.

The JCSS Probabilistic Model Code for concrete material properties was followed to investigate the influence of spatially varying material properties on the behaviour and performance of a non-linear analysis. The model code is implemented as a material model in the general purpose FEM program DIANA. For the suggested correlation function in the model code, the one sided SDF and the variance function, which are needed in some of the random field generators, are derived in this research (see appendix G). A concrete floor submitted to shrinkage was considered. In this model the developed material model and some variations on this model have been used in the analyses. The results are compared with a model with no spatial variation in the material properties. Based on the observation in these analyses the following conclusion can be drawn:

1. When spatially varying material properties are used, it was observed that cracking starts at the weakest point in the structure. From this location the cracks develop over the full length and width of the plate. The crack patterns in the floor are non-symmetric.
2. All the analyses did not reached the convergence norm in the load steps where cracking occurs. In the floor a large strain energy is build up before cracking occurs. This leads to an explosive growth of the crack after the initialization of the first crack. The application of spatially varying material properties gave no change in this behaviour.
3. In other problems, the application of spatially varying material properties may improve the numerical stability. This is based on the following indications:
   - The width of the area with crack strains became smaller, especially when a correlation length of 0.5 m was used. Localization of cracks is an important condition for an analysis to be numerical stable.
   - The growth in total number of integration points which were in the cracked stage was more gradual in the case that spatially varying material properties were applied.
4. When a random field is applied in a non-linear FEA, its not known a priori when the model switches form linear to non-linear behaviour. Smaller load steps have to be selected over a zone where it could be the case that no non-linear behaviour is present. This problem can be omitted by doing a linear analysis prior to the non-linear analysis with the same random field. The load factor where the first cracks arise can be estimated in this way.

7.2 Response to the research question

Based on the findings of this research, an answer on the raised research question is formulated in this section. The research question of this study is formulated as:

> *"To what extent are the available random field generators suitable to represent the statistical characteristics of concrete in a general purpose program, and what is the influence of spatially varying concrete material properies on a non-linear FEA?"*

With regard to the first part of the research question, it can be concluded that all the assessed class 1 methods are appropriate in representing the statistical characteristics of concrete. None of the methods gave erroneous results. The full potential of class 2 methods is not considered in this research. These methods could be competitive to class 1 methods or even outperform them. Further research is necessary to explore the applicability of such methods in a general purpose FEM program. Of the class 1 methods, the CMD method is the most accurate method. However, this method becomes very slow when the number of nodes increases which makes it less suited for a general purpose program. The FFT method is slightly less accurate and stays very efficient with increasing number of nodes. If another correlation function is desired by the user, the one sided SDF of this correlation function have to be derived first. This makes the method less flexible, which is a disadvantage when the method is used in a general purpose program. The LAS method is the most efficient method, but is less accurate then the other methods. In this method the random field is automatically discretized with the spatial average method. This gives some restriction on how the method could be implemented in a general purpose program, which is disadvantageous.

When spatially varying material properties are incorporated in a FE analysis the cracking behaviour is affected. In the investigated example, it resulted in non-symmetric cracks which initialize at the weakest point in the structure. Additionally, indications of a more stable analysis where observed in the different analyses. More cases and applications have to be explored to come to a more solid foundation for the answer of the second part of the research question.

7.3 Recommendations for further research

In the last section of this report, direction for future research are identified, as shown in the following paragraphs:

More experiments In the literature review it was found that there is no agreement on what statistical characteristics of reinforced concrete should be used. In the FE analyses it was shown that the correlation length has a significant influence on the cracking behaviour in the concrete floor. To come to more realistic models it is very important to have more information on what statistical characteristics should be used. Especially, in a reliability analysis of a reinforced concrete structure this is desirable. More experimental data and analysis of such data is needed in future research to determine the statistical characteristics of reinforced concrete.

Class 1 methods vs. Class 2 methods In this report the available methods are divided into two classes. In literature, assessments of the class 1 methods and of the class 2 methods can be found. The performance of the methods in class 1 are however not compared to the performance of the class 2 methods in literature. This would be very interesting to investigate.

Reliability analysis In a reliability analysis of a reinforced concrete structure it is recommended to use the FFT method as random field generator. To come to an accurate estimate of the reliability, the focus should be on finding the most appropriate sampling technique which can be used in the MCS method. Other reliability methods seems to be become too involved when they are combined with a non-linear analysis of a reinforced concrete structure.

More cases and applications Before coming to a conclusion on the additional value of random fields in a probabilistic analysis, more cases and applications with spatially varying material properties need to be explored. When this is done, the influence of spatially varying material properties on a non-linear FEA can be determined on a more solid ground. Next, it is important to explore other practical issues which are faced during such analyses. With this information, robust solution procedures can be selected which can be used in a MCS.

Improving material model The developed material model can be improved by varying the fracture energy in the model. No suggestions are given in the JCSS model code for this material parameter. It can be related deterministically to the tensile strength and Young's modulus or a random field which is correlated to these material parameters can be generated to include variations of the fracture energy in the model. Stimulation of localization of cracks in a FEA is expected.

Bibliography

[1] B. Sudret and A. Der Kiureghian. „Stochastic Finite Element Methods and Reliability. A state-of-the-art-report“. In: *Technical Rep. UCB/SEMM-2000/08, Univ. of California, Berkeley, CA* November (2000).

[2] G. Stefanou. „The Stochastic Finite Element Method : Past , Present and Future“. In: *Computer Methods in Applied Mechanics and Engineering* 198.9-12 (2009), pp. 1031–1051.

[3] CUR pubication 190. *Probabilities in Civil Engineering, Part1: Probabilistic Design Theory*. Gouda: Stichting CUR, 1997.

[4] G.A. Fenton. „Simulation and Analysis of Random Fields“. Dissertation thesis. Princeton University, 1990.

[5] M.A. Hicks and C. Jommi. *ALERT Doctoral School 2014 Stochastic Analysis and Inverse Modelling*. 2014.

[6] E. Vanmarcke. *Random Fields: Analysis and Synthesis*. 1983.

[7] G.E.P. Box and MullerM.E. „A Note on the Generation of Random Normal Deviates“. In: *Ann. Math. Statist* 29.2 (1958), pp. 610–611.

[8] J. Neveau. *Introduction aux Probabilités*. Cours de l'Ecole Polytechnique, 1992.

[9] M.B. Priestley. *Spectral Analysis and Time Series, Vol. 1: Univariate Series*. Academic Press, New York, 1981.

[10] W.A. Spencer. „Parallel Stochastic and Finite Element Modelling of Clay Slope Stability in 3D“. Dissertation thesis. University of Manchester, 2007.

[11] M. de Vasconcellos Real et al. „Response Variability in Reinforced Concrete Structures with Uncertain Geometrical and Material Properties“. In: *Nuclear Engineering and Design* 226.3 (2003), pp. 205–220.

[12] A. Nataf. „Détermination des distributions de probabilités dont les marges sont données“. In: *Comptes Rendus de l'Academie des Sciences* 225 (1958), pp. 42–43.

[13] W. Pula. *Simulation of random fields*. URL: `http://www.ib.pwr.wroc.pl/wpula/W14.pdf` (visited on June 19, 2015).

[14] G.A. Fenton. „Error evaluation of three Random-Field Generators“. In: *Journal of Engineering Mechanics* 120.12 (1994), pp. 2478–2497.

[15] G. Matheron. „The Intrinsic Random Functions and Their Applications“. In: *Advances in Applied Probability* 5 (1973), pp. 439–468.

[16] C.C. Li and A. Der Kiureghian. „Optimal Discretization of Random Fields“. In: *Journal of Engineering Mechanics* 119.6 (1993), pp. 1136–1154.

[17] H.G. Matthies et al. „Uncertainties in Probabilistic Numerical Analysis of Structures and Solids Stochastic Finite Elements“. In: 19.3 (1997), pp. 283–336.

[18] S.K. Jha and J. Ching. „Simulating Spatial Averages of Stationary Random Field Using Fourier Series Method“. In: *Journal of Engineering Mechanics* (2013), pp. 594–605.

[19] I. Papaioannou W. Betz and D. Straub. „Numerical methods for the discretization of random fields by means of the Karhunen – Loève expansion“. In: *Computer Methods in Applied Mechanics and Engineering* 271 (2014), pp. 109–129.

[20] T. Vrouwenvelder. *JCSS probabilistic model Code*. Ed. by M.H. Faber. Mar. 28, 2015. URL: `http://www.jcss.byg.dtu.dk/Publications/Probabilistic_Model_Code`.

[21] F. Biondini et al. „Reliability of material and geometrically non-linear reinforced and prestressed concrete structures“. In: *Computers and Structures* 82.13-14 (2004), pp. 1021–1031.

[22] D. Pukl R. Novak and k. Bergmeister. „Reliability Assessment of Concrete Structures“. In: *Computational Modelling of Concrete Structures-proc. Europ. conf. EURO-C 2003* (2003).

[23] J. Milton de Araújo. „Probabilistic analysis of reinforced concrete columns“. In: *Advances in Engineering Software* 32.12 (2001), pp. 871–879.

[24] J. Zhang and B Ellingwood. „SFEM for Reliability of Structures with Material Nonlinearities“. In: *Journal of Structural Engineering* 122.6 (1996), pp. 701–704.

[25] T.H. Lee and K.M. Mosalam. „Probabilistic Fiber Element Modelling of Reinforced Concrete Structures“. In: *Computers and Structures* 82.27 (2004), pp. 2285–2299.

[26] K. Bergmeister et al. „Structural Assessment and Reliability Analysis for Existing Engineering Structures, Theoretical Background“. In: *Structure and Infrastructure Engineering* 5.4 (2009), pp. 267–275.

[27] D. Novák et al. „Stochastic Nonlinear Fracture Mechanics Finite Element Analysis of Concrete Structures“. In: *Proc. of ICoSSaR'05 the 9th Int. Conference on Structural Safety and Reliability.* Ed. by G.I. Schuëller G. Augusti and M. Ciampoli. 2005.

[28] D. Novák and R. Pukl. „Reliable / reliability computing for concrete structures : Methodology and software tools“. In: *5th International Conference on Reliable Engineering Computing (REC 2012)* (2012), pp. 427–438.

[29] A. Strauss et al. „Advanced Life-Cycle Analysis of Existing Concrete Bridges“. In: *Journal of Materials in Civil Engineering* 20.1 (2008), pp. 9–19.

[30] M.I.B. Valente J.C. Matos and P.J.S. Cruz. „Non Linear Probabilistic Analysis of Reinforced Concrete Structures“. In: *34th IABSE Symposium – Large structures and infrastructures for environmentally constrained and urbanized areas* (2010).

[31] J.C. Matos. „Uncertainty Evaluation of Reinforced Concrete and Composite Structures Behavior“. Dissertation thesis. Universidade do Minho, 2013.

[32] R. Pukl et al. „Spatial variability of material properties in nonlinear computer simulation“. In: *Computational Modelling of Concrete Structures* (2006), pp. 891–896.

[33] Institute for Risk and Uncertainty, University of Liverpool. *COSSAN-X*. URL: `http://www.cossan.co.uk/` (visited on June 12, 2015).

[34] Southwest Research Institute. *NESSUS*. URL: `http://www.nessus.com` (visited on June 12, 2015).

[35] A. Der Kiureghian. *Reliability software*. URL: `http://www.ce.berkeley.edu/~adk/software.html` (visited on June 12, 2015).

[36] Y. Shi and M.G. Stewart. „Spatial Reliability Analysis of Explosive Blast Load Damage to Reinforced Concrete Columns“. In: *Structural Safety* 53 (2015), pp. 13–25.

[37] J. Milton de Araújo and A.M. Awruch. „Probabilistic Finite Element Analysis of Concrete Gravity Dams“. In: 29.2 (1998), pp. 97–104.

[38] A.M. Awruch. „Reliability of Reinforced Concrete Structures Using Stochastic Finite Elements“. In: *Engineering Computations* 19.7 (2002), pp. 764–786.

[39] T. Most and C. Bucher. „Stochastic Simulation of Cracking in Concrete Structures Using Multiparameter Random Fields“. In: *International Journal of Reliability and Safety* 1 (2006), pp. 168–186.

[40] T. Most and C. Bucher. „Probabilistic analysis of concrete cracking using neural networks and random fields“. In: *Probabilistic Engineering Mechanics* 22 (2007), pp. 219–229.

[41] R. de Vries. „The Stochastic Finite Element Method (SFEM) Intrusive formulations and an application in“. Master of Science thesis. Delft University of Technology, 2013.

[42] W.K. Liu et al. „Random Field Finite Elements“. In: *International Journal for Numerical Methods in Engineering* 23 (1986), pp. 1831–1845.

[43] M.A. Gutiérrez et al. „Finite element Reliability methods using Diana“. In: *DIANA Computational Mechanics* (1994), pp. 255–263.

[44] J. Zhang and B. Ellingwood. „Orthogonal Series Expansions of Random Fields in Reliability Analysis“. In: *Journal of Engineering Mechanics* 120.12 (1995), pp. 2660–2677.

[45] G.N. Wells. *The Finite Element Method: An Introduction*. 2009.

[46] RD de Borst and L.J. Sluijs. *Computational Methods in Non-Linear Solid Mechanics*. 2013.

[47] G.A. Fenton. „Error evaluation of three Random-Field Generators“. In: *Journal of Engineering Mechanics* 120.12 (1995), pp. 2478–2497.

[48] I.S. Gradshteyn and I.M. Ryzhik. *Tables of Integrals, Series, and Products (7th edition)*. 2007, p. 958.

[49] M. Matsumoto and T. Nishimura. „Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator“. In: *ACM Trans. Model. Comput. Simul.* 8.1 (Jan. 1998), pp. 3–30.

List of Figures

List of Tables

Appendices

Project outline

A

Stochastic spatial distributed properties in nonlinear crack-analysis

Usually in finite element analysis the domain is considered as a continuum with constant properties in space. In non-linear analysis where softening material failure is considered, strains can localize and onset and development of failure can be strongly effected by stochastic spatial distributions of material properties. Recent design code allows that lower safety factors are applied when spatially distributed properties are used in analysis. There are also indications that spatially distributed material properties are advantageous for the robustness of the non-linear analysis, especially for conditions with large areas with constant loadings.

In the present DIANA product version variations of material properties can only be defined by assigning different properties to individual elements by hand. TNO DIANA BV wants to extend the DIANA program with option to define spatially distributed material properties automatically for the next release and provide users guidance and technical background for using this option. TNO DIANA BV wishes to have a Master-student involved in this development project, to make sure that the relevant procedures are considered for implementation. As each parameter has a particular domain and uncertainty, probably different distributions must be considered.

The following tasks are foreseen:

- Literature review, resulting in overview of relevant procedures to define spatial distribution based on standard deviations and characteristic length and may be other engineering parameters.
- Selection of procedure for implementation together with TNO DIANA developers
- Software implementation together with TNO DIANA Developers, inclusive documentation and testing
- Application on real structures, such as assessment of loading capacity of existing bridge or stability of a soil-slope
- Reporting

This project shall result in a functional extension of the DIANA product version.

The project shall be done in the TNO DIANA office in Delft, preferably in the period of February-August 2015.

Potential literature for review:

- Simulation of random fields for stochastic finite element analysis, M.Vorechovsky and D. Novak, ICOSSAR 2005, ISBN 90 5966 040 4

- Influence of heterogeneity on the reliability and failure of a long 3D slope, M.A. Hicks and A. Spencer, Computers and Geotechnics 37 (2010)
- Stochastic approach to slope stability analysis with in-situ data, J. Nuttall, M. Hicks and M. Lloret-Cabot
- The stochastic Finite Element Method (SFEM), Master thesis Rein de Vries TUD, May 2013.

B Derivation of the Nataf Transformation

In this appendix the transformation of a normally distributed random field to a log-normally distributed random field is derived. This transformation named after scientist A. Nataf.

As given in equation 2.23a, the PDF of the normal distribution is given by:

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma_Y}\text{exp}\left(-\frac{(y-\mu_Y)^2}{2\sigma_Y^2}\right) \tag{B.1}$$

The mapping to a normal distributed random variable Y to a log-normal distributed random variable X is given by:

$$X = \text{exp}(Y) \quad \text{or} \quad Y = ln(X) \tag{B.2}$$

According to the *change of variable theorem*, the PDF of the log-normal random variable can then be written as:

$$f_X(x) = g_Y(y)\frac{dy}{dx} = g(ln(x))\frac{1}{x} = \frac{1}{\sqrt{2\pi}\sigma_Y x}\text{exp}\left(-\frac{(ln(x)-\mu_Y)^2}{2\sigma_Y^2}\right) \tag{B.3}$$

The t-th order moment can be determined as follows:

$$\text{E}[X^t] = \text{E}[\text{exp}(tY)] = \int_{-\infty}^{\infty}\text{exp}(ty)\frac{1}{\sqrt{2\pi}\sigma_Y}\text{exp}\left(-\frac{(y-\mu_Y)^2}{2\sigma_Y^2}\right)dy \tag{B.4}$$

Now the following substitution is applied: $z = y - \mu_Y$, which results in:

$$\begin{aligned}
\text{E}[X^t] &= \int_{-\infty}^{\infty}\text{exp}(t(z+\mu_Y))\frac{1}{\sqrt{2\pi}\sigma_Y}\text{exp}\left(-\frac{z^2}{2\sigma_Y^2}\right)dz \\
&= \text{exp}(t\mu_Y)\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma_Y}\text{exp}\left(\frac{2tz\sigma_Y^2 - z^2}{2\sigma_Y^2}\right)dz \\
&= \text{exp}(t\mu_Y)\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma_Y}\text{exp}\left(\frac{-(z-t\sigma_Y^2)^2 + t^2\sigma_Y^4}{2\sigma_Y^2}\right)dz \\
&= \text{exp}(t\mu_Y + \frac{t^2\sigma_Y^2}{2})\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma_Y}\text{exp}\left(-\left(\frac{z-t\sigma_Y^2}{\sqrt{2}\sigma_Y}\right)^2\right)dz \\
&= \text{exp}(t\mu_Y + \frac{t^2\sigma_Y^2}{2})I
\end{aligned} \tag{B.5}$$

To determine integral I, the following substitution is applied: $x = \frac{z - t\sigma_Y^2}{\sqrt{2}\sigma_Y}$. Thereafter the integral is squared which results in:

$$\begin{aligned} I &= \int_{-\infty}^{\infty} \frac{\sqrt{2}}{\sqrt{2\pi}} \text{exp}(-x^2)dx \\ I^2 &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\sqrt{2}}{\sqrt{2\pi}} \text{exp}(-x^2) \frac{\sqrt{2}}{\sqrt{2\pi}} \text{exp}(-y^2)dxdy \\ &= \frac{1}{\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \text{exp}(-(x^2+y^2))dxdy \end{aligned} \tag{B.6}$$

No the variables are substituted by polar coordinates.

$$\begin{aligned} I^2 &= \frac{1}{\pi}\int_{0}^{2\pi}\int_{0}^{\infty} \text{exp}(-r^2)rdrd\theta \\ &= \frac{2\pi}{\pi}\left[-\frac{1}{2}\text{exp}(-r^2)\right]_0^{\infty} = 1 \quad \rightarrow \quad I = 1 \end{aligned} \tag{B.7}$$

The t$^{\text{th}}$ order moment is then equal to:

$$\text{E}[X^t] = \text{exp}(t\mu_Y + \frac{t^2\sigma_Y^2}{2}) \tag{B.8}$$

The mean and second moment of a log-normal random variable can be expressed as:

$$\mu_X = \text{E}[X] = \text{exp}(\mu_Y + \frac{\sigma_Y^2}{2}) \tag{B.9}$$

$$\text{E}[X^2] = \text{exp}(2\mu_Y + 2\sigma_Y^2) \tag{B.10}$$

According with equation 2.14, the variance of a log-normal random variable can then be expressed as:

$$\begin{aligned} \text{Var} &= \text{E}[X^2] - \text{E}[X]^2 = \text{exp}(2\mu_Y + 2\sigma_Y^2) - \left(\text{exp}(2\mu_Y + 2\sigma_Y^2)\right)^2 \\ &= \text{exp}(2\mu_Y + \sigma_Y^2)(\text{exp}(\sigma_Y^2) - 1) \end{aligned} \tag{B.11}$$

The standard deviation of a log-normal random variable is then equal to:

$$\begin{aligned} \sigma_X &= \sqrt{\text{exp}(2\mu_Y + \sigma_Y^2)(\text{exp}(\sigma_Y^2) - 1)} = \sqrt{\text{exp}(2\mu_Y + \sigma_Y^2)}\sqrt{\text{exp}(\sigma_Y^2) - 1} \\ &= \mu_X\sqrt{\text{exp}(\sigma_Y^2) - 1} \end{aligned} \tag{B.12}$$

Equation B.9 and B.12 can be rewritten such that equation 2.27 is obtained.

According to equation 2.30, the correlation function for two log-normal random variables can be written as:

$$\rho_{X_1,X_2} = \frac{\text{Cov}[X_1, X_2]}{\sigma_{X_1}\sigma_{X_2}} = \frac{\text{E}[X_1X_2] - \text{E}[X_1]\text{E}[X_2]}{\sigma_{X_1}\sigma_{X_2}} \tag{B.13}$$

$\text{E}[X_1X_2]$ can be evaluated as follows:

$$\begin{aligned}\text{E}[X_1X_2] &= \text{E}[\exp(\mu_{Y_1} + \sigma_{Y_1}Y_1)\exp(\mu_{Y_2} + \sigma_{Y_2}Y_2)] \\ &= \exp(\mu_{Y_1} + \mu_{Y_2})\text{E}[\exp(\sigma_{Y_1}Y_1 + \sigma_{Y_2}Y_2)] \\ &= \exp(\mu_{Y_1} + \mu_{Y_2})\exp(\frac{1}{2}(\sigma_{Y_1}^2 + 2\sigma_{Y_1}\sigma_{Y_2}\rho_{Y_1,Y_2} + \sigma_{Y_2}) \qquad \text{(B.14)}\end{aligned}$$

The correlation function then becomes equal to:

$$\begin{aligned}\rho_{X_1,X_2} &= \frac{\exp(\mu_{Y_1} + \mu_{Y_2})\exp(\frac{1}{2}(\sigma_{Y_1}^2 + 2\sigma_{Y_1}\sigma_{Y_2}\rho_{Y_1,Y_2} + \sigma_{Y_2}) - \exp(\mu_{Y_2} + \frac{\sigma_{Y_2}^2}{2})\exp(\mu_{Y_2} + \frac{\sigma_{Y_2}^2}{2})}{\sqrt{\exp(2\mu_{Y_1} + \sigma_{Y_1}^2)(\exp(\sigma_{Y_1}^2) - 1)}\sqrt{\exp(2\mu_{Y_2} + \sigma_{Y_2}^2)(\exp(\sigma_{Y_2}^2) - 1)}} \\ &= \frac{\exp(\mu_{Y_1} + \mu_{Y_2} + \frac{1}{2}\sigma_{Y_1}\frac{1}{2}\sigma_{Y_2})(\exp(\sigma_{Y_1}\sigma_{Y_2}\rho_{Y_1,Y_2}) - 1)}{\exp(\mu_{Y_1} + \mu_{Y_2} + \frac{1}{2}\sigma_{Y_1}\frac{1}{2}\sigma_{Y_2})\sqrt{\exp(\sigma_{Y_1}^2) - 1}\sqrt{\exp(\sigma_{Y_2}^2) - 1}} \\ &= \frac{(\exp(\sigma_{Y_1}\sigma_{Y_2}\rho_{Y_1,Y_2}) - 1)}{\sqrt{\exp(\sigma_{Y_1}^2) - 1}\sqrt{\exp(\sigma_{Y_2}^2) - 1}} \qquad \text{(B.15)}\end{aligned}$$

For a homogeneous field, i.e. $\sigma_{Y_1} = \sigma_{Y_2}$ and $\mu_{Y_1} = \mu_{Y_2}$, the correlation function for log-normal random fields then becomes equal to:

$$\rho_X = \frac{\exp(\sigma_Y\rho_Y) - 1}{\exp(\sigma_Y) - 1} \quad \rightarrow \quad \rho_T(\Delta x) = \frac{\exp(\sigma_T^2\rho(\Delta x)) - 1}{\exp(\sigma_T^2) - 1} \tag{B.16}$$

Which is equal to equation 2.65. The transformed standard deviation σ_Y can be determined with equation 2.27.

C Derivation of the covariance function for two local averages having the same domain size

The covariance between two local averages can be derived by considering two averaging domains of size $|D_a|$ and $|D_b|$, centred at the points $\mathbf{x}_a$ and $\mathbf{x}_b$ and taking expectations.

$$\begin{aligned} \mathrm{E}[H_{D_a}H_{D_b}] &= \mathrm{E}\left[\frac{1}{|D_a|}\int_{D_a} H(\boldsymbol{\xi})d\boldsymbol{\xi}\frac{1}{|D_b|}\int_{D_b} H(\boldsymbol{\eta})d\boldsymbol{\eta}\right] \\ &= \frac{1}{|D_a|\,|D_b|}\int_{D_a}\int_{D_b} \mathrm{E}[H(\boldsymbol{\xi})H(\boldsymbol{\eta})]d\boldsymbol{\xi}d\boldsymbol{\eta} \end{aligned} \tag{C.1}$$

According to equation 2.29, the covariance function for two local averages can be written as:

$$B_{D_aD_b}(\mathbf{x}_a - \mathbf{x}_b) = \frac{1}{|D_a|\,|D_b|}\int_{D_a}\int_{D_b} \mathrm{E}[H(\boldsymbol{\xi})H(\boldsymbol{\eta})]\,d\boldsymbol{\xi}d\boldsymbol{\eta} - \mathrm{E}^2[H(\mathbf{x})] \tag{C.2}$$

For a homogeneous zero mean random field the covariance of two local averages, having the same domain of size $|D|$ and separated by a distance equal to k_nD_n in every direction, can be written as follows:

$$\begin{aligned} B_D(\mathbf{kD}) &= \frac{1}{|D|^2}\int_0^D\int_{nD}^{(n+1)D} \mathrm{E}[H(\boldsymbol{\xi})H(\boldsymbol{\eta})]\,d\boldsymbol{\xi}d\boldsymbol{\eta} \\ &= \frac{1}{|D|^2}\int_0^{D_n}\int_{k_nD_n}^{(k_n+1)D_n}\ldots\int_0^{D_1}\int_{k_1D_1}^{(k_1+1)D_1} B(\xi_1-\eta_1,\ldots,\xi_n-\eta_n)d\xi_1d\eta_1\ldots d\xi_nd\eta_n \end{aligned} \tag{C.3}$$

Where k_n is a positive real number.

To simplify this integral the following change of variables is applied:

$$\xi_n = \Delta y_n + \Delta x_n \qquad \text{and} \qquad \eta_n = \Delta y_n \qquad\qquad \text{for all n} \tag{C.4}$$

Thereafter the 2n-fold integral is reduced to a n-fold integral by integrating over Δy_n for every n. First for n=1 the change of variables and integration over Δy_1 is carried out, which results in:

$$\begin{aligned}
B_D(\mathbf{kD}) &= \frac{1}{|D|^2}\int_0^{D_n}\int_{k_nD_n}^{(k_n+1)D_n}\ldots\int_0^{D_2}\int_{k_2D_2}^{(k_2+1)D_2}\ldots \\
&\ldots\int_{(k_1-1)D_1}^{k_1D_1}\int_{k_1D_1-\Delta x_1}^{D_1} B(\Delta x_1,\xi_2-\eta_2,\ldots,\xi_n-\eta_n)d\Delta x_1 d\Delta y_1 d\xi_2 d\eta_2\ldots d\xi_n d\eta_n \\
&+ \frac{1}{|D|^2}\int_0^{D_n}\int_{k_nD_n}^{(k_n+1)D_n}\ldots\int_0^{D_2}\int_{k_2D_2}^{(k_2+1)D_2}\ldots \\
&\ldots\int_{k_1D_1}^{(k_1+1)D_1}\int_0^{(k_1+1)D_1-\Delta x_1} B(\Delta x_1,\xi_2-\eta_2,\ldots,\xi_n-\eta_n)d\Delta x_1 d\Delta y_1 d\xi_2 d\eta_2\ldots d\xi_n d\eta_n
\end{aligned} \tag{C.5}$$

Where $d\xi_1 d\eta_1$ could be replaced according:

$$\begin{aligned}
d\xi_1 d\eta_1 &= \left|\frac{\partial(\xi_1,\eta_1)}{\partial(\Delta x_1,\Delta y_1)}\right| d\Delta x_1 d\Delta y_1 = \left(\frac{\partial\xi_1}{\partial\Delta x_1}\frac{\partial\eta_1}{\partial\Delta y_1}-\frac{\partial\xi_1}{\partial\Delta y_1}\frac{\partial\eta_1}{\partial\Delta x_1}\right) d\Delta x_1 d\Delta y_1 \\
&= (1\cdot 1-1\cdot 0)d\Delta x_1 d\Delta y_1 = d\Delta x_1 d\Delta y_1
\end{aligned} \tag{C.6}$$

The new domains of the integrals can be found by substitution of the new variables into the boundaries of the original domain like:

$$\begin{aligned}
\xi_1 = k_1D_1 &\rightarrow \Delta y_1+\Delta x_1 = k_1D_1 \\
\xi_1 = (k_1+1)D_1 &\rightarrow \Delta y_1+\Delta x_1 = (k_1+1)D_1 \\
\eta_1 = 0 &\rightarrow \Delta y_1 = 0 \\
\eta_1 = D &\rightarrow \Delta y_1 = D
\end{aligned} \tag{C.7}$$

The old and new domain are drawn in figure C.1. The area is split up in two to find the new domains for the integrals.

Now equation C.5 is integrated over Δy_1, which results in:

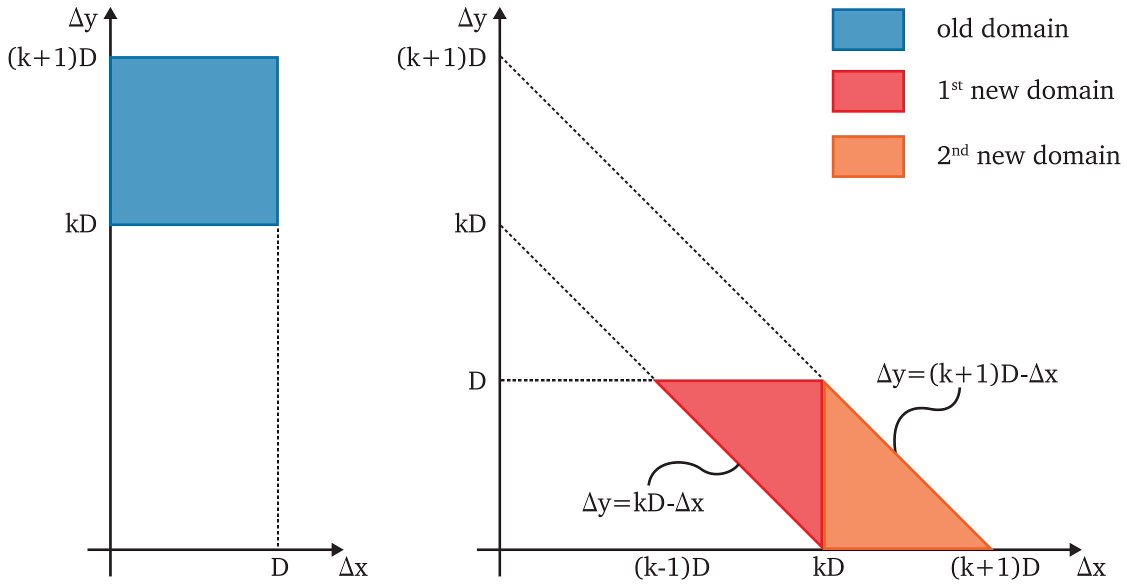


Fig. C.1.: Old and new domains of the integrals.

$$
\begin{aligned}
B_D(\mathbf{kD}) &= \frac{1}{|D|^2}\int_0^{D_n}\int_{k_nD_n}^{(k_n+1)D_n}\dots\int_0^{D_2}\int_{k_2D_2}^{(k_2+1)D_2}\dots \\
&\dots\int_{(k_1-1)D_1}^{k_1D_1}(\Delta x_1-(k_1-1)D_1)B(\Delta x_1,\xi_2-\eta_2,\dots,\xi_n-\eta_n)d\Delta x_1 d\xi_2 d\eta_2\dots d\xi_n d\eta_n \\
&+ \frac{1}{|D|^2}\int_0^{D_n}\int_{k_nD_n}^{(k_n+1)D_n}\dots\int_0^{D_2}\int_{k_2D_2}^{(k_2+1)D_2}\dots \\
&\dots\int_{k_1D_1}^{(k_1+1)D_1}((k_1+1)D_1-\Delta x_1)B(\Delta x_1,\xi_2-\eta_2,\dots,\xi_n-\eta_n)d\Delta x_1 d\xi_2 d\eta_2\dots d\xi_n d\eta_n
\end{aligned}
\tag{C.8}
$$

This procedure can be repeated n times to acquire the following expression for the covariance function:

$$B_D(\mathbf{kD}) = \frac{1}{|D|^2} \sum_{j_1=1}^{2} \cdots \sum_{j_n=1}^{2} \int_{A_{j_n,n}} \cdots \int_{A_{j_1,1}} (C_{j_1,1}) \dots (C_{j_n,n}) B(\Delta x_1, \dots, \Delta x_2) d\Delta x_1 \dots d\Delta x_n \tag{C.9a}$$

Where

$$\begin{aligned} \int_{A_{1,n}} &= \int_{(k_n-1)D_n}^{k_n D_n} \\ \int_{A_{2,n}} &= \int_{k_n D_n}^{(k_n+1)D_n} \end{aligned} \tag{C.9b}$$

$$\begin{aligned} C_{1,n} &= (k_n+1)D_n - \Delta x_n \\ C_{2,n} &= \Delta x_n - (k_n-1)D_n \end{aligned} \tag{C.9c}$$

Δx_1 till Δx_n are the lag distances in different directions which are bounded by the size of the domain.

D Derivation of the Karhunen-Loève expansion

The Karhunen-loève expansion is a series expansion method where the mean square error is minimized. To do so, use is made of orthonormal function in the Hilbert Space. The sequence of orthonormal functions $\{\phi_j(\mathbf{x})\}_{j=1}^{\infty}$ are such that:

$$\int_a^b \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x} = \delta_{ij} \tag{D.1}$$

Which is the inner product of two real functions on the interval [a, b]. Just as for vectors, the inner product of two orthonormal functions is equal to zero.

A random field $H(x,\theta)$ can be expressed as a convergent series of these orthonormal functions:

$$H(\mathbf{x},\theta) = \sum_{i=1}^{\infty} c_i(\theta)\phi_i(\mathbf{x}) \tag{D.2}$$

With $c_j(\theta)$ being random variables and $H(x,\theta)$ not necessarily mean-square periodic, Gaussian or stationary.

To find the correct values for $c_i(\theta)$ the mean square error is minimized. The mean square error is represented as:

$$\epsilon = \int_\Omega [H(\mathbf{x},\theta) - \sum_{i=1}^{\infty} c_i(\theta)\phi_i(\mathbf{x})]^2 d\mathbf{x} \tag{D.3}$$

To find the minimum the derivative is set equal to zero which leads to:

$$\frac{\partial\epsilon}{\partial c_k} = \int_\Omega 2[H(\mathbf{x},\theta) - \sum_{i=1}^{\infty} c_i(\theta)\phi_i(\mathbf{x})]\phi_j(\mathbf{x})d\mathbf{x} = 0 \tag{D.4}$$

Now an expression for $c_k(\theta)$ can be found.

$$\int_\Omega H(\mathbf{x},\theta)\phi_j(\mathbf{x})d\mathbf{x} = \int_\Omega \sum_{i=1}^{\infty} c_i(\theta)\phi_i(\mathbf{x})\phi_j(\mathbf{x})dx = \sum_{i=1}^{\infty}\int_\Omega c_i(\theta)\phi_i(\mathbf{x})\phi_j(\mathbf{x})dx = c_k(\theta) \tag{D.5}$$

Assumed is that the variables $\{c_j(\theta)\}_{j=1}^{\infty}$ are mutually independent, so:

$$\mathrm{E}[c_i(\theta)c_j(\theta)] = \int_P c_i(\theta)c_j(\theta)dP(\theta) =< c_i(\theta)c_j(\theta) >= \lambda\delta_{ij} \tag{D.6}$$

Where $dP(\theta)$ is the joint PDF of the random variables.

Multiplying equation D.2 by $c_j(\theta)$ and taking expectations results in:

$$\mathrm{E}[c_j(\theta)H(\mathbf{x})] = \sum_{i=1}^{\infty} \mathrm{E}[c_i(\theta)c_j(\theta)]\phi_i(\mathbf{x}) = \lambda_j \phi_j(\mathbf{x}) \tag{D.7}$$

Now $c_k(\theta) = \int_\Omega H(\mathbf{x},\theta)\phi_j(\mathbf{x})d\mathbf{x}$ is inserted twice in equation D.7 which results in:

$$\begin{aligned} \mathrm{E}[c_j(\theta)H(\mathbf{x})] &= \sum_{i=1}^{\infty} \mathrm{E}[H(\mathbf{x}_2,\theta)\phi_j(\mathbf{x}_2)d\mathbf{x}_2 c_i(\theta)]\phi_i(\mathbf{x}_1) = \dots \\ \dots &= \int_\Omega \mathrm{E}[H(\mathbf{x}_2,\theta)H(\mathbf{x}_1,\theta)]\phi_j(\mathbf{x}_2)d\mathbf{x}_2 = \lambda_j\phi_j(\mathbf{x}_1) \end{aligned} \tag{D.8}$$

It follows from equation D.8 that:

$$\int_\Omega B(\mathbf{x}_1,\mathbf{x}_2)\phi_j(\mathbf{x}_2)d\mathbf{x}_2 = \lambda_j\phi_j(\mathbf{x}_1) \tag{D.9}$$

Where $B(\mathbf{x}_1,\mathbf{x}_2)$ is the covariance function which behaves as a kernel for the eigenvalue problem where $\phi_k(\mathbf{x})$ are the eigenfunctions and λ_k are the corresponding eigenvalues. The random field can now be expressed as:

$$H(\mathbf{x},\theta) = \mu(\mathbf{x}) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(\mathbf{x})\chi_i(\theta) \tag{D.10}$$

Where $\mu(\mathbf{x})$ is the mean and $\chi(\theta)$ are uncorrelated N(0,1) random variables. The optimal discretized expression for the KL expansion as given in equation 2.119 can be obtained by truncating the summation in this derivation.

E

Bar in tension with a spatially varying Young's modulus

In this appendix, the Matlab code of a bar in tension with a spatially varying Young's modulus is given. It is modelled with 3 noded bar elements in 1D. In this model the integration point methods is used as discretization method. The function file for the generation of the random field according to the CMD methods is given. In the last code, the bar is modelled with 3 noded triangular elements in 2D. In this code the midpoint methods is used as discretization method.

3 noded bar in 1D with integration point discretization method

```
%%  Integraion point method with correlated random variables for a bar in 1D
%   made by Robin van der Have

clear; close all; clc;
tic                         % start timer
%eta=sym('eta');

%% Input model
% Input model parameters
E=10;                       % Mean youngs modulus
E_std=5;                    % Standard deviation youngs modulus
A=1;                        % Cross-sectional area
LL=10;                      % Length of the beam
NE=10;                      % Number of elements
NI=2;                       % Number of integration points

% Dependend model parameters
NN=2*NE+1;                  % number of nodes
xx=linspace(0,LL,NN);       % X-coordinates
NC=[1:NN;xx]';              % Nodal coordinates (Node number, Nodal coordinate)
ENC=zeros(NE+1,2);          % Matrix for Endnode coordinates (Node number, Nodal coordinate)
for i=1:NE+1
    for j=1:2
    ENC(i,j)=NC(2*i-1,j);
    end
end
MNC=zeros(NE,2);            % Matrix for midnode coordinates (Node number, Nodal coordinate)
for i=1:NE
    for j=1:2
    MNC(i,j)=NC(i*2,j);
    end
end
% Matrix with element connectivity (element, node 1, node 2, node 3)
CM=[(1:NE)' ENC(1:length(ENC)-1,1) MNC(:,1) ENC(2:length(ENC),1)];
ll=LL/NE;                   % Length of one element
```

```

% Boundary conditions (1=dirichlet, 0=neuman)
%   node    bc type     value
BC=[1       1           0;          % Clamped at x=0
    NN      0           10];        % Force at end of beam in kN
NB=size(BC,1);

%% Shape functions for three noded bar element
%N1= 0.5*eta^2-0.5*eta;
%N2= -eta^2+1;
%N3= 0.5*eta^2+0.5*eta
%B1=eta-0.5;
%B2=-2*eta;
%B3=eta+0.5;
%Diff=[B1 B2 B3];
%J=Diff*CMX;

%Gaussian integration with 2 integrationpoints
G=[-1/sqrt(3) 1/sqrt(3)];           % Locations of Gausspoints
W=[1 1];                            % Weigths
J=ll/2;                             % dx=Length/2*deta

%% Random field
% Input random field
Dim=1;                      % Model/RF dimension
mean=E;                     % Mean RF
std=E_std;                  % Standard deviation RF
CF='SExp';                  % Type of correlation function (Exp, SExp, CSin)
c1=0;                       % Threshold value for correlation
lc=1;                       % Correlation length
Dec='eigen';                % Decomposition method
distribution='normal';      % Distribtuion type

% Random field mesh (midpoint method)
IP=zeros(NE*NI,1);             % Vector for coordinates midpoints
for i=1:NE                     % Loop to determine coordinates of midpoints elements
IP(i*2-1)=MNC(i,2)+G(1)*ll/2;
IP(i*2)=MNC(i,2)+G(2)*ll/2;
end
Mesh=IP;

% Random field mesh (integration point method)

% Generate random field
RF=CovDec(Dim,mean,std,CF,c1,lc,Mesh,Dec,distribution);
CRV=RF(:,2);

% Constant field
%CRV=linspace(E,E,NE)';
%RF=[Mesh CRV];

% plot random field
y=linspace(0,0,NN);
yy=[RF(:,2); RF(length(Mesh),2)];
figure()
stem(RF(:,1),RF(:,2),'-x')
hold on
plot(xx,y,'o')
stairs(NC(1:length(NC),2),yy)
hold off

% Check RF (add display of error)
```

```
m=sum(CRV)/length(Mesh);
sigma=sqrt(sum((CRV-m).^2)/length(Mesh));


%% start solve
% Assemble global stiffness matrix
k_e=zeros(3,3);                     % Create matrix for element stiffness matrix
K=zeros(NN,NN);                     % Create matrix for global stiffness matrix
for i=1:NE
    for j=1:2
        g=G(j);
    B=2/ll*[g-0.5 -2*g g+0.5];
    k_e=k_e+B'*RF(i*2+j-2,2)*A*B*W(j)*J;
    end
    for p=2:4
        for q=2:4
            r=CM(i,p);
            s=CM(i,q);
            K(r,s)=K(r,s)+k_e(p-1,q-1);
        end
    end
k_e=zeros(3,3);
end
%K=sparse(K);

% Create Force vector
F=zeros(NN,1);

% Impose dirichlet boundary conditions
for i=1:NB
    if BC(i,2)==1;
        K(BC(i,1),:)=0;
        K(:,BC(i,1))=0;
        K(BC(i,1),BC(i,1))=1;
        F(BC(i,1))=BC(i,3);
    end
end

% Apply neuman boundary conditions
for i=1:NB
if BC(i,2)==0;
F(BC(i,1),1)=BC(i,3);
end
end

%Solve system Ku=f
u=K\F;

%analytical solution based on the mean value of E
u_anal=LL*BC(2,3)/(m*A);

%plot displacements
figure()
plot(xx,u)
hold on
plot([0 LL],[0 u_anal])
hold off
toc
```

Covariance matrix decomposition method for random field generation

```
function RF = CovDec(Dim,mean,std,CF,c1,lc,Mesh,Dec,distribution)
%CovDec  returns correlated random variables for a isotropic gaussain field
%
% Output:
%   RF:     Matrix containing the coordinates and correlated random variables
%           belonging to that coordinate
%
% Input:
%   Dim:     Model dimension (1, 2, 3)
%   mean:    Mean value random field
%   std:     Standard deviation random field
%   CF:      Type of correlation function ('Exp', 'SExp')
%                Exp:    Exponential correlation function
%                            Exp = c1+(1-c1)*exp(-delta_x/lc)
%                SExp:   Squared exponential correlation function
%                            SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%            Where delta_x is the lag distance which can be detemined with
%            the coordinates of the Mesh
%   c1:      threshold value for correlation function
%   lc:      Correlation length/scale of fluctuation in both directions
%   Mesh:    Matrix containing the coordinates of the random field mesh
%   Dec:     Decomposition method (chol/eigen)

if Dim==1
    %Determine Correlation matrix
    Cor_M=zeros(length(Mesh),length(Mesh));
    if strcmp(CF,'Exp')==1
        for i=1:length(Mesh)
            for j=1:length(Mesh)
                Cor_M(i,j)=c1+(1-c1)*exp(-abs((Mesh(i)-Mesh(j)))/lc);
            end
        end
    elseif strcmp(CF,'SExp')==1
        for i=1:length(Mesh)
            for j=1:length(Mesh)
                Cor_M(i,j)=c1+(1-c1)*exp(-((abs(Mesh(i)-Mesh(j)))/lc)^2);
            end
        end
    else
        disp('CorF must be equal to Exp or SExp')
    end

elseif Dim==2
    %Determine Correlation matrix
    Cor_M=zeros(length(Mesh),length(Mesh));
    if strcmp(CF,'Exp')==1
        for i=1:length(Mesh)
            for j=1:length(Mesh)
                Cor_M(i,j)=c1+(1-c1)*exp(-(sqrt((Mesh(i,1)-Mesh(j,1))^2 ...
                +(Mesh(i,2)-Mesh(j,2))^2))/lc);
            end
        end
    elseif strcmp(CF,'SExp')==1
        for i=1:length(Mesh)
            for j=1:length(Mesh)
```

```
                Cor_M(i,j)=c1+(1-c1)*exp(-((sqrt((Mesh(i,1)-Mesh(j,1))^2 ...
                +(Mesh(i,2)-Mesh(j,2))^2))/lc)^2);
            end
        end
    else
        disp('CorF must be equal to Exp or SExp')
    end

elseif Dim==3
    disp('not programmed')
end

if strcmp(distribution,'lognormal')==1
    std_norm=sqrt(log(1+(std/mean)^2));
    mean_norm=log(mean)-0.5*std_norm^2;
    Cor_M=(exp(Cor_M*std_norm^2)-1)/(exp(std_norm^2)-1);
end

% Decomposition correlation matrix
if strcmp(Dec,'chol')==1
    L=chol(Cor_M,'lower');
elseif strcmp(Dec,'eigen')==1
    [vec,lambda]=eig(Cor_M);
    L=vec*sqrt(abs(lambda));
elseif strcmp(Dec,'mod chol')==1
    L=chol2(Cor_M);
    L=L';
else
    disp('Dec must be equal to chol, mod chol or eigen')
end
RV=normrnd(0,1,[length(Mesh),1]);
if strcmp(distribution,'normal')==1
    CRV=mean+L*RV*std;    % Generate random field
elseif strcmp(distribution,'lognormal')==1
    CRV=exp(mean_norm+L*RV*std_norm);    % Generate random field
end
RF=[Mesh CRV];
end
```

3 noded bar in 2D with midpoint discretization method

```
%%  Midpoint method with correlated random variables for a bar in 2D
%   made by Robin van der Have

clear; close all; clc;
tic                         % start timer

%% Input model
% Input model parameters
E_m=10;                     % Mean youngs modulus
E_std=5;                    % Standard deviation youngs modulus
nu=0;                       % Poisson ratio
A=1;                        % Cross-sectional area
LL=10;                      % Length of the beam
HH=1;                       % Height of the beam
t=1;                        % Thickness
NB=10;                      % Number of collections of 4 triangle elemtns
NI=1;                       % Number of integration points
DOF=2;                      % Number of degrees of freedom per node

% Dependend model parameters
NE=4*NB;
NN=3*NB+2;                  % number of nodes
xx=linspace(0,LL,NB*2+1);   % X-coordinates
yy=linspace(0,HH,3);        % Y-coordinates
ll=LL/NB;                   % Length of one block of 4 elements

% Form D matrix for plane stress
D = 1/(1-nu^2)*[1 nu 0 ; nu 1 0; 0 0 (1-nu)/2] ;

% Matrix with nodal coordinates (Node number, x-coordinate, y-coordinate)
NC=zeros(NN,3);
NC(:,1)=1:NN;
NC(1,2)=xx(1); NC(2,2)=xx(1); NC(1,3)=yy(1); NC(2,3)=yy(3);
for i=1:NB
    NC(i*3,2)=xx(i*2+1);
end
for i=1:NB
    NC(i*3+1,2)=xx(i*2+1);
end
for i=1:NB
    NC(i*3+2,2)=xx(i*2);
end
for i=1:NB
    NC(i*3,3)=yy(1);
end
for i=1:NB
    NC(i*3+1,3)=yy(3);
end
for i=1:NB
    NC(i*3+2,3)=yy(2);
end

% Matrix with element connectivity (element, node 1, node 2, node 3)
CM=zeros(NE,4);
CM(:,1)=1:NE;
for i=1:NB
    CM(i*4-3:i*4,2:4)=[i*3-3 i*3 i*3+2; i*3 i*3+1 i*3+2; i*3+2 i*3+1 i*3-2; i*3-3 i*3+2 i*3-2];
end
```

```
CM(1,2)=1; CM(4,2)=1; CM(3,4)=2; CM(4,4)=2;

% Coordinates of nodes per element
CMX=changem(CM(:,2:4),NC(:,2)',1:NN);
CMX=CMX';
CMY=changem(CM(:,2:4),NC(:,3)',1:NN);
CMY=CMY';

% Boundary conditions (1=dirichlet, 0=neuman) dof 1=x and dof 2=y
%   node    bc type dof     value
BC=[1       1       1       0;        % Clamped at x=0
    1       1       2       0         % Clamped at x=0
    2       1       1       0         % Clamped at x=0
    NN-2    0       1       5         % Half of the force at end of beam in kN
    NN-1    0       1       5];       % Half of the force at end of beam in kN
NBC=size(BC,1);                       % Number of boundary conditions

%% Random field
% Input random field
Dim=2;                      % Model/RF dimension
mean=E_m;                   % Mean RF
std=E_std;                  % Standard deviation RF
CF='SExp';                  % Type of correlation function (Exp, SExp)
c1=0;                       % Threshold value for correlation
lc=1;                       % Correlation length
Dec='eigen';                % Decomposition method
distribution='normal';      % Distribtuion type

% Random field mesh (midpoint method)
MP=zeros(NE,2);             % Vector for coordinates midpoints
for i=1:NB                  % Loop to determine coordinates of midpoints elements
MP(i*4-3:i*4,1:2)=[1/2*ll+ll*(i-1) 1/6*HH; 5/6*ll+ll*(i-1) 1/2*HH; ...
    1/2*ll+ll*(i-1) 5/6*HH; 1/6*ll+ll*(i-1) 1/2*HH];
end
Mesh=MP;

% Generate random field
RF=CovDec(Dim,mean,std,CF,c1,lc,Mesh,Dec,distribution);
CRV=RF(:,3);

% Constant field
%CRV=linspace(E,E,NE)';
%RF=[Mesh CRV];

% plot random field
CRVP=[CRV'; CRV'; CRV'];

figure()
fill3(CMX,CMY,CRVP,CRV')
colormap autumn
axis equal
view (0,90)

% Check RF (add display of error)
m=sum(CRV)/length(Mesh);
sigma=sqrt(sum((CRV-m).^2)/length(Mesh));

%% Derivatives of shape functions evaluated at integraion points for two noded bar element
%N1= -xi-eta+1;
%N2= xi;
%N3= eta
%N=[N1 N2 N3];
```

```

Diff=[-1 1 0;       % Derivatives of shape functions (dN1/dx dN2/dx dN3/dx; dN1/dy dN2/dy dN3/dy)
      -1 0 1];
Jx=Diff*CMX;
Jy=Diff*CMY;
J=[Jx;Jy];          % Jacobian matrix

% Derivatives of shapes function with respect to x and y for every element
dN=zeros(6,NE);
for i=1:NE
    for j=1:3
    dN(j*2-1:j*2,i)=1/(J(1,i)*J(4,i)-J(3,i)*J(2,i))*[J(4,i) -J(3,i); -J(2,i) J(1,i)]*Diff(:,j);
    end
end

% Weigth for gauss integration, equal to 0.5 since the surface of the parent element is 0.5
W=1/2;
%% start solve
% Assemble global stiffness matrix
k_e=zeros(6,6);                     % Create matrix for element stiffness matrix
K=zeros(NN*DOF,NN*DOF);             % Create matrix for global stiffness matrix
for i=1:NE
    B=[dN(1,i) 0 dN(3,i) 0 dN(5,i) 0;
        0 dN(2,i) 0 dN(4,i) 0 dN(6,i);
        dN(2,i) dN(1,i) dN(4,i) dN(3,i) dN(6,i) dN(5,i)];
    j=(J(1,i)*J(4,i)-J(3,i)*J(2,i));
    k_e=B'*RF(i,3)*D*B*j*t*W;
    for p=2:4
        for q=2:4
            r=2*CM(i,p)-1;          % Assemble values corresponding to dof in x direction
            s=2*CM(i,q)-1;
            K(r,s)=K(r,s)+k_e((p-1)*2-1,(q-1)*2-1);
            r=2*CM(i,p);          % Assemble values corresponding to dof in y direction
            s=2*CM(i,q);
            K(r,s)=K(r,s)+k_e((p-1)*2,(q-1)*2);
        end
    end
k_e=zeros(3,3);
end
%K=sparse(K);

% Create Force vector
F=zeros(NN*DOF,1);

% Impose dirichlet boundary conditions
for i=1:NBC
    if BC(i,2)==1;
        if BC(i,3)==1                   %BC in x direction
            K(2*BC(i,1)-1,:)=0;
            K(:,2*BC(i,1)-1)=0;
            K(2*BC(i,1)-1,2*BC(i,1)-1)=1;
            F(2*BC(i,1)-1)=BC(i,4);
        elseif BC(i,3)==2               % BC in y direction
            K(2*BC(i,1),:)=0;
            K(:,2*BC(i,1))=0;
            K(2*BC(i,1),2*BC(i,1))=1;
            F(2*BC(i,1))=BC(i,4);
        end
    end
end

% Apply neuman boundary conditions
```

```
for i=1:NBC
    if BC(i,2)==0;
        if BC(i,3)==1                   %BC in x direction
            F(2*BC(i,1)-1,1)=BC(i,4);
        elseif BC(i,3)==2               %BC in y direction
            F(2*BC(i,1),1)=BC(i,4);
        end
    end
end

%Solve system Ku=f
u=K\F;

%analytical solution based on the mean value of E
u_anal=LL*BC(4,4)*2/(m*A);

%plot displacements
Ux=u(1:2:NN*2-1);
Uy=u(2:2:NN*2);
X=NC(:,2);
Y=NC(:,3);
figure()
scatter(X,Ux)
hold on
plot([0 LL],[0 u_anal])
hold off
toc
```

F

Matlab code: Determination statistical properties random fields

```matlab
function [mean_m,mean_s,std_m,std_s,c_err_m, c_err_s]= stat_prop( RF,runs,dim,Mesh,CF,c1,d)
% stat_prop determines the statistical properties of a number of random
% fields
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Output:
%   mean_m: mean of the mean values of the random fields
%   mean_s: standard deviation of the mean values of the random fields
%   std_m:  Mean of the standard deviations of the random fields
%   std_s:  Standard deviation of the standard deviations of the random
%           fields
%
% Input:
%   RF:     Matrix containing the random field
%   runs:   Number of random fields
%   dim:    Dimension of the random field

if dim==1
    N=size(RF,1);
    m=zeros(1,runs);            % Allocate vector for mean values
    sigma=zeros(1,runs);        % Allocate vector for standard deviations

    % Determine mean and std per field
    for i=1:runs
    m(i)=sum(RF(:,i))/N;
    sigma(i)=sqrt(sum((RF(:,i)-m(i)).^2)/(N));
    end

    % Determine statistical properties for all fields
    mean_m=sum(m)/length(m);
    mean_s=sqrt(sum((m-mean_m).^2)/length(m));
    std_m=sum(sigma)/length(sigma);
    std_s=sqrt(sum((sigma-std_m).^2)/length(sigma));

    % Determine correlation of random fields
    lag=Mesh(1:N-1);                    % Lag distance
    C=zeros(N-1,1);
    for i=1:(N-1)           % Loop for every lag
        s=0;
        RF1=0;
        RF2=0;
        RF1_s=0;
        RF2_s=0;
        for k=i:N           % Summations to determine mean value
            for j=1:runs    % Loop for every run
                RF1=RF1+RF(k-i+1,j);
                RF2=RF2+RF(k,j);
            end
        end
        RF1_m=RF1/(runs*(N-i+1));
        RF2_m=RF2/(runs*(N-i+1));
            for k=i:N
                for j=1:runs
                s=s+(RF(k-i+1,j)-RF1_m)*(RF(k,j)-RF2_m);
                RF1_s=RF1_s+(RF(k-i+1,j)-RF1_m)^2;
                RF2_s=RF2_s+(RF(k,j)-RF2_m)^2;
            end
        end
        C(i)=s/(sqrt(RF1_s)*sqrt(RF2_s));
    end

    % Exact correlation function
    if strcmp(CF,'Exp')==1
        CF=c1+(1-c1)*exp(-(lag./d));
    elseif strcmp(CF,'SExp')==1
        CF=c1+(1-c1)*exp(-(lag./d).^2);
    else
        disp('CF must be equal to Exp or SExp')
    end

    % Plot of covariance
    figure()
    axis([0 17.5 0 1])
    hold on
    plot(lag,C)
    plot(lag,CF)
    hold off

    % Determine error correlation
    err=abs(C-CF');
    c_err_m=sum(err)/length(err);
    c_err_s=sqrt(sum((err-c_err_m).^2)/length(err));

elseif dim==2
    N1=size(RF,1);
    N2=size(RF,2);
    L_cx=d(1,1);
    N=N1*N2;
    m=zeros(1,runs);            % Allocate vector for mean values
    sigma=zeros(1,runs);        % Allocate vector for standard deviations

    % Determine mean and std per field
    for i=1:runs
    m(i)=sum(sum(RF(:,:,i)))/N;
    sigma(i)=sqrt(sum(sum((RF(:,:,i)-m(i)).^2))/N);
    end

    % Determine statistical properties for all fields
    mean_m=sum(m)/length(m);
    mean_s=sqrt(sum((m-mean_m).^2)/length(m));
    std_m=sum(sigma)/length(sigma);
    std_s=sqrt(sum((sigma-std_m).^2)/length(sigma));

    % Determine correlation of random fields
    lag=Mesh(1:N1-1);                    % Lag distance
    C=zeros(N1-1,1);
    for i=1:(N1-1)           % Loop for every lag
        s=0;
        RF1=0;
        RF2=0;
        RF1_s=0;
        RF2_s=0;
        for k=i:N1           % Summations to determine mean value
            for l=1:N2
                for j=1:runs    % Loop for every run
                    RF1=RF1+RF(k-i+1,l,j);
                    RF2=RF2+RF(k,l,j);
                end
            end
        end
        RF1_m=RF1/(runs*N2*(N1-i+1));
        RF2_m=RF2/(runs*N2*(N1-i+1));
        for k=i:N1
            for l=1:N2
                for j=1:runs
                    s=s+(RF(k-i+1,l,j)-RF1_m)*(RF(k,l,j)-RF2_m);
                    RF1_s=RF1_s+(RF(k-i+1,l,j)-RF1_m)^2;
                    RF2_s=RF2_s+(RF(k,l,j)-RF2_m)^2;
                end
            end
        end
        C(i)=s/(sqrt(RF1_s)*sqrt(RF2_s));
    end

    % Exact correlation function
    if strcmp(CF,'Exp')==1
        CF=c1+(1-c1)*exp(-(lag./L_cx));
    elseif strcmp(CF,'SExp')==1
        CF=c1+(1-c1)*exp(-(lag./L_cx).^2);
    else
        disp('CF must be equal to Exp or SExp')
    end

    % Plot of covariance
    figure()
    axis([0 17.5 0 1])
    hold on
    plot(lag,C)
    plot(lag,CF)
    hold off

    % Determine error correlation
    err=abs(C-CF');
    c_err_m=sum(err)/length(err);
    c_err_s=sqrt(sum((err-c_err_m).^2)/length(err));

else
   disp('Not programmed')
end
```

G

SDF and variance function for Exponential and Squared Exponential covariance function

In this appendix the one sided Spectral Density Functions (SDF) and variance functions which are given in section 3.2.1 will be derived.

SDF 1D Exponential Covariance function

$$\begin{aligned}
G(\omega) &= \frac{2}{\pi}\int_0^\infty B(\Delta x)\cos(\omega\Delta x)d\Delta x \\
&= \frac{2\sigma^2}{\pi}\int_0^\infty \left(c_1 + (1-c_1)\exp\left(-\frac{\Delta x}{L_c}\right)\right)\cos(\omega\Delta x)d\Delta x \\
&= \frac{2\sigma^2}{\pi}\left[\int_0^\infty c_1\cos(\omega\Delta x)d\Delta x + \int_0^\infty (1-c_1)\exp\left(-\frac{\Delta x}{L_c}\right)\cos(\omega\Delta x)d\Delta x\right] \\
&= \frac{2\sigma^2}{\pi}(I_1 + (1-c_1)I_2) \qquad \text{(G.1)}
\end{aligned}$$

The first integral can be solved easily since it is known that the Fourier transform of a constant is equal to the Dirac delta functions times that constant. Because c_1 is a real and even function, the integral can be rewritten as equation 2.47, which results in:

$$\begin{aligned}
I_1 &= \int_0^\infty c_1\cos(\omega\Delta x)d\Delta x \\
&= \frac{2\pi}{2}\frac{1}{2\pi}\int_{-\infty}^\infty c_1\exp(-i\omega\Delta x)d\Delta x \\
&= \pi c_1\delta(\omega) \qquad \text{(G.2)}
\end{aligned}$$

To solve the second integral, integration by parts is applied twice:

$$\begin{aligned}
I_2 &= \left[\exp(-\frac{\Delta x}{L_c})\sin(\omega\Delta x)\frac{1}{\omega}\right]_{\Delta x=0}^{\Delta x=\infty} - \int_0^\infty -\frac{(1)}{L_c}\exp\left(-\frac{\Delta x}{L_c}\right)\sin(\omega\Delta x)\frac{1}{\omega}d\Delta x \\
&= \left[\exp(-\frac{\Delta x}{L_c})\sin(\omega\Delta x)\frac{1}{\omega}\right]_{\Delta x=0}^{\Delta x=\infty} - \left[\frac{1}{L_c\omega^2}\exp(-\frac{\Delta x}{L_c})\cos(\omega\Delta x)\right]_{\Delta x=0}^{\Delta x=\infty} \\
&- \frac{1}{L_c^2\omega^2}\int_0^\infty \exp\left(-\frac{\Delta x}{L_c}\right)\cos(\omega\Delta x)d\Delta x \qquad \text{(G.3)}
\end{aligned}$$

In the last term the original integral can now be recognized. This part can be taken to the LHS to find an expression for I_2.

$$\begin{aligned}
I_2\left(1+\frac{1}{L_c^2\omega^2}\right) &= \left[\exp(-\frac{\Delta x}{L_c})\sin(\omega\Delta x)\frac{1}{\omega}\right]_{\Delta x=0}^{\Delta x=\infty} - \left[\frac{1}{L_c\omega^2}\exp(-\frac{\Delta x}{L_c})\cos(\omega\Delta x)\right]_{\Delta x=0}^{\Delta x=\infty} \\
I_2 &= \frac{\left[\exp(-\frac{\Delta x}{L_c})\sin(\omega\Delta x)\frac{1}{\omega}\right]_{\Delta x=0}^{\Delta x=\infty} - \left[\frac{1}{L_c\omega^2}\exp(-\frac{\Delta x}{L_c})\cos(\omega\Delta x)\right]_{\Delta x=0}^{\Delta x=\infty}}{\left(1+\frac{1}{L_c^2\omega^2}\right)} \\
&= \frac{[0-0]+\left[0+\frac{1}{L_c\omega^2}\right]}{\left(1+\frac{(1-c_1)}{L_c^2\omega^2}\right)} = \frac{L_c}{L_c^2\omega^2+1} \qquad \text{(G.4)}
\end{aligned}$$

Now I_1 and I_2 are substituted in equation G.6 to find the expression for the one sided SDF.

$$G(\omega) = \frac{2\sigma^2}{\pi}(I_1+(1-c_1)I_2) = 2\sigma^2 c_1\delta(\omega) + \frac{2(1-c_1)\sigma^2 L_c}{\pi(L_c^2\omega^2+1)} \qquad \text{(G.5)}$$

SDF 1D Squared Exponential Covariance function

$$\begin{aligned}
G(\omega) &= \frac{2}{\pi}\int_0^\infty B(\Delta x)\cos(\omega\Delta x)d\Delta x \\
&= \frac{2\sigma^2}{\pi}\int_0^\infty \left[c_1+(1-c_1)\exp\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)\right]\cos(\omega\Delta x)d\Delta x \\
&= \frac{2\sigma^2}{\pi}\left[\int_0^\infty c_1\cos(\omega\Delta x)d\Delta x + \int_0^\infty (1-c_1)\exp\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)\cos(\omega\Delta x)d\Delta x\right] \\
&= \frac{2\sigma^2}{\pi}(I_1+(1-c_1)I_2(\omega)) \qquad \text{(G.6)}
\end{aligned}$$

The first integral is given in equation G.2. To solve the second integral, differentiating under the integral sign is applied.

$$I_2'(\omega) = \int_0^\infty -\Delta x\exp\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)\sin(\omega\Delta x)d\Delta x \qquad \text{(G.7)}$$

Now integration by parts is applied.

$$\begin{aligned} I_2'(\omega) &= \left[\frac{L_c^2}{2}\text{sin}(\omega\Delta x)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)\right]_{\Delta x=0}^{\Delta x=\infty} \\ &- \frac{\omega L_c^2}{2}\int_0^\infty \text{cos}(\omega\Delta x)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)d\Delta x \\ &= 0 - \frac{\omega L_c^2}{2}I_2(\omega) \to I_2'(\omega) + \frac{\omega L_c^2}{2}I_2(\omega) = 0 \end{aligned} \tag{G.8}$$

A solution of this differential equation is equal to $I_2(\omega) = C\text{exp}\left(-\frac{\omega^2 L_c^2}{4}\right)$. The constant C is the only unknown in this equation. It can be found by setting ω equal to 0.

$$\begin{aligned} I_2(0) &= \int_0^\infty \text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)\text{cos}(0\cdot\Delta x)d\Delta x = C\text{exp}(0) \\ C &= \int_0^\infty \text{exp}xp\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)\Delta x \\ C^2 &= \left(\int_0^\infty \text{exp}\left(-\left(\frac{\Delta x_1}{L_c}\right)^2\right)\Delta x_1\right)\left(\int_0^\infty \text{exp}\left(-\left(\frac{\Delta x_2}{L_c}\right)^2\right)\Delta x_2\right) \\ &= \int_0^\infty\int_0^\infty \text{exp}\left(-\frac{\Delta x_1^2+\Delta x_2^2}{L_c^2}\right)\Delta x_1\Delta x_2 \end{aligned} \tag{G.9}$$

The variables are now substituted by polar coordinates.

$$\begin{aligned} I_2(0) &= \int_0^{\frac{\pi}{2}}\int_0^\infty r\,\text{exp}\left(-\frac{r^2}{L_c^2}\right)\Delta r\Delta\theta \\ &= \frac{\pi}{2}\int_0^\infty r\,\text{exp}\left(-\frac{r^2}{L_c^2}\right)\Delta r \end{aligned} \tag{G.10}$$

The following substitution is applied: $u = \frac{r^2}{L_c^2}$.

$$C^2 = \frac{\pi L_c^2}{4}\int_0^\infty \text{exp}(-u)du = \frac{\pi L_c^2}{4} \tag{G.11}$$

$$C = \sqrt{\frac{\pi L_c^2}{4}} = \frac{L_c\sqrt{\pi}}{2} \tag{G.12}$$

Now I_1 and I_2 are substituted in equation G.6 to find the expression for the one sided SDF.

$$G(\omega) = \frac{2\sigma^2}{\pi}(I_1 + (1-c_1)I_2) = 2\sigma^2 c_1\delta(\omega) + \frac{(1-c_1)\sigma^2 L_c}{\sqrt{\pi}}\text{exp}\left(-\frac{\omega^2 L_c^2}{4}\right) \tag{G.13}$$

Variance function 1D Exponential Covariance function

$$\begin{aligned}
\gamma(D) &= \frac{2}{D^2}\int_0^D (D-\Delta x)\rho(\Delta x)d\Delta x \\
&= \frac{2}{D^2}\int_0^D (D-\Delta x)\left(c_1+(1-c_1)\text{exp}\left(-\frac{\Delta x}{L_c}\right)\right)d\Delta x \\
&= \frac{2}{D^2}\int_0^D Dc_1 d\Delta x - \frac{2}{D^2}\int_0^D \Delta x c_1 d\Delta x + \frac{2}{D^2}\int_0^D D(1-c_1)\text{exp}\left(-\frac{\Delta x}{L_c}\right)d\Delta x \\
&- \frac{2}{D^2}\int_0^D \Delta x(1-c_1)\text{exp}\left(-\frac{\Delta x}{L_c}\right)d\Delta x \\
&= \frac{2}{D^2}\left[I_1 - I_2 + I_3 - I_4\right] \qquad \text{(G.14)}
\end{aligned}$$

Now the different integrals are solved and substituted in equation G.14. To solve the fourth integral integration by parts is applied.

$$I_1 = \int_0^D Dc_1 d\Delta x = Dc_1\left[\Delta x\right]_{\Delta x=0}^{\Delta x=D} = D^2c_1 \qquad \text{(G.15)}$$

$$I_2 = \int_0^D \Delta x c_1 d\Delta x = c_1\left[\frac{1}{2}\Delta x^2\right]_{\Delta x=0}^{\Delta x=D} = \frac{1}{2}D^2c_1 \qquad \text{(G.16)}$$

$$\begin{aligned}
I_3 &= \int_0^D D(1-c_1)\text{exp}\left(-\frac{\Delta x}{L_c}\right)d\Delta x = D(1-c_1)\left[-L_c\text{exp}\left(-\frac{\Delta x}{L_c}\right)\right]_{\Delta x=0}^{\Delta x=D} \\
&= -L_cD(1-c_1)\text{exp}\left(-\frac{D}{L_c}\right) + L_cD(1-c1) \qquad \text{(G.17)}
\end{aligned}$$

$$\begin{aligned}
I_4 &= \int_0^D \Delta x(1-c_1)\text{exp}\left(-\frac{\Delta x}{L_c}\right)d\Delta x \\
&= \left[-(1-c_1)\Delta x L_c\text{exp}\left(-\frac{\Delta x}{L_c}\right)\right]_{\Delta x=0}^{\Delta x=D} + L_c(1-c_1)\int_0^D \text{exp}\left(-\frac{\Delta x}{L_c}\right)d\Delta x \\
&= -(1-c_1)DL_c\text{exp}\left(-\frac{D}{L_c}\right) + 0 + \left[-L_c^2(1-c_1)\text{exp}\left(-\frac{\Delta x}{L_c}\right)\right]_{\Delta x=0}^{\Delta x=D} \\
&= -(1-c_1)DL_c\text{exp}\left(-\frac{D}{L_c}\right) - (1-c_1)L_c^2\text{exp}\left(-\frac{D}{L_c}\right) + (1-c_1)L_c^2 \qquad \text{(G.18)}
\end{aligned}$$

$$\begin{aligned}
\gamma(D) &= \frac{2}{D^2}\left[I_1 - I_2 + I_3 - I_4\right] \\
&= c_1 + \frac{2(1-c_1)}{D^2}\left(DL_c + L_c^2\text{exp}(-\frac{D}{L_c}) - L_c^2\right) \qquad \text{(G.19)}
\end{aligned}$$

Variance function 1D Squared Exponential Covariance function

$$\begin{aligned}
\gamma(D) &= \frac{2}{D^2}\int_0^D (D-\Delta x)\rho(\Delta x)d\Delta x \\
&= \frac{2}{D^2}\int_0^D (D-\Delta x)\left[c_1+(1-c_1)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)\right]d\Delta x \\
&= \frac{2}{D^2}\int_0^D Dc_1 d\Delta x - \frac{2}{D^2}\int_0^D \Delta x c_1 d\Delta x \\
&+ \frac{2}{D^2}\int_0^D D(1-c_1)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)d\Delta x \\
&- \frac{2}{D^2}\int_0^D \Delta x(1-c_1)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)d\Delta x \\
&= \frac{2}{D^2}\left[I_1-I_2+I_3-I_4\right] \qquad \text{(G.20)}
\end{aligned}$$

The first and the second integral are already solved (G.16 and G.17). The third integral is solved by substituting $\frac{\Delta x}{L_c}$ by u. Thereafter the error functions, as defined in equation 2.24, is used.

$$\begin{aligned}
I_3 &= \int_0^D D(1-c_1)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)d\Delta x \\
&= D(1-c_1)L_c\int_0^{\frac{D}{L_c}} \text{exp}\left(-u^2\right)d\Delta x \\
&= \frac{D(1-c_1)L_c\sqrt{\pi}}{2}erf\left(\frac{D}{L_c}\right) \qquad \text{(G.21)}
\end{aligned}$$

To solve the fourth integral $\frac{\Delta x^2}{L_c^2}$ is substituted by u.

$$\begin{aligned}
I_4 &= \int_0^D \Delta x(1-c_1)\text{exp}\left(-\left(\frac{\Delta x}{L_c}\right)^2\right)d\Delta x \\
&= \frac{(1-c_1)L_c^2}{2}\int_0^{\frac{D^2}{L_c^2}} \text{exp}(-u)du \\
&= \frac{(1-c_1)L_c^2}{2}\left(1-\text{exp}\left(-\frac{D^2}{L_c^2}\right)\right) \qquad \text{(G.22)}
\end{aligned}$$

Now the solutions to the integrals are substituted in equation G.20 to find the expression for the variance function.

$$\begin{aligned}
\gamma(D) &= \frac{2}{D^2}\left[I_1-I_2+I_3-I_4\right] \\
&= c_1+\frac{(1-c_1)}{D^2}\left(L_c^2(\text{exp}(-\frac{D^2}{L_c^2})-1)+D\sqrt{\pi}L_c erf(\frac{D}{L_c})\right) \qquad \text{(G.23)}
\end{aligned}$$

SDF 2D Squared Exponential Covariance function First the one sided SDF for the Squared Exponential (SExp) covariance function is derived. The SExp covariance function is separable, i.e. $\rho(\Delta x_1, \Delta x_1) = \rho(\Delta x_1)\rho(\Delta x_2)$. According to Vanmarcke [6], the normalized spectral density function of which the covariance function is separable can be written as: $g(\omega_1, \omega_2) = g(\omega_1)g(\omega_2)$. The normalized spectral density function is defined as: $g(\omega) = \frac{1}{\sigma^2}G(\omega)$.

$$
\begin{aligned}
G(\omega_1,\omega_2) &= \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_0^\infty \int_0^\infty B(\Delta x_1, \Delta x_2)\text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2)d\Delta x_1 d\Delta x_2 \\
&= \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_0^\infty \int_0^\infty \left[c_1 + (1-c_1)\text{exp}\left(-\left((\frac{\Delta x_1}{L_{c,1}})^2 + \frac{\Delta x_2}{L_{c,2}})^2\right)\right)\right] \\
&\quad \text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2)d\Delta x_1 d\Delta x_2 \\
&= \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_0^\infty \int_0^\infty c_1\text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2)d\Delta x_1 d\Delta x_2 \\
&+ \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_0^\infty \int_0^\infty (1-c_1)\text{exp}\left[-\left((\frac{\Delta x_1}{L_{c,1}})^2 + \frac{\Delta x_2}{L_{c,2}})^2\right)\right] \\
&\quad \text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2)d\Delta x_1 d\Delta x_2 \\
&= \sigma^2 g_1(\omega_1,\omega_2) + \sigma^2 g_2(\omega_1,\omega_2)
\end{aligned}
\tag{G.24}
$$

For both $g_1(\omega_1,\omega_2)$ and $g_2(\omega_1,\omega_2)$ the covariance kernel can be separated as follows:

$$
\begin{aligned}
\rho_1(\Delta x_1, \Delta x_2) &= c1 \\
&\rightarrow \rho_1(\Delta x_1)\rho_1(\Delta x_2) = \sqrt{c_1}\sqrt{c_1}
\end{aligned}
\tag{G.25}
$$

$$
\begin{aligned}
\rho_2(\Delta x_1, \Delta x_2) &= (1-c_1)\text{exp}\left[-\left((\frac{\Delta x_1}{L_{c,1}})^2 + \frac{\Delta x_2}{L_{c,2}})^2\right)\right] \\
&\rightarrow \rho_2(\Delta x_1)\rho_2(\Delta x_2) = (1-c_1)\text{exp}\left(-(\frac{\Delta x_1}{L_{c,1}})^2\right)\text{exp}\left(-(\frac{\Delta x_2}{L_{c,2}})^2\right)
\end{aligned}
\tag{G.26}
$$

The SDF for the SExp covariance function in 2d then becomes equal to:

$$
\begin{aligned}
G(\omega_1,\omega_2) &= \sigma^2 g_1(\omega_1,\omega_2) + \sigma^2 g_2(\omega_1,\omega_2) \\
&= \sigma^2\left(2\sqrt{c_1}\delta(\omega_1)\right)\left(2\sqrt{c_1}\delta(\omega_2)\right) \\
&+ \left[\frac{(1-c_1)L_{c,1}}{\sqrt{\pi}}\text{exp}\left(-\frac{\omega_1^2 L_{c,1}^2}{4}\right)\right]\left[\frac{L_{c,2}}{\sqrt{\pi}}\text{exp}\left(-\frac{\omega_2^2 L_{c,2}^2}{4}\right)\right] \\
&= 4\sigma^2 c_1\delta(\omega_1)\delta(\omega_2) \\
&+ \frac{\sigma^2(1-c_1)L_{c,1}L_{c,2}}{\pi}\text{exp}\left[-\left(\frac{\omega_1^2 L_{c,1}^2}{4} + \frac{\omega_2^2 L_{c,2}^2}{4}\right)\right]
\end{aligned}
\tag{G.27}
$$

SDF 2D Exponential Covariance function Now the one sided SDF for Exp is derived. First it is rewritten.

$$
\begin{aligned}
G(\omega_1,\omega_2) &= \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_0^\infty \int_0^\infty B(\Delta x_1,\Delta x_2)\text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2) d\Delta x_1 d\Delta x_2 \\
&= \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_{-\infty}^\infty \int_{-\infty}^\infty \left[c_1 + (1-c_1)\text{exp}\left(-\sqrt{(\frac{\Delta x_1}{L_{c,1}})^2 + \frac{\Delta x_2}{L_{c,2}})^2}\right)\right] \\
&\quad \text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2) d\Delta x_1 d\Delta x_2 \\
&= \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_0^\infty \int_0^\infty c_1 \text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2) d\Delta x_1 d\Delta x_2 \\
&+ \left(\frac{2}{\pi}\right)^2 \sigma^2 \int_0^\infty \int_0^\infty (1-c_1)\left(-\sqrt{(\frac{\Delta x_1}{L_{c,1}})^2 + \frac{\Delta x_2}{L_{c,2}})^2}\right) \\
&\quad \text{cos}(\omega_1\Delta x_1)\text{cos}(\omega_2\Delta x_2) d\Delta x_1 d\Delta x_2 \\
&= \sigma^2 g_1(\omega_1,\omega_2) + \sigma^2 g_2(\omega_1,\omega_2)
\end{aligned}
\tag{G.28}
$$

$g_1(\omega_1,\omega_2)$ is given in equation G.27, and is equal to $4c_1\delta(\omega_1)\delta(\omega_2)$. Now $g_2(\omega_1,\omega_2)$ is derived, of which the covariance kernel is not separable. First the one sided normalized SDF is first written as a two sided normalized SDF. Then the following substitutions are made.

$$
\Delta x_1' = \frac{\Delta x_1}{L_{c,1}}, \qquad \Delta x_2' = \frac{\Delta x_2}{L_{c,2}}, \qquad \omega_1' = \omega_1 L_{c,1}, \qquad \omega_2' = \omega_2 L_{c,2}
\tag{G.29}
$$

Which results in:

$$
\begin{aligned}
g_2(\omega_1',\omega_2') &= \frac{4}{(2\pi)^2}\sigma^2(1-c_1)L_{c,1}L_{c,2}\int_{-\infty}^\infty \int_{-\infty}^\infty \text{exp}\left(-(\Delta x_1' + \Delta x_2')^2\right) \\
&\quad \text{exp}\left(-i(\omega_1'\Delta x_1' + \omega_2'\Delta x_2')\right) d\Delta x_1' d\Delta x_2'
\end{aligned}
\tag{G.30}
$$

Then the variables are substituted by polar coordinates and then the equation is rewritten.

$$
\begin{aligned}
g_2(\omega_1',\omega_2') &= \frac{4}{(2\pi)^2}\sigma^2(1-c_1)L_{c,1}L_{c,2}\int_0^\infty \int_0^{2\pi} r\,\text{exp}\left(-r\right) \\
&\quad \text{exp}\left(-ir\zeta\text{cos}(\theta-\phi)\right) d\theta dr \\
&= \frac{2}{(\pi)}\sigma^2(1-c_1)L_{c,1}L_{c,2}\int_0^\infty r\,\text{exp}\left(-r\right) J_0(r\zeta) dr
\end{aligned}
\tag{G.31}
$$

Where J_0 is the zero order Bessel function of the first kind and $\zeta^2 = \omega_1'^2 + \omega_2'^2$. The solution of this integral can be found in [48] (page 702, equation 6.623-2) and is equal to:

$$\begin{aligned} g_2(\omega_1, \omega_2) &= \frac{2}{(\pi)}\sigma^2(1-c_1)L_{c,1}L_{c,2}\int_0^\infty r \exp(-r)\, J_0(r\zeta)dr \\ &= \frac{2}{(\pi)}\sigma^2(1-c_1)L_{c,1}L_{c,2}\frac{2\Gamma(3/2)}{\sqrt{\pi}(1+\zeta^2)^{\frac{3}{2}}} \\ &= \frac{2\sigma^2(1-c_1)L_{c,1}L_{c,2}}{\pi(1+\zeta^2)^{\frac{3}{2}}} \\ &= \frac{2\sigma^2(1-c_1)L_{c,1}L_{c,2}}{\pi(1+\omega_1^2L_{c,1}^2+\omega_2^2L_{c,2}^2)^{\frac{3}{2}}} \end{aligned} \tag{G.32}$$

The SDF for the Exp covariance function in 2d then becomes equal to:

$$\begin{aligned} G(\omega_1, \omega_2) &= \sigma^2 g_1(\omega_1, \omega_2) + \sigma^2 g_2(\omega_1, \omega_2) \\ &= 4\sigma^2 c_1 \delta(\omega_1)\delta(\omega_2) + \frac{2\sigma^2(1-c_1)L_{c,1}L_{c,2}}{\pi(1+\omega_1^2L_{c,1}^2+\omega_2^2L_{c,2}^2)^{\frac{3}{2}}} \end{aligned} \tag{G.33}$$

H

Matlab code: Gauss quadrature to determine covariance and variance according local average theory

```
function vario = varfn_Gauss(CF,distribution,Dx,lag,L_c,c1,var)
% Numeric integration to determine covariances of two local averages
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Based on subroutine dcvab1.f and dcvaa1.f of RFEM software which is written by
% Gordon A. Fenton. Free available on http://courses.engmath.dal.ca/rfem/

ng=16;
z=[-.989400934991649932596; -.944575023073232576078; -.865631202387831743880; ...
    -.755404408355003033895; -.617876244402643748447; -.458016777657227386342; ...
    -.281603550779258913230; -.095012509837637440185; 0.095012509837637440185; ...
    0.281603550779258913230; 0.458016777657227386342; 0.617876244402643748447; ...
    0.755404408355003033895; 0.865631202387831743880; 0.944575023073232576078; ...
    0.989400934991649932596];
w=[0.027152459411754094852; 0.062253523938647892863; 0.095158511682492784810; ...
    0.124628971255533872052; 0.149595988816576732081; 0.169156519395002538189; ...
    0.182603415044923588867; 0.189450610455068496285; 0.189450610455068496285; ...
    0.182603415044923588867; 0.169156519395002538189; 0.149595988816576732081; ...
    0.124628971255533872052; 0.095158511682492784810; 0.062253523938647892863; ...
    0.027152459411754094852];

r1=0.5*Dx;
d1=0;
if lag==0
    for i=1:ng
        xi=r1*(1+z(i));
        if strcmp(distribution,'normal')==1
            if strcmp(CF,'Exp')==1
                d1=d1+w(i)*(1-z(i))*(c1+(1-c1)*exp(-xi/L_c));
            elseif strcmp(CF,'SExp')==1
                d1=d1+w(i)*(1-z(i))*(c1+(1-c1)*exp(-(xi/L_c)^2));
            else
                Disp('Correlation function not programmed')
            end
        elseif strcmp(distribution,'lognormal')==1
            if strcmp(CF,'Exp')==1
                d1=d1+w(i)*(1-z(i))*((exp(var*(c1+(1-c1)*exp(-xi/L_c)))-1)/(exp(var)-1));
            elseif strcmp(CF,'SExp')==1
                d1=d1+w(i)*(1-z(i))*((exp(var*(c1+(1-c1)*exp(-(xi/L_c)^2)))-1)/(exp(var)-1));
            else
                Disp('Correlation function not programmed')
            end
        else
            Disp('Distribution type not programmed')
        end
    end
    vario=0.5*d1;
else
    s1=2*lag-1;
    s2=2*lag+1;
    for i=1:ng
        xi1=r1*(z(i)+s1);
        xi2=r1*(z(i)+s2);
        if strcmp(distribution,'normal')==1
            if strcmp(CF,'Exp')==1
                d1=d1+w(i)*((1+z(i))*(c1+(1-c1)*exp(-xi1/L_c)) ...
                    +(1-z(i))*(c1+(1-c1)*exp(-xi2/L_c)));
            elseif strcmp(CF,'SExp')==1
                d1=d1+w(i)*((1+z(i))*(c1+(1-c1)*exp(-(xi1/L_c)^2)) ...
                     +(1-z(i))*(c1+(1-c1)*exp(-(xi2/L_c)^2)));
            else
                Disp('Correlation function not programmed')
            end
        elseif strcmp(distribution,'lognormal')==1
            if strcmp(CF,'Exp')==1
                d1=d1+w(i)*((1+z(i))*((exp(var*(c1+(1-c1)*exp(-xi1/L_c)))-1)/(exp(var)-1)) ...
                     +(1-z(i))*((exp(var*(c1+(1-c1)*exp(-xi2/L_c)))-1)/(exp(var)-1)));
            elseif strcmp(CF,'SExp')==1
                d1=d1+w(i)*((1+z(i))*((exp(var*(c1+(1-c1)*exp(-(xi1/L_c)^2)))-1)/(exp(var)-1)) ...
                     +(1-z(i))*((exp(var*(c1+(1-c1)*exp(-(xi2/L_c)^2)))-1)/(exp(var)-1)));
            else
                Disp('Correlation function not programmed')
            end
        else
            Disp('Distribution type not programmed')
        end
    end
    vario=0.25*d1;
end

end
```

Matlab codes random field generators

In this appendix, the different Matlab codes are given for the random field generators which are assessed in chapter 3. The following codes can be found in this appendix:

- Covariance Matrix Decomposition 1D
- Covariance Matrix Decomposition 2D
- Fast Fourier Transform method in 1D
- Fast Fourier Transform method in 2D
- Local Average Subdivision in 1D
- Local Average Subdivision in 2D
- Expansion Optimal Linear Estimation in 1D
- Expansion Optimal Linear Estimation in 2D
- Modified Cholesky decomposition
- Inverse Fourier Transform algorithm

I.1 Matlab code: Covariance Matrix Decomposition 1D

```
% CMD_1D returns correlated random variables for a homogeneous gaussain field
% in 1D applying the Covariance Matrix Decomposition method
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:       Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_C:        Correlation length/scale of fluctuation
%   D:          Size random field
%   N:          Number of nodes in random field
%   Dec:        Decomposition method (chol/eigen)
%   Var_fn:     Reduction of variance applied (yes/no)

clc, close all, clear all
t=cputime;

% Input values:
runs=2000;
dim=1;
mean_RF=0;
std_RF=1;
CF='SExp';
c1=0;
L_c=2.5;
D=17.5;
N=64;
Dec='mod chol';
Var_fn='no';
distribution='normal';

Mesh=linspace(0,D,N);                   % Random field Mesh
RF=zeros(length(Mesh),runs);            % Allocate matrix voor values RF
var=std_RF^2;

%Determine Correlation matrix
Cor_M=zeros(length(Mesh),length(Mesh));
if strcmp(Var_fn,'yes')==1
   dx=D/(N-1);
   Cov=zeros(N,1);
   for i=1:N
        lag=i-1;
        Cov(i)=varfn_Gauss(CF,distribution,dx,lag,L_c,c1,var);
    end
    for i=1:length(Mesh)
        for j=1:i
            Cor_M(i,j)=Cov(i-j+1);
        end
    end
    Cor_M=Cor_M+Cor_M'-diag(diag(Cor_M));
elseif strcmp(Var_fn,'no')==1
    if strcmp(CF,'Exp')==1
        for i=1:length(Mesh)
            for j=1:length(Mesh)
                Cor_M(i,j)=c1+(1-c1)*exp(-abs((Mesh(i)-Mesh(j)))/L_c);
            end
        end
    elseif strcmp(CF,'SExp')==1
        for i=1:length(Mesh)
            for j=1:length(Mesh)
                Cor_M(i,j)=c1+(1-c1)*exp(-((abs(Mesh(i)-Mesh(j)))/L_c)^2);
            end
        end
    else
```

```
        disp('CF must be equal to Exp or SExp')
    end
else
    disp('Var_fn must be equal to yes or no')
end

if strcmp(distribution,'lognormal')==1
    std_norm=sqrt(log(1+(std_RF/mean_RF)^2));
    mean_norm=log(mean_RF)-0.5*std_norm^2;
    Cor_M=(exp(Cor_M*std_norm^2)-1)/(exp(std_norm^2)-1);
end

% Decomposition correlation matrix
if strcmp(Dec,'chol')==1
    L=chol(Cor_M,'lower');
elseif strcmp(Dec,'eigen')==1
    [vec,lambda]=eig(Cor_M);
    L=vec*sqrt(abs(lambda));
elseif strcmp(Dec,'mod chol')==1
    L=chol2(Cor_M);
    L=L';
else
    disp('Dec must be equal to chol, mod chol or eigen')
end

it=cputime-t;

% Generation random fields
RV=randn(length(Mesh),runs);       % Standard normal random variables

for j=1:runs
    if strcmp(distribution,'normal')==1
        RF(:,j)=mean_RF+L*RV(:,j)*std_RF;    % Generate random field
    elseif strcmp(distribution,'lognormal')==1
        RF(:,j)=exp(mean_norm+L*RV(:,j)*std_norm);    % Generate random field
    end
end
rt=cputime-t;

% plot of 1st random field
figure()
plot(Mesh,RF(:,1))

% Determin statistical properties random field
[mean_m,mean_s,std_m,std__s,c_err_m,c_err_s]=stat_prop(RF,runs,dim,Mesh,CF,c1,L_c);
st=cputime-t;

table=[std_RF it rt st mean_m mean_s std_m std__s c_err_m c_err_s];
```

I.2 Matlab code: Covariance Matrix Decomposition 2D

```matlab
% CMD_2D returns correlated random variables for a homogeneous gaussain field
% in 2D applying Covariance Matrix decomposition
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%               in x-direction
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields in x-direction
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:         Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_cx:       Correlation length/scale of fluctuation in x-direcion
%   L_cy:       Correlation length/scale of fluctuation in y-direcion
%   Dx:         Size random field in x direction
%   Dy:         Size random field in y direction
%   Nx:         Number of nodes in random field in y direction
%   Ny:         Number of nodes in random field in y direction
%   Dec:        Decomposition method (chol/eigen)

clc, close all, clear all
t=cputime;

% Input values:
runs=200;
dim=2;
mean_RF=0;
std_RF=1;
CF='Exp';
c1=0.5;
L_cx=2.5;
L_cy=2.5;
Dx=17.5;
Dy=17.5;
Nx=32;
```

```
Ny=32;
Dec='eigen';
Var_fn='no';
distribution='normal';

Meshxy=zeros(Nx*Ny,2);                    % Allocate matrix voor RF-mesh
CRV=zeros(length(Meshxy),runs);           % Allocate matrix voor values RF
RF=zeros(Nx,Ny,runs);

for i=1:Nx
    for j=1:Ny
        Meshxy((j-1)*Nx+i,1:2)=[(i-1)*Dx/(Nx-1) (j-1)*Dy/(Ny-1)];
    end
end
Meshx=linspace(0,Dx,Nx);            % X-Coordinates
Meshy=linspace(0,Dy,Ny);            % Y-Coordinates
var=std_RF^2;

%Determine Correlation matrix
Cor_M=zeros(length(Meshxy),length(Meshxy));
if strcmp(Var_fn,'yes')==1
    ii=0;
    dx=Dx/(Nx-1);
    dy=Dy/(Ny-1);
    for j=1:Ny
        lagy=j-1;
        for i=1:Nx
            ii=ii+1;
            lagx=i-1;
            Cor_M(1,ii)=varfn2D_Gauss(CF,distribution,dx,dy,lagx,lagy,L_cx,L_cy,c1,var);
        end
    end
    for j=2:length(Meshxy)
        mxj=mod(j-1,Nx);
        myj=floor((j-1)/Nx);
        for i=2:j
            mxi=mod(i-1,Nx);
            myi=floor((i-1)/Nx);
            m=1+(abs(mxj-mxi))+Nx*(abs(myj-myi));
            Cor_M(i,j)=Cor_M(1,m);
        end
    end
    Cor_M=Cor_M+Cor_M'-diag(diag(Cor_M));
elseif strcmp(Var_fn,'no')==1
    if strcmp(CF,'Exp')==1
        for i=1:length(Meshxy)
            for j=1:length(Meshxy)
                Cor_M(i,j)=c1+(1-c1)*exp(-sqrt(((Meshxy(i,1)-Meshxy(j,1))/L_cx)^2 ...
                    +((Meshxy(i,2)-Meshxy(j,2))/L_cy)^2));
            end
        end
    elseif strcmp(CF,'SExp')==1
        for i=1:length(Meshxy)
            for j=1:length(Meshxy)
                Cor_M(i,j)=c1+(1-c1)*exp(-(((Meshxy(i,1)-Meshxy(j,1))/L_cx)^2 ...
                    +((Meshxy(i,2)-Meshxy(j,2))/L_cy)^2));
            end
        end
    else
        disp('CF must be equal to Exp or SExp')
    end
else
```

```
    disp('Var_fn must be equal to yes or no')
end

if strcmp(distribution,'lognormal')==1
    std_norm=sqrt(log(1+(std_RF/mean_RF)^2));
    mean_norm=log(mean_RF)-0.5*std_norm^2;
    Cor_M=(exp(Cor_M*std_norm^2)-1)/(exp(std_norm^2)-1);
end

% Decomposition correlation matrix
if strcmp(Dec,'chol')==1
    L=chol(Cor_M,'lower');
elseif strcmp(Dec,'eigen')==1
    [vec,lambda]=eig(Cor_M);
    L=vec*sqrt(abs(lambda));
elseif strcmp(Dec,'mod chol')==1
    L=chol2(Cor_M);
    L=L';
else
    disp('Dec must be equal to chol, mod chol or eigen')
end

it=cputime-t;

% Generation random fields
RV=randn(length(Meshxy),runs);

for j=1:runs
    if strcmp(distribution,'normal')==1
        CRV(:,j)=mean_RF+L*RV(:,j)*std_RF;   % Generate random field
    elseif strcmp(distribution,'lognormal')==1
        CRV(:,j)=exp(mean_norm+L*RV(:,j)*std_norm);   % Generate random field
    end
end
for j=1:runs
    for i=1:Ny
        RF(:,i,j)=CRV((i-1)*Nx+1:i*Nx,j);
    end
end
rt=cputime-t;

% plot of 1st random field
figure()
surf(Meshx,Meshy,RF(:,:,1)')

% Determin statistical properties random field
Mesh=Meshx;
L_c=[L_cx; L_cy];
[mean_m,mean_s,std_m,std_s,c_err_m,c_err_s]=stat_prop(RF,runs,dim,Mesh,CF,c1,L_c);
st=cputime-t;
table=[it rt st mean_m mean_s std_m std_s c_err_m c_err_s];
```

I.3 Matlab code: Fast Fourier Transform method in 1D

```
% FFT_1D  returns correlated random variables for a homogeneous gaussain field
% in 1D applying the Fast Fourier Transformation method
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:         Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_c:        Correlation length/scale of fluctuation
%   D:          Size random field
%   N:          Number of nodes in random field
%   Var_fn:     Reduction of variance applied (yes/no)

clc,clear,close all
t=cputime;

% Input values:
runs=1;
dim=1;
mean_RF=30.5;
std_RF=1;
CF='SExp';
c1=0.5;
L_c=5;
D=35;
N=16;
distribution='lognormal';
doublesize='yes';

if strcmp(distribution,'normal')==1
    var=std_RF^2;            % Variance of random field
elseif strcmp(distribution,'lognormal')==1
```

```
    std_norm=sqrt(log(1+(std_RF/mean_RF)^2));
    mean_norm=log(mean_RF)-0.5*std_norm^2;
    var=std_norm^2;
end

if strcmp(doublesize,'yes')==1
    N2=N*2;
    D2=D*2+D/(N-1);
elseif strcmp(doublesize,'no')==1
    N2=N;
    D2=D;
end

Mesh=linspace(0,D,N);               % X-Coordinates
delta_w=2*pi*(N2-1)/(N2*D2);        % Delta w (width of a inteval in power spectrum density function)
w=linspace(0,delta_w*(N2-1),N2);    % Coordinates w for power spectrum density function
RF=zeros(N2,runs);                  % Allocate matrix voor values RF

% Determine one sided SDF
if strcmp(CF,'Exp')==1
    G=(1-c1)*2*var*L_c./(pi*(1+L_c^2*w.^2));
elseif strcmp(CF,'SExp')==1
    G=(1-c1)*var*L_c./(sqrt(pi)*exp(L_c^2*w.^2/4));
else
    disp('CF must be equal to Exp or SExp')
end

% Plot of power spectrum density function
figure()
plot (w,G)

% Determening the standard deviation of coefficients Ak and Bk
V=2:N2/2;

Var_A(V)=0.25*(G(V)+G(N2+2-V))*delta_w;

if strcmp(distribution,'normal')
    Var_A(1)=G(1)*0.5*delta_w+c1*var;         % Addition for dirac delta function c1(w)*std^2
elseif strcmp(distribution,'lognormal')
    Var_A(1)=G(1)*0.5*delta_w+c1*var;
end
Var_A(N2/2+1)=G(N2/2+1)*delta_w;
std_A=sqrt(Var_A);

it=cputime-t;
% Generation random fields
for i=1:runs                        % Loop for every random field
    Z_CRV=zeros(N2,1);               % Starting values loop
    Ak=zeros(N2,1);
    Bk=zeros(N2,1);
    for k=1:N2/2+1
        Ak(k)=normrnd(0,std_A(k));      % Determine Ak coefficients
        Bk(k)=normrnd(0,std_A(k));      % Determine Bk coefficients
    end
    Bk(1)=0;
    Bk(N2/2+1)=0;
    Ak(N2/2+V)=Ak(fliplr(V));            % Make use of symmetry condition
    Bk(N2/2+V)=-Bk(fliplr(V));

    % Determine values of random field at x-coordinates
    [Ak,Bk]=invFFT1D(Ak,Bk,log2(N2));

    % tranformation of standard normal gaussian field
    if strcmp(distribution,'normal')==1
        RF(:,i)=mean_RF+Ak(:,1);
    elseif strcmp(distribution,'lognormal')==1
        RF(:,i)=exp(Ak(:,1)+mean_norm);
    end
end
rt=cputime-t;

% plot of 1st random field
figure()
plot(Mesh,RF(1:N,1))

% Determin statistical properties random field
[mean_m,mean_s,std_m,std__s,c_err_m,c_err_s]=stat_prop(RF(1:N,1:runs),runs,dim,Mesh,CF,c1,L_c); % mean and std RF
st=cputime-t;
table=[std_RF it rt st mean_m mean_s std_m std__s c_err_m c_err_s];
```

I.4 Matlab code: Fast Fourier Transform method in 2D

```
% FFT_2D returns correlated random variables for a homogeneous gaussain field
% in 2D applying the Fast Fourier Transform method
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%               in x-direction
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields in x-direction
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:         Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_c:        Correlation length/scale of fluctuation in both directions
%   Dx:         Size random field in x direction
%   Dy:         Size random field in y direction
%   Nx:         Number of nodes in random field in y direction
%   Ny:         Number of nodes in random field in y direction

clc, close all, clear all
t=cputime;

%% Input values:
runs=200;
dim=2;
mean_RF=30.5;
std_RF=5.9;
CF='SExp';
c1=0;
L_cx=5;
L_cy=5;
Dx=40;
Dy=40;
Nx=32;
Ny=32;
doublesize='yes';
```

```
distribution='lognormal';

if strcmp(distribution,'normal')==1
    var=std_RF^2;            % Variance of random field
elseif strcmp(distribution,'lognormal')==1
    std_norm=sqrt(log(1+(std_RF/mean_RF)^2));
    mean_norm=log(mean_RF)-0.5*std_norm^2;
    var=std_norm^2;
end

if strcmp(doublesize,'yes')==1
    Nx2=Nx*2;
    Ny2=Ny*2;
    Dx2=Dx*2+Dx/(Nx-1);
    Dy2=Dy*2+Dy/(Ny-1);
elseif strcmp(doublesize,'no')==1
    Nx2=Nx;
    Ny2=Ny;
    Dx2=Dx;
    Dy2=Dy;
end

RF=zeros(Nx2,Ny2,runs);
Meshx=linspace(0,Dx,Nx);             % X-Coordinates
Meshy=linspace(0,Dy,Ny);             % Y-Coordinates
delta_wx=2*pi*(Nx2-1)/(Nx2*Dx2);     % Delta w
delta_wy=2*pi*(Ny2-1)/(Ny2*Dy2);     % Delta w
wx=linspace(0,delta_wx*(Nx2-1),Nx2); % Coordinates w for power spectrum density function
wy=linspace(0,delta_wy*(Ny2-1),Ny2); % Coordinates w for power spectrum density function

%% Determine one sided SDF
G_c=zeros(Nx2,Ny2);
if strcmp(CF,'Exp')==1
    for i=1:Nx2
        for j=1:Ny2
            G_c(i,j)=2*(1-c1)*var*L_cx*L_cy/(pi*(1+wx(i)^2*L_cx^2+wy(j)^2*L_cy^2)^1.5); % spectral density function of Exp with c1=0
        end
    end
elseif strcmp(CF,'SExp')==1
    for i=1:Nx2
        for j=1:Ny2
            G_c(i,j)=(1-c1)*var*L_cx*L_cy/(pi*exp(wx(i)^2*L_cx^2/4+wy(j)^2*L_cy^2/4));
        end
    end
else
    disp('CF must be equal to Exp or SExp')
end

G=G_c;
G(1,:)=G(1,:).*0.5;
G(:,1)=G(:,1).*0.5;
delta_w=delta_wx*delta_wy;

% Plot of one sided SDF
figure()
surf(wx,wy,G_c')

%% Determening the standard deviation of coefficients Ak and Bk
Var=zeros(Nx2,Ny2/2+1);
for i=2:Nx2
    for j=2:Ny2/2
```

```
        Var(i,j)=1/8*delta_w*(G(i,j)+G(i,Ny2+2-j)+G(Nx2+2-i,j)+G(Nx2+2-i,Ny2+2-j));
    end
end
i=1;
for j=2:Ny2/2
    Var(i,j)=1/4*delta_w*(G(i,j)+G(i,Ny2-j+2));
end
i=1+Nx2/2;
for j=2:Ny2/2
    Var(i,j)=1/4*delta_w*(G(i,j)+G(i,Ny2-j+2));
end
j=1;
for i=2:Nx2/2
    Var(i,j)=1/4*delta_w*(G(i,j)+G(Nx2-i+2,j));
end
j=1+Ny2/2;
for i=2:Nx2/2
    Var(i,j)=1/4*delta_w*(G(i,j)+G(Nx2-i+2,j));
end
Var(1,1)=delta_w*G(1,1)+var*c1;
Var(1+Nx2/2,1)=delta_w*G((1+Nx2/2),1);
Var(1,1+Ny2/2)=delta_w*G(1,1+Ny2/2);
Var(1+Nx2/2,1+Ny2/2)=delta_w*G(1+Nx2/2,1+Ny2/2);

std_coef=sqrt(Var);

it=cputime-t;

%% Generation random fields
RF=zeros(Nx2,Ny2,runs);                  % Matrix for realisations of random field
for run=1:runs                           % Loop for every random field
    Ak=zeros(Nx2,Ny2);                   % Starting values loop
    Bk=zeros(Nx2,Ny2);
for j=1:Nx2
    for k=1:Ny2/2+1
        Ak(j,k)=normrnd(0,std_coef(j,k));      % Determine Ak coefficients
        Bk(j,k)=normrnd(0,std_coef(j,k));      % Determine Bk coefficients
    end
end
Bk(1,1)=0;
Bk(Nx2/2+1,1)=0;
Bk(1,Ny2/2+1)=0;
Bk(Nx2/2+1,Ny2/2+1)=0;

Ak1=rot90(Ak(2:Nx2,2:Ny2),2);
Ak(2:Nx2,Ny2/2+2:Ny2)=Ak1(1:Nx2-1,Ny2/2+1:Ny2-1);
Ak(Nx2/2+1:Nx2,Ny2/2+1)=Ak1(Nx2/2:Nx2-1,Ny2/2);
Ak2=flipud(Ak(2:Nx2/2,1));
Ak3=fliplr(Ak(1,2:Ny2/2));
Ak(Nx2/2+2:Nx2,1)=Ak2;
Ak(1,Ny2/2+2:Ny2)=Ak3;

Bk1=rot90(Bk(2:Nx2,2:Ny2),2);
Bk(2:Nx2,Ny2/2+2:Ny2)=-Bk1(1:Nx2-1,Ny2/2+1:Ny2-1);
Bk(Nx2/2+1:Nx2,Ny2/2+1)=-Bk1(Nx2/2:Nx2-1,Ny2/2);
Bk2=flipud(Bk(2:Nx2/2,1));
Bk3=fliplr(Bk(1,2:Ny2/2));
Bk(Nx2/2+2:Nx2,1)=-Bk2;
Bk(1,Ny2/2+2:Ny2)=-Bk3;

m1=log2(Nx2);
m2=log2(Ny2);
```

```
ta=zeros(Ny2,1);
tb=zeros(Ny2,1);
for i=1:Nx2
    for j=1:Ny2
        ta(j)=Ak(i,j);
        tb(j)=Bk(i,j);
    end
    [ta,tb]=invFFT1D(ta,tb,m2);
    for j=1:Ny2
        Ak(i,j)=ta(j);
        Bk(i,j)=tb(j);
    end
end
for j=1:Ny2
    [Ak(:,j),Bk(:,j)]=invFFT1D(Ak(:,j),Bk(:,j),m1);
end

if strcmp(distribution,'normal')==1
    RF(:,:,run)=mean_RF+Ak;
elseif strcmp(distribution,'lognormal')==1
    RF(:,:,run)=exp(mean_norm+Ak);

end
end
rt=cputime-t;
%% Plot the random field of the first run
figure()
surf(Meshx,Meshy,RF(1:Nx,1:Ny,1)')

% Determin statistical properties random field
Mesh=Meshx;
[mean_m,mean_s,std_m,std_s,c_err_m,c_err_s] ...
    =stat_prop(RF(1:Nx,1:Ny,1:runs),runs,dim,Mesh,CF,c1,L_cx);     % mean and std RF
st=cputime-t;

table=[it rt st mean_m mean_s std_m std_s c_err_m c_err_s];
```

I.5 Matlab code: Local Average Subdivision in 1D

```
% LAS_1D  returns correlated random variables for a homogeneous gaussain field
% in 1D applying the Local Average Subdivision method
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Based on subroutine las1g.f of RFEM software which is written by
% Gordon A. Fenton. Free available on http://courses.engmath.dal.ca/rfem/
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:         Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_c:        Correlation length/scale of fluctuation
%   D:          Size random field
%   m:          Number of devisions
%   k1:         Number of cells for stage 0
%   nbh:        Neigberhood cells (3 or 5)

clc,clear,close all
t=cputime;

% Input values:
runs=2000;
dim=1;
mean_RF=1;
std_RF=1;
CF='SExp';
c1=0.5;
L_c=5;
D=17.5;
m=1;
k1=8;
nbh=3;
```

```
distribution='lognormal';

if strcmp(distribution,'normal')==1
    var=std_RF^2;            % Variance of random field
elseif strcmp(distribution,'lognormal')==1
    std_norm=sqrt(log(1+(std_RF/mean_RF)^2));
    mean_norm=log(mean_RF)-0.5*std_norm^2;
    var=std_norm^2;
end

N=k1*2^m;                   % Number of Cells of random field
RF=zeros(N,runs);           % Allocate matrix voor values RF

%% Determine initial values
% Generate RF values for stage 0
Dx=D/k1;                    % Cell size stage 0

% Determine Correlation matrix stage 0 using the variance function
Cov=zeros(k1,1);
for i=1:k1
    lag=i-1;
    Cov(i)=varfn_Gauss(CF,distribution,Dx,lag,L_c,c1,var);
end
Cor_m=zeros(k1,k1);
for j=1:k1
    for i=1:j
        Cor_m(j,i)=Cov(j-i+1);
    end
end
L=chol(Cor_m,'lower');      % Decomposition correlation matrix

% Determine a_l and c coefficients
Cov=zeros(nbh,1);           % Allocate vector for covariances
C_Cov=zeros(nbh,1);         % Allocate vector for cross-covariances
Cor_M=zeros(nbh,nbh);       % Allocate matrix for correlation matrix
A=zeros(3,m);               % Allocate matrix for a_l coefficients
c=zeros(m,1);               % Allocate vector for c coefficients
for j=1:m
    % Determin Correlation matrix for stage m
    for i=1:nbh
        lag=i-1;
        Cov(i)=varfn_Gauss(CF,distribution,Dx,lag,L_c,c1,var);
    end
    for l=1:nbh
        for k=1:l
            Cor_M(k,l)=Cov(l-k+1);
            Cor_M(l,k)=Cov(l-k+1);
        end
    end
    Dx=Dx/2;
    % Determin cross-covariances for stage m
    for i=1:nbh
        lag1=abs(i*2-nbh-2);
        lag2=abs(i*2-nbh-1);
        C_Cov1=varfn_Gauss(CF,distribution,Dx,lag1,L_c,c1,var);
        C_Cov2=varfn_Gauss(CF,distribution,Dx,lag2,L_c,c1,var);
        C_Cov(i)=0.5*C_Cov1+0.5*C_Cov2;
    end
    % Collect a_l and c coefficients
    A(1:nbh,j)=Cor_M\C_Cov;
    c(j)=varfn_Gauss(CF,distribution,Dx,0,L_c,c1,var)-A(:,j)'*C_Cov;
end
```

```
cc=sqrt(c);
it=cputime-t;

%% Generation random fields
RV=randn(N,runs);             % k1*2^m random variables are needed
for k=1:runs
    z=zeros(N,1);
    z(1:k1)=L*RV(1:k1,k);
    % Exepion if 1 cell at stage 0
    if k1==1
        z(2)=A(nbh/2+0.5,1)*z(1)+RV(2,k)*cc(1);
        z(1)=2*z(1)-z(2);
    else
        z_p=z;
        j=1;
        for i=1:k1*2^(j-1)
            % Boundary values
            if i==1
                z(2*i)=A(nbh/2+0.5:nbh,j)'*z_p(i:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
            elseif i==2
                if nbh==3
                    z(2*i)=A(:,j)'*z_p(i-nbh/2+0.5:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
                else
                    z(2*i)=A(2:nbh,j)'*z_p(i:i+3)+RV(k1*2^(j-1)+i,k)*cc(j);
                end
            elseif i==k1*2^(j-1)-1
                if nbh==3
                    z(2*i)=A(:,j)'*z_p(i-nbh/2+0.5:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
                else
                    z(2*i)=A(1:4,j)'*z_p(i-3:i)+RV(k1*2^(j-1)+i,k)*cc(j);
                end
            elseif i==k1*2^(j-1)
                z(2*i)=A(1:nbh/2+0.5,j)'*z_p(i-nbh/2+0.5:i)+RV(k1*2^(j-1)+i,k)*cc(j);
            % Internal values
            else
                z(2*i)=A(:,j)'*z_p(i-nbh/2+0.5:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
            end
            z(2*i-1)=2*z_p(i)-z(2*i);
        end
    end
    if m>=2
    for j=2:m
        z_p=z;
        for i=1:k1*2^(j-1)
            % Boundary values
           if i==1
               z(2*i)=A(nbh/2+0.5:nbh,j)'*z_p(i:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
           elseif i==2
               if nbh==3
                   z(2*i)=A(:,j)'*z_p(i-nbh/2+0.5:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
               else
                   z(2*i)=A(2:nbh,j)'*z_p(i:i+3)+RV(k1*2^(j-1)+i,k)*cc(j);
               end
           elseif i==k1*2^(j-1)-1
               if nbh==3
                   z(2*i)=A(:,j)'*z_p(i-nbh/2+0.5:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
               else
                   z(2*i)=A(1:4,j)'*z_p(i-3:i)+RV(k1*2^(j-1)+i,k)*cc(j);
               end
           elseif i==k1*2^(j-1)
                z(2*i)=A(1:nbh/2+0.5,j)'*z_p(i-nbh/2+0.5:i)+RV(k1*2^(j-1)+i,k)*cc(j);
           % Internal values
           else
                z(2*i)=A(:,j)'*z_p(i-nbh/2+0.5:i+nbh/2-0.5)+RV(k1*2^(j-1)+i,k)*cc(j);
           end
            z(2*i-1)=2*z_p(i)-z(2*i);
        end
    end
    end

    % tranformation of standard normal gaussian field
    if strcmp(distribution,'normal')==1
        RF(:,k)=std_RF*z+mean_RF;
    elseif strcmp(distribution,'lognormal')==1
        RF(:,k)=exp(std_norm*z+mean_norm);
    end

end
rt=cputime-t;

% plot of 1st random field
figure()
Mesh=linspace(0,D-Dx,N);        % X-Coordinates
plot(Mesh,RF(:,1))

% Determin statistical properties random field
[mean_m,mean_s,std_m,std__s,c_err_m,c_err_s] ...
    =stat_prop(RF,runs,dim,Mesh,CF,c1,L_c);     % mean and std RF
st=cputime-t;
table=[it rt st mean_m mean_s std_m std__s c_err_m c_err_s];
```

I.6 Matlab code: Local Average Subdivision in 2D

```
% LAS_2D returns correlated random variables for a homogeneous gaussain field
% in 2D applying the Local Average Subdivision
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Based on subroutine las2g.f of RFEM software which is written by
% Gordon A. Fenton. Free available on http://courses.engmath.dal.ca/rfem/
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%               in x-direction
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields in x-direction
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:         Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_cx:       Correlation length/scale of fluctuation in x-direcion
%   L_cy:       Correlation length/scale of fluctuation in y-direcion
%   Dx:         Size random field in x direction
%   Dy:         Size random field in y direction
%   mm:         Number of devisions
%   k1:         Number of cells in x direction for stage 0
%   k2:         Number of celss in y direction for stage 0
%   nbh:        Neighborhood cells
%   Int:        Exact integration or Gauss integration for varaince
%               function (Exact/Gauss) Exact only works for SEXP with c1=0

clc, close all, clear all
t=cputime;

%% Input values:
runs=200;
dim=2;
mean_RF=30.5;
std_RF=5.9;
CF='SExp';
```

```
c1=0.5;
L_cx=5;
L_cy=5;
Dx=17.5;
Dy=17.5;
k1=8;
k2=8;
mm=2;
nbh=3;
Int='Gauss';
distribution='normal';
if strcmp(distribution,'normal')==1
    var=std_RF^2;            % Variance of random field
elseif strcmp(distribution,'lognormal')==1
    std_norm=sqrt(log(1+(std_RF/mean_RF)^2));
    mean_norm=log(mean_RF)-0.5*std_norm^2;
    var=std_norm^2;
end

N1=k1*2^mm;             % Number of cells in random field in x direction
N2=k2*2^mm;             % Number of cells in random field in y direction
NN=N1*N2;               % Number of Cells of random field
RF=zeros(N2,N1,runs);   % Allocate matrix voor values RF

% Matices for boundaries (corners and sides)
mc=[1 4 2 5; 2 5 3 6;4 7 5 8; 5 8 6 9];
ms=[4 2 1 1; 5 3 2 2; 6 5 4 3; 7 6 5 4; 8 8 7 5; 9 9 8 6];

%% Determine initial values
dx=Dx/k1;                    % Cell size stage 0
dy=Dy/k2;

% Determine covariances
ii=0;
kk=k1*k2;
Q=zeros(kk,kk);
if strcmp(Int,'Exact')==1
    cov=zeros(1,kk);
    for i=1:k2
        n=i-1;
        for j=1:k1
            m=j-1;
            ii=ii+1;
            if n==0 && m==0
                cov(ii)=var*varfn2D(dx,dy,L_cx,L_cy,c1,CF);
            else
                var1=(m-1)^2*(n-1)^2*varfn2D((m-1)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                var2=-2*(m-1)^2*(n)^2*varfn2D((m-1)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                var3=(m-1)^2*(n+1)^2*varfn2D((m-1)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                var4=-2*(m)^2*(n-1)^2*varfn2D((m)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                var5=4*(m)^2*(n)^2*varfn2D((m)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                var6=-2*(m)^2*(n+1)^2*varfn2D((m)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                var7=(m+1)^2*(n-1)^2*varfn2D((m+1)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                var8=-2*(m+1)^2*(n)^2*varfn2D((m+1)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                var9=(m+1)^2*(n+1)^2*varfn2D((m+1)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                if n==0 && m==1
                    cov(ii)=var/4*(var4+var6+var7+var9);
                elseif n==0 && m>=2
                    cov(ii)=var/4*(var1+var3+var4+var6+var7+var9);
                elseif n==1 && m==0
                    cov(ii)=var/4*(var2+var3+var8+var9);
                elseif n==1 && m==1
```

```
                    cov(ii)=var/4*(var5+var6+var8+var9);
                elseif n==1 && m>=2
                    cov(ii)=var/4*(var2+var3+var5+var6+var8+var9);
                elseif n>=2 && m==0
                    cov(ii)=var/4*(var1+var2+var3+var7+var8+var9);
                elseif n>=2 && m==1
                    cov(ii)=var/4*(var4+var5+var6+var7+var8+var9);
                else
                    cov(ii)=var/4*(var1+var2+var3+var4+var5+var6+var7+var8+var9);
                end
            end
        end
    end
    Q(1,:)=cov;
elseif strcmp(Int,'Gauss')==1
    for j=1:k2
        lagy=j-1;
        for i=1:k1
            ii=ii+1;
            lagx=i-1;
            Q(1,ii)=varfn2D_Gauss(CF,distribution,dx,dy,lagx,lagy,L_cx,L_cy,c1,var);
        end
    end
else
    Disp('Int must be equal to Exact or Gauss')
end

% Assemble correlation matrix stage 0
for j=2:kk
    mxj=mod(j-1,k1);
    myj=floor((j-1)/k1);
    for i=2:j
        mxi=mod(i-1,k1);
        myi=floor((i-1)/k1);
        m=1+(abs(mxj-mxi))+k1*(abs(myj-myi));
        Q(i,j)=Q(1,m);
    end
end
Q0=chol2(Q);

% Determine correlation matrix stage 1
if k1<=2 || k2<=2
    cov=zeros(nbh^2,1);
    Cor_M=zeros(nbh^2,nbh^2);
    ii=0;
    if strcmp(Int,'Exact')==1
        for i=1:3
            n=i-1;
            for j=1:3
                m=j-1;
                ii=ii+1;
                if n==0 && m==0
                    cov(ii)=var*varfn2D(dx,dy,L_cx,L_cy,c1,CF);
                else
                    var1=(m-1)^2*(n-1)^2*varfn2D((m-1)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                    var2=-2*(m-1)^2*(n)^2*varfn2D((m-1)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                    var3=(m-1)^2*(n+1)^2*varfn2D((m-1)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                    var4=-2*(m)^2*(n-1)^2*varfn2D((m)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                    var5=4*(m)^2*(n)^2*varfn2D((m)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                    var6=-2*(m)^2*(n+1)^2*varfn2D((m)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                    var7=(m+1)^2*(n-1)^2*varfn2D((m+1)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                    var8=-2*(m+1)^2*(n)^2*varfn2D((m+1)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                    var9=(m+1)^2*(n+1)^2*varfn2D((m+1)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                    if n==0 && m==1
                        cov(ii)=var/4*(var4+var6+var7+var9);
                    elseif n==0 && m==2
                        cov(ii)=var/4*(var1+var3+var4+var6+var7+var9);
                    elseif n==1 && m==0
                        cov(ii)=var/4*(var2+var3+var8+var9);
                    elseif n==1 && m==1
                        cov(ii)=var/4*(var5+var6+var8+var9);
                    elseif n==1 && m==2
                        cov(ii)=var/4*(var2+var3+var5+var6+var8+var9);
                    elseif n==2 && m==0
                        cov(ii)=var/4*(var1+var2+var3+var7+var8+var9);
                    elseif n==2 && m==1
                        cov(ii)=var/4*(var4+var5+var6+var7+var8+var9);
                    else
                        cov(ii)=var/4*(var1+var2+var3+var4+var5+var6+var7+var8+var9);
                    end
                end
            end
        end
    elseif strcmp(Int,'Gauss')==1
        for i=1:3
            lagy=i-1;
            for j=1:3
                ii=ii+1;
                lagx=j-1;
                cov(ii)=varfn2D_Gauss(CF,distribution,dx,dy,lagx,lagy,L_cx,L_cy,c1,var);
            end
        end
    else
        Disp('Int must be equal to Exact or Gauss')
    end
    kx=[0 1 2 0 1 2 0 1 2];
    ky=[0 0 0 1 1 1 2 2 2];
    for l=1:nbh^2
        for k=1:l
            m=1+abs(kx(l)-kx(k))+3*abs(ky(l)-ky(k));
            Cor_M(k,l)=cov(m);
            Cor_M(l,k)=cov(m);
        end
    end
else
    Cor_M=zeros(9,9);
    kx=[0 1 2 0 1 2 0 1 2];
    ky=[0 0 0 1 1 1 2 2 2];
    for j=1:9
        for i=1:j;
            m=1+abs(kx(j)-kx(i))+k1*abs(ky(j)-ky(i));
            Cor_M(i,j)=Q(1,m);
        end
    end
    Cor_M=Cor_M+Cor_M'-diag(diag(Cor_M));
end

%% Determine a_l and c coefficients
 Ai=zeros(9,3,mm);      % Allocate matrix for interior values for a_l coefficients
 Ci=zeros(6,mm);        % Allocate matrix for interior values for c coefficients
 Ac=zeros(4,3,4,mm);    % Allocate matrix for corner values for a_l coefficients
 Cc=zeros(6,4,mm);      % Allocate matrix for corner values for c coefficients
 As=zeros(6,3,4,mm);    % Allocate matrix for side values for a_l coefficients
 Cs=zeros(6,4,mm);      % Allocate matrix for side values for c coefficients
```

```
% Loop for every stage
for div=1:mm
    dx=dx/2;
    dy=dy/2;
    cov=zeros(1,16);
    c_cov=zeros(9,3);
    if strcmp(Int,'Exact')==1
        ii=0;
        for i=1:4
            n=i-1;
            for j=1:4
                m=j-1;
                ii=ii+1;
                if n==0 && m==0
                    cov(ii)=var*varfn2D(dx,dy,L_cx,L_cy,c1,CF);
                else
                    var1=(m-1)^2*(n-1)^2*varfn2D((m-1)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                    var2=-2*(m-1)^2*(n)^2*varfn2D((m-1)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                    var3=(m-1)^2*(n+1)^2*varfn2D((m-1)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                    var4=-2*(m)^2*(n-1)^2*varfn2D((m)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                    var5=4*(m)^2*(n)^2*varfn2D((m)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                    var6=-2*(m)^2*(n+1)^2*varfn2D((m)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                    var7=(m+1)^2*(n-1)^2*varfn2D((m+1)*dx,(n-1)*dy,L_cx,L_cy,c1,CF);
                    var8=-2*(m+1)^2*(n)^2*varfn2D((m+1)*dx,(n)*dy,L_cx,L_cy,c1,CF);
                    var9=(m+1)^2*(n+1)^2*varfn2D((m+1)*dx,(n+1)*dy,L_cx,L_cy,c1,CF);
                    if n==0 && m==1
                        cov(ii)=var/4*(var4+var6+var7+var9);
                    elseif n==0 && m>=2
                        cov(ii)=var/4*(var1+var3+var4+var6+var7+var9);
                    elseif n==1 && m==0
                        cov(ii)=var/4*(var2+var3+var8+var9);
                    elseif n==1 && m==1
                        cov(ii)=var/4*(var5+var6+var8+var9);
                    elseif n==1 && m>=2
                        cov(ii)=var/4*(var2+var3+var5+var6+var8+var9);
                    elseif n>=2 && m==0
                        cov(ii)=var/4*(var1+var2+var3+var7+var8+var9);
                    elseif n>=2 && m==1
                        cov(ii)=var/4*(var4+var5+var6+var7+var8+var9);
                    else
                        cov(ii)=var/4*(var1+var2+var3+var4+var5+var6+var7+var8+var9);
                    end
                end
            end
        end

    elseif strcmp(Int,'Gauss')==1
        ii=0;
        for i=1:4
            lagy=i-1;
            for j=1:4
                lagx=j-1;
                ii=ii+1;
                cov(ii)=varfn2D_Gauss(CF,distribution,dx,dy,lagx,lagy,L_cx,L_cy,c1,var);
            end
        end
    else
        Disp('Int must be equal to Exact or Gauss')
    end

    B=[cov(1) cov(2) cov(5); cov(2) cov(1) cov(6); cov(5) cov(6) cov(1)];

    c_cov(1,1)=0.25*(cov(11)+cov(12)+cov(15)+cov(16));
    c_cov(2,1)=0.25*(cov(9)+cov(10)+cov(13)+cov(14));
    c_cov(3,1)=0.25*(cov(10)+cov(11)+cov(14)+cov(15));
    c_cov(4,1)=0.25*(cov(3)+cov(4)+cov(7)+cov(8));
    c_cov(5,1)=0.25*(cov(1)+cov(2)+cov(5)+cov(6));
    c_cov(6,1)=0.25*(cov(2)+cov(3)+cov(6)+cov(7));
    c_cov(7,1)=0.25*(cov(7)+cov(8)+cov(11)+cov(12));
    c_cov(8,1)=0.25*(cov(5)+cov(6)+cov(9)+cov(10));
    c_cov(9,1)=0.25*(cov(6)+cov(7)+cov(10)+cov(11));

    c_cov(1,2)=c_cov(7,1);
    c_cov(2,2)=c_cov(8,1);
    c_cov(3,2)=c_cov(9,1);
    c_cov(4,2)=c_cov(4,1);
    c_cov(5,2)=c_cov(5,1);
    c_cov(6,2)=c_cov(6,1);
    c_cov(7,2)=c_cov(1,1);
    c_cov(8,2)=c_cov(2,1);
    c_cov(9,2)=c_cov(3,1);

    c_cov(1,3)=c_cov(3,1);
    c_cov(2,3)=c_cov(2,1);
    c_cov(3,3)=c_cov(1,1);
    c_cov(4,3)=c_cov(6,1);
    c_cov(5,3)=c_cov(5,1);
    c_cov(6,3)=c_cov(4,1);
    c_cov(7,3)=c_cov(9,1);
    c_cov(8,3)=c_cov(8,1);
    c_cov(9,3)=c_cov(7,1);

    % Determine interior parameters
    Ai(:,:,div)=Cor_M\c_cov;
    RR=B-c_cov'*Ai(:,:,div);
    BBchol=chol2(RR);
    ii=0;
    for j=1:3
        for i=1:j
            ii=ii+1;
            Ci(ii,div)=BBchol(i,j);
        end
    end

    % Determine corner parameters
    RC=zeros(4);
    DA=zeros(4,1);
    BB=zeros(3,3);
    for j=1:4
        for i=1:j
            RC(i,j)=Cor_M(mc(i,1),mc(j,1));
        end
    end
    RC=RC+RC'-diag(diag(RC));
    for nc=1:4
        for j=1:3
            for i=1:4
                DA(i)=c_cov(mc(i,nc),j);
            end
            DA=RC\DA;
            for i=1:4
                Ac(i,j,nc,div)=DA(i);
            end
            for i=1:j
                BB(i,j)=B(i,j)-c_cov(mc(1,nc),i)*DA(1)- c_cov(mc(2,nc),i)...
                    *DA(2)-c_cov(mc(3,nc),i)*DA(3)-c_cov(mc(4,nc),i)*DA(4);
            end
        end
        BBchol=chol2(BB);
        ii=0;
        for j=1:3
            for i=1:j
                ii=ii+1;
                Cc(ii,nc,div)=BBchol(i,j);
            end
        end
    end

    % Determine side parameters
    BB=zeros(3,3);

    for ns=1:4
        RS=zeros(6,6);
        for j=1:6
            for i=1:j
                RS(i,j)=Cor_M(ms(i,ns),ms(j,ns));
            end
        end
        RS=RS+RS'-diag(diag(RS));
        for j=1:3
            for i=1:6
                DA(i)=c_cov(ms(i,ns),j);
            end
            DA=RS\DA;
            for i=1:6
            As(i,j,ns,div)=DA(i);
            end
            for i=1:j
                 BB(i,j)=B(i,j)-c_cov(ms(1,ns),i)*DA(1)-c_cov(ms(2,ns),i) ...
                     *DA(2)-c_cov(ms(3,ns),i)*DA(3)-c_cov(ms(4,ns),i) ...
                     *DA(4)-c_cov(ms(5,ns),i)*DA(5)-c_cov(ms(6,ns),i)*DA(6);
            end
        end
        BBchol=chol2(BB);
        ii=0;
        for j=1:3
            for i=1:j
                ii=ii+1;
                Cs(ii,ns,div)=BBchol(i,j);
            end
        end
    end

    % Determine Correlation matrix for next stage
    kx=[0 1 2 0 1 2 0 1 2];
    ky=[0 0 0 1 1 1 2 2 2];
    Cor_M=zeros(9,9);
    for l=1:nbh^2
        for k=1:l
            m=1+abs(kx(l)-kx(k))+4*abs(ky(l)-ky(k));
            Cor_M(k,l)=cov(m);
            Cor_M(l,k)=cov(m);
        end
    end
end
```

```
it=cputime-t;

%% generate field
for run=1:runs
    %Generate values stage 0
    RV0=randn(kk,1);
    z(1:kk)=Q0'*RV0;

    for i=1:mm
        z_p=z;

        % Corners
        % Old positions corners
        cor12=k1*2^(i-1)*(k2*2^(i-1)-1);
        cor11=cor12-1;
        cor14=k1*2^(i-1)*(k2*2^(i-1));
        cor13=cor14-1;

        cor22=k1*2^(i-1);
        cor21=cor22-1;
        cor24=2*k1*2^(i-1);
        cor23=cor24-1;

        cor31=k1*2^(i-1)*(k2*2^(i-1)-2)+1;
        cor32=cor31+1;
        cor33=k1*2^(i-1)*(k2*2^(i-1)-1)+1;
        cor34=cor33+1;

        cor41=1;
        cor42=2;
        cor43=k1*2^(i-1)+1;
        cor44=cor43+1;

        % New positions corners
        v11=k1*2^i*(k2*2^i);
        v12=k1*2^i*(k2*2^i-1);
        v13=v11-1;
        v14=v12-1;


        v21=2*k1*2^i;
        v22=k1*2^i;
        v23=v21-1;
        v24=v22-1;

        v33=k1*2^i*(k2*2^i-1)+1;
        v34=k1*2^i*(k2*2^i-2)+1;
        v31=v33+1;
        v32=v34+1;


        v43=k1*2^i+1;
        v44=1;
        v41=v43+1;
        v42=v44+1;

        RV=randn(3,4);

        %Corner 1
        z(v11)=Cc(1,1,i)*RV(1,1)+z_p([cor11 cor12 cor13 cor14])*Ac(:,1,1,i);
        z(v12)=Cc(2:3,1,i)'*RV(1:2,1)+z_p([cor11 cor12 cor13 cor14])*Ac(:,2,1,i);
        z(v13)=Cc(4:6,1,i)'*RV(1:3,1)+z_p([cor11 cor12 cor13 cor14])*Ac(:,3,1,i);
        z(v14)=4*z_p(cor14)-z(v13)-z(v12)-z(v11);

        %Corner 2
        z(v21)=Cc(1,2,i)*RV(1,2)+z_p([cor21 cor22 cor23 cor24])*Ac(:,1,2,i);
        z(v22)=Cc(2:3,2,i)'*RV(1:2,2)++z_p([cor21 cor22 cor23 cor24])*Ac(:,2,2,i);
        z(v23)=Cc(4:6,2,i)'*RV(1:3,2)++z_p([cor21 cor22 cor23 cor24])*Ac(:,3,2,i);
        z(v24)=4*z_p(cor22)-z(v23)-z(v22)-z(v21);

        %Corner 3
        z(v31)=Cc(1,3,i)*RV(1,3)+z_p([cor31 cor32 cor33 cor34])*Ac(:,1,3,i);
        z(v32)=Cc(2:3,3,i)'*RV(1:2,3)+z_p([cor31 cor32 cor33 cor34])*Ac(:,2,3,i);
        z(v33)=Cc(4:6,3,i)'*RV(1:3,3)+z_p([cor31 cor32 cor33 cor34])*Ac(:,3,3,i);
        z(v34)=4*z_p(cor33)-z(v33)-z(v32)-z(v31);

        %Corner 4
        z(v41)=Cc(1,4,i)*RV(1,4)+z_p([cor41 cor42 cor43 cor44])*Ac(:,1,4,i);
        z(v42)=Cc(2:3,4,i)'*RV(1:2,4)+z_p([cor41 cor42 cor43 cor44])*Ac(:,2,4,i);
        z(v43)=Cc(4:6,4,i)'*RV(1:3,4)+z_p([cor41 cor42 cor43 cor44])*Ac(:,3,4,i);
        z(v44)=4*z_p(cor41)-z(v43)-z(v42)-z(v41);

        % sides
        if k1<=2 && i==1
        else
            % old starting positions sides
            sid11=1-1;
            sid12=2-1;
            sid13=3-1;
            sid14=k1*2^(i-1)+1-1;
            sid15=sid14+1;
            sid16=sid15+1;

            sid41=k1*2^(i-1)*(k2*2^(i-1)-2)+1-1;
            sid42=sid41+1;
            sid43=sid42+1;
            sid44=k1*2^(i-1)*(k2*2^(i-1)-1)+1-1;
            sid45=sid44+1;
            sid46=sid45+1;

            % new positions sides
            w11=k1*2^i+4-2;
            w12=4-2;
            w13=w11-1;
            w14=3-2;

            w41=k1*2^i*(k2*2^i-1)+4-2;
            w42=k1*2^i*(k2*2^i-2)+4-2;
            w43=w41-1;
            w44=w42-1;

            % Side 1
            for j=1:k1*2^(i-1)-2
                RV=randn(3,1);
                sid11=sid11+1;
                sid12=sid12+1;
                sid13=sid13+1;
                sid14=sid14+1;
                sid15=sid15+1;
                sid16=sid16+1;

                w11=w11+2;
                w12=w12+2;
                w13=w13+2;
                w14=w14+2;

                z(w11)=Cs(1,1,i)*RV(1,1)+z_p([sid11 sid12 sid13 sid14 sid15 sid16]) ...
                       *As(:,1,1,i);
                z(w12)=Cs(2:3,1,i)'*RV(1:2,1)+z_p([sid11 sid12 sid13 sid14 sid15 sid16]) ...
                       *As(:,2,1,i);
                z(w13)=Cs(4:6,1,i)'*RV(1:3,1)+z_p([sid11 sid12 sid13 sid14 sid15 sid16]) ...
                       *As(:,3,1,i);
                z(w14)=4*z_p(sid12)-z(w13)-z(w12)-z(w11);
            end
            % Side 4
             for j=1:k1*2^(i-1)-2
                RV=randn(3,1);
                sid41=sid41+1;
                sid42=sid42+1;
                sid43=sid43+1;
                sid44=sid44+1;
                sid45=sid45+1;
                sid46=sid46+1;

                w41=w41+2;
                w42=w42+2;
                w43=w43+2;
                w44=w44+2;

                z(w41)=Cs(1,4,i)*RV(1,1)+z_p([sid41 sid42 sid43 sid44 sid45 sid46]) ...
                       *As(:,1,4,i);
                z(w42)=Cs(2:3,4,i)'*RV(1:2,1)+z_p([sid41 sid42 sid43 sid44 sid45 sid46])...
                       *As(:,2,4,i);
                z(w43)=Cs(4:6,4,i)'*RV(1:3,1)+z_p([sid41 sid42 sid43 sid44 sid45 sid46])...
                       *As(:,3,4,i);
                z(w44)=4*z_p(sid45)-z(w43)-z(w42)-z(w41);
            end
        end

        if k2<=2 && i==1
            else
            % old starting positions sides
            sid21=1-k1*2^(i-1);
            sid22=2-k1*2^(i-1);
            sid23=k1*2^(i-1)+1-k1*2^(i-1);
            sid24=sid23+1;
            sid25=2*k1*2^(i-1)+1-k1*2^(i-1);
            sid26=sid25+1;

            sid31=k1*2^(i-1)-1-k1*2^(i-1);
            sid32=sid31+1;
            sid33=2*k1*2^(i-1)-1-k1*2^(i-1);
            sid34=sid33+1;
            sid35=3*k1*2^(i-1)-1-k1*2^(i-1);
            sid36=sid35+1;

            % new positions sides
            w21=3*k1*2^i+2-2*k1*2^i;
            w22=2*k1*2^i+2-2*k1*2^i;
            w23=w21-1;
            w24=w22-1;

            w31=4*k1*2^i-2*k1*2^i;
            w32=3*k1*2^i-2*k1*2^i;
            w33=w31-1;
            w34=w32-1;

            % Side 2
            for j=1:k2*2^(i-1)-2
                RV=randn(3,1);
                sid21=sid21+k1*2^(i-1);
                sid22=sid22+k1*2^(i-1);
                sid23=sid23+k1*2^(i-1);
                sid24=sid24+k1*2^(i-1);
                sid25=sid25+k1*2^(i-1);
                sid26=sid26+k1*2^(i-1);

                w21=w21+2*k1*2^i;
                w22=w22+2*k1*2^i;
                w23=w23+2*k1*2^i;
                w24=w24+2*k1*2^i;

                z(w21)=Cs(1,2,i)*RV(1,1)+z_p([sid21 sid22 sid23 sid24 sid25 sid26])...
                       *As(:,1,2,i);
                z(w22)=Cs(2:3,2,i)'*RV(1:2,1)+z_p([sid21 sid22 sid23 sid24 sid25 sid26])...
                       *As(:,2,2,i);
                z(w23)=Cs(4:6,2,i)'*RV(1:3,1)+z_p([sid21 sid22 sid23 sid24 sid25 sid26])...
                       *As(:,3,2,i);
                z(w24)=4*z_p(sid23)-z(w23)-z(w22)-z(w21);
            end
            % Side 3
            for j=1:k2*2^(i-1)-2
                RV=randn(3,1);
                k=j-1;
                sid31=sid31+k1*2^(i-1);
                sid32=sid32+k1*2^(i-1);
                sid33=sid33+k1*2^(i-1);
                sid34=sid34+k1*2^(i-1);
                sid35=sid35+k1*2^(i-1);
                sid36=sid36+k1*2^(i-1);

                w31=w31+2*k1*2^i;
                w32=w32+2*k1*2^i;
                w33=w33+2*k1*2^i;
                w34=w34+2*k1*2^i;

                z(w31)=Cs(1,3,i)*RV(1,1)+z_p([sid31 sid32 sid33 sid34 sid35 sid36])...
                       *As(:,1,3,i);
                z(w32)=Cs(2:3,3,i)'*RV(1:2,1)+z_p([sid31 sid32 sid33 sid34 sid35 sid36])...
                       *As(:,2,3,i);
                z(w33)=Cs(4:6,3,i)'*RV(1:3,1)+z_p([sid31 sid32 sid33 sid34 sid35 sid36])...
                       *As(:,3,3,i);
                z(w34)=4*z_p(sid34)-z(w33)-z(w32)-z(w31);
            end
        end

        % interior values
        % old starting positions
        int1=1-3;
        int2=2-3;
        int3=3-3;
        int4=k1*2^(i-1)+1-3;
        int5=int4+1;
        int6=int5+1;
        int7=2*k1*2^(i-1)+1-3;
        int8=int7+1;
        int9=int8+1;

        % new positions
        u1=3*k1*2^i+4-6-k1*2^i;
        u2=2*k1*2^i+4-6-k1*2^i;
        u3=u1-1;
        u4=u2-1;

        for j=1:k2*2^(i-1)-2
            int1=int1+2;
            int2=int2+2;
            int3=int3+2;
            int4=int4+2;
            int5=int5+2;
            int6=int6+2;
            int7=int7+2;
            int8=int8+2;
            int9=int9+2;

            u1=u1+4+k1*2^i;
            u2=u2+4+k1*2^i;
            u3=u3+4+k1*2^i;
            u4=u4+4+k1*2^i;
            for k=1:k1*2^(i-1)-2
                RV=randn(3,1);
                int1=int1+1;
                int2=int2+1;
                int3=int3+1;
                int4=int4+1;
                int5=int5+1;
                int6=int6+1;
                int7=int7+1;
                int8=int8+1;
                int9=int9+1;

                u1=u1+2;
                u2=u2+2;
                u3=u3+2;
                u4=u4+2;

                z(u1)=Ci(1,i)*RV(1,1)+z_p([int1 int2 int3 int4 int5 int6 int7 int8 int9])...
                      *Ai(:,1,i);
                z(u2)=Ci(2:3,i)'*RV(1:2,1)+z_p([int1 int2 int3 int4 int5 int6 int7 int8 int9])...
                      *Ai(:,2,i);
                z(u3)=Ci(4:6,i)'*RV(1:3,1)+z_p([int1 int2 int3 int4 int5 int6 int7 int8 int9])...
                      *Ai(:,3,i);
                z(u4)=4*z_p(int5)-z(u3)-z(u2)-z(u1);
            end
        end

    end
    for j=1:N2
        RF(1:N1,j,run)=z((j-1)*N1+1:j*N1);
    end
end

% tranformation of standard normal gaussian field
if strcmp(distribution,'normal')==1
    RF=RF*std_RF+mean_RF;
elseif strcmp(distribution,'lognormal')==1
    RF=exp(std_norm*RF+mean_norm);
end

rt=cputime-t;
```

```

% plot of 1st random field
figure()
Meshx=linspace(1/2*dx,Dx-1/2*dx,N1);
Meshy=linspace(1/2*dy,Dy-1/2*dy,N2);
surf(Meshx,Meshy,RF(:,:,1)')

% Determin statistical properties random field

Mesh=linspace(0,Dx-dx,N1);
L_c=[L_cx; L_cy];
[mean_m,mean_s,std_m,std_s,c_err_m,c_err_s]=stat_prop(RF,runs,dim,Mesh,CF,c1,L_c);
st=cputime-t;
table=[it rt st mean_m mean_s std_m std_s c_err_m c_err_s];
```

I.7 Matlab code: Expansion Optimal Linear Estimation in 1D

```matlab
% EOLE_1D  returns correlated random variables for a homogeneous gaussain field
% in 1D applying the Expension Optimal Linear Estimation method
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:         Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_c:        Correlation length/scale of fluctuation
%   D:          Size random field
%   N:          Number of nodes in random field

clc, close all, clear all
t=cputime;

% Input values:
runs=2000;
dim=1;
mean_RF=0;
std_RF=1;
CF='SExp';
c1=0;
L_c=5;
D=17.5;
N=8;

if strcmp(CF,'Exp')==1
    M=round(D/(L_c/5));         % Mesh RF for generation
elseif strcmp(CF,'SExp')==1
```

```
    M=round(D/(L_c/2));         % Mesh RF for generation
else
    disp('CF must be equal to Exp or SExp')
end

K=round((3/4)*M);               % Truncation order for EOLE expansion

RF_Mesh=linspace(0,D,M);        % Random field Mesh
RF=zeros(N,runs);               % Allocate matrix voor values RF

%Determine Correlation matrix
Cor_M=zeros(length(RF_Mesh),length(RF_Mesh));
if strcmp(CF,'Exp')==1
    for i=1:length(RF_Mesh)
        for j=1:length(RF_Mesh)
            Cor_M(i,j)=std_RF^2*(c1+(1-c1)*exp(-abs((RF_Mesh(i)-RF_Mesh(j)))/L_c));
        end
    end
elseif strcmp(CF,'SExp')==1
    for i=1:length(RF_Mesh)
        for j=1:length(RF_Mesh)
            Cor_M(i,j)=std_RF^2*(c1+(1-c1)*exp(-((abs(RF_Mesh(i)-RF_Mesh(j)))/L_c)^2));
        end
    end
else
    disp('CF must be equal to Exp or SExp')
end

x=sym('x');
[vec,lambda]=eig(Cor_M);
lambda=diag(lambda);
lambda=flipud(lambda);
vec=fliplr(vec);
RF_Mesh=fliplr(RF_Mesh);

Cor=zeros(1,M);
Cor=sym(Cor);
for i=1:M
    if strcmp(CF,'Exp')==1
        Cor(i)=symfun(c1+(1-c1)*exp(-abs((x-RF_Mesh(i))/L_c)),x);
    elseif strcmp(CF,'SExp')==1
        Cor(i)=symfun(c1+(1-c1)*exp(-((x-RF_Mesh(i))/L_c)^2),x);
    else
        disp('CF must be equal to Exp or SExp')
    end
end
it=cputime-t;

Mesh=linspace(0,D,N);
for run=1:runs
    CRV=mean_RF;
    RV=randn(K,1);
    for i=1:K
        CRV=CRV+std_RF*RV(i)/sqrt(lambda(i))*vec(:,i)'*Cor';
    end

    j=1;
    for i=1:N
        RF(i,run)=subs(CRV,x,Mesh(i));
    end
end
rt=cputime-t;
```

```

% plot of 1st random field
figure()
plot(Mesh,RF(:,1))

% Determin statistical properties random field
[mean_m,mean_s,std_m,std__s,c_err_m,c_err_s]=stat_prop(RF,runs,dim,Mesh,CF,c1,L_c);
st=cputime-t;
table=[it rt st mean_m mean_s std_m std__s c_err_m c_err_s];
```

I.8 Matlab code: Expansion Optimal Linear Estimation in 2D

```matlab
% EOLE_2D  returns correlated random variables for a homogeneous gaussain field
% in 2D applying the Expension Optimal Linear Estimation method
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Output:
%   RF:         Matrix containing the coordinates and correlated random variables
%               belonging to that coordinate
%   it:         Intialization time random fields
%   rt:         Realization time random fields
%   st:         Time to determine the statistical properties of random
%               fields
%   mean_m:     Mean of the mean values of the random fields
%   mean_s:     Standard deviation of the mean values of the random fields
%   std_m:      Mean of the standard deviations of the random fields
%   std_s:      Standard deviation of the standard deviations of the random
%               fields
%   c_err_m:    Mean absolute error in correlation in all the random fields
%   c_err_s:    Standard deviations in the absolute errors of the the
%               correlation in all the random fields
%
% Input:
%   runs:       Number of random fields created
%   dim:        dimension random field
%   Mean_RF:    Mean value random field
%   std_RF:     Standard deviation random field
%   CF:         Type of correlation function ('Exp', 'SExp')
%                   Exp:    Exponential correlation function
%                           Exp = c1+(1-c1)*exp(-delta_x/lc)
%                   SExp:   Squared exponential correlation function
%                           SExp = c1+(1-c1)*exp(-(delta_x/lc)^2)
%               Where delta_x is the lag distance which can be detemined with
%               the coordinates of the Mesh
%   c1:         threshold value for correlation function
%   L_cx:       Correlation length/scale of fluctuation in x-direcion
%   L_cy:       Correlation length/scale of fluctuation in y-direcion
%   Dx:         Size random field in x direction
%   Dy:         Size random field in y direction
%   Nx:         Number of nodes in random field in y direction
%   Ny:         Number of nodes in random field in y direction

clc, close all, clear all
t=cputime;

% Input values:
runs=1;
dim=2;
mean_RF=0;
std_RF=1;
CF='Exp';
c1=0.5;
L_cx=5;
L_cy=5;
Dx=17.5;
Dy=17.5;
```

```
Nx=6;
Ny=8;

if strcmp(CF,'Exp')==1
    Mx=round(Dx/(L_cx/5));      % Mesh RF for generation in x-direction
    My=round(Dy/(L_cy/5));      % Mesh RF for generation in y-direction
elseif strcmp(CF,'SExp')==1
    Mx=round(Dx/(L_cx/2));      % Mesh RF for generation in x-direction
    My=round(Dy/(L_cy/2));      % Mesh RF for generation in y-direction
else
    disp('CF must be equal to Exp or SExp')
end

Kx=round((3/4)*Mx);               % Truncation order for EOLE expansion
Ky=round((3/4)*My);
K=Kx*Ky;
RF=zeros(Nx,Ny,runs);

RF_Mesh=zeros(Mx*My,2);
for i=1:Mx
    for j=1:My
        RF_Mesh((j-1)*Mx+i,1:2)=[(i-1)*Dx/(Mx-1)  (j-1)*Dy/(My-1)];
    end
end

%Determine Correlation matrix
Cor_M=zeros(length(RF_Mesh),length(RF_Mesh));
if strcmp(CF,'Exp')==1
    for i=1:length(RF_Mesh)
        for j=1:length(RF_Mesh)
            Cor_M(i,j)=std_RF^2*(c1+(1-c1)*exp(-sqrt(((RF_Mesh(i,1)-RF_Mesh(j,1))/L_cx)^2 ...
            +((RF_Mesh(i,2)-RF_Mesh(j,2))/L_cy)^2)));
        end
    end
elseif strcmp(CF,'SExp')==1
    for i=1:length(RF_Mesh)
        for j=1:length(RF_Mesh)
            Cor_M(i,j)=std_RF^2*(c1+(1-c1)*exp(-(((RF_Mesh(i,1)-RF_Mesh(j,1))/L_cx)^2 ...
                +((RF_Mesh(i,2)-RF_Mesh(j,2))/L_cy)^2)));
        end
    end
else
    disp('CF must be equal to Exp or SExp')
end

x=sym('x');
y=sym('y');
[vec,lambda]=eig(Cor_M);
lambda=diag(lambda);
lambda=flipud(lambda);
vec=fliplr(vec);
RF_Mesh=flipud(RF_Mesh);

Cor=zeros(1,length(RF_Mesh));
Cor=sym(Cor);
for i=1:length(RF_Mesh)
    if strcmp(CF,'Exp')==1
        Cor(i)=symfun(c1+(1-c1)*exp(-(sqrt((x-RF_Mesh(i,1))^2 ...
            +(y-RF_Mesh(i,2))^2))/L_cx),[x y]);
    elseif strcmp(CF,'SExp')==1
        Cor(i)=symfun(c1+(1-c1)*exp(-((sqrt((x-RF_Mesh(i,1))^2 ...
        +(y-RF_Mesh(i,2))^2))/L_cx)^2),[x y]);
    else
        disp('CorF must be equal to Exp, SExp or CSin')
    end
end
it=cputime-t;

for run=1:runs
    CRV=mean_RF;
    RV=randn(K,1);
    for i=1:K
        CRV=CRV+std_RF*RV(i)/sqrt(lambda(i))*vec(:,i)'*Cor';
    end
    Meshx=linspace(0,Dx,Nx);
    Meshy=linspace(0,Dy,Ny);

    for j=1:length(Meshx)
        for i=1:length(Meshy)
        RF(i,j,run)=subs(CRV,[x y],[Meshx(j) Meshy(i)]);
        end
    end
end

rt=cputime-t;

% plot of 1st random field
figure()
surf(Meshx,Meshy,RF(:,:,1)')

% Determin statistical properties random field
Mesh=Meshx;
L_c=[L_cx; L_cy];
[mean_m,mean_s,std_m,std_s,c_err_m,c_err_s]=stat_prop(RF,runs,dim,Mesh,CF,c1,L_c);
st=cputime-t;

table=[std_RF it rt st mean_m mean_s std_m std_s c_err_m c_err_s];
```

J Matlab code: Modified Cholesky decomposition

```matlab
function [A,error] = chol2(Q)
% Modified Cholesky decomposition where a tollerance value is set to ommit
% negative values in the square root of the diagonal terms.
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Based on subroutine chol2.f of RFEM software which is written by
% Gordon A. Fenton. Free available on http://courses.engmath.dal.ca/rfem/
%
% Output:
%   A:      Matrix containing the values of the approximated cholesky
%           decomposition.
%   error:  Estimated error in the decomposition.
%
% Input:
%   Q:      Symmetric matrix which have to be decomposed.

%% Decomposition matrix Q
n=size(Q,1);
error=0;
Q1=Q(n,n);
if n>=3
    Q2=Q(n-1,n-1);
    Q3=Q(n-2,n-2);
end
if n>=128
    tol=10^-4;
else
    tol=10^-8;
end

A=zeros(n,n);
for i=1:n
    s=0;
        for j=1:i-1
            s=s+Q(j,i)*Q(j,i);
        end
        t=Q(i,i)-s;
    if t<=tol
        Q(i,i)=0;
        for j=i+1:n
            Q(i,j)=0;
        end
    else
        Q(i,i)=sqrt(t);
        for j=i+1:n
            s=0;
            for k=1:i-1
                s=s+Q(k,i)*Q(k,j);
            end
            Q(i,j)=(Q(i,j)/Q(i,i))-(s/Q(i,i));
        end
    end
end
for i=1:n
    for j=i:n
        A(i,j)=Q(i,j);
    end
end

%% estimation of the error
if n>=3
    t1=Q(1,n)*Q(1,n)+Q(n-1,n)*Q(n-1,n)+Q(n,n)*Q(n,n);
    t2=Q(1,n-1)*Q(1,n-1)+Q(n-1,n-1)*Q(n-1,n-1);
    t3=Q(1,n-2)*Q(1,n-2);
    for i=2:n-2
        t1=t1+Q(i,n)*Q(i,n);
        t2=t2+Q(i,n-1)*Q(i,n-1);
        t3=t3+Q(i,n-2)*Q(i,n-2);
    end
    r1=abs((Q1-t1)/Q1);
    r2=abs((Q2-t2)/Q2);
    if r1>=r2
        r2=r1;
    end
    r3=abs((Q3-t3)/Q3);
    if r2>=r3
        r3=r2;
    end
    error=r3;
else
    t=Q(1,n)*Q(1,n);
    for i=2:n
        t=t+Q(i,n)*Q(i,n);
    end
    error=abs((Q1-t)/Q1);
end
```

K

Matlab code: Inverse Fourier Transform algorithm

```matlab
function [Ak,Bk] = invFFT1D( Ak,Bk,m1)
% invFFT1D returns the inverse fourier transform applying a fast fourier
% algorithm
%
% Written by Robin van der Have, Delft, 25 September 2015
%
% Based on subroutine fft1d.f of RFEM software which is written by
% Gordon A. Fenton. It is originally written by J.W. Cooley et al.
% http://courses.engmath.dal.ca/rfem/
%
% Ouput:
%   Ak:     Real vector of length N=2^m1  containing the real part of the
%           random field
%   Bk:     Real vector of length N=2^m1  containing the imaginary part of
%           the random field
% Input:
%   Ak:     Real vector of length N=2^m1 containing the fourier
%           coefficients
%   Bk:     Real vector of length N=2^m1 containing the fourier
%           coefficients
%   m1:     The length of the fourier sequence considered is N=2^m1

j=1;
N=2^m1;
for l=1:(N-1)
    if l<j
        t=Ak(j);
        Ak(j)=Ak(l);
        Ak(l)=t;
        t=Bk(j);
        Bk(j)=Bk(l);
        Bk(l)=t;
    end
    k=N/2;
    while k<j
        j=j-k;
        k=k/2;
    end
    j=j+k;
end
me=1;
for mm=1:m1
    k=me;
    pibyk=pi/k;
    me=2*me;
    wr=cos(pibyk);
    wi=sin(pibyk);
    qr=1;
    qi=0;
    for j=1:k
```

```
        for l=j:me:N
            sr=Ak(l+k)*qr-Bk(l+k)*qi;
            si=Bk(l+k)*qr+Ak(l+k)*qi;
            Ak(l+k)=Ak(l)-sr;
            Bk(l+k)=Bk(l)-si;
            Ak(l)=Ak(l)+sr;
            Bk(l)=Bk(l)+si;
        end
        tr=qr*wr-qi*wi;
        ti=qr*wi+qi*wr;
        qr=tr;
        qi=ti;
    end
end

end
```

L Z-Test random number generators

To test the random number generators, used in this research, a Z-test is performed on a sample of $2 \cdot 10^5$ generated standard normal random numbers. In Matlab the *randn* function is used to generate the numbers. In DIANA a generator of uniformly distributed numbers is combined with the transformation method of Box and Muller. The generator of uniform distributed random number generators is called the *Mersenne Twister* and was developed by Takuji Nishimura and Makoto Matsumoto in 1997 [49]. This pseudorandom number generator has a period equal to $2^{19937} - 1$.

The mean value and standard deviation of both samples are given in table L.1. It can be seen that the values are very close to the desired values.

Tab. L.1.: Mean value and standard deviation of samples.

	Desired	Matlab	DIANA
Mean value	0	-0.00414808	0.00161677
Standard deviation	1	0.99944467	0.99978786

The graphs where the results of the Z-tests are plotted can be found in figure L.1 and L.2. The sample data are the generated numbers sorted from small to large. The expected values are the inverse values of a cummulative distribution with mean and standard deviation as specified in table L.1. The z-values are the inverse values of the standard cumulative distribution. If the sample data is close to the expected values the sample can be classified as normally distributed. For both Matlab and DIANA this is the case as can be seen in the graphs.

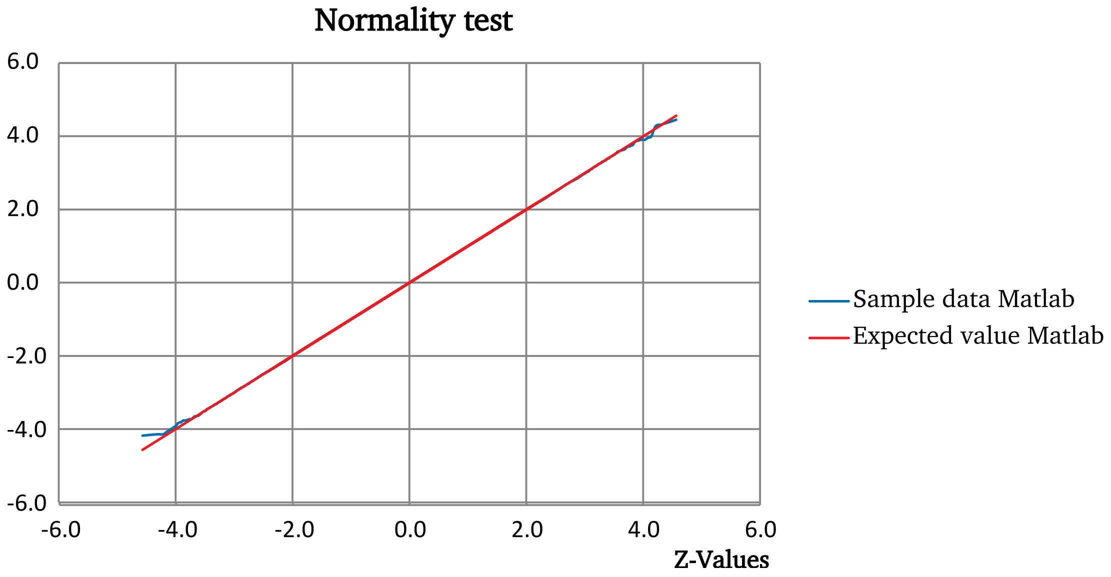


Fig. L.1.: Z-test for sample generated with Matlab.

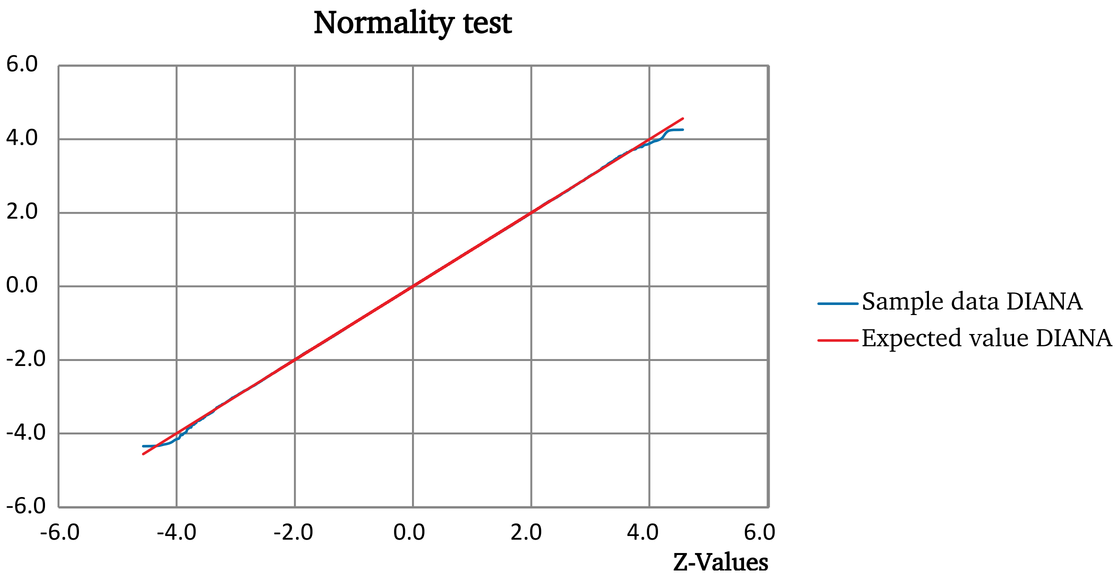


Fig. L.2.: Z-test for sample generated with DIANA.

M

Data comparison of random field generators

In this appendix the data of the assessment, carried out in chapter 3, can be found. First, the results for the 1D fields are given in table M.1 to M.36. Thereafter, the results for the 2D fields are given in table M.37 to M.72

Tab. M.1.: Initialization time of the different methods in 1D for normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0312	0.0156	0.0312	0.0000	0.0312	0.0000	0.0624	0.0468	0.0156	0.0156	-	-
5	16	0.0000	0.0000	0.0000	0.0000	0.0156	0.0000	0.0468	0.0468	0.0000	0.0156	0.0156	0.0156
5	32	0.0000	-	0.0624	0.0000	0.0000	0.0000	0.1092	0.0468	0.0000	0.0156	0.0000	0.0000
2.5	16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0468	0.0468	0.0000	0.0000	0.0000	0.0156
2.5	32	0.0000	0.0000	0.0000	0.0000	0.0000	0.0156	0.0468	0.0468	0.0000	0.0156	0.0156	0.0156
2.5	64	0.0000	-	0.0000	0.0000	0.0000	0.0000	0.0468	0.0624	0.0000	0.0156	0.0000	0.0156
0.5	64	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0468	0.0624	0.0000	0.0000	0.0000	0.0156
0.5	128	0.0468	0.0000	0.0156	0.0624	0.0156	0.0156	0.0624	0.0624	0.0156	0.0156	0.0156	0.0156
0.5	256	0.0000	-	0.0780	0.0156	0.0624	0.0468	0.0312	0.0624	0.0156	0.0156	0.0156	0.0156

Tab. M.2.: Initialization time of the different methods in 1D for normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0156	0.0000	0.0312	0.0156	0.0312	0.0000	0.0624	0.0468	0.0312	0.0156	-	-
5	16	0.0000	0.0000	0.0000	0.0000	0.0156	0.0000	0.0468	0.0624	0.0156	0.0156	0.0312	0.0156
5	32	0.0000	-	0.0000	0.0000	0.0000	0.0000	0.0468	0.0468	0.0000	0.0156	0.0000	0.0156
2.5	16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0624	0.0468	0.0000	0.0000	0.0156	0.0156
2.5	32	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0624	0.0468	0.0000	0.0000	0.0000	0.0000
2.5	64	0.0000	-	0.0000	0.0156	0.0000	0.0156	0.0624	0.0468	0.0000	0.0156	0.0156	0.0156
0.5	64	0.0000	0.0000	0.0624	0.0468	0.0000	0.0000	0.0468	0.0468	0.0000	0.0000	0.0156	0.0156
0.5	128	0.0000	0.0000	0.0000	0.0156	0.0156	0.0156	0.0468	0.0468	0.0156	0.0156	0.0156	0.0156
0.5	256	0.0000	-	0.0780	0.0624	0.0624	0.0312	0.0624	0.0624	0.0156	0.0156	0.0156	0.0156

Tab. M.3.: Initialization time of the different methods in 1D for log-normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0156	0.0156	0.0156	0.0312	0.0624	0.0156	0.2028	0.0468	0.0312	0.0156	-	-
5	16	0.0000	0.0000	0.0000	0.0000	0.0156	0.0000	0.0468	0.0624	0.0156	0.0156	0.0312	0.0000
5	32	0.0000	-	0.0000	0.0000	0.0000	0.0000	0.0624	0.0468	0.0156	0.0156	0.0156	0.0156
2.5	16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0468	0.0468	0.0156	0.0000	0.0000	0.0156
2.5	32	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0624	0.0468	0.0000	0.0156	0.0156	0.0156
2.5	64	0.0000	-	0.0000	0.0000	0.0000	0.0000	0.0468	0.0624	0.0000	0.0156	0.0156	0.0156
0.5	64	0.0000	0.0000	0.0000	0.0156	0.0000	0.0000	0.0468	0.0468	0.0156	0.0000	0.0156	0.0156
0.5	128	0.0000	0.0000	0.0156	0.0468	0.0156	0.0000	0.0624	0.0468	0.0156	0.0156	0.0156	0.0312
0.5	256	0.0000	-	0.0468	0.0624	0.0468	0.0624	0.0468	0.0624	0.0156	0.0156	0.0312	0.0312

Tab. M.4.: Initialization time of the different methods in 1D for log-normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0156	0.0000	0.0312	0.0000	0.0312	0.0000	0.0624	0.0468	0.0468	0.0156	-	-
5	16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0780	0.0624	0.0000	0.0000	0.0312	0.0156
5	32	0.0000	-	0.0000	0.0000	0.0000	0.0000	0.0468	0.0468	0.0000	0.0000	0.0156	0.0156
2.5	16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0468	0.0468	0.0000	0.0000	0.0156	0.0000
2.5	32	0.0000	0.0000	0.0000	0.0156	0.0000	0.0000	0.0624	0.0468	0.0000	0.0156	0.0000	0.0000
2.5	64	0.0000	-	0.0156	0.0000	0.0156	0.0156	0.0468	0.0468	0.0000	0.0156	0.0156	0.0312
0.5	64	0.0000	0.0000	0.0000	0.0000	0.0000	0.0156	0.0624	0.0468	0.0000	0.0000	0.0156	0.0312
0.5	128	0.0000	0.0156	0.0156	0.0000	0.0156	0.0000	0.0624	0.0780	0.0156	0.0156	0.0156	0.0156
0.5	256	0.0156	-	0.0156	0.0156	0.0624	0.0468	0.0624	0.0468	0.0156	0.0156	0.0156	0.0312

Tab. M.5.: Realization time of the different methods in 1D for the realization of 2000 normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0312	0.0312	0.0312	0.0156	0.0312	0.0156	0.8892	0.8892	0.1248	0.1092	-	-
5	16	0.0156	0.0000	0.0156	0.0156	0.0156	0.0000	1.5132	1.5600	0.1248	0.1248	0.1560	0.1404
5	32	0.0156	-	0.0624	0.0000	0.0000	0.0156	2.8392	2.8704	0.3276	0.3120	0.3900	0.3588
2.5	16	0.0000	0.0000	0.0000	0.0000	0.0156	0.0156	1.4976	1.5444	0.1248	0.1248	0.1248	0.1404
2.5	32	0.0156	0.0000	0.0156	0.0156	0.0000	0.0156	2.8392	2.8860	0.3900	0.3432	0.3588	0.3588
2.5	64	0.0156	-	0.0624	0.0156	0.0156	0.0156	5.5848	5.5068	0.7020	0.7176	0.7488	0.7488
0.5	64	0.0000	0.0156	0.0156	0.0000	0.0156	0.0156	5.5068	5.6160	0.7332	0.6864	0.7488	0.7644
0.5	128	0.1092	0.1872	0.0936	0.1248	0.0780	0.1560	11.0449	11.1073	1.4352	1.3572	1.4508	1.4508
0.5	256	0.2028	-	0.3120	0.2184	0.2652	0.3588	21.5749	22.3393	2.9640	3.0108	3.0576	3.0576

Tab. M.6.: Realization time of the different methods in 1D for the realization of 2000 normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0156	0.0000	0.0468	0.0312	0.0468	0.0000	0.9516	0.9516	0.2652	0.1248	-	-
5	16	0.0156	0.0000	0.0156	0.0000	0.0156	0.0000	1.5444	1.5288	0.1560	0.1404	0.1716	0.1404
5	32	0.0156	-	0.0000	0.0000	0.0156	0.0000	2.8236	2.8392	0.3744	0.3744	0.3744	0.3744
2.5	16	0.0156	0.0000	0.0000	0.0000	0.0000	0.0156	1.5444	1.5132	0.1248	0.1248	0.1404	0.1404
2.5	32	0.0156	0.0000	0.0000	0.0156	0.0156	0.0156	2.8548	2.8392	0.3588	0.3744	0.3744	0.3588
2.5	64	0.0000	-	0.0312	0.0156	0.0156	0.0156	5.4288	5.5224	0.7332	0.7488	0.8268	0.7644
0.5	64	0.0156	0.0000	0.0624	0.0468	0.0156	0.0156	5.4288	5.3976	0.7644	0.7332	0.7800	0.7644
0.5	128	0.0780	0.1248	0.1092	0.0780	0.0780	0.0780	10.7173	10.7173	1.3884	1.3884	1.4508	1.4664
0.5	256	0.1872	-	0.2652	0.2496	0.2652	0.1716	20.9821	21.2161	3.1512	3.1044	3.1668	3.1044

Tab. M.7.: Realization time of the different methods in 1D for the realization of 2000 log-normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0312	0.0312	0.0156	0.0468	0.0780	0.0156	1.0452	0.8892	0.1560	0.1248	-	-
5	16	0.0156	0.0156	0.0156	0.0312	0.0312	0.0156	1.5288	1.5288	0.1404	0.1404	0.3276	0.1248
5	32	0.0156	-	0.0156	0.0624	0.0156	0.0000	2.8704	2.8080	0.3432	0.3276	0.3900	0.3900
2.5	16	0.0156	0.0156	0.0156	0.0000	0.0156	0.0156	1.5444	1.5756	0.1404	0.1404	0.1404	0.1404
2.5	32	0.0156	0.0156	0.0156	0.0156	0.0156	0.0000	2.8080	2.9328	0.3744	0.3432	0.3900	0.3744
2.5	64	0.0156	-	0.0156	0.0156	0.0156	0.0156	5.3820	5.5848	0.7332	0.7020	0.7644	0.7644
0.5	64	0.0156	0.0156	0.0156	0.0312	0.0156	0.0156	5.4132	5.6160	0.7332	0.6708	0.7800	0.7644
0.5	128	0.1248	0.1404	0.1404	0.2184	0.1716	0.1560	10.5925	10.9201	1.4040	1.3572	1.4508	1.4508
0.5	256	0.2652	-	0.2964	0.2652	0.3120	0.3120	20.6857	21.4969	3.1512	3.1044	3.1356	3.2136

Tab. M.8.: Realization time of the different methods in 1D for the realization of 2000 log-normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0312	0.0000	0.0468	0.0156	0.0312	0.0000	0.8892	0.8580	0.1560	0.1248	-	-
5	16	0.0000	0.0156	0.0156	0.0156	0.0156	0.0000	1.5132	1.4820	0.1248	0.1248	0.2340	0.1404
5	32	0.0000	-	0.0156	0.0624	0.0000	0.0000	2.7144	2.7144	0.3120	0.3432	0.3900	0.3900
2.5	16	0.0000	0.0156	0.0156	0.0156	0.0312	0.0156	1.4820	1.4820	0.1248	0.1404	0.1248	0.1404
2.5	32	0.0156	0.0156	0.0000	0.0156	0.0000	0.0000	2.7300	2.6520	0.3120	0.3276	0.3588	0.3744
2.5	64	0.0156	-	0.0312	0.0156	0.0312	0.0312	5.2884	5.1480	0.7176	0.6864	0.7956	0.7800
0.5	64	0.0156	0.0156	0.0156	0.0156	0.0156	0.0312	5.1948	5.2260	0.7332	0.7332	0.7332	0.7800
0.5	128	0.0780	0.1404	0.1404	0.0780	0.1404	0.0780	10.3117	10.3585	1.3884	1.4196	1.4664	1.4820
0.5	256	0.2652	-	0.3276	0.3276	0.3120	0.2496	20.5453	20.1865	3.0732	3.1200	3.1356	3.0732

Tab. M.9.: Time to determine the statistical properties of the generated fields for the different methods in 1D. The fields are normal distributed with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.2184	0.2184	0.1872	0.1248	0.2028	0.1248	1.0140	1.0140	0.2496	0.2184	-	-
5	16	0.1560	0.1872	0.1716	0.1560	0.1560	0.1404	1.6692	1.7004	0.2652	0.2652	0.2964	0.2808
5	32	0.2496	-	0.3120	0.2340	0.2652	0.2340	3.0576	3.1044	0.5460	0.5460	0.6864	0.5772
2.5	16	0.1404	0.1404	0.1404	0.1404	0.1872	0.1404	1.6380	1.7004	0.2808	0.2652	0.2652	0.2652
2.5	32	0.3120	0.2340	0.2340	0.2340	0.2340	0.2340	3.0888	3.1200	0.6084	0.0000	0.6240	0.6084
2.5	64	0.6084	-	0.6708	0.6552	0.6396	0.6084	6.2088	6.1152	1.3104	1.3260	1.3416	1.3572
0.5	64	0.6084	0.6240	0.6084	0.6084	0.6084	0.6084	6.1152	6.2244	1.3104	1.2792	1.3416	1.3416
0.5	128	2.4804	2.5116	2.4180	2.4492	2.3400	2.4492	13.4161	13.4941	3.8844	3.7284	3.7752	3.7752
0.5	256	11.6689	-	12.6673	12.1681	11.6377	11.8093	33.9458	34.8818	15.5533	14.2741	14.2897	15.3817

Tab. M.10.: Time to determine the statistical properties of the generated fields for the different methods in 1D. The fields are normal distributed with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.2028	0.1404	0.2184	0.2184	0.2184	0.1248	1.0608	1.0608	0.3900	0.2340	-	-
5	16	0.1560	0.1404	0.1560	0.1716	0.1560	0.1404	1.6848	1.6692	0.3120	0.2496	0.3432	0.2808
5	32	0.2652	-	0.2652	0.2340	0.2340	0.2184	3.0576	3.0576	0.5928	0.5772	0.6084	0.5928
2.5	16	0.1560	0.1404	0.1404	0.1404	0.1404	0.1404	1.6692	1.6536	0.2808	0.2496	0.2964	0.2808
2.5	32	0.2340	0.2340	0.2184	0.2652	0.2340	0.2340	3.0888	3.0576	0.5928	0.5928	0.6084	0.5928
2.5	64	0.5928	-	0.6552	0.6240	0.7020	0.5928	6.0372	6.1152	1.3416	1.3572	1.4196	1.3728
0.5	64	0.6552	0.5928	0.6708	0.6864	0.5928	0.5928	6.0528	6.0060	1.3416	1.3104	1.3884	1.3572
0.5	128	2.3556	2.3868	2.4804	2.5584	2.3868	2.2932	12.9637	12.9793	3.7752	3.6504	3.7752	3.7440
0.5	256	11.6689	-	12.3397	12.5737	11.1229	11.4505	32.5886	32.3234	15.0697	14.7109	15.1165	14.8825

Tab. M.11.: Time to determine the statistical properties of the generated fields for the different methods in 1D. The fields are lognormal distributed with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.1872	0.2028	0.2184	0.2652	0.2652	0.1248	1.1700	0.9984	0.2652	0.2340	-	-
5	16	0.1560	0.1560	0.1716	0.1716	0.1560	0.1560	1.6692	1.6692	0.2652	0.2808	0.5616	0.2652
5	32	0.2340	-	0.2496	0.2964	0.2340	0.2184	3.1044	3.0420	0.5772	0.5616	0.6240	0.6084
2.5	16	0.2184	0.1560	0.1560	0.1404	0.1404	0.1560	1.7004	1.7160	0.2808	0.2808	0.2808	0.2652
2.5	32	0.2340	0.2808	0.2340	0.2340	0.2652	0.2340	3.0420	3.1668	0.6396	0.5772	0.6240	0.6084
2.5	64	0.6240	-	0.6084	0.6084	0.6084	0.5928	5.9592	6.1932	1.3416	1.2948	1.3572	1.3572
0.5	64	0.6240	0.6084	0.6084	0.6864	0.6240	0.6708	5.9904	6.2244	1.3416	1.2948	1.3728	1.3572
0.5	128	2.4804	2.4024	2.3556	2.5740	2.8392	2.5428	12.8077	13.2289	3.7752	3.6504	3.7284	3.6972
0.5	256	11.6377	-	11.9185	11.9653	11.8249	11.5597	32.0738	34.0394	15.0697	14.7733	14.6485	15.8185

Tab. M.12.: Time to determine the statistical properties of the generated fields for the different methods in 1D. The fields are log-normal distributed with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.3120	0.1248	0.2184	0.1248	0.2028	0.1248	1.0296	0.9672	0.2964	0.2496	-	-
5	16	0.1404	0.1404	0.1560	0.1560	0.1560	0.1404	1.6536	1.6380	0.2808	0.2808	0.4056	0.2808
5	32	0.2652	-	0.2496	0.2964	0.2184	0.2184	2.9328	2.9328	0.6396	0.5772	0.6084	0.6240
2.5	16	0.1404	0.1404	0.1560	0.1560	0.1872	0.1560	1.6224	1.6224	0.2652	0.2652	0.2808	0.2808
2.5	32	0.2340	0.2340	0.2652	0.2340	0.2184	0.2184	2.9484	2.8860	0.5460	0.5616	0.5928	0.5928
2.5	64	0.6240	-	0.6084	0.6084	0.6084	0.5928	5.8812	5.7408	1.2792	1.3104	1.3884	1.3728
0.5	64	0.6084	0.6396	0.6084	0.6084	0.5928	0.6084	5.7720	5.8188	1.3260	1.3416	1.3260	1.3728
0.5	128	2.3868	2.3868	2.4024	2.3868	2.3712	2.4024	12.5893	12.5893	3.6348	3.7752	3.7596	3.8220
0.5	256	11.8249	-	11.6221	11.5753	11.6845	11.1073	32.3078	31.3094	15.5533	15.3661	14.6797	15.5689

Tab. M.13.: Mean values of the mean values of the 2000 normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0109	0.0024	0.0007	-0.0031	0.0110	-0.0043	0.0187	0.0096	0.0139	-0.0052	-	-
5	16	-0.0088	-0.0068	-0.0032	-0.0116	-0.0014	0.0023	-0.0057	0.0242	-0.0039	-0.0095	0.0101	0.0006
5	32	-0.0100	-	-0.0018	0.0154	-0.0193	0.0004	-0.0093	0.0095	-0.0037	-0.0159	-0.0103	-0.0121
2.5	16	-0.0069	-0.0087	-0.0064	-0.0020	-0.0104	0.0095	-0.0159	0.0111	0.0046	0.0035	-0.0106	-0.0042
2.5	32	-0.0133	-0.0015	-0.0162	0.0025	-0.0268	-0.0030	0.0130	0.0100	0.0038	0.0128	-0.0067	0.0091
2.5	64	-0.0143	-	-0.0100	-0.0088	0.0074	0.0178	0.0143	-0.0011	-0.0023	0.0023	0.0242	0.0090
0.5	64	-0.0048	-0.0078	-0.0027	0.0030	0.0021	0.0015	0.0043	-0.0017	0.0004	0.0017	0.0126	0.0089
0.5	128	0.0015	-0.0001	-0.0019	-0.0073	-0.0010	-0.0044	0.0015	-0.0071	-0.0048	-0.0016	0.0040	-0.0011
0.5	256	-0.0027	-	0.0061	-0.0040	-0.0002	0.0002	-0.0038	-0.0057	0.0048	-0.0005	-0.0063	0.0045

Tab. M.14.: Mean values of the mean values of the 2000 normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0072	0.0200	-0.0007	-0.0227	0.0022	0.0289	0.0066	0.0124	-0.0066	-0.0063	-	-
5	16	-0.0386	-0.0075	0.0123	-0.0050	-0.0226	-0.0179	-0.0342	-0.0242	0.0023	-0.0288	-0.0360	-0.0205
5	32	-0.0098	-	-0.0052	0.0242	0.0061	-0.0065	0.0171	0.0149	-0.0026	0.0065	0.0017	0.0191
2.5	16	-0.0259	0.0016	-0.0071	-0.0038	-0.0032	0.0152	-0.0194	0.0122	-0.0125	-0.0068	0.0125	-0.0044
2.5	32	0.0030	-0.0192	0.0278	0.0034	-0.0104	-0.0147	-0.0200	0.0192	0.0081	-0.0085	0.0027	-0.0074
2.5	64	0.0062	-	-0.0144	-0.0130	-0.0198	-0.0221	-0.0214	0.0146	-0.0156	-0.0351	0.0017	-0.0142
0.5	64	-0.0134	-0.0348	-0.0259	-0.0101	0.0259	0.0050	-0.0117	-0.0368	0.0101	-0.0023	0.0061	-0.0027
0.5	128	0.0245	-0.0128	-0.0188	0.0139	-0.0325	-0.0103	-0.0352	0.0154	-0.0075	0.0060	-0.0167	-0.0115
0.5	256	-0.0124	-	-0.0037	-0.0335	0.0511	0.0010	-0.0137	-0.0092	-0.0183	-0.0196	-0.0405	-0.0044

Tab. M.15.: Mean values of the mean values of the 2000 log-normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	1.0106	0.9809	0.9891	1.0009	1.0017	0.9885	0.9902	1.0007	0.9200	0.9576	-	-
5	16	0.9842	0.9939	1.0166	0.9838	0.9981	1.0030	0.9953	0.9826	0.9748	0.9891	1.0213	1.0225
5	32	1.0316	-	0.9841	0.9917	1.0086	1.0006	1.0026	1.0324	0.9636	1.0128	1.1468	1.1110
2.5	16	0.9988	1.0087	1.0078	1.0059	1.0116	1.0008	0.9872	0.9997	0.9510	0.9761	1.0154	1.0621
2.5	32	0.9917	1.0152	1.0041	0.9939	0.9973	1.0063	0.9921	1.0040	0.9685	0.9848	1.1374	1.2452
2.5	64	1.0093	-	1.0002	0.9958	0.9994	0.9857	1.0031	0.9979	0.9733	1.0051	1.2586	1.4531
0.5	64	0.9939	0.9934	0.9932	0.9995	0.9963	1.0030	0.9751	0.9997	0.9310	0.9803	1.1595	1.3114
0.5	128	1.0004	0.9968	0.9981	0.9990	1.0003	0.9973	0.9935	1.0066	0.9529	0.9794	1.1086	1.2428
0.5	256	0.9973	-	0.9915	0.9970	0.9975	0.9965	1.0016	1.0083	0.9842	0.9926	1.1854	1.3535

Tab. M.16.: Mean values of the mean values of the 2000 log-normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.9811	0.9932	1.0145	0.9968	1.0358	1.0126	0.9938	1.0137	0.9537	0.9832	-	-
5	16	0.9935	0.9848	0.9796	0.9915	0.9674	0.9713	1.0068	1.0013	0.9553	1.0073	0.9817	1.0481
5	32	1.0101	-	1.0050	1.0098	0.9822	1.0266	0.9845	0.9944	0.9739	0.9980	1.0341	1.0681
2.5	16	1.0005	0.9983	0.9970	0.9964	0.9934	0.9886	0.9719	1.0070	0.9972	1.0168	0.9960	1.0480
2.5	32	0.9838	0.9701	1.0013	1.0270	0.9929	0.9969	0.9853	1.0167	0.9896	0.9937	1.0804	1.1010
2.5	64	1.0004	-	1.0099	0.9648	0.9873	1.0119	0.9982	1.0126	0.9831	0.9890	1.1399	1.1421
0.5	64	0.9894	0.9925	1.0032	0.9830	0.9956	0.9994	0.9796	1.0196	0.9526	0.9725	1.0864	1.1958
0.5	128	0.9852	1.0185	0.9964	0.9883	0.9993	1.0141	0.9585	1.0128	0.9655	0.9900	1.0770	1.1108
0.5	256	0.9889	-	0.9944	1.0007	1.0136	0.9678	0.9943	0.9863	0.9970	0.9980	1.0939	1.1557

Tab. M.17.: Deviation in the mean values of the 2000 normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.6242	0.6232	0.6240	0.6067	0.6295	0.6141	0.6275	0.5963	0.6352	0.6726	-	-
5	16	0.6200	0.6346	0.6479	0.6287	0.6413	0.6427	0.6414	0.6358	0.6502	0.6357	0.6553	0.6421
5	32	0.6300	-	0.6294	0.6291	0.6234	0.6698	0.6448	0.6332	0.6416	0.6390	0.6426	0.6581
2.5	16	0.4800	0.4655	0.4894	0.4718	0.4867	0.4717	0.4824	0.4705	0.5019	0.4790	0.4969	0.4813
2.5	32	0.4970	0.4797	0.4921	0.4785	0.4973	0.4721	0.4957	0.4682	0.4996	0.4698	0.4972	0.4903
2.5	64	0.4936	-	0.4787	0.4868	0.4917	0.7134	0.4889	0.4797	0.4988	0.4717	0.4878	0.4834
0.5	64	0.2376	0.2226	0.2442	0.2281	0.2391	0.2176	0.2369	0.2199	0.2320	0.2221	0.2354	0.2237
0.5	128	0.2346	0.2151	0.2345	0.2221	0.2410	0.2225	0.2380	0.2233	0.2340	0.2248	0.2337	0.2246
0.5	256	0.2384	-	0.2344	0.2258	0.2368	0.2246	0.2294	0.2240	0.2395	0.2231	0.2337	0.2197

Tab. M.18.: Deviation in the mean values of the 2000 normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.8356	0.8253	0.8113	0.8287	0.8348	0.8253	0.8199	0.8225	0.8459	0.8511	-	-
5	16	0.8067	0.8229	0.8417	0.8184	0.8496	0.8310	0.8212	0.8064	0.8453	0.8498	0.8510	0.8301
5	32	0.8434	-	0.8498	0.8295	0.8135	0.8687	0.8531	0.8466	0.8335	0.8600	0.8467	0.8511
2.5	16	0.7762	0.7740	0.8012	0.7860	0.7773	0.7609	0.7707	0.7919	0.7871	0.7884	0.7793	0.7756
2.5	32	0.7937	0.7950	0.8024	0.7901	0.7978	0.8018	0.8050	0.7650	0.7924	0.7891	0.8065	0.7813
2.5	64	0.7868	-	0.7918	0.7975	0.8046	0.9979	0.7679	0.7990	0.7748	0.7857	0.7829	0.8131
0.5	64	0.7239	0.7503	0.7306	0.7394	0.7228	0.7393	0.7394	0.7387	0.6883	0.7187	0.7297	0.7090
0.5	128	0.7399	0.7319	0.7354	0.7451	0.7191	0.7187	0.7323	0.7136	0.7388	0.7255	0.7179	0.7379
0.5	256	0.7351	-	0.7330	0.7351	0.7248	0.7068	0.7409	0.7464	0.7328	0.7239	0.7356	0.7165

Tab. M.19.: Deviation in the mean values of the 2000 log-normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.5587	0.5434	0.5092	0.5564	0.5279	0.5302	0.5785	0.6143	0.5245	0.5449	-	-
5	16	0.5349	0.5717	0.5331	0.5500	0.5297	0.5813	0.6017	0.5696	0.5446	0.6019	0.5784	0.6393
5	32	0.5966	-	0.5264	0.6114	0.5640	0.5864	0.6050	0.6538	0.5148	0.5942	0.8330	0.7309
2.5	16	0.3958	0.4487	0.4160	0.4300	0.4189	0.4361	0.4518	0.4521	0.3908	0.4056	0.4589	0.5171
2.5	32	0.3943	0.4397	0.4317	0.4205	0.4136	0.4427	0.4397	0.4512	0.3999	0.4325	0.6548	0.9132
2.5	64	0.4079	-	0.4020	0.4160	0.4123	0.4372	0.4566	0.4560	0.4101	0.4349	0.7415	2.4778
0.5	64	0.2023	0.1989	0.1976	0.1996	0.2013	0.2055	0.2019	0.2156	0.1786	0.1990	0.5006	0.9887
0.5	128	0.2029	0.1932	0.1934	0.2018	0.1977	0.1961	0.2125	0.2131	0.1865	0.1956	0.3571	0.7971
0.5	256	0.1953	-	0.1993	0.1973	0.1925	0.1960	0.2150	0.2190	0.1935	0.1973	0.4506	0.8430

Tab. M.20.: Deviation in the mean values of the 2000 log-normal distributed fields in 1D, with a threshold value of 0.5

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.7224	0.7367	0.7578	0.7334	0.7712	0.8010	0.7459	0.8184	0.6909	0.7413	-	-
5	16	0.7752	0.7526	0.6831	0.7836	0.6633	0.7118	0.7874	0.7605	0.6986	0.7558	0.7321	0.8723
5	32	0.7427	-	0.7878	0.7462	0.7282	0.8134	0.7828	0.7849	0.7224	0.7576	0.7632	0.8913
2.5	16	0.6959	0.6965	0.6700	0.6848	0.7008	0.7034	0.6749	0.7232	0.6966	0.7081	0.6794	0.7817
2.5	32	0.6685	0.6628	0.6766	0.6748	0.6703	0.6951	0.7565	0.7624	0.6982	0.6699	0.8174	0.8968
2.5	64	0.7073	-	0.6663	0.6327	0.6497	0.6982	0.7500	0.7468	0.6957	0.6740	0.8250	0.8834
0.5	64	0.5981	0.5901	0.6049	0.5814	0.6025	0.6255	0.6422	0.6830	0.5894	0.5722	0.7273	0.8685
0.5	128	0.6026	0.6714	0.5940	0.6083	0.5968	0.6071	0.6230	0.6580	0.5873	0.5750	0.6496	0.7225
0.5	256	0.5745	-	0.6001	0.6195	0.6099	0.5685	0.6870	0.6243	0.5861	0.6007	0.6650	0.7843

5.

Tab. M.21.: Mean value of the deviation in the 2000 normal distributed fields with in 1D, a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.7278	0.7240	0.7444	0.7172	0.7481	0.7260	0.7170	0.7384	0.6134	0.6547	-	-
5	16	0.7392	0.7002	0.7367	0.7143	0.7410	0.7156	0.7239	0.7221	0.6735	0.6962	0.7499	0.7360
5	32	0.7452	-	0.7447	0.6969	0.7402	0.7443	0.7330	0.6992	0.7145	0.6938	0.8414	0.7882
2.5	16	0.8454	0.8438	0.8464	0.8561	0.8476	0.8473	0.8228	0.8370	0.7565	0.8118	0.8467	0.9265
2.5	32	0.8363	0.8521	0.8432	0.8416	0.8491	0.8357	0.8406	0.8475	0.8002	0.8268	0.9824	1.0283
2.5	64	0.8455	-	0.8413	0.8333	0.8493	1.2835	0.8390	0.8278	0.8188	0.8340	1.0493	1.0773
0.5	64	0.9653	0.9697	0.9658	0.9640	0.9649	0.9667	0.9387	0.9666	0.8710	0.9382	1.1219	1.2102
0.5	128	0.9653	0.9723	0.9693	0.9643	0.9657	0.9685	0.9528	0.9638	0.9156	0.9535	1.0598	1.1372
0.5	256	0.9646	-	0.9649	0.9676	0.9688	0.9625	0.9616	0.9637	0.9386	0.9595	1.1016	1.1652

Tab. M.22.: Mean value of the deviation in the 2000 normal distributed fields with in 1D, a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.5245	0.5092	0.5644	0.5108	0.5312	0.5005	0.5072	0.5123	0.4561	0.4860	-	-
5	16	0.5223	0.5030	0.5390	0.5035	0.5208	0.5035	0.5174	0.5088	0.4848	0.4908	0.5354	0.5253
5	32	0.5257	-	0.5354	0.4948	0.5249	0.6658	0.5130	0.4950	0.5093	0.4987	0.6074	0.5660
2.5	16	0.5977	0.5945	0.6243	0.5954	0.5988	0.5964	0.5798	0.6049	0.5366	0.5758	0.6067	0.6532
2.5	32	0.6006	0.5971	0.6138	0.5965	0.6034	0.5915	0.5863	0.5963	0.5722	0.5873	0.7002	0.7402
2.5	64	0.5992	-	0.6016	0.5927	0.5969	1.2622	0.5908	0.5902	0.5873	0.5932	0.7500	0.7785
0.5	64	0.6796	0.6829	0.6862	0.6819	0.6792	0.6840	0.6630	0.6888	0.6196	0.6617	0.7907	0.8556
0.5	128	0.6847	0.6867	0.6838	0.6836	0.6831	0.6786	0.6710	0.6812	0.6508	0.6780	0.7499	0.8072
0.5	256	0.6831	-	0.6830	0.6849	0.6808	0.6802	0.6784	0.6839	0.6681	0.6819	0.7829	0.8242

Tab. M.23.: Mean value of the deviation in the 2000 log-normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.6755	0.6153	0.6513	0.6382	0.6695	0.6172	0.5967	0.6026	0.4997	0.5373	-	-
5	16	0.6623	0.6164	0.6776	0.6020	0.6712	0.6181	0.6150	0.5681	0.5931	0.6003	0.7585	0.7123
5	32	0.7018	-	0.6676	0.6088	0.6942	0.6122	0.6318	0.6095	0.6247	0.6228	1.1386	0.9627
2.5	16	0.7738	0.7679	0.7767	0.7562	0.7896	0.7522	0.7091	0.7264	0.6193	0.6985	0.8203	0.9753
2.5	32	0.7701	0.7602	0.7978	0.7407	0.7792	0.7571	0.7312	0.7252	0.6980	0.7142	1.2975	1.5998
2.5	64	0.7915	-	0.7830	0.7411	0.7948	0.7333	0.7434	0.7236	0.7254	0.7311	1.7334	2.4667
0.5	64	0.9104	0.9080	0.9064	0.9109	0.9177	0.9296	0.8509	0.9092	0.7276	0.8603	1.6104	2.3050
0.5	128	0.9277	0.9099	0.9225	0.9144	0.9249	0.9193	0.8918	0.9225	0.8106	0.8708	1.5257	2.2761
0.5	256	0.9301	-	0.9168	0.9137	0.9209	0.9110	0.9123	0.9150	0.8704	0.9073	1.9221	2.9385

Tab. M.24.: Mean value of the deviation in the 2000 log-normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.4849	0.4780	0.4938	0.4757	0.4965	0.4731	0.4179	0.4342	0.3928	0.4368	-	-
5	16	0.4943	0.4628	0.4801	0.4614	0.4810	0.4468	0.4342	0.4094	0.4421	0.4575	0.5074	0.5272
5	32	0.4997	-	0.4964	0.4646	0.4845	0.4746	0.4335	0.4096	0.4638	0.4564	0.6620	0.6506
2.5	16	0.5637	0.5533	0.5653	0.5591	0.5553	0.5601	0.4871	0.5214	0.4961	0.5516	0.5964	0.6981
2.5	32	0.5614	0.5368	0.5734	0.5861	0.5645	0.5522	0.4960	0.5072	0.5287	0.5475	0.8415	0.9563
2.5	64	0.5699	-	0.5770	0.5327	0.5700	0.5659	0.5120	0.5112	0.5582	0.5533	0.9959	1.0698
0.5	64	0.6614	0.6663	0.6710	0.6545	0.6612	0.6604	0.5724	0.6180	0.5636	0.6209	1.0035	1.3682
0.5	128	0.6590	0.6799	0.6690	0.6587	0.6706	0.6709	0.5672	0.6145	0.6016	0.6481	0.9347	1.2360
0.5	256	0.6595	-	0.6710	0.6695	0.6839	0.6449	0.5995	0.5991	0.6488	0.6593	1.0791	1.4262

Tab. M.25.: Standard deviation of the deviations in the 2000 normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.2421	0.2936	0.2363	0.2965	0.2401	0.2878	0.2384	0.3034	0.2308	0.2790	-	-
5	16	0.2238	0.2865	0.2202	0.2957	0.2265	0.3040	0.2225	0.3115	0.2280	0.2966	0.2493	0.3124
5	32	0.2220	-	0.2180	0.2899	0.2150	0.3246	0.2214	0.2955	0.2292	0.2924	0.2549	0.3311
2.5	16	0.2105	0.2525	0.2115	0.2510	0.2099	0.2573	0.2097	0.2464	0.2164	0.2675	0.2441	0.3023
2.5	32	0.1988	0.2568	0.1999	0.2595	0.2087	0.2474	0.2013	0.2557	0.2168	0.2561	0.2729	0.3282
2.5	64	0.1949	-	0.1942	0.2578	0.1966	0.5183	0.2028	0.2475	0.2048	0.2618	0.2838	0.3385
0.5	64	0.1185	0.1265	0.1182	0.1279	0.1193	0.1314	0.1206	0.1275	0.1194	0.1269	0.2582	0.2721
0.5	128	0.1155	0.1324	0.1145	0.1327	0.1149	0.1280	0.1148	0.1296	0.1168	0.1272	0.1809	0.2079
0.5	256	0.1147	-	0.1119	0.1313	0.1126	0.1303	0.1158	0.1272	0.1154	0.1301	0.1832	0.2109

Tab. M.26.: Standard deviation of the deviations in the 2000 normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.1733	0.2045	0.1874	0.2076	0.1746	0.2040	0.1693	0.2054	0.1661	0.2088	-	-
5	16	0.1603	0.2073	0.1601	0.2103	0.1563	0.2114	0.1622	0.2192	0.1584	0.2052	0.1745	0.2147
5	32	0.1516	-	0.1595	0.2064	0.1503	0.3126	0.1506	0.2081	0.1599	0.2098	0.1824	0.2276
2.5	16	0.1519	0.1791	0.1573	0.1782	0.1491	0.1752	0.1485	0.1770	0.1518	0.1760	0.1726	0.2122
2.5	32	0.1445	0.1811	0.1486	0.1801	0.1444	0.1742	0.1441	0.1765	0.1498	0.1835	0.1921	0.2430
2.5	64	0.1383	-	0.1410	0.1817	0.1459	0.6545	0.1445	0.1783	0.1487	0.1821	0.2042	0.2458
0.5	64	0.0815	0.0879	0.0824	0.0903	0.0841	0.0918	0.0821	0.0929	0.0849	0.0924	0.1760	0.1870
0.5	128	0.0843	0.0917	0.0815	0.0913	0.0790	0.0899	0.0801	0.0914	0.0825	0.0926	0.1262	0.1423
0.5	256	0.0802	-	0.0785	0.0926	0.0810	0.0913	0.0813	0.0923	0.0830	0.0894	0.1305	0.1571

Tab. M.27.: Standard deviation of the deviations in the 2000 log-normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.5344	0.5141	0.4767	0.5390	0.5011	0.4980	0.4962	0.5398	0.2109	0.4471	-	-
5	16	0.4908	0.5081	0.4706	0.4875	0.4604	0.5168	0.4992	0.4680	0.4133	0.5340	0.8044	0.7831
5	32	0.5591	-	0.4761	0.5873	0.5111	0.5323	0.4899	0.5641	0.4521	0.5387	2.7115	1.3496
2.5	16	0.4553	0.5733	0.4743	0.5385	0.4942	0.5084	0.4896	0.5094	0.4606	0.4575	0.7746	1.1755
2.5	32	0.4259	0.5209	0.4866	0.4812	0.4710	0.5159	0.4458	0.5040	0.3887	0.4871	1.9926	2.9518
2.5	64	0.4467	-	0.4452	0.4859	0.4655	0.5010	0.4659	0.5255	0.4318	0.5030	2.5511	11.4292
0.5	64	0.3227	0.3451	0.3099	0.3455	0.3349	0.3937	0.3100	0.3805	0.4667	0.3291	2.3955	6.5882
0.5	128	0.3170	0.3270	0.3084	0.3501	0.3058	0.3539	0.3210	0.3888	0.2425	0.3212	1.8660	5.7186
0.5	256	0.2985	-	0.2926	0.3555	0.3024	0.3359	0.3209	0.3737	0.2710	0.3365	2.8679	5.9602

Tab. M.28.: Standard deviation of the deviations in the 2000 log-normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.4612	0.4464	0.4556	0.4424	0.4642	0.4837	0.3637	0.4354	0.3614	0.4620	-	-
5	16	0.4450	0.4603	0.3964	0.4686	0.3915	0.4013	0.3873	0.3741	0.4493	0.4347	0.4723	0.5809
5	32	0.4131	-	0.4342	0.4366	0.4163	0.4888	0.4142	0.4144	0.4382	0.4516	0.6700	0.8643
2.5	16	0.4605	0.4862	0.4518	0.4853	0.4496	0.5441	0.3914	0.4673	0.4171	0.4718	0.5930	0.8185
2.5	32	0.4525	0.4777	0.4500	0.5386	0.4591	0.4631	0.4379	0.4498	0.4541	0.4737	1.2225	2.4091
2.5	64	0.4573	-	0.4548	0.4379	0.4344	0.4782	0.4326	0.4541	0.4771	0.4629	1.2815	1.6458
0.5	64	0.4353	0.4505	0.4416	0.4185	0.4283	0.4525	0.3982	0.4475	0.3827	0.4081	1.4287	2.3377
0.5	128	0.4281	0.5016	0.4237	0.4408	0.4282	0.4370	0.3986	0.4402	0.3914	0.4178	0.8795	1.8850
0.5	256	0.4068	-	0.4385	0.4715	0.4372	0.4220	0.4375	0.4232	0.4202	0.4383	1.0946	2.7322

Tab. M.29.: Mean error in the correlation structure, estimated over 2000 normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0117	0.0077	0.0035	0.0065	0.0016	0.0058	0.0147	0.0280	0.0563	0.0550	-	-
5	16	0.0073	0.0128	0.0186	0.0068	0.0068	0.0030	0.0169	0.0126	0.0326	0.0127	0.0138	0.0238
5	32	0.0130	-	0.0087	0.0110	0.0094	0.0136	0.0124	0.0105	0.0119	0.0059	0.0432	0.0373
2.5	16	0.0053	0.0047	0.0021	0.0048	0.0059	0.0053	0.0088	0.0093	0.0289	0.0204	0.0152	0.0297
2.5	32	0.0202	0.0102	0.0077	0.0062	0.0094	0.0062	0.0088	0.0093	0.0177	0.0157	0.0321	0.0417
2.5	64	0.0052	-	0.0103	0.0107	0.0089	0.0243	0.0061	0.0101	0.0124	0.0173	0.0455	0.0474
0.5	64	0.0047	0.0055	0.0063	0.0073	0.0053	0.0054	0.0074	0.0078	0.0127	0.0102	0.0176	0.0171
0.5	128	0.0041	0.0059	0.0044	0.0073	0.0056	0.0061	0.0057	0.0049	0.0077	0.0114	0.0119	0.0142
0.5	256	0.0030	-	0.0050	0.0052	0.0039	0.0070	0.0045	0.0074	0.0088	0.0100	0.0118	0.0149

Tab. M.30.: Mean error in the correlation structure, estimated over 2000 normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0018	0.0063	0.0165	0.0026	0.0053	0.0075	0.0070	0.0019	0.0416	0.0465	-	-
5	16	0.0154	0.0068	0.0065	0.0094	0.0098	0.0041	0.0085	0.0266	0.0245	0.0091	0.0120	0.0303
5	32	0.0033	-	0.0061	0.0087	0.0149	0.1322	0.0177	0.0099	0.0118	0.0073	0.0693	0.0504
2.5	16	0.0063	0.0034	0.0063	0.0063	0.0069	0.0090	0.0048	0.0045	0.0357	0.0158	0.0217	0.0531
2.5	32	0.0054	0.0055	0.0048	0.0066	0.0046	0.0173	0.0204	0.0098	0.0201	0.0115	0.0736	0.1152
2.5	64	0.0015	-	0.0066	0.0104	0.0131	0.2592	0.0100	0.0158	0.0190	0.0136	0.1245	0.1227
0.5	64	0.0031	0.0170	0.0038	0.0108	0.0027	0.0085	0.0223	0.0073	0.0219	0.0159	0.0970	0.1448
0.5	128	0.0065	0.0039	0.0071	0.0138	0.0047	0.0029	0.0109	0.0063	0.0322	0.0118	0.0809	0.1004
0.5	256	0.0061	-	0.0063	0.0082	0.0028	0.0111	0.0115	0.0141	0.0188	0.0115	0.0942	0.1308

Tab. M.31.: Mean error in the correlation structure, estimated over 2000 log-normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
L_c	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0779	0.0525	0.0883	0.0585	0.0891	0.0548	0.0330	0.0186	0.0370	0.0571	-	-
5	16	0.0799	0.0552	0.0795	0.0475	0.0846	0.0616	0.0287	0.0245	0.0540	0.0495	0.1526	0.1033
5	32	0.0788	-	0.0889	0.0522	0.0790	0.0499	0.0319	0.0255	0.0933	0.0621	0.2336	0.1599
2.5	16	0.0552	0.0286	0.0452	0.0284	0.0523	0.0385	0.0167	0.0191	0.0248	0.0322	0.0710	0.0847
2.5	32	0.0505	0.0266	0.0458	0.0290	0.0442	0.0338	0.0237	0.0161	0.0423	0.0243	0.1085	0.0995
2.5	64	0.0519	-	0.0530	0.0355	0.0507	0.0323	0.0340	0.0191	0.0410	0.0280	0.1162	0.1043
0.5	64	0.0112	0.0079	0.0116	0.0094	0.0129	0.0092	0.0093	0.0067	0.0068	0.0098	0.0195	0.0213
0.5	128	0.0138	0.0101	0.0127	0.0094	0.0122	0.0105	0.0080	0.0081	0.0111	0.0105	0.0212	0.0191
0.5	256	0.0120	-	0.0129	0.0099	0.0111	0.0090	0.0085	0.0079	0.0110	0.0110	0.0224	0.0204

Tab. M.32.: Mean error in the correlation structure, estimated over 2000 log-normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
L_c	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.1554	0.1400	0.1312	0.1323	0.1307	0.1068	0.0505	0.0520	0.1191	0.1673	-	-
5	16	0.1280	0.1457	0.1579	0.1284	0.1736	0.1257	0.0529	0.0503	0.1762	0.1427	0.1774	0.1529
5	32	0.1442	-	0.1186	0.1575	0.1472	0.1337	0.0865	0.0729	0.1594	0.1426	0.3000	0.2889
2.5	16	0.1414	0.1436	0.1615	0.1544	0.1282	0.1609	0.0800	0.0996	0.0980	0.1413	0.2252	0.2559
2.5	32	0.1622	0.1591	0.1604	0.2085	0.1664	0.1410	0.0598	0.0593	0.1334	0.1721	0.3621	0.4524
2.5	64	0.1417	-	0.1787	0.1712	0.1755	0.1517	0.0662	0.0780	0.1725	0.1679	0.3981	0.4230
0.5	64	0.1679	0.1740	0.1687	0.1618	0.1602	0.1468	0.0688	0.0787	0.1085	0.1613	0.3655	0.4238
0.5	128	0.1579	0.1395	0.1706	0.1551	0.1707	0.1598	0.0847	0.0953	0.1354	0.1714	0.3322	0.4283
0.5	256	0.1720	-	0.1725	0.1618	0.1674	0.1753	0.0656	0.1038	0.1713	0.1704	0.3764	0.4605

Tab. M.33.: Standard deviation of the errors in the correlation structure, estimated over 2000 normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0103	0.0077	0.0038	0.0056	0.0009	0.0034	0.0138	0.0171	0.0375	0.0452	-	-
5	16	0.0065	0.0101	0.0083	0.0041	0.0042	0.0024	0.0104	0.0106	0.0169	0.0080	0.0232	0.0333
5	32	0.0097	-	0.0042	0.0085	0.0063	0.0084	0.0090	0.0065	0.0098	0.0045	0.0487	0.0498
2.5	16	0.0055	0.0035	0.0024	0.0055	0.0046	0.0054	0.0076	0.0122	0.0328	0.0173	0.0302	0.0631
2.5	32	0.0241	0.0068	0.0072	0.0039	0.0049	0.0067	0.0061	0.0051	0.0191	0.0122	0.0590	0.0919
2.5	64	0.0040	-	0.0078	0.0066	0.0087	0.0255	0.0032	0.0073	0.0112	0.0122	0.0710	0.1021
0.5	64	0.0037	0.0041	0.0055	0.0067	0.0055	0.0039	0.0071	0.0062	0.0161	0.0134	0.0310	0.0353
0.5	128	0.0050	0.0055	0.0035	0.0088	0.0042	0.0049	0.0062	0.0043	0.0112	0.0137	0.0222	0.0394
0.5	256	0.0022	-	0.0050	0.0043	0.0032	0.0081	0.0037	0.0068	0.0090	0.0135	0.0277	0.0415

Tab. M.34.: Standard deviation of the errors in the correlation structure, estimated over 2000 normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0022	0.0056	0.0093	0.0027	0.0043	0.0047	0.0050	0.0016	0.0265	0.0487	-	-
5	16	0.0080	0.0040	0.0036	0.0064	0.0053	0.0023	0.0055	0.0143	0.0087	0.0081	0.0149	0.0165
5	32	0.0033	-	0.0079	0.0081	0.0057	0.0736	0.0062	0.0082	0.0163	0.0041	0.0283	0.0223
2.5	16	0.0038	0.0038	0.0060	0.0042	0.0067	0.0079	0.0041	0.0049	0.0146	0.0113	0.0190	0.0357
2.5	32	0.0038	0.0049	0.0033	0.0055	0.0043	0.0086	0.0046	0.0056	0.0131	0.0167	0.0361	0.0486
2.5	64	0.0013	-	0.0075	0.0049	0.0063	0.0949	0.0081	0.0072	0.0260	0.0209	0.0453	0.0583
0.5	64	0.0050	0.0060	0.0019	0.0044	0.0032	0.0045	0.0038	0.0049	0.0222	0.0240	0.0519	0.0523
0.5	128	0.0031	0.0025	0.0030	0.0045	0.0023	0.0026	0.0039	0.0038	0.0132	0.0249	0.0605	0.0612
0.5	256	0.0031	-	0.0030	0.0057	0.0027	0.0050	0.0044	0.0047	0.0265	0.0310	0.0699	0.0708

Tab. M.35.: Standard deviation of the errors in the correlation structure, estimated over 2000 log-normal distributed fields in 1D, with a threshold value of 0.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0576	0.0628	0.0623	0.0639	0.0567	0.0592	0.0137	0.0134	0.0212	0.0455	-	-
5	16	0.0543	0.0508	0.0614	0.0538	0.0478	0.0487	0.0231	0.0254	0.0431	0.0496	0.1058	0.1230
5	32	0.0466	-	0.0552	0.0582	0.0579	0.0542	0.0199	0.0311	0.0444	0.0621	0.2076	0.1946
2.5	16	0.0509	0.0460	0.0562	0.0532	0.0487	0.0459	0.0186	0.0167	0.0192	0.0352	0.1078	0.1643
2.5	32	0.0561	0.0452	0.0493	0.0463	0.0589	0.0440	0.0202	0.0250	0.0326	0.0362	0.1530	0.1946
2.5	64	0.0529	-	0.0486	0.0497	0.0546	0.0446	0.0233	0.0224	0.0465	0.0392	0.1600	0.2168
0.5	64	0.0281	0.0241	0.0297	0.0228	0.0292	0.0233	0.0111	0.0112	0.0092	0.0218	0.0508	0.0893
0.5	128	0.0291	0.0241	0.0289	0.0226	0.0297	0.0246	0.0126	0.0132	0.0183	0.0253	0.0631	0.0908
0.5	256	0.0303	-	0.0301	0.0246	0.0301	0.0240	0.0141	0.0127	0.0235	0.0263	0.0701	0.0881

Tab. M.36.: Standard deviation of the errors in the correlation structure, estimated over 2000 log-normal distributed fields in 1D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**			
		Cholesky		*Eigen*		*Modified Eigen*				*nbh=3*		*nbh=5*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp	Exp	SExp
5	8	0.0670	0.0670	0.0545	0.0622	0.0577	0.0488	0.0241	0.0241	0.0810	0.0876	-	-
5	16	0.0391	0.0571	0.0496	0.0511	0.0550	0.0495	0.0181	0.0201	0.0748	0.0566	0.0505	0.0429
5	32	0.0390	-	0.0291	0.0662	0.0365	0.0497	0.0344	0.0319	0.0496	0.0485	0.0602	0.0598
2.5	16	0.0398	0.0460	0.0457	0.0484	0.0355	0.0556	0.0263	0.0348	0.0377	0.0492	0.0651	0.0849
2.5	32	0.0366	0.0477	0.0364	0.0616	0.0426	0.0406	0.0198	0.0232	0.0397	0.0512	0.0897	0.1265
2.5	64	0.0257	-	0.0359	0.0456	0.0347	0.0376	0.0158	0.0227	0.0536	0.0466	0.0672	0.0899
0.5	64	0.0233	0.0265	0.0239	0.0261	0.0221	0.0208	0.0119	0.0128	0.0318	0.0335	0.0616	0.0616
0.5	128	0.0165	0.0184	0.0187	0.0193	0.0190	0.0209	0.0130	0.0158	0.0295	0.0306	0.0553	0.0553
0.5	256	0.0164	-	0.0175	0.0196	0.0165	0.0220	0.0091	0.0190	0.0354	0.0405	0.0490	0.0481

Tab. M.37.: Initialization time of the different methods in 2D for normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.03	0.03	0.27	0.66	0.06	0.28	1.08	0.08	0.47	0.22
5	16	0.02	-	0.06	0.11	0.06	0.05	0.06	0.05	0.33	0.33
5	32	0.20	-	1.26	1.01	3.17	0.70	0.05	0.06	0.36	0.42
2.5	16	0.02	0.06	0.06	0.06	0.06	0.05	0.06	0.05	0.30	0.34
2.5	32	0.25	-	1.34	1.09	3.14	1.79	0.05	0.06	0.34	0.41
2.5	64	5.68	-	62.18	57.10	192.02	29.84	0.06	0.08	0.42	0.48
0.5	64	5.62	114.52	68.16	62.96	193.92	371.16	0.06	0.06	0.44	0.50
0.5	128	193.52	-	3940.24	3801.39	12142.65	12091.45	0.09	0.09	1.51	1.42
0.5	256	-	-	-	-	-	-	0.25	1.19	1.31	1.51

Tab. M.38.: Initialization time of the different methods in 2D for normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.05	0.34	0.03	0.67	0.28	0.30	0.41	0.08	0.64	0.20
5	16	0.09	-	0.06	0.02	0.06	0.03	0.08	0.05	0.36	0.34
5	32	0.31	-	1.11	1.03	3.12	0.66	0.06	0.05	0.36	0.41
2.5	16	0.00	0.02	0.09	0.08	0.06	0.06	0.05	0.05	0.30	0.34
2.5	32	0.25	-	1.31	0.98	3.12	1.65	0.06	0.06	0.36	0.41
2.5	64	5.88	-	60.65	57.52	190.99	27.36	0.08	0.06	0.42	0.48
0.5	64	6.18	5.16	61.00	60.47	191.24	190.80	0.06	0.06	0.42	0.50
0.5	128	193.29	-	3803.49	3619.44	11974.15	7444.41	0.09	0.09	1.19	1.36
0.5	256	-	-	-	-	-	-	0.27	0.20	1.29	1.42

Tab. M.39.: Initialization time of the different methods in 2D for log-normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.03	0.34	0.69	0.75	0.28	0.28	0.17	0.08	0.23	0.23
5	16	0.02	-	0.06	0.05	0.06	0.06	0.06	0.05	0.37	0.44
5	32	0.27	-	1.19	1.31	3.14	1.39	0.08	0.05	0.45	0.51
2.5	16	0.02	0.02	0.08	0.06	0.06	0.06	0.06	0.05	0.39	0.42
2.5	32	0.27	-	1.26	1.19	3.17	3.12	0.05	0.06	0.45	0.53
2.5	64	5.82	-	62.43	57.83	190.21	72.17	0.06	0.06	0.53	0.61
0.5	64	5.74	5.51	61.84	61.21	192.15	190.02	0.06	0.08	0.55	0.61
0.5	128	197.56	-	3950.44	3873.15	11962.34	11972.03	0.09	0.09	1.50	1.73
0.5	256	-	-	-	-	-	-	0.23	0.22	1.61	1.79

Tab. M.40.: Initialization time of the different methods in 2D for log-normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.03	0.34	0.67	0.72	0.28	0.28	0.61	0.08	0.22	0.25
5	16	0.02	-	0.06	0.12	0.06	0.08	0.06	0.06	0.36	0.42
5	32	0.31	-	1.12	1.03	3.18	1.17	0.05	0.06	0.45	0.50
2.5	16	0.05	0.02	0.02	0.02	0.06	0.08	0.06	0.05	0.37	0.42
2.5	32	0.25	-	1.12	1.03	3.21	3.18	0.05	0.06	0.45	0.50
2.5	64	6.47	-	61.34	57.30	191.44	56.14	0.06	0.06	0.53	0.59
0.5	64	5.80	5.58	61.53	61.17	190.98	190.62	0.08	0.06	0.51	0.59
0.5	128	198.89	-	3816.83	3732.29	11966.76	11963.81	0.11	0.09	1.45	1.65
0.5	256	-	-	-	-	-	-	0.28	0.23	1.56	1.75

Tab. M.41.: Realization time of the different methods in 2D for the realization of 200 normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.05	0.05	0.37	0.66	0.08	0.28	2.32	1.31	1.36	0.56
5	16	0.08	-	0.12	0.16	0.08	0.08	4.45	4.62	1.15	1.15
5	32	0.53	-	1.70	1.45	3.56	1.19	17.43	17.60	4.87	4.73
2.5	16	0.08	0.08	0.11	0.12	0.08	0.11	4.51	4.66	1.11	1.17
2.5	32	0.64	-	1.67	1.51	3.46	2.22	16.89	17.57	4.48	4.66
2.5	64	12.70	-	69.30	64.33	198.99	36.40	68.22	69.50	19.08	19.27
0.5	64	12.68	124.27	74.71	70.04	200.94	379.72	66.74	69.65	19.66	19.38
0.5	128	301.43	-	4051.89	3912.19	12251.79	12197.69	261.86	277.79	80.47	80.70
0.5	256		-	-	-	-	-	1085.25	1129.18	319.33	322.19

Tab. M.42.: Realization time of the different methods in 2D for the realization of 200 normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.05	0.34	0.05	0.67	0.30	0.30	1.75	1.33	1.50	0.53
5	16	0.14	-	0.08	0.03	0.12	0.05	5.04	4.65	1.17	1.17
5	32	0.75	-	1.50	1.36	3.51	1.05	18.97	17.61	4.68	4.59
2.5	16	0.05	0.06	0.16	0.97	0.08	0.12	4.88	4.60	1.09	1.17
2.5	32	0.67	-	1.64	1.36	3.51	1.98	18.83	17.75	5.12	4.59
2.5	64	13.07	-	67.64	64.51	198.04	34.43	73.91	68.66	19.13	19.23
0.5	64	13.21	11.58	67.97	67.53	198.34	197.92	73.82	70.11	19.11	19.20
0.5	128	305.11	-	3915.81	3732.68	12087.13	7554.89	297.48	280.08	80.68	80.20
0.5	256	-	-	-	-	-	-	1107.01	1110.21	323.30	327.07

Tab. M.43.: Realization time of the different methods in 2D for the realization of 200 log-normal distributed fields with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.05	0.36	0.70	0.76	0.30	0.28	1.47	1.31	0.58	0.56
5	16	0.08	-	0.12	0.14	0.08	0.11	4.71	4.56	1.22	1.26
5	32	0.66	-	1.68	1.73	3.46	1.73	17.92	17.44	4.68	4.79
2.5	16	0.08	0.06	0.14	0.19	0.09	0.08	4.70	4.51	1.22	1.25
2.5	32	0.66	-	1.76	1.62	3.57	3.51	17.89	17.33	4.74	4.79
2.5	64	13.03	-	69.70	65.33	197.33	79.28	70.17	68.98	19.19	19.50
0.5	64	12.90	12.53	68.98	68.47	199.28	197.19	69.37	68.47	19.03	19.33
0.5	128	299.48	-	4060.35	3985.58	12075.16	12084.76	276.48	270.99	79.95	80.53
0.5	256	-	-	-	-	-	-	1090.82	1084.78	324.20	321.89

Tab. M.44.: Realization time of the different methods in 2D for the realization of 200 log-normal distributed fields with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.03	0.36	0.69	0.73	0.28	0.30	1.84	1.33	0.55	0.56
5	16	0.06	-	0.14	0.14	0.11	0.09	4.90	4.51	1.17	1.22
5	32	0.76	-	1.51	1.42	3.57	1.56	18.14	18.28	4.60	4.56
2.5	16	0.09	0.06	0.09	0.06	0.12	0.09	4.73	4.54	1.17	1.22
2.5	32	0.73	-	1.53	1.48	3.48	3.40	18.39	18.21	4.51	4.60
2.5	64	13.15	-	68.38	64.46	198.57	63.31	70.75	70.79	18.49	18.66
0.5	64	12.90	12.71	68.52	68.19	197.28	197.75	70.70	70.72	18.45	18.55
0.5	128	311.16	-	3928.96	3845.58	12079.31	12075.99	281.43	273.64	94.21	95.13
0.5	256	-	-	-	-	-	-	1100.79	1098.04	326.46	328.12

Tab. M.45.: Time to determine the statistical properties of the generated fields for the different methods in 2D. The fields are normal distributed with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.44	0.48	2.22	0.92	0.45	0.53	2.64	1.59	1.79	0.84
5	16	1.39	-	1.62	1.47	1.48	1.47	5.73	5.93	2.48	2.43
5	32	10.08	-	11.26	10.84	16.66	11.26	27.00	27.11	14.98	14.27
2.5	16	1.40	1.51	1.45	1.51	1.36	1.47	5.79	5.99	2.48	2.48
2.5	32	10.23	-	11.01	10.80	13.24	12.18	26.15	27.24	14.18	14.21
2.5	64	89.50	-	144.21	138.06	274.58	116.25	142.54	146.53	96.17	96.47
0.5	64	89.15	199.82	151.12	141.95	279.35	459.34	139.06	146.27	95.96	96.30
0.5	128	922.06	-	4648.56	4508.63	12880.97	12806.00	888.02	899.38	694.42	697.65
0.5	256	-	-	-	-	-	-	6049.31	6129.33	5220.53	5229.26

Tab. M.46.: Time to determine the statistical properties of the generated fields for the different methods in 2D. The fields are normal distributed with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	1.76	0.62	0.45	0.97	0.56	0.58	2.07	1.61	1.89	0.80
5	16	1.48	-	1.39	1.37	1.51	1.34	6.38	5.91	2.48	2.46
5	32	10.11	-	10.90	10.83	12.87	10.37	28.67	26.77	14.20	13.88
2.5	16	1.44	1.40	1.70	14.15	1.37	1.40	6.18	5.87	2.43	2.45
2.5	32	10.00	-	10.73	10.67	12.87	11.39	28.53	26.91	14.82	13.90
2.5	64	89.37	-	141.13	138.87	272.49	107.83	151.51	140.60	95.52	93.80
0.5	64	89.62	90.59	141.54	142.23	271.85	271.85	151.02	142.04	95.32	93.60
0.5	128	926.60	-	4523.31	4336.91	12696.62	8163.81	917.54	874.73	691.80	673.60
0.5	256	-	-	-	-	-	-	5852.78	5857.12	5178.67	5018.41

Tab. M.47.: Time to determine the statistical properties of the generated fields for the different methods in 2D. The fields are lognormal distributed with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.44	0.62	0.98	1.03	0.58	0.56	1.78	1.56	0.84	0.86
5	16	1.51	-	1.44	1.48	1.40	1.40	6.80	5.83	2.54	2.57
5	32	10.31	-	11.06	11.23	12.90	11.17	27.41	26.60	14.18	14.34
2.5	16	1.39	1.37	1.54	1.51	1.37	1.34	6.01	5.77	2.53	2.57
2.5	32	10.12	-	11.09	10.98	12.96	12.78	27.30	26.49	14.23	14.38
2.5	64	88.75	-	143.38	139.12	271.16	152.55	144.99	141.12	95.22	96.58
0.5	64	88.64	88.28	140.99	140.51	273.31	270.65	144.44	140.39	95.18	95.85
0.5	128	915.12	-	4656.58	4582.36	12682.51	12692.15	884.09	865.91	692.88	695.87
0.5	256	-	-	-	-	-	-	5833.55	5833.17	5236.20	5224.33

Tab. M.48.: Time to determine the statistical properties of the generated fields for the different methods in 2D. The fields are log-normal distributed with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.31	0.62	1.00	1.00	0.56	0.56	2.17	1.58	0.80	0.81
5	16	1.42	-	1.45	1.48	1.40	1.47	6.29	5.77	2.40	2.46
5	32	10.75	-	10.75	10.92	13.09	11.06	27.80	27.89	13.62	13.60
2.5	16	1.44	1.36	1.39	1.36	1.40	1.36	6.02	5.83	2.42	2.46
2.5	32	10.73	-	10.94	11.01	12.76	12.76	27.86	27.61	13.59	13.62
2.5	64	91.56	-	141.79	138.84	271.89	137.08	146.44	144.18	89.73	89.73
0.5	64	88.00	88.84	142.19	142.52	271.83	271.80	145.89	144.55	89.48	89.56
0.5	128	927.69	-	4529.89	4448.23	12687.28	12684.44	885.93	876.73	681.51	682.25
0.5	256	-	-	-	-	-	-	5921.64	5908.77	5017.49	5018.05

Tab. M.49.: Mean values of the mean values of the 200 normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0896	-0.0171	0.0135	0.0385	0.0188	-0.0109	0.0058	-0.0355	-0.0188	0.0369
5	16	0.0074	-	-0.0387	0.0378	0.0137	0.0158	-0.0043	-0.0227	-0.0104	-0.0317
5	32	-0.0238	-	-0.0640	-0.0005	0.0274	0.0504	-0.0580	-0.0318	0.0090	-0.0110
2.5	16	-0.0130	0.0104	-0.0085	-0.0067	-0.0456	-0.0089	-0.0096	0.0034	0.0364	-0.0085
2.5	32	0.0113	-	0.0174	0.0135	0.0091	0.0296	-0.0292	-0.0174	0.0049	0.0278
2.5	64	0.0050	-	0.0157	-0.0020	-0.0095	-0.0043	-0.0159	0.0167	-0.0066	-0.0143
0.5	64	-0.0064	0.0033	-0.0011	-0.0030	0.0018	-0.0034	-0.0033	0.0042	0.0074	0.0041
0.5	128	-0.0109	-	-0.0031	-0.0010	0.0030	-0.0057	0.0023	-0.0038	-0.0060	-0.0022
0.5	256	-	-	-	-	-	-	0.0060	-0.0016	-0.0053	0.0044

Tab. M.50.: Mean values of the mean values of the 200 normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0563	-0.0331	0.0142	-0.0003	0.0315	-0.0231	0.0872	0.0377	-0.0637	-0.1446
5	16	-0.0215	-	-0.0371	-0.0306	-0.0109	0.0279	0.0881	-0.0494	0.0896	0.0781
5	32	-0.0111	-	0.0233	0.0612	0.0150	-0.1480	-0.0684	0.0524	0.0015	0.0748
2.5	16	0.0120	-0.0094	0.0784	-0.0056	-0.0555	0.1363	0.1362	-0.0336	0.0024	-0.0425
2.5	32	0.0370	-	0.0598	0.0962	-0.0475	-0.0291	-0.0338	0.0542	-0.0104	0.0420
2.5	64	-0.0389	-	0.0210	0.0443	0.0036	0.0677	0.0157	-0.0719	0.0895	0.0457
0.5	64	0.0086	-0.0792	0.0105	0.0714	-0.0403	-0.0495	-0.0441	-0.0030	-0.0569	0.0221
0.5	128	0.0049	-	0.0087	-0.0334	0.0402	-0.0302	-0.0750	-0.0532	-0.0037	0.0255
0.5	256	-	-	-	-	-	-	0.0952	-0.0663	-0.0052	0.0231

Tab. M.51.: Mean values of the mean values of the 200 log-normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.9905	0.9990	0.9868	1.0018	0.9640	0.9744	0.9669	1.0014	0.9468	0.9746
5	16	0.9928	-	0.9848	1.0199	1.0165	1.0011	0.9904	1.0331	0.9363	0.9838
5	32	1.0016	-	0.9976	1.0300	1.0105	0.9734	1.0097	0.9718	0.9451	1.0248
2.5	16	0.9755	1.0057	1.0255	0.9960	0.9733	1.0136	0.9827	0.9966	0.9358	0.9618
2.5	32	0.9865	-	1.0054	0.9841	0.9921	0.9919	1.0237	0.9829	0.9353	0.9814
2.5	64	0.9815	-	1.0078	1.0072	1.0155	1.0153	0.9860	0.9957	1.0169	0.9915
0.5	64	0.9980	0.9958	0.9977	1.0029	1.0018	1.0006	0.9749	1.0011	0.9011	0.9541
0.5	128	0.9997	-	0.9995	0.9973	0.9974	0.9973	0.9854	0.9972	0.9379	0.9896
0.5	256	-	-	-	-	-	-	0.9932	0.9965	0.9629	1.0043

Tab. M.52.: Mean values of the mean values of the 200 log-normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.9602	0.9879	1.0939	0.9485	0.9441	0.9871	1.0205	0.9812	1.0287	0.9992
5	16	0.9709	-	1.0344	0.9880	1.0398	1.1046	0.9813	0.9183	0.9020	1.0758
5	32	0.9682	-	0.9961	1.0000	1.0040	0.9821	0.9685	0.9829	1.0410	1.0873
2.5	16	0.9376	1.0455	1.0307	1.0126	1.0410	0.9620	1.0179	1.0477	1.0617	0.9919
2.5	32	0.9284	-	0.9799	0.9962	0.9263	1.0194	0.9293	0.9412	1.0010	0.9376
2.5	64	0.9517	-	1.0201	1.0344	1.0187	1.0464	0.9642	1.0187	0.9740	0.9764
0.5	64	0.9262	0.9648	1.0355	0.9295	0.9921	1.0459	0.9879	1.0024	0.9164	0.9470
0.5	128	1.0502	-	1.0232	1.0350	0.9232	0.9385	1.0454	0.9965	0.9707	1.0109
0.5	256	-	-	-	-	-	-	0.9904	0.9848	0.9737	1.0076

Tab. M.53.: Deviation in the mean values of the 200 normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L$_c$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.4659	0.3847	0.4005	0.3878	0.4580	0.3603	0.4322	0.3713	0.4879	0.3909
5	16	0.4441	-	0.4768	0.3934	0.4784	0.4101	0.4354	0.3941	0.4406	0.3879
5	32	0.4954	-	0.4894	0.4411	0.4696	0.4377	0.5117	0.4340	0.4838	0.3863
2.5	16	0.2733	0.2184	0.2743	0.2149	0.2939	0.2558	0.2644	0.2248	0.3009	0.2245
2.5	32	0.2617	-	0.2811	0.2349	0.2835	0.2386	0.3035	0.2166	0.2930	0.2217
2.5	64	0.3144	-	0.3158	0.2469	0.2622	0.2197	0.2904	0.2215	0.2877	0.2353
0.5	64	0.0630	0.0464	0.0633	0.0468	0.0679	0.0507	0.0705	0.0454	0.0737	0.0513
0.5	128	0.0631	-	0.0633	0.0488	0.0706	0.0456	0.0673	0.0475	0.0697	0.0442
0.5	256	-	-	-	-	-	-	0.0674	0.0487	0.0683	0.0523

Tab. M.54.: Deviation in the mean values of the 200 normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.6891	0.7801	0.8651	0.7090	0.8451	0.7177	0.8845	0.7379	0.7752	0.8008
5	16	0.7300	-	0.8180	0.7564	0.7765	0.7397	0.7624	0.7835	0.7993	0.7686
5	32	0.7062	-	0.7884	0.7260	0.7973	0.8198	0.7396	0.7699	0.8082	0.7260
2.5	16	0.7831	0.7201	0.7043	0.7228	0.7353	0.6981	0.7891	0.6910	0.7157	0.7192
2.5	32	0.7378	-	0.7484	0.7706	0.7845	0.6699	0.6847	0.6799	0.7261	0.7424
2.5	64	0.7232	-	0.7131	0.7840	0.7167	0.7155	0.6784	0.7432	0.7653	0.7129
0.5	64	0.7273	0.7010	0.7308	0.6959	0.6590	0.6557	0.6500	0.7170	0.6461	0.7097
0.5	128	0.7334	-	0.7384	0.7762	0.7124	0.7330	0.6787	0.7153	0.6489	0.7395
0.5	256	-	-	-	-	-	-	0.6615	0.6937	0.7192	0.7399

Tab. M.55.: Deviation in the mean values of the 200 log-normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.4015	0.3112	0.3026	0.3310	0.3175	0.2899	0.3758	0.3330	0.3542	0.3652
5	16	0.3443	-	0.3394	0.3836	0.3768	0.3444	0.4039	0.4013	0.4147	0.3487
5	32	0.3827	-	0.3208	0.3577	0.3649	0.3863	0.4385	0.3254	0.3294	0.3547
2.5	16	0.2119	0.2037	0.2294	0.1785	0.2174	0.1956	0.2668	0.1904	0.1979	0.1871
2.5	32	0.2069	-	0.2252	0.1759	0.2055	0.1827	0.2601	0.1783	0.1981	0.2009
2.5	64	0.2332	-	0.2184	0.1793	0.2170	0.1979	0.2670	0.2123	0.2328	0.1880
0.5	64	0.0512	0.0416	0.0521	0.0379	0.0532	0.0438	0.0581	0.0438	0.0492	0.0409
0.5	128	0.0521	-	0.0501	0.0423	0.0491	0.0412	0.0611	0.0444	0.0459	0.0366
0.5	256	-	-	-	-	-	-	0.0600	0.0477	0.0507	0.0421

Tab. M.56.: Deviation in the mean values of the 200 log-normal distributed fields in 2D, with a threshold value of 0.5

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.5574	0.5287	0.7185	0.4997	0.6081	0.6063	0.7170	0.7033	0.7278	0.6266
5	16	0.6176	-	0.7261	0.6266	0.6486	0.8067	0.6796	0.5053	0.5092	0.6715
5	32	0.6228	-	0.5726	0.6456	0.5540	0.7087	0.7785	0.8565	0.7107	0.7677
2.5	16	0.5063	0.5811	0.6843	0.5996	0.6687	0.5332	0.5471	0.6364	0.6768	0.5652
2.5	32	0.5397	-	0.6713	0.6423	0.5232	0.5480	0.6145	0.6939	0.5978	0.5019
2.5	64	0.5760	-	0.6354	0.7310	0.5724	0.6828	0.7087	0.5626	0.5758	0.5407
0.5	64	0.4660	0.5684	0.5636	0.5447	0.5536	0.5973	0.7063	0.6158	0.5956	0.5460
0.5	128	0.6143	-	0.5839	0.6118	0.4877	0.5971	0.7190	0.7288	0.5900	0.6244
0.5	256	-	-	-	-	-	-	0.6530	0.6381	0.5484	0.5552

Tab. M.57.: Mean value of the deviation in the 200 normal distributed fields with in 2D, a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.8766	0.9279	0.8647	0.9061	0.8721	0.8991	0.8463	0.8900	0.7493	0.8416
5	16	0.8993	-	0.8709	0.9100	0.8814	0.9012	0.8543	0.9082	0.7902	0.8634
5	32	0.8749	-	0.8770	0.8889	0.8726	0.8786	0.8591	0.8801	0.8312	0.8670
2.5	16	0.9536	0.9752	0.9397	0.9784	0.9642	0.9776	0.9281	0.9603	0.8238	0.9401
2.5	32	0.9620	-	0.9595	0.9618	0.9630	0.9678	0.9434	0.9777	0.8975	0.9554
2.5	64	0.9479	-	0.9500	0.9750	0.9483	0.9974	0.9378	0.9621	0.9168	0.9675
0.5	64	0.9959	0.9978	0.9942	0.9982	0.9974	1.0008	0.9557	0.9980	0.8675	0.9468
0.5	128	0.9954	-	0.9962	0.9981	0.9973	0.9988	0.9751	0.9970	0.9271	0.9836
0.5	256	-	-	-	-	-	-	0.9877	1.0003	0.9626	0.9994

Tab. M.58.: Mean value of the deviation in the 200 normal distributed fields with in 2D, a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.6143	0.6220	0.6274	0.6428	0.6141	0.6344	0.5893	0.6176	0.5235	0.5989
5	16	0.6228	-	0.6273	0.6341	0.6306	0.6335	0.6129	0.6391	0.5571	0.6250
5	32	0.6124	-	0.6154	0.6437	0.6179	0.6321	0.6028	0.6242	0.5905	0.6327
2.5	16	0.6806	0.6861	0.6702	0.6849	0.6800	0.6859	0.6465	0.6863	0.6003	0.6559
2.5	32	0.6764	-	0.6756	0.6743	0.6691	0.6876	0.6574	0.6856	0.6337	0.6771
2.5	64	0.6735	-	0.6762	0.6858	0.6721	0.6867	0.6687	0.6899	0.6416	0.6784
0.5	64	0.7059	0.7061	0.7060	0.7057	0.7062	0.7050	0.6762	0.7069	0.6129	0.6711
0.5	128	0.7080	-	0.7068	0.7069	0.7056	0.7060	0.6919	0.7047	0.6588	0.6960
0.5	256	-	-	-	-	-	-	0.6973	0.7033	0.6808	0.7052

Tab. M.59.: Mean value of the deviation in the 200 log-normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.8389	0.8378	0.8400	0.8449	0.8259	0.8137	0.7188	0.8152	0.6110	0.7415
5	16	0.8322	-	0.8221	0.8304	0.8847	0.8324	0.8024	0.8133	0.6801	0.7951
5	32	0.8518	-	0.8441	0.8655	0.8342	0.8091	0.8149	0.7739	0.7351	0.8354
2.5	16	0.9115	0.9633	0.9603	0.9468	0.9246	0.9429	0.8617	0.9122	0.7124	0.8604
2.5	32	0.9183	-	0.9542	0.9332	0.9277	0.9366	0.9218	0.9246	0.7780	0.9094
2.5	64	0.9102	-	0.9608	0.9519	0.9421	0.9488	0.9130	0.9247	0.8980	0.9319
0.5	64	0.9968	0.9895	0.9965	1.0029	0.9996	0.9988	0.9133	1.0024	0.7038	0.8681
0.5	128	0.9970	-	0.9927	0.9934	0.9914	0.9916	0.9513	0.9855	0.8210	0.9712
0.5	256	-	-	-	-	-	-	0.9744	0.9894	0.9031	1.0110

Tab. M.60.: Mean value of the deviation in the 200 log-normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	N	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.5728	0.6115	0.6737	0.6058	0.5925	0.6079	0.5387	0.5454	0.5298	0.5840
5	16	0.5763	-	0.6180	0.5837	0.6308	0.6615	0.5156	0.5134	0.4787	0.6652
5	32	0.5753	-	0.5976	0.5931	0.6154	0.5905	0.5195	0.5356	0.5861	0.6566
2.5	16	0.6217	0.7059	0.6889	0.6802	0.6891	0.6517	0.6030	0.6322	0.6052	0.6191
2.5	32	0.6172	-	0.6528	0.6554	0.6212	0.6883	0.5611	0.5790	0.6058	0.6005
2.5	64	0.6324	-	0.6726	0.7013	0.6869	0.7174	0.5671	0.6212	0.6315	0.6550
0.5	64	0.6526	0.6808	0.7265	0.6581	0.7019	0.7398	0.6003	0.6473	0.5360	0.6262
0.5	128	0.7393	-	0.7209	0.7295	0.6477	0.6600	0.6512	0.6387	0.6181	0.7082
0.5	256	-	-	-	-	-	-	0.6260	0.6342	0.6568	0.7148

Tab. M.61.: Standard deviation of the deviations in the 200 normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.1315	0.1786	0.1298	0.1795	0.1337	0.1801	0.1607	0.1779	0.1556	0.1933
5	16	0.1518	-	0.1203	0.1846	0.1243	0.1971	0.1235	0.1851	0.1406	0.1866
5	32	0.1512	-	0.1286	0.1699	0.1303	0.1709	0.1313	0.1840	0.1342	0.1805
2.5	16	0.1025	0.0992	0.0953	0.1157	0.0955	0.1124	0.0982	0.1037	0.1133	0.1296
2.5	32	0.0981	-	0.0970	0.1164	0.0924	0.1075	0.0978	0.1161	0.0987	0.1144
2.5	64	0.1032	-	0.0918	0.1193	0.1006	0.1154	0.0872	0.1134	0.1040	0.1136
0.5	64	0.0250	0.0243	0.0219	0.0231	0.0235	0.0260	0.0278	0.0234	0.0273	0.0221
0.5	128	0.0248	-	0.0244	0.0253	0.0244	0.0251	0.0244	0.0246	0.0239	0.0236
0.5	256	-	-	-	-	-	-	0.0249	0.0239	0.0276	0.0233

Tab. M.62.: Standard deviation of the deviations in the 200 normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0940	0.1108	0.0974	0.1206	0.0887	0.1108	0.0970	0.1172	0.1032	0.1275
5	16	0.0946	-	0.0918	0.1294	0.0908	0.1348	0.1029	0.1414	0.0905	0.1468
5	32	0.0881	-	0.0872	0.1217	0.0909	0.1430	0.0895	0.1412	0.0938	0.1316
2.5	16	0.0687	0.0729	0.0691	0.0765	0.0702	0.0772	0.0631	0.0771	0.0771	0.0794
2.5	32	0.0700	-	0.0729	0.0734	0.0649	0.0842	0.0642	0.0796	0.0796	0.0835
2.5	64	0.0630	-	0.0639	0.0789	0.0694	0.0777	0.0679	0.0779	0.0673	0.0812
0.5	64	0.0176	0.0181	0.0167	0.0169	0.0177	0.0185	0.0184	0.0178	0.0195	0.0160
0.5	128	0.0171	-	0.0181	0.0180	0.0179	0.0190	0.0176	0.0183	0.0170	0.0163
0.5	256	-	-	-	-	-	-	0.0161	0.0170	0.0176	0.0173

Tab. M.63.: Standard deviation of the deviations in the 200 log-normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.5026	0.3724	0.3789	0.4674	0.4166	0.3882	0.3243	0.4106	0.2919	0.3981
5	16	0.3711	-	0.3476	0.4118	0.4163	0.4109	0.4798	0.4140	0.3557	0.4036
5	32	0.4050	-	0.3655	0.4277	0.3517	0.4470	0.4306	0.3863	0.2971	0.4289
2.5	16	0.2589	0.3152	0.2916	0.3365	0.3093	0.2735	0.3115	0.2964	0.2210	0.2940
2.5	32	0.2555	-	0.2860	0.2868	0.2489	0.2679	0.3200	0.2885	0.2347	0.2963
2.5	64	0.2867	-	0.3307	0.2968	0.2528	0.3205	0.3491	0.3693	0.2718	0.2955
0.5	64	0.0769	0.0762	0.0909	0.0859	0.0851	0.0792	0.0856	0.0921	0.0582	0.0648
0.5	128	0.0922	-	0.0693	0.0800	0.0745	0.0813	0.0879	0.0829	0.0597	0.0719
0.5	256	-	-	-	-	-	-	0.0894	0.1058	0.0812	0.0797

Tab. M.64.: Standard deviation of the deviations in the 200 log-normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.3663	0.3618	0.4746	0.3989	0.4089	0.4275	0.4107	0.4600	0.4610	0.4089
5	16	0.4074	-	0.4511	0.4620	0.3988	0.4901	0.3825	0.3320	0.2816	0.5360
5	32	0.3782	-	0.3530	0.4203	0.3482	0.4551	0.4392	0.5427	0.4429	0.5059
2.5	16	0.3478	0.3907	0.5028	0.4347	0.4641	0.3952	0.3468	0.3992	0.4247	0.3718
2.5	32	0.3593	-	0.4782	0.4105	0.3599	0.3885	0.4033	0.4772	0.3782	0.3412
2.5	64	0.4046	-	0.4290	0.4966	0.4052	0.5017	0.4085	0.3777	0.3929	0.3849
0.5	64	0.3310	0.3987	0.3934	0.3871	0.3967	0.4289	0.4288	0.4041	0.3548	0.3658
0.5	128	0.4333	-	0.4113	0.4277	0.3401	0.4213	0.4482	0.4641	0.3744	0.4409
0.5	256	-	-	-	-	-	-	0.4083	0.4108	0.3701	0.4003

Tab. M.65.: Mean error in the correlation structure, estimated over 200 normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0121	0.0267	0.0180	0.0157	0.0200	0.0114	0.0146	0.0067	0.0462	0.0396
5	16	0.0125	-	0.0186	0.0032	0.0103	0.0108	0.0166	0.0255	0.0181	0.0127
5	32	0.0149	-	0.0104	0.0123	0.0217	0.0072	0.0159	0.0108	0.0387	0.0316
2.5	16	0.0056	0.0106	0.0048	0.0152	0.0100	0.0119	0.0111	0.0093	0.0390	0.0121
2.5	32	0.0135	-	0.0080	0.0121	0.0125	0.0200	0.0223	0.0180	0.0193	0.0181
2.5	64	0.0141	-	0.0102	0.0055	0.0194	0.0134	0.0105	0.0091	0.0118	0.0162
0.5	64	0.0040	0.0030	0.0038	0.0037	0.0034	0.0036	0.0051	0.0028	0.0103	0.0061
0.5	128	0.0043	-	0.0026	0.0025	0.0031	0.0025	0.0033	0.0036	0.0077	0.0069
0.5	256	-	-	-	-	-	-	0.0034	0.0032	0.0054	0.0067

Tab. M.66.: Mean error in the correlation structure, estimated over 200 normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
L_c	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0281	0.0149	0.0398	0.0267	0.0398	0.0127	0.0651	0.0207	0.0492	0.0368
5	16	0.0355	-	0.0143	0.0026	0.0025	0.0194	0.0061	0.0174	0.0409	0.0065
5	32	0.0290	-	0.0096	0.0272	0.0125	0.0214	0.0099	0.0162	0.0353	0.0237
2.5	16	0.0269	0.0060	0.0215	0.0016	0.0046	0.0153	0.0557	0.0256	0.0396	0.0196
2.5	32	0.0077	-	0.0053	0.0262	0.0338	0.0353	0.0305	0.0209	0.0180	0.0131
2.5	64	0.0149	-	0.0262	0.0334	0.0252	0.0101	0.0347	0.0103	0.0378	0.0100
0.5	64	0.0129	0.0040	0.0151	0.0096	0.0357	0.0367	0.0199	0.0046	0.0226	0.0263
0.5	128	0.0144	-	0.0211	0.0428	0.0021	0.0171	0.0128	0.0064	0.0110	0.0268
0.5	256	-	-	-	-	-	-	0.0286	0.0092	0.0239	0.0217

Tab. M.67.: Mean error in the correlation structure, estimated over 200 log-normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0754	0.0473	0.0979	0.0462	0.0913	0.0637	0.0335	0.0448	0.0259	0.0576
5	16	0.0857	-	0.0983	0.0578	0.1085	0.0702	0.0442	0.0307	0.0251	0.0744
5	32	0.0806	-	0.1253	0.0697	0.0832	0.0405	0.0462	0.0655	0.0712	0.0488
2.5	16	0.0461	0.0309	0.0480	0.0414	0.0433	0.0387	0.0176	0.0205	0.0352	0.0308
2.5	32	0.0503	-	0.0476	0.0288	0.0557	0.0301	0.0208	0.0231	0.0322	0.0282
2.5	64	0.0480	-	0.0486	0.0335	0.0537	0.0266	0.0238	0.0277	0.0446	0.0350
0.5	64	0.0098	0.0076	0.0099	0.0082	0.0102	0.0076	0.0046	0.0045	0.0074	0.0075
0.5	128	0.0113	-	0.0118	0.0081	0.0114	0.0077	0.0063	0.0053	0.0089	0.0092
0.5	256	-	-	-	-	-	-	0.0066	0.0065	0.0118	0.0106

Tab. M.68.: Mean error in the correlation structure, estimated over 200 log-normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.1491	0.1694	0.1263	0.1929	0.1489	0.1373	0.0589	0.0632	0.0748	0.1485
5	16	0.1544	-	0.1112	0.1540	0.1700	0.0730	0.0617	0.1539	0.1124	0.1582
5	32	0.1413	-	0.1810	0.1245	0.2222	0.0950	0.0618	0.0501	0.1282	0.1393
2.5	16	0.1964	0.1650	0.1437	0.1506	0.1440	0.1710	0.1274	0.0890	0.0709	0.1405
2.5	32	0.1717	-	0.1126	0.1257	0.1736	0.1949	0.0622	0.0709	0.1174	0.1647
2.5	64	0.1568	-	0.1433	0.0984	0.1735	0.1469	0.0460	0.1252	0.1655	0.1844
0.5	64	0.2080	0.1561	0.1799	0.1636	0.1778	0.1692	0.0210	0.1048	0.0427	0.1371
0.5	128	0.1608	-	0.1683	0.1534	0.1923	0.1335	0.0504	0.0395	0.1039	0.1427
0.5	256	-	-	-	-	-	-	0.0671	0.0818	0.1569	0.1844

Tab. M.69.: Standard deviation of the errors in the correlation structure, estimated over 200 normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0076	0.0198	0.0131	0.0128	0.0126	0.0145	0.0132	0.0042	0.0338	0.0269
5	16	0.0087	-	0.0124	0.0029	0.0075	0.0079	0.0077	0.0131	0.0227	0.0071
5	32	0.0077	-	0.0086	0.0107	0.0098	0.0070	0.0068	0.0110	0.0132	0.0217
2.5	16	0.0048	0.0147	0.0036	0.0100	0.0070	0.0074	0.0105	0.0061	0.0394	0.0117
2.5	32	0.0089	-	0.0047	0.0109	0.0101	0.0145	0.0153	0.0165	0.0187	0.0139
2.5	64	0.0102	-	0.0042	0.0041	0.0086	0.0087	0.0077	0.0054	0.0087	0.0125
0.5	64	0.0032	0.0024	0.0050	0.0041	0.0032	0.0031	0.0071	0.0022	0.0146	0.0100
0.5	128	0.0036	-	0.0027	0.0028	0.0021	0.0026	0.0046	0.0035	0.0083	0.0111
0.5	256	-	-	-	-	-	-	0.0030	0.0032	0.0053	0.0125

Tab. M.70.: Standard deviation of the errors in the correlation structure, estimated over 200 normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0146	0.0107	0.0188	0.0173	0.0212	0.0074	0.0283	0.0213	0.0229	0.0307
5	16	0.0162	-	0.0061	0.0017	0.0026	0.0121	0.0036	0.0115	0.0118	0.0047
5	32	0.0082	-	0.0153	0.0149	0.0104	0.0100	0.0044	0.0122	0.0087	0.0131
2.5	16	0.0097	0.0064	0.0092	0.0013	0.0039	0.0107	0.0189	0.0217	0.0207	0.0129
2.5	32	0.0098	-	0.0031	0.0094	0.0133	0.0130	0.0181	0.0129	0.0129	0.0085
2.5	64	0.0063	-	0.0119	0.0125	0.0199	0.0061	0.0117	0.0139	0.0093	0.0089
0.5	64	0.0030	0.0018	0.0028	0.0032	0.0058	0.0068	0.0049	0.0021	0.0079	0.0062
0.5	128	0.0026	-	0.0046	0.0081	0.0017	0.0032	0.0036	0.0016	0.0053	0.0058
0.5	256	-	-	-	-	-	-	0.0054	0.0039	0.0051	0.0056

Tab. M.71.: Standard deviation of the errors in the correlation structure, estimated over 200 log-normal distributed fields in 2D, with a threshold value of 0.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0632	0.0476	0.0735	0.0510	0.0730	0.0505	0.0192	0.0436	0.0214	0.0505
5	16	0.0565	-	0.0535	0.0352	0.0412	0.0572	0.0242	0.0186	0.0256	0.0430
5	32	0.0583	-	0.0417	0.0590	0.0590	0.0421	0.0281	0.0304	0.0346	0.0503
2.5	16	0.0507	0.0453	0.0552	0.0467	0.0523	0.0428	0.0106	0.0231	0.0241	0.0388
2.5	32	0.0576	-	0.0512	0.0454	0.0537	0.0435	0.0246	0.0228	0.0279	0.0383
2.5	64	0.0490	-	0.0450	0.0486	0.0561	0.0418	0.0122	0.0247	0.0390	0.0503
0.5	64	0.0294	0.0240	0.0293	0.0239	0.0292	0.0238	0.0075	0.0120	0.0094	0.0273
0.5	128	0.0298	-	0.0299	0.0234	0.0298	0.0237	0.0112	0.0113	0.0192	0.0316
0.5	256	-	-	-	-	-	-	0.0133	0.0127	0.0266	0.0338

Tab. M.72.: Standard deviation of the errors in the correlation structure, estimated over 200 log-normal distributed fields in 2D, with a threshold value of 0.5.

		CMD						**FFT**		**LAS**	
		Cholesky		*Eigen*		*Modified Eigen*				*Gauss*	
$\mathbf{L_c}$	**N**	Exp	SExp	Exp	SExp	Exp	SEXP	Exp	SEXP	Exp	SExp
5	8	0.0633	0.0799	0.0541	0.0913	0.0649	0.0657	0.0286	0.0347	0.0420	0.0685
5	16	0.0489	-	0.0369	0.0623	0.0538	0.0329	0.0287	0.0678	0.0444	0.0694
5	32	0.0355	-	0.0488	0.0455	0.0671	0.0383	0.0199	0.0390	0.0479	0.0689
2.5	16	0.0574	0.0523	0.0421	0.0492	0.0422	0.0550	0.0433	0.0295	0.0388	0.0487
2.5	32	0.0406	-	0.0265	0.0384	0.0468	0.0555	0.0222	0.0384	0.0340	0.0462
2.5	64	0.0316	-	0.0268	0.0247	0.0354	0.0396	0.0192	0.0413	0.0406	0.0455
0.5	64	0.0276	0.0220	0.0238	0.0236	0.0238	0.0238	0.0063	0.0160	0.0110	0.0186
0.5	128	0.0165	-	0.0173	0.0193	0.0203	0.0168	0.0116	0.0064	0.0149	0.0165
0.5	256	-	-	-	-	-	-	0.0085	0.0123	0.0180	0.0198