Multivariate Geostatistical Grid-Free Simulation of Natural Phenomena

by

Yevgeniy Zagayevskiy

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mining Engineering

Department of Civil and Environmental Engineering

University of Alberta

Abstract

Geostatistical modeling of natural phenomena is an essential step in the development petroleum reservoirs and mineral deposits. This numerical modeling involves simulation of the geological attributes conditional to available relevant data sampled at various scales. The simulation is usually performed at a point scale on a grid of regularly spaced nodes. The simulated results are non-reproducible and order dependent when another simulation is performed on a closer spacing in some areas or for an expanded study area.

A grid-free geostatistical simulation (GFS) method is proposed and developed in this thesis, where the properties of natural phenomena are modeled as a function of the coordinates of the simulation locations. The resulting realizations are conditioned to the data values, preserve the spatial structure of the modeled system, and the relationship between system's variables. Simulation is performed at the point scale and can be upscaled to larger volumes to establish block-scale realizations. The conditioning data can be expressed as a set of scattered point-scale data values or a set of regularly sampled block-scale data values. Gridded block-scale data are converted to pseudo point-scale values using a point-scale block value representation technique to avoid artifacts in the simulated values.

The grid-free simulation proceeds as follows. The 1-D unconditional stochastic processes are generated in a grid-free manner with a proposed Fourier series simulation (FSS), where the target covariance function is decomposed with a Fourier series, and simulation is reconstructed with linear model of regionalization as a weighted sum of the Fourier coefficients and periodic stochastic cosine functions expressed as a function of the coordinates and a random phase. 2-D and 3-D unconditional stochastic processes are simulated grid-free using the modified turning bands concept, where a set of 1-D line processes with covariance functions related to the target 2-D/3-D covariance function is linearly combined to obtain realizations in the higher dimensional space. In the modified turning bands method, the bands are replaced with the points and line process simulation is carried out with the FSS as a function the projected simulation location on a line. The anisotropy is addressed by affine transformation of the simulation space. The conditioning is performed using kriging in a dual form to reduce computational time. Modeling of multivariate systems is possible with the linear model of coregionalization. The weighted random factors are summed to obtain valid multivariate realizations. The assimilation of the secondary gridded data is performed with intrinsic cokriging, where the secondary data at a simulation location and all primary data locations are used in the conditioning. When the secondary data do not cover the entire simulation domain, the secondary data is projected to the simulation location and weighted appropriately to avoid edge

artifacts. The simulation of the nugget effect component is expressed as a function of the coordinates of the gridded space at the lowest possible refinement level.

The grid-free simulation methodology is implemented in Fortran code. Applicability and efficacy of the proposed method are shown with numerous 2-D and 3-D synthetic examples and real case study based on the Firebag oil sands project located in northern Alberta, Canada. Implementation aspects are documented and optimal simulation parameters are defined. Computational time is reduced through the adoption of the turning bands simulation of higher dimensional systems, conditioning with the dual cokriging, intrinsic data assimilation, and block matrix inversion in presence of the exhaustively sampled gridded data at the data assimilation step.

"If you only read the books that everyone else is reading, you can only think what everyone else is thinking."

Haruki Murakami, *Norwegian Wood*

In memory of my grandfather

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Clayton V. Deutsch, the professor and director of the Centre for Computational Geostatistics (CCG), for his continuous stream of challenging engineering problems and non-trivial research ideas, encouragement, motivation, and guidance. This helped me overcome all graduate student obstacles smoothly and produce this valuable piece of research.

It is a pleasure to have Alice Da Silva in our research group, who would shield me from the university paper work and let me work in a stress-free research environment. The heated discussions with my great colleagues Diogo Silva, Jared Deutsch, Ryan Barnett, and Saina Lajevardi expanded my geostatistical knowledge beyond the classroom. It was also great to be pushed beyond my research comfort zone almost on daily basis by junior members of our group Jianan Qu, Cole Mooney, and Yaroslav Vasilchuk. I am happy to establish long working relationships with John Manchuk, Dan Khan, Daniel Silva, Miguel Cuba, and Dawib Fenik. Thank you to all my colleagues for support.

Also, I would like to thank all examining committee members and especially Dr. Denis Marcotte for their valuable comments and thoughtful contributions to the final version of my thesis. By following their guidelines, I have considerably improved the quality of the work.

The financial support and study-oriented environment provided by the sponsoring companies of our research group and the University of Alberta are highly appreciated. I was motivated by the University of Alberta resources: modern libraries, abundant foreign language classes, various sport clubs and social events, just to name few, which harmoniously supplemented the productivity of my research work. The accommodating academic atmosphere has provided me with an opportunity to grow as an individual.

I am thankful to all my friends, younger and elder, with whom I have established warm relationship through these three plus years at the university. I still stay in touch with many of them. It is a great experience to have friends from all over the world.

Last, but not the least, I thank my loving family for letting me be childish for a longer time. My decision freedom allowed me rambling on a foreign soil for a search of myself. Now I can proudly call Canada my second home. Most of my questions have been answered, but even more have risen during my journey.

This thesis of mine has been stipulated by industrial needs and, hopefully will find application in commercial modeling.

Contents

List of Tables

List of Figures

List of Symbols

$\boldsymbol{A}_p$	weighting matrix of the independent random factors $\boldsymbol{X}_p$
A	modeling domain
$A(\boldsymbol{u}, \boldsymbol{\omega}_m)$	anisotropy function in the spectral simulation
a_p	weight of the independent random factor $X_p(\boldsymbol{u})$
$a_{p,kl}$	weight element in the weighting matrix $\boldsymbol{A}_p$
$a_{\alpha'}$	intercept of the local regression model at point α'
$\boldsymbol{B}_p$	sill contribution matrix of the independent random factor $\boldsymbol{X}_p$
b_p	sill contribution of independent random factor X_p
$b_{p,kl}$	element of the sill contribution matrix $\boldsymbol{B}_p$
$b_{alpha'}$	slope of the local regression model at anchor point α'
$\boldsymbol{C}$	1. model covariance matrix
	2. data covariance matrix
$\boldsymbol{C}_Y$	covariance matrix of the modeled system in normal scores
$\boldsymbol{C}_{Y_F Y_F}$	covariance matrix for fixed data
$\boldsymbol{C}_{Y_F Y_V}$	covariance matrix for fixed – varying data
$\boldsymbol{C}_{Y_V Y_V}$	covariance matrix for varying data
$\boldsymbol{C}_{Y_V Y_F}$	covariance matrix for varying – fixed data
$\boldsymbol{C}_Z$	covariance matrix of the modeled system in original units
$C_k(\boldsymbol{h})$	stationary covariance function of the k^{th} category
$C_{X^{(1)},2-D}(h)$	line process covariance function for simulation 2-D systems

$C_{X^{(1)},3-D}(h)$	line process covariance function for simulation 3-D systems
$C_Z(\boldsymbol{h})$	stationary covariance function of the random function Z
$C_Z(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{h})$	covariance function of the random function Z
$C_{Z_k Z_l}(\boldsymbol{h})$	stationary covariance function of the functions Z_k and Z_l
$\hat{C}_Y(\boldsymbol{\omega})$	covariance of random function Y in the frequency domain
$\boldsymbol{c}$	data-estimate covariance matrix
$det\{\}$	determinant operator
$\boldsymbol{e}$	eigenvector
$E\{\}$	expected value operator
$erf()$	error function
$Exp(\boldsymbol{h})$	exponential semivariogram model or covariance function
$\boldsymbol{F}$	matrix with cumulative distribution function values
$F_{Z(\boldsymbol{u})}(z)$	cumulative distribution function of random variable $Z(\boldsymbol{u})$
$\hat{F}_{Z(\boldsymbol{u})}(z)$	approximated cumulative distribution function of $Z(\boldsymbol{u})$
f	function
f_s	coefficient in the point-scale block value interpolation
$f(h)$	periodic continuous function with argument h
$f_{Z(\boldsymbol{u})}(z)$	probability density function of random variable $Z(\boldsymbol{u})$
f^*	reconstructed function
$Gaus(\boldsymbol{h})$	Gaussian semivariogram model or covariance function
GR_{max}	maximum gamma ray measurement along a well bore
GR_{min}	minimum gamma ray measurement along a well bore
$G_{Y(\boldsymbol{u})}(y)$	cumulative probability function of random variable $Y(\boldsymbol{u})$
g_p	p^{th} cosine Fourier coefficient
g'_p	p^{th} sine Fourier coefficient
$g_{Y(\boldsymbol{u})}(y)$	probability density function of random variable $Y(\boldsymbol{u})$
H	Hurst exponent
$\boldsymbol{h}$ and $\boldsymbol{h}'$	separation lag vector

h	separation lag distance in 1-D space
h_X	projection of lag vector $\boldsymbol{h}$ on X axis
h_Y	projection of lag vector $\boldsymbol{h}$ on Y axis
h_Z	projection of lag vector $\boldsymbol{h}$ on Z axis
h_1	projection of $\boldsymbol{h}$ on first principal direction of continuity
h_2	projection of $\boldsymbol{h}$ on second principal direction of continuity
h_3	projection of $\boldsymbol{h}$ on third principal direction of continuity
Δh	distance increment
$\boldsymbol{I}$	identity matrix with zero elements except unit diagonal
i	1. realization index
	2. grid block index in first orthogonal direction
	3. imaginary unit of the complex number
$i_k(\boldsymbol{u})$	indicator of k^{th} category at location $\boldsymbol{u}$
$i_k^s(\boldsymbol{u})$	simulated indicator of k^{th} category at location $\boldsymbol{u}$
$i_k^*(\boldsymbol{u})$	estimated indicator of k^{th} category at location $\boldsymbol{u}$
$i_k'^*(\boldsymbol{u})$	corrected estimated indicator of k^{th} category at location $\boldsymbol{u}$
$int\{\}$	integer operator
j	1. location index
	2. grid block index in second orthogonal direction
	3. white noise location index
$\boldsymbol{K}_{\alpha'}$	matrix with kernel values for local linear regression model
K	1. number of multivariate random functions
	2. number of categories
K_E	number of random functions sampled exhaustively
K_S	number of random functions sampled at scatter
K_b	elastic bulk modulus in GPa
$K_{b,clay}$	elastic bulk modulus of clay mineral in GPa
$K_{b,dry}$	elastic bulk modulus of dry rock in GPa

$K_{b,fluid}$	elastic bulk modulus of fluid in GPa
$K_{b,matrix}$	elastic bulk modulus of rock matrix in GPa
$K_{b,oil}$	elastic bulk modulus of oil in GPa
$K_{b,quartz}$	elastic bulk modulus of quartz mineral in GPa
$K_{b,sat}$	elastic bulk modulus of saturated rock in GPa
$K_{b,water}$	elastic bulk modulus of water in GPa
K_s	elastic shear modulus in GPa
$K_{s,clay}$	elastic shear modulus of clay mineral in GPa
$K_{s,dry}$	elastic shear modulus of dry rock in GPa
$K_{s,matrix}$	elastic shear modulus of rock matrix in GPa
$K_{s,quartz}$	elastic shear modulus of quartz mineral in GPa
$K_{s,sat}$	elastic shear modulus of saturated rock in GPa
$K_\omega(\alpha', \boldsymbol{u}_\alpha)$	kernel function for local linear regression model
k	1. random function index in the multivariate system
	2. permeability in mD
	3. grid block index in third orthogonal direction
$\boldsymbol{L}$	lower triangular matrix
$\boldsymbol{L}_Y$	lower triangular matrix of the covariance matrix $\boldsymbol{C}_Y$
L	number of the turning lines processes
l	1. random function index in the multivariate system
	2. turning line index
$\boldsymbol{M}$	number of the discretization nodes in spectral simulation
$\boldsymbol{M}_R$	coordinate rotation matrix around orthogonal axes
$\boldsymbol{M}_{R_X}$	coordinate rotation matrix around X axis
$\boldsymbol{M}_{R_Y}$	coordinate rotation matrix around Y axis
$\boldsymbol{M}_{R_Z}$	coordinate rotation matrix around Z axis
$\boldsymbol{M}_S$	coordinate stretching matrix
M	1. number of the simulation nodes

	2. number of discretization nodes in spectral simulation
	3. number of nodes in Fourier series decomposition
m	1. index used in Fourier coefficients
	2. random function index in the multivariate system
$m_{X_p}(\boldsymbol{u})$	local mean of independent factor $X_p(\boldsymbol{u})$ at location $\boldsymbol{u}$
$m_{X_{p,k}}(\boldsymbol{u})$	local mean of factor $X_p(\boldsymbol{u})$ related to Z_k
$m_Y(\boldsymbol{u})$	local mean of random variable $Y(\boldsymbol{u})$ in normal scors
m_Z	stationary mean of random function Z
$m_Z(\boldsymbol{u})$	local mean of random variable $Z(\boldsymbol{u})$ at location $\boldsymbol{u}$
m_{Z_k}	stationary mean of k^{th} random function Z_k
$m_{\tilde{Z}}(\boldsymbol{u})$	local mean of non-stationary random variable $\tilde{Z}(\boldsymbol{u})$
N	number of the data
N_A	number of anchor points on the CDF
N_E	number of data of random function sampled exhaustively
N_R	number of realizations
N_S	number of data of random function sampled at scatter
N_X	number of discretization points in X direction
N_Y	number of discretization points in Y direction
N_Z	number of discretization points in Z direction
N_k	number of the data of the k^{th} random function
N_l	number of the data of the l^{th} random function
N_p	number of the data pairs
$NS(\boldsymbol{u})$	normal score value at location $\boldsymbol{u}$
$Nugget(\boldsymbol{h})$	nugget effect semivariogram model or covariance function
$N(0,1)$	normal distribution with zero mean and unit variance
ΔN_X	number of discretization points in X direction
ΔN_Y	number of discretization points in Y direction
ΔN_Z	number of discretization points in Z direction

$\boldsymbol{0}$	zero matrix
$\boldsymbol{P}$	vector with local linear regression coefficients
$P+1$	number of the independent random factors $X_p(\boldsymbol{u})$
$Prob\{\}$	probability
p	index of the independent random factor $X_p(\boldsymbol{u})$
$p_k(\boldsymbol{u})$	probability of k^{th} category to be present at location $\boldsymbol{u}$
$2Q+1$	number of the Fourier series terms
$2Q_1+1$	number of Fourier series terms in first orthogonal direction
$2Q_2+1$	number of Fourier series terms in second orthogonal direction
$2Q_3+1$	number of Fourier series terms in third orthogonal direction
q	1. index of the random factor $X_q(\boldsymbol{u})$
	2. index of a Fourier series term
$\boldsymbol{R}$	vector with $N(0,1)$ values
R	random function with $N(0,1)$ distribution
$R(\boldsymbol{u})$	random variable with $N(0,1)$ distribution at location $\boldsymbol{u}$
$\boldsymbol{r}$	variogram/correlation range vector
r	variogram range of isotropic random function
r_1	projection of $\boldsymbol{r}$ on first principal direction of continuity
r_2	projection of $\boldsymbol{r}$ on second principal direction of continuity
r_3	projection of $\boldsymbol{r}$ on third principal direction of continuity
$2S$	periodic domain of the Fourier series
S_1	size of the data block in first orthogonal direction
S_2	size of the data block in second orthogonal direction
S_3	size of the data block in third orthogonal direction
$Sph(\boldsymbol{h})$	spherical semivariogram model or covariance function
s	1. iteration index
	2. water saturation
$\boldsymbol{U}$	upper triangular matrix

$\boldsymbol{U}_Y$	upper triangular matrix of covariance matrix $\boldsymbol{C}_Y$
U	uniformly distributed random function
$U(\boldsymbol{u})$	uniformly distributed random variable at location $\boldsymbol{u}$
$\boldsymbol{u}$ and $\boldsymbol{u}'$	location vector
$\boldsymbol{u}_\alpha$	data location vector
u	location coordinate
u_1	coordinate on first principal direction of continuity
u_2	coordinate on second principal direction of continuity
u_3	coordinate on third principal direction of continuity
$\hat{\boldsymbol{u}}$	unit vector
V	1. volume in space
	2. block scale
$V(\boldsymbol{u})$	volume in space centered at location $\boldsymbol{u}$
V-shale or V_{shale}	volume fraction of shale mineral
$\boldsymbol{X}_p(\boldsymbol{u})$	vector with K independent random factors of index p
X	one of orthogonal axes of Cartesian coordinate system
$X_l^{(1)}$	l^{th} stochastic line process
$X_p(\boldsymbol{u})$	p^{th} independent random factor at location $\boldsymbol{u}$
$X_{p,k}(\boldsymbol{u})$	p^{th} random factor for k^{th} random function
x	coordinate on X axis
$x(\boldsymbol{u})$	coordinate of location $\boldsymbol{u}$ on X axis
$\boldsymbol{Y}$	matrix of random variables in normal scores
$\boldsymbol{Y}^d$	vector of data values in normal scores
Y	one of orthogonal axes of Cartesian coordinate system
Y	random function in normal scores
Y'	random function without nugget effect in normal scores
$Y(\boldsymbol{u})$	point-scale random variable at location $\boldsymbol{u}$ in normal scores
$Y_k(\boldsymbol{u})$	k^{th} random variable of multivariate system in normal scores

$Y(\boldsymbol{u}_\alpha)$	data of random variable Y at location $\boldsymbol{u}_\alpha$ in normal space
$Y^s(\boldsymbol{u})$	conditionally simulated random variable $Y(\boldsymbol{u})$
$Y_i^s(\boldsymbol{u})$	i^{th} realization of conditionally simulated variable $Y(\boldsymbol{u})$
$Y^{us}(\boldsymbol{u})$	unconditionally simulated random variable $Y(\boldsymbol{u})$
$Y_i^{us}(\boldsymbol{u})$	i^{th} realization of unconditionally simulated variable $Y(\boldsymbol{u})$
$Y^*(\boldsymbol{u})$	estimated random variable $Y(\boldsymbol{u})$ at location $\boldsymbol{u}$
$Y^{*\|us}(\boldsymbol{u})$	estimated random variable $Y(\boldsymbol{u})$ conditioned to simulation
$Y_i^{*\|us}(\boldsymbol{u})$	i^{th} realization of estimated $Y(\boldsymbol{u})$ conditioned to simulation
$\Delta Y^*(\boldsymbol{u})$	$Y(\boldsymbol{u})$ estimate difference conditioned to data and simulation
y	1. value of random function Y
	2. value random variable $Y(\boldsymbol{u})$
	3. coordinate on Y axis
$y(\boldsymbol{u})$	coordinate of location $\boldsymbol{u}$ on Y axis
$\boldsymbol{Z}$	vector/matrix of random variables in original units
$\boldsymbol{Z}^d$	vector of data values of random function Z in original units
Z	one of orthogonal axes of Cartesian coordinate system
Z	random function in original units
Z_k	k^{th} random function of multivariate system in original units
Z_l	l^{th} random function of multivariate system in original units
$Z(\boldsymbol{u})$	1. point-scale random variable in original units
	2. simulated value of random functions Z at location $\boldsymbol{u}$
$Z_k(\boldsymbol{u})$	k^{th} random variable of multivariate system at location $\boldsymbol{u}$
$Z(\boldsymbol{u}_\alpha)$	1. random variable Z at data location $\boldsymbol{u}_\alpha$
	2. data value of random function Z at location $\boldsymbol{u}_\alpha$
$Z(V(\boldsymbol{u}))$	block-scale random variable centered at location $\boldsymbol{u}$
$Z^s(\boldsymbol{u})$	conditionally simulated value of random variable $Z(\boldsymbol{u})$
$Z^{us}(\boldsymbol{u})$	unconditionally simulated value of random variable $Z(\boldsymbol{u})$
$Z^*(\boldsymbol{u})$	estimated random variable $Z(\boldsymbol{u})$ at location $\boldsymbol{u}$ in original units

$Z_k^*(\boldsymbol{u})$	estimated k^{th} variable $Z_k(\boldsymbol{u})$ of the multivariate system
$\bar{Z}$	average of random function in original units
$\tilde{Z}$	non-stationary random function in original units
z	1. value of random function Z
	2. value of random variable $Z(\boldsymbol{u})$
	3. coordinate on Z axis
z'	adjusted to the reservoir top elevation coordinate z
$z(\boldsymbol{u})$	coordinate of location $\boldsymbol{u}$ on the Z axis
$z_{top}(x, y)$	reservoir top elevation of location $\boldsymbol{u}(x, y)$
$\hat{z}(\boldsymbol{u}_{\alpha'})$	fitted data at anchor point α'
α	data location index
α'	anchor location index for local linear regression
α_a	azimuth rotation angle
α_d	dip rotation angle
α_p	plunge rotation angle
α_X	rotation angle of maximum continuity direction around X
α_Y	rotation angle of maximum continuity direction around Y
α_Z	rotation angle of maximum continuity direction around Z
β	estimation or simulation location index
$\Gamma(\boldsymbol{h})$	fundamental stationary semivariogram model
$\Gamma^0(\boldsymbol{h})$	nugget effect semivariogram model
$\Gamma^1(\boldsymbol{h})$	spherical semivariogram model
$\Gamma^2(\boldsymbol{h})$	exponential semivariogram model
$\Gamma^3(\boldsymbol{h})$	Gaussian semivariogram model
$\gamma_Z(\boldsymbol{h})$	stationary semivariogram of random function Z
$\gamma_Z(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{h})$	semivariogram of random function Z
$\gamma_{Z_k Z_l}(\boldsymbol{h})$	stationary cross-semivariogram of functions Z_k and Z_l

$\hat{\gamma}_Z(\boldsymbol{h})$	experimental semivariogram of random function Z
$\hat{\gamma}_{Z_k Z_l}(\boldsymbol{h})$	experimental cross-semivariogram of functions Z_k and Z_l
Δ_X	grid resolution in X direction
Δ_Y	grid resolution in Y direction
Δ_Z	grid resolution in Z direction
ε	estimation error random function
$\varepsilon(\boldsymbol{u})$	estimation error random variable at location $\boldsymbol{u}$
ϵ	mean square error
θ	additional phase shift attributed to multivariate systems
θ_l	directional angle for l^{th} line process for 2-D systems
$\theta_{1,l}$	first directional angle for l^{th} line process for 3-D systems
$\theta_{2,l}$	second directional angle for l^{th} line process for 3-D systems
κ	scaling parameter
$\boldsymbol{\lambda}$	vector with kriging weights
$\lambda(\boldsymbol{u})$	weighting function
$\lambda_\alpha(\boldsymbol{u})$	kriging weight of datum at $\boldsymbol{u}_\alpha$ for estimation at location $\boldsymbol{u}$
$\lambda_{k,\alpha}(\boldsymbol{u})$	kriging weight of datum of k at $\boldsymbol{u}_\alpha$ to estimate i_k at $\boldsymbol{u}$
$\lambda_{k,\alpha_l}(\boldsymbol{u})$	kriging weight of datum of Z_l at $\boldsymbol{u}_\alpha$ to estimate Z_k at $\boldsymbol{u}$
$\hat{\lambda}(\boldsymbol{\omega})$	weighting function in frequency domain
$\breve{\lambda}(\boldsymbol{u})$	reverse of weighting function
μ	dimensional size of space
$\boldsymbol{\nu}$	vector with dual kriging weights
$\nu_\alpha(\boldsymbol{u})$	dual kriging weight of datum at $\boldsymbol{u}_\alpha$ to estimate at $\boldsymbol{u}$
ρ	density
ρ_{clay}	density of clay mineral
ρ_{fluid}	density of fluid
ρ_{matrix}	density of rock matrix
ρ_{oil}	density of oil mineral

ρ_{quartz}	density of quartz mineral
ρ_{sat}	density of saturated rock
ρ_{water}	density of water mineral
$\rho_{Z_k Z_l}(\boldsymbol{h})$	correlation coefficient of random functions Z_k and Z_l
$\sigma_Y(\boldsymbol{u})$	local standard deviation of random variable $Y(\boldsymbol{u})$
$\sigma_Z(\boldsymbol{u})$	local standard deviation of random variable $Z(\boldsymbol{u})$
$\sigma_{Z_k Z_l}(\boldsymbol{u})$	standard deviation of random variables $Z_k(\boldsymbol{u})$ and $Z_l(\boldsymbol{u})$
$\sigma_k^2(\boldsymbol{u})$	local variance of k^{th} category at location $\boldsymbol{u}$
$\sigma_Y^2(\boldsymbol{u})$	local variance of random variable $Y(\boldsymbol{u})$
$\sigma_Z^2(\boldsymbol{u})$	local variance of random variable $Z(\boldsymbol{u})$
σ_Z^2	stationary variance of random function Z
$\sigma_{Z_k Z_l}^2(\boldsymbol{u})$	covariance between random variables $Z_k(\boldsymbol{u})$ and $Z_l(\boldsymbol{u})$
$\sigma_{Z_k Z_l}^2$	stationary covariance between $Z_k(\boldsymbol{u})$ and $Z_l(\boldsymbol{u})$
$\sigma_\varepsilon^2(\boldsymbol{u})$	estimation error variance at location $\boldsymbol{u}$
$\sigma_{\varepsilon^*}^2(\boldsymbol{u})$	kriging variance at location $\boldsymbol{u}$ of random variable $Z(\boldsymbol{u})$
$\sigma_{\varepsilon^*,k}^2(\boldsymbol{u})$	kriging variance at location $\boldsymbol{u}$ of random variable $Z_k(\boldsymbol{u})$
τ	eigenvalue
υ	weight of data on cumulative distribution function
$\Phi(\boldsymbol{h})$	fundamental stationary covariance function model
$\Phi^0(\boldsymbol{h})$	nugget effect covariance function model
$\Phi^1(\boldsymbol{h})$	spherical covariance function model
$\Phi^2(\boldsymbol{h})$	exponential covariance function model
$\Phi^3(\boldsymbol{h})$	Gaussian covariance function model
ϕ	porosity
ϕ'	corrected porosity
ϕ_q	random phase of Fourier series simulation
ϕ_Z	transfer function applied to a random function Z
$\phi(\omega_m)$	random phase

$\boldsymbol{\omega_m}$ and $\boldsymbol{\omega'_m}$	discretized frequency on $\boldsymbol{m}^{th}$ node
$\boldsymbol{\omega}$	diagonal matrix with zonal anisotropy parameters
ω	1. frequency
	2. kernel window in the local linear regression
ω_l	power for lower extrapolation tail
ω_m and ω'_m	discretized frequency on m^{th} node
ω_u	power for upper extrapolation tail
ω_1	parameter in first principal direction of zonal anisotropy
ω_2	parameter in second principal direction of zonal anisotropy
ω_3	parameter in third principal direction of zonal anisotropy
$\Delta\omega$	frequency increments
$\Im$	imaginary part of the complex number
$\Re$	real part of the complex number
*	estimation superscript
$\star$	convolution operator
$\breve{}$	reverse operator
$\hat{}$	1. unit vector operator
	2. regression operator
$\|\|\ \|\|$	magnitude operator for vectors

Abbreviations

1-D	**1** (**One**) **D**imensional
2-D	**2** (**Two**) **D**imensional
3-D	**3** (**Three**) **D**imensional
AI	**A**coustic **I**mpedance
BLUE	**B**est **L**inear **U**biased **E**stimator
BMI	**B**lock **M**atrix **I**nversion
CDF	**C**umulative **D**istribution **F**unction
CLT	**C**entral **L**imit **T**heorem
fBm	**F**ractal **B**rownian **M**otion
FFT	**F**ast **F**ourier **T**ransform
fGn	**F**ractal **G**aussian **N**oise
FIM	**F**ourier **I**ntegral **M**ethod
FS	**F**ractal **S**imulation
FSS	**F**ourier **S**eries **S**imulation
GFS	**G**rid-**F**ree **S**imulation
GR	**G**amma **R**ay
IHS	**I**nclined **H**eterolithic **S**trata
LAS	**L**og **A**SCII **S**tandard
LMC	**L**inear **M**odel of **C**oregionalization
LMR	**L**inear **M**odel of **R**egionalization

LUS	**L**ower Triangular Matrix - **U**pper Triangular Matrix **S**imulation
MA	**M**oving **A**verage Simulation
MCS	**M**onte **C**arlo **S**imulation
MPS	**M**ultiple **P**oint **S**tatistics
NST	**N**ormal **S**core **T**ransformation
PCA	**P**rincipal **C**omponent **A**nalysis
PDF	**P**robability **D**ensity **F**unction
PE	**P**hotoelectric **E**ffect
PFS	**P**robability **F**ield **S**imulation
PSB	**P**oint **S**cale **B**lock value
SA	**S**imulated **A**nnealing
SAGD	**S**team **A**ssisted **G**ravity **D**rainage oil extraction method
SCK	**S**imple **C**o**K**riging
SGS	**S**equential **G**aussian **S**imulation
SIS	**S**equential **I**ndicator **S**imulation
SK	**S**imple **K**riging
SS	**S**pectral **S**imulation
TBS	**T**urning **B**ands **S**imulation
TLS	**T**urning **L**ines **S**imulation

Chapter 1

Introduction

Human society depends on natural resources including minerals and hydrocarbons. They are key requirements for the stable development and security of society. Unfortunately, these vital components are not renewable and they will become scarce and more costly at some point in the future. The extraction of minerals and hydrocarbons must be effectively managed using the best available technology. An improved understanding of the distribution of the natural resources will help to develop them in the most sustainable and economic manner possible.

A field of applied statistics, called geostatistics, serves to understand, characterize, and predict the spatial and temporal properties of the natural phenomena and to quantify uncertainty in the predictions [36, 50]. Geostatisticians estimate and simulate attributes of natural phenomena by analyzing available data and communicating with experts from other geosciences and engineering fields including geologists, geophysicists, and petroleum or mining engineers. This makes geostatistics a cornerstone of geomodeling natural phenomena [30, 64].

Conventionally, numerical geostatistical models of the natural phenomena are constructed on a grid of fixed node spacing. This approach is inflexible for

current applications where the resolution is increased in some areas and different representations are needed to evaluate alternative recovery mechanisms. This research work primarily focuses on removing the dependency of geostatistical simulation methods on a predetermined simulation grid. The final realizations of natural phenomena is constrained to scattered point-scale hard data and exhaustively sampled block-scale measurements. Both univariate and multivariate systems are studied.

1.1 Problem Statement

Natural phenomena are uncertain because of limited data and somewhat chaotic nature of the original deposition and subsequent diagenesis. Geostatistical methods aim to generate numerical models that honour not only data, but also spatial characteristics of the modeled system. Moreover, these techniques are capable of characterizing uncertainty [17, 26].

Two classes of geostatistical techniques are distinguished: estimation and simulation methods [17]. The estimation algorithms produce a single estimate at each location leading to a smooth model of the property. Simulation methods generate an ensemble of stochastic models or realizations with higher local variability to describe the natural phenomena with a reasonable pattern of fluctuations and a measure of uncertainty due to limited data. Both local and global uncertainty can be derived from a set of simulated realizations. Local estimation error is computed at each location. Global uncertainty is assessed by processing each realization through a recovery transfer function, and, therefore, directly observing uncertainty in all critical model responses [36]. As a result, simulation methods have gained popularity to model natural phenomena.

Although the theory for geostatistical simulation is quite well developed, there are several aspects frequently faced in practice that make modeling petroleum

reservoirs or mineral deposits more difficult. Model regridding, modeling of multivariate systems, and accounting for the multiscale data are key challenges.

Many geostatistical simulation techniques are inherently free of the simulation grid, but not implemented or applied as such. Conventional simulation is performed on a regular grid – a set of evenly organized nodes in the space. The grid has to be defined before any simulation starts. The data locations do not have to align with the grid nodes. The grid-based discretization of the simulation introduces complexity to include additional simulated values at a finer node density, or to perform simulation at irregularly distributed locations. The model has to be rebuilt with another grid of different node density or with multiple grids to simulate local variations at different resolutions and various grid densities. In all these cases, the simulation results are non-unique and depend on the order of simulation.

Fig. 1.1 demonstrates the importance of modeling flexibility for simulation methods for characterization of steam assisted gravity drainage (SAGD) petroleum reservoirs in northern Alberta, Canada [9]. The geological properties must initially be characterized over many square kilometers at a low resolution model grid, then multiple high resolution model grids at the meter scale are required to assess potential development areas. Regridding of the geological property is always required. The challenge is to keep models of different resolutions consistent and reproducible. The high-resolution model for flow simulation shown in brown should comply with low-resolution model for resources estimation shown in black. A grid-free simulation method, where simulation of the natural phenomena is presented continuously in the space as a function of the coordinates of the simulation locations, would eliminate the need for regridding and make modeling very flexible with respect to the configuration and number of the simulation locations.

The modeling of natural deposits often considers simultaneous simulation

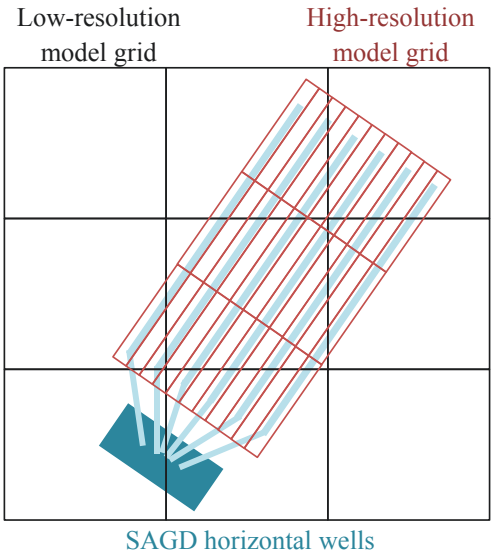


Figure 1.1: Illustrative example of one of the challenges faced in the modelling SAGD petroleum reservoirs – model regridding shown in a plan view.

of multiple correlated properties. Most of the properties modeled by geostatistics for the petroleum and mining industries are geological in nature and time invariant. There are many correlated variables. Porosity, which defines fluid storage ability of the rock, permeability, which characterizes ease of fluid to flow through the rock, and reservoir fluid saturations, which quantify the volumes of oil, gas, and water stored in the void space of the rock, are key correlated petrophysical properties used to manage petroleum reservoirs, and should be modeled simultaneously preserving their relationship [18]. Various minerals are also found to be deposited in some correlated fashion [13]. Such interaction between multiple variables should be preserved in the final grid-free simulation model.

In practice, there are different data types sampled at various scales. There could be point-scale core data sampled along wells and exhaustively sampled block-scale seismic data measured over large volumes as shown in Fig. 1.2. All these data types sampled at various scales should be properly assimilated into

the model within the grid-free simulation framework. In addition, there is sometimes a need to generate geostatistical models at an arbitrarily high resolution to permit upscaling into the desired model size. The output model size and orientation could change for different goals.

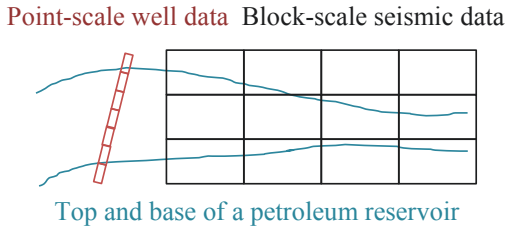


Figure 1.2: Illustrative example of one of the challenges faced in the modelling petroleum reservoirs – data assimilation of various scales shown in cross-sectional view.

A grid-free simulation (GFS) algorithm is sought to address aforementioned aspects of modeling properties of natural phenomena encountered in the practice of petroleum and mining projects independent from the modeling grid. An example of the univariate GFS is shown in Fig. 1.3. The realization is conditioned to data sampled along a vertical well in the middle of the cross-section. The spatial structure of porosity field is anisotropic. The realization is shown on three sets of the simulation nodes. The first set of nodes is scattered irregularly over the cross-section. The next two sets are placed evenly with various densities. The resulting anisotropic realizations are shown next to the simulation nodes maps in a pixel format. The area closest to each node is assigned to the node and colored with the nodal value. The realizations are independent of the grid and can be resolved at any locations at arbitrarily high resolution.

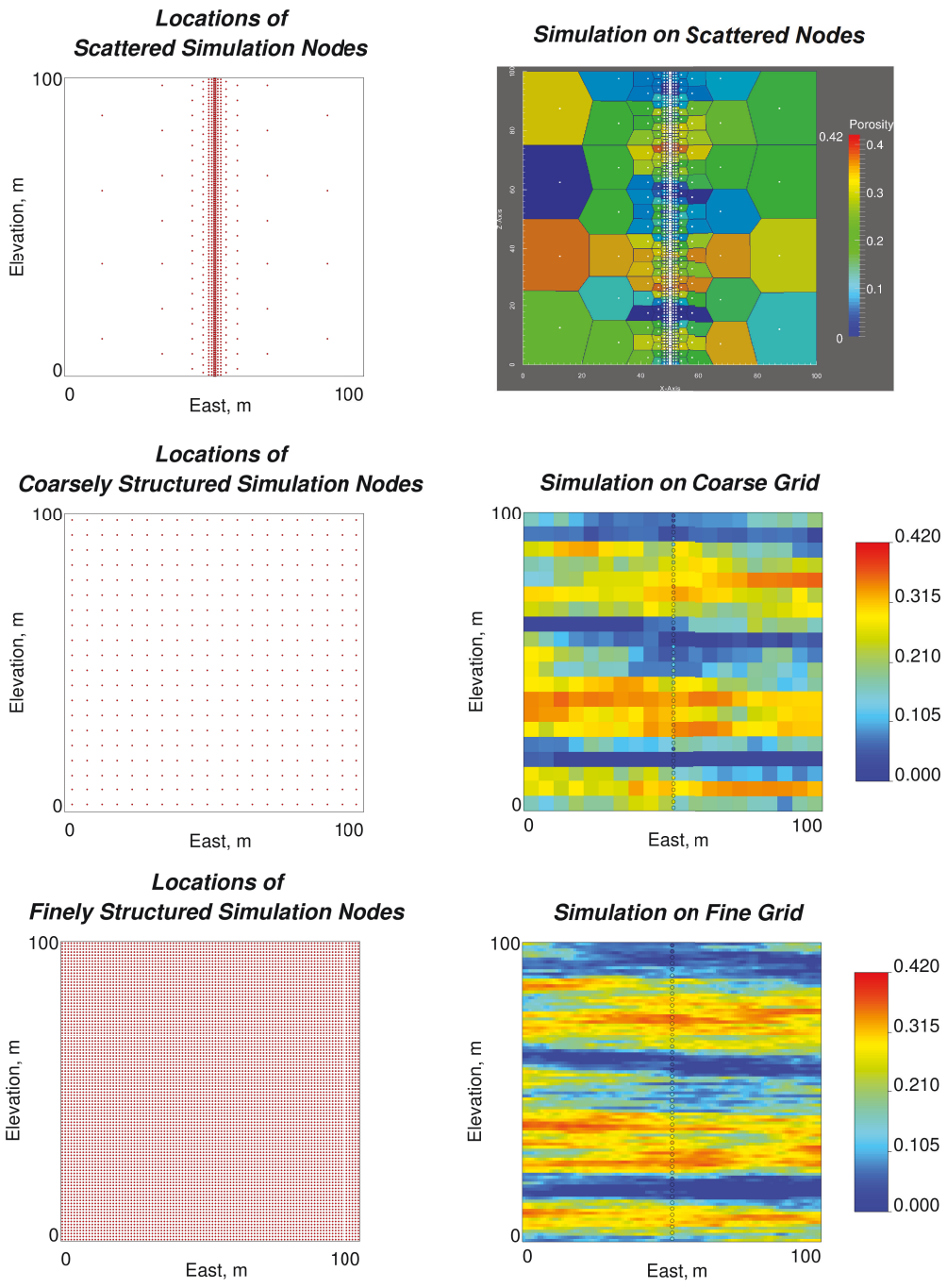


Figure 1.3: Grid-free simulation example on different sets of the simulation nodes – irregularly scattered nodes, coarsely structured nodes, and finely structured nodes.

1.2 Problem Solution

A grid-free conditional simulation engine will be developed to provide a flexible tool for modeling multivariate natural phenomena as a function of the coordinates, data values, and spatial structure of the modeled system. This approach will provide flexibility to resolve the model of a multivariate system at any grid configuration (scattered nodes, structured or unstructured grids) and resolution conditional to multiscale data with complete consistency and a possible reduction in computational time. The simulated values are tied to location and are always the same regardless of the order of simulation. The relationship between multiple variables will be explicitly reproduced. The point-scale hard data and gridded block-scale soft data will be used as conditioning data in the grid-free simulation. The hard data denote direct measurements of the simulated geological properties. The soft data imply indirect measurements of the simulated variables that are correlated with hard data.

A comparison of the proposed simulation engine with conventional simulation algorithms is schematically depicted in Fig. 1.4 for modeling geologic properties of the petroleum reservoirs [64]. The data come from multiple sources sampled at different scales. Point-scale laboratory measurements of core, larger scale well log and well test data, and exhaustively sampled seismic attributes are a few examples of frequently encountered data types in petroleum reservoir modeling. These measurements are analyzed to depict spatial structure of the modeled primary variables such as porosity, permeability, or fluid saturations. The spatial structure of the system is quantified through variogram models for two-point geostatistical models [37]. In conventional geomodeling, the geologic properties are simulated on a regular model grid at fixed scale. With the proposed grid-free simulation algorithm, the model can be realized at any location in

the space at arbitrary resolution and upscaled to any scale. The scale of reporting the simulated values can be addressed by an adequate discretization before upscaling. For instance, more heterogeneous regions or regions of higher interest can be resolved at finer simulation nodes density than more homogenous or less important regions of the model. Or, a larger lease area could be modeled on a coarse grid for resource estimation, and some smaller areas around particular wells could be modeled on a finer grid for flow simulation to accurately account for local geologic heterogeneities. Both models would be consistent with each other. This will provide modeling flexibility and reduce computational time by simulating at coarser resolution for less vital parts of the model. The model is not stored as a set of gridded values, but rather as a function, that reduces memory storage requirements. Simulation could be performed independently and simultaneously at various locations with parallel programming and multiple processor computers.

The fundamentals of the proposed grid-free simulation engine algorithm are summarized in Fig. 1.5. The key steps are the definition of the key variables to model, assemblage of relevant data, simulation in a grid-free form analytically, and retrieval of the simulated values of any realization of any variable at any location in the space. A more detailed workflow of the GFS is provided in Fig. 1.6. The simulation is performed in normal units at a point scale. The two-step conditional simulation has been chosen, because kriging-based conditioning of unconditional simulation is inherently grid-free, and many fast unconditional simulation methods are available and can be enhanced to become grid-free [10]. A new fast unconditional simulation method called Fourier series simulation (FSS) is proposed in this thesis. This method appears to fall into a class of the spectral simulation techniques, but has a distinct interpretation [58, 71]. The unconditional simulation is represented as a linear finite sum of weighted independent random factors. Independent random factors are stochastic in nature

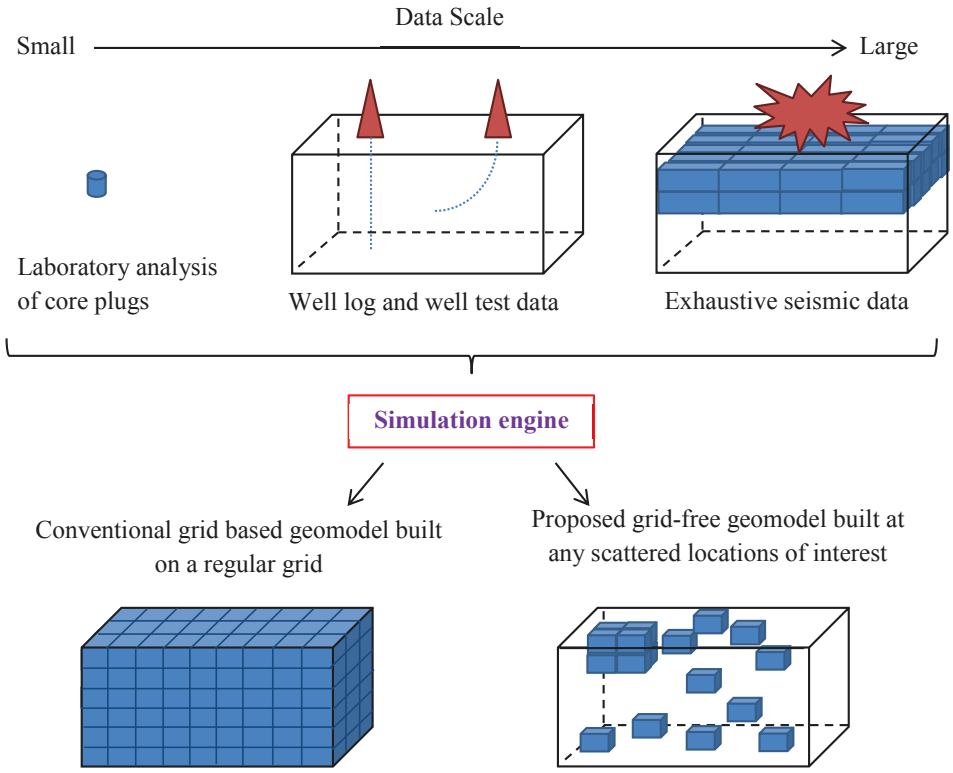


Figure 1.4: Comparison of conventional and grid-free simulation approaches to modeling petroleum systems in presence of multiscale data.

and expressed analytically as a function of the simulation location through cosine functions with random phase. The corresponding weights are derived from Fourier series-based decomposition of the target spatial function of the system structure [8, 26]. Therefore, the resulting conditional grid-free simulation is expressed explicitly in analytical form as a function of some fixed parameters, and most importantly, as a function of the coordinates of the simulation locations.

An illustrative example of a grid-free simulation of a geologic property within reservoir boundaries is shown in Fig. 1.7 in interactive 3-D mode, where the property is presented continuously in space. Several slices are shown, which can be viewed individually in the figure.

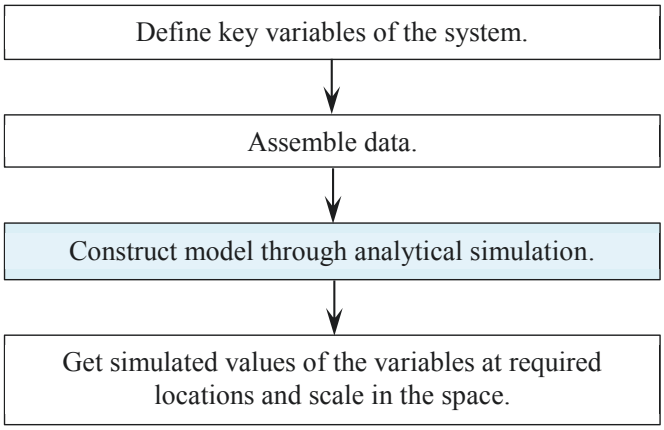


Figure 1.5: Concept of grid-free simulation.

1.3 Thesis Statement

A grid-free simulation (GFS) method provides an algorithm for the grid independent conditional geostatistical simulation of multiple correlated properties of natural phenomena, where simulation is expressed as a spatial continuous function of the coordinates of the simulation locations, data values, and spatial structure of the modeled system, to improve accuracy of numerical geomodels by ensuring consistency in realizations resolved at any resolution.

1.4 Delimitations and Limitations

The delimitations of the proposed grid-free conditional simulation algorithm are as follows. The discussion will be focused only on static properties with no time component. In addition, only continuous properties will be studied. For possible grid-free modeling of categorical variables see [31–33]. The geostatistical tools operate within the multivariate spatial Gaussian (normal) space. The preprocessing of the data, which includes normal score transformation, correlation

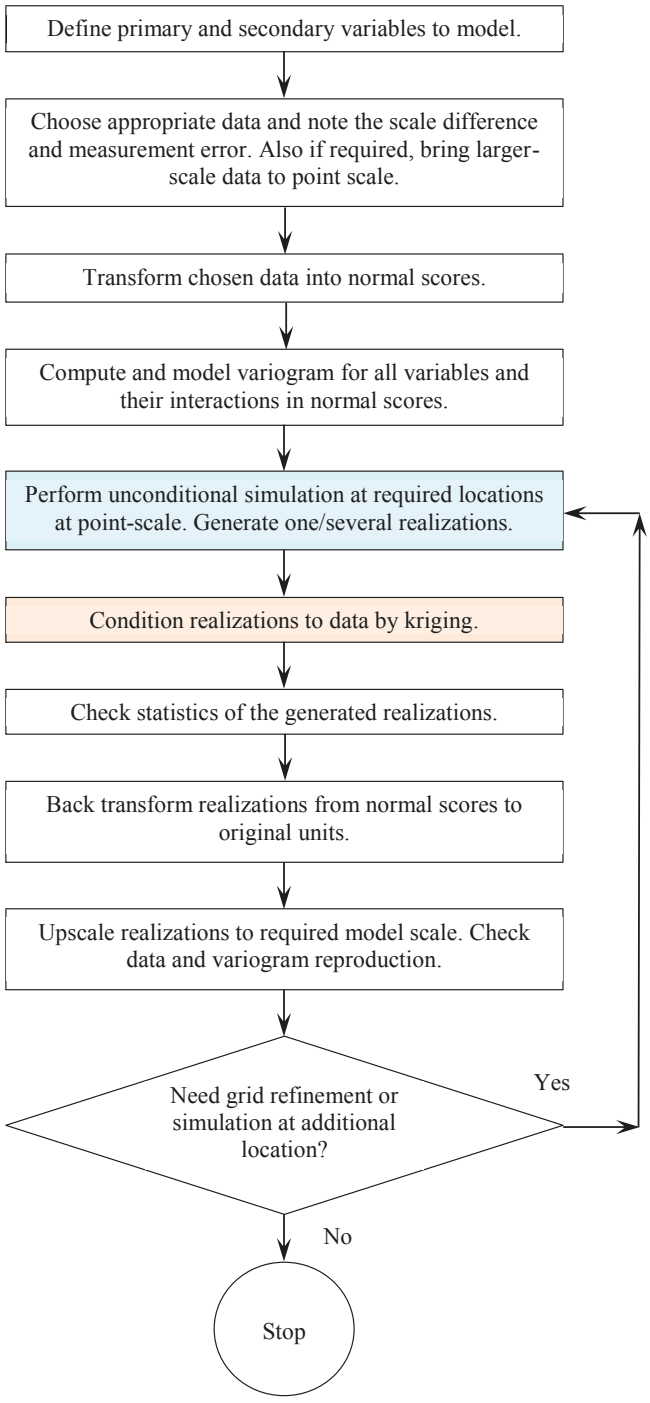


Figure 1.6: Workflow for modeling deposits with grid-free simulation.

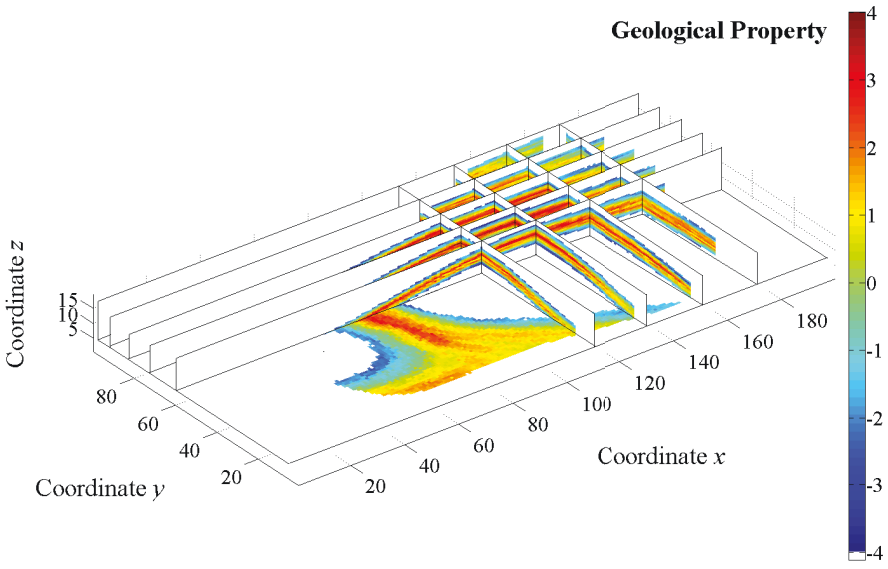


Figure 1.7: Conceptual geostatistical grid-free model of a geologic property in interactive 3-D mode. Please click on the figure to activate 3-D view mode.

and de-correlation techniques, will not be emphasized, but are crucial [17]. Characterization of the multivariate spatial structure is important. Conventionally, the relationship between multiple correlated variables is preserved through the linear model of coregionalization (LMC) [26]. It is assumed that the form of the LMC is well known before the simulation of the multivariate system is performed. The multiscale nature of the data is acknowledged, but not addressed explicitly.

Some limitations of the proposed grid-free conditional simulation approach are present and may be managed or overcome with additional work. The distribution parameters of the multiple variables are assumed stationary and constant over the modeling region. Any non-stationarity could be resolved through trend modeling, or by decomposing the non-stationary domain into multiple stationary domains based on categorical variable modeling [64]. An example of how

non-stationarity can be addressed within the GFS by dividing non-stationary region into two stationary domains A and B is presented in Fig. 1.8 as a single GFS realization of a mineral grade. The stationary domains are separated by a geological fault, which is defined before the simulation is performed.

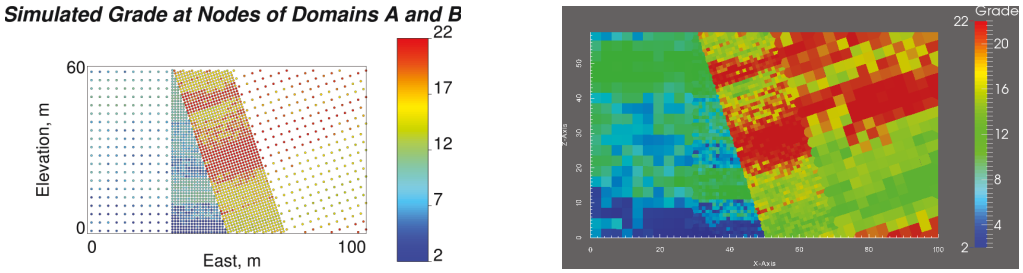


Figure 1.8: Example of conditional grid-free simulation of non-stationary geological property at point scale – (a) point representation of the simulation, and (b) pixel representation of the simulation. The non-stationary domain consists of two stationary domains A and B separated by a fault.

1.5 Outline

The thesis consists of seven chapters. Chapter 1 is the introduction. Chapter 2 covers theoretical background of the essentials of geostatistics and simulation methods. The next two chapters are devoted to the framework of two-step grid-free simulation. Chapter 3 provides information on the unconditional grid-free simulation. Chapter 4 covers the conditioning of unconditional simulation. Once the theoretical framework is presented, some practical aspects are discussed. Chapter 5 shows implementation aspects of the proposed grid-free simulation method. Chapter 6 demonstrates practical implementation of grid-free simulation to a real case study. The last Chapter 7 wraps up the thesis with Conclusions. The description of developed Fortran codes is provided in an Appendix.

Chapter 2

Background

Geostatistics has evolved from applied statistics initially to satisfy the needs of the mining engineers to delineate ore bodies [50]. Later, it spread to other engineering fields including petroleum engineering, where oil and gas reservoirs are characterized with the help of geostatistical tools [56, 64]. Key definitions and concepts are covered in this chapter. Spatial properties of the natural entity such as mineral grade or rock porosity are studied.

Numerous geostatistical algorithms have been developed and implemented abundantly in the software. Care is taken to choose an algorithm that reasonably describes the geological property under consideration. It is essential to understand the theory behind each algorithm to avoid blunders and mistakes that could come from blind implementation. The important estimation and simulation techniques are reviewed and classified below. It is worth to mention that simulation methods are placed at the top of the list of geostatistical tools due to their increased use in creating realistic models of natural variability and their quantification of uncertainty.

2.1 Essentials of Geostatistics

Definitions, assumptions, and statistical calculations are explained before presenting geostatistical algorithms for modeling regionalized variables.

2.1.1 Modeling Attributes

Geostatistical tools are capable of modeling properties of the natural phenomena in space. The modeling can occur in one-dimensional (1-D), two-dimensional (2-D), or three-dimensional (3-D) space. A location in the space $\boldsymbol{u}$ is often determined by Cartesian coordinates mapped along orthogonal axes. The location vector $\boldsymbol{u}$ in μ-dimensional space consists of μ location coordinates as shown in Eq. (2.1), where A is the modeling domain.

$$\boldsymbol{u} = [u_1 \ ... \ u_\mu]^T, \ \boldsymbol{u} \in A \tag{2.1}$$

A stochastic or random process is the probabilistic interpretation of the properties of the natural phenomenon. A random function Z is the analytical representation of the stochastic process continuous in space. When random function is tied to a location $\boldsymbol{u}$, the random variable $Z(\boldsymbol{u})$ at a point scale is defined. Thus, a random variable is the subset of the random function assigned to a particular location and volume. The relationship between the random function and a random variable at a point scale is shown in Eq. (2.2). The random variable $Z(\boldsymbol{u})$ can take continuous or categorical values z. Continuous random variables take values from continuous spectrum, while categorical variables can take one of a discrete set of values. The corresponding random function Z is denoted continuous or categorical, respectively.

$$Z = \{Z(\boldsymbol{u}), \ \boldsymbol{u} \in A\} \tag{2.2}$$

The point scale is the smallest practical volume that is often considered to be the scale or support of the data. This is the smallest scale that the space can be discretized. Values of a property at larger scales can be uniquely derived from the values of the same property at a point scale. The reverse procedure (downscaling) is not unique. The scale difference of the random variables can be adjusted through the regularization operator for properties that change linearly with scale [36]. For instance, if additive random variable describes a volume V centered at location $\boldsymbol{u}$, which is larger than a point scale, the block-scale variable $Z(V(\boldsymbol{u}))$ can be defined from its point scale counterparts $Z(\boldsymbol{u})$ contained within $V(\boldsymbol{u})$ as shown in Eq. (2.3).

$$Z(V(\boldsymbol{u})) = \frac{1}{V(\boldsymbol{u})} \int_{V(\boldsymbol{u})} Z(\boldsymbol{u}')d\boldsymbol{u}',\ \boldsymbol{u}' \in V(\boldsymbol{u}) \tag{2.3}$$

Values of the random function Z may be known at some locations, while values at other locations should be inferred conditional to these known values. Locations with known values are called data or sampled, while locations with unknown values are called unsampled. Data values $Z(\boldsymbol{u}_\alpha)$, $\alpha = 1, ..., N$, are differentiated from the estimation values $Z^*(\boldsymbol{u})$ or conditional simulation values $Z^s(\boldsymbol{u})$ of the random function Z. Both estimation and simulation are conditioned to the data and are constructed using the spatial structure of the data. While simulation would preserve imposed spatial structure, estimation fails to do so. Simulated values of the random function Z that only preserve spatial structure, but do not honor data values, are called unconditional simulation values $Z^{us}(\boldsymbol{u})$. Note that often the conditionally or unconditionally simulated value at location $\boldsymbol{u}$ is also expressed as $Z(\boldsymbol{u})$. The data can be either sampled at scattered locations or measured on a grid. These data types are called scattered data and exhaustively sampled data, respectively. All data values can be stored in a vector $\boldsymbol{Z}^d$ shown in Eq. (2.4). Here, α denotes data location index.

$$\boldsymbol{Z}^d = [Z(\boldsymbol{u}_1) \; ... \; Z(\boldsymbol{u}_\alpha) \; ... \; Z(\boldsymbol{u}_N)]^T \tag{2.4}$$

The modeled system is defined by a set of random variables distributed regularly on a grid in a conventional modeling or at scattered locations in modeling with unstructured grids. The grid-free approach does not imply any grid or a set of scattered nodes before simulation starts. It rather constructs a model in a polynomial form, values of which can be recalled at any location in the space. The conventional representation of the model is shown in Eq. (2.5), where M is the number of simulation nodes. Here β denotes location index of modeled variable.

$$\boldsymbol{Z} = [Z(\boldsymbol{u}_1) \; ... \; Z(\boldsymbol{u}_\beta) \; ... \; Z(\boldsymbol{u}_M)]^T \tag{2.5}$$

When multiple correlated modeling variables are present, a multivariate system is considered. Therefore, when K multiple random functions are combined together into a single function, it is denoted as a random set of functions Z shown in Eq. (2.6). The random function Z denotes a univariate system, unless it is specified otherwise. In the multivariate context, the variable that is modeled is called primary. A secondary variable constrains or conditions the primary variable at unsampled locations.

$$Z = \{Z_k, \; k = 1, ..., K\} \tag{2.6}$$

All features inherent for univariate systems can be extended to multivariate systems. For instance, the data of the multivariate system can be stored in a vector form of size $\sum_{k=1}^{K} N_k \times 1$ as shown in Eq. (2.7). Also, the model of the multivariate system can be presented in matrix form of size $M \times K$ as in Eq. (2.8).

$$\boldsymbol{Z}^d = [Z_1(\boldsymbol{u}_1) \; ... \; Z_1(\boldsymbol{u}_{N_1}) \; ... \; Z_k(\boldsymbol{u}_1) \; ... \; Z_k(\boldsymbol{u}_{N_k}) \; ... \; Z_K(\boldsymbol{u}_1) \; ... \; Z_K(\boldsymbol{u}_{N_K})]^T \quad (2.7)$$

$$\boldsymbol{Z} = \begin{bmatrix} Z_1(\boldsymbol{u}_1) & ... & Z_k(\boldsymbol{u}_1) & ... & Z_K(\boldsymbol{u}_1) \\ & & ... & & \\ Z_1(\boldsymbol{u}_\beta) & ... & Z_k(\boldsymbol{u}_\beta) & ... & Z_K(\boldsymbol{u}_\beta) \\ & & ... & & \\ Z_1(\boldsymbol{u}_M) & ... & Z_k(\boldsymbol{u}_M) & ... & Z_K(\boldsymbol{u}_M) \end{bmatrix} \quad (2.8)$$

The essential geostatistical measures are reviewed next.

2.1.2 Geostatistical Measures

The random variable $Z(\boldsymbol{u})$ is characterized by a statistical distribution. The distribution could be written as a cumulative distribution function (CDF) $F_{Z(\boldsymbol{u})}(z)$ or probability density function (PDF) $f_{Z(\boldsymbol{u})}(z)$, where z represents outcomes or values of $Z(\boldsymbol{u})$ [77]. The CDF and PDF for a univariate case are shown in Eqs. (2.9) and (2.10). The $Prob\{Z(\boldsymbol{u}) \leq z\}$ is the probability of random variable $Z(\boldsymbol{u})$ to be less than or equal to value z. The $Prob\{z' \leq Z(\boldsymbol{u}) \leq z''\}$ states the probability of random variable $Z(\boldsymbol{u})$ to be between values z' and z''. The schematic of local distribution of random variable $Z(\boldsymbol{u})$ is shown in Fig. 2.1 in terms of the PDF. A distribution that is tied to a particular location in the space is called local distribution. The distribution that characterizes the entire random function or all random variables in the modeling domain at once is called a global distribution. Both PDF and CDF are used for the uncertainty quantification in estimation and simulation.

$$F_{Z(\boldsymbol{u})}(z) = Prob\{Z(\boldsymbol{u}) \leq z\},\ -\infty < z < \infty \tag{2.9}$$

$$\int_{z'}^{z''} f_{Z(\boldsymbol{u})}(z)dz = Prob\{z' \leq Z(\boldsymbol{u}) \leq z''\} \tag{2.10}$$

The CDF and PDF of the multiple random variables of same random function or CDF and PDF of multivariate system are depicted in Eqs. (2.11) and (2.12).

$$F_{Z_1(\boldsymbol{u}_1)...Z_K(\boldsymbol{u}_K)}(z_1, ..., z_K) = Prob\{Z_1(\boldsymbol{u}_1) \leq z_1, ..., Z_K(\boldsymbol{u}_K) \leq z_K\} \tag{2.11}$$

$$\begin{aligned} &\int_{z_1'}^{z_1''} ... \int_{z_K'}^{z_K''} f_{Z_1(\boldsymbol{u}_1)...Z_K(\boldsymbol{u}_K)}(z_1, ..., z_K)dz_1...dz_K \\ &\qquad = Prob\{z_1' \leq Z_1(\boldsymbol{u}_1) \leq z_1'', ..., z_K' \leq Z_K(\boldsymbol{u}_K) \leq z_K''\} \end{aligned} \tag{2.12}$$

The relationship between CDF and PDF can be expressed as shown in Eqs. (2.13) and (2.14) for univariate systems, and as in Eqs. (2.15) and (2.16) for multivariate systems.

$$F_{Z(\boldsymbol{u})}(z) = \int_{-\infty}^{z} f_{Z(\boldsymbol{u})}(z')dz' \tag{2.13}$$

$$f_{Z(\boldsymbol{u})}(z) = \frac{dF_{Z(\boldsymbol{u})}(z)}{dz} \tag{2.14}$$

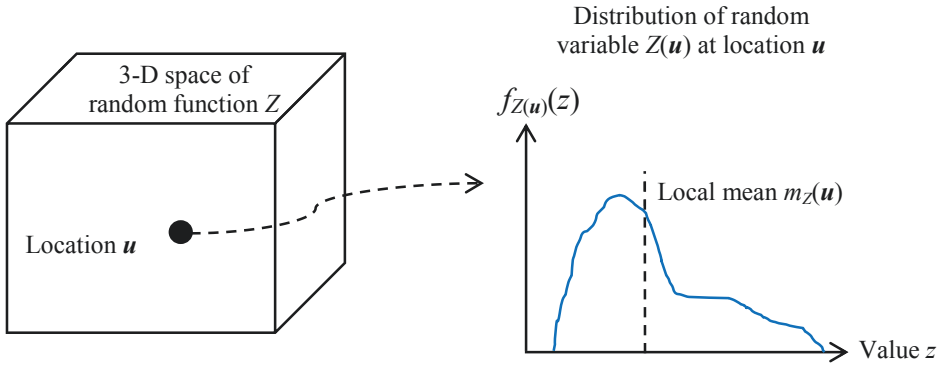


Figure 2.1: Schematic of local distribution of random variable $Z(\boldsymbol{u})$.

$$F_{Z_1(\boldsymbol{u}_1)...Z_K(\boldsymbol{u}_K)}(z_1, ..., z_K) = \int_{-\infty}^{z_1} ... \int_{-\infty}^{z_K} f_{Z_1(\boldsymbol{u}_1)...Z_K(\boldsymbol{u}_K)}(z'_1, ..., z'_K) dz'_1 ... dz'_K \tag{2.15}$$

$$f_{Z_1(\boldsymbol{u}_1)...Z_K(\boldsymbol{u}_K)}(z_1, ..., z_K) = \frac{\partial^K F_{Z_1(\boldsymbol{u}_1)...Z_K(\boldsymbol{u}_K)}(z_1, ..., z_K)}{\partial z_1 ... \partial z_K} \tag{2.16}$$

The distribution of a random variable $Z(\boldsymbol{u})$ can be summarized by some parameters such as the mean $m_Z(\boldsymbol{u})$ or variance $\sigma_Z^2(\boldsymbol{u})$. The distribution of several random variables of the same random function Z can be characterized by the covariance $C_Z(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{h})$ [26]. The variance is a special case of covariance, when $\boldsymbol{h} = \boldsymbol{0}$. The vector $\boldsymbol{h}$ between two locations $\boldsymbol{u}$ and $\boldsymbol{u} + \boldsymbol{h}$ is called the separation lag vector. Mean stands for the average value of the outcomes of the random variable. Variance quantifies the squared deviations of the outcomes from the mean. The covariance function characterizes variations of the random variables in space. The mean is also called a first-order moment of the distribution, while

the covariance is a second-order moment. Both of them are expressed through the expected value operator $E\{\}$ shown in Eq. (2.17) for univariate system, where ϕ_Z is some transfer function applied to a random function Z. The expected value operator accounts for the probability density function. The mean is derived with the expected value operator as presented in Eqs. (2.18). The covariance has the form shown in Eq. (2.19). A special form of the covariance is variance $\sigma_Z^2(\boldsymbol{u})$, which is shown in Eq. (2.20). Note that all these distributional parameters are tied to the location in the space. The covariance is related to two random variables at a time, and, therefore, is referred as a two-point statistic.

$$E\{\phi_Z\} = \int_{-\infty}^{\infty} \phi_Z f_{Z(\boldsymbol{u})}(z)dz \tag{2.17}$$

$$m_Z(\boldsymbol{u}) = E\{Z(\boldsymbol{u})\} = \int_{-\infty}^{\infty} z f_{Z(\boldsymbol{u})}(z)dz \tag{2.18}$$

$$\begin{aligned} C_Z(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{h}) &= E\{(Z(\boldsymbol{u}) - m_Z(\boldsymbol{u}))\,(Z(\boldsymbol{u}+\boldsymbol{h}) - m_Z(\boldsymbol{u}+\boldsymbol{h}))\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (z' - m_Z(\boldsymbol{u}))(z'' - m_Z(\boldsymbol{u}+\boldsymbol{h})) f_{Z(\boldsymbol{u})Z(\boldsymbol{u}+\boldsymbol{h})}(z', z'')dz'dz'' \end{aligned} \tag{2.19}$$

$$\sigma_Z^2(\boldsymbol{u}) = C_Z(\boldsymbol{u}, \boldsymbol{u}) = E\{(Z(\boldsymbol{u}) - m_Z(\boldsymbol{u}))^2\} = \int_{-\infty}^{\infty} (z - m(\boldsymbol{u}))^2 f_{Z(\boldsymbol{u})}(z)dz \tag{2.20}$$

The semivariogram or simply variogram $\gamma_Z(\boldsymbol{u},\boldsymbol{u}+\boldsymbol{h})$ is an alternative to the covariance function to describe spatial structure of the random function Z. The variogram is also a two-point statistic. The advantage of the variogram over the covariance is avoidance of the local mean in the calculation. The expression for the variogram is shown in Eq. (2.21) for the univariate case.

$$
\begin{aligned}
2\gamma_Z(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{h}) &= E\{(Z(\boldsymbol{u}) - Z(\boldsymbol{u}+\boldsymbol{h}))^2\} \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (z' - z'')^2 f_{Z(\boldsymbol{u})Z(\boldsymbol{u}+\boldsymbol{h})}(z', z'')dz'dz''
\end{aligned}
\tag{2.21}
$$

Stationarity is a vital geostatistical assumption that implies constant statistical parameters across the modeling domain A [36]. Stationarity imposes constraints to a spatial structure of the modeled property of the natural phenomenon. The physical interpretation is that regions of similar geological deposition or character are grouped together and modeled separately from other regions with distinct geological properties. This assumption defines boundaries between statistical populations, and suggests appropriate geomodeling methodology.

First-order stationarity assumption implies that local mean $m_Z(\boldsymbol{u})$ is the same over the entire modeling domain A (Eq. (2.22)). Second-order stationarity states that spatial covariance and variogram do not depend on particular two locations $\boldsymbol{u}$ and $\boldsymbol{u} + \boldsymbol{h}$ in the space, but rather depend only on the separation lag vector $\boldsymbol{h}$ between any two points in the modeling domain A as shown in Eqs. (2.23) – (2.25). The covariance function is related to the variogram in an elegant form for random variables from a first- and second-order stationary random function. This relationship between the stationary variogram and covariance is expressed in Eq. (2.26). The first- and second-order stationarity assumption is commonly made in practice.

$$
m_Z(\boldsymbol{u}) = m_Z, \ \forall \boldsymbol{u} \in A \tag{2.22}
$$

$$
C_Z(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{h}) = C_Z(\boldsymbol{h}), \ \forall \boldsymbol{u} \in A \tag{2.23}
$$

$$\gamma_Z(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{h}) = \gamma_Z(\boldsymbol{h}),\ \forall \boldsymbol{u} \in A \tag{2.24}$$

$$\sigma_Z^2(\boldsymbol{u}) = C_Z(\boldsymbol{0}) = \sigma_Z^2,\ \forall \boldsymbol{u} \in A \tag{2.25}$$

$$\gamma_Z(\boldsymbol{h}) = \sigma_Z^2 - C_Z(\boldsymbol{h}),\ \forall \boldsymbol{u} \in A \tag{2.26}$$

A non-stationary random process $\tilde{Z}$ can be modified to a stationary one by removing a local non-stationary mean $m_{\tilde{Z}}(\boldsymbol{u})$, which is also called detrending, as presented in Eq. (2.27). The challenge lies in the computation of the trend and possible artifacts such as negative estimates.

$$Z(\boldsymbol{u}) = \tilde{Z}(\boldsymbol{u}) - m_{\tilde{Z}}(\boldsymbol{u}) \tag{2.27}$$

For the multivariate stationary case, expressions for the mean, covariance, and variogram are depicted in Eq. (2.28) – (2.30). The covariance and variogram are referred to as the cross-covariance $C_{Z_k Z_l}(\boldsymbol{h})$ and cross-variogram $\gamma_{Z_k Z_l}(\boldsymbol{h})$ for different random functions Z_k and Z_l. The relationship between the covariance and variogram of the stationary random functions is shown in Eq. (2.31). Same relationship between the symmetric covariance and variogram of the stationary random functions is shown in Eq. (2.32). The cross-correlation coefficient $\rho_{Z_k Z_l}(\boldsymbol{h})$ is another frequently used statistical parameter, which is the standardized covariance between two random variables. Its expression for the stationary multivariate case is shown in Eq. (2.33).

$$m_{Z_k} = E\{Z_k(\boldsymbol{u})\},\ \forall \boldsymbol{u} \in A,\ k = 1, ..., K \tag{2.28}$$

$$C_{Z_k Z_l}(\boldsymbol{h}) = E\{(Z_k(\boldsymbol{u}) - m_{Z_k})\,(Z_l(\boldsymbol{u}+\boldsymbol{h}) - m_{Z_l}))\},\ k,l = 1,...,K \tag{2.29}$$

$$2\gamma_{Z_k Z_l}(\boldsymbol{h}) = E\{(Z_k(\boldsymbol{u}) - Z_k(\boldsymbol{u}+\boldsymbol{h}))\,(Z_l(\boldsymbol{u}) - Z_l(\boldsymbol{u}+\boldsymbol{h}))\},\quad k,l = 1,...,K \tag{2.30}$$

$$2\gamma_{Z_k Z_l}(\boldsymbol{h}) = 2\sigma^2_{Z_k Z_l} - C_{Z_k Z_l}(\boldsymbol{h}) - C_{Z_l Z_k}(\boldsymbol{h}),\ \forall \boldsymbol{u} \in A,\ k,l = 1,...,K \tag{2.31}$$

$$\gamma_{Z_k Z_l}(\boldsymbol{h}) = \sigma^2_{Z_k Z_l} - C_{Z_k Z_l}(\boldsymbol{h}),\ \forall \boldsymbol{u} \in A,\ k,l = 1,...,K \tag{2.32}$$

$$\rho_{Z_k Z_l}(\boldsymbol{h}) = \frac{C_{Z_k Z_l}(\boldsymbol{h})}{\sqrt{\sigma^2_{Z_k}\sigma^2_{Z_l}}},\ \forall \boldsymbol{u} \in A,\ k,l = 1,...,K \tag{2.33}$$

In practice, the experimental variograms $\hat{\gamma}_Z(\boldsymbol{h})$ and $\hat{\gamma}_{Z_k Z_l}(\boldsymbol{h})$ are computed from the data points $\{Z_k(\boldsymbol{u}_\alpha), \alpha = 1,...,N_k, k = 1,...,K\}$ according to Eqs. (2.34) and (2.35) for univariate and multivariate systems to estimate the spatial distribution of the random functions, where N_p is the number of data pairs *approximately* separated by lag vector $\boldsymbol{h}$. The tolerance is introduced into search parameters to increase possible number of data pairs in the experimental variogram calculation for modeling stability [17]. Note that the data of random function pair Z_k-Z_l for multivariate case should be collocated to compute the experimental cross-semivariogram from Eq. (2.35).

$$\hat{\gamma}_Z(\boldsymbol{h}) = \frac{1}{2N_p}\sum_{\alpha=1}^{N_p}(Z(\boldsymbol{u}_\alpha) - Z(\boldsymbol{u}_\alpha + \boldsymbol{h}))^2 \tag{2.34}$$

$$\hat{\gamma}_{Z_k Z_l}(\boldsymbol{h}) = \frac{1}{2N_p} \sum_{\alpha=1}^{N_p} \left(Z_k(\boldsymbol{u}_\alpha) - Z_k(\boldsymbol{u}_\alpha + \boldsymbol{h})\right) \left(Z_l(\boldsymbol{u}_\alpha) - Z_l(\boldsymbol{u}_\alpha + \boldsymbol{h})\right) \tag{2.35}$$

The experimental variograms depend on the magnitude of the lag vector $\boldsymbol{h}$ and also on the direction. This anisotropy can be quantified through three principal orthogonal directions of continuity – major, minor, and medium directions of the continuity. The schematic of the idealized anisotropic experimental variogram ellipsoid in 3-D is shown in Fig. 2.2. The principal directions of continuity are not usually aligned with the axes of the coordinate system.

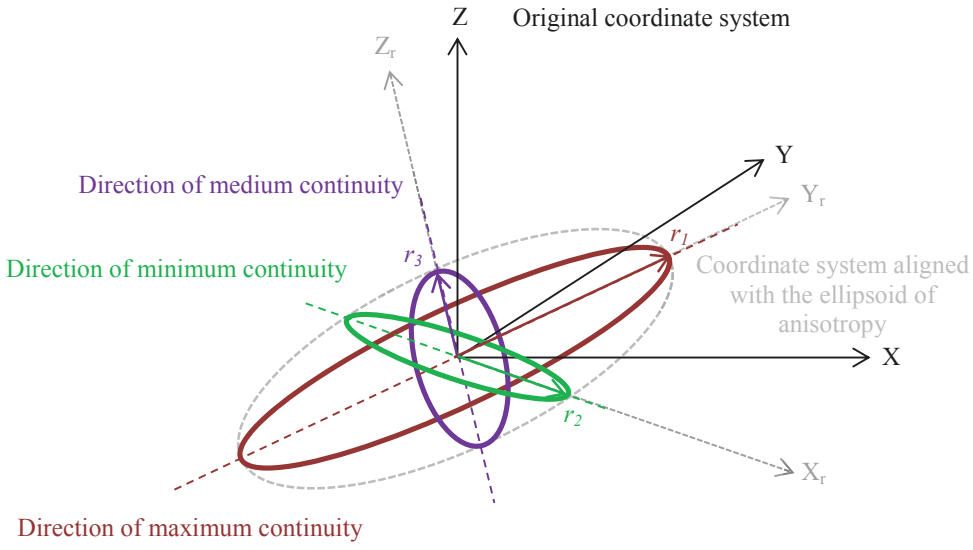


Figure 2.2: Schematic of the idealized anisotropic experimental variogram ellipsoid in 3-D.

Valid variogram models are fitted to the experimental variograms to have continuous uninterrupted variogram values for all possible lag vectors $\boldsymbol{h}$. Variogram models are used in the geostatistical estimation and simulation of the properties of the natural phenomena at unsampled locations. Most variogram

models require multiple structures, where each structure is one fundamental variogram function. The number of structures and their contribution to the variogram model are derived from a linear representation of the random function Z as a sum of weighted orthogonal (independent) random factors. Each factor has licit variogram structure ensuring that the resulting variogram structure of the random function Z is also licit.

This linear model is referred to as a linear model of regionalization (LMR) for univariate systems and linear model of coregionalization (LMC) for multivariate systems as shown in Eqs. (2.36) and (2.37). More generalized version of the LMC can be found in [48]. In the LMR, the random variable $Z(\boldsymbol{u})$ at a location $\boldsymbol{u}$ in the space is represented by a set of independent random factors $X_p(\boldsymbol{u}), p = 0, ..., P$, with zero mean and unit variance and weighted by coefficients $a_p, p = 0, ..., P$. In the LMC, K random variables stored in a vector $\boldsymbol{Z}(\boldsymbol{u})$, which are expressed as a sum of products of weighting matrix $\boldsymbol{A}_p$ and vector of random factors $\boldsymbol{X}_p(\boldsymbol{u})$ at some location $\boldsymbol{u}$ in the space. Random factors $X_{p,k}$ and $X_{p,l}$ for any k and l have same spatial structure for same p. Expressions for the elements of the multivariate systems are summarized in Eqs. (2.38) – (2.41). The coefficients $a_{p,kl}$ must be real numbers. They are derived from the sill contribution coefficients $b_{p,kl}$ of the licit variogram model, which are found by analyzing the experimental variograms. The LMC coefficient matrices $\boldsymbol{A}_p, p = 1, ..., P$, are the lower triangular matrices, what means that $a_{p,kl} = 0$ for $k < l$, $p = 1, ..., P$.

$$Z(\boldsymbol{u}) = m_Z(\boldsymbol{u}) + \sum_{p=0}^{P} a_p X_p(\boldsymbol{u}) \tag{2.36}$$

$$\boldsymbol{Z}(\boldsymbol{u}) = \boldsymbol{m}_Z(\boldsymbol{u}) + \sum_{p=0}^{P} \boldsymbol{A}_p \boldsymbol{X}_p(\boldsymbol{u}) \tag{2.37}$$

$$\boldsymbol{Z}(\boldsymbol{u}) = [Z_1(\boldsymbol{u}) \; ... \; Z_K(\boldsymbol{u})]^T \tag{2.38}$$

$$\boldsymbol{m}_Z(\boldsymbol{u}) = [m_{Z_1}(\boldsymbol{u}) \ ... \ m_{Z_K}(\boldsymbol{u})]^T \tag{2.39}$$

$$\boldsymbol{A}_p = \begin{bmatrix} a_{p,11} & ... & a_{p,1K} \\ ... & & ... \\ a_{p,K1} & ... & a_{p,KK} \end{bmatrix}, \ p = 1, ..., P \tag{2.40}$$

$$\boldsymbol{X}_p(\boldsymbol{u}) = [X_{p,1}(\boldsymbol{u}) \ ... \ X_{p,K}(\boldsymbol{u})]^T \tag{2.41}$$

The mean and spatial structure of the independent random factors $X_p(\boldsymbol{u}), p = 0, ..., P$, for the univariate case are presented in Eqs. (2.42) and (2.43). The spatial structure of each random factor has a fundamental form $\Gamma(\boldsymbol{h})$. The relationship between the variogram model of the random function Z and the variogram models of random factors is shown in Eq. (2.44), which is the key expression for fitting experimental variograms with multiple structures. The sill contribution coefficients b_p in Eq. (2.44) are either derived by simple trial-error approach, so the variogram model would approximate the experimental points the best, or by an iterative method such as the one presented in [26]. Here, a licit variogram model implies a semi-positive definite variogram model with all b_p coefficients being non-negative. Eq. (2.45) states the relationship between LMR coefficients $a_p, p = 0, ..., P$, and sill contribution coefficients $b_p, p = 0, ..., P$, for the univariate systems. The relationship between the variance of the random function and the contribution of the random factors should be held as shown in Eq. (2.46).

$$m_{X_p}(\boldsymbol{u}) = 0, \ \forall \boldsymbol{u} \in A, \ p = 0, ..., P \tag{2.42}$$

$$\gamma_{X_p X_q}(\boldsymbol{h}) = \begin{cases} \Gamma_{X_p}(\boldsymbol{h}), & \forall p = q \\ 0, & \forall p \neq q \end{cases} \tag{2.43}$$

$$\gamma_Z(\boldsymbol{h}) = \sum_{p=0}^{P} b_p \Gamma_{X_p}(\boldsymbol{h}) \tag{2.44}$$

$$b_p = a_p^2,\ p = 0, ..., P \tag{2.45}$$

$$\sigma_Z^2 = \sum_{p=0}^{P} b_p \tag{2.46}$$

The independent random factors $X_{p,k}(\boldsymbol{u}), p = 1, ..., P, k = 1, ..., K$, have similar properties in the multivariate case, where K random functions Z_k are modeled. Properties of the independent random factors are summarized in Eqs. (2.47) – (2.51). Eq. (2.49) states how direct- or cross-variogram models of the modeled random functions are related to the variogram models of the factors. The requirement for positive semi-definite variogram model is presented in Eq. (2.52), which states that eigenvalues τ_p of the sill contribution matrix $\boldsymbol{B}_p$ should be positive for all $p = 0, ..., P$ with $\boldsymbol{e}_p$ being eigenvectors. Eq. (2.53) presents additional requirements for positive semi-definiteness check of the sill contribution matrix $\boldsymbol{B}_p$ for licit variogram models. It states that determinant of the sill contribution matrix must be greater or equal to zero, and diagonal terms of the matrix $\boldsymbol{B}_p$ must be non-negative. The covariance between multiple variables has to be preserved as presented in Eq. (2.54).

$$m_{X_{p,k}}(\boldsymbol{u}) = 0,\ \forall \boldsymbol{u} \in A,\ p = 1, ..., P,\ k = 1, ..., K \tag{2.47}$$

$$\gamma_{X_{p,k}X_{q,l}}(\boldsymbol{h}) = \begin{cases} \Gamma_{X_p}(\boldsymbol{h}), & \forall p = q \\ 0, & \forall p \neq q \end{cases} \tag{2.48}$$

$$\gamma_{Z_k Z_l}(\boldsymbol{h}) = \sum_{p=0}^{P} b_{p,kl} \Gamma_{X_p}(\boldsymbol{h}) \tag{2.49}$$

$$\boldsymbol{B}_p = \boldsymbol{A}_p \boldsymbol{A}_p^T,\ p = 0, ..., P \tag{2.50}$$

$$\boldsymbol{B}_p = \begin{bmatrix} b_{p,11} & ... & b_{p,1K} \\ ... & & ... \\ b_{p,K1} & ... & b_{p,KK} \end{bmatrix},\ p = 0, ..., P \tag{2.51}$$

$$\boldsymbol{B}_p \boldsymbol{e}_p = \tau_p \boldsymbol{e}_p \rightarrow \tau_p > 0,\ p = 0, ..., P \tag{2.52}$$

$$\begin{cases} \begin{vmatrix} b_{p,11} & ... & b_{p,1K} \\ ... & & ... \\ b_{p,K1} & ... & b_{p,KK} \end{vmatrix} \geq 0,\ p = 0, ..., P \\ b_{p,kl} \geq 0,\ \forall k = l,\ p = 0, ..., P,\ k, l = 1, ..., K \end{cases} \tag{2.53}$$

$$\sigma^2_{Z_k Z_l} = \sum_{p=0}^{P} b_{p,kl} \tag{2.54}$$

The licit forms of the variogram models $\Gamma_{X_p}(\boldsymbol{h})$ for the geostatistical inference are restricted to particular forms. The most common licit variogram models are nugget $\Gamma^0(\boldsymbol{h})$, spherical $\Gamma^1(\boldsymbol{h})$, exponential $\Gamma^2(\boldsymbol{h})$, and Gaussian $\Gamma^3(\boldsymbol{h})$ variogram models. The ananlytical expressions of these four fundamental variogram models are presented in Eqs. (2.55) – (2.59). Here $\boldsymbol{r}$ is the variogram range vector, beyond which spatial correlation disappears. The variogram range is also defined as a distance on the variogram plot, at which variogram function reaches (or almost reaches) sill of the variogram (the variance of the random function). Corresponding covariance functions are denoted as $\Phi^0(\boldsymbol{h})$, $\Phi^1(\boldsymbol{h})$, $\Phi^2(\boldsymbol{h})$, and $\Phi^3(\boldsymbol{h})$, which are related to the variogram types as shown above in Eq. (2.26). Plots of the four basic variograms models and covariance functions are presented

in Fig. 2.3. Conventionally, the nugget effect variogram model takes the index $p = 0$. Both nugget effect variogram model and nugget effect covariance function can be denoted as $Nugget(\boldsymbol{h})$, spherical – $Sph(\boldsymbol{h})$, exponential – $Exp(\boldsymbol{h})$, and Gaussian – $Gaus(\boldsymbol{h})$.

$$\Gamma^0(\boldsymbol{h}) = \begin{cases} 0, & \boldsymbol{h} = 0 \\ 1, & \boldsymbol{h} \neq 0 \end{cases} \tag{2.55}$$

$$\Gamma^1(\boldsymbol{h}) = \begin{cases} 1.5\frac{\boldsymbol{h}}{\boldsymbol{r}} - 0.5\left(\frac{\boldsymbol{h}}{\boldsymbol{r}}\right)^3, & \frac{\boldsymbol{h}}{\boldsymbol{r}} \leq 1 \\ 1, & \frac{\boldsymbol{h}}{\boldsymbol{r}} > 1 \end{cases} \tag{2.56}$$

$$\Gamma^2(\boldsymbol{h}) = 1 - exp\left(-3\frac{\boldsymbol{h}}{\boldsymbol{r}}\right) \tag{2.57}$$

$$\Gamma^3(\boldsymbol{h}) = 1 - exp\left(-3\left(\frac{\boldsymbol{h}}{\boldsymbol{r}}\right)^2\right) \tag{2.58}$$

$$\frac{\boldsymbol{h}}{\boldsymbol{r}} = \sqrt{\left(\frac{h_1}{r_1}\right)^2 + \left(\frac{h_2}{r_2}\right)^2 + \left(\frac{h_3}{r_3}\right)^2} \tag{2.59}$$

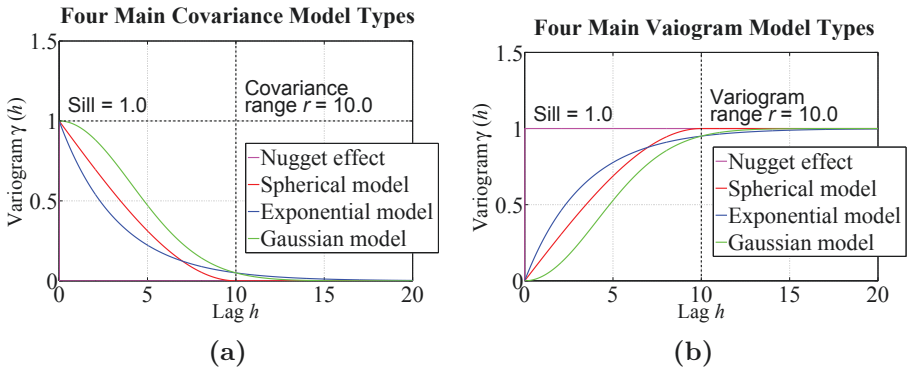


Figure 2.3: Four main spatial structure model types – nugget effect, spherical, exponential, and Gaussian models shown as (a) covariance functions, and (b) variogram models.

In Eq. (2.59), the separation distances h_1, h_2, and h_3 are projections of the vector $\boldsymbol{h} = (h_X, h_Y, h_Z)$ on orthogonal axes r_1, r_2, and r_3 of the anisotropic vector field $\boldsymbol{r}$. The variogram ranges r_1, r_2, and r_3 correspond to three principal orthogonal directions of the continuity. Because the principal directions of anisotropy are orthogonal, the original X, Y, and Z axes of the coordinate system could be rotated to match r_1, r_2, and r_3. By doing so, the rotated coordinates of the lag vector $\boldsymbol{h}$ become projections h_1, h_2, and h_3. Directional angle of the major continuity r_1 is measured between Y axis and r_1 and is called azimuth angle α_a (rotation around Z axis). Other two angles are called dip angle α_d and plunge angle α_p, where rotation happens around Y and X axis, respectively [17]. The schematic of the rotation is presented in Fig. 2.4. The projections h_1, h_2, and h_3 can be computed as shown in Eqs. (2.60) – (2.63) from coordinates h_X, h_Y, h_Z. Matrices $\boldsymbol{M}_{R_X}$, $\boldsymbol{M}_{R_Y}$, and $\boldsymbol{M}_{R_Z}$ are the rotation matrices around X, Y, and Z axes, respectively. In the isotropic case, ratio $\boldsymbol{h}/\boldsymbol{r}$ becomes scalar $—h—/r$.

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \boldsymbol{M}_{R_X} \boldsymbol{M}_{R_Y} \boldsymbol{M}_{R_Z} \begin{bmatrix} h_X \\ h_Y \\ h_Z \end{bmatrix} \tag{2.60}$$

$$\boldsymbol{M}_{R_X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_p & \sin\alpha_p \\ 0 & -\sin\alpha_p & \cos\alpha_p \end{bmatrix} \tag{2.61}$$

$$\boldsymbol{M}_{R_Y} = \begin{bmatrix} \cos\alpha_d & 0 & \sin\alpha_d \\ 0 & 1 & 0 \\ -\sin\alpha_d & 0 & \cos\alpha_d \end{bmatrix} \tag{2.62}$$

$$\boldsymbol{M}_{R_Z} = \begin{bmatrix} \cos\alpha_a & -\sin\alpha_a & 0 \\ \sin\alpha_a & \cos\alpha_a & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.63}$$

Once the variogram model is fitted, the geostatistical estimation and simulation methods can be carried out to estimate properties of the natural phenomena at unsampled locations. Estimation methods produce a unique smooth model. Simulation methods generate several models or realizations of the unknown truth, which depict more realistic local variations.

2.2 Estimation

Geostatistical estimation is the general term for spatial interpolation methods, which define unknown value of a random function at an unsampled location using relevant data. There are two general types of the random functions dealt with: continuous and categorical. The estimation methods are slightly different for each type.

2.2.1 Continuous Variables

Kriging is a conventional geostatistical method widely used for the estimation of properties of natural phenomena [36]. Simple kriging (SK) and simple cokriging (SCK) estimate univariate and multivariate systems. They are spatial interpolation techniques that are sometimes called best linear unbiased estimators (BLUE). The mean of the local estimates is unbiased, that is the expectation of the estimate equals the expectation of the variable it seeks to estimate, while local variance of the estimation error is minimized. Kriging produces an estimate conditional to the available primary data values. In cokriging, a primary variable

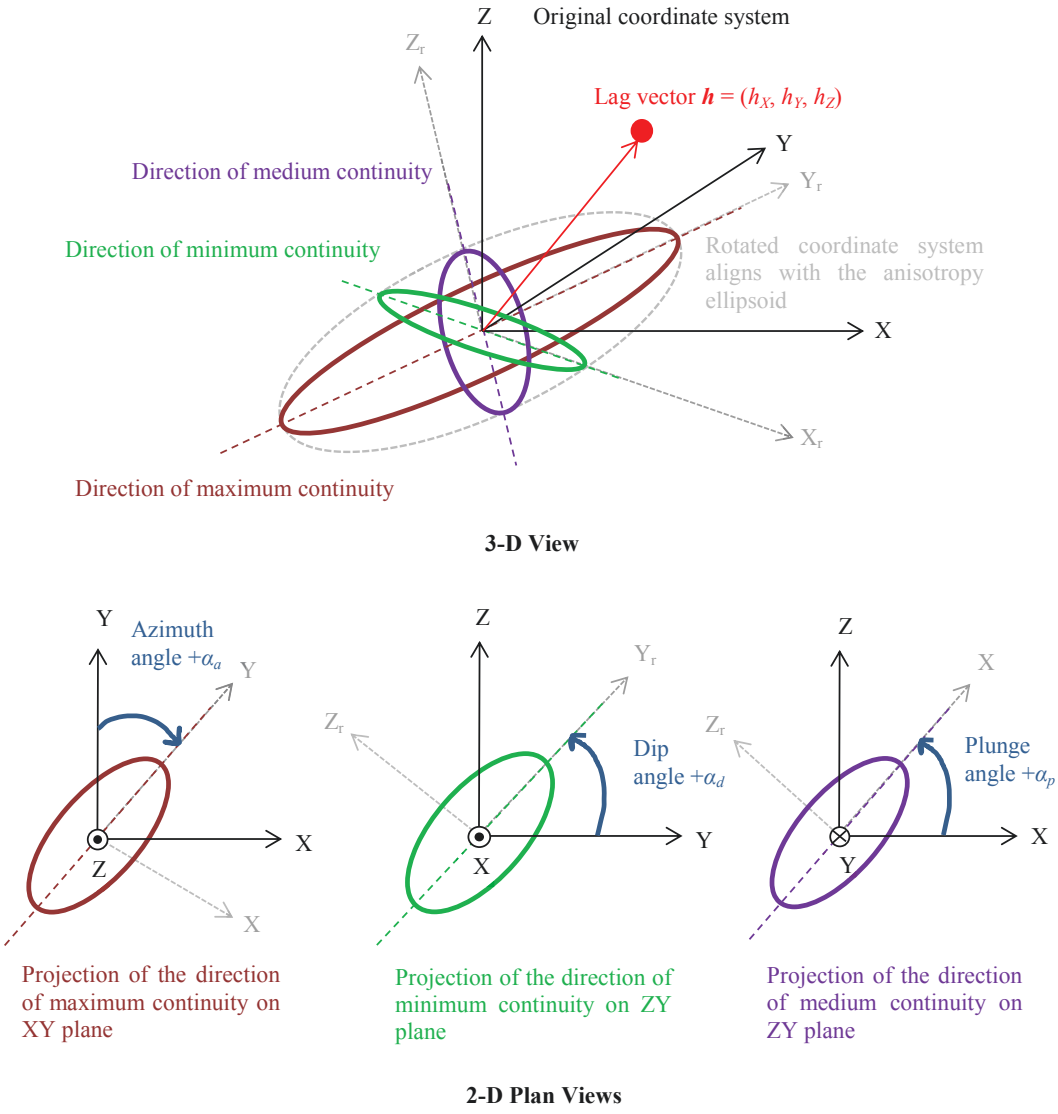


Figure 2.4: Schematic of projection of the lag vector $\boldsymbol{h}$ on the principal directions $\boldsymbol{r}$ of anisotropy.

is estimated conditional to primary data and relevant secondary data of correlated secondary variables. An assumption of first- and second-order stationarity is crucial in both techniques. Simple kriging is explained first, which is followed by simple cokriging.

The simple kriging equations take form shown in Eq. (2.64), where $Z^*(\boldsymbol{u})$ is the estimate of random function Z at location $\boldsymbol{u}$, $\lambda_\alpha(\boldsymbol{u})$ is the kriging weight assigned to a datum of random function Z at location $\boldsymbol{u}_\alpha$ to compute the estimate of the random function at location $\boldsymbol{u}$, and N is the number of data [10, 17, 26, 36]. Note that simple kriging estimate is unbiased (Eq. (2.65)). The kriging weights are found from the minimization of the variance of the estimation error ε. The error itself is location-dependent and defined as the difference between an estimate and the truth at the estimation location $\boldsymbol{u}$ in Eq. (2.66). The variance of estimation error $\sigma^2_\varepsilon(\boldsymbol{u})$ is presented as an expected value of the square difference between an estimate and the truth at the estimation location $\boldsymbol{u}$ as shown in Eq. (2.67), from which kriging weights are derived by finding local minima through the partial derivatives. This optimized estimation error variance is also called simple kriging variance. The system of kriging equations for simple kriging as shown in Eq. (2.68) is the resulting solution to the estimation error variance minimization. The kriging weights are derived from this system of kriging equations. Covariances on the left-hand side are between data locations, and covariances on the right-hand side are between data and the unsampled location. The spatial structure of the modeling system in terms of the covariance or variogram function strongly influences the solution. Kriging assesses the uncertainty in the estimate by the minimized estimation error variance (kriging variance), which takes form shown in Eq. (2.69) [36].

$$Z^*(\boldsymbol{u}) = m_Z(\boldsymbol{u}) + \sum_{\alpha=1}^{N} \lambda_\alpha(\boldsymbol{u}) \left[Z(\boldsymbol{u}_\alpha) - m_Z(\boldsymbol{u}_\alpha) \right] \tag{2.64}$$

$$E\{Z^*(\boldsymbol{u})\} = m_Z(\boldsymbol{u}) \tag{2.65}$$

$$\varepsilon(\boldsymbol{u}) = Z^*(\boldsymbol{u}) - Z(\boldsymbol{u}) \tag{2.66}$$

$$\sigma_\varepsilon^2(\boldsymbol{u}) = E\{(Z^*(\boldsymbol{u}) - Z(\boldsymbol{u}))^2\} \;\Rightarrow \min \sigma_\varepsilon^2(\boldsymbol{u}) \;\Rightarrow \lambda(\boldsymbol{u}) \tag{2.67}$$

$$\sum_{\alpha=1}^{N} C_Z(\boldsymbol{u}_\alpha - \boldsymbol{u}_{\alpha'})\lambda_\alpha(\boldsymbol{u}) = C_Z(\boldsymbol{u} - \boldsymbol{u}_{\alpha'}),\ \alpha' = 1, ..., N \tag{2.68}$$

$$\sigma_{\varepsilon^*}^2(\boldsymbol{u}) = \sigma_Z^2 - \sum_{\alpha=1}^{N} C_Z(\boldsymbol{u} - \boldsymbol{u}_\alpha)\lambda_\alpha(\boldsymbol{u}) \tag{2.69}$$

Simple cokriging is presented in Eq. (2.70), where $Z_k^*(\boldsymbol{u})$ is the estimate of random function Z_k at location $\boldsymbol{u}$, $\lambda_{k,\alpha_l}(\boldsymbol{u})$ is the kriging weight assigned to a datum of random function Z_l at location $\boldsymbol{u}_\alpha$ to compute estimate of the random function Z_k at location $\boldsymbol{u}$, N_l is the number of data of random function Z_l, and K is the total number of random functions. Kriging weights are computed from the system of cokriging equations presented in Eq. (2.71). The minimized estimation error variance is presented in Eq. (2.72).

$$Z_k^*(\boldsymbol{u}) = m_{Z_k}(\boldsymbol{u}) + \sum_{l=1}^{K}\sum_{\alpha_l=1}^{N_l} \lambda_{k,\alpha_l}(\boldsymbol{u})\left[Z_l(\boldsymbol{u}_\alpha) - m_{Z_l}(\boldsymbol{u}_\alpha)\right],\ k = 1, ..., K \tag{2.70}$$

$$\begin{gathered}\sum_{l=1}^{K}\sum_{\alpha_l=1}^{N_l} \lambda_{k,\alpha_l}(\boldsymbol{u}) C_{Z_l Z_m}(\boldsymbol{u}_{\alpha_l} - \boldsymbol{u}_{\alpha'_m}) = C_{Z_k Z_m}(\boldsymbol{u} - \boldsymbol{u}_{\alpha'_m}), \\ \alpha'_m = 1, ..., N_m,\ m = 1, ..., K,\ k = 1, ..., K\end{gathered} \tag{2.71}$$

$$\sigma^2_{\varepsilon^*,k}(\boldsymbol{u}) = \sigma^2_{Z_k} - \sum_{l=1}^{K}\sum_{\alpha_l=1}^{N_l} \lambda_{k,\alpha_l}(\boldsymbol{u}) C_{Z_k Z_l}(\boldsymbol{u} - \boldsymbol{u}_{\alpha_l}),\ k = 1, ..., K \tag{2.72}$$

Therefore, simple kriging and cokriging provide an unique estimation model of the studied property along with the uncertainty in the estimate based on the available relevant data and variogram model.

2.2.2 Categorical Variables

As stated several times, the assumption of the stationarity is a central idea in the geostatistical modeling [36]. A non-stationary setting can sometimes be addressed by dividing the original domain into several stationary domains, which are usually modeled with categorical variable modeling techniques.

Kriging can be extended to categorical random functions. Each random function Z_k out of K random functions corresponds to a category, which is presented by an indicator i_k [10, 17, 26]. Indicators $i_k(\boldsymbol{u}), k = 1, ..., K$, are similar to probabilities and indicate a proportion of times the particular category would be present at location $\boldsymbol{u}$. Values of the indicators vary between 0 and 1. Indicator of zero implies that particular category could never present at a location, while an indicator of one indicates the certain presence of a category at a location. The mathematical definition of the indicators is shown in Eq. (2.73) with their properties summarized in Eq. (2.74). Probabilities $p_k(\boldsymbol{u}), k = 1, ..., K$, are means (expected values) of the indicators $i_k(\boldsymbol{u}), k = 1, ..., K$. Definition of the probabilities is shown in Eq. (2.75) with properties in Eq. (2.76). Variance $\sigma^2_k(\boldsymbol{u})$ of the indicators is defined similar to the continuous variables with different outcome presented in Eq. (2.77).

$$i_k(\boldsymbol{u}) = \begin{cases} 1, \text{if category } k \text{ is present at location } \boldsymbol{u} \\ 0, \text{if category } k \text{ is missing at location } \boldsymbol{u} \end{cases}, \; k = 1, ..., K \tag{2.73}$$

$$\begin{cases} 0 \leq i_k(\boldsymbol{u}) \leq 1, \; k = 1, ..., K \\ \sum_{k=1}^{K} i_k(\boldsymbol{u}) = 1 \end{cases} \tag{2.74}$$

$$p_k(\boldsymbol{u}) = E\{i_k(\boldsymbol{u})\}, \; k = 1, ..., K \tag{2.75}$$

$$\begin{cases} 0 \leq p_k(\boldsymbol{u}) \leq 1, \; k = 1, ..., K \\ \sum_{k=1}^{K} p_k(\boldsymbol{u}) = 1 \end{cases} \tag{2.76}$$

$$\sigma_k^2(\boldsymbol{u}) = E\{(i_k(\boldsymbol{u}) - p_k(\boldsymbol{u}))^2\} = p_k(\boldsymbol{u})(1 - p_k(\boldsymbol{u})), \; k = 1, ..., K \tag{2.77}$$

The indicators are estimated at unsampled locations $\boldsymbol{u}$ similar to the simple kriging estimation of continuous variables. The estimated indicator $i_k^*(\boldsymbol{u})$ is computed as in Eq. (2.78). A system of kriging equations (2.79) is required to derive kriging weights $\lambda_{k,\alpha}(\boldsymbol{u})$ for each indicator data location $\boldsymbol{u}_\alpha, \alpha = 1, ..., N$, to compute estimate $i_k^*(\boldsymbol{u})$ at unsampled location $\boldsymbol{u}$. The covariance functions $C_k(\boldsymbol{h}), k = 1, ..., K$, of the indicators are computed from the fitted variogram models in the same way as it has been explained for the continuous variables. Because the estimate of each category is computed independently, it may happen that the indicator estimates do not satisfy the required properties in Eq. (2.74). For this reason, the estimates are reset to $i_k'^*(\boldsymbol{u})$ according to Eq. (2.80).

$$i_k^*(\boldsymbol{u}) = p_k(\boldsymbol{u}) + \sum_{\alpha=1}^{N} \lambda_{k,\alpha}(\boldsymbol{u}) \left[i_k(\boldsymbol{u}_\alpha) - p_k(\boldsymbol{u}_\alpha) \right], \; k = 1, ..., K \tag{2.78}$$

$$\sum_{\alpha=1}^{N} \lambda_{k,\alpha}(\boldsymbol{u}) C_k(\boldsymbol{u}_\alpha - \boldsymbol{u}_{\alpha'}) = C_k(\boldsymbol{u} - \boldsymbol{u}_{\alpha'}), \; \alpha' = 1, ..., N, \; k = 1, ..., K \tag{2.79}$$

$$\begin{cases} i_k^*(\boldsymbol{u}) = 0, \text{if } i_k^*(\boldsymbol{u}) < 0 \\ i_k^*(\boldsymbol{u}) = 1, \text{if } i_k^*(\boldsymbol{u}) > 1 \\ i_k'^*(\boldsymbol{u}) = \frac{i_k^*(\boldsymbol{u})}{\sum_{l=1}^{K} i_l^*(\boldsymbol{u})}, \; \forall \boldsymbol{u} \in A \end{cases}, \; k = 1, ..., K \tag{2.80}$$

The categories can be drawn from the estimated corresponding indicators, what leads to the simulation method explained later. Note that indicator kriging utilizes variogram models, and, therefore, is based on a two-point statistic.

2.3 Simulation

Estimation methods produce a unique smooth models that do not represent realistic short scale variability [36]. Simulation methods have been developed to generate a set of equally likely to be drawn stochastic realizations of the reality with the expected degree of local variation. Uncertainty is assessed by variations between the realizations. The realizations can be processed through the transfer function to get characteristics of the system.

The normal or Gaussian distribution is very important in most simulation algorithms. Simulated variables of a stationary random process tend to follow normal distribution because of the central limit theorem (CLT). It states that averages or sums of large number of independent equally distributed variables

would follow a normal distribution [19]. Some simulation methods linearly combine values and lead to results that are normally distributed. Other simulation methods take advantage of the unique mathematical properties of the multivariate normal distribution. Therefore, the simulation is almost always performed in normal units. The random function that follows a normal distribution is denoted as Y. The normal or Gaussian distribution is a parametric distribution, where the mean and standard deviation fully describe the entire distribution as shown in Eq. (2.81) through the PDF $g_{Y(\boldsymbol{u})}(y)$ for a single random variable $Y(\boldsymbol{u})$. One of the unique properties of the multivariate normal distribution is that all conditional and marginal distributions are also normal.

$$g_{Y(\boldsymbol{u})}(y, m_Y(\boldsymbol{u}), \sigma_Y(\boldsymbol{u})) = \frac{1}{\sqrt{2\pi}\sigma_Y(\boldsymbol{u})} \exp\left(-\frac{(y - m_Y(\boldsymbol{u}))^2}{2\sigma_Y^2(\boldsymbol{u})}\right) \tag{2.81}$$

The normal score transformation (NST) can be applied to convert random function Z, which follows arbitrary distribution $F_{Z(\boldsymbol{u})}(z)$, to a random function Y, which follows strictly the normal distribution $G_{Y(\boldsymbol{u})}(y)$. The unbiased data distribution of the random function Z in original units should be used in the transformation [17]. The normal score transformation of the univariate system can be presented as a nonlinear quantile-quantile transform of variable distribution as shown in Eq. (2.82). A graphical representation is shown in Fig. 2.5. It is generally assumed that the normal scores of random function Y are stationary and follow normal distribution with a zero mean and unit variance $Y \sim N(0,1)$ [36].

$$y = G_{Y(\boldsymbol{u})}^{-1}[F_{Z(\boldsymbol{u})}(z)] \tag{2.82}$$

The NST can be applied to random functions of the multivariate system

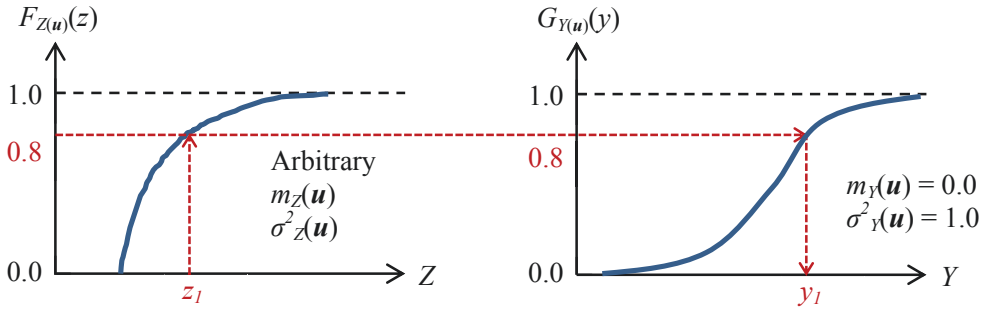


Figure 2.5: Graphical representation of the normal score transformation.

individually [42]. The multivariate distribution would then be assumed multivariate Gaussian. The NST is not a linear transform, thus only quantiles can be back transformed (not means and variances).

2.3.1 Continuous Variables

The theory of the common geostatistical simulation methods for modeling continuous properties of the natural phenomena is reviewed in this subsection. These simulation techniques are assessed for their implementation in a grid-free form. Current developments of the grid-free conditional simulation are also highlighted.

Despite large abundance of the geostatistical simulation techniques, most methods are somehow grid dependent from an implementation perspective. The conventional conditional and unconditional geostatistical simulation techniques are designed in a way that takes advantage of a grid for the simulation [26]. Moreover, most techniques are not expressed as a function of the coordinates of a simulation location and could not be easily refined to higher resolution in current computational implementation [36].

The simulation methods of continuous random functions can be divided

into two groups: direct conditional simulations and two-step conditional simulations [40]. Sequential Gaussian simulation (SGS) and simulated annealing (SA) generate conditional realizations directly [16, 64]. LU simulation (LUS), moving average (MA), spectral (SS) and fractal simulations (FS) generate unconditional realizations, which are later conditioned to data values. Note that the LUS can be also implemented to generate conditional realizations [14]. Virtually all simulation algorithms are implemented on a grid despite the fact that many are not intrinsically limited to gridded simulation [17, 64]. The grid type varies from regularly structured grids, where the modeling domain is discretized into a regular configuration of grid nodes organized according to some defined structure, to irregular unstructured grids, where the modeling domain is divided into nodes irregularly distributed in the space [45]. Turning bands simulation (TBS) method is a special algorithm that combines 1-D simulations into simulation of higher dimensional systems. It is possible to express the realization as a function of the coordinates with the TBS. The TBS method is discussed in Chapter 3. All simulation algorithms can generate univariate and multivariate simulations. Simulation is performed at the scale of the data referred to as the point scale. Point-scale values can be brought to a larger scale by a regularization operator or by some means of upscaling [36, 78]. Both approaches average point-scale spatial properties of the variable within some volume associated with larger scale. Such averaging is unique for additive variables such as rock porosity or mineral grade.

Direct conditional simulation techniques are extensively used in practice for modeling continuous properties. Conditioning realizations to data is performed simultaneously in the simulation algorithm. The concept of Monte Carlo simulation (MCS) is utilized in all simulation methods [51]. A principle of MCS states that simulated values can be drawn from the local distribution of the simulated variable at the simulation location, where the local distribution is characterized by conditional mean and variance. A random number between values 0 and

1 is drawn. The corresponding quantile value from the estimated CDF would correspond to the simulated value of the random variable.

The sequential Gaussian simulation (SGS) is widely used geostatistical simulation technique for modeling continuous variables [17, 64, 65]. SGS is based on a sequential implementation of the Bayesian law of conditional probabilities. The core of the simulation engine is shown in Eq. (2.83), where N is the number of the conditioning data and M is the number of the simulation nodes [17]. The simulation of the continuous variables is performed in the normal space Y. Therefore, the NST is performed before and after the simulation. After NST, a regular grid is established and a simulation path is defined randomly. The simulated values are drawn randomly from the estimated local distributions at the nodes. These local distributions are derived from the simple kriging mean and variance calculated conditional to data values and previously simulated values taken within some search neighborhood using Eqs. (2.64) and (2.69) for univariate case. The simulated values are back transformed to the original space once all simulation nodes have been visited. The random component does not depend on the location of the simulation node, but rather on the order of simulation node. Thus, each simulated realization is not represented just as a function of the coordinates. The realization at a location would change with any grid parameter changes. The SGS realizations could easily be refined using previously simulated values as conditioning data. The drawback of such approach is that all previously simulated values should be stored in the memory, because SGS realizations are not expressed as a function of the coordinates of the simulation locations. One concern with SGS is that the variograms of the realizations may not reproduce the target variogram model at longer ranges [21].

$$\begin{aligned}\text{Prob}\{Y_1 \leq y_1, ..., Y_M \leq y_M\} = &\text{Prob}\{Y_1 \leq y_1 | (N+M-1)\}... \\ &\text{Prob}\{Y_{M-1} \leq y_{M-1} | (N+1)\} \\ &\text{Prob}\{Y_M \leq y_M | (N)\}\end{aligned} \tag{2.83}$$

Simulated annealing (SA) is an optimization technique that could be adapted to be geostatistical simulation technique [16]. The objective function includes the mismatch between simulated values at the data locations and data values, and minimization of the difference between reproduced and target spatial structures of the system. The simulation is generated iteratively on any grid or a set of simulation nodes. The initial iteration is generated arbitrarily from the variable's distribution. The objective function is computed and compared to some minimal value, at which the simulation terminates. If the objective function is greater than terminating value, some realization values are perturbed. A new objective function is computed and compared with old one and the terminating value. If new objective function is smaller than an old one, new perturbed realization values are accepted. Otherwise, new realization values are accepted or rejected based on some acceptance/rejection strategy controlled by a cooling function. The cooling function tends to accept more perturbation in the beginning of the simulation and less with increasing number of iterations. Other realization values are perturbed in the same fashion until the objective function reaches minimal terminating value. Careful selection of the perturbation algorithm and cooling function is required. The implementation of the SA may be computationally expensive.

The two-step conditional simulation procedures are also common in practice. The choice between direct and two-step simulation methods depends on

the modeler [17, 69]. Two-step conditional simulation algorithms produce conditional realizations in two steps [36]. The data values are also transformed to normal units. First, unconditional realizations $Y_i^{us}(\boldsymbol{u}), i = 1, ..., N_R$, are generated in normal scores with various fast unconditional simulation algorithms. The resulting realizations have the specified spatial structure, but do not honor the data. The unconditional simulation is fast. Second, the unconditional realizations are conditioned by adding kriging estimate $Y^*(\boldsymbol{u})$ conditioned to data in normal scores and subtracting kriging estimate $Y_i^{*|us}(\boldsymbol{u})$ conditioned to unconditionally simulated values of i^{th} realization at the data locations. This step is expressed mathematically in Eq. (2.84) for both univariate and multivariate systems. The equation is applied to every realization i of total N_R realizations. In case of modeling multivariate system, kriging is replaced with cokriging and simulation is replaced with cosimulation.

$$Y_i^s(\boldsymbol{u}) = Y_i^{us}(\boldsymbol{u}) + \left(Y^*(\boldsymbol{u}) - Y_i^{*|us}(\boldsymbol{u})\right), \ \forall \boldsymbol{u} \in A, \ i = 1, ..., N_R \tag{2.84}$$

The LU simulation, moving average, spectral and fractal simulation techniques are geostatistical simulation engines for generation of unconditional realizations [15, 28, 34, 55, 70]. They are applied to any dimensional problem and lead to accurate variogram reproduction.

The LU simulation (LUS) is based on the LU decomposition of the covariance matrix $\boldsymbol{C}_Y$ of the modeled system in the normal space [1, 14]. The entries in the covariance matrix are the covariance values between all simulation nodes. Therefore, the simulation nodes should be defined before the simulation starts. The node locations can be specified arbitrarily, but usually are organized according to a regular grid. The simulated realizations can be refined, but are

not expressed as a function of the simulation node locations. Because the covariance matrix is symmetric, the LU decomposition algorithm is replaced with the faster Cholesky decomposition algorithm shown in Eq. (2.85). Matrices $\boldsymbol{L}_Y$ and $\boldsymbol{U}_Y$ are the lower and upper triangular matrices. The simulated values $\boldsymbol{Y}$ are generated by matrix multiplication of the lower triangular matrix and vector of normally distributed zero-mean unit-variance random values $\boldsymbol{R} \sim N(\boldsymbol{0},\boldsymbol{I})$ presented in Eq. (2.86). The matrix $\boldsymbol{I}$ is the identity matrix with zero elements except diagonal entries, which are one. The random components in $\boldsymbol{R}$ are not controlled through the coordinates. The realizations are generated by drawing different elements for random vector $\boldsymbol{R}$. The resulting simulated values in $\boldsymbol{Y}$ would follow normal distribution with zero mean and unit variance. The LU matrix decomposition becomes intractable and computationally demanding for large models. The method reproduces the variogram well. The unconditional realizations from LU simulation can be elegantly conditioned by kriging in a matrix form [1, 14].

$$\boldsymbol{C}_Y = \boldsymbol{L}_Y \boldsymbol{U}_Y = \boldsymbol{L}_Y \boldsymbol{L}_Y^T \tag{2.85}$$

$$\boldsymbol{Y} = \boldsymbol{L}_Y \boldsymbol{R} \tag{2.86}$$

In the moving average (MA) simulation method, the unconditional simulation $Y^{us}(u)$ of 1-D systems is generated as a convolution of a weighting function $\lambda(u)$ and random Gaussian signal $R(u)$ as shown in Eq. (2.87) [55]. The MA is usually applied together with the turning bands simulation method [36]. Transformation of the covariance function $C_Y(h)$ of the simulated random variable $Y^{us}(u)$ to the frequency space is applied in this technique. The Fourier transform and inverse Fourier transform, which form the Fourier transform pair of a continuous function f, can be calculated as shown in Eqs. (2.88) and (2.89),

where i is the imaginary unit of the complex number [7, 72]. An elegant property of the Fourier transform is that convolution in the space domain translates to a multiplication in the frequency domain [8]. Because the relationship between the covariance $C_Y(h)$ and weighting function $\lambda(u)$ in the space domain is expressed through a convolution of the weighting function and its reverse $\breve{\lambda}(u)$ as in Eq. (2.90), the weighting function $\hat{\lambda}(\omega)$ in the frequency domain is derived from the Fourier transform $\hat{C}_Y(\omega)$ of the target covariance function as presented in Eq. (2.91). The Fourier transform of the weighting function is transformed back to original space with the inverse Fourier transform as presented in Eq. (2.89). The expression for the weighting function can be easily derived analytically for most covariance functions. Forms of the weighting functions for the most common covariance functions are summarized in [55] and can be extended to any other licit covariance function models. The weighting function $\lambda(u)$ is symmetric about the origin. The random signal $R(u)$ has a pure nugget effect spatial structure. It is not represented as a function of the coordinate. The convolution operator $[\lambda \star R](u)$ is discretized through the summation in the numerical implementation. Therefore, the moving average simulation depends on the grid.

$$Y^{us}(u) = [\lambda \star R](u) = \int_{-\infty}^{\infty} \lambda(u - u')R(u')du' \tag{2.87}$$

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(h)\exp(-i\omega h)dh \tag{2.88}$$

$$f(h) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)\exp(i\omega h)d\omega \tag{2.89}$$

$$C_Y(h) = [\lambda \star \breve{\lambda}](h) = \int_{-\infty}^{\infty} \lambda(h - h')\lambda(h')dh' \tag{2.90}$$

$$\hat{\lambda}(\omega) = \sqrt{\hat{C}_Y(\omega)} \tag{2.91}$$

Spectral simulation (SS) is an important class of fast unconditional simulation techniques. Fractal simulation is a special case of spectral simulation with a particular covariance function. The SS methods are global techniques in a sense that they assume stationarity over entire modeling area. The key idea of the SS is the reciprocal transition between space and frequency domains. Any continuous function f in Eqs. (2.88) and (2.89) is discretized into M equally spaced nodes in both space and frequency domains. The resulting Fourier transform pair is shown in Eqs. (2.92) and (2.93) for 1-D case. Note that the frequency domain is periodic and, thus, frequency domain can be completely characterized in $[0,\, 2\pi)$ interval. This class of simulation methods often uses the fast Fourier transform (FFT) algorithm to compute the Fourier transform pair [8, 28, 66]. The FFT is grid dependent and requires a grid size that is a multiple of 2. An example of Cooley-Tukey FFT of the function from space to frequency domain is presented in Eqs. (2.94) – (2.96), where $\hat{f}^e(\omega_m)$ and $\hat{f}^o(\omega_m)$ denote even and odd parts of the $\hat{f}(\omega_m)$ [11]. The computational time reduces from M^2 to $M\log(M)$ for the transformation of entire function f by recursively applying FFT.

$$\hat{f}(\omega_m) = \sum_{q=1}^{M} f(u_q) \exp\left(-i2\pi \frac{(m-1)(q-1)}{M}\right) \tag{2.92}$$

$$f(u_q) = \frac{1}{M} \sum_{m=1}^{M} \hat{f}(\omega_m) \exp\left(i2\pi \frac{(m-1)(q-1)}{M}\right) \tag{2.93}$$

$$\begin{cases} \hat{f}(\omega_m) = \hat{f}^e(\omega_m) + \exp\left(-i2\pi \frac{(m-1)}{M}\right) \hat{f}^o(\omega_m) \\ \hat{f}(\omega_{m+M/2}) = \hat{f}^e(\omega_m) - \exp\left(-i2\pi \frac{(m-1)}{M}\right) \hat{f}^o(\omega_m) \end{cases}, \; m = 1, ..., M/2 \tag{2.94}$$

$$\hat{f}^e(\omega_m) = \sum_{q=1}^{M/2} f(u_{2q}) \exp\left(-i2\pi \frac{(m-1)(q-1)}{M/2}\right) \tag{2.95}$$

$$\hat{f}^o(\omega_m) = \sum_{q=1}^{M/2} f(u_{2q-1}) \exp\left(-i2\pi \frac{(m-1)(q-1)}{M/2}\right) \tag{2.96}$$

Spectral simulation is well researched, but not implemented extensively for geostatistical applications [17]. The origin of SS is electrical engineering. One of first papers on SS proposed a technique for the generation of multidimensional multivariate anisotropic realizations without reliance on the FFT [71]. Isotropic univariate random function $Y^{us}(\boldsymbol{u})$ can be generated unconditionally according to Eq. (2.97). Here $\boldsymbol{\omega_m}$ is the discrete frequency vector for μ-dimensional system; $\boldsymbol{\omega'_m}$ is the discrete frequency vector with some small random value introduced to avoid the periodicity in the simulated function $Y^{us}(\boldsymbol{u})$; $\Delta\boldsymbol{\omega}$ is the vector of frequency increments for all dimensions μ; $\phi(\boldsymbol{\omega_m})$ is the random phase independently and uniformly distributed between 0 and 2π; and $\hat{C}_Y(\boldsymbol{\omega_m})$ is the spectral density of the covariance function $C_Y(\boldsymbol{h})$ computed discretely as shown in Eq. (2.98). The discretization indices $\boldsymbol{m}$ and $\boldsymbol{q}$ include indices over high dimensional domain, i.e., $\boldsymbol{m} = [m_1 \ ... \ m_\mu]^T$ and $\boldsymbol{q} = [q_1 \ ... \ q_\mu]^T$. The total number of the discretization points $\boldsymbol{M}$ has M^μ components. An anisotropic random function can be simulated according to Eq. (2.99), where $A(\boldsymbol{u}, \boldsymbol{\omega_m})$ is the deterministic function that addresses anisotropy of the modeled system. Isotropic multiple correlated random functions can be generated according to Eq. (2.100). $L_{kl}(\boldsymbol{\omega_m})$ is the element of the lower triangular matrix $\boldsymbol{L}(\boldsymbol{\omega_m})$ computed from the Cholesky decomposition of the spectral density matrix $\hat{\boldsymbol{C}}_Y(\boldsymbol{\omega_m})$ as shown in Eq. (2.101). θ_{kl} is the additional phase shift attributed to the multivariate case. Its expression is shown in Eq. (2.102). In univariate isotropic, univariate anisotropic, and multivariate cases, the unconditional simulation $Y^{us}(\boldsymbol{u})$ is proportional to a sum

of the product of the square root of the spectral density and cosine function of a particular frequency. The simulated realization can be expressed as a function of the coordinates of simulation location $\boldsymbol{u}$ allowing zooming in and out the process, which is briefly described in [70]. The major concern with this technique is an occurrence of the periodicity in the reproduced variograms of the simulations [59, 60].

$$Y^{us}(\boldsymbol{u}) = \sum_{\boldsymbol{m}=1}^{\boldsymbol{M}} \sqrt{\hat{C}_Y(\boldsymbol{\omega_m}) 2\Delta\boldsymbol{\omega}} \cos(\boldsymbol{\omega}'_{\boldsymbol{m}}\boldsymbol{u} + \phi(\boldsymbol{\omega_m})) \tag{2.97}$$

$$\hat{C}_Y(\boldsymbol{\omega_m}) = \frac{1}{M^{\mu}} \sum_{\boldsymbol{q}=1}^{\boldsymbol{M}} C_Y(\boldsymbol{h_q}) \cos(\boldsymbol{\omega_m h_q}) \tag{2.98}$$

$$Y^{us}(\boldsymbol{u}) = \sum_{\boldsymbol{m}=1}^{\boldsymbol{M}} A(\boldsymbol{u}, \boldsymbol{\omega_m}) \sqrt{\hat{C}_Y(\boldsymbol{\omega_m}) 2\Delta\boldsymbol{\omega}} \cos(\boldsymbol{\omega}'_{\boldsymbol{m}}\boldsymbol{u} + \phi(\boldsymbol{\omega_m})) \tag{2.99}$$

$$Y^{us}_k(\boldsymbol{u}) = \sum_{l=1}^{k} \sum_{\boldsymbol{m}=1}^{\boldsymbol{M}} |L_{Y,kl}(\boldsymbol{\omega_m})| \sqrt{2\Delta\boldsymbol{\omega}} \cos(\boldsymbol{\omega}'_{\boldsymbol{m}}\boldsymbol{u} + \theta_{kl}(\boldsymbol{\omega_m}) + \phi_l(\boldsymbol{\omega_m})), \quad k = 1, ..., K \tag{2.100}$$

$$\hat{\boldsymbol{C}}_Y(\boldsymbol{\omega_m}) = \boldsymbol{L}_Y(\boldsymbol{\omega_m}) \boldsymbol{L}_Y^T(\boldsymbol{\omega_m}),\ L_{Y,kl}(\boldsymbol{\omega_m}) \in \boldsymbol{L}_{Y,kl}(\boldsymbol{\omega_m}),\ k, l = 1, ..., K \tag{2.101}$$

$$\theta_{kl}(\boldsymbol{\omega_m}) = \tan^{-1}\left(\frac{\Im L_{kl}(\boldsymbol{\omega_m})}{\Re L_{kl}(\boldsymbol{\omega_m})}\right),\ k, l = 1, ..., K \tag{2.102}$$

A similar frequency-based approach to the simulation of unconditional Gaussian fields in any dimensional space is the Fourier integral method (FIM),

which does not show significant periodicity in the reproduced experimental variograms of the simulations [58]. The gridded realization $Y^{us}(\boldsymbol{u_q}), \boldsymbol{q} = 1, ..., \boldsymbol{M}$, in higher dimensional space is generated on a regular grid by means of the FFT, where discrete spectral density function $\hat{C}_Y(\boldsymbol{\omega_m})$ represents the square magnitude of the Fourier transform of the simulation as shown in Eq. (2.103). This simulation is possible, because the relationship between random function and covariance can be expressed in a convolution form as in Eq. (2.104), what becomes Eq. (2.105) in a frequency domain [82]. The spectral density is computed according to Eq. (2.106). The phase $\phi(\boldsymbol{\omega_m})$ is added randomly to introduce the stochastic feature. The FIM allows generating realizations with any licit anisotropic variogram models.

$$Y^{us}(\boldsymbol{u_q}) = \sum_{\boldsymbol{m}=1}^{\boldsymbol{M}} \sqrt{\hat{C}_Y(\boldsymbol{\omega_m})} \exp\left(i2\pi \frac{(\boldsymbol{m}-1)(\boldsymbol{q}-1)}{\boldsymbol{M}} + \phi(\boldsymbol{\omega_m})\right) \tag{2.103}$$

$$C_Y(\boldsymbol{h}) = [Y \star \breve{Y}](\boldsymbol{h}) = \int_{-\infty}^{\infty} Y(\boldsymbol{h}-\boldsymbol{h}')Y(\boldsymbol{h}')d\boldsymbol{h}' \tag{2.104}$$

$$\hat{C}_Y(\boldsymbol{\omega}) = \hat{Y}(\boldsymbol{\omega})\hat{Y}(-\boldsymbol{\omega}) = |\hat{Y}(\boldsymbol{\omega})|^2 \tag{2.105}$$

$$\hat{C}_Y(\boldsymbol{\omega_m}) = \frac{1}{M^{\mu}} \sum_{\boldsymbol{q}=1}^{\boldsymbol{M}} C_Y(\boldsymbol{h_q}) \exp\left(-i2\pi \frac{(\boldsymbol{m}-1)(\boldsymbol{q}-1)}{\boldsymbol{M}}\right) \tag{2.106}$$

Multivariate simulation is possible with the FIM [59]. First, the spectral density matrix $\hat{\boldsymbol{C}}_Y$ is defined by applying Fourier transform to the target direct- and cross-covariances $\boldsymbol{C}_Y$ obtained from the experimental variograms of the simulated random functions. The covariance matrix $\boldsymbol{C}_Y$ has to be positive

definite and may be corrected by setting negative eigenvalues of the matrix to zero. The eigenvalues can be derived as shown in Eq. (2.107), where τ is the possible eigenvalue solution to the equation and $\boldsymbol{e}$ is the eigenvector [67]. Second, Cholesky decomposition of the spectral density matrix $\hat{\boldsymbol{C}}_Y$ is performed. Each row of the lower triangular matrix represents the set of modules of the spectral amplitude functions. Third, each variable $\hat{Y}_k^{us}(\boldsymbol{\omega_m}), k = 1, ..., K$, is simulated in the frequency domain by adding the spectral amplitude modules from the corresponding row of the lower matrix multiplied by a complex exponent of the random phase. Last, the simulated values $\hat{Y}_k^{us}(\boldsymbol{\omega_m}), k = 1, ..., K$, in the frequency domain are transformed back to the original units $Y_k^{us}(\boldsymbol{u_q}), k = 1, ..., K$, through the inverse Fourier transformation. All direct and cross terms of the covariance matrix are reproduced. Some corrections to get strictly Gaussian simulations may be necessary [59]. All calculations are performed using the FFT, which makes this technique grid dependent.

$$\boldsymbol{C}_Y \boldsymbol{e} = \tau \boldsymbol{e} \tag{2.107}$$

Fast Fourier transform-moving average (FFT-MA) simulation method takes advantage of both fast Fourier transformation and simplicity of the moving average to simulate unconditional Gaussian realizations [41] . In this method, the FFT is applied separately to weighting function $\lambda(u)$ and random signal $R(u)$ to transform both functions to the frequency domain independently. The product of these two transforms in frequency domain is back transformed with FFT to original space to get unconditional simulation $Y^{us}(u)$. Such approach allows to update unconditional simulation locally by changing only some local values of the random signal $R(u)$. Because FFT is used in this simulation method, final realizations depend on the grid, size of which is a multiple of 2.

There are several spectral simulation programs available. The generation of

isotropic unconditional univariate random processes in 1-D, 2-D, and 3-D spaces has been implemented with Fortran-77 based on the Fourier integral method [60]. The program can generate the process on up to $2^{15} = 32{,}768$ grid nodes. Modifications have been made to generate unconditional nested anisotropic processes excluding zonal anisotropy on a 2-D grid [61]. Another Fortran-77 program for conditional spectral simulation of anisotropic realizations in 1-D, 2-D, or 3-D is available [83]. All programs require a simulation grid, and the simulation is performed in normal score units.

The fractal simulation is implemented similar to the spectral simulation with the difference that simulated random function has to have a particular covariance form [22, 46]. The fractal Gaussian noise (fGn) and fractal Brownian motion (fBm) are two of the most commonly encountered fractals in geostatistics. Their spatial structures differ from the ones that are usually used in conventional geostatistics. The covariance of the fBm in space and frequency domains can be presented for 1-D case as in Eqs. (2.108) and (2.109), respectively, where H is the Hurst exponent [76]. fGn is similar to white noise and can be described by a nugget effect spatial structure.

$$C_Y(h) \propto |h|^{-2H},\ 0 < H < 1 \tag{2.108}$$

$$\hat{C}_Y(\omega) \propto \omega^{-2H-1},\ 0 < H < 1 \tag{2.109}$$

Some spatial properties of petroleum reservoirs can be modeled with fBm in 2-D and 3-D using the Fourier integral method explained before [28, 58]. The resulting simulation preserves spatial structure of the data. The fBm process is continuous and non-differentiable. The fBm possesses a statistical self-similarity and self-affinity feature at various scales, which implies that some statistical properties change in the same manner in all dimensions and other properties

change in a proportional manner in all dimensions. The statistical self-affinity scaling behaviour is shown in Eq. (2.110), where κ is the scaling parameter [76].

$$\gamma_Y(\kappa h) \propto \kappa^{2H}\gamma_Y(h),\ 0 < H < 1 \tag{2.110}$$

Most of the covered techniques address multivariate systems based on some modifications of the linear model of coregionalization. The LMC is a conventional way to model spatial structure of multiple correlated variables, but requires tedious derivation of all direct- and cross-covariances, which is not an easy task [17, 26, 36, 44, 52, 53].

Surprisingly, but not many authors have addressed the importance of grid-free simulation algorithm in the geostatistical literature. As it was mentioned before, SS can be used potentially as a grid-free simulation technique [70]. However, periodicity issue in the reproduced covariance should be first solved [21].

The most obvious way to characterize the simulation of a spatial random function as a function of the coordinates of the simulation locations is to represent this simulation through a well-defined polynomial function. This research work proposes to represent unconditional simulation $Y^{us}(\boldsymbol{u})$ as a weighted sum of stochastic cosine functions, where weighting coefficients are derived from the Fourier series decomposition of the target covariance function (see Chapter 3). This approach bears common features with the spectral simulation techniques.

2.3.2 Categorical Variables

The simulation methods for categorical variables can be based on a two- and multiple-point statistics. First group is represented by a sequential indicator simulation (SIS), truncated Gaussian simulation for simple relationship between categories or truncated pluriGaussian simulation for more complex relationships, Markovian-type categorical prediction (MCP) [2, 10, 17, 64]. Second group is

represented by multiple point statistics (MPS) simulation method for categorical variables [74]. Training images reflecting similar spatial structure of the system are strongly required for the MPS algorithm. Because the implementation of these techniques is gridded, simulated values are not expressed freely as a function of the simulation node coordinates.

Current state of the grid-free simulation of discrete properties of the natural phenomena is tied to an object-based modeling [63]. They define categories through the geometrical shapes [31, 32]. First, the geological bodies are generated as analytical objects at a particular scale. Second, these objects are rasterized on a grid or on an irregular set of simulation nodes. This enables to express spatial distribution of the stationary domains as a function of the simulation location coordinates. The shape of the objects is limited to a particular depositional environment, and conditioning to the hard data is challenging [33].

The sequential indicator simulation is a widely used for the simulation of categorical variables. It borrows major aspects from the categorical kriging of probabilities explained in Section 2.2.2 and is very similar to SGS in the implementation aspect. The algorithm of the SIS is briefly reviewed, because it is used in Chapter 6 to simulate categorical domains of stationary geological properties [17]. First, the order of the categories is chosen to be used later in local CDF. Second, the random simulation path is defined. Third, all or some indicator data within search area $i_k(\boldsymbol{u}_\alpha)$ and previously simulated indicators $i_k^s(\boldsymbol{u}_\beta)$ are used in Eq. (2.78) to get an estimated indicator $i_k^*(\boldsymbol{u})$ for every category $k = 1, ..., K$, at simulation location $\boldsymbol{u}$. The estimated indicators are corrected if necessary, to add to one and be within range [0,1]. Forth, the local CDF of the estimated indicators is built. Fifth, the simulated categories $i_k^s(\boldsymbol{u})$ are generated by MCS concept from the local CDF of the estimated indicators. The random number is drawn by random number generator, which is later matched with the quantile of the CDF. Corresponding category is recorded. Simulated

indicator is one for the simulated category, and it is zero for the rest of the categories [64]. The procedure is repeated from third step to get simulated categories at other simulation locations. The procedure is terminated, when all simulation nodes are visited.

Chapter 3

Unconditional Grid-Free Simulation

To address conditional simulation of multivariate systems of natural phenomena, a simulation is generated as a two-step algorithm in normal scores. There is a remarkable advantage of reduced computational time in simulation of unconditional realizations. The conditioning can be performed by dual kriging, which is faster than the primal form of kriging and is conveniently represented as a function of the coordinates.

The two-step conditional simulation algorithm has been presented in Fig. 1.6 of Chapter 1. Major points of the multivariate grid-free simulation of the natural phenomena can be summarized as follows.

1. Define variables for the modeling
    - study available data
    - assess the relationship between variables
    - note the scale difference in the data
2. Pre-process data

- bring all data to point scale
- transform data to normal scores

3. Obtain spatial structure of the modeling system
   - compute experimental variograms in all possible directions
   - define principal directions of the continuity
   - fit variogram model
4. Perform simulation in the normal units in a grid-free format at the required simulation locations
   - generate unconditional realizations as a function of the coordinates at the simulation locations
   - condition realizations to data values
5. Check target statistics in the conditional simulation
   - check data reproduction
   - check normality of the realizations
   - check compliance of experimental variograms from realizations to data variogram and variogram model
   - check reproduction of the relationship between variables
6. Post-process realizations
   - transform realizations back to original units
   - upscale point-scale realizations to a larger scale as required
7. Perform simulation at another set of locations if necessary

The unconditional simulation is described in this chapter, which is one of two major steps of the proposed grid-free simulation algorithm. The second major step for conditioning unconditional realizations to the data is explained in Chapter 4. The unconditional simulation is represented as a polynomial of the coordinates of the simulation locations, and corresponding coefficients are derived from the target covariance function of the random process. The decomposition of the covariance function can be performed through the well-known Fourier series decomposition. The expression for the Fourier coefficients are straightforward and can be computed both analytically and numerically. The Fourier series representation of the covariance function is similar to spectral simulation algorithms. The resulting random function will follow a normal distribution. Multivariate realizations have the potential to be described well with LMC concept.

3.1 Fourier Series Simulation

The Fourier series simulation (FSS) has been developed for the grid-free representation of the simulated system. It is based on the linear model of regionalization for univariate systems or linear model of coregionalization for multivariate systems, where weights are derived from the decomposition of the target covariance function with the Fourier series.

3.1.1 Fourier Series

The Fourier series can be used to decompose any periodic continuous function $f(h)$ into a weighted sum of cosine and sine terms of various frequencies as shown in Eq. (3.1) with h being an argument of the function and q being an index [8]. The Fourier coefficients are represented by two kinds. The ones that correspond to cosine functions are denoted as g_q and are computed by integration of the continuous function as shown in Eq. (3.2). The others that correspond to sine

functions are denoted g'_q and are computed similarly to cosine Fourier coefficients as presented in Eq. (3.3). The product of a Fourier coefficient with the cosine or sine function is called the Fourier series term. The $2S$ stands for the periodic domain, where S is half of the periodic domain.

$$f(h) = \sum_{q=-\infty}^{\infty} \left(g_q \cos(\frac{\pi q h}{S}) + g'_q \sin(\frac{\pi q h}{S}) \right) \tag{3.1}$$

$$g_q = \frac{1}{2S} \int_{-S}^{S} f(h) \cos(\frac{\pi q h}{S}) dh \tag{3.2}$$

$$g'_q = \frac{1}{2S} \int_{-S}^{S} f(h) \sin(\frac{\pi q h}{S}) dh \tag{3.3}$$

In case when function $f(h)$ has a form that does not allow strict mathematical integration of Eqs. (3.2) and (3.3), it is possible to approximate integration with the discrete numerical summation of $2M + 1$ terms with distance increment $\Delta h = 2S/(2M+1)$ as shown in Eqs. (3.4) and (3.5).

$$g_q = \frac{1}{2S} \sum_{m=-M}^{M} f(m\Delta h) \cos(\frac{2\pi q m}{2M+1}) \tag{3.4}$$

$$g'_q = \frac{1}{2S} \sum_{m=-M}^{M} f(m\Delta h) \sin(\frac{2\pi q m}{2M+1}) \tag{3.5}$$

In order to better understand the concept of the Fourier series decomposition of the continuous periodic function, an example is provided below. The goal is to decompose function $f(h)$ with periodicity of $2S = 20.0$, shown in Eq. (3.6), into Fourier series terms with various term numbers. The parameter C equals to 1.0. The index i of peaks varies between -∞ and ∞. The base of this function can be represented as in Eq. (3.7), when peak number $i = 0$ is selected. Therefore, the function $f(h)$ is an even function, i.e. $f(h) = f(\text{-}h)$. Because

of this property, sine Fourier coefficients g' are all zero. It can be checked by substituting Eq. (3.6) or Eq. (3.7) into Eq. (3.3). On the other hand, cosine Fourier coefficients have a form shown in Eq. (3.8).

$$f(h) = \begin{cases} [h - (2i-1)S]\,\frac{C}{S}, & (2i-1)S < h \leq 2iS \\ [-h + (2i+1)S]\,\frac{C}{S}, & 2iS < h \leq (2i+1)S \end{cases} \tag{3.6}$$

$$f(h) = \begin{cases} [h + S]\,\frac{C}{S}, & -S < h \leq 0 \\ [-h + S]\,\frac{C}{S}, & 0 < h \leq S \end{cases} \tag{3.7}$$

$$g_p = \begin{cases} \frac{C}{2}, & p = 0 \\ \frac{C}{\pi^2 p^2}\left[1 - cos(\pi p)\right], & p \neq 0 \end{cases} \tag{3.8}$$

Fig. 3.1 shows original continuous periodic function $f(h)$ along with the corresponding 99 significant Fourier coefficients $\{g_q,\ q$ = -49, ..., 49$\}$ and two reconstructed functions $f^*(h)$ with the Fourier series of different length. The first function is approximated by 11 Fourier coefficient terms q = -5, ..., 5. The second function is represented by 99 Fourier coefficient terms q = -49, ..., 49. It is observed that Fourier series terms with indices closer to the zero index are more significant and possess more information about the original function.

The following figures are intended to better illustrate principles of the Fourier series-based decomposition of a continuous periodic function $f(h)$. Fig. 3.2 presents cosine Fourier coefficients starting with index $q = 0$ and expanding to left and right. It can be observed that all even Fourier coefficients are zero except g_0. All odd Fourier coefficients are non-zero and becomes smaller as one moves from the index $q = 0$ to both sides. Fig. 3.3 summarizes cosine functions that correspond to eleven Fourier series terms centered around $q = 0$ and last two Fourier series terms for q = -49 and q = 49. Note the periodic nature of each cosine function of various frequency. Also note that the last plot shows distorted high frequency plot, which should be very similar to other lower

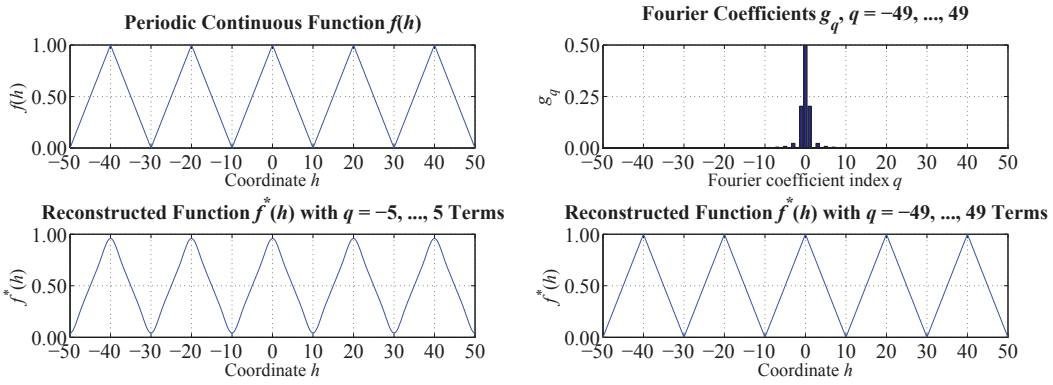


Figure 3.1: Fourier series example – original continuous periodic function $f(h)$, corresponding Fourier coefficients $\{g_p,\ p$ = -49, ..., 49$\}$, reconstructed function $f^*(h)$ with Fourier coefficients $\{g_{-5}$, ..., $g_5\}$, and reconstructed function $f^*(h)$ with Fourier coefficients $\{g_{-49},$ ..., $g_{49}\}$. Recall that all sine Fourier coefficients g' are zero.

frequency plots with higher intensity of lines. The reason of misrepresentation lies in the discretization level used for the plotting, which is smaller than a frequency of the cosine function. Next, Fig. 3.6 depicts Fourier series terms, that is, products of Fourier coefficients with corresponding cosine functions. The magnitude of each Fourier term decreases as its frequency increases. This can be clearly seen on Fig. 3.7. Finally, Fourier series-based representation of the original function $f(h)$ with various numbers of Fourier series terms can be seen in Fig. 3.8. A prominent feature of the periodic function decomposition with the Fourier series is that limited number of Fourier series terms around index q = 0 would contain the most information about the function. It can be seen by comparing reconstructed functions $f^*(h)$ with 11 and 99 terms. Both of them look very much alike, despite great difference in the number of terms used in the function $f(h)$ reconstruction.

It is crucial for this research work to notice that covariance function can be

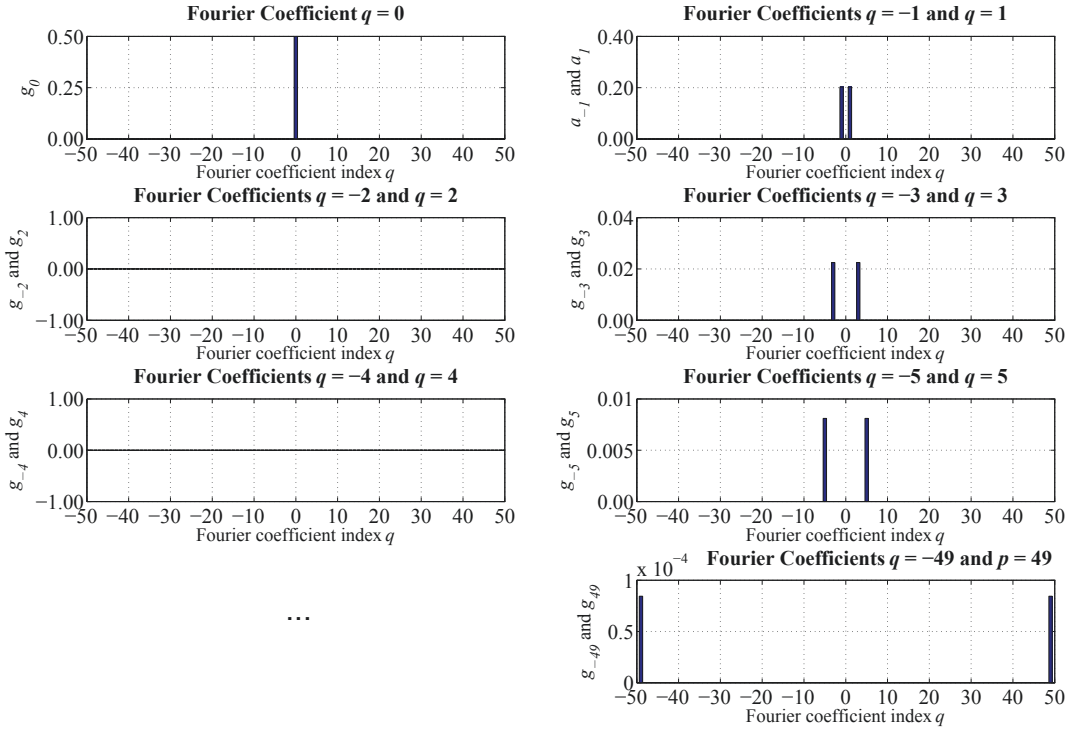


Figure 3.2: Fourier series example – first eleven and last two Fourier coefficients from $\{g_{-49}, ..., g_{49}\}$ range.

also deemed as a periodic continuous function as long as half of the periodic domain S is larger than the longest variogram range and larger than the simulation domain. By making this assumption, periodicity in the variogram reproduction and simulation is avoided. For further details, consult Fig. 3.7, where continuous periodic function in the form of the spherical covariance function $Sph(h)$ is shown. The variogram range r is 10.0 units. Half of the periodic domain S is 20.0 units. The variogram range r is smaller than S, as it is supposed to be. This spherical covariance function has a continuous periodic form. Therefore, to simulate realizations without periodic features, the simulation domain should be

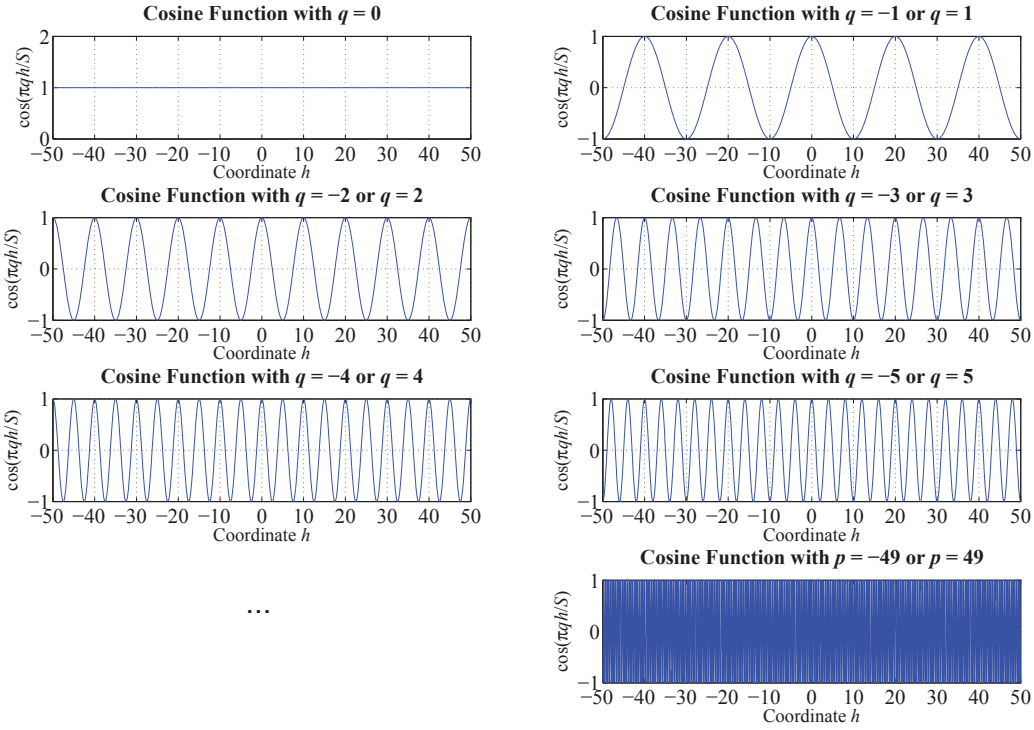


Figure 3.3: Fourier series example – cosine functions, which correspond to first eleven and last two Fourier series terms from $\{q$ = -49, ..., 49$\}$ range.

less than half of the periodic domain S.

The Fourier series can be easily used to decompose not just 1-D functions, but also higher dimensional ones. The general expression for the Fourier series for a μ-dimensional function is shown in Eq. (3.9). The Fourier coefficients are computed with Eqs. (3.10) and (3.11). Here the scalar terms are replaced with vectors.

$$f(\boldsymbol{h}) = \sum_{\boldsymbol{q}=-\infty}^{\infty} \left(g_{\boldsymbol{q}} \cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) + g'_{\boldsymbol{q}} \sin(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) \right) \tag{3.9}$$

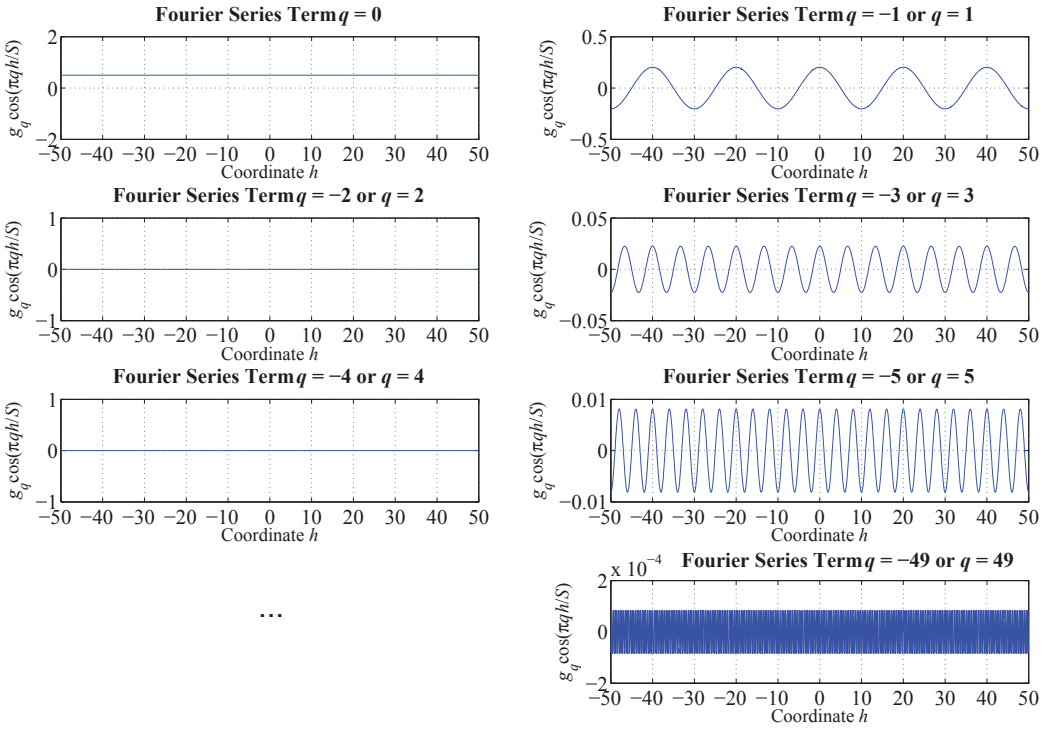


Figure 3.4: Fourier series example – first eleven and last two Fourier series terms from $\{q$ = -49, ..., 49$\}$ range.

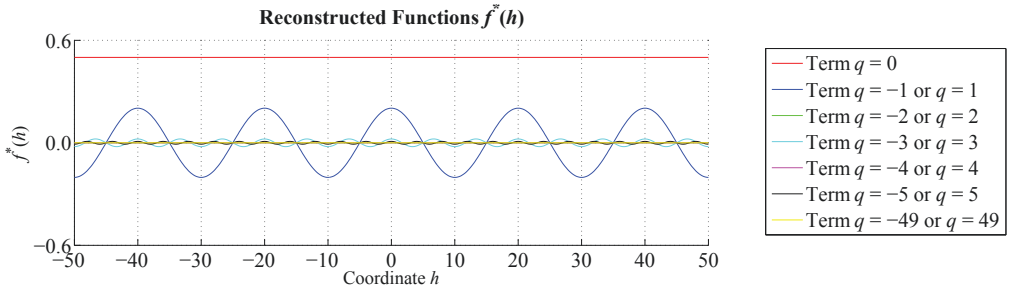


Figure 3.5: Fourier series example – first eleven and last two Fourier series terms from $\{q$ = -49, ..., 49$\}$ range imposed on one plot.

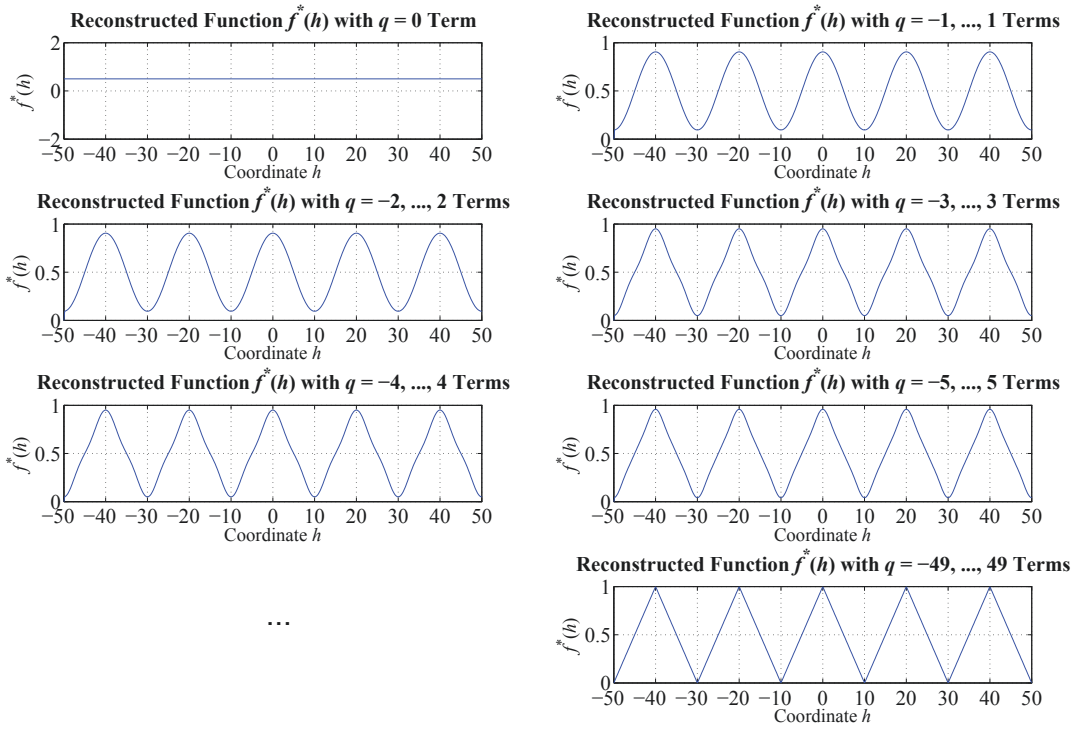


Figure 3.6: Fourier series example – Fourier series-based representation of the function $f(h)$ with various numbers of the Fourier series terms.

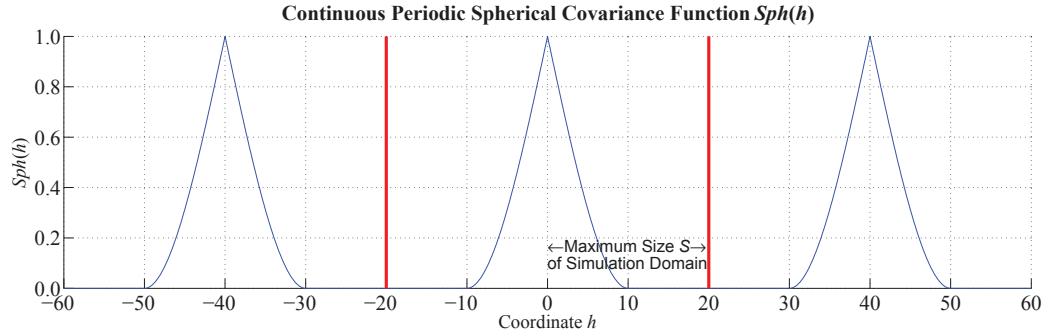


Figure 3.7: Continuous periodic representation of spherical covariance function $f(h) = Sph(h)$.

$$g_{\boldsymbol{q}} = \frac{1}{2^{\mu} S_1 ... S_{\mu}} \int_{-\boldsymbol{S}}^{\boldsymbol{S}} f(\boldsymbol{h}) \cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) d\boldsymbol{h} \tag{3.10}$$

$$g'_{\boldsymbol{q}} = \frac{1}{2^{\mu} S_1 ... S_{\mu}} \int_{-\boldsymbol{S}}^{\boldsymbol{S}} f(\boldsymbol{h}) \sin(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) d\boldsymbol{h} \tag{3.11}$$

Note that in the equations above, the enlarged dimensionality does not change the structure of the original Fourier series presented in Eqs. (3.1) – (3.3). The vectors in Eqs. (3.9) – (3.11) have forms shown in Eqs. (3.12) – (3.14). The unfolded expression for the Fourier series and its coefficients in μ-dimensional space can be rewritten as in Eqs. (3.15) – (3.17).

$$\boldsymbol{h} = \begin{bmatrix} h_1 & ... & h_{\mu} \end{bmatrix}^T \tag{3.12}$$

$$\boldsymbol{q} = \begin{bmatrix} q_1 & ... & q_{\mu} \end{bmatrix}^T \tag{3.13}$$

$$\boldsymbol{S} = \begin{bmatrix} S_1 & ... & 0 \\ & ... & \\ 0 & ... & S_{\mu} \end{bmatrix} \tag{3.14}$$

$$\begin{aligned} f(h_1, ..., h_{\mu}) = \sum_{q_1=-\infty}^{\infty} ... \sum_{q_{\mu}=-\infty}^{\infty} \Bigg(& g_{q_1 ... q_{\mu}} \cos(\frac{\pi q_1 h_1}{S_1} + ... + \frac{\pi q_{\mu} h_{\mu}}{S_{\mu}}) \\ & + g'_{q_1 ... q_{\mu}} \sin(\frac{\pi q_1 h_1}{S_1} + ... + \frac{\pi q_{\mu} h_{\mu}}{S_{\mu}}) \Bigg) \end{aligned} \tag{3.15}$$

$$g_{q_1 \ldots q_\mu} = \frac{1}{2^\mu S_1 \ldots S_\mu} \int_{-S_1}^{S_1} \ldots \int_{-S_\mu}^{S_\mu} f(h_1, \ldots, h_\mu) \cdot \cos\left(\frac{\pi q_1 h_1}{S_1} + \ldots + \frac{\pi q_\mu h_\mu}{S_\mu}\right) dh_1 \ldots dh_\mu \tag{3.16}$$

$$g'_{q_1 \ldots q_\mu} = \frac{1}{2^\mu S_1 \ldots S_\mu} \int_{-S_1}^{S_1} \ldots \int_{-S_\mu}^{S_\mu} f(h_1, \ldots, h_\mu) \cdot \sin\left(\frac{\pi q_1 h_1}{S_1} + \ldots + \frac{\pi q_\mu h_\mu}{S_\mu}\right) dh_1 \ldots dh_\mu \tag{3.17}$$

An example of anisotropic covariance function decomposition in 2-D ($\mu = 2$) is shown in Figs. 3.8 – 3.10 using the numerical integration of the Fourier coefficients. The covariance function takes the form of the most common covariance functions: spherical, exponential, and Gaussian covariances. The covariance function is deemed to be a continuous periodic function with periodic ranges $2S_1 = 40.0$ units in h_1 direction, which makes 90^o with the azimuth, and $2S_2 = 40.0$ units in h_2 direction. The anisotropy is presented by different covariance ranges: $r_1 = 10.0$ units in h_1 direction and $r_2 = 5.0$ units in h_2 direction. Only 81 Fourier coefficients $g_{q_1 q_2}$ (discrete values) are computed for each index q_1 and q_2. The direction of the anisotropy is reversed by 90^o in the space of Fourier coefficients. It is observed again that even though the summation of the Fourier series has to be performed over the infinite space, in practice, the number of the Fourier coefficients required for the Fourier series-based representation of the continuous periodic function is finite and is not large. Only limited terms are necessary to represent the function. As a recommendation, the Fourier series terms $\boldsymbol{q} = \{-Q_1, \ldots, Q_1; \ldots; -Q_\mu, \ldots, Q_\mu\}$ with the corresponding Fourier coefficients larger than 0.01 (1%) of the largest Fourier coefficient, which happens to be coefficient

g_0, need to be retained in the function decomposition for the accurate and precise results. The study that justifies the choice of the optimal number of the Fourier coefficients used for covariance decomposition is presented in Section 5.3 of Chapter 5. This optimal threshold was defined visually from the realization maps and reproduction of the target variogram.

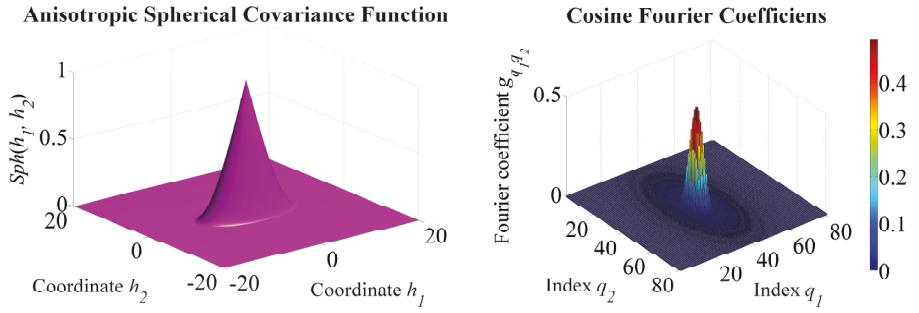


Figure 3.8: (a) Anisotropic spherical covariance function in 2-D space, and (b) corresponding Fourier coefficients in 2-D space.

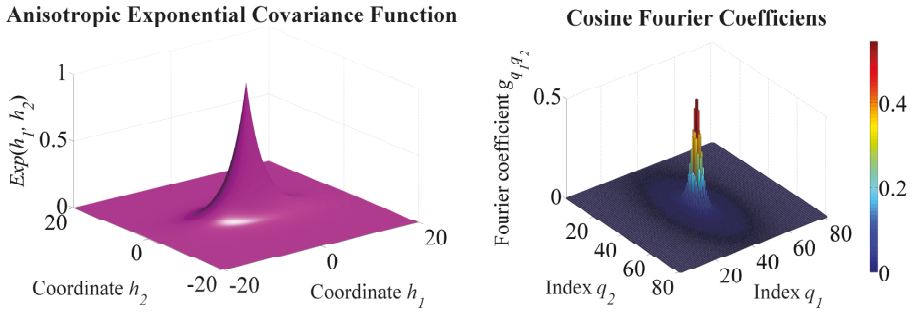


Figure 3.9: (a) Anisotropic exponential covariance function in 2-D space, and (b) corresponding Fourier coefficients in 2-D space.

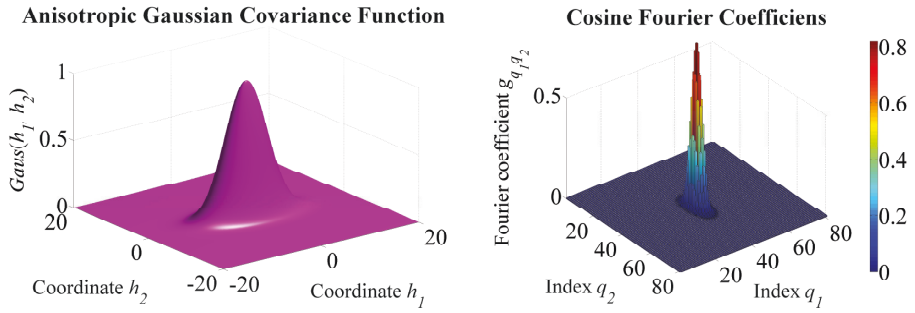


Figure 3.10: (a) Anisotropic Gaussian covariance function in 2-D space, and (b) corresponding Fourier coefficients in 2-D space.

3.1.2 Univariate Simulation

The concept of the Fourier series-based representation of the covariance function can be expanded further for the unconditional simulation. For this reason, the LMR concept comes in handy. The LMR Eq. (2.36) is rewritten in Eq. (3.18), where instead of random function Z, random function Y is to be simulated in a normal space with zero mean and unit variance. The zero term independent random factor X_0 is left aside, because of its special covariance function. This independent factor is also called white noise; it has a nugget covariance function. The covariance of the target random function Y is expressed as in Eq. (3.19), the schematic graph of which is shown in Fig. 3.11 for 1-D. The covariance functions are mostly defined through the input or known variogram models, which would have been derived from the data and have a form as in Eq. (3.20). Fig. 3.11(a) shows that a target covariance function with a nugget effect is not continuous. This means that the Fourier series approach cannot be used to decompose the covariance function $C_Y(\boldsymbol{h})$. On the other hand, when the white noise is removed from the LMR, the corresponding covariance function becomes continuous as required (see Fig. 3.11(b)). For now, we will work with the random function Y' $= Y - g_0 X_0$, expression of which along with its covariance is shown in Eqs. (3.21)

and (3.22). Therefore, by comparing Eq. (3.22) with Eq. (3.9) of the Fourier series, it can be argued that these two equations are identical, i.e. $C_{Y'}(\boldsymbol{h})$ can be easily presented as $f(\boldsymbol{h})$. To illustrate this claim, let's recall that sine Fourier coefficients $g'_{\boldsymbol{q}}$ are zero for the even functions (symmetric about vertical axis). The parallel between notations should be also drawn: p index is replaced with $\boldsymbol{q}$, and $C_{X_q}(\boldsymbol{h})$ becomes a cosine function. To put this in a mathematical expression, Eq. (3.23) can be used. Also, a finite number of the Fourier coefficients $\boldsymbol{Q}$ is used in the decomposition, because only coefficients centered around zero index contribute to the decomposed function.

$$Y(\boldsymbol{u}) = a_0 X_0(\boldsymbol{u}) + \sum_{p=1}^{P} a_p X_p(\boldsymbol{u}) \tag{3.18}$$

$$C_Y(\boldsymbol{h}) = a_0^2 C_{X_0}(\boldsymbol{h}) + \sum_{p=1}^{P} a_p^2 C_{X_p}(\boldsymbol{h}) \tag{3.19}$$

$$\gamma_Y(\boldsymbol{h}) = b_0 \Gamma_0(\boldsymbol{h}) + \sum_{p=1}^{P} b_p \Gamma_p(\boldsymbol{h}) \tag{3.20}$$

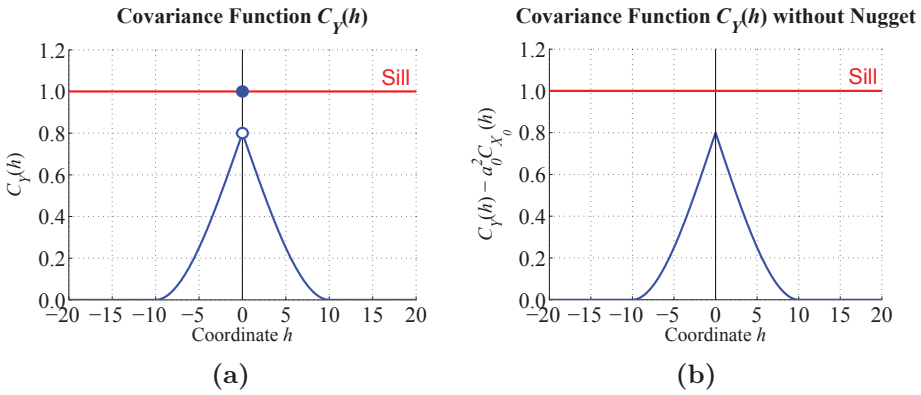


Figure 3.11: Plot of the covariance function of the random function Y (a) with nugget and (b) without nugget.

$$Y'(\boldsymbol{u}) = \sum_{p=1}^{P} a_p X_p(\boldsymbol{u}) \tag{3.21}$$

$$C_{Y'}(\boldsymbol{h}) = \sum_{p=1}^{P} a_p^2 C_{X_p}(\boldsymbol{h}) \tag{3.22}$$

$$C_{Y'}(\boldsymbol{h}) = f(\boldsymbol{h}) \Rightarrow \begin{cases} p \rightarrow \boldsymbol{q} \\ P \rightarrow \boldsymbol{Q} \Rightarrow Q = 2(Q_1 + ...Q_\mu) + \mu \\ \sum\limits_{p=1}^{P} \rightarrow \sum\limits_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} \\ a_p^2 \rightarrow a_{\boldsymbol{q}}^2 \\ C_{X_p}(\boldsymbol{h}) \rightarrow C_{X_{\boldsymbol{q}}}(\boldsymbol{h}) \end{cases} \tag{3.23}$$

To extend further, the LMR changes from Eq. (3.21) to Eq. (3.24), where the LMR coefficients $a_{\boldsymbol{q}}$ are found from the cosine Fourier coefficients $g_{\boldsymbol{q}}$ as shown in Eq. (3.25), and random factor $X_{\boldsymbol{q}}(\boldsymbol{u})$ has the form as presented in Eq. (3.26). The covariance function of the proposed random function has to be in the cosine form as shown in Eq. (3.27). The most important part is that random factors (and also target random function Y') are represented by a random component $\phi_{\boldsymbol{q}}$, which is a function of the Fourier coefficient index $\boldsymbol{q}$ and is not a function of the location $\boldsymbol{u}$. Therefore, final expression for the unconditional realization $Y_i^{us}(\boldsymbol{u})$ of random function Y at location $\boldsymbol{u}$ in normal scores is shown in Eq. (3.28), where the random phase $\phi_{\boldsymbol{q},i}$ of uniform distribution is tied to index of the Fourier series coefficient $\boldsymbol{q}$ and a realization index i. The random variable $Y'(\boldsymbol{u})$ is expressed as a function of the coordinates of the simulation location $\boldsymbol{u}$. This simulation method is named Fourier series simulation. The simulation of a white noise X_0 (nugget component) in a grid-free manner is explained later in Section 3.3.

$$Y'(\boldsymbol{u}) = \sum_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} a_{\boldsymbol{q}} X_{\boldsymbol{q}}(\boldsymbol{u}) \tag{3.24}$$

$$a_{\boldsymbol{q}} = \sqrt{g_{\boldsymbol{q}}} \tag{3.25}$$

$$X_{\boldsymbol{q}}(\boldsymbol{u}) = \sqrt{2} cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{u} + \phi_{\boldsymbol{q}}), \;\; \phi_{\boldsymbol{q}} \in [0, 2\pi] \tag{3.26}$$

$$C_{X_{\boldsymbol{q}}}(\boldsymbol{h}) = \cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) \tag{3.27}$$

$$Y_i^{us}(\boldsymbol{u}) = \sqrt{b_0} X_{0,i}(\boldsymbol{u}) + \frac{1}{\sqrt{2^{\mu-1} S_1 ... S_\mu}} \sum_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{u} + \phi_{\boldsymbol{q},i}) \cdot \sqrt{\int_{-\boldsymbol{S}}^{\boldsymbol{S}} (C_Y(\boldsymbol{h}) - b_0 C_{X_0}(\boldsymbol{h})) \cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) d\boldsymbol{h}} \tag{3.28}$$

To illustrate the principles of grid-free simulation based on the Fourier series decomposition of the covariance function of Y', 1-D and 2-D examples are provided below. Following examples show that the FSS generates a realization of a random function Y' as a continuous function in space, which can be resolved at any set of locations (simulation nodes). In other words, FSS produces grid-free simulation.

Consider the generation of a stochastic process in 1-D over the [0 – 1000] unit interval with the exponential covariance function $C_{Y'}(h)$ expressed by Eq. (3.29), where correlation range r is 50.0 units. The cosine Fourier coefficients g_q can be easily computed from Eq. (3.2), where $f(h)$ is replaced with the target exponential covariance function. These Fourier coefficients have analytical form shown in Eq. (3.30). A total of 1001 cosine Fourier coefficients are computed.

The half periodic domain S is 2000.0 units. The sine Fourier coefficients g'_q are zero. The plots of the exponential covariance function and 1001 Fourier coefficients are shown in Fig. 3.12. The covariance function is shown for the full range of the periodic domain.

$$C_{Y'}(h) = exp\left(-3\frac{|h|}{r}\right), \quad r = 50.0 \tag{3.29}$$

$$g_q = \frac{3\left(1-(-1)^q exp\left(-3\frac{|h|}{r}\right)\right)}{rS\left(\left(\frac{3}{r}\right)^2+\left(q\frac{\pi}{S}\right)^2\right)}, \quad S = 2000.0, \quad q = -500, ..., 500 \tag{3.30}$$

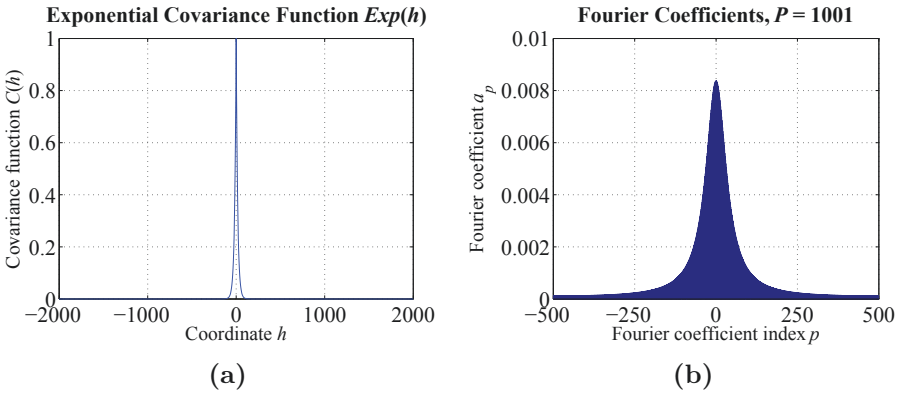


Figure 3.12: Univariate 1-D Fourier series simulation example – (a) exponential target covariance function and (b) corresponding Fourier coefficients.

Four grids are established to represent the single realization of the spatial random function Y' on the [0.0 – 1000.0] unit interval: three evenly structured grids with 100.0, 10.0, and 1.0 unit intervals, and one randomly structured grid. A single realization is shown in Fig. 3.13, where point-scale values simulated at specified simulation nodes are connected to each other by lines. It is clear that despite the grid resolution, the general trend of the simulation can be seen. The portion [470.0 – 530.0] unit of the interval bounded by two vertical black lines

is zoomed in to further explain the grid-free nature of the simulation, and is shown in Fig. 3.14. The realization is represented by dots, which is the correct way of thinking about the proposed grid-free simulation at a point scale. The locations of the simulation nodes overlap each other for evenly structured grids, and simulated values are identical for these overlapping nodes. The simulated values at the randomly distributed simulation locations of the randomly structured grid also follow the realization shape. To emphasize the reason behind it, the simulation is already calculated by the FSS algorithm for the entire space. The grid specification defines the coordinates of the simulation nodes, at which the stochastic process is resolved. The variogram reproduction computed from the simulation resolved on evenly structured grids is shown in Fig. 3.15. The experimental variogram closely follows the variogram model shown by black line for short ranges and diverges from the variogram model for longer ranges due to expected statistical fluctuations (physical limitations of the simulation interval). Experimental variograms computed from various grids closely follow each other as expected, since all realizations are values of the same simulation resolved at different node spacing.

Another example of the grid-free simulation with the FSS is presented below for a 2-D space. Four various grids are selected to emphasize grid-free nature of the proposed simulation algorithm. Fig. 3.16 shows the grids. The simulation domain has a rectangular shape with the longest side of 80.0 units and shortest side of 50.0 units rotated 45° azimuth. The first grid has a coarse resolution of evenly distributed $17 \times 11 = 187$ simulation nodes. The spacing between nodes is 5.0×5.0 units^2. The second grid has a fine resolution of evenly distributed $80 \times 50 = 4000$ simulation nodes with 1.0×1.0 units^2 spacing. The third grid is a combination of the first two grids. It contains 1133 simulation nodes. Last, the fourth grid has a random structure and consists of 187 simulation nodes. A single realization of the 2-D random function Y' is shown on the same figure.

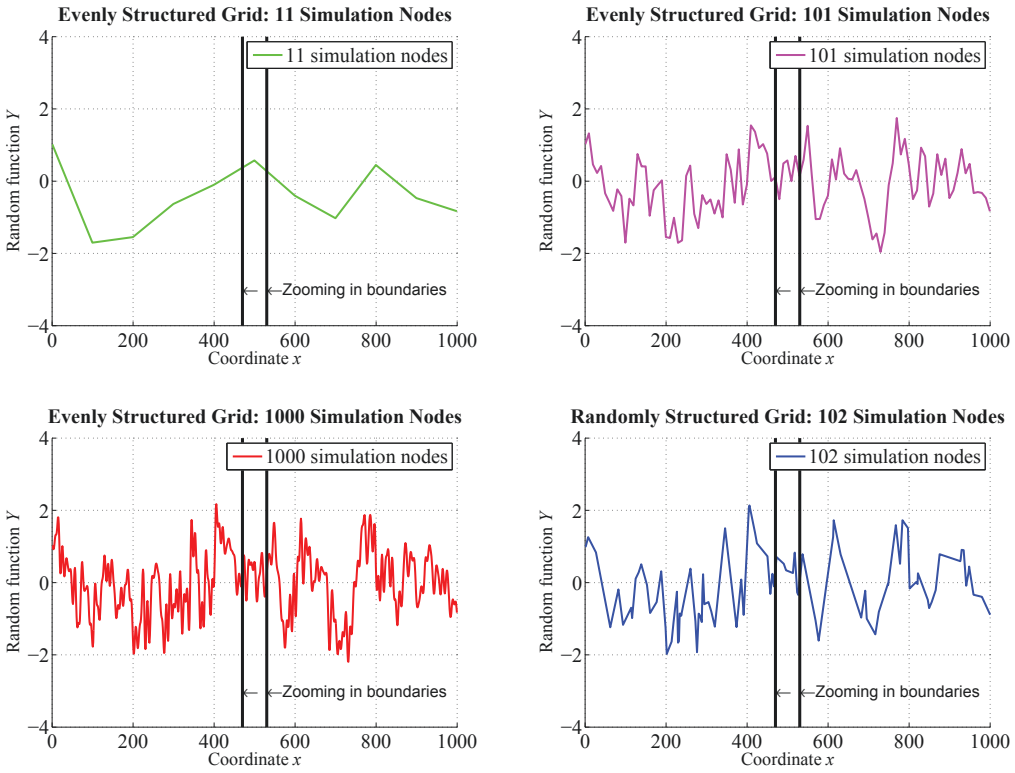


Figure 3.13: Univariate 1-D Fourier series simulation example – single grid-free realization simulated on four grids of various resolutions: (a) evenly structured 11-point grid, (b) evenly structured 101-point grid, (c) evenly structured 1000-point grid, and (d) random 102-point grid.

Note that despite the difference in the grid resolutions, the same realization of the simulation is presented. The target covariance function has an anisotropic spherical form shown in Eq. (3.31).

$$C_{Y'}(\boldsymbol{h}) = Sph_{\substack{r_1=20.0 \\ r_2=10.0 \\ \alpha=120°}}(\boldsymbol{h}) \tag{3.31}$$

A total of 100 realizations are simulated over the finely resolved grid 2. The first four realizations are shown in Fig. 3.17. The anisotropy is clearly seen

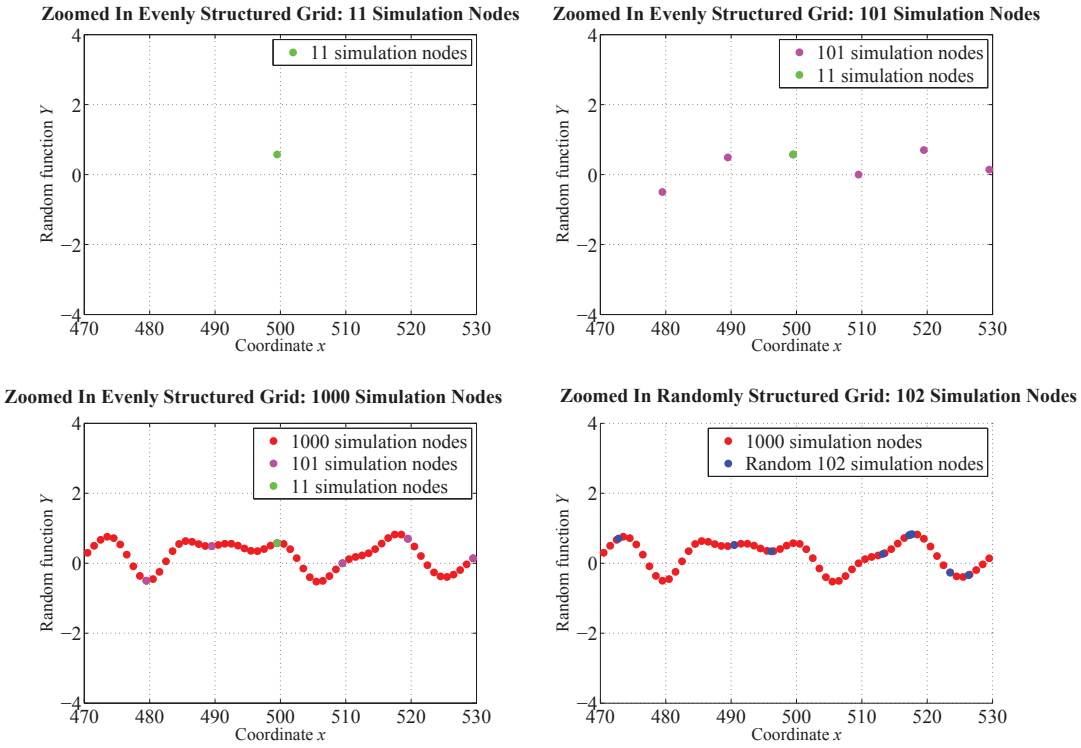


Figure 3.14: Univariate 1-D Fourier series simulation example – zoomed in grid-free realization simulated on four grids of various resolutions: (a) evenly structured 11-point grid, (b) evenly structured 101-point grid, (c) evenly structured 1000-point grid, and (d) random 102-point grid.

in the realizations with major direction of continuity being at 120° azimuth. The histograms of the simulated values from first two realizations and all realizations at once are shown in Fig. 3.18 along with the normal probability plot for all 100 realizations. Because of the ergodic fluctuations, there are deviations from a perfect normal distribution. There is less deviation when all realizations are combined together. Mean and variance maps of 100 realizations along with the corresponding histograms are presented in Fig. 3.19. The local mean is close to zero and local variance is close to 1.0 for most of the locations. The histograms

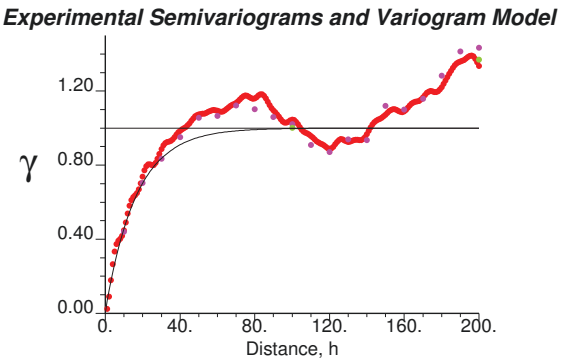


Figure 3.15: Univariate 1-D Fourier series simulation example – target variogram model shown by black line and experimental variograms computed from grid-free realization simulated on three grids of various resolutions shown by dots: (a) experimental variogram in light green computed from evenly structured 11-point grid, (b) experimental variogram in purple computed from evenly structured 101-point grid, and (c) experimental variogram in red computed from evenly structured 1000-point grid.

suggest some variability present in the local mean and variance of the realizations. The experimental variograms computed from the first three realizations and the average experimental variogram computed from all realizations are shown in Fig. 3.20 with the anisotropic spherical variogram model. As described before, the ergodicity introduces deviations of the experimental variograms from the target model. But on average, the variogram reproduction appears reasonable. All 100 experimental variograms are shown in green with the average experimental variogram shown in red and the variogram model shown in black in Fig. 3.21 for both major and minor directions of continuity. The variogram is reproduced better for the major direction of continuity. Fig. 3.22 illustrates experimental variograms computed from the unconditional realizations generated by SGS with the same variogram model. The locations of the simulation nodes are slightly different for the SGS realizations. The variogram is reproduced well for shorter ranges with greater deviations at larger distances. This is a characteristic feature of the SGS simulation method, which is outperformed by the proposed grid-free

simulation method FSS [21]. The spread of individual experimental variograms is less for the SGS than for the FSS generated realizations. It may be explained by larger number of the data pairs used in the computation of the experimental variograms from the SGS realizations.

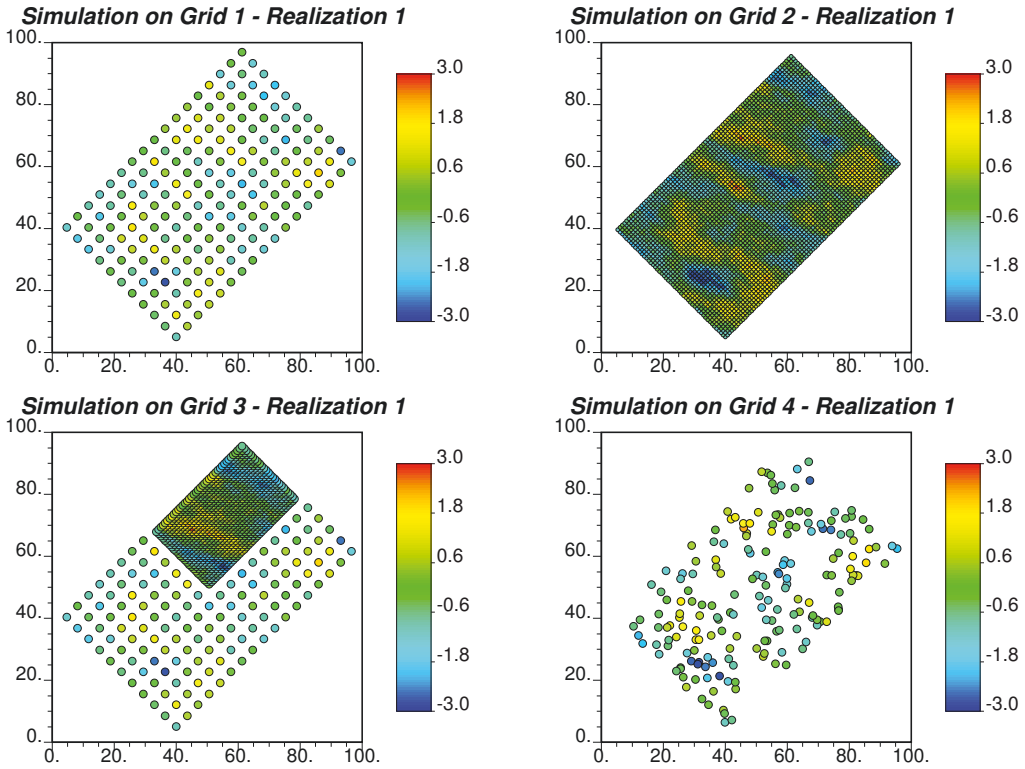


Figure 3.16: Univariate 2-D Fourier series simulation example – four distinct simulation grids: (a) coarse evenly structured grid 1, (b) fine evenly structured grid 2, (c) combination of coarse and fine evenly placed nodes on grid 3, and (d) random grid 4.

3.1.3 Multivariate Simulation

Most systems consist of more than just one property. Therefore, there is a need to develop a multivariate simulation algorithm in a grid-free manner. The simplest solution is to replace the LMR with the LMC, where the linear model's

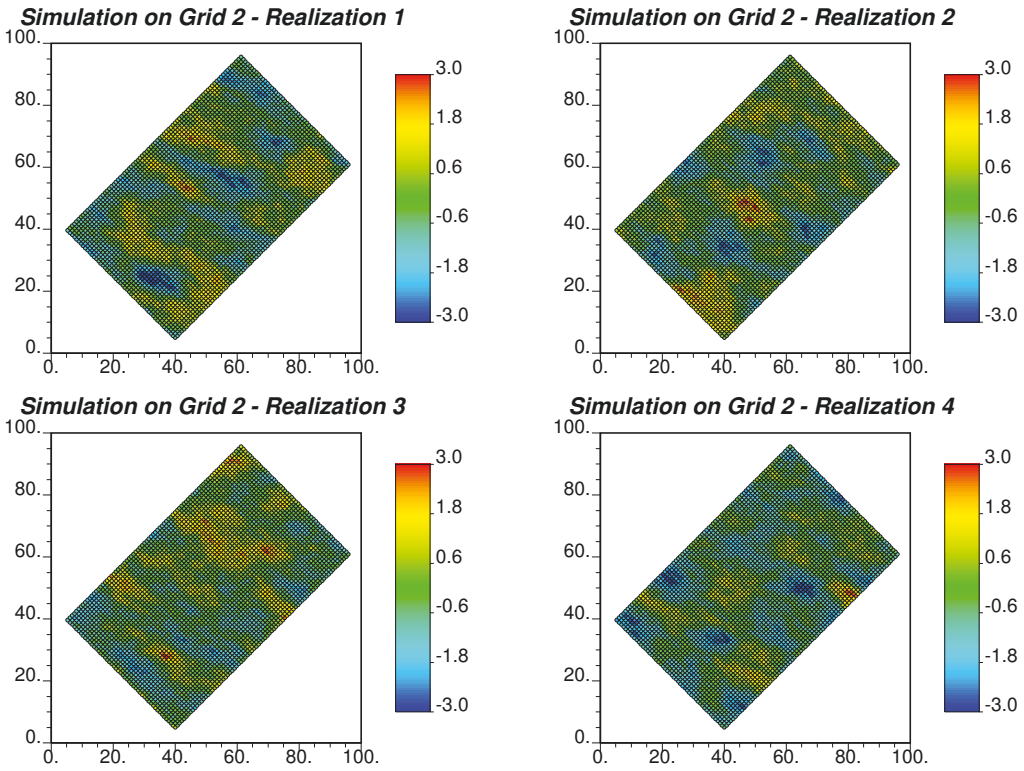


Figure 3.17: Univariate 2-D Fourier series simulation example – realizations 1 – 4 on the fine grid 2.

coefficients take a matrix form and independent random factors are stored in a vector form. The linear model of coregionalization is shown in Eq. (3.32) for a multivariate system Y in normal units with zero mean and unit variance. Description of the matrices $\boldsymbol{A}_p$ and $\boldsymbol{X}_p(\boldsymbol{u}), p = 1, ..., P$, are provided in Eqs. (2.40) and (2.41) of Section 2.1.2, Chapter 2. The random variables of the multivariate system are stored in the matrix $\boldsymbol{Y}(\boldsymbol{u})$ as shown in Eq. (3.33). The covariance matrix of the modeled system can be presented as shown in Eq. (3.34), the terms of which are explained in Eq. (3.35). The variogram model in Eq. (3.36) is the input for the simulation of random variables of the system. Matrices $\boldsymbol{B}_p, p = 1, ..., P$, have form as in Eq. (2.51), and are related to the LMR coefficients as shown in Eq. (3.37).

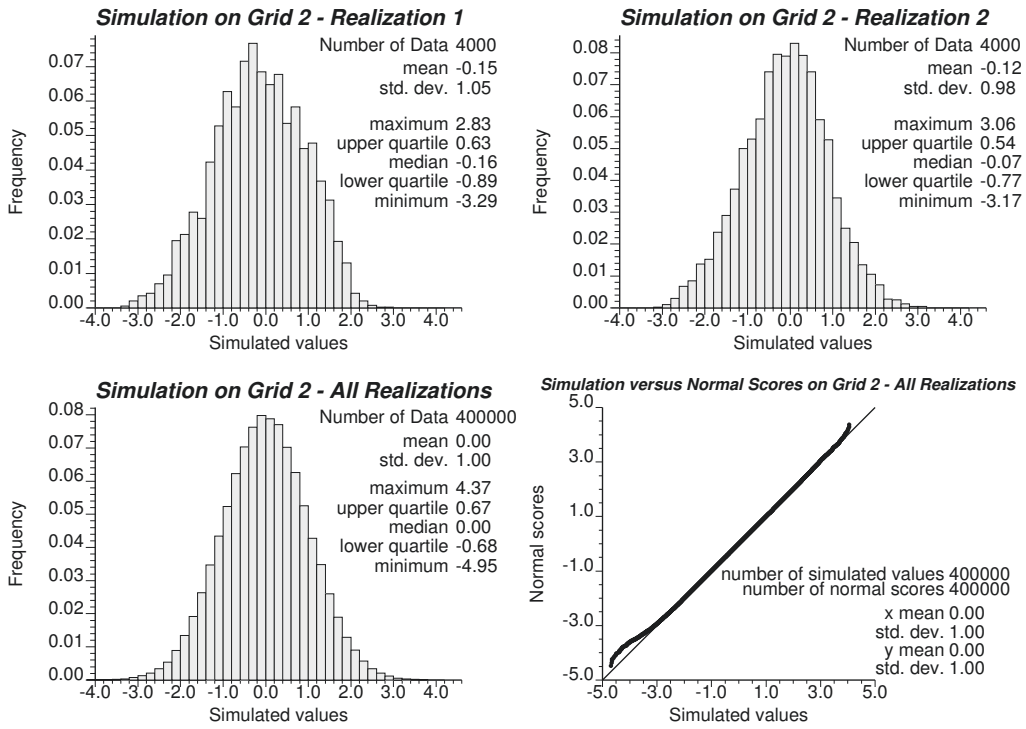


Figure 3.18: Univariate 2-D Fourier series simulation example – histograms of realization 1 and 2 and all 100 realizations on the fine grid 2 along with the normal probability plot of all realization values.

$$\boldsymbol{Y}(\boldsymbol{u}) = \boldsymbol{A}_0 \boldsymbol{X}_0(\boldsymbol{u}) + \sum_{p=1}^{P} \boldsymbol{A}_p \boldsymbol{X}_p(\boldsymbol{u}) \tag{3.32}$$

$$\boldsymbol{Y}(\boldsymbol{u}) = [Y_1(\boldsymbol{u}) \; ... \; Y_K(\boldsymbol{u})]^T \tag{3.33}$$

$$\boldsymbol{C}_Y(\boldsymbol{h}) = \boldsymbol{A}_0 [\boldsymbol{A}_0]^T C_{X_0}(\boldsymbol{h}) + \sum_{p=1}^{P} \boldsymbol{A}_p [\boldsymbol{A}_p]^T C_{X_p}(\boldsymbol{h}) \tag{3.34}$$

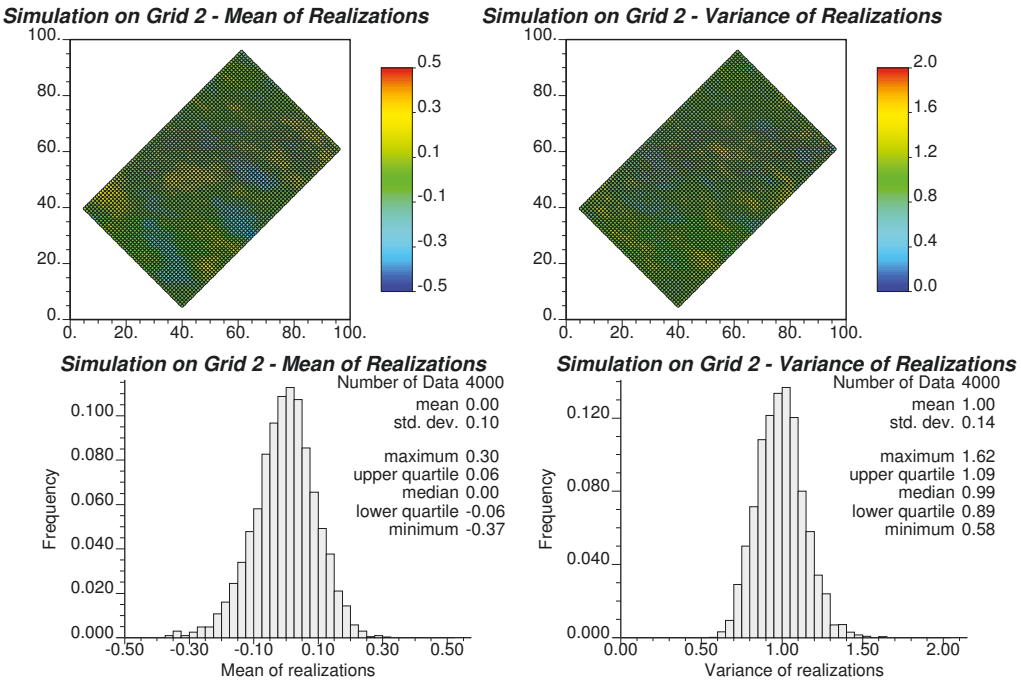


Figure 3.19: Univariate 2-D Fourier series simulation example – mean and variance fine grid 2 maps computed from 100 realizations along with the corresponding histograms.

$$\boldsymbol{C}_Y(\boldsymbol{h}) = \begin{bmatrix} C_{Y_1Y_1}(\boldsymbol{h}) & ... & C_{Y_1Y_K}(\boldsymbol{h}) \\ ... & & ... \\ C_{Y_KY_1}(\boldsymbol{h}) & ... & C_{Y_KY_K}(\boldsymbol{h}) \end{bmatrix} \tag{3.35}$$

$$\gamma_Y(\boldsymbol{h}) = \boldsymbol{B}_0\Gamma_{X_0}(\boldsymbol{h}) + \sum_{p=1}^{P} \boldsymbol{B}_p\Gamma_{X_p}(\boldsymbol{h}) \tag{3.36}$$

$$\boldsymbol{B}_p = \boldsymbol{A}_p[\boldsymbol{A}_p]^T, \;\; p = 0, ..., P \tag{3.37}$$

Following the same logical path as for the univariate unconditional grid-free simulation, the unconditional simulation of the multivariate system without

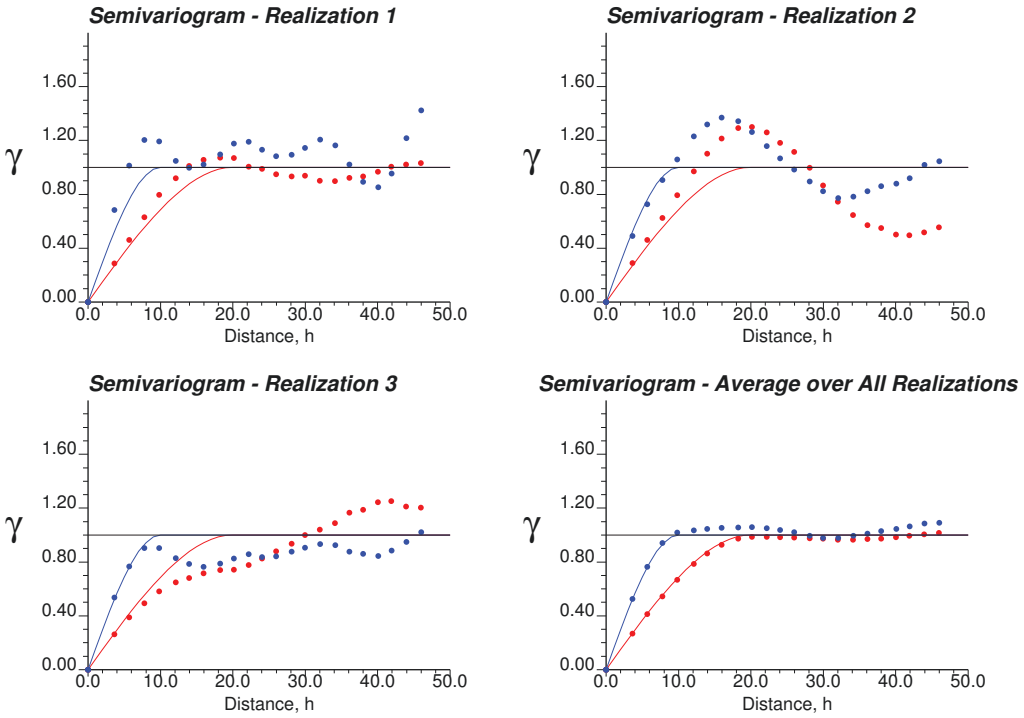


Figure 3.20: Univariate 2-D Fourier series simulation example – experimental variograms (dots) for major (in red) and minor (in blue) directions for realizations 1 – 3 and average variogram from all 100 realizations simulated on a fine grid 2 and corresponding variogram models (lines).

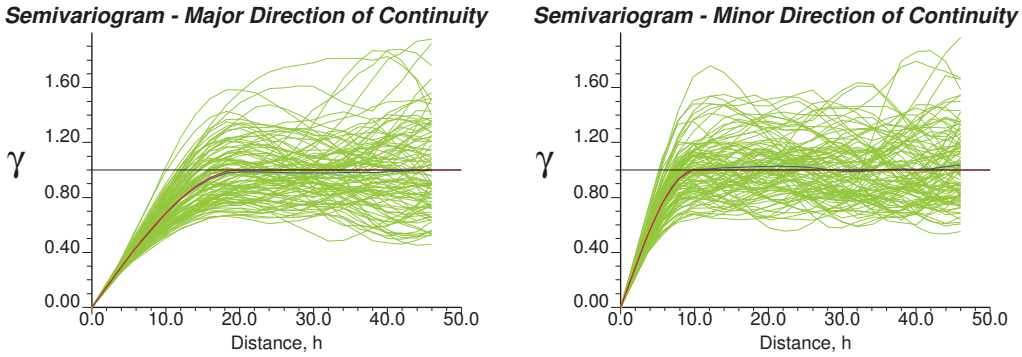


Figure 3.21: Univariate 2-D Fourier series simulation example – all 100 experimental variograms (in green) and average experimental variogram (in blue) from grid 2 FFS realizations along with the variogram model (in red) for major and minor directions of continuity computed.

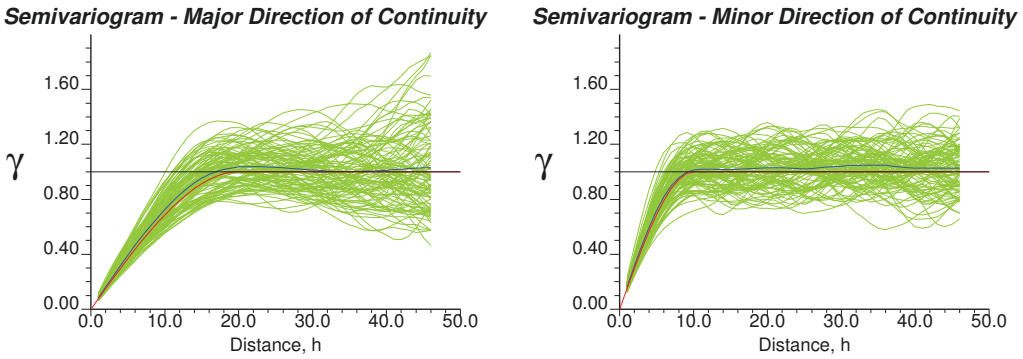


Figure 3.22: Univariate 2-D Fourier series simulation example – all 100 experimental variograms (in green) and average experimental variogram (in blue) computed from the SGS realizations resolved on grid 2 along with the variogram model (in red) for major and minor directions of continuity.

nugget effect is defined first $\boldsymbol{Y}'(\boldsymbol{u})$ as in Eq. (3.38). Each term of the covariance matrix $\boldsymbol{C}_{Y'}(\boldsymbol{h})$ of $\boldsymbol{Y}'(\boldsymbol{u})$ in Eq. (3.39) can be decomposed with the Fourier series. The analogy between this covariance matrix and the Fourier series decomposition is presented in Eq. (3.40) for the multivariate case. As before, all sine Fourier coefficients $g'_{\boldsymbol{q}}$ are zero, because the covariance is an even function. Therefore, the system $\boldsymbol{Y}'(\boldsymbol{u})$ and covariance function $\boldsymbol{C}_{Y'}(\boldsymbol{h})$ can be rewritten as in Eqs. (3.41) and (3.42), respectively, with parameters computed as shown in Eqs. (3.43) – (3.47).

$$\boldsymbol{Y}'(\boldsymbol{u}) = \sum_{p=1}^{P} \boldsymbol{A}_p \boldsymbol{X}_p(\boldsymbol{u}) \tag{3.38}$$

$$\boldsymbol{C}_{Y'}(\boldsymbol{h}) = \sum_{p=1}^{P} \boldsymbol{A}_p [\boldsymbol{A}_p]^T C_{X_p}(\boldsymbol{h}) \tag{3.39}$$

$$C_{Y'}(\boldsymbol{h}) = f(\boldsymbol{h}) \Rightarrow \begin{cases} p \rightarrow \boldsymbol{q} \\ P \rightarrow \boldsymbol{Q} \Rightarrow Q = 2(Q_1 + ... + Q_\mu) + \mu \\ \sum\limits_{p=1}^{P} \rightarrow \sum\limits_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} \\ a_{p,kl}^2 \rightarrow a_{\boldsymbol{q},kl}^2 \\ C_{X_p}(\boldsymbol{h}) \rightarrow C_{X_q}(\boldsymbol{h}) \end{cases} \tag{3.40}$$

$$\boldsymbol{Y'}(\boldsymbol{u}) = \sum_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} \boldsymbol{A_q} \boldsymbol{X_q}(\boldsymbol{u}) \tag{3.41}$$

$$\boldsymbol{C}_{Y'}(\boldsymbol{h}) = \sum_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} \boldsymbol{G_q} C_{X_q}(\boldsymbol{h}) \tag{3.42}$$

$$\boldsymbol{G_q} = \boldsymbol{A_q}[\boldsymbol{A_q}]^T, \;\; -\boldsymbol{Q} < \boldsymbol{q} < \boldsymbol{Q} \tag{3.43}$$

$$a_{\boldsymbol{q},kl} = \begin{cases} |a_{\boldsymbol{q},11}| = \sqrt{g_{\boldsymbol{q},11}} \\ |a_{\boldsymbol{q},k1}| = \frac{g_{\boldsymbol{q},k1}}{|a_{\boldsymbol{q},11}|} \\ |a_{\boldsymbol{q},kk}| = \sqrt{g_{\boldsymbol{q},kk} - \sum\limits_{m=1}^{k-1} a_{\boldsymbol{q},km}^2} \\ |a_{\boldsymbol{q},kl}| = \frac{g_{q,kl} - \sum\limits_{m=1}^{l-1} |a_{q,km}||a_{q,lm}|}{|a_{q,ll}|} \end{cases} \tag{3.44}$$

$$g_{\boldsymbol{q},kl} = \frac{1}{2^\mu S_1 ... S_\mu} \int_{-\boldsymbol{S}}^{\boldsymbol{S}} (C_{Y_k Y_l}(\boldsymbol{h}) - b_{0,kl} C_{X_0}(\boldsymbol{h})) \cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) d\boldsymbol{h} \tag{3.45}$$

$$X_{\boldsymbol{q},k}(\boldsymbol{u}) = \sqrt{2} cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{u} + \phi_{\boldsymbol{q},k}), \;\; \phi_{\boldsymbol{q},k} \in [0, 2\pi] \tag{3.46}$$

$$C_{X_{\boldsymbol{q}}}(\boldsymbol{h}) = \cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{h}) \tag{3.47}$$

Thus, the final expression for the unconditional multivariate grid-free realization $\boldsymbol{Y}_i^{us}(\boldsymbol{u})$ at location $\boldsymbol{u}$ in normal scores is shown in Eq. (3.48) or Eq. (3.49), where the random phase $\phi_{\boldsymbol{q},k,i}$ is tied to index of the Fourier series coefficient $\boldsymbol{q}$, random function index k, and a realization index i. Note that matrices $\boldsymbol{A}_{\boldsymbol{q}}$ are the lower triangular matrices, i.e. $\mathrm{a}_{\boldsymbol{q},kl} = 0$ for $k < l \; \forall \boldsymbol{q}$.

$$\boldsymbol{Y}(\boldsymbol{u}) = \sqrt{b_0}\boldsymbol{X}_0(\boldsymbol{u}) + \sum_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} \boldsymbol{A}_{\boldsymbol{q}} \boldsymbol{X}_{\boldsymbol{q}}(\boldsymbol{u}) \tag{3.48}$$

$$\begin{bmatrix} Y_{1,i}^{us}(\boldsymbol{u}) \\ \dots \\ Y_{K,i}^{us}(\boldsymbol{u}) \end{bmatrix} = \sqrt{b_0} \begin{bmatrix} X_{0,1,i}(\boldsymbol{u}) \\ \dots \\ X_{0,K,i}(\boldsymbol{u}) \end{bmatrix} + \sqrt{2} \sum_{\boldsymbol{q}=-\boldsymbol{Q}}^{\boldsymbol{Q}} \begin{bmatrix} a_{\boldsymbol{q},11} & \dots & a_{\boldsymbol{q},1K} \\ \dots & & \dots \\ a_{\boldsymbol{q},K1} & \dots & a_{\boldsymbol{q},KK} \end{bmatrix} \cdot \begin{bmatrix} cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{u} + \phi_{\boldsymbol{q},1,i}) \\ \dots \\ cos(\pi \boldsymbol{q}^T \boldsymbol{S}^{-1} \boldsymbol{u} + \phi_{\boldsymbol{q},K,i}) \end{bmatrix} \tag{3.49}$$

The example below is provided to show grid-free simulation of the bivariate system Y' consisting of random functions Y_1' and Y_2'. The simulation is resolved on the grid same as grid 2 in the univariate grid-free simulation of Section 3.1.2: evenly spaced 80 × 50 simulation nodes rotated by -45° azimuth and separated by 1.0 × 1.0 units^2 from each other. The covariance function of the system is provided by Eq. (3.50), which is used in the simulation. The first two realizations of each random function from 100 realization pool are shown in Fig. 3.23. The realizations show the anisotropy and correlation between realizations of different

random functions. Histograms of the first realization and all realizations of each random function are shown in Fig. 3.24. Histograms of individual realizations fluctuate, yet the histograms of all realizations follow the expected normal distribution, which is confirmed by a normal probability plots on the same figure. Note the means and standard deviations are close to 0.0 and 1.0, respectively. The relationship between the realizations of the two random functions is checked through a scatter plot. Fig. 3.25 presents a bivariate plot between first realizations and all realizations of the random functions. The scatter plots show a bivariate Gaussian nature of the simulated random functions. The bivariate relationship is preserved better for the case when all realizations are plotted. The correlation coefficient is 0.71, which is close to the target 0.70. Finally, Fig. 3.26 contains experimental direct- and cross-variograms computed for each realization and their average for major and minor directions of continuity along with the variogram models. The average experimental variograms closely reproduce variogram models. The spread between individual experimental variograms are quite significant due to the ergodic statistical fluctuations.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_1}(\boldsymbol{h}) = Sph_{\substack{r_1=20.0 \\ r_2=10.0 \\ \alpha=120^\circ}}(\boldsymbol{h}) \\ C_{Y_1Y_2}(\boldsymbol{h}) = 0.7 Sph_{\substack{r_1=20.0 \\ r_2=10.0 \\ \alpha=120^\circ}}(\boldsymbol{h}) \\ C_{Y_2}(\boldsymbol{h}) = Sph_{\substack{r_1=20.0 \\ r_2=10.0 \\ \alpha=120^\circ}}(\boldsymbol{h}) \end{cases} \tag{3.50}$$

3.2 Turning Lines Simulation

The issue with FSS is the computational time required for 2-D and 3-D systems. The number of Fourier coefficients becomes very large for higher dimensions. If the 1-D multivariate case requires $K(2Q+1)$ coefficients to generate single realization without a nugget effect, this number increases to $K(2Q+1)^2$ and

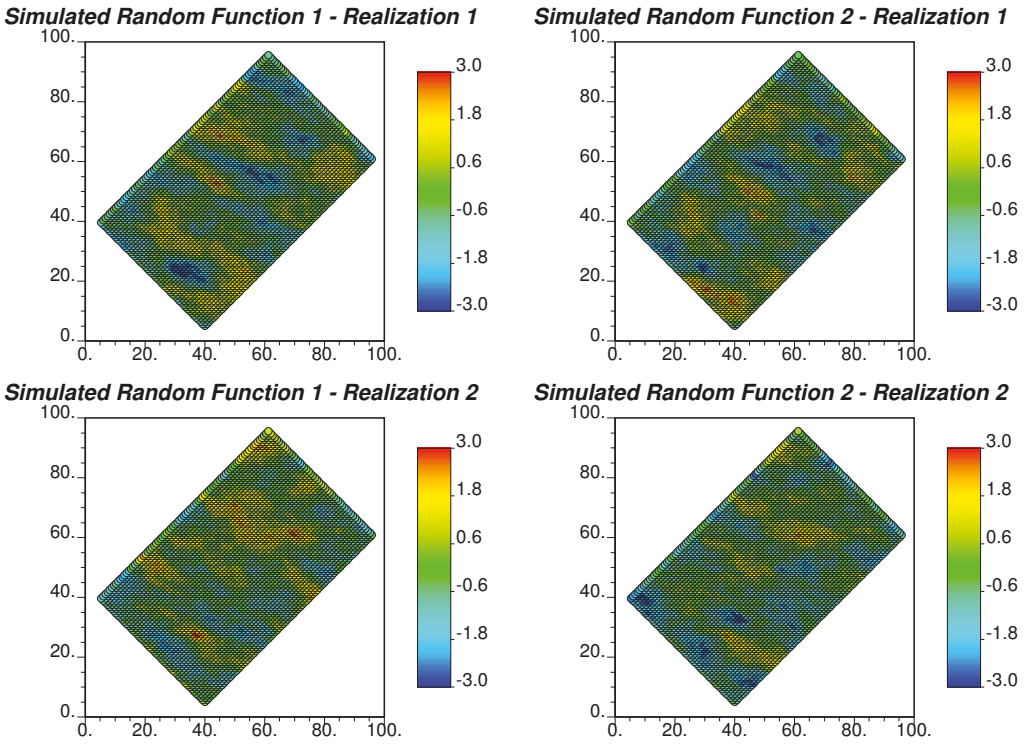


Figure 3.23: Multivariate 2-D Fourier series simulation example – realizations 1 and 2 maps of random functions Y_1 and Y_2 of the bivariate system.

$K(2Q+1)^3$ for 2-D and 3-D multivariate cases, respectively. The computational time can be improved by using the turning bands simulation approach to combine 1-D line simulations into higher dimensional simulations. The number of lines required to generate a simulation is much smaller than the number of Fourier coefficients required for each dimension. The bands are replaced with the points, when 1-D line simulations are performed with the FSS.

3.2.1 Turning Lines

The turning bands simulation method has been briefly mentioned in Section 2.3.1 [36, 37]. The random function Y' is generated in 2-D or 3-D by combining independent 1-D stochastic processes. In the conventional TBS, the 1-D line

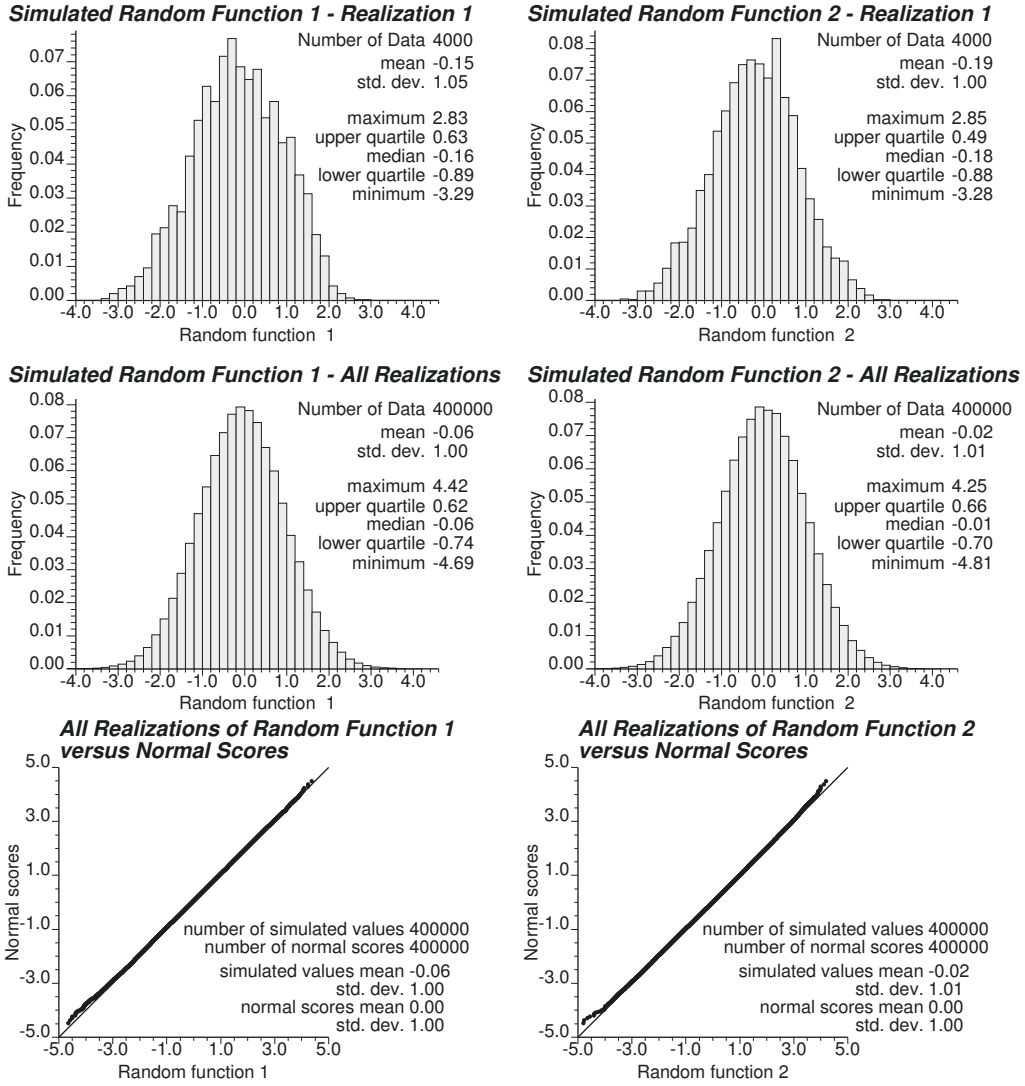


Figure 3.24: Multivariate 2-D Fourier series simulation example – histograms of first realizations and all realizations along with the normality check of all realizations of random functions Y_1 and Y_2 of the bivariate system.

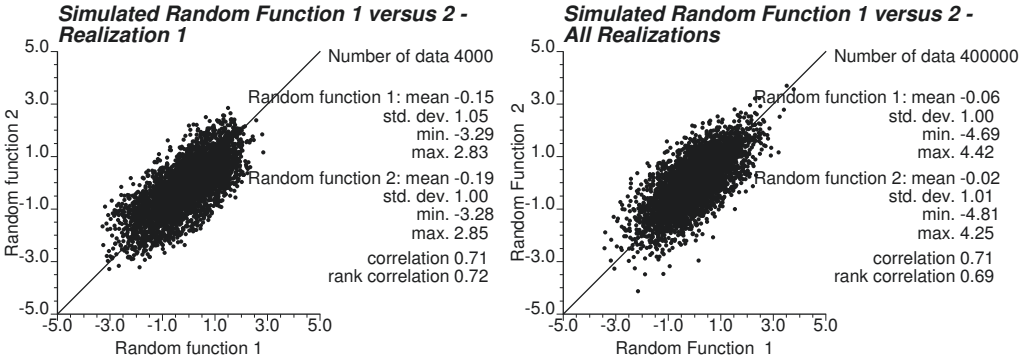


Figure 3.25: Multivariate 2-D Fourier series simulation example – scatter plots between first realizations and all realizations of random functions Y_1 and Y_2 of the bivariate system.

processes are generated with a moving average and, therefore, have a gridded nature. A discretization interval in the moving average simulation is called the band. The bands are spaced equally along the lines. The projections of the simulation location $\boldsymbol{u}$ onto the lines are adjusted to the bands. In the proposed grid-free simulation paradigm, turning bands are coupled with the FSS to generate independent random factors $X_p, p = 1, ..., P$, in a more precise manner. These random factors are combined together through LMR or LMC to simulate one or more random functions Y'. The procedure is explained in detail in Sections 3.2.4 and 3.2.5 for univariate and multivariate cases, respectively. Because, the FSS simulates as a function of the simulation location, there is no longer any need to divide the 1-D line stochastic processes into bands, but rather present these processes continuously over turning lines [36]. Therefore, the turning bands method is renamed as turning lines simulation (TLS) when the FSS is applied. In the turning lines simulation, the bands are replaced with points. There is no need for a discretization of the 1-D line simulations.

The TLS is an efficient way for higher dimensional simulation of independent random factor X with zero mean and unit variance in normal space through

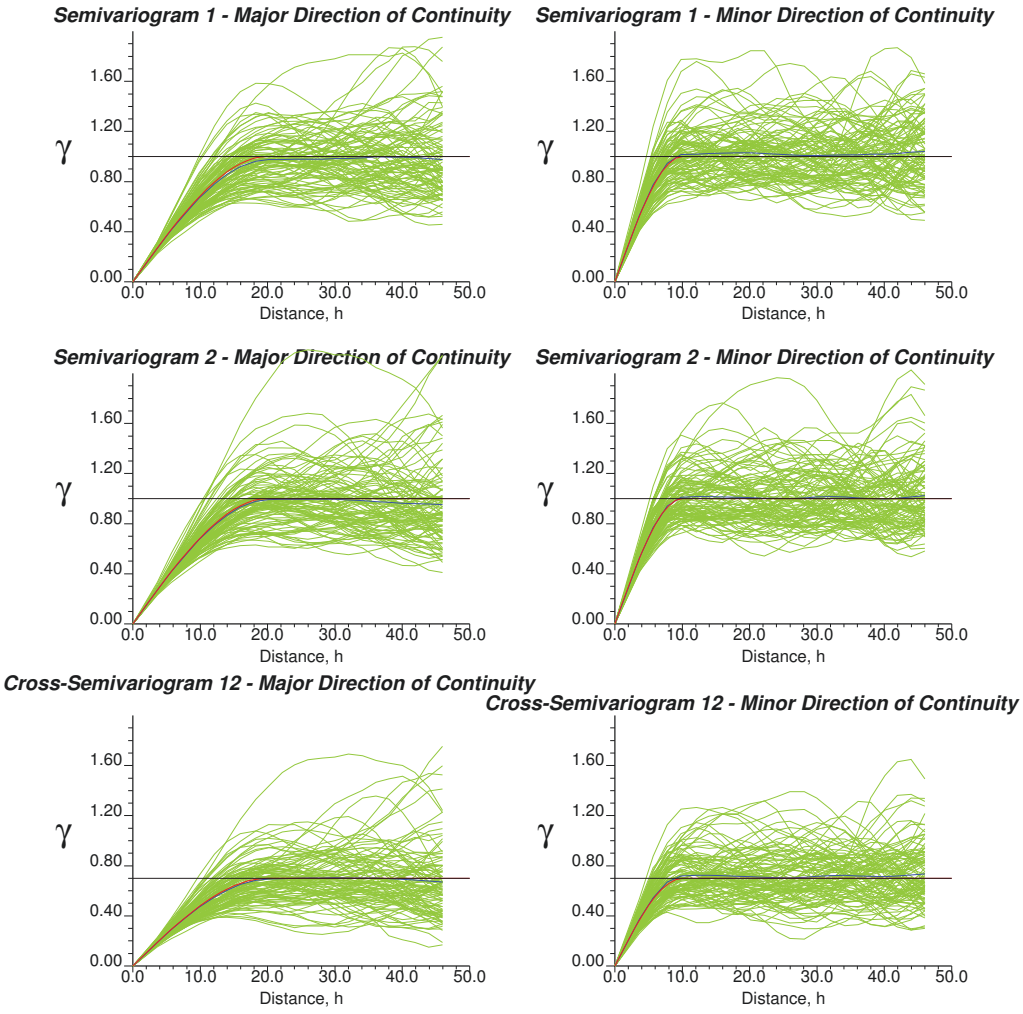


Figure 3.26: Multivariate 2-D Fourier series simulation example – experimental direct- and cross-variograms (in green) computed for all realizations and their average experimental variograms (in red) along with the variogram model (in dark blue) of random functions Y_1 and Y_2 of the bivariate system for major and minor directions of continuity.

a weighted sum of independent 1-D line stochastic processes $X^{(1)} = \{X_l^{(1)},\ l = 1,\ ...,\ L\}$ of a particular covariance function $C_{X^{(1)}}(h)$ as shown in Eq. (3.51). A total number of the turning lines is L, which is chosen to reproduce Y' precisely (see Section 5.3.2 in Chapter 5 for choice of optimal number L of turning lines). The simulated value $X(\boldsymbol{u})$ is found as a summation of the equally weighted line simulations $X_l^{(1)}(\boldsymbol{u}), l = 1, ..., L$, values of which are retrieved at the projected location $\boldsymbol{u}$ on these lines with directional unit vectors $\hat{\boldsymbol{u}}_l, l = 1, ..., L$. It is assumed that all lines intersect each other at the origin of the coordinate system. Fig. 3.27 shows the projection of the simulation location $\boldsymbol{u}$ on line l with directional unit vector $\hat{\boldsymbol{u}}_l$. The scalar product $\boldsymbol{u} \cdot \hat{\boldsymbol{u}}_l$ of two location vectors is computed as shown in Eq. (3.52), where location vector $\boldsymbol{u}$ is represented by three Cartesian coordinates $\{x,\ y,\ z\}$. The unit vector $\hat{\boldsymbol{u}}_l$ of magnitude one ($||\hat{\boldsymbol{u}}_l|| = 1$) is also represented by three Cartesian components $\{\hat{x}_l,\ \hat{y}_l,\ \hat{z}_l\}$. The simulated value $X_l^{(1)}(\boldsymbol{u} \cdot \boldsymbol{u}_l)$ of any line process $X_l^{(1)}$ is tied precisely to the local coordinate on this line, which is expressed as a product of magnitude $||\boldsymbol{u}||$ of the simulation location $\boldsymbol{u}$ and cosine of angle α_l between simulation location vector $\boldsymbol{u}$ and direction $\hat{\boldsymbol{u}}_l$ of line process l.

$$X(\boldsymbol{u}) = \frac{1}{\sqrt{L}} \sum_{l=1}^{L} X_l^{(1)}(\boldsymbol{u} \cdot \hat{\boldsymbol{u}}_l) \tag{3.51}$$

$$\boldsymbol{u} \cdot \hat{\boldsymbol{u}}_l = ||\boldsymbol{u}|| \cos \alpha_l = \frac{|x\hat{x}_l + y\hat{y}_l + z\hat{z}_l|}{\sqrt{\hat{x}_l^2 + \hat{y}_l^2 + \hat{z}_l^2}} \tag{3.52}$$

The turning lines should be evenly distributed in the space to avoid artifacts in the simulation. For the 2-D space, the lines are placed in such positions, which ensure that the perimeter of the circle with unit radius is divided equally. Only one angle θ_l in range of $[0 - \pi]$ is enough to define the location of line l. This angle is measured in a clockwise fashion from the Y axis. Therefore, the location of the line in 2-D space is computed evenly as shown in Eq. (3.53) and Fig.

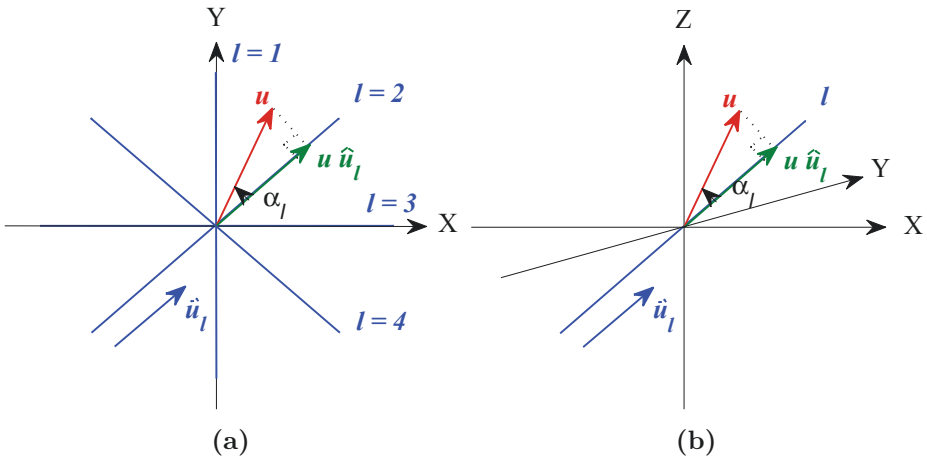


Figure 3.27: Projection of the simulation location $\boldsymbol{u}$ onto a turning line – (a) in 2-D, and (b) in 3-D.

3.27. The Cartesian coordinates $\hat{x}_l$ and $\hat{y}_l$ of the unit vector $\hat{\boldsymbol{u}}_l$ of the line l are computed as presented in Eq. (3.54). The Cartesian coordinate in the direction Z is omitted arbitrarily for 2-D space. To be more precise, any two coordinates along two coordinate axes are considered, while the third one is discarded.

$$\theta_l = (l-1)\frac{\pi}{L},\ l = 1, ..., L \tag{3.53}$$

$$\hat{\boldsymbol{u}}_l = \begin{cases} \hat{x}_l = ||\hat{\boldsymbol{u}}_l|| \sin\theta_l \\ \hat{y}_l = ||\hat{\boldsymbol{u}}_l|| \cos\theta_l \end{cases},\ l = 1, ..., L \tag{3.54}$$

It is a challenge to place turning lines evenly in 3-D space. Recall that all lines intersect at the coordinate system origin. Two angles $\theta_{1,l}$ and $\theta_{2,l}$ are required to specify the location of each line l in the 3-D space. The first angle $\theta_{1,l}$ is measured in the same way as angle θ_l for 2-D systems. The second angle $\theta_{2,l}$ is measured clockwise from the Z axis. Once the angles for evenly spaced lines are defined, the corresponding coordinates $\hat{x}_l$, $\hat{y}_l$, and $\hat{z}_l$ of the unit vector $\hat{\boldsymbol{u}}_l$ of the line l can be computed as presented in Eq. (3.55).

$$\hat{\boldsymbol{u}}_l = \begin{cases} \hat{x}_l = ||\hat{\boldsymbol{u}}_l|| \sin\theta_{2,l} \sin\theta_{1,l} \\ \hat{y}_l = ||\hat{\boldsymbol{u}}_l|| \sin\theta_{2,l} \cos\theta_{1,l} \\ \hat{z}_l = ||\hat{\boldsymbol{u}}_l|| \cos\theta_{2,l} \end{cases} , \; l = 1, ..., L \tag{3.55}$$

Alternatively, the vertices of an icosahedron and its subdivision can be used to define directions of the lines that would evenly cut the surface of the sphere with unit radius in 3-D [29, 38, 79]. That is the intersection points formed between lines and the unit sphere surface are placed evenly on the sphere surface, and lines themselves are approximately divide 3-D space in quasi-even sectors. The icosahedron and two levels of its subdivision are shown in Fig. 3.28, which represent an approximate partition of the unit sphere surface with equilateral triangles. The icosahedron consists of 20 equilateral triangles. All 20 triangles share 12 vertices that are placed exactly on the surface of the unit sphere. Therefore, the icosahedron vertices divide the unit sphere surface equally by $L_0 = 12$ lines originated in the centre of the unit sphere. The locations of vertices on a unit sphere in a Cartesian coordinate system are defined by a combination of zero, two numbers, and their positive and negative values (0.0, $\pm$0.85065, $\pm$0.52573), where the ratio of the last two is a golden ratio $\frac{1+\sqrt{5}}{2} \approx$ 1.6180. The resulting coordinates $\hat{x}_{l_0}$, $\hat{y}_{l_0}$, and $\hat{z}_{l_0}$ of the icosahedron vertices are summarized in Table 3.1 [12]. Note that projections of the simulation location on the lines are the same for pair of lines that originate at sphere center in opposite directions. Therefore, it would have been enough to use only half of the vertices, which define a semi-sphere, in the definition of the line direction in the turning lines simulation. For instance, first six vertices from 3.1 could be good candidates for definition of line directions, because all of them lie in a semi-sphere region. To avoid confusion of the choice of the lines in the subsequent subdivision of the icosahedron, all vertices are used in the definition of the lines. This ignorance of the overlapping lines do not lead to any harm to simulation as shown later in

many examples.

Each face of the icosahedron has the shape of an equilateral triangle that can be subdivided into four equal smaller equilateral triangles. The coordinates of new vertices are computed first as the coordinates $\hat{x}^*_{l_1}$, $\hat{y}^*_{l_1}$, and $\hat{z}^*_{l_1}$ of a middle point, which divides any two adjacent vertices of icosahedron, as shown in Eq. (3.56). Because all new and old vertices should fall on the surface of the unit sphere, the coordinates of the middle points are adjusted to the final coordinates $\hat{x}_{l_1}$, $\hat{y}_{l_1}$, and $\hat{z}_{l_1}$ of unit vectors of evenly placed lines in 3-D as presented in Eq. (3.57). The resulting triangles define a first-level subdivision of the icosahedron and form a geometric shape closer to the unit sphere. The total number of the lines $L_1 = 42$ for first-level subdivision of the icosahedron is equal to the number of previous vertices 12 and number of newly obtained vertices 30. Each new equilateral triangle can be further subdivided into four even smaller triangles. Coordinates $\hat{x}_{l_2}$, $\hat{y}_{l_2}$, and $\hat{z}_{l_2}$ of new vertices of the second-level icosahedron subdivision are found by applying Eqs. (3.58) and (3.59). The total number of turning lines, which cut a 3-D unit sphere surface in a quasi-even fashion, is $L_2 = 162$. This number includes 12 vertices of original icosahedron, 30 additional vertices of first-order icosahedron subdivision, and 120 vertices of second-order icosahedron subdivision. The subdivision can be continued further. Note that angles between lines are not exactly the same, but close to each other [10]. It is shown in later chapters that a larger number of the turning lines improves the quality of the simulated model, but increases computational time.

$$\hat{\boldsymbol{u}}^*_{l_1} = \begin{cases} \hat{x}^*_{l_1} = \frac{\hat{x}_{l_0} + \hat{x}_{l'_0}}{2} \\ \hat{y}^*_{l_1} = \frac{\hat{y}_{l_0} + \hat{y}_{l'_0}}{2} \\ \hat{z}^*_{l_1} = \frac{\hat{z}_{l_0} + \hat{z}_{l'_0}}{2} \end{cases}, \; l_1 = 1, ..., 30, \; l_0, l'_0 \in [1, 12] \tag{3.56}$$

Table 3.1: Cartesian coordinates of the icosahedron vertices.

Vertex	$\hat{x}_{l_0}$	$\hat{y}_{l_0}$	$\hat{z}_{l_0}$
1	-0.52573	0.0	0.85065
2	0.52573	0.0	0.85065
3	0.0	0.85065	0.52573
4	-0.85065	0.52573	0.0
5	-0.85065	-0.52573	0.0
6	0.0	-0.85065	0.52573
7	0.85065	-0.52573	0.0
8	0.85065	0.52573	0.0
9	0.0	0.85065	-0.52573
10	-0.52573	0.0	-0.85065
11	0.0	-0.85065	-0.52573
12	0.52573	0.0	-0.85065

$$\hat{\boldsymbol{u}}_{l_1} = \begin{cases} \hat{x}_{l_1} = \frac{\hat{x}^*_{l_1}}{\sqrt{(\hat{x}^*_{l_1})^2+(\hat{y}^*_{l_1})^2+(\hat{z}^*_{l_1})^2}} \\ \hat{y}_{l_1} = \frac{\hat{y}^*_{l_1}}{\sqrt{(\hat{x}^*_{l_1})^2+(\hat{y}^*_{l_1})^2+(\hat{z}^*_{l_1})^2}} \\ \hat{z}_{l_1} = \frac{\hat{z}^*_{l_1}}{\sqrt{(\hat{x}^*_{l_1})^2+(\hat{y}^*_{l_1})^2+(\hat{z}^*_{l_1})^2}} \end{cases} , \; l_1 = 1, ..., 30 \tag{3.57}$$

$$\hat{\boldsymbol{u}}^*_{l_2} = \begin{cases} \hat{x}^*_{l_2} = \frac{\hat{x}_{l_{[0,1]}}+\hat{x}_{l'_{[0,1]}}}{2} \\ \hat{y}^*_{l_2} = \frac{\hat{y}_{l_{[0,1]}}+\hat{y}_{l'_{[0,1]}}}{2} \\ \hat{z}^*_{l_2} = \frac{\hat{z}_{l_{[0,1]}}+\hat{z}_{l'_{[0,1]}}}{2} \end{cases} , \; l_2 = 1, ..., 120, \; l_{[0,1]}, l'_{[0,1]} \in [1, 42] \tag{3.58}$$

$$\hat{\boldsymbol{u}}_{l_2} = \begin{cases} \hat{x}_{l_2} = \frac{\hat{x}^*_{l_2}}{\sqrt{(\hat{x}^*_{l_2})^2+(\hat{y}^*_{l_2})^2+(\hat{z}^*_{l_2})^2}} \\ \hat{y}_{l_2} = \frac{\hat{y}^*_{l_2}}{\sqrt{(\hat{x}^*_{l_2})^2+(\hat{y}^*_{l_2})^2+(\hat{z}^*_{l_2})^2}} \\ \hat{z}_{l_2} = \frac{\hat{z}^*_{l_2}}{\sqrt{(\hat{x}^*_{l_2})^2+(\hat{y}^*_{l_2})^2+(\hat{z}^*_{l_2})^2}} \end{cases} , \; l_2 = 1, ..., 120 \tag{3.59}$$

Once the directions of the lines are defined, the values of line stochastic

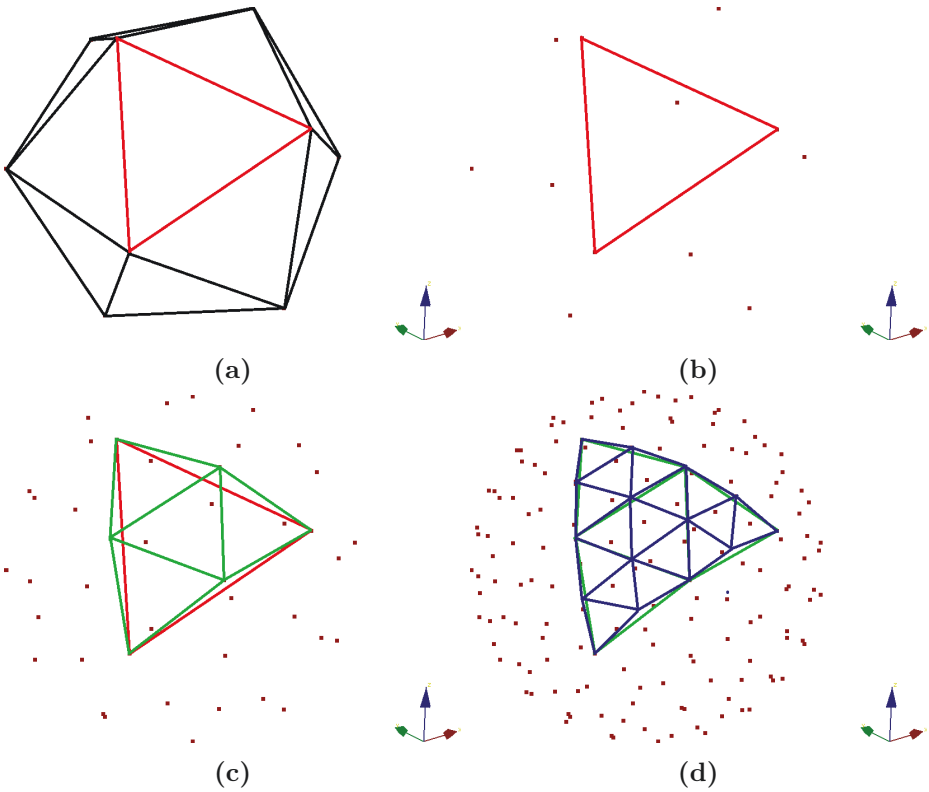


Figure 3.28: Intersection points of evenly placed lines with unit sphere surface: (a) icosahedron, (b) partition of the sphere surface of unit radius with 12 evenly placed lines that cut sphere at icosahedron vertices, (c) partition of the sphere surface of unit radius with 42 evenly placed lines that cut sphere at icosahedron vertices subdivided once, and (d) partition of the sphere surface of unit radius with 162 evenly placed lines that cut sphere at icosahedron vertices subdivided twice.

processes $X_l^{(1)}, l = 1, ..., L$, are simulated at the required locations that correspond to a projection of the simulation location $\boldsymbol{u}$ on 1-D lines. The form of a line process $X_l^{(1)}$ is defined by a relationship between the covariance $C_{X^{(1)}}(h)$ of this line process and the covariance $C_X(h)$ of target random factor X. Eq. (3.60) shows how covariances are related to each other for 2-D systems, and Eq. (3.61) establishes covariance relationship for 3-D processes. The derivation of these relationships are elegantly explained in [47]. To summarize, the line processes

$X_l^{(1)}, l = 1, ..., L$, are generated unconditionally by the FSS with a particular covariance function $C_{X^{(1)}}(h)$, which is derived from the covariance function of the random factor X. Then, these 1-D line processes are used in Eq. (3.51) to simulate the random factor X.

$$C_X(h) = \frac{2}{\pi} \int_0^h \frac{C_{X^{(1),2-D}}(h')}{\sqrt{h^2 - h'^2}} dh' \tag{3.60}$$

$$C_{X^{(1),3-D}}(h) = \frac{d}{dh} \left(h C_X(h) \right) \tag{3.61}$$

Because the turning bands/lines method allows to represent only isotropic random functions X, a coordinate transformation of the simulation domain is proposed to account for the anisotropy. This is discussed later in Section 3.2.3.

3.2.2 Covariance Function of Line Processes

Covariances of 1-D line stochastic processes are provided for the most common covariances used in the simulation of 2-D and 3-D systems. The derivation of the 1-D covariances $C_{X^{(1)}}(h)$ can be found in [24] based on Eqs. (3.60) and (3.61). The target covariance of random factor X can be spherical, exponential, or Gaussian. Corresponding 1-D line covariances for simulation of the 2-D and 3-D random functions are shown in Eq. (3.62) and Eq. (3.63) for spherical covariance function, in Eq. (3.64) and Eq. (3.65) for exponential covariance function, and in Eq. (3.66) and Eq. (3.67) for Gaussian covariance function. In these equations $I_0(h)$ is the modified Bessel function of zero order, and $L_0(h)$ is the modified Struve function of zero order. The plots of these 1-D covariance functions along with the target covariance functions are shown in Figs. 3.29 – 3.31.

$$Sph_{X^{(1)},2-D}(h) = \begin{cases} 1 - \frac{3\pi|h|}{8r}\left(2 - \left(\frac{h}{r}\right)^2\right), & |h| \leq r \\ 1 - \frac{3|h|}{4r}\left(\left(2 - \left(\frac{h}{r}\right)^2\right)\arcsin\left(\frac{r}{|h|}\right) + \sqrt{\left(\frac{h}{r}\right)^2 - 1}\right), & |h| > r \end{cases} \tag{3.62}$$

$$Sph_{X^{(1)},3-D}(h) = \begin{cases} 1 - \frac{3|h|}{r} + 2\left(\frac{|h|}{r}\right)^3, & |h| \leq r \\ 0, & |h| > r \end{cases} \tag{3.63}$$

$$Exp_{X^{(1)},2-D}(h) = 1 - \frac{3\pi|h|}{2r}\left(I_0\left(\frac{3|h|}{r}\right) - L_0\left(\frac{3|h|}{r}\right)\right) \tag{3.64}$$

$$Exp_{X^{(1)},3-D}(h) = \left(1 - \frac{3|h|}{r}\right)\exp\left(-\frac{|3|h}{r}\right) \tag{3.65}$$

$$Gauss_{X^{(1)},2-D}(h) = 1 - \frac{2\sqrt{3}|h|}{r}\exp\left(-3\left(\frac{h}{r}\right)^2\right)\int_0^{\frac{\sqrt{3}|h|}{r}}\exp\left(\left(h'\right)^2\right)dh' \tag{3.66}$$

$$Gauss_{X^{(1)},3-D}(h) = \left(1 - 6\left(\frac{h}{r}\right)^2\right)\exp\left(-3\left(\frac{h}{r}\right)^2\right) \tag{3.67}$$

3.2.3 Anisotropic Simulation

The turning lines approach does not produce anisotropic processes in its current form. For this reason, the coordinates of the simulation nodes are modified to enable simulation of anisotropic fields [56]. Two modifications are required for the coordinates of the simulation locations $\boldsymbol{u}$ to properly account for anisotropy in the simulation: stretching and rotation operations. These two operations can be expressed as the matrix product between an operation matrix and a location vector. This modification is performed for every anisotropic structure, or, in

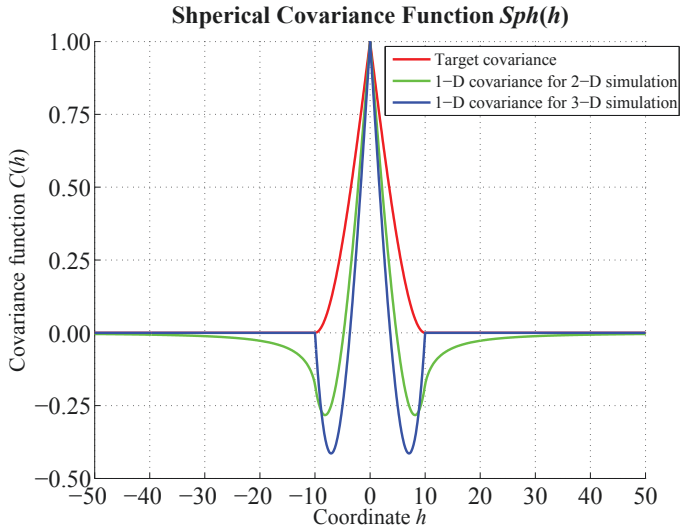


Figure 3.29: Spherical covariance function and its 1-D line covariance functions equivalents for 2-D and 3-D simulations.

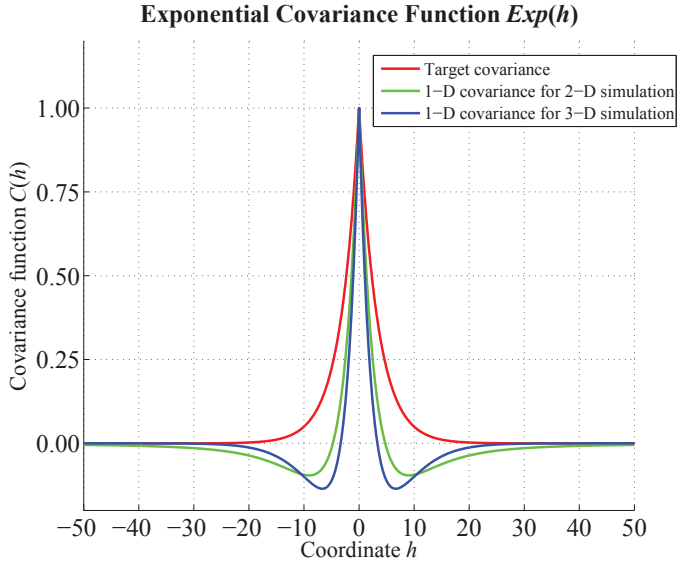


Figure 3.30: Exponential covariance function and its 1-D line covariance functions equivalents for 2-D and 3-D simulations.

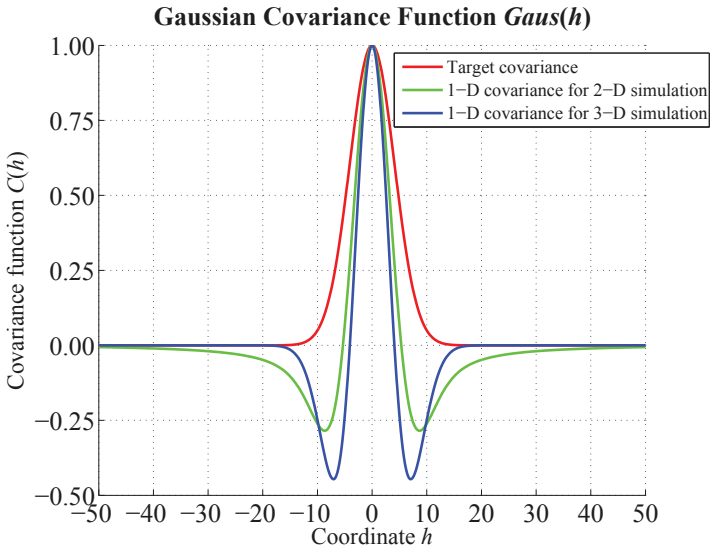


Figure 3.31: Gaussian covariance function and its 1-D line covariance functions equivalents for 2-D and 3-D simulations.

other words, for each random factor X_p in the LMR/LMC of the target random function Y.

The original simulation location $\boldsymbol{u}$ is represented by three Cartesian coordinates as shown in Eq. (3.68). However, the actual simulation is performed at location $\boldsymbol{u}'$ to address anisotropy. The coordinates of the modified location are presented in Eq. (3.69). The resulting simulated value is tied to the original coordinates $\boldsymbol{u}$. Two matrix operations are applied to the original location vector $\boldsymbol{u}$ to get $\boldsymbol{u}'$ as Eq. (3.70) states. The stretching matrix $\boldsymbol{M}_S$ is a diagonal matrix, elements of which are represented by a ratio of the variogram ranges $r_i, i = 1, 2, 3$, to the variogram range r_2 as presented in Eq. (3.71). The rotation matrix $\boldsymbol{M}_R$ is a product of rotation matrices $\boldsymbol{M}_{R_X}$, $\boldsymbol{M}_{R_Y}$, and $\boldsymbol{M}_{R_Z}$ around X, Y, and Z axes. The details are provided in Eqs. (3.72) – (3.75). Angles α_a, α_p, and α_d are the angles of the maximum direction of the continuity in the space. The isotropic simulation with variogram range r_1 is performed at simulation locations with

transformed coordinates.

$$\boldsymbol{u} = [x \;\; y \;\; z]^T \tag{3.68}$$

$$\boldsymbol{u}' = [x' \;\; y' \;\; z']^T \tag{3.69}$$

$$\boldsymbol{u}' = \boldsymbol{M}_S \boldsymbol{M}_R \boldsymbol{u} \tag{3.70}$$

$$\boldsymbol{M}_S = \begin{bmatrix} \frac{r_1}{r_2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{r_1}{r_3} \end{bmatrix} \tag{3.71}$$

$$\boldsymbol{M}_R = \boldsymbol{M}_{R_X} \boldsymbol{M}_{R_Y} \boldsymbol{M}_{R_Z} \tag{3.72}$$

$$\boldsymbol{M}_{R_X} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_p & \sin\alpha_p \\ 0 & -\sin\alpha_p & \cos\alpha_p \end{bmatrix} \tag{3.73}$$

$$\boldsymbol{M}_{R_Y} = \begin{bmatrix} \cos\alpha_d & 0 & \sin\alpha_d \\ 0 & 1 & 0 \\ -\sin\alpha_d & 0 & \cos\alpha_d \end{bmatrix} \tag{3.74}$$

$$\boldsymbol{M}_{R_Z} = \begin{bmatrix} \cos\alpha_a & -\sin\alpha_a & 0 \\ \sin\alpha_a & \cos\alpha_a & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.75}$$

An example of the coordinate transformation to account for anisotropy in the simulation generated by the turning lines is shown in Fig. 3.32. The variogram model has a spherical form with a direction of maximum continuity at 60° of azimuth. The maximum continuity variogram range r_1 is 20 units, and the minimum continuity variogram range r_2 is 10 units. First, the original coordinates of the simulation locations $\boldsymbol{u}$ are defined. Then, the coordinate transformation is applied to get new coordinates $\boldsymbol{u}'$, on which isotropic simulation is performed. The coordinates rotation operator $\boldsymbol{M}_R$ and coordinates stretching operator $\boldsymbol{M}_S$ are applied to get the final coordinates. The coordinates are rotated by 60° counter-clockwise and stretched twice ($r_1/r_2 = 2.0$) in the transformed direction of the minimum continuity to account for the target anisotropic variogram model. Next, the simulation is carried out using turning lines and isotropic variogram model with variogram range 20 units of the maximum direction of the continuity. Finally, the coordinate transformation can be reversed to get simulated anisotropic field at the proper simulation node locations $\boldsymbol{u}$, or simulated values can be simply mapped back to the original simulation node coordinates $\boldsymbol{u}$.

3.2.4 Univariate Simulation

The univarite simulation of anisotropic realizations with the turning lines sums the independent random factors $X_p, p = 1, ..., P$, that have one of three basic covariance function forms – spherical, exponential, or Gaussian form, while random factor X_0 has the nugget effect spatial structure. The equations required for the univarite grid-free simulation are summarized in Eqs. (3.76) – (3.81), where i is the realization index, and $C'_{X_p^{(1)}}$ is the isotropic covariance function of 1-D line process with the variogram range r equal to the variogram model range r_1 in the direction of maximum continuity. The linear model of the regionalization is depicted in Eq. (3.76) for the simulation random variable $Y(\boldsymbol{u})$ at location $\boldsymbol{u}$. The

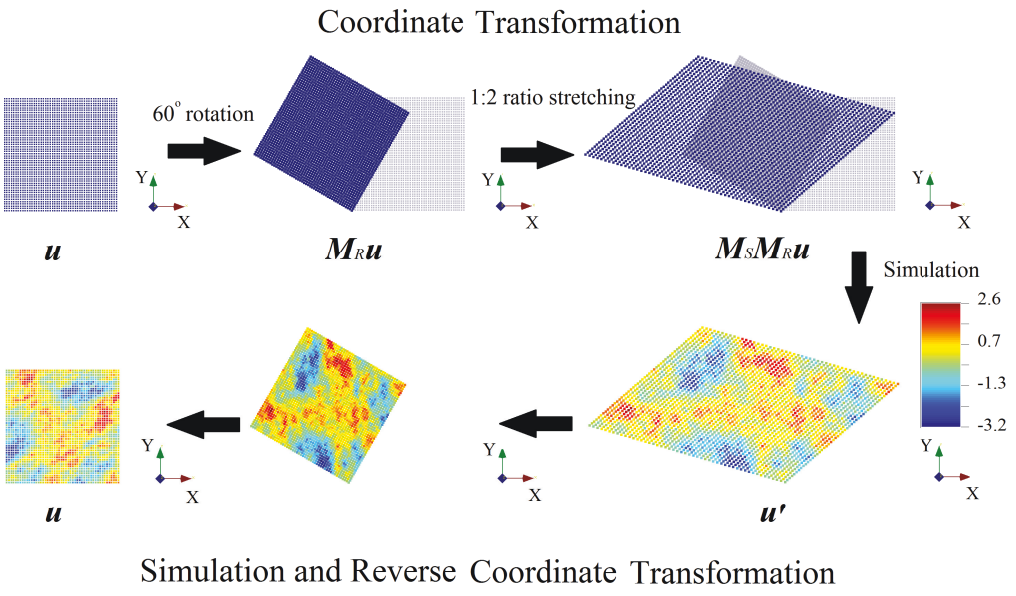


Figure 3.32: Example of the coordinate transformation for the anisotropic simulation with turning lines. Original coordinates are rotated to align with the principal directions of the continuity. Then they are stretched along second and third principal directions of the continuity to match the variogram range in the first direction of continuity. The isotropic simulation is performed at locations with these modified coordinates, where variogram range in the first principal direction is used in the isotropic variogram model. Finally, the simulated values are mapped back to the original locations or coordinate transformation is performed in the reverse order for the simulation nodes.

weighting coefficients are derived as shown in Eq. (3.79) from the contribution coefficients of the input target variogram model $\gamma_Y(\boldsymbol{h})$. The target variogram model in Eq. (3.78) is related to the covariance structure of the simulation in Eq. (3.77). Random factors $X_p, p = 1, ..., P$, are constructed from the 1-D line processes by using the TLS in Eq. (3.80). The 1-D line simulations are generated with the FSS at modified location $\boldsymbol{u}'$ with isotropic variogram model as shown in Eq. (3.81). Here, random phase $\phi_{p,q,l,i}$ depends on the indices of random factor p, Fourier coefficient q, turning line l, and realization i.

$$Y(\boldsymbol{u}) = a_0 X_0(\boldsymbol{u}) + \sum_{p=1}^{P} a_p X_p(\boldsymbol{u}) \tag{3.76}$$

$$C_Y(\boldsymbol{h}) = a_0^2 C_{X_0}(\boldsymbol{h}) + \sum_{p=1}^{P} a_p^2 C_{X_p}(\boldsymbol{h}) \tag{3.77}$$

$$\gamma_Y(\boldsymbol{h}) = b_0 \Gamma_0(\boldsymbol{h}) + \sum_{p=1}^{P} b_p \Gamma_p(\boldsymbol{h}) \tag{3.78}$$

$$a_p = \sqrt{b_p}, \;\; p = 0, ..., P \tag{3.79}$$

$$X_p(\boldsymbol{u}) = \frac{1}{\sqrt{L}} \sum_{l=1}^{L} X_{p,l}^{(1)}(\boldsymbol{u} \cdot \hat{\boldsymbol{u}}_l), \;\; p = 1, ..., P \tag{3.80}$$

$$X_{p,l}^{(1)}(\boldsymbol{u} \cdot \hat{\boldsymbol{u}}_l) = \frac{1}{\sqrt{S}} \sum_{q=-Q}^{Q} cos(\frac{\pi q \boldsymbol{u}' \cdot \hat{\boldsymbol{u}}_l}{S} + \phi_{p,q,l,i}) \sqrt{\int_{-S}^{S} C'_{X_p^{(1)}}(h) \cos(\frac{\pi q h}{S}) dh} \tag{3.81}$$

Note that the resulting sequence of the equations leads to the grid-free simulation of the univariate systems with reduced computational time in comparison with the FSS approach originally presented above. The proposed simulation approach is called grid-free simulation (GFS). In this case, the computational time is proportional to $PL(2Q+1)$ for each realization in comparison to $(2Q+1)^2$ for simulation of 2-D systems with FSS or $(2Q+1)^3$ for simulation of 3-D systems with FSS. The product of the number of random factors P and number of the lines L is usually smaller than $2Q+1$ and definitely smaller than $(2Q+1)^2$.

The example of GFS of a univariate system Y' shown here is similar to the FSS example presented in Section 3.1.2. The grid-free simulation method is used to generated 100 unconditional realizations on a 2-D plane with anisotropic

spherical variogram model shown in Eq. (3.82). The periodic domain $2S$ = 400.0 is chosen to be larger than the simulation domain. The corresponding 1-D isotropic covariance function with variogram range r_1 is discretized on this periodic domain in [-200.0 – 200.0] interval to compute 801 Fourier coefficients g_q, q = -400, ..., 400, which are both shown in Fig. 3.33. Number L of the turning lines is 200.

$$\gamma_Y(\boldsymbol{h}) = Sph_{\substack{r_1=20.0 \\ r_2=10.0 \\ \alpha=120°}}(\boldsymbol{h}) \tag{3.82}$$

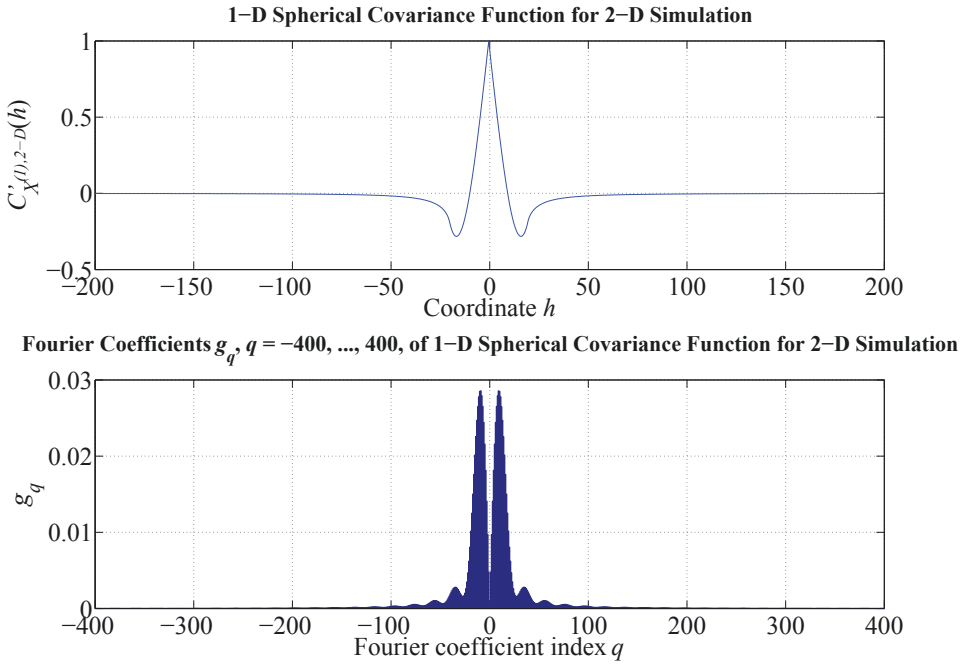


Figure 3.33: Univariate 2-D grid-free simulation example – (a) 1-D spherical target covariance function for 2-D simulation and (b) corresponding Fourier coefficients.

Four grids of various simulation node densities and simulation node distributions are used to resolve the simulation with the GFS, and are shown in Fig.

3.34. These grids are the same as grids presented in the example of previous Section 3.1. The gridding choice does not impose any change to the simulation. The simulation is truly grid-free and just a function of the coordinates of the simulation locations. In other words, the simulation is represented as a spatial function and can be easily recalled at any location in space. The first 4 realizations on a fine evenly structured grid are shown in Fig. 3.35. All four realizations are stochastic with prominent anisotropic features, where the direction of the maximum continuity is at 120° azimuth. Fig. 3.36 contains histograms of the first two realizations and all realizations resolved on the fine grid 2 along with the normality check for all 100 realizations. As mentioned before, individual histograms deviate from normality due to ergodicity (the small simulation domain in comparison with the variogram range). On the other hand, the combined histogram of all realizations looks normal with 0.01 mean and 1.01 variance, which are close to the normal distribution parameters. Maps of the local means and variances are presented in Fig. 3.37 and are complemented with the corresponding histograms. The local means and variances of the realizations are distributed normally around target statistics of 0.0 and 1.0. Fig. 3.38 shows variogram model reproduction for first two realizations, the average experimental variogram, and the average experimental variogram map. The variogram model values in the major and minor directions of continuity are shown as red and blue lines, respectively. The experimental variogram points are computed from the realizations generated on fine grid 2. The experimental variograms computed for individual realizations deviate from variogram model especially for longer ranges due to ergodicity. A cyclic nature to the variograms is present, due to the Fourier series implementation. However, the average experimental variogram computed from all realizations reproduces variogram model quite well. The average experimental variogram map is further evidence of good variogram reproduction.

Note, that there are no any visual artifacts of the turning lines, because of relatively large number of the lines used in the simulation. If a smaller number of the lines were selected in the turning lines simulation method, there would be artifacts present in the simulation and variogram map computed from the realizations [17]. Finally, experimental variograms for all 100 realizations for both major and minor directions of the continuity are shown in Fig. 3.39 with the average experimental variogram and variogram model. The spread of individual variograms is quite significant, partially because they are computed from the unconditional realizations. The average variogram closely follows the variogram model.

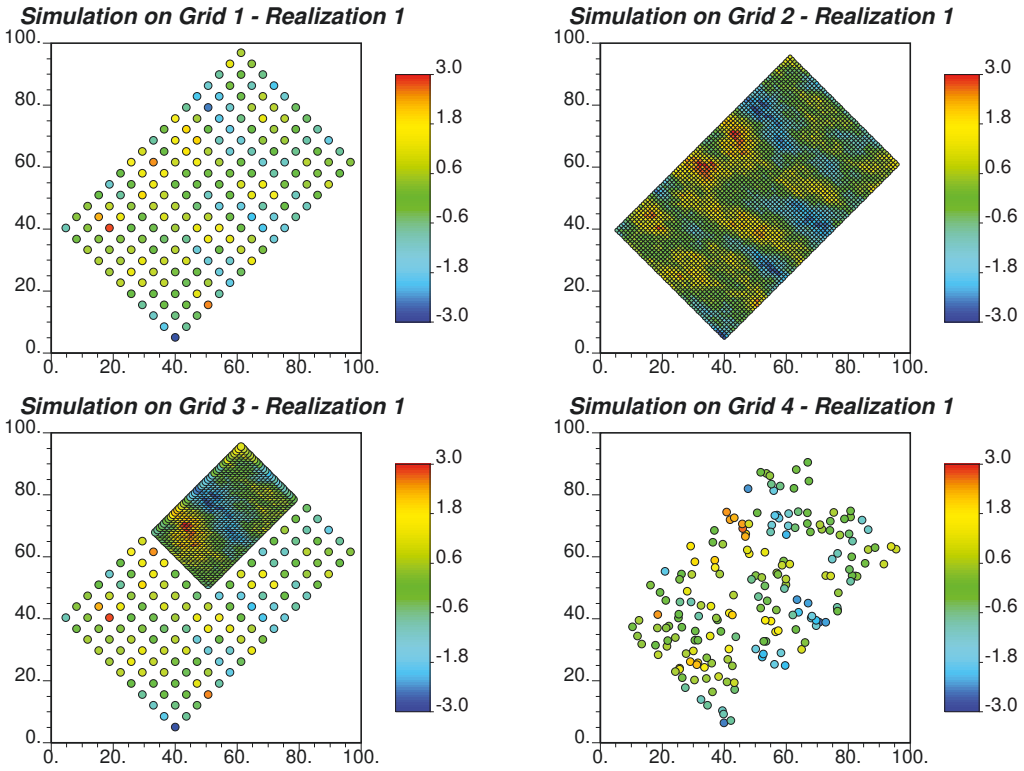


Figure 3.34: Univariate 2-D grid-free simulation example – four distinct simulation grids: (a) coarse evenly structured grid 1, (b) fine evenly structured grid 2, (c) combination of coarse and fine evenly placed nodes on grid 3, and (d) random grid 4.

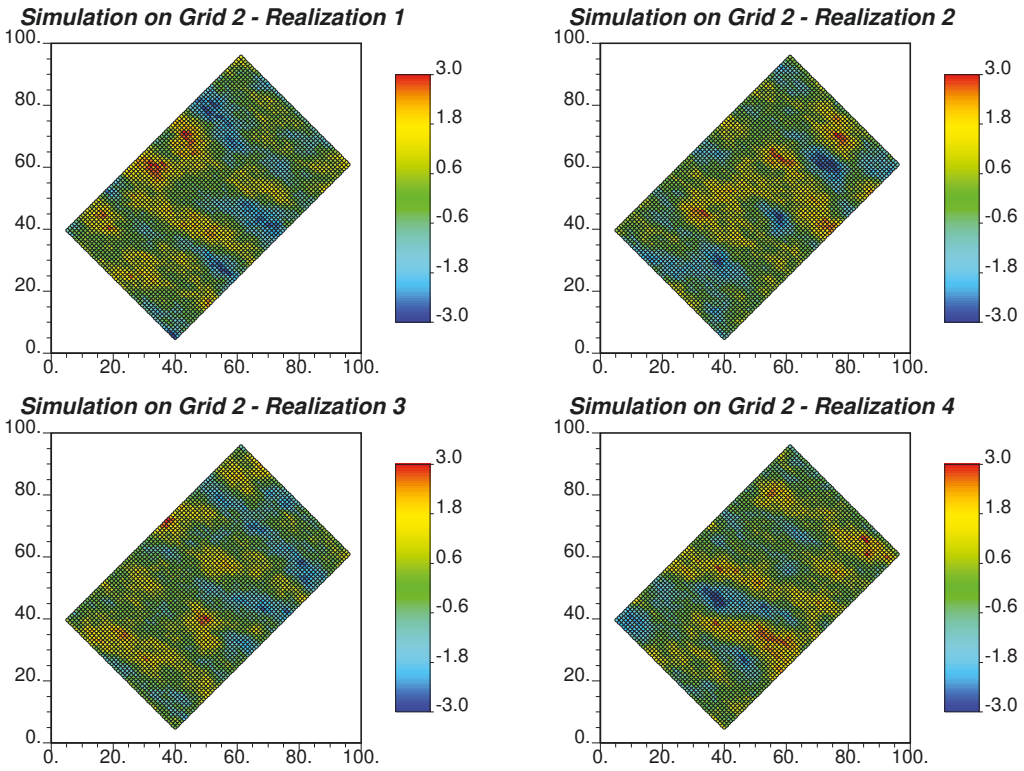


Figure 3.35: Univariate 2-D grid-free simulation example – realizations 1 – 4 on the fine grid 2.

3.2.5 Multivariate Simulation

The multivariate simulation of the anisotropic realizations in a grid-free manner is performed through the LMC shown in Eq. (3.83). The definition of the matrices has been explained in Section 2.1.2 of Chapter 2. The random factors $X_{p,k}(\boldsymbol{u}), p = 1, ..., P, k = 1, ..., K$, which are stored in $\boldsymbol{X}_p(\boldsymbol{u}), p = 1, ..., P$, are computed at location $\boldsymbol{u}$ exactly the same way as random factors $X_p(\boldsymbol{u}), p = 1, ..., P$, have been computed for the univariate case in previous Section 3.2.4. The 1-D line simulations are generated with the FSS and combined through turning lines simulation framework to get random factors. The covariance of

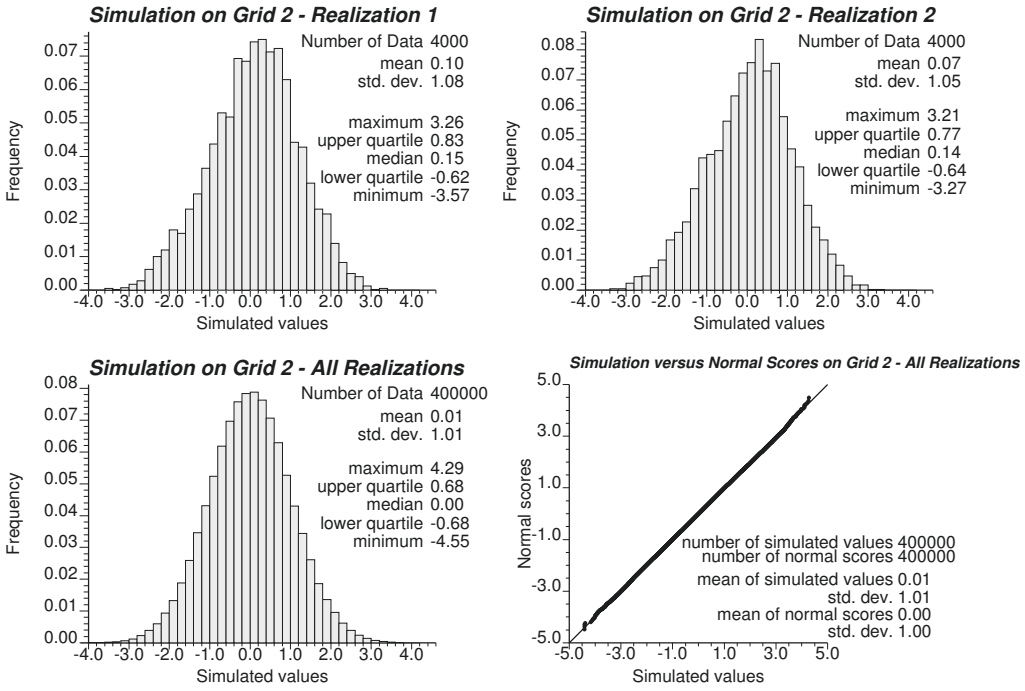


Figure 3.36: Univariate 2-D grid-free simulation example – histograms of realization 1 and 2 and all 100 realizations on the fine grid 2 along with the normal probability plot of all realization values.

the multivariate simulation $\boldsymbol{Y}(\boldsymbol{u})$ can be expressed by Eq. (3.84). Input target variogram model in Eq. (3.85) is used to derive LMC weights according to Eq. (3.86). More details on how to derive the LMC coefficients are provided in Eq. (3.87) through the LU decomposition of the matrix $\boldsymbol{B}_p$ with structural contributions $b_{p,kl}$. The computational cost of single multivariate realization is proportional to $KPL(2Q+1)$.

$$\boldsymbol{Y}(\boldsymbol{u}) = \boldsymbol{A}_0 \boldsymbol{X}_0(\boldsymbol{u}) + \sum_{p=1}^{P} \boldsymbol{A}_p \boldsymbol{X}_p(\boldsymbol{u}) \tag{3.83}$$

$$\boldsymbol{C}_Y(\boldsymbol{h}) = \boldsymbol{A}_0 [\boldsymbol{A}_0]^T C_{X_0}(\boldsymbol{h}) + \sum_{p=1}^{P} \boldsymbol{A}_p [\boldsymbol{A}_p]^T C_{X_p}(\boldsymbol{h}) \tag{3.84}$$

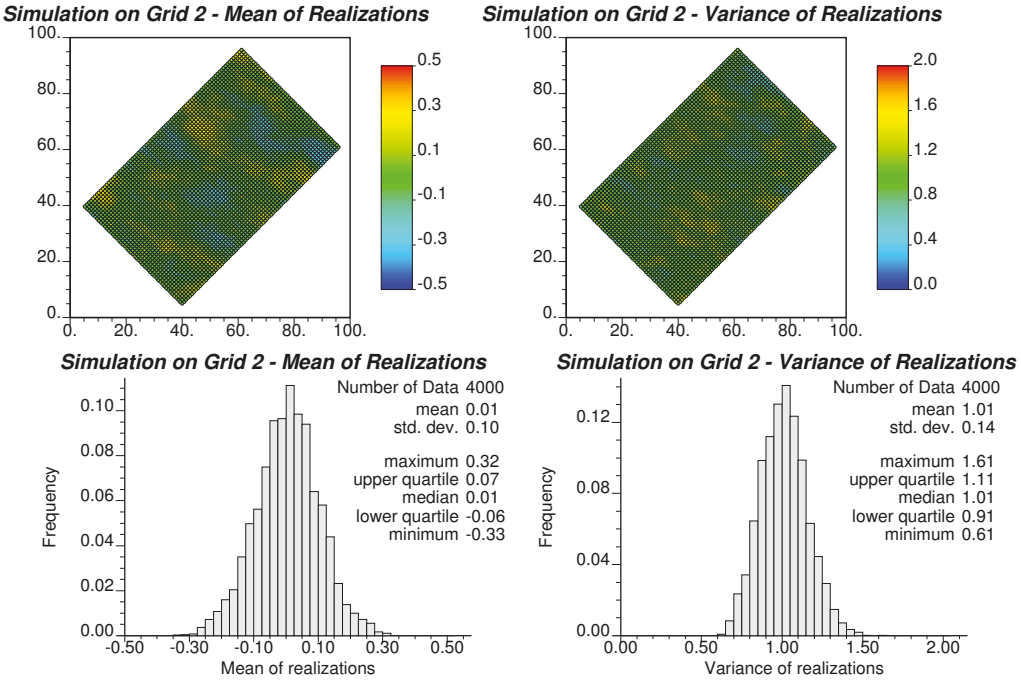


Figure 3.37: Univariate 2-D grid-free simulation example – mean and variance fine grid 2 maps computed from 100 realizations along with the corresponding histograms.

$$\gamma_Y(\boldsymbol{h}) = \boldsymbol{B}_0 \Gamma_{X_0}(\boldsymbol{h}) + \sum_{p=1}^{P} \boldsymbol{B}_p \Gamma_{X_p}(\boldsymbol{h}) \tag{3.85}$$

$$\boldsymbol{B}_p = \boldsymbol{A}_p [\boldsymbol{A}_p]^T, \;\; p = 0, ..., P \tag{3.86}$$

$$a_{p,kl} = \begin{cases} |a_{p,11}| = \sqrt{b_{p,11}} \\ |a_{p,k1}| = \frac{b_{p,k1}}{|a_{p,11}|} \\ |a_{p,kk}| = \sqrt{b_{p,kk} - \sum_{m=1}^{k-1} a_{p,km}^2} \\ |a_{p,kl}| = \frac{b_{p,kl} - \sum_{m=1}^{l-1} |a_{p,km}||a_{p,lm}|}{|a_{p,ll}|} \end{cases}, \; p = 1, ..., P, \; k, l = 1, ..., K \tag{3.87}$$

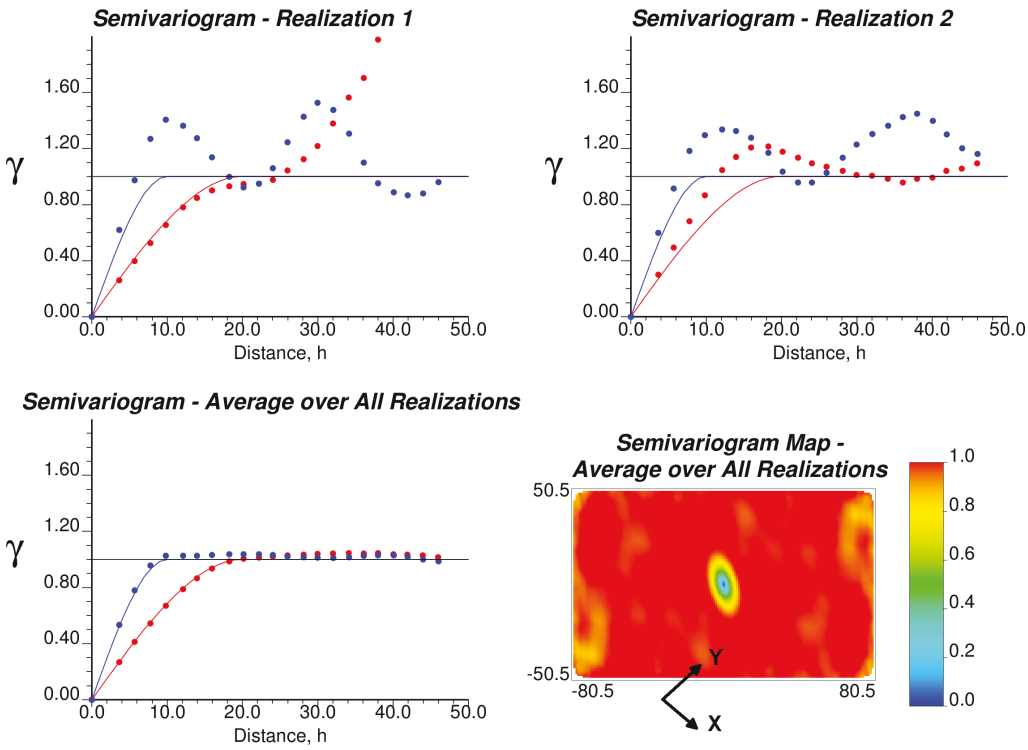


Figure 3.38: Univariate 2-D grid-free simulation example – experimental variograms (dots) for major (in red) and minor (in blue) directions for realizations 1 and 2, average variogram from all 100 realizations simulated on a fine grid 2 and corresponding variogram models (lines), and average variogram map from all 100 realizations.

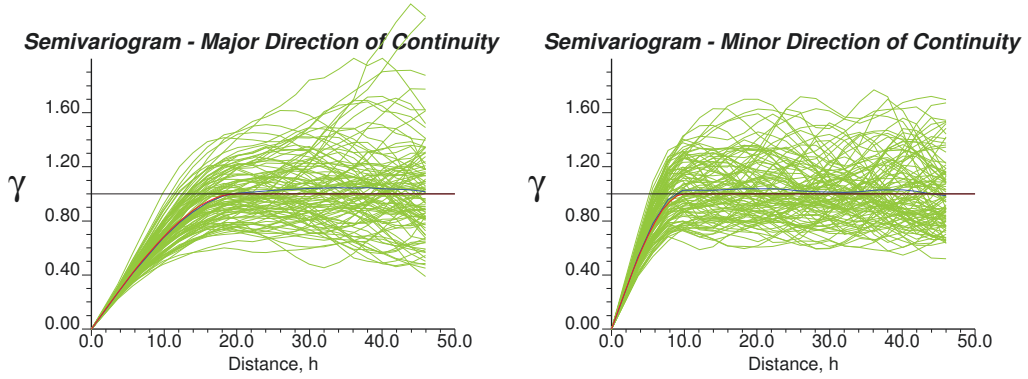


Figure 3.39: Univariate 2-D grid-free simulation example – all 100 experimental variograms (in green) and average experimental variogram (in blue) along with the variogram model (in red) for major and minor directions of continuity.

The example shown below demonstrates the applicability of GFS to multivariate modeling. Here, the bivariate system $Y = \{Y_1, Y_2\}$ is simulated according to the two nested structure variogram model, the covariance form $C_Y(\boldsymbol{h})$ of which is shown in Eq. (3.88) and does not possess nugget. The first variogram structure is represented by an isotropic spherical covariance function with correlation range $r = 5.0$ units. The second structure is presented by an anisotropic covariance function with correlation ranges $r_1 = 30.0$ units and $r_2 = 20.0$ units, where the major direction of continuity is at an azimuth of 120°. The correlation between random functions Y_1 and Y_2 is 0.7. The contributions of the structures ensure validity of the target variogram model. Corresponding 1-D isotropic covariance functions used for the simulation for each structure are shown in Fig. 3.40 with the appropriate 1201 Fourier coefficients. Because correlation ranges are different for both spherical covariance functions, there is a difference between Fourier coefficients for each structure. Note that the general distribution shape of the Fourier coefficients remains the same for both structures. The periodic domain $2S$ is 400.0 units.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_1}(\boldsymbol{h}) = 0.7Sph_{r=5.0}(\boldsymbol{h}) + 0.3Sph_{\substack{r_1=30.0\\ r_2=20.0\\ \alpha=120°}}(\boldsymbol{h}) \\ C_{Y_1Y_2}(\boldsymbol{h}) = 0.5Sph_{r=5.0}(\boldsymbol{h}) + 0.2Sph_{\substack{r_1=30.0\\ r_2=20.0\\ \alpha=120°}}(\boldsymbol{h}) \\ C_{Y_2}(\boldsymbol{h}) = 0.7Sph_{r=5.0}(\boldsymbol{h}) + 0.3Sph_{\substack{r_1=30.0\\ r_2=20.0\\ \alpha=120°}}(\boldsymbol{h}) \end{cases} \tag{3.88}$$

A total of 100 realizations were generated. The first two realizations of each random function Y_1 and Y_2 are shown in Fig. 3.41. The realizations of the two random functions are correlated to each other and follow the spatial distribution specified by the input variogram model. Next Fig. 3.42 presents histograms of individual realizations and all realizations, and normal probability plots of

all realizations versus normal scores. Histograms of individual realizations look normal to some extent. Histograms of all realizations follow normal distribution quite closely, which is also supported by the normality check (points are lying on a straight line with some small deviations at the tails). Simulated means and variances for each random function are 0.03, 0.01 and 1.01, 1.01, respectively, which are very close to the theoretical values of 0.0 and 1.0. The relationship between simulations of two random functions are checked through the cross plots shown in Fig. 3.43. The clouds of points appear bivariate Gaussian. The correlation coefficient is computed precisely to be 0.7 for the cloud of points from all 100 realizations. Finally, experimental direct- and cross-variograms for major and minor directions of continuity computed for each realization and their average along with the varogram model are shown in Fig. 3.44. The variogram model is closely reproduced on average with smaller spread of individual experimental variograms in comparison with the univariate example. It might be partially due to more constrained target covariance function (two structures versus one).

3.3 White Noise Simulation

This section is devoted to the simulation of the remaining nugget effect component $X_0(\boldsymbol{u})$ of the simulation $Y(\boldsymbol{u})$.

3.3.1 Random Number Generation

The random independent factor $X_0(\boldsymbol{u})$ is a white noise with the nugget covariance function that follows a Gaussian distribution with zero mean and unit variance. Its expression is shown in Eq. (3.89), where $U(\boldsymbol{u})$ is a completely random noise that follows uniform distribution with 0.5 mean and 1/12 variance and its values lie between 0 and 1 as shown in Eq. (3.90). The function f transforms

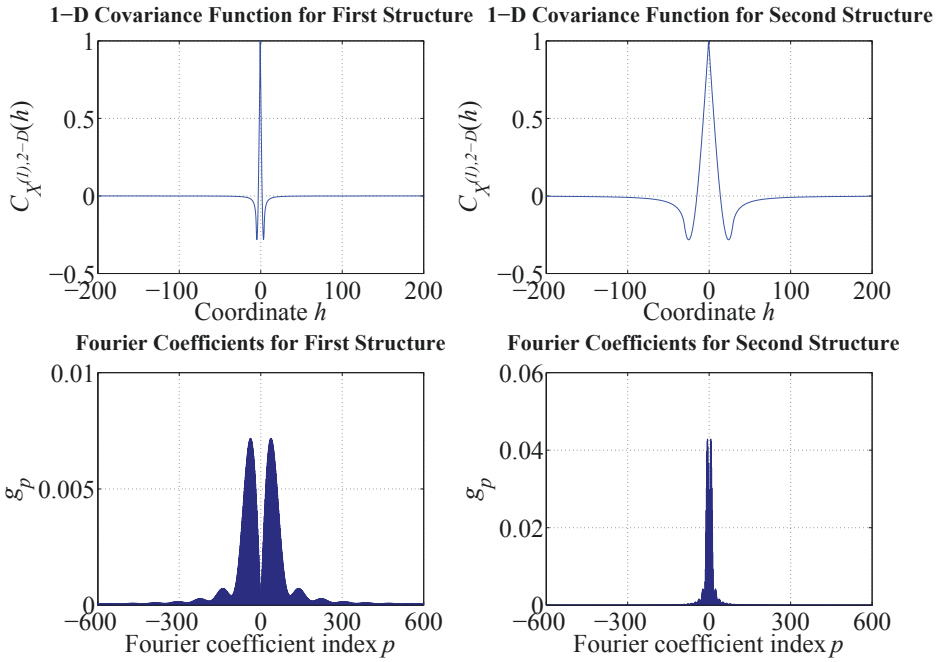


Figure 3.40: Multivariate 2-D grid-free simulation example – (a) 1-D spherical covariance function used as a first structure of the target covariance function for 2-D simulation, (b) 1-D spherical covariance function used as a second structure of the target covariance function for 2-D simulation, and (c) & (d) their corresponding Fourier coefficients.

random numbers U to a random factor X_0 with specified distribution parameters. This section describes numerical generation of the random numbers U with pseudo-random number generator. The algorithm of how to tie the inherently random component $U(\boldsymbol{u})$ to the coordinates the simulation location $\boldsymbol{u}$ and the form of the function f are explained below in Section 3.3.2.

$$X_0(\boldsymbol{u}) = f\left(U(\boldsymbol{u})\right) \tag{3.89}$$

$$U(\boldsymbol{u}) \sim U(0,1) \tag{3.90}$$

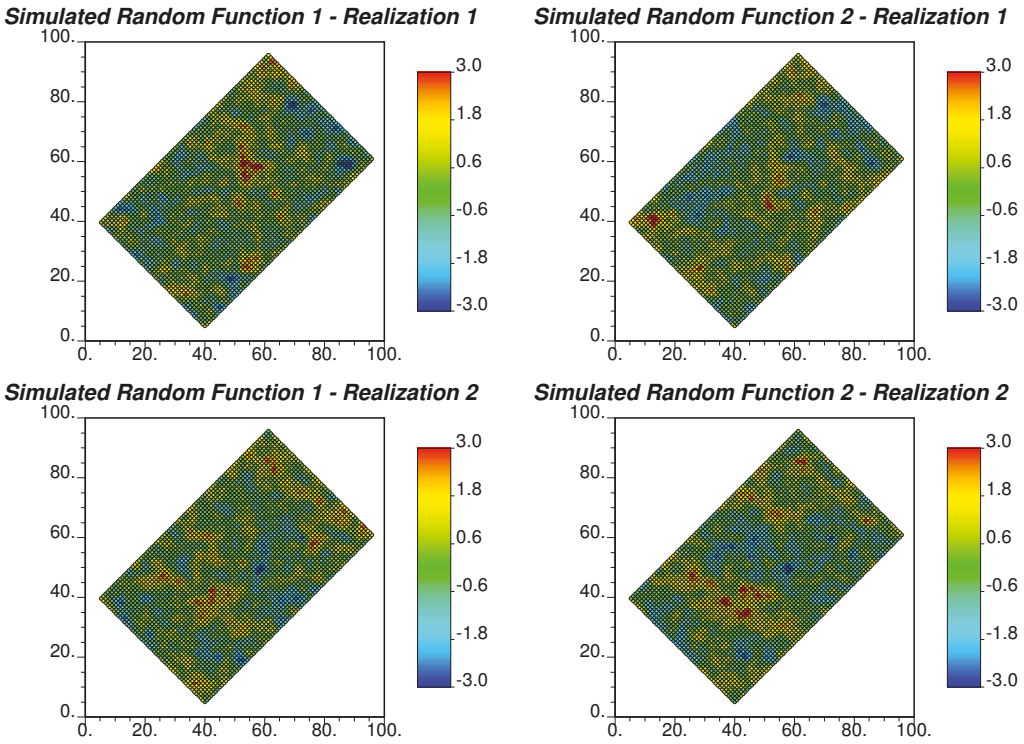


Figure 3.41: Multivariate 2-D grid-free simulation example – realizations 1 and 2 maps of random functions Y_1 and Y_2 of the bivariate system.

The uniform random numbers can be generated with various numerical techniques. The most common methods are additive congruential [81]. The `acorn` algorithm is selected as a robust random number generator, which is a variation of the congruential methods [80]. The general `acorn` algorithm is presented in Fig. 3.45. The procedure itself is simple, but it produces high quality pseudo-random numbers. The algorithm starts with the specification of total amount N_{runs} of random numbers U to be generated. There is a specification block for the initialization of parameters such as the number of iterations M, which is set up at 12 by default, array of the changing parameters $value(1{:}M{+}1)$, maximum threshold of the changing values $value_{max}$, which is set at 2×10^6

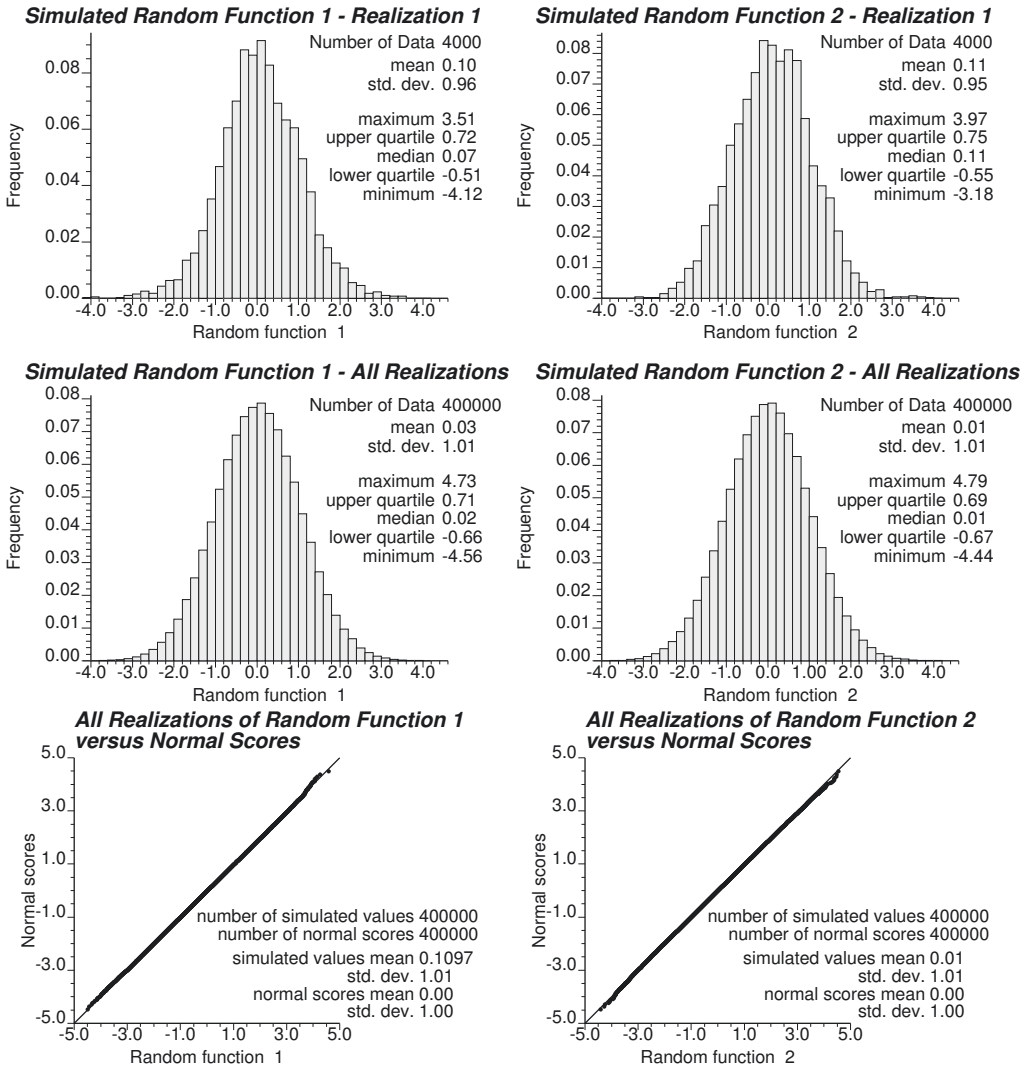


Figure 3.42: Multivariate 2-D grid-free simulation example – histograms of first realizations and all 100 realizations along with the normality check of all realizations of random functions Y_1 and Y_2 of the bivariate system.

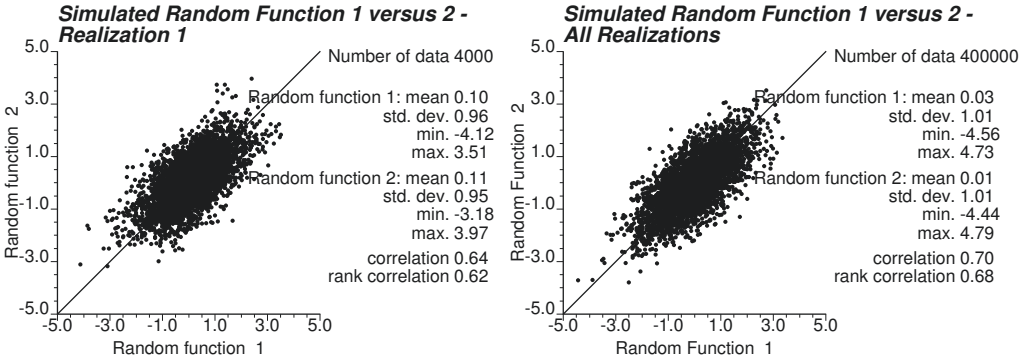


Figure 3.43: Multivariate 2-D grid-free simulation example – scatter plots between first realizations and all realizations of random functions Y_1 and Y_2 of the bivariate system.

by default, and initial value of the changing parameters $value(1)$, which is defined by a *seed* number. Once parameters are set, the iteration begins between $j = 1$ and $j = M$. While the iteration is performed, it is ensured that the last changing value $value(M+1)$ in the array does not exceed a specified maximum value $value_{max}$. Once this condition is not honored, the last $value(M+1)$ is reduced by $value_{max}$. After the final iteration is performed, the random number U is computed by dividing the last changing value in the array by the maximum value. The procedure is repeated for another random number, for which $i = i + 1$, until all random numbers are generated. The parameters are not restored, but rather kept at current values except iteration index j, which is set back to 1. The resulting random numbers are uniformly distributed between 0 and 1.

3.3.2 White Noise as Location Function

For the white noise $X_0(\boldsymbol{u})$, which is also called zero-order random factor, to be expressed as a function of the location $\boldsymbol{u}$ in the simulation domain, the following three steps can be carried out for each simulation location.

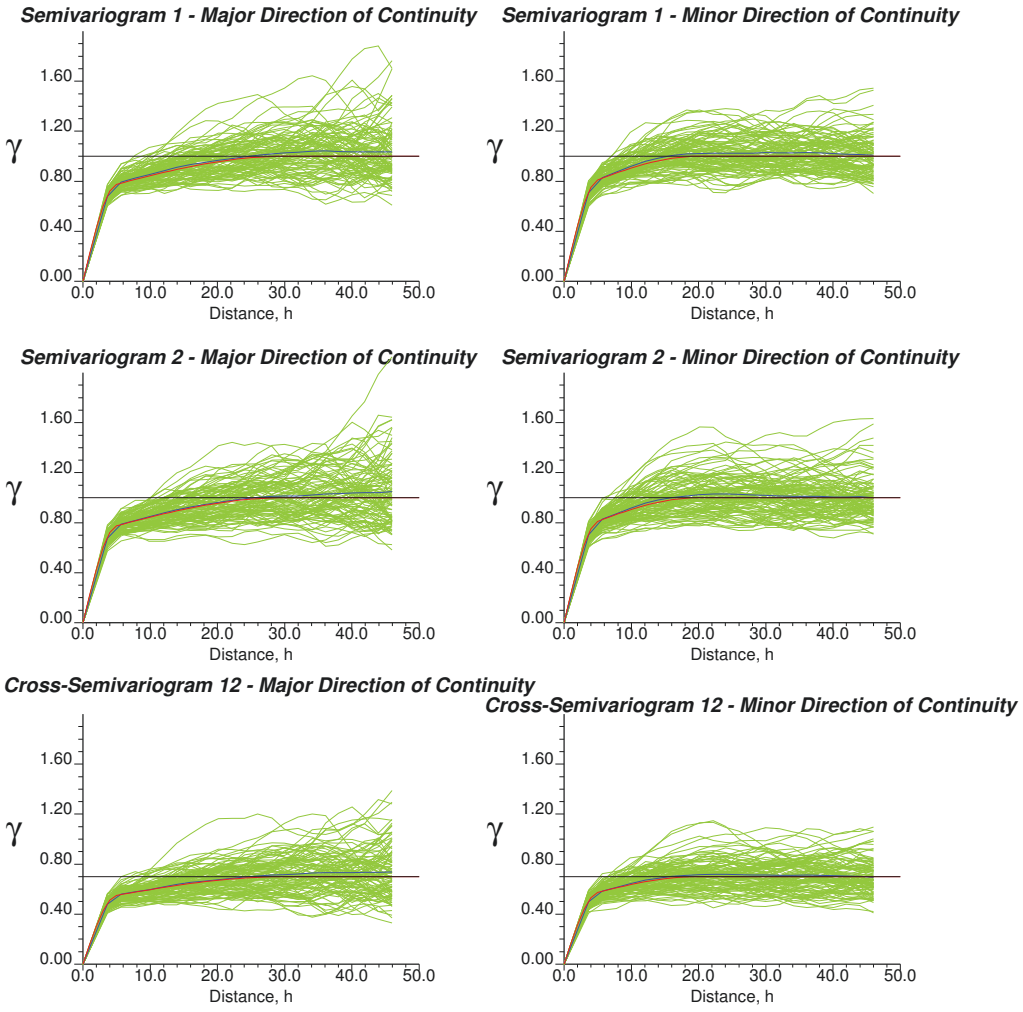


Figure 3.44: Multivariate 2-D grid-free simulation example – experimental direct- and cross-variograms (in green) computed for all realizations and their average experimental variograms (in blue) along with the variogram model (in red) of random functions Y_1 and Y_2 of the bivariate system for major and minor directions of continuity.

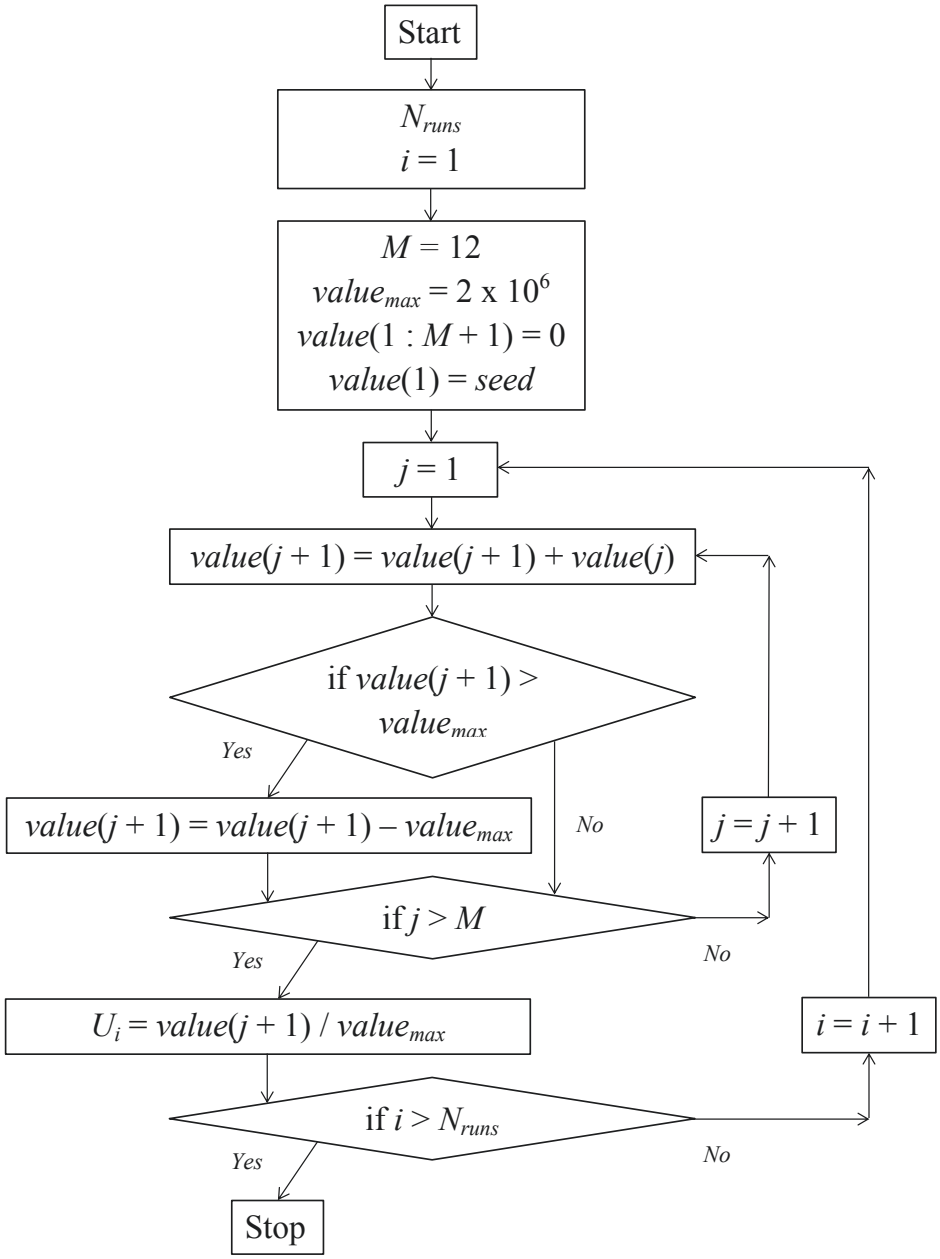


Figure 3.45: Algorithm of random number generator `acorn`.

First, the coordinates of the simulation location $\boldsymbol{u}$ should be indexed. Index $j(\boldsymbol{u})$ is presented in the integer form, and is function of the coordinates $x(\boldsymbol{u})$, $y(\boldsymbol{u})$, and $z(\boldsymbol{u})$ of the simulation location $\boldsymbol{u}$. To make conversion unique, the minimum resolution of the simulation should be assumed, which are Δ_X, Δ_Y, and Δ_Z resolution sizes in X, Y, and Z directions, respectively. The boundary of the simulation domain should be also specified. For this purpose in addition to resolution sizes, the minimum x_{min}, y_{min}, and z_{min} coordinates of the simulation domain and total number of the discretization nodes N_X, N_Y, and N_Z in X, Y, and Z directions are required. The expression to convert location coordinates to the index is shown in Eq. (3.91), where $int\{\}$ is the operator to convert real number to an integer. The location index will be used as a random number generation seed. As a result of such indexing, a very dense mesh is imposed over the simulation domain, and the simulation location of the white noise is shifted to the center of the closest grid block of this fine mesh.

$$\begin{aligned} j(\boldsymbol{u}) = & \; int\left\{\frac{x(\boldsymbol{u})-x_{min}}{\Delta_x}\right\} \\ & + int\left\{\frac{y(\boldsymbol{u})-y_{min}}{\Delta_y}\right\} N_X \\ & + int\left\{\frac{z(\boldsymbol{u})-z_{min}}{\Delta_z}\right\} N_X N_Y \end{aligned} \tag{3.91}$$

Second, because most of the random number generators utilize some simple arithmetical operation such as addition or multiplication in congruential methods including `acorn`, the generated random numbers might show some index-based pattern. To avoid this non-randomness, the random number generator should be rerun as many times as $N_{runs,0}$, which happens to be at least 2000 for `acorn` random number generator, to get final random number for the simulation location $\boldsymbol{u}$. Also, to account for simulation of multiple realizations and variables, additional

reruns are required. The Eq. (3.92) shows how the random numbers are generated, what the random number seed is, and how many times the random number generator should be rerun to generate random numbers of acceptable quality for multiple realization of the several random factors $X_{0,k,i}(\boldsymbol{u}), k = 1, ..., K$ and $i = 1, ..., N_R$, at location $\boldsymbol{u}$.

$$U_{k,i}(\boldsymbol{u}) = rand\{\text{seed} = j(\boldsymbol{u}), N_{runs} = N_{runs,0} + (k-1) + (i-1)K\} \tag{3.92}$$

Third, convert random number $U_{k,i}(\boldsymbol{u})$ to the normal score $NS_{k,i}(\boldsymbol{u})$ with zero mean and unit variance using quantile transformation procedure. This normal score is the realization i value of the zero-order random factor $X_{0,k,i}(\boldsymbol{u})$ at location $\boldsymbol{u}$. The random numbers represent quantile values, which are used to read normal scores from the cumulative distribution function of the $N(0,1)$ distribution as shown in Eq. (3.93), where G is the CDF of the $N(0,1)$ normal score values, and G^{-1} is the inverse of the G.

$$X_{0,k,i}(\boldsymbol{u}) = G^{-1}\left[U_{k,i}(\boldsymbol{u})\right] \tag{3.93}$$

The need to represent white noise in a grid-free manner is "sequence independence", that is, the same white noise values are obtained everywhere in space regardless of the sequence. Alternatively, simulated white noise values could be kept for each simulation location and restored from the memory as needed. However, this alternative approach does not allow to represent white noise in the form of analytical function of the coordinates of the simulation locations.

Two small case studies are prepared to show the functionality of the proposed approach to the simulation of the random factor $X_0(\boldsymbol{u})$ in a semi-analytical grid-free fashion.

The first case study consists of a single random function, which is generated by the GFS with the nugget covariance model shown in Eq. (3.94). The random numbers involved in the simulation of the random function and their difference along with the resulting white noise simulation resolved on two grids are presented in Fig. 3.46. By looking at the difference maps, it is obvious that white noise is indeed expressed as a function of the coordinates of the simulation location. Fig. 3.47 further justifies this observation, where histograms of the random numbers, white noises, and their differences are shown. A larger sample size leads to a more uniform distribution of the random numbers and to more normal distribution of the white noise.

$$C_Y(\boldsymbol{h}) = Nugget(\boldsymbol{h}) \tag{3.94}$$

The second case study consists of two random functions generated with the covariance models according to Eqs. (3.95) and (3.96). The difference between these two random functions is presence of a nugget effect in the second simulation. The total number of the Fourier coefficients is 801. The number of the turning lines is 200. The size of the periodic domain is 200.0. The maps of the two simulations, their difference, and corresponding histograms are shown in Fig. 3.48. The general pattern of two maps are similar, where difference is due to the white noise present in second random function. The experimental variograms computed for both simulations in X direction are shown in Fig. 3.49. The experimental variograms more or less follow the variogram model. Deviations are explained by the ergodicity. The nugget effect is clearly seen on the right plot. The general shape of the experimental variograms are quite close to each other for random functions generated without and with a nugget effect. Therefore, it has been shown how to generate unconditional realizations with the nugget effect in a grid-free fashion.

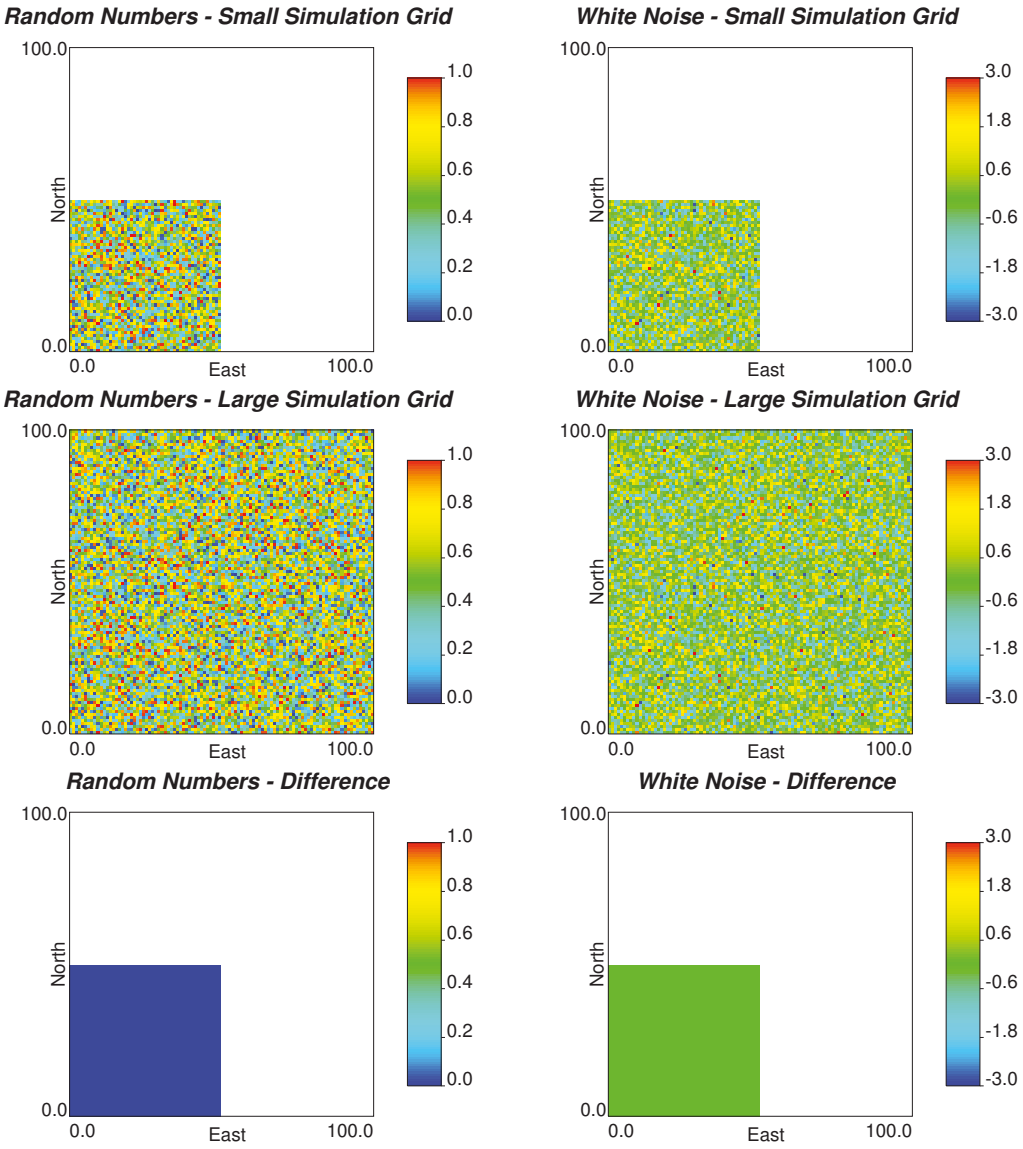


Figure 3.46: Example of white noise generation – left column is represented by maps of random numbers generated on two different grid and their difference, where smaller grid coincides with the bottom left portion of the larger grid, and right column is represented by corresponding maps of white noise realizations and their difference between two grids.

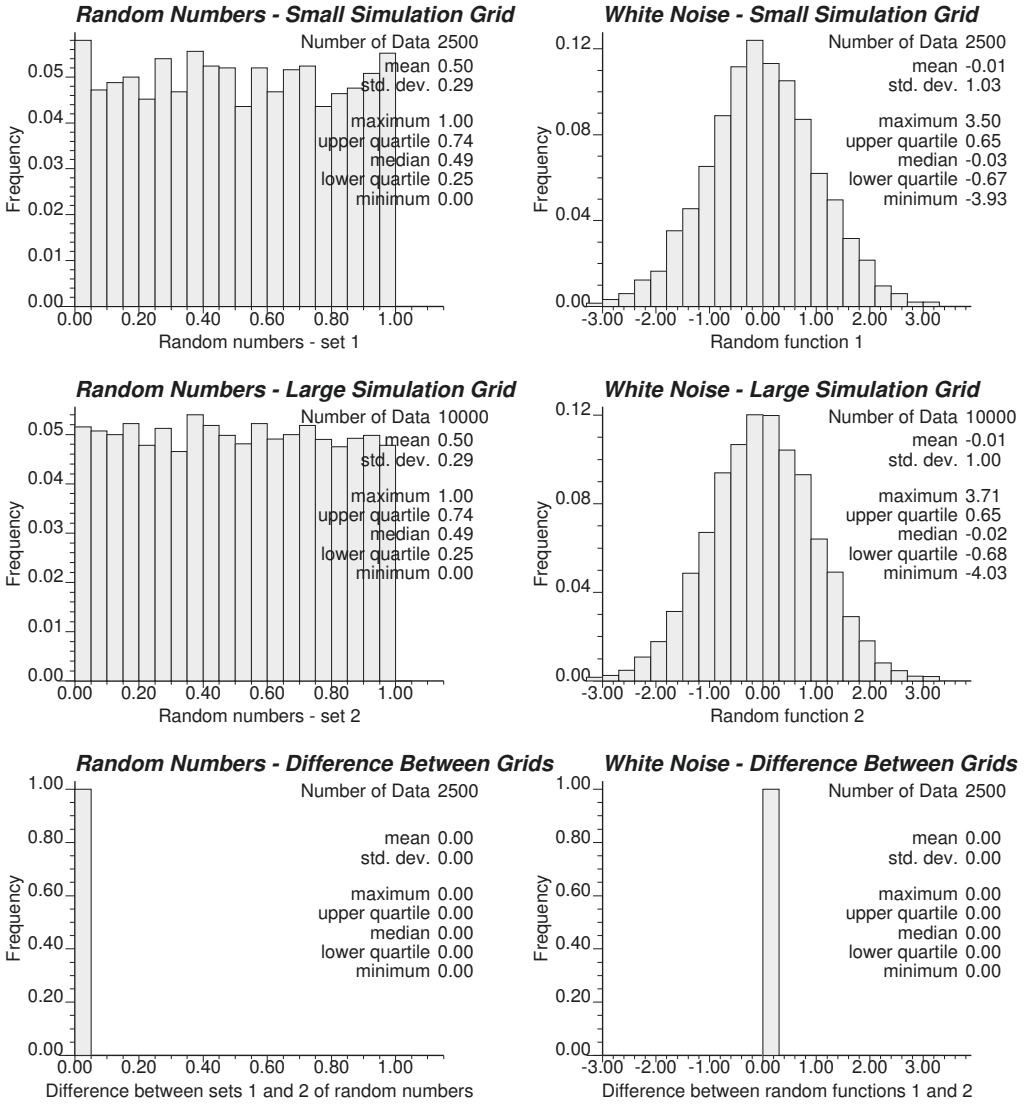


Figure 3.47: Example of white noise generation – left column is represented by histograms of random numbers generated on two different grid (see Fig. 3.46) and their difference, and right column is represented by corresponding histograms of white noise realizations and their difference between two grids.

$$C_{Y_1}(\boldsymbol{h}) = Sph_{r=40.0}(\boldsymbol{h}) \tag{3.95}$$

$$C_{Y_2}(\boldsymbol{h}) = 0.2 + 0.8Sph_{r=40.0}(\boldsymbol{h}) \tag{3.96}$$

Thus, representation of the unconditional realization Y^{us} of the random function Y in a grid-free manner is complete with nugget effect component X_0 presented as a function of the simulation location coordinates.

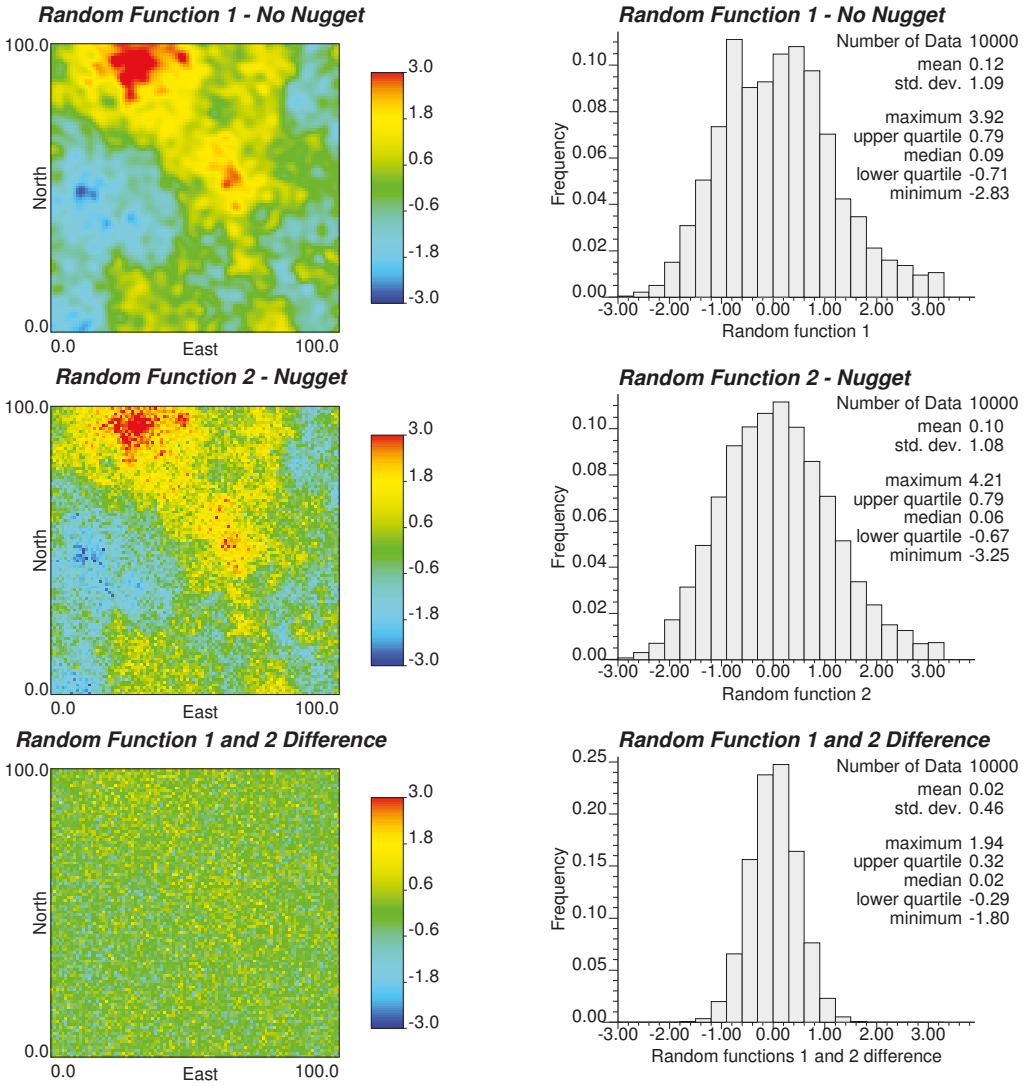


Figure 3.48: Example of grid-free simulation with nugget effect – maps and corresponding histograms of two realizations generated without and with nugget effect and their difference.

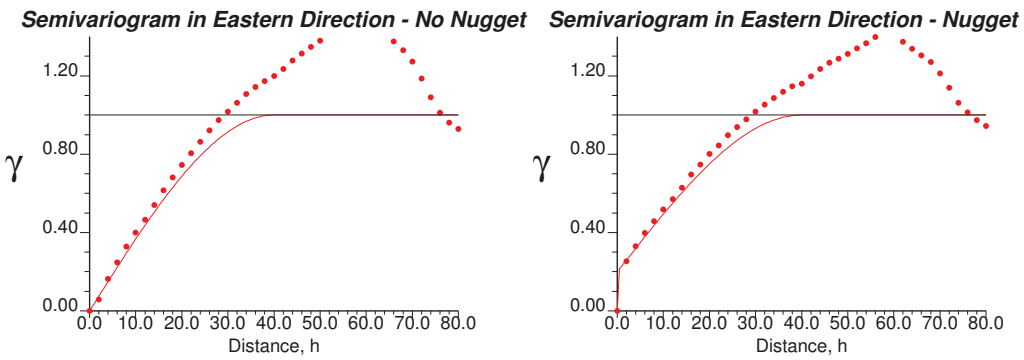


Figure 3.49: Example of grid-free simulation with nugget effect – experimental variograms (shown by dots) computed in eastern direction from realizations generated without and with nugget effect and variogram model (shown by line).

Chapter 4

Conditioning

The previous Chapter 3 presented unconditional grid-free simulation at point scale. This chapter focuses on the kriging-based conditioning in presence of various data types sampled at point and block scales [36, 37]. An illustrative example in Fig. 4.1 depicts main points of the conditioning univariate simulation through the kriging. The procedure behind this conditioning is as follows. First, an unconditional simulation Y^{us} with required spatial structure is generated. Second, an estimate is removed from this unconditional simulation by subtracting the kriging estimate $Y^{*|us}$, where the actual data values are replaced with the unconditionally simulated values at the data locations, that is, from the simulation Y^{us}. Third, the kriging estimate Y^* derived from actual data values is added back to arrive at a conditional simulation Y^s that preserves the spatial structure of the random function and reproduces the data at the data locations. The mathematical form of the conditioning at a particular location $\boldsymbol{u}$ is shown in Eq. (4.1) for a point scale, where $Y^s(\boldsymbol{u})$ is the conditional realization at location $\boldsymbol{u}$, $Y^{us}(\boldsymbol{u})$ is the unconditional realization at location $\boldsymbol{u}$, $Y^{*|us}(\boldsymbol{u})$ is the kriging estimate at location $\boldsymbol{u}$, in which data values are replaced with unconditionally simulated values at the same data locations, and $Y^*(\boldsymbol{u})$ is the kriging estimate at

location $\boldsymbol{u}$. Because kriging is a grid-free estimation technique, final conditional realization has also a grid-free form. This approach appears ad-hoc; however, it is based on sound theory and the correctness of this has been established in many references [10, 36].

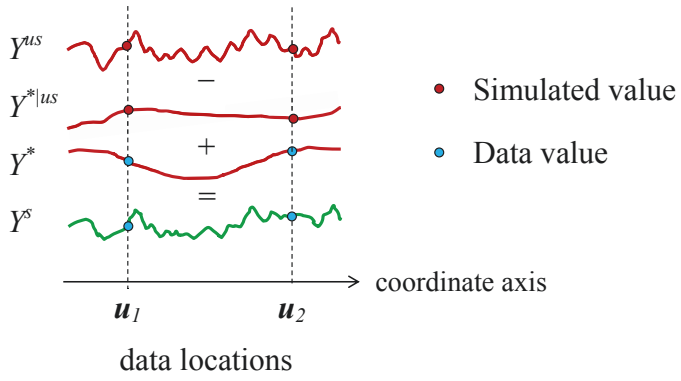


Figure 4.1: Illustrative 1-D example of conditioning a realization by kriging.

$$Y^s(\boldsymbol{u}) = Y^{us}(\boldsymbol{u}) - Y^{*|us}(\boldsymbol{u}) + Y^*(\boldsymbol{u}) \tag{4.1}$$

For the multivariate systems consisting of K random functions, the scalar values Y in Eq. (4.1) are replaced with the vector forms $\boldsymbol{Y}$ as in Eq. (4.2), where each vector has $K \times 1$ size. Matrices of conditional and unconditional realizations are shown in Eqs. (4.3) and (4.4), respectively.

$$\boldsymbol{Y}^s(\boldsymbol{u}) = \boldsymbol{Y}^{us}(\boldsymbol{u}) - \boldsymbol{Y}^{*|us}(\boldsymbol{u}) + \boldsymbol{Y}^*(\boldsymbol{u}) \tag{4.2}$$

$$\boldsymbol{Y}^s(\boldsymbol{u}) = \begin{bmatrix} Y_1^s(\boldsymbol{u}) \\ \dots \\ Y_K^s(\boldsymbol{u}) \end{bmatrix} \tag{4.3}$$

$$\boldsymbol{Y}^{us}(\boldsymbol{u}) = \begin{bmatrix} Y_1^{us}(\boldsymbol{u}) \\ ... \\ Y_K^{us}(\boldsymbol{u}) \end{bmatrix} \tag{4.4}$$

An example of the kriging-based conditioning of unconditional realization of a 2-D univariate system is presented in Fig. 4.2. First, unconditional simulation with isotropic spherical covariance structure is simulated on a 100 × 100 grid, which covers 100 × 100 m^2 area. The correlation range is 20.0 m. There are 9 data values evenly measured over the modeling area. Second, the unconditionally simulated values are sampled at these 9 data locations and used to get an estimate of the unconditional simulation through kriging. Third, the kriging estimate conditioned to original data values is calculated. Fourth, the first estimate model is removed from the unconditional simulation. Last, kriging estimate is added back for data reproduction. As a result, the final realization preserves input spatial structure and honors the data. While the kriging estimates look artificially smooth, the unconditional and conditional realizations reproduce the specified spatial correlation structure.

4.1 Conditional Grid-Free Simulation with Scattered Data

This section is devoted to the conditioning of unconditional realizations with data values sampled arbitrarily over the modeling domain.

4.1.1 Conditioning with Primal Cokriging

The primal form of the kriging equations is the conventional representation of the spatial interpolation method in geostatistics [10, 26, 35, 36, 77]. In practice,

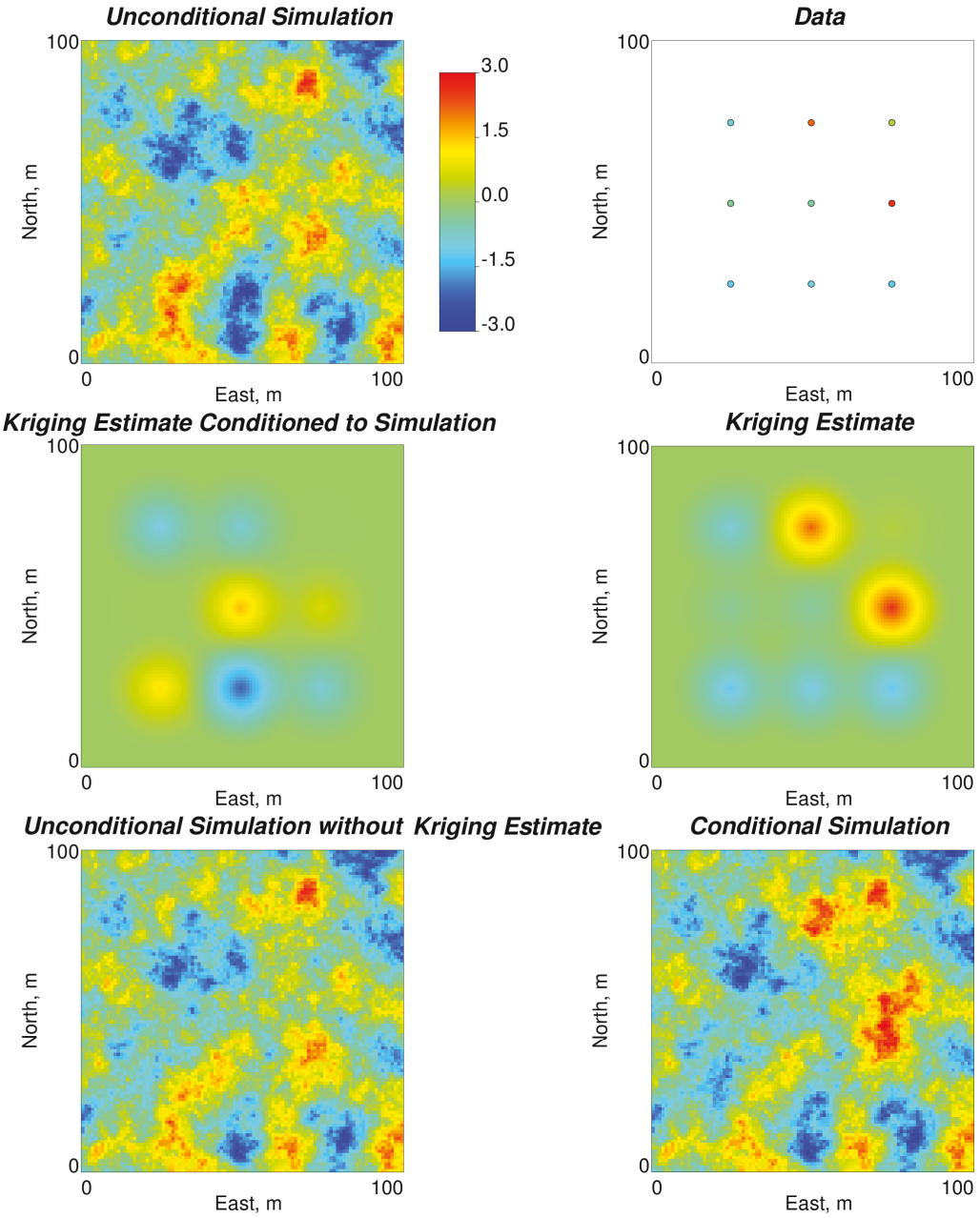


Figure 4.2: Example of conditioning a realization by kriging in 2-D.

all calculations are done in the normal units with stationarity assumption, the global mean of the random function is zero and variance is one. The simple kriging estimate can be expressed as shown in Eq. (4.5) for actual conditioning data values and in Eq. (4.6) for the unconditionally simulated conditioning data values, where N is the number of the data. More explanation of the kriging estimate was given in Section 2.2.1 of Chapter 2.

$$Y^*(\boldsymbol{u}) = m_Y(\boldsymbol{u}) + \sum_{\alpha=1}^{N} \lambda_\alpha(\boldsymbol{u}) \left(Y(\boldsymbol{u}_\alpha) - m_Y(\boldsymbol{u}_\alpha)\right) \tag{4.5}$$

$$Y^{*|us}(\boldsymbol{u}) = m_Y(\boldsymbol{u}) + \sum_{\alpha=1}^{N} \lambda_\alpha(\boldsymbol{u}) \left(Y^{us}(\boldsymbol{u}_\alpha) - m_Y(\boldsymbol{u}_\alpha)\right) \tag{4.6}$$

Because the only difference between these two equations is the conditioning data values, conditioning Eq. (4.1) can be rewritten into simpler expression shown in Eq. (4.7), which states that the conditional simulation $Y^s(\boldsymbol{u})$ is represented as a sum of the unconditional simulation $Y^{us}(\boldsymbol{u})$ and a single conditioning kriging estimate $\Delta Y^*(\boldsymbol{u})$ at location $\boldsymbol{u}$. The conditioning kriging estimate is shown in Eq. (4.8) or in Eq. (4.9) in matrix form. Here, the data values are replaced with the difference between actual data values and unconditionally simulated values at data locations $\boldsymbol{u}_\alpha, \alpha = 1, ..., N$, as shown in Eq. (4.10). The kriging weights $\boldsymbol{\lambda}(\boldsymbol{u})$ are stored in the vector form as presented in Eq. (4.11).

$$Y^s(\boldsymbol{u}) = Y^{us}(\boldsymbol{u}) + \Delta Y^*(\boldsymbol{u}) \tag{4.7}$$

$$\Delta Y^*(\boldsymbol{u}) = \sum_{\alpha=1}^{N} \lambda_\alpha(\boldsymbol{u}) \left(Y(\boldsymbol{u}_\alpha) - Y^{us}(\boldsymbol{u}_\alpha)\right) \tag{4.8}$$

$$\Delta Y^*(\boldsymbol{u}) = \boldsymbol{\lambda}^T(\boldsymbol{u})\, \boldsymbol{Y}^d \tag{4.9}$$

$$\boldsymbol{Y}^d = \begin{bmatrix} Y(\boldsymbol{u}_1) - Y^{us}(\boldsymbol{u}_1) \\ \dots \\ Y(\boldsymbol{u}_\alpha) - Y^{us}(\boldsymbol{u}_\alpha) \\ \dots \\ Y(\boldsymbol{u}_N) - Y^{us}(\boldsymbol{u}_N) \end{bmatrix} \tag{4.10}$$

$$\boldsymbol{\lambda}(\boldsymbol{u}) = [\lambda_1(\boldsymbol{u}) \dots \lambda_\alpha(\boldsymbol{u}) \dots \lambda_N(\boldsymbol{u})]^T \tag{4.11}$$

The kriging weights $\lambda_\alpha(\boldsymbol{u})$ are derived by minimizing the variance of the estimation error $\varepsilon(\boldsymbol{u}) = Y^*(\boldsymbol{u}) - Y(\boldsymbol{u})$ at the estimation location $\boldsymbol{u}$. The expression for the kriging weights is presented as a system of kriging equations Eq. (4.12) or as Eq. (4.13) in a matrix form. More details on the derivation of the kriging weights can be found in Section 2.2.1 of Chapter 2. Expressions of the data covariance matrix $\boldsymbol{C}_Y$ and simulation – data covariance vector $\boldsymbol{c}_Y$ are shown in Eqs. (4.14) and (4.15), respectively.

$$\sum_{\alpha=1}^{N} C_Y(\boldsymbol{u}_\alpha - \boldsymbol{u}_{\alpha'})\lambda_\alpha(\boldsymbol{u}) = C_Y(\boldsymbol{u} - \boldsymbol{u}_{\alpha'}), \;\; \alpha' = 1, ..., N \tag{4.12}$$

$$\boldsymbol{C}_Y \boldsymbol{\lambda}(\boldsymbol{u}) = \boldsymbol{c}_Y(\boldsymbol{u}) \tag{4.13}$$

$$\boldsymbol{C}_Y = \begin{bmatrix} C_Y(\boldsymbol{u}_1 - \boldsymbol{u}_1) & \dots & C_Y(\boldsymbol{u}_1 - \boldsymbol{u}_{\alpha'}) & \dots & C_Y(\boldsymbol{u}_1 - \boldsymbol{u}_N) \\ \dots & & \dots & & \dots \\ C_Y(\boldsymbol{u}_\alpha - \boldsymbol{u}_1) & \dots & C_Y(\boldsymbol{u}_\alpha - \boldsymbol{u}_{\alpha'}) & \dots & C_Y(\boldsymbol{u}_\alpha - \boldsymbol{u}_N) \\ \dots & & \dots & & \dots \\ C_Y(\boldsymbol{u}_N - \boldsymbol{u}_1) & \dots & C_Y(\boldsymbol{u}_N - \boldsymbol{u}_{\alpha'}) & \dots & C_Y(\boldsymbol{u}_N - \boldsymbol{u}_N) \end{bmatrix} \tag{4.14}$$

$$\boldsymbol{c}_Y = \begin{bmatrix} C_Y(\boldsymbol{u}-\boldsymbol{u}_1) \\ \cdots \\ C_Y(\boldsymbol{u}-\boldsymbol{u}_{\alpha'}) \\ \cdots \\ C_Y(\boldsymbol{u}-\boldsymbol{u}_N) \end{bmatrix} \tag{4.15}$$

An example is provided to show the generation of a conditional univariate grid-free simulation in 2-D. Consider a data set consisting of 62 points distributed as shown in Fig. 4.3 in plan view. These data represent measurements from 20 vertical and 2 horizontal wells over a 5000.0 × 5000.0 m^2 modeling area. The data are synthetically generated by unconditional grid-free simulation algorithm with anisotropic covariance function specified in Eq. (4.16). Therefore, the data follow a normal distribution with zero mean and unit variance. The histogram of the data is also shown in Fig. 4.3. The mean and variance of the data slightly deviate from the theoretical values, because the sample size is small.

$$C_Y(\boldsymbol{h}) = Sph_{\substack{r_1=1000.0\ m \\ r_2=500.0\ m \\ \alpha=60^\circ}}(\boldsymbol{h}) \tag{4.16}$$

A total of 50 conditional realizations are generated within three simulation domains: over the entire domain at a coarse resolution 50.0 × 50.0 m^2, around horizontal wells at a finer resolution 5.0 × 5.0 m^2, and around some portion of the upper horizontal well at an even finer resolution 1.0 × 1.0 m^2. The covariance function used in the simulation is the same as one used for the data generation. The number of Fourier coefficients is 301, and number of the 1-D line processes is 50. These parameters are chosen to generate realizations with required accuracy artifact-free. The details on choice of the simulation parameters are provided later in Chapter 5. The locations of the simulation

nodes are placed on regular grids as shown in Fig. 4.4: large grid 1, medium grid 2, and small grid 3, where a smaller grid is a subset of the larger grids. The purpose of simulation at various grid resolutions is a reserve estimation of the entire domain, construction of geological model for subsequent flow simulation, and refinement of some geological properties within particular area of interest. The GFS does not depend on the simulation grid, and therefore three simulations resolved on three grids will be identical in the overlapping areas. This example represents the zooming in feature of GFS.

Maps of first two realizations on grid 1 with the local mean and estimation error variance of all 50 realizations are shown in Fig. 4.5. The data are reproduced, which can be seen on the mean and estimation error variance maps. The anisotropy is obvious from the realizations and post-processed maps. The uncertainty map coincides with the theoretical prediction, which states that the uncertainty is zero at the data locations and increases up to the global stationary variance at locations lying further than correlation range from the data. Fig. 4.6 presents the same first two realizations resolved on grid 2 around the horizontal wells. This second set of realizations can be called zoomed in realizations. Fig. 4.7 presents the first two realizations on grid 3 at even finer resolution. Because the realization maps represent the same realizations at various resolutions, the overlapping region of the realizations resolved on different grids are identical. The histogram of all 50 realizations generated on grid 1 and their normal probability plot are shown in Fig. 4.8. The statistics of the realizations are in compliance with the data statistics and required theoretical statistics of the normal distribution. Fig. 4.9 contains experimental variogram map computed from 50 realizations and experimental variograms computed in eastern direction shown as green lines for each realization along with their average shown as a blue line and variogram model shown as a red line. The variogram reproduction is good for individual realizations and quite good for their average. The spread

of the individual experimental variograms is less for the conditional realizations than for the unconditional realizations (see Fig. 3.21 or Fig. 3.39 in Chapter 3) [54].

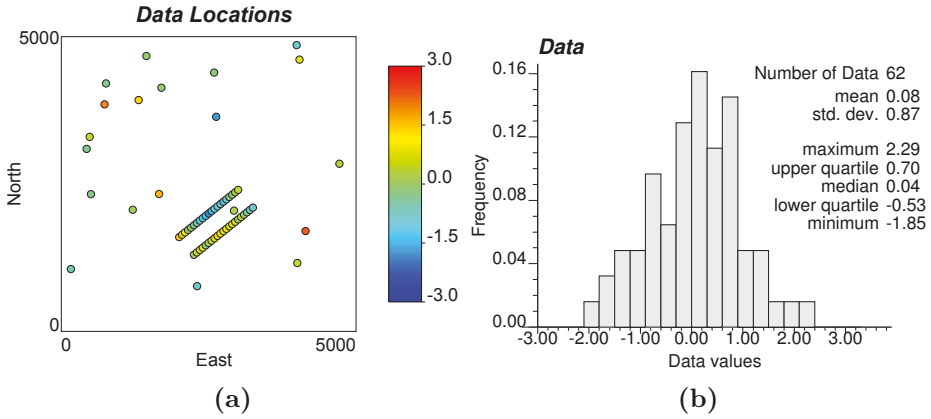


Figure 4.3: Example of univariate conditional grid-free simulation – (a) data locations and (b) data histogram.

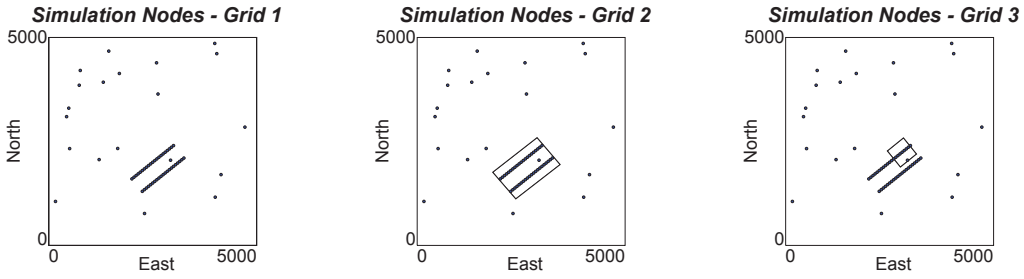


Figure 4.4: Example of univariate conditional grid-free simulation – three gridded configurations of the simulation nodes.

The equations for multivariate conditioning remain similar to univariate case with only difference that more random functions and associated data are taken into consideration. Eqs. (4.17) – (4.24) show the kriging estimates used for the conditioning of the unconditional simulation. Kriging estimate of random variable $Y_k^*(\boldsymbol{u})$ of K-variate system with actual conditioning data is depicted in Eq. (4.17). Note that data of all types are used for the conditioning. Eq. (4.17)

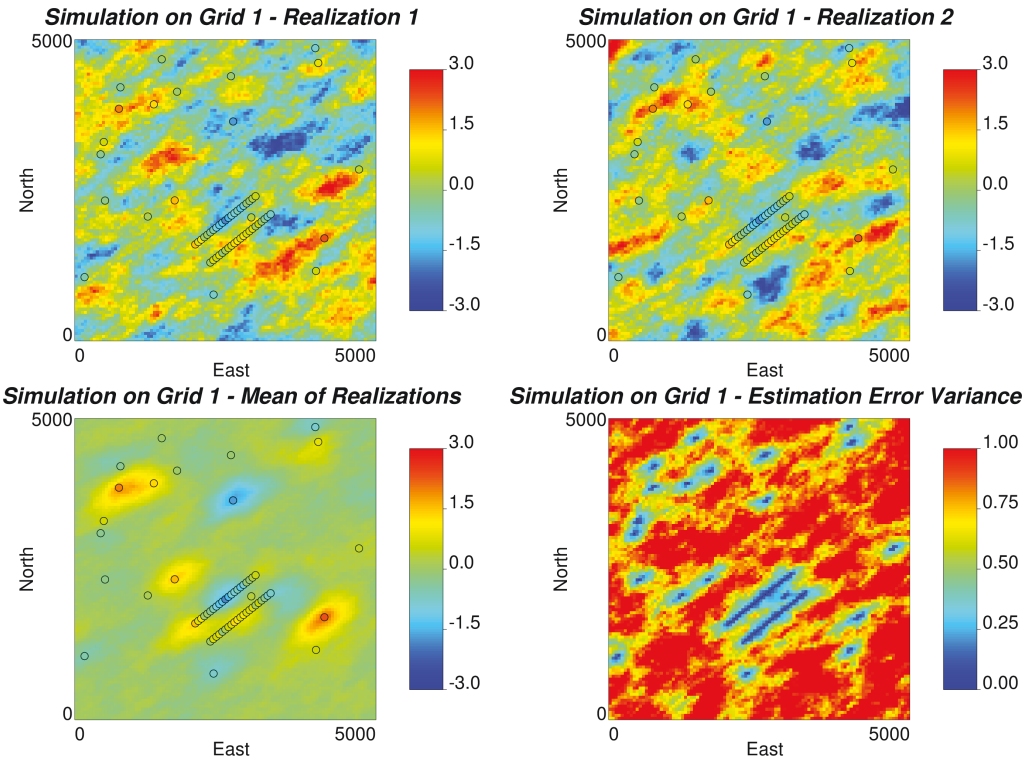


Figure 4.5: Example of univariate conditional grid-free simulation – maps of first two realizations, mean of 50 realizations, and estimation error variance simulated on large grid 1.

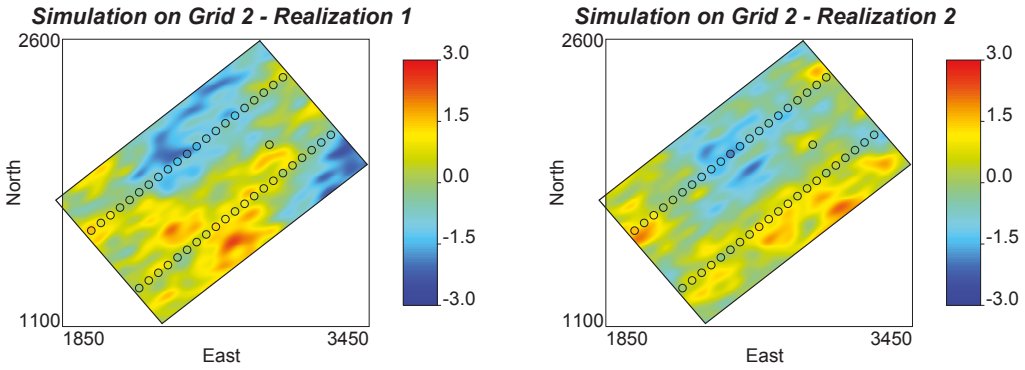


Figure 4.6: Example of univariate conditional grid-free simulation – maps of first two realizations on medium grid 2.

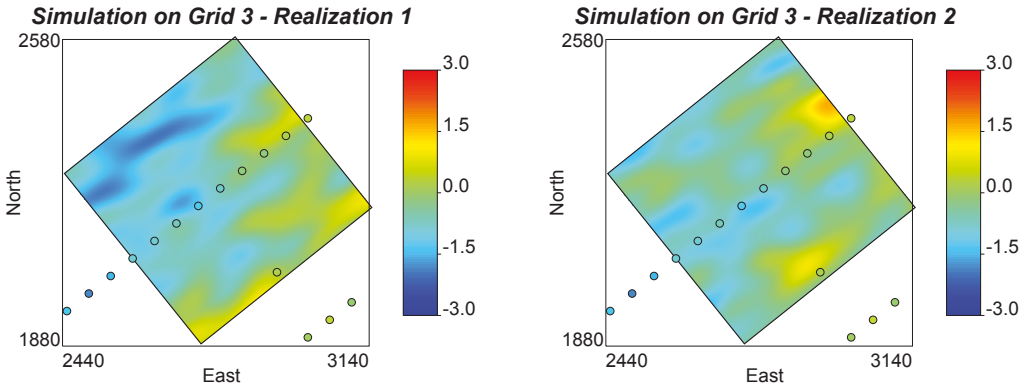


Figure 4.7: Example of univariate conditional grid-free simulation – maps of first two realizations on small grid 2.

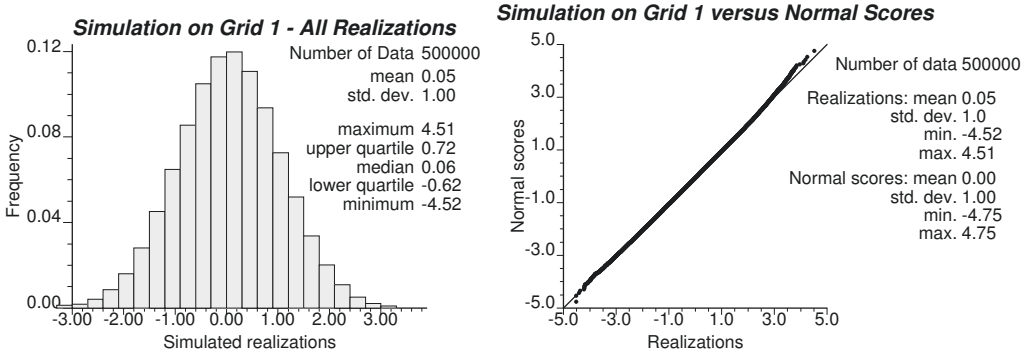


Figure 4.8: Example of univariate conditional grid-free simulation – histogram of all realizations resolved on grid 1 and probability plot to check normality of these realizations.

shows how the kriging estimate is computed with unconditionally simulated values at data locations. When these two kriging estimate equations are combined together according to conditioning in Eq. (4.2), the resulting kriging estimate can be expressed as in Eq. (4.19) or in Eq. (4.20) in matrix form. The matrices are explained in Eqs. (4.21) – (4.23). Therefore, the resulting expression for conditioning unconditional multivariate simulation with kriging is presented in Eq. (4.24).

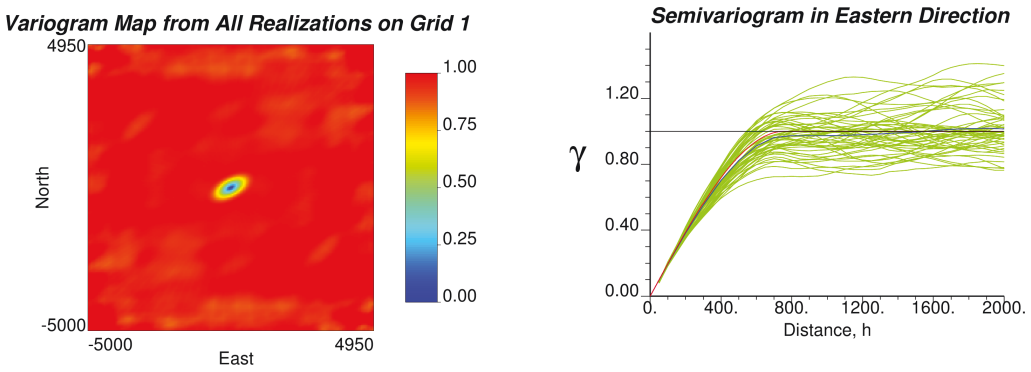


Figure 4.9: Example of univariate conditional grid-free simulation – experimental variogram map computed from all 50 realizations resolved on grid 1, and individual experimental variograms for each realization (green) along with their average variogram (blue) and corresponding variogram model (red) in eastern direction.

$$Y_k^*(\boldsymbol{u}) = m_{Y_k}(\boldsymbol{u}) + \sum_{l=1}^{K} \sum_{\alpha_l=1}^{N_l} \lambda_{k,\alpha_l}(\boldsymbol{u}) \left(Y_l(\boldsymbol{u}_{\alpha_l}) - m_{Y_l}(\boldsymbol{u}_{\alpha_l}) \right), \quad k = 1, ..., K \tag{4.17}$$

$$Y_k^{*|us}(\boldsymbol{u}) = m_{Y_k}(\boldsymbol{u}) + \sum_{l=1}^{K} \sum_{\alpha_l=1}^{N_l} \lambda_{k,\alpha_l}(\boldsymbol{u}) \left(Y_l^{us}(\boldsymbol{u}_{\alpha_l}) - m_{Y_l}(\boldsymbol{u}_{\alpha_l}) \right), \quad k = 1, ..., K \tag{4.18}$$

$$\Delta Y_k^*(\boldsymbol{u}) = \sum_{l=1}^{K} \sum_{\alpha_l=1}^{N_l} \lambda_{k,\alpha_l}(\boldsymbol{u}) \left(Y_l(\boldsymbol{u}_{\alpha_l}) - Y_l^{us}(\boldsymbol{u}_{\alpha_l}) \right), \quad k = 1, ..., K \tag{4.19}$$

$$\Delta \boldsymbol{Y}^*(\boldsymbol{u}) = \boldsymbol{\lambda}^T(\boldsymbol{u}) \, \boldsymbol{Y}^d \tag{4.20}$$

$$\Delta \boldsymbol{Y}^*(\boldsymbol{u}) = \begin{bmatrix} \Delta Y_1^*(\boldsymbol{u}) \\ ... \\ \Delta Y_K^*(\boldsymbol{u}) \end{bmatrix} \tag{4.21}$$

$$\boldsymbol{Y}^d = \begin{bmatrix} Y_1(\boldsymbol{u}_1) - Y_1^{us}(\boldsymbol{u}_1) \\ ... \\ Y_1(\boldsymbol{u}_{N_1}) - Y_1^{us}(\boldsymbol{u}_{N_1}) \\ ... \\ Y_K(\boldsymbol{u}_1) - Y_K^{us}(\boldsymbol{u}_1) \\ ... \\ Y_K(\boldsymbol{u}_{N_K}) - Y_K^{us}(\boldsymbol{u}_{N_K}) \end{bmatrix} \tag{4.22}$$

$$\boldsymbol{\lambda}(\boldsymbol{u}) = \begin{bmatrix} \lambda_{1,1}(\boldsymbol{u}) & ... & \lambda_{1,N_1}(\boldsymbol{u}) & ... & \lambda_{1,1}(\boldsymbol{u}) & ... & \lambda_{1,N_K}(\boldsymbol{u}) \\ ... & & ... & & ... & & ... \\ \lambda_{K,1}(\boldsymbol{u}) & ... & \lambda_{K,N_1}(\boldsymbol{u}) & ... & \lambda_{K,1}(\boldsymbol{u}) & ... & \lambda_{K,N_K}(\boldsymbol{u}) \end{bmatrix}^T \tag{4.23}$$

$$\boldsymbol{Y}^s(\boldsymbol{u}) = \boldsymbol{Y}^{us}(\boldsymbol{u}) + \Delta \boldsymbol{Y}^*(\boldsymbol{u}) \tag{4.24}$$

The kriging weights for the multivariate conditioning are derived in a similar way to the univariate case by minimizing the estimation variance. The resulting system of kriging equations are shown in Eq. (4.25) in the open form and in Eq. (4.26) in the matrix form. The matrices are explained in Eqs. (4.27) and (4.28). The size of data-data covariance matrix $\boldsymbol{C}_Y$ is $\sum_{k=1}^{K} N_k \times \sum_{k=1}^{K} N_k$. The size of simulation-data covariance matrix $\boldsymbol{c}_Y(\boldsymbol{u})$ is $\sum_{k=1}^{K} N_k \times K$.

$$\left(\sum_{l=1}^{K} \sum_{\alpha_l=1}^{N_l} C_{Y_l Y_m}(\boldsymbol{u}_{\alpha_l} - \boldsymbol{u}_{\alpha'_m}) \lambda_{k,\alpha_l}(\boldsymbol{u}) = C_{Y_k Y_m}(\boldsymbol{u} - \boldsymbol{u}_{\alpha'_m}), \right. \\ \left. \alpha'_m = 1, ..., N_m, \;\; m = 1, ..., K \right), \;\; k = 1, ..., K \tag{4.25}$$

$$\boldsymbol{C}_Y \boldsymbol{\lambda}(\boldsymbol{u}) = \boldsymbol{c}_Y(\boldsymbol{u}) \tag{4.26}$$

$$\boldsymbol{C}_Y = \begin{bmatrix} C_{Y_1 Y_1}(\boldsymbol{u}_1 - \boldsymbol{u}_1) & ... & C_{Y_1 Y_K}(\boldsymbol{u}_{N_1} - \boldsymbol{u}_{N_K}) \\ ... & & ... \\ C_{Y_K Y_1}(\boldsymbol{u}_{N_K} - \boldsymbol{u}_{N_1}) & ... & C_{Y_K Y_K}(\boldsymbol{u}_{N_K} - \boldsymbol{u}_{N_K}) \end{bmatrix} \tag{4.27}$$

$$\boldsymbol{c}_Y(\boldsymbol{u}) = \begin{bmatrix} C_{Y_1 Y_1}(\boldsymbol{u} - \boldsymbol{u}_1) & ... & C_{Y_1 Y_K}(\boldsymbol{u} - \boldsymbol{u}_1) \\ ... & & ... \\ C_{Y_K Y_1}(\boldsymbol{u} - \boldsymbol{u}_{N_K}) & ... & C_{Y_K Y_K}(\boldsymbol{u} - \boldsymbol{u}_{N_K}) \end{bmatrix} \tag{4.28}$$

The 3-D example below is provided to show multivariate conditional simulation with GFS. The bivariate system has anisotropic spatial structure shown in Eq. (4.29). Two random functions Y_1 and Y_2 are correlated with correlation coefficient of 0.60. The data for each random function are measured along randomly placed 20 vertical wells at 5.0 m intervals and along 2 horizontal wells at about 10.0 m intervals. The data locations are shown in Fig. 4.10. Next, Fig. 4.11 contains histograms of the data along with a scatter plot. The data look normal and correlated. The modeling domain is defined by the following ranges: $x \in$ [1500.0 m – 2500.0 m], $y \in$ [2500.0 m – 3500.0 m], and in $z \in$ [90.0 m – 170.0 m]. The objective is to show the regridding feature with the conditional GFS. It is achieved through simulation of 50 realizations of the random functions Y_1

and Y_2 at coarse grid 25.0 × 25.0 × 2.5 m^3, which covers the entire modeling domain, and simulation of a single realization of each random function at a fine grid 5.0 × 5.0 × 1.0 m^3 around a portion of the horizontal wells. Nodes of the second grid are not aligned with the nodes of the first coarsely resolved grid. The resulting realizations at the two resolutions are part of the same realization, and, therefore, are identical in the overlapping regions.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_1}(\boldsymbol{h}) = Sph_{\substack{r_1=250.0 \\ r_2=125.0 \\ r_3=10.0 \\ \alpha=0°}}(\boldsymbol{h}) \\ C_{Y_1Y_2}(\boldsymbol{h}) = 0.6Sph_{\substack{r_1=250.0 \\ r_2=125.0 \\ r_3=10.0 \\ \alpha=0°}}(\boldsymbol{h}) \\ C_{Y_2}(\boldsymbol{h}) = Sph_{\substack{r_1=250.0 \\ r_2=125.0 \\ r_3=10.0 \\ \alpha=0°}}(\boldsymbol{h}) \end{cases} \tag{4.29}$$

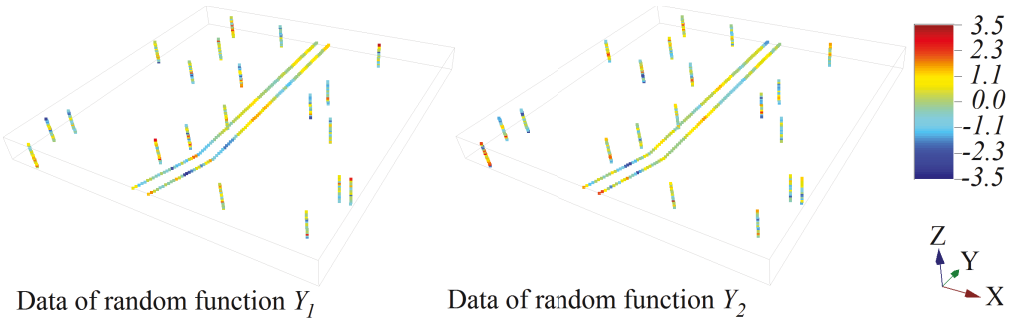


Figure 4.10: Example of multivariate conditional grid-free simulation – data location maps of random functions Y_1 and Y_2 of the bivariate system.

The simulation is performed at point scale. The realizations of both random functions resolved at coarse grid 1 are shown in Fig. 4.12 with different plotting techniques. While the realization of random function Y_1 is shown in a gridded format, the realization of random function Y_2 is visualized only at the simulation locations. Note the data reproduction on both models. Histograms of all 50 realizations and a scatter plot between realizations of the two random

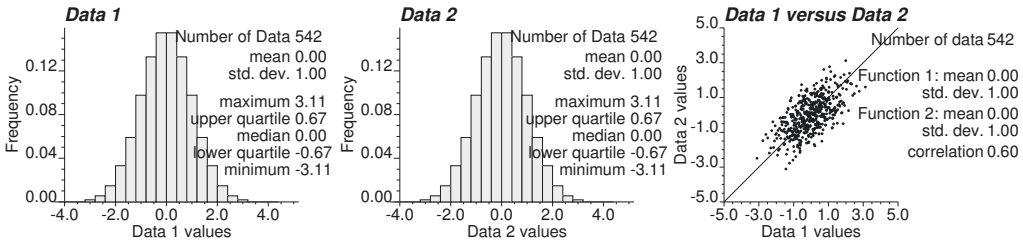


Figure 4.11: Example of multivariate conditional grid-free simulation – histograms and scatter plot of data of random functions Y_1 and Y_2 of the bivariate system.

functions are shown in Fig. 4.13. The realizations appear univariate and bivariate normal. The correlation coefficient is close to the target one. Variogram reproduction is depicted on Fig. 4.14. Experimental direct- and cross-variograms are computed for all 50 realizations in three principal directions of the continuity. These individual variograms are shown in green with their averages in blue. The variogram model is presented by the red line. The variogram model is reproduced exactly for the shorter ranges. Slight deviations are present for longer ranges. The mean and estimation error variance models of 50 realizations of random function Y_1 are presented in Fig. 4.15. The mean is smooth as expected. The data are reproduced. The estimation error variance is zero at data locations and increases as the separation from the data locations becomes larger. Finally, Fig. 4.16 depicts the first realizations of both spatially continuous functions at coarse and fine grids, which are not aligned with each other. Both realizations coincide. Thus, the grid-free nature of the conditional GFS is confirmed through this small example.

4.1.2 Conditioning with Dual Cokriging

Dual kriging for univariate conditioning and dual cokriging for multivariate conditioning with global data neighborhood are proposed to reduce computational time during kriging weights computation [10]. The computational reduction is

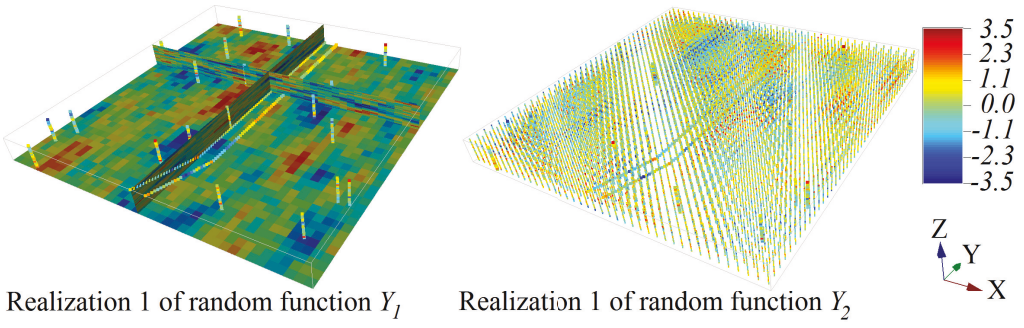


Figure 4.12: Example of multivariate conditional grid-free simulation – first realizations of random functions Y_1 and Y_2: (a) pixel representation of a realization, (b) point representation of a realization.

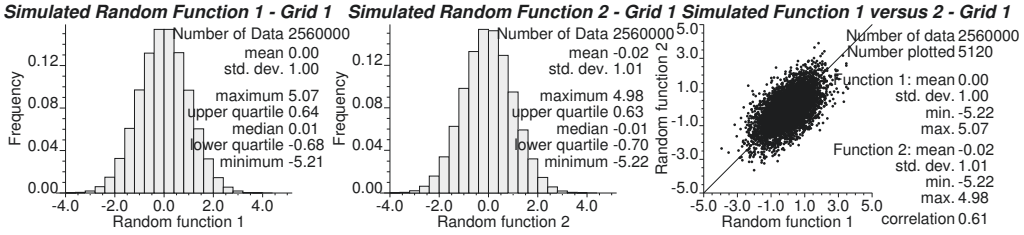


Figure 4.13: Example of multivariate conditional grid-free simulation – histograms and scatter plot of all 50 conditional realizations of the bivariate system.

achieved through matrix manipulation. The dual (co)kriging weights $\boldsymbol{\nu}$ are computed only once per realization for all simulation locations $\boldsymbol{u}$ (computational time is proportional to the number of realizations N_R) in comparison with the primary (co)kriging weights $\boldsymbol{\lambda}(\boldsymbol{u})$, which are computed for every simulation location of all realizations (computational time is proportional to the number of simulation nodes M). This reduces the number of times a matrix multiplication operation is performed for the kriging weights derivation with conditioning to all available data at large number of the simulation nodes in comparison with number of the realizations.

The dual kriging estimation for the univariate case is shown in Eq. (4.30)

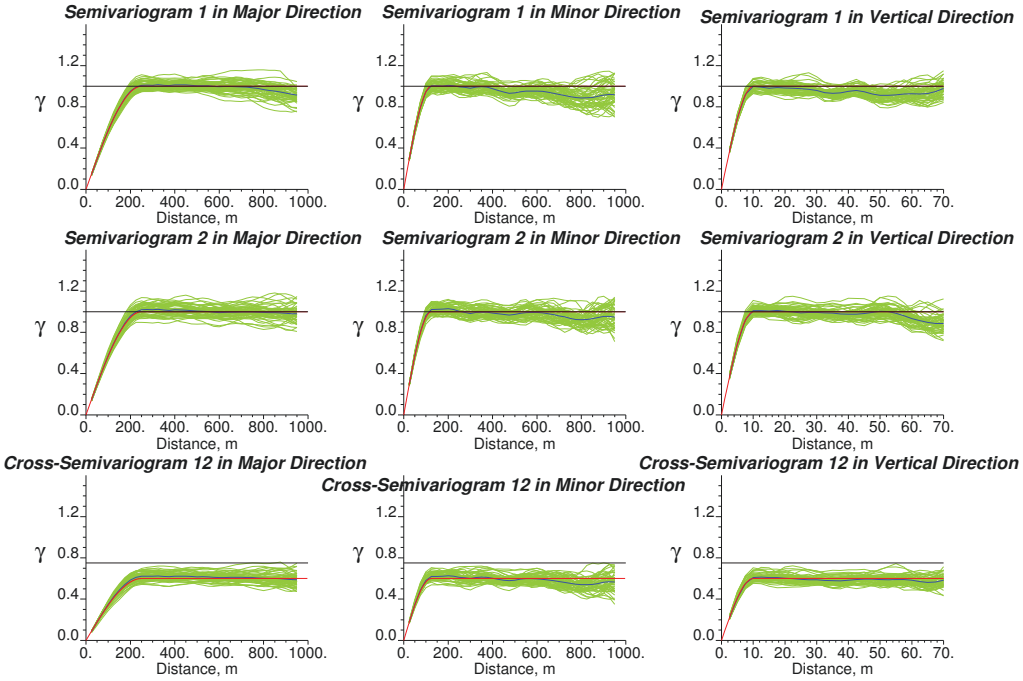


Figure 4.14: Example of multivariate conditional grid-free simulation – experimental direct- and cross-variograms (in green) computed for each realization of the bivariate system for major, minor, and vertical directions of the continuity along with their average (in blue) and variogram model (in red).

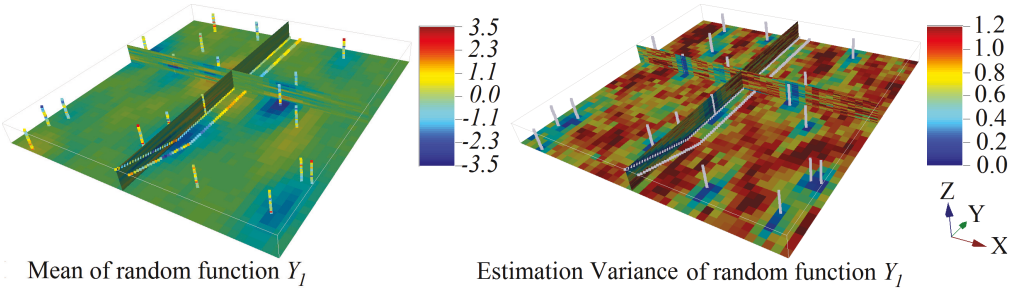


Figure 4.15: Example of multivariate conditional grid-free simulation – mean and estimation variance maps computed from 50 realizations of the random function Y_1.

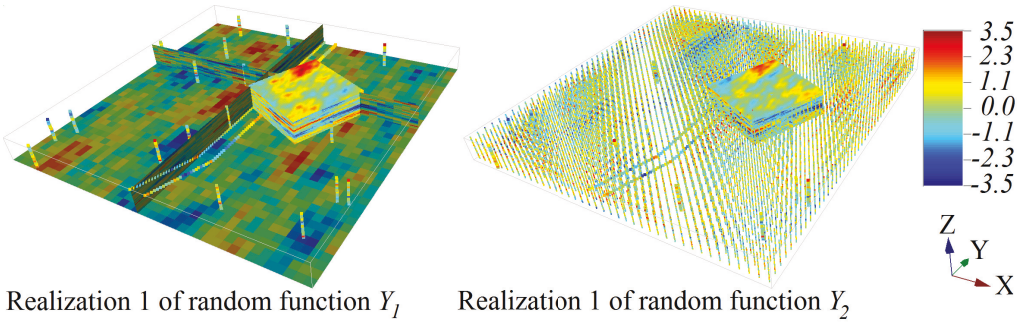


Figure 4.16: Example of multivariate conditional grid-free simulation – coarse and fine resolution representation of first realizations of the bivariate system. Note data reproduction and model consistency between resolutions.

in matrix notation, which is derived by combining Eqs. (4.9) and (4.13). Note that left hand side covariance matrix $\boldsymbol{C}_Y$ is symmetric.

$$\Delta Y^*(\boldsymbol{u}) = [\boldsymbol{c}_Y(\boldsymbol{u})]^T \boldsymbol{\nu} \tag{4.30}$$

$$\boldsymbol{C}_Y \boldsymbol{\nu} = \boldsymbol{Y}^d \tag{4.31}$$

The dual cokriging estimation equations for the multivariate case are identical to the univariate case and are shown in Eqs. (4.32) and (4.33). Note that matrices have slightly different forms for univariate and multivariate cases as has been explained in previous Section 4.1.1.

$$\Delta \boldsymbol{Y}^*(\boldsymbol{u}) = [\boldsymbol{c}_Y(\boldsymbol{u})]^T \boldsymbol{\nu} \tag{4.32}$$

$$\boldsymbol{C}_Y \boldsymbol{\nu} = \boldsymbol{Y}^d \tag{4.33}$$

An example is shown below to demonstrate how the implementation of cokriging in a dual form can reduce computational time keeping the estimate

unchanged. Consider a bivariate 2-D system with spatial structure presented by Eq. (4.34). Random function Y_1 has been sampled at 250 scattered locations, see the location map and histogram in Fig. 4.17. Random function Y_2 has been exhaustively sampled at all 2500 simulation locations. A location map of the Y_2 data and the corresponding histogram are also shown in the same figure. The distributions of both random functions are normal with zero mean and unit variance. These two functions are linearly correlated to each other.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_1}(\boldsymbol{h}) = 0.1Nugget\,(\boldsymbol{h}) + 0.9Sph_{r=20.0}\,(\boldsymbol{h}) \\ C_{Y_1Y_2}(\boldsymbol{h}) = 0.1Nugget\,(\boldsymbol{h}) + 0.6Sph_{r=20.0}\,(\boldsymbol{h}) \\ C_{Y_2}(\boldsymbol{h}) = 0.1Nugget\,(\boldsymbol{h}) + 0.9Sph_{r=20.0}\,(\boldsymbol{h}) \end{cases} \tag{4.34}$$

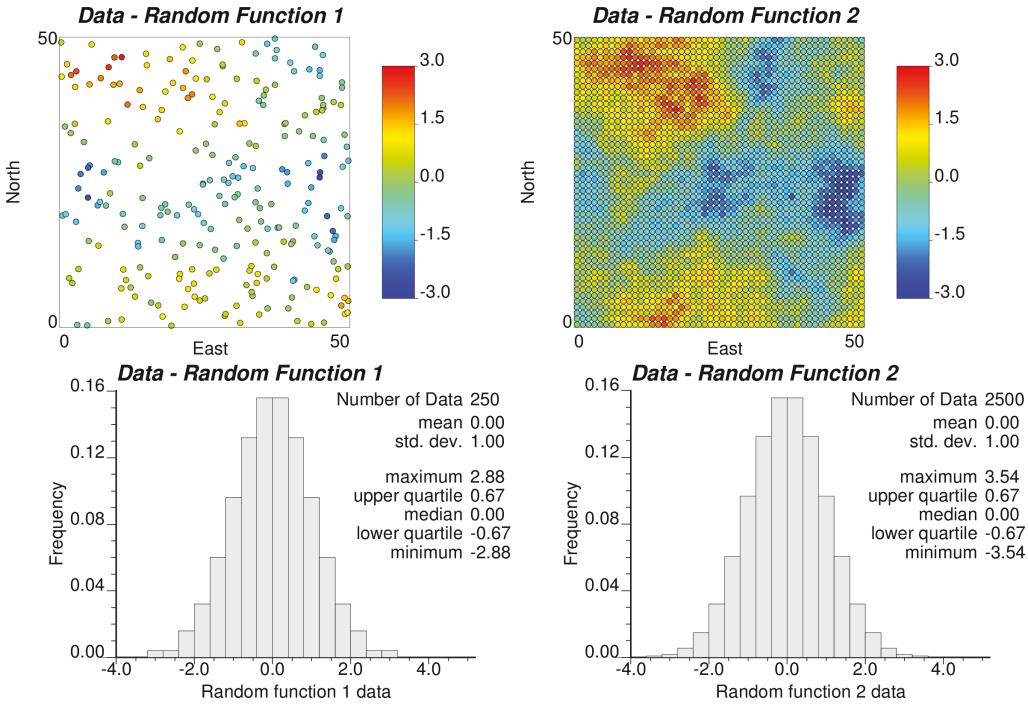


Figure 4.17: Example of dual cokriging – data location maps and histograms of the bivariate system.

The kriging estimation is performed on a given data set with several approaches. Fig. 4.18 contains the resulting estimation maps of random function Y_1 and random function Y_2. In the first approach, random function Y_1 is estimated using only primary data with simple kriging. The estimation map is smooth as expected. In the second approach, both random functions are estimated with the primal form of the simple cokriging conditional to all primary and secondary data. In last third approach, both random functions are estimated with the dual form of the simple cokriging using all given data. It is obvious that the data are reproduced in all approaches. Both cokriging estimates look identical as expected, because these two kriging estimate definitions are mathematically identical and the same conditioning data are used. Kriging estimate Y_1^* differs from cokriging estimates Y_1^* of random function Y_1 because secondary data are used in the conditioning in addition to primary data and secondary data do not have smooth pattern. Computational time associated with these three cases is summarized in Fig. 4.19. The kriging algorithm does not require much time, because only small number of the primary data is taken into account and only random function Y_1 is estimated. In this case study, implementation of the dual form of cokriging reduces computational time by 10% in comparison with the primal cokriging form. The reduction in the computational time will be more significant for bigger models, larger number of conditioning data, and less realizations being conditioned to data from global neighborhood.

4.2 Conditional Grid-Free Simulation with Exhaustively Sampled Data

Quite often in petroleum projects, a secondary geophysical variable is sampled exhaustively and evenly over a large area or volume to infer scarcely sampled primary variables, which are correlated with this secondary variable. Because

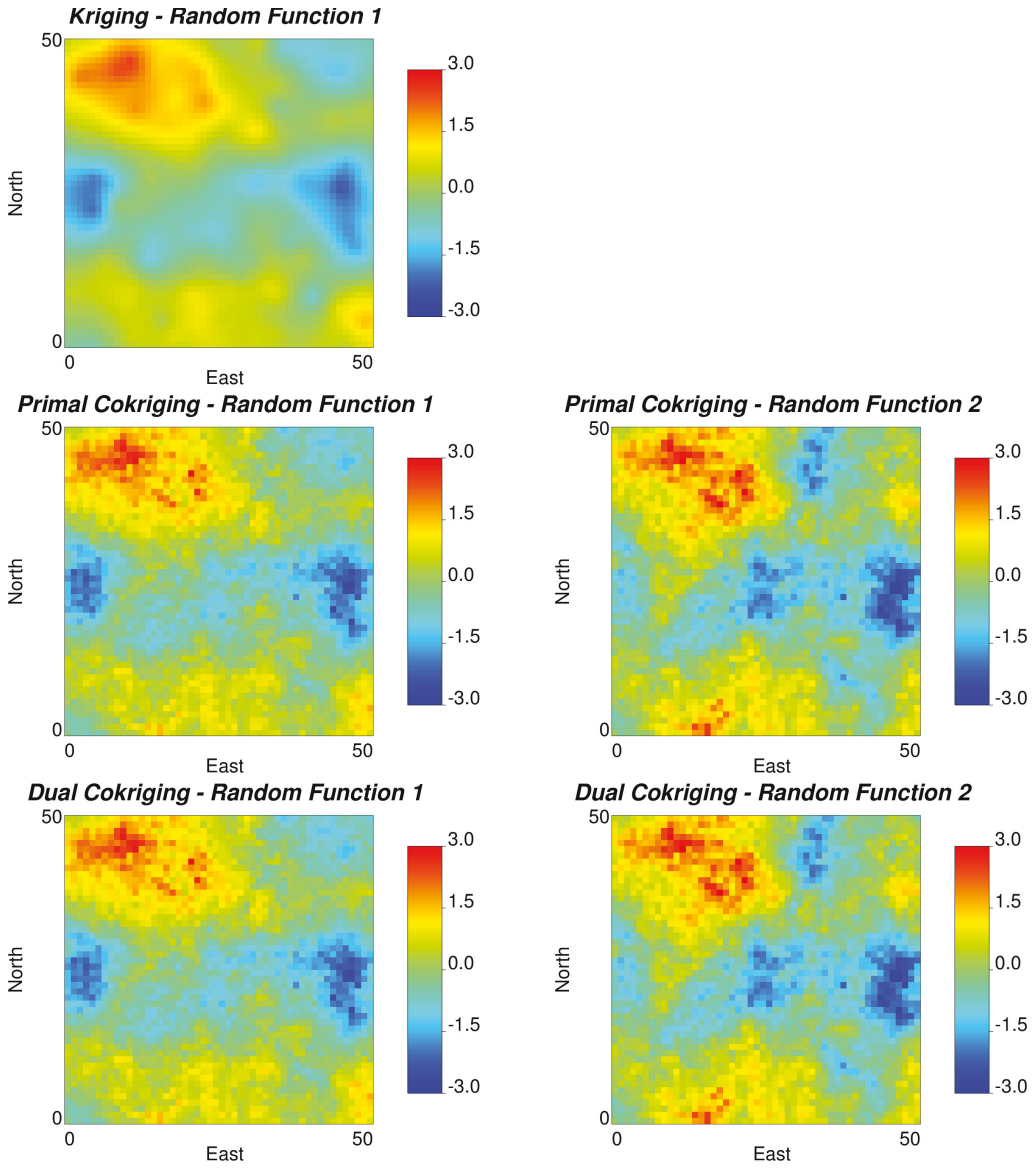


Figure 4.18: Example of dual cokriging – (a) kriging estimation map of the random function 1, (b) - (c) kriging estimation maps of the random functions 1 and 2 in primal form, and (d) - (e) kriging estimation maps of the random functions 1 and 2 in dual form.

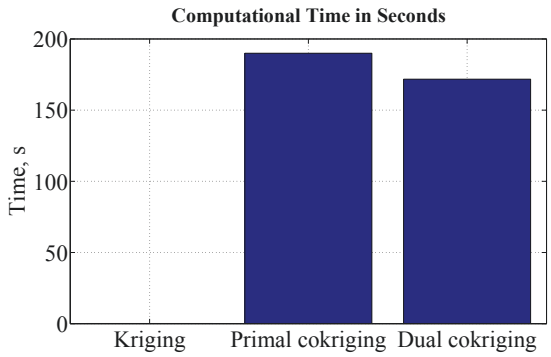


Figure 4.19: Example of dual cokriging – computational time of kriging, primal cokriging, and dual cokriging in seconds.

the number of the data is very large, directly conditioning to all data cannot be performed at a computationally affordable cost as discussed in previous Section 4.1. The intrinsic cokriging (ICK) approach is suggested for conditioning realizations to exhaustively sampled data to ensure manageable computational time and keep estimation quality about the same as if all secondary data are used in the conditioning. The key assumption in ICK is that the secondary data are collocated with all primary data and the secondary at the estimation location screen the rest of the secondary data. This implies that only a small portion of the exhaustively sampled secondary data is used in the conditioning at any one location, which is associated with much less computational burden.

4.2.1 Conditioning with Intrinsic Cokriging

Intrinsic cokriging has been chosen for conditioning multivariate systems mainly to reduce computational time with large exhaustively sampled data sets. Also, ICK reduces variance inflation that is often the case for collocated cokriging, where secondary data only at the estimation location are used along with all primary data [4]. Intrinsic cokriging implies that there are more than one random

function to be simulated, at least one of which is exhaustively sampled either over entire simulation domain or over its portion.

Primary random functions are being estimated, while secondary random functions are used mainly to further constrain estimates of the primary functions. The ICK estimate $Y_k^*(\boldsymbol{u})$ of a primary or secondary random function at location $\boldsymbol{u}$ is conditioned to all scattered data $\sum_{l=1}^{K_S} N_{S,l}$, all exhaustively sampled data $\sum_{l'=1}^{K_E} N_{E,l'}$ at scattered data locations, and all exhaustively sampled data K_E at estimation location $\boldsymbol{u}$, where K_S is the number of the modeled properties sampled at scattered locations, K_E is the number of the modeled properties sampled exhaustively and evenly, and $K_S + K_E = K$. A schematic of the conditioning data in the intrinsic cokriging is shown in Fig. 4.20. The equations used to compute intrinsic cokriging estimate $Y_k^*(\boldsymbol{u})$ conditional to data values, intrinsic cokriging estimate $Y_k^{*|us}(\boldsymbol{u})$ conditional to unconditionally simulated values at data locations, and updating intrinsic cokriging estimate $\Delta Y_k^*(\boldsymbol{u})$ for conditioning unconditional simulation are provided in Eqs. (4.35) – (4.37). Eq. (4.38) is a matrix form of the Eq. (4.37) for all updating intrinsic cokriging estimates $\Delta Y_k^*(\boldsymbol{u}), k = 1, ..., K$. The matrices have been explained in Section 4.1. Note that the number of exhaustively sampled data can be smaller than the total number of the scattered data, when some scattered data of various types are collocated with each other. Therefore, a general inequality between numbers of these two data types is $\sum_{l'=1}^{K_E} N_{E,l'} \leq \sum_{l=1}^{K_S} N_{S,l}$. Also, when the estimation location is collocated with one of the scattered data locations, the last sum in Eqs. (4.35) – (4.37) is ignored to avoid the singularity in the inverse computation of data-data covariance matrix. These equations remain the same despite random function being estimated. Kriging weights are derived in a similar way as before. The system of kriging equations is shown in Eq. (4.39) or in Eq. (4.40) in matrix form. The size of data-data covariance matrix $\boldsymbol{C}_Y$ is $\left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'} + K_E\right) \times$

$\left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'} + K_E\right)$. The size of estimation-data covariance matrix $\boldsymbol{c}_Y(\boldsymbol{u})$ is $\left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'} + K_E\right) \times (K_S + K_E)$. The size of the kriging weights matrix $\boldsymbol{\lambda}(\boldsymbol{u})$ is also $\left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'} + K_E\right) \times (K_S + K_E)$. The conditioning based on ICK is performed in a dual form as explained in Section 4.1.2.

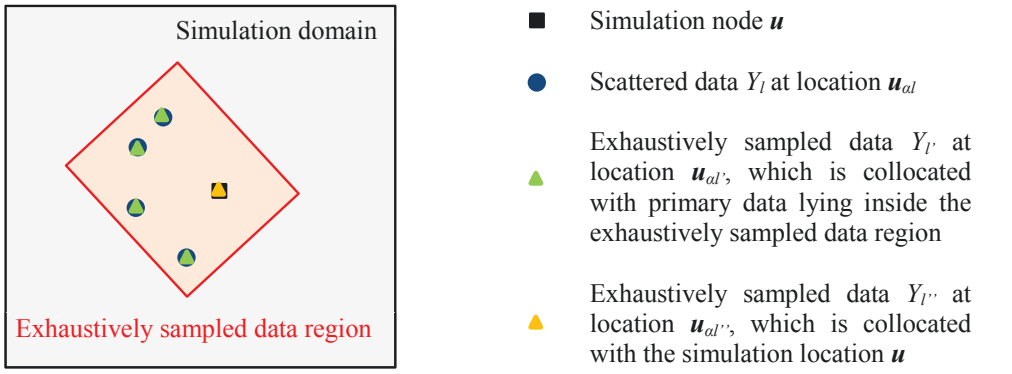


Figure 4.20: Schematic of the conditioning data in the intrinsic cokriging (ICK).

$$
\begin{aligned}
Y_k^*(\boldsymbol{u}) = m_{Y_k}(\boldsymbol{u}) &+ \sum_{l=1}^{K_S} \sum_{\alpha_l=1}^{N_{S,l}} \lambda_{k,\alpha_l}(\boldsymbol{u}) \left(Y_l(\boldsymbol{u}_{\alpha_l}) - m_{Y_l}(\boldsymbol{u}_{\alpha_l})\right) \\
&+ \sum_{l'=1}^{K_E} \sum_{\alpha_{l'}=1}^{N_{E,l'}} \lambda_{k,\alpha_{l'}}(\boldsymbol{u}) \left(Y_{l'}(\boldsymbol{u}_{\alpha_{l'}}) - m_{Y_{l'}}(\boldsymbol{u}_{\alpha_{l'}})\right) \\
&+ \sum_{l''=1}^{K_E} \lambda_{k,\alpha_{l''}}(\boldsymbol{u}) \left(Y_{l''}(\boldsymbol{u}) - m_{Y_{l''}}(\boldsymbol{u})\right), \quad k = 1, ..., K
\end{aligned}
\tag{4.35}
$$

$$
\begin{aligned}
Y_k^{*|us}(\boldsymbol{u}) = m_{Y_k}(\boldsymbol{u}) &+ \sum_{l=1}^{K_S} \sum_{\alpha_l=1}^{N_{S,l}} \lambda_{k,\alpha_l}(\boldsymbol{u}) \left(Y_l^{us}(\boldsymbol{u}_{\alpha_l}) - m_{Y_l}(\boldsymbol{u}_{\alpha_l})\right) \\
&+ \sum_{l'=1}^{K_E} \sum_{\alpha_{l'}=1}^{N_{E,l'}} \lambda_{k,\alpha_{l'}}(\boldsymbol{u}) \left(Y_{l'}^{us}(\boldsymbol{u}_{\alpha_{l'}}) - m_{Y_{l'}}(\boldsymbol{u}_{\alpha_{l'}})\right) \\
&+ \sum_{l''=1}^{K_E} \lambda_{k,\alpha_{l''}}(\boldsymbol{u}) \left(Y_{l''}^{us}(\boldsymbol{u}) - m_{Y_{l''}}(\boldsymbol{u})\right), \quad k = 1, ..., K
\end{aligned} \tag{4.36}
$$

$$
\begin{aligned}
\Delta Y_k^{*}(\boldsymbol{u}) = &\sum_{l=1}^{K_S} \sum_{\alpha_l=1}^{N_{S,l}} \lambda_{k,\alpha_l}(\boldsymbol{u}) \left(Y_l(\boldsymbol{u}_{\alpha_l}) - Y_l^{us}(\boldsymbol{u}_{\alpha_l})\right) \\
&+ \sum_{l'=1}^{K_E} \sum_{\alpha_{l'}=1}^{N_{E,l'}} \lambda_{k,\alpha_{l'}}(\boldsymbol{u}) \left(Y_{l'}(\boldsymbol{u}_{\alpha_{l'}}) - Y_{l'}^{us}(\boldsymbol{u}_{\alpha_{l'}})\right) \\
&+ \sum_{l''=1}^{K_E} \lambda_{k,\alpha_{l''}}(\boldsymbol{u}) \left(Y_{l''}(\boldsymbol{u}) - Y_{l''}^{us}(\boldsymbol{u})\right), \quad k = 1, ..., K
\end{aligned} \tag{4.37}
$$

$$
\Delta\, \boldsymbol{Y}^{*}(\boldsymbol{u}) = \boldsymbol{\lambda}^{T}(\boldsymbol{u})\, \boldsymbol{Y}^{d} \tag{4.38}
$$

$$
\begin{cases}
\sum_{l=1}^{K_S} \sum_{\alpha_l=1}^{N_{S,l}} C_{Y_l Y_j}(\boldsymbol{u}_{\alpha_l} - \boldsymbol{u}_{\alpha'_j}) \lambda_{k,\alpha_l}(\boldsymbol{u}) \\
+ \sum_{l'=1}^{K_E} \sum_{\alpha_{l'}=1}^{N_{E,l'}} C_{Y_{l'} Y_j}(\boldsymbol{u}_{\alpha_{l'}} - \boldsymbol{u}_{\alpha'_j}) \lambda_{k,\alpha_{l'}}(\boldsymbol{u}) \\
+ \sum_{l''=1}^{K_E} C_{Y_{l''} Y_j}(\boldsymbol{u}_{\alpha_{l''}} - \boldsymbol{u}_{\alpha'_j}) \lambda_{k,\alpha_{l''}}(\boldsymbol{u}) \qquad , \; k = 1, ..., K \\
= C_{Y_k Y_j}(\boldsymbol{u} - \boldsymbol{u}_{\alpha'_j}), \\
j = m, \;\; \alpha'_m = 1, ..., N_{S,m}, \;\; m = 1, ..., K_S, \\
j = m', \;\; \alpha'_{m'} = 1, ..., N_{E,m'}, \;\; m' = 1, ..., K_E, \\
j = m'', \;\; \alpha'_{m''} = 1, ..., K_E
\end{cases}
\tag{4.39}
$$

$$
\boldsymbol{C}_Y \boldsymbol{\lambda}(\boldsymbol{u}) = \boldsymbol{c}_Y(\boldsymbol{u}) \tag{4.40}
$$

Three examples are prepared to show that conditioning of unconditional realizations with intrinsic cokriging leads to similar results as conditioning with conventional cokriging. Simulation results are compared for kriging, cokriging, and intrinsic cokriging in a dual form. Intrinsic cokriging enables assimilating a large quantity of the exhaustively sampled data not worrying about matrix inversion size limitation of the data-data covariance matrix $\boldsymbol{C}_Y$ in kriging weights computation. The computational time in dual form implementation of the ICK becomes longer though, because the data-data covariance matrix has to be inverted for every estimation location due to changing collocated data value with estimation location. The computational time is improved by introducing the block matrix inversion scheme explained in the next Section 4.2.2, where a smaller portion of the covariance matrix has to be inverted for every estimation location.

The set up of the modeling system is explained next. The studied system consists of two correlated random functions primary Y_1 and secondary Y_2 in normal space. The correlation coefficient is 0.7. The first random function is sampled at 250 scattered locations. The second random function is measured evenly over the entire simulation domain at 2500 locations. The exhaustively sampled data are collocated with every simulation location. The data locations maps are shown in Fig. 4.21. Both data follow a standard normal distribution $N(0,1)$. The spatial structure of the system is highlighted in Eq. (4.41), where the experimental spatial structure of the random function Y_1 has more importance in the definition of the linear model of coregionalization. A total of $N_R = 50$ conditional realizations are generated, where 1001 Fourier coefficients are used for each of 100 turning lines simulations to avoid obvious artifacts and properly decompose target covariance functions (see Chapter 5 for more details).

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_1}(\boldsymbol{h}) = 0.1Nugget(\boldsymbol{h}) + 0.3Sph_{\substack{r_1=6.0\\r_2=19.0\\\alpha=90°}}(\boldsymbol{h}) \\ \qquad\qquad + 0.6Sph_{\substack{r_1=23.0\\r_2=19.0\\\alpha=90°}}(\boldsymbol{h}) \\ C_{Y_1Y_2}(\boldsymbol{h}) = 0.1Nugget(\boldsymbol{h}) + 0.2Sph_{\substack{r_1=6.0\\r_2=19.0\\\alpha=90°}}(\boldsymbol{h}) \\ \qquad\qquad + 0.4Sph_{\substack{r_1=23.0\\r_2=19.0\\\alpha=90°}}(\boldsymbol{h}) \\ C_{Y_2}(\boldsymbol{h}) = 0.1Nugget(\boldsymbol{h}) + 0.3Sph_{\substack{r_1=6.0\\r_2=19.0\\\alpha=90°}}(\boldsymbol{h}) \\ \qquad\qquad + 0.6Sph_{\substack{r_1=23.0\\r_2=19.0\\\alpha=90°}}(\boldsymbol{h}) \end{cases} \tag{4.41}$$

In the first case, only the random function Y_1 is simulated and conditioned with kriging to its scattered data. Simulation results, which include a map of the first realization, histogram of all realizations, mean map of all realizations, and

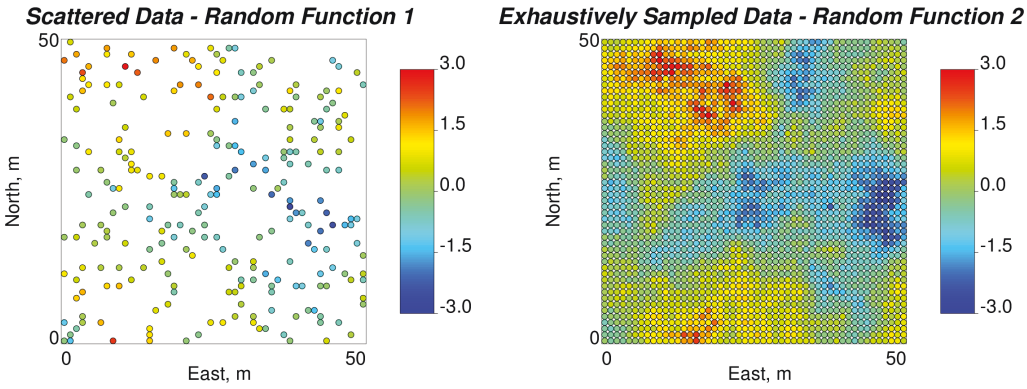


Figure 4.21: Example of the conditioning using the conventional cokriging and intrinsic cokriging – data locations.

estimation error variance map, are shown in Fig. 4.22. Data reproduction and normality of the simulated realizations are ensured. The variogram reproduction is shown in Fig. 4.23. The experimental variograms computed from individual realizations are shown in green with their average in blue, the variogram model is presented by a red line. Experimental variogram points computed from the data are shown as blue dots. The experimental variograms closely follow the variogram model for shorter ranges with deviations towards the data experimental variogram for longer ranges.

In the second case, both random functions Y_1 and Y_2 are simulated and conditioned with cokriging to all primary scattered data and all secondary exhaustively sampled data. Simulation results of both random functions are shown in Fig. 4.24 and Fig. 4.25. Data are reproduced, and realizations closely follow standardized normal distribution. Because there is a large number of primary conditioning data, the first realization of random function Y_1 conditioned to all primary and secondary data looks very similar to the first realization of the same random function conditioned to only primary data by kriging in the previous case. The exhaustively sampled data of random function Y_2 are not so smooth, which leads to the mean of 50 realizations of random function Y_1 being not so

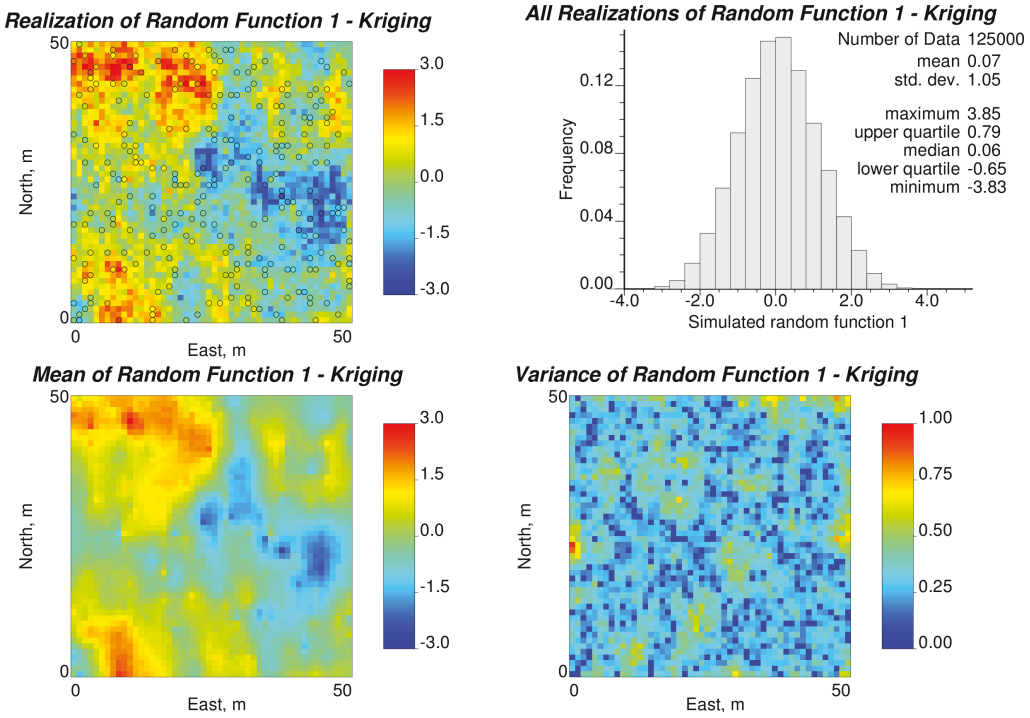


Figure 4.22: Example of the conditioning using the conventional kriging – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_1.

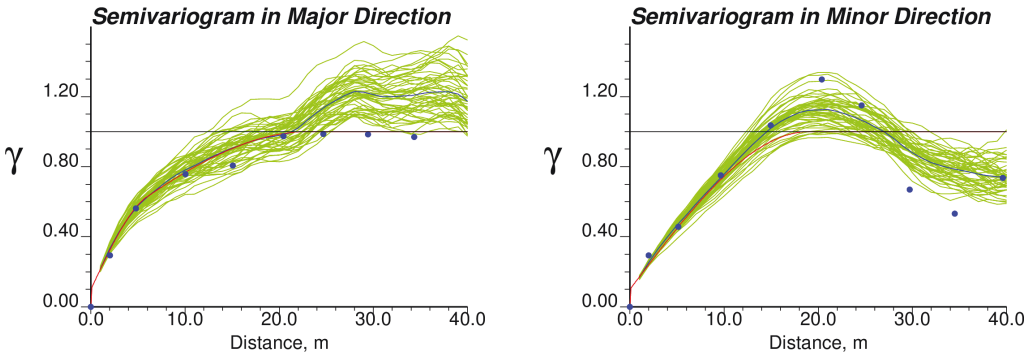


Figure 4.23: Example of the conditioning using the conventional kriging – experimental variograms in green computed from 50 realizations, their average in blue, and variogram model in red of the random function Y_1 in major and minor directions of the continuity. Blue dots are experimental variograms of random function Y_1 computed from data.

smooth either in comparison with the simulation mean conditioned by kriging in previous case. Also, the estimation error variance map of random function Y_1 is less in magnitude due to a larger number of conditioning data. The estimation error variance map of random function Y_2 is zero as expected at every simulation location, because there is a data value at every simulation location. The correlation coefficient of the generated realizations is shown on a scatter plot in Fig. 4.26. It is 0.8, which is a bit higher than target 0.7. The bivariate distribution does not appear binormal. These might be caused by absence of degrees of freedom for random function Y_2, which is sampled at every simulation location. This fact may also over-constrain the simulated values of random function Y_1. The variogram reproduction can be found in Fig. 4.27. Because exhaustively sampled data are dominant data type, the experimental variograms of random function Y_1 and cross-variogram between random function Y_1 and Y_2, which are shown in green lines, lie between variogram model in red and data experimental variogram of random function Y_2, shown as violet dots.

In the third case, simulation of both random functions is conditioned with the intrinsic cokriging formalism. Here, all scattered data of random function Y_1 and exhaustively sampled data of random function Y_2 collocated with the scattered data and simulation location are used in conditioning of the simulation. Simulation results are presented in Figs. 4.28 – 4.30 with the variogram reproduction in Fig. 4.31. The realizations are identical to the realizations from previous case with the cokriging-based conditioning. The reason for the similarity lies in the screening effect, which states that among all exhaustively sampled secondary data the highest weight is assigned to the data collocated with the simulation location [10]. Therefore, under these circumstances it is plausible to replace cokriging with the intrinsic cokriging, which leads to similar simulation results. The comparison of the computational time is discussed next.

The simulations are performed on a 64-bit Windows machine with Intel i7

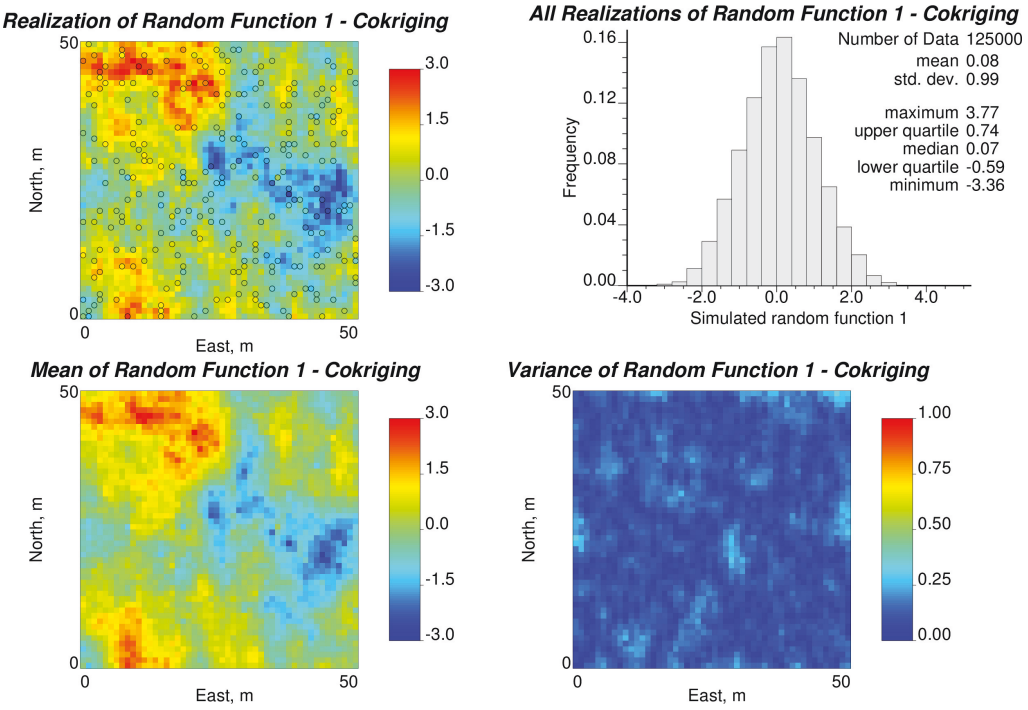


Figure 4.24: Example of the conditioning using the cokriging – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_1.

processor of 2.8 GHz and 24.0 GB RAM. The required computational time varies for each of these three cases, and is summarized in Fig. 4.32. It is obvious that the least computationally demanding case is kriging, because the least number of data (250) among three cases are used in the conditioning. It requires 769 seconds to simulate 50 conditional realizations conditioned by kriging. Simulation based on the cokriging takes about eleven times longer or 9063 seconds. Mostly it is because the number of conditioning data is eleven times larger (2750 data values). When intrinsic cokriging is implemented for the conditioning, the data-data covariance matrix has to be inverted every time for new simulation location. This leads to significantly increased computational time for the third case – approximately 52150 seconds for 50 realizations. Conditioning data changes

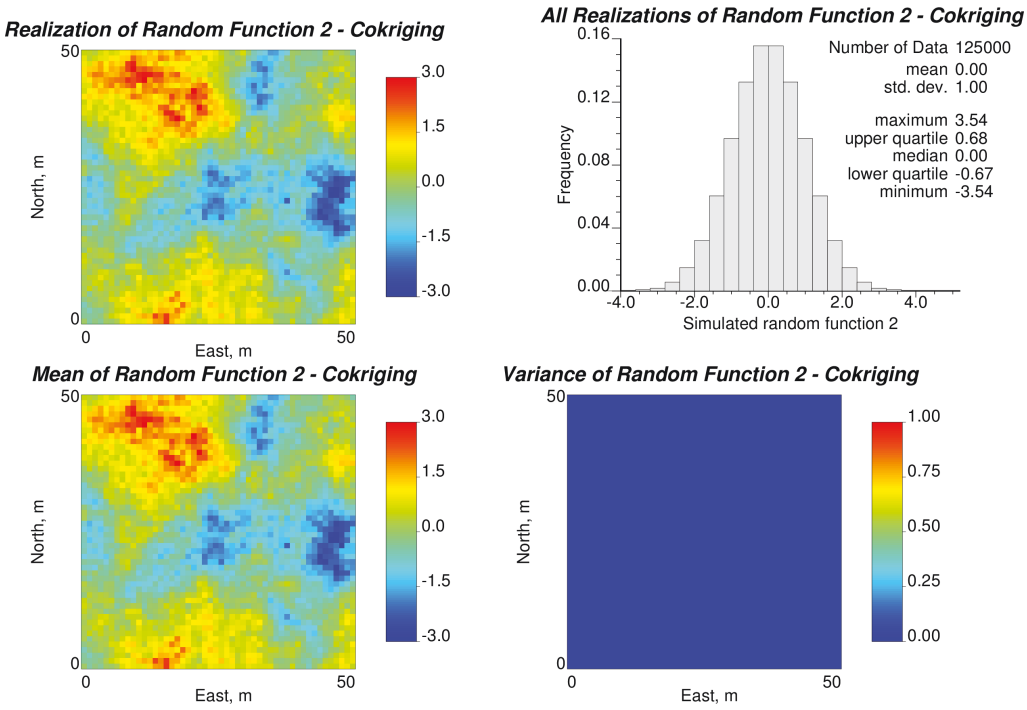


Figure 4.25: Example of the conditioning using the cokriging – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_2.

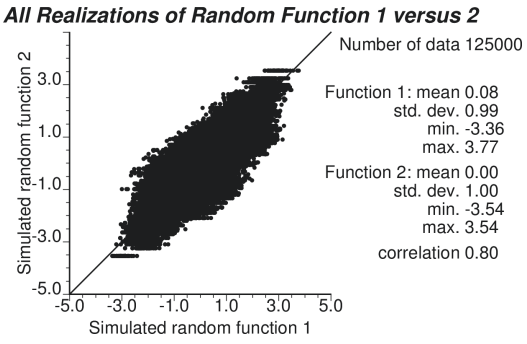


Figure 4.26: Example of the conditioning using the cokriging – scatter plot between all realizations of the simulated random functions Y_1 and Y_2.

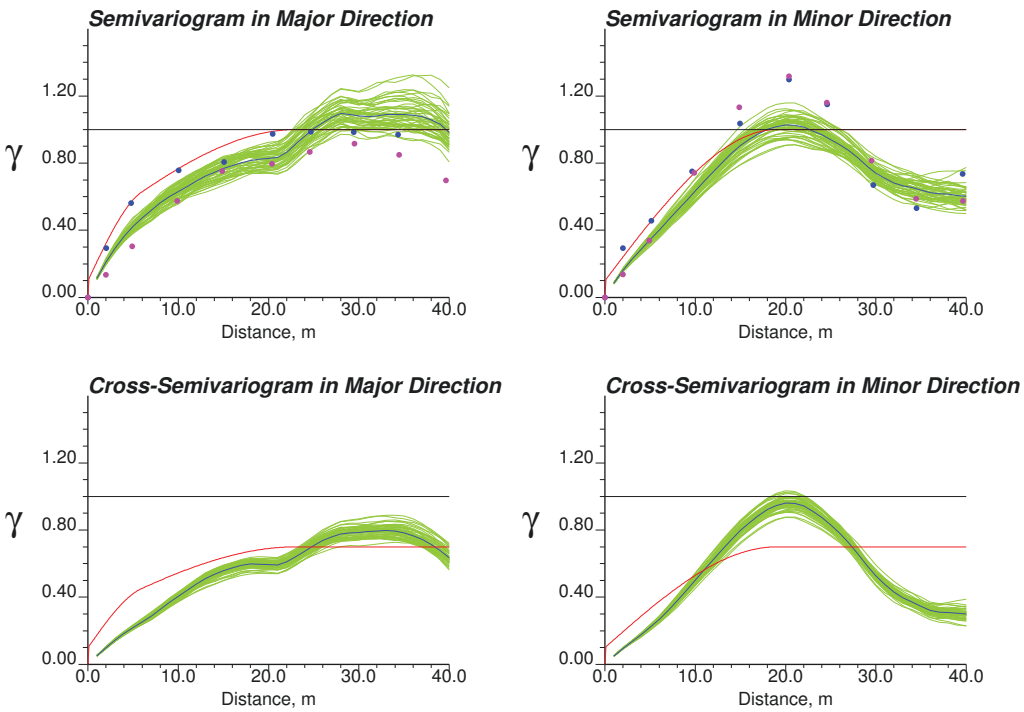


Figure 4.27: Example of the conditioning using the cokriging – experimental variograms in green computed from 50 realizations, their average in blue, and variogram model in red of the random functions Y_1 and Y_2 in major and minor directions of the continuity. Blue dots are experimental variograms of random function Y_1 computed from data, and violet dots are computed from data of random function Y_2.

for every simulation location, which is either 501, when simulation location does not overlap with any scattered data of random function Y_1, or 500, when there is an overlap. Thus, intrinsic cokriging is a computationally demanding approach, unless a modification is applied to the implementation of the algorithm. Block matrix inversion (BMI) is proposed for the data covariance matrix inversion step to reduce computational time. The BMI is discussed next in Section 4.2.2. Just to note that computational time reduces significantly to 9104 seconds for the similar simulation study with the BMI. The reduced time is similar to the cokriging case. The BMI approach surpasses the cokriging substantially in computational

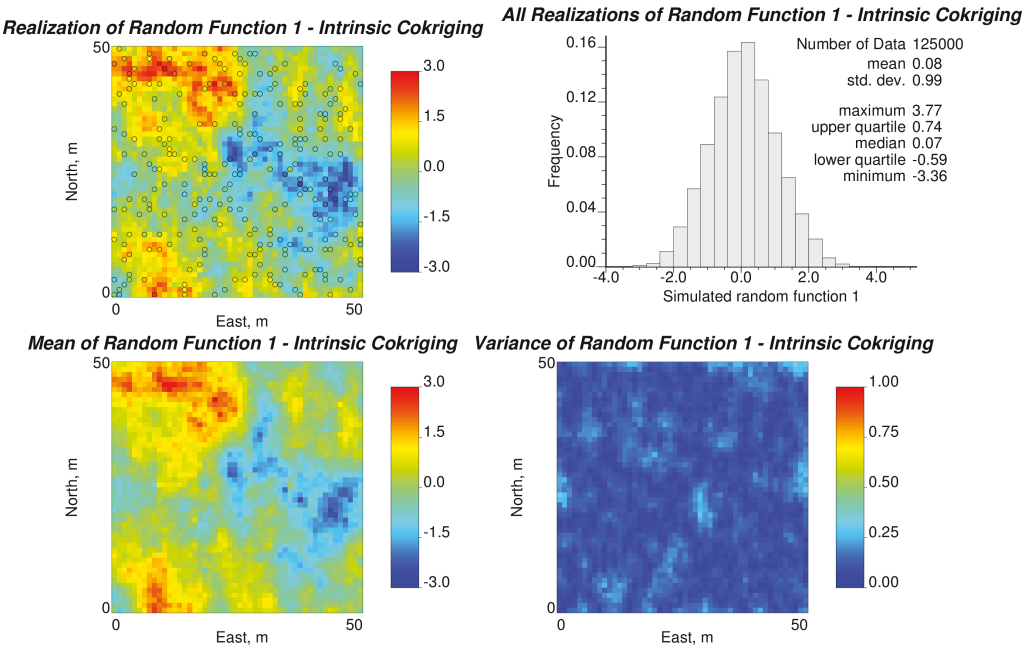


Figure 4.28: Example of the conditioning using the intrinsic cokriging – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_1.

time in practice, when exhaustively sampled data are very large in number in comparison with the scattered data.

As a conclusion to these three cases, in the presence of exhaustively sampled data, the simulation results are similar for realizations conditioned by cokriging and intrinsic cokriging. It is also found that the spatial structure of the random function Y_1 depends on the spatial structure of the random function Y_2, because of its larger number of data and a reasonably high correlation in this case. The computational time is lower for the block matrix inversion implementation of the intrinsic cokriging algorithm than original intrinsic cokriging.

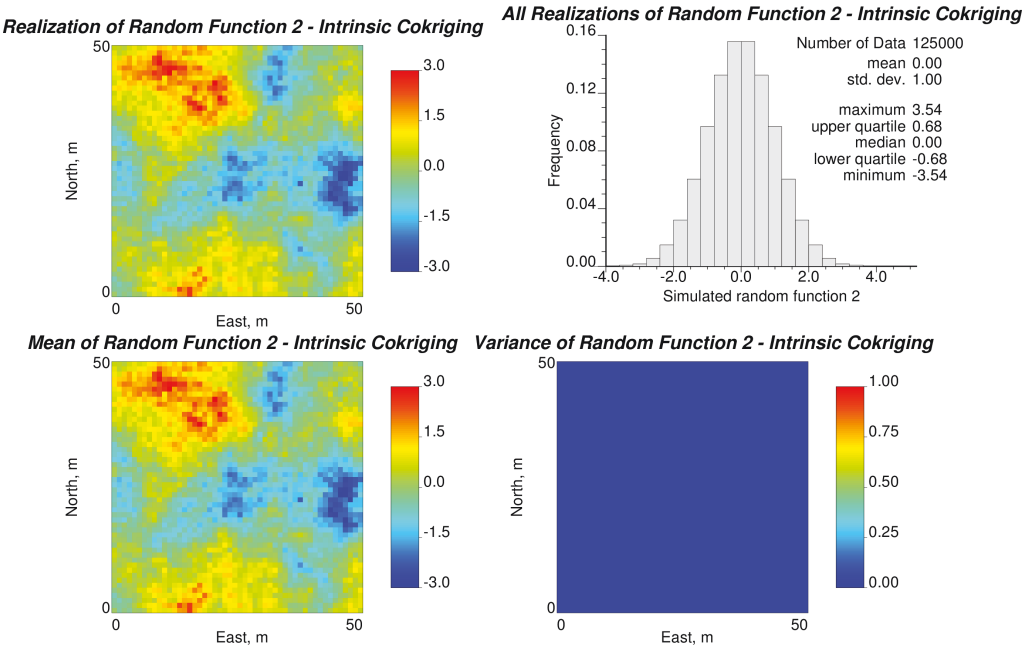


Figure 4.29: Example of the conditioning using the intrinsic cokriging – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_2.

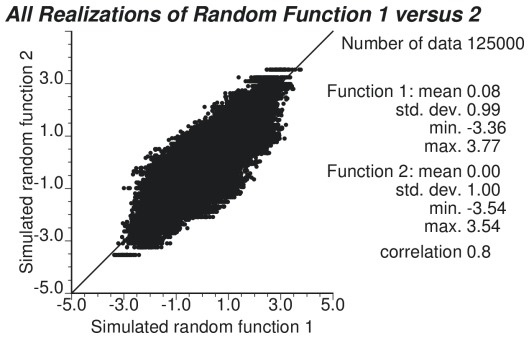


Figure 4.30: Example of the conditioning using the intrinsic cokriging – scatter plot between all realizations of the simulated random functions Y_1 and Y_2.

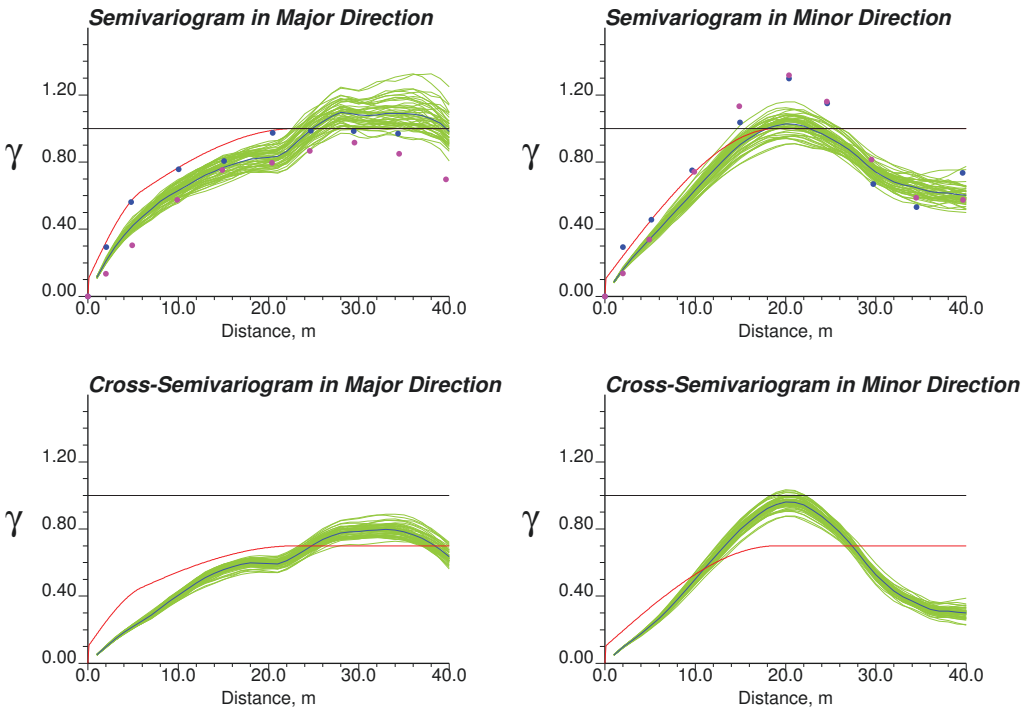


Figure 4.31: Example of the conditioning using the intrinsic cokriging – experimental variograms in green computed from 50 realizations, their average in blue, and variogram model in red of the random functions Y_1 and Y_2 in major and minor directions of the continuity. Blue dots are experimental variograms of random function Y_1 computed from data, and violet dots are computed from data of random function Y_2.

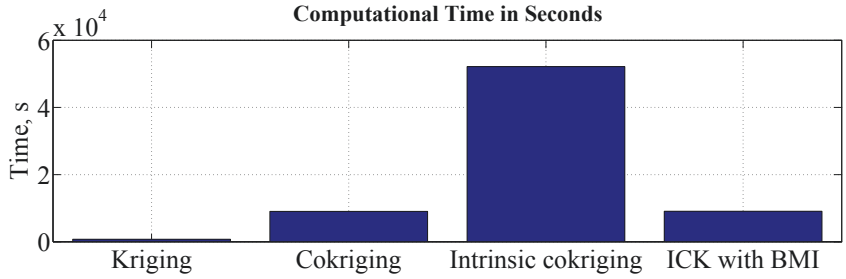


Figure 4.32: Example of the conditioning using the kriging, cokriging, intrinsic cokriging, and intrinsic cokriging with block matrix inversion – comparison of computational time.

4.2.2 Block Matrix Inversion

The block matrix inversion is proposed for inversion of the data-data covariance matrix $\boldsymbol{C}_Y$ to reduce computational time in the intrinsic cokriging-based conditioning, when a large number of exhaustively sampled data is assimilated into the model along with the scattered data.

The data within the data-data covariance matrix can be grouped into fixed data, which do not change over simulation locations, and varying data, which vary from one simulation location to another. The fixed data comprise scattered data and exhaustively sampled data collocated with the scattered data. Their number $\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'}$ is relatively large. The varying data consist of the exhaustively sampled data collocated with the simulation location. Their number K_E is limited only to the number of the random functions that are exhaustively sampled. Thus, when the data covariance matrix $\boldsymbol{C}_Y$ of size $\left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'} + K_E\right) \times \left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'} + K_E\right)$ shown in Eq. (4.42) is examined closely, it can be divided into four parts based on the covariances $\boldsymbol{C}_{Y_F Y_F}$ between fixed data, covariances $\boldsymbol{C}_{Y_F Y_V}$ between fixed and varying data, covariances $\boldsymbol{C}_{Y_V Y_F}$ between varying and fixed data, and covariances $\boldsymbol{C}_{Y_V Y_V}$ between and varying and varying data for each simulation location. This subdivision implies that data-data covariance matrix has a block matrix form. The largest covariance part $\boldsymbol{C}_{Y_F Y_F}$ of size $\left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'}\right) \times \left(\sum_{l=1}^{K_S} N_{S,l} + \sum_{l'=1}^{K_E} N_{E,l'}\right)$ is fixed, while values of other three parts of the data-data covariance matrix vary with the simulation location. The inverse of the data-data covariance matrix is shown in Eq. (4.43), which is also subdivided into four parts (matrices). The combined size of these four parts is the same as the size of the original covariance matrix $\boldsymbol{C}_Y$. Each part is computed independently from the other parts as shown in Eqs. (4.44) – (4.47) according to the block matrix

inversion theorem [25]. The inverse $\boldsymbol{C}^{-1}_{Y_F Y_F}$ is computed only once for all simulation locations and realizations. It is the most computationally intense step in the block matrix inversion. Another inversion operation in the block matrix inversion algorithm involves inverse $\boldsymbol{C}^{-1}_{Y_V Y_V}$ of covariance matrix between varying data of small size K_E x K_E. Other operations carried out in the block matrix inversion are matrix multiplication. Matrix multiplication is faster than matrix inversion. Therefore, the inclusion of the block matrix inversion algorithm to the intrinsic cokriging improves computational time. The proposed matrix inversion algorithm does not call for expensive matrix inversion for each simulation location, but rather requires matrix multiplication.

$$\boldsymbol{C}_Y = \begin{bmatrix} \boldsymbol{C}_{Y_F Y_F} & \boldsymbol{C}_{Y_F Y_V} \\ \boldsymbol{C}_{Y_V Y_F} & \boldsymbol{C}_{Y_V Y_V} \end{bmatrix} \tag{4.42}$$

$$\boldsymbol{C}_Y^{-1} = \begin{bmatrix} \boldsymbol{C}_{Y_F Y_F} & \boldsymbol{C}_{Y_F Y_V} \\ \boldsymbol{C}_{Y_V Y_F} & \boldsymbol{C}_{Y_V Y_V} \end{bmatrix}^{-1} = \begin{bmatrix} \left[\boldsymbol{C}_Y^{-1}\right]_{FF} & \left[\boldsymbol{C}_Y^{-1}\right]_{FV} \\ \left[\boldsymbol{C}_Y^{-1}\right]_{VF} & \left[\boldsymbol{C}_Y^{-1}\right]_{VV} \end{bmatrix} \tag{4.43}$$

$$\begin{aligned} &\left[\boldsymbol{C}_Y^{-1}\right]_{FF} = \boldsymbol{C}^{-1}_{Y_F Y_F} \\ &+ \boldsymbol{C}^{-1}_{Y_F Y_F} \boldsymbol{C}_{Y_V Y_F} \left[\boldsymbol{C}_{Y_V Y_V} - \boldsymbol{C}_{Y_F Y_V} \boldsymbol{C}^{-1}_{Y_F Y_F} \boldsymbol{C}_{Y_V Y_F}\right]^{-1} \boldsymbol{C}_{Y_F Y_V} \boldsymbol{C}^{-1}_{Y_F Y_F} \end{aligned} \tag{4.44}$$

$$\left[\boldsymbol{C}_Y^{-1}\right]_{FV} = -\boldsymbol{C}^{-1}_{Y_F Y_F} \boldsymbol{C}_{Y_V Y_F} \left[\boldsymbol{C}_{Y_V Y_V} - \boldsymbol{C}_{Y_F Y_V} \boldsymbol{C}^{-1}_{Y_F Y_F} \boldsymbol{C}_{Y_V Y_F}\right]^{-1} \tag{4.45}$$

$$\left[\boldsymbol{C}_Y^{-1}\right]_{VF} = -\left[\boldsymbol{C}_{Y_V Y_V} - \boldsymbol{C}_{Y_F Y_V} \boldsymbol{C}^{-1}_{Y_F Y_F} \boldsymbol{C}_{Y_V Y_F}\right]^{-1} \boldsymbol{C}_{Y_F Y_V} \boldsymbol{C}^{-1}_{Y_F Y_F} \tag{4.46}$$

$$\left[\boldsymbol{C}_{Y}^{-1}\right]_{VV} = \left[\boldsymbol{C}_{Y_V Y_V} - \boldsymbol{C}_{Y_F Y_V}\, \boldsymbol{C}_{Y_F Y_F}^{-1}\, \boldsymbol{C}_{Y_V Y_F}\right]^{-1} \tag{4.47}$$

An example of conditioning simulation with the intrinsic cokriging implemented along with the block matrix inversion is given in the third case study in the previous Section 4.2.1. Figs. 4.33 – 4.35 present simulation results for both random functions Y_1 and Y_2. Fig. 4.36 shows variogram reproduction. The results are identical to the intrinsic cokriging without block matrix inversion. On the other hand, computational time drops by 5.7 times from 52150 seconds to 9104 seconds (Fig. 4.32). The BMI is an effective algorithm to improve computational time of the intrinsic cokriging without sacrificing any estimation quality.

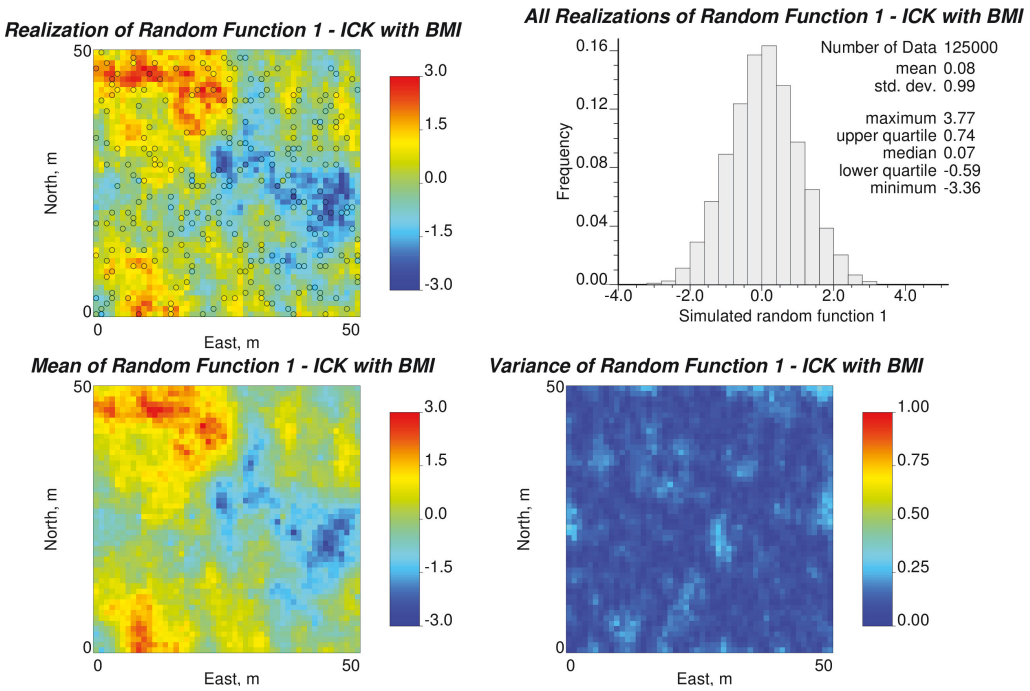


Figure 4.33: Example of the conditioning using the intrinsic cokriging with block matrix inversion – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_1.

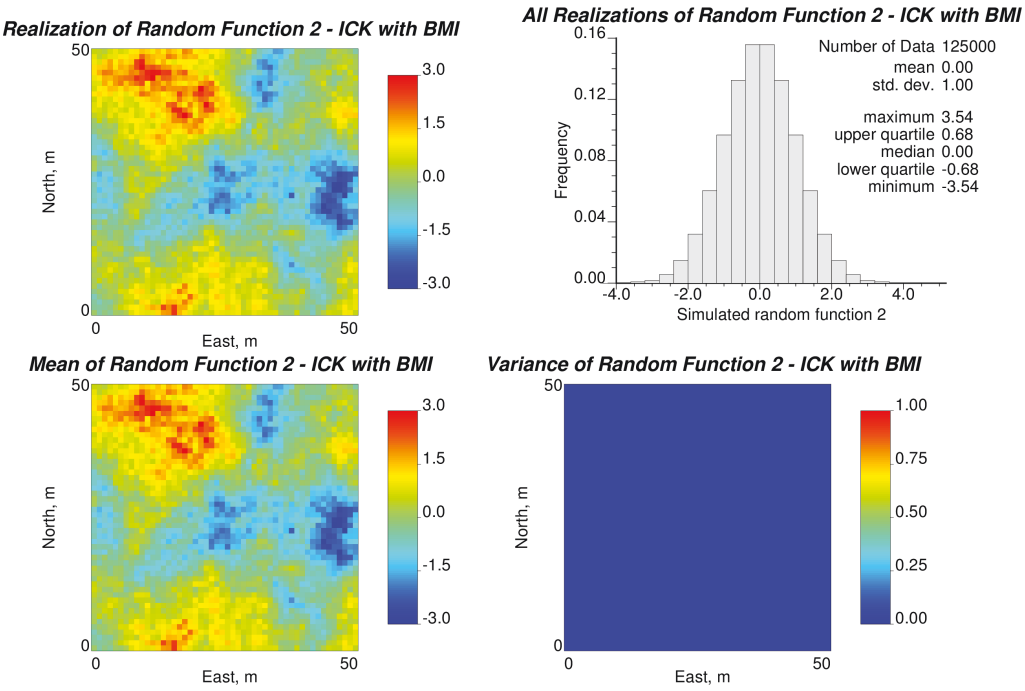


Figure 4.34: Example of the conditioning using the intrinsic cokriging with block matrix inversion – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_2.

4.2.3 Projection in Intrinsic Cokriging

In most of the cases the "exhaustively" sampled data do not cover the entire modeling region. When intrinsic cokriging is applied for the conditioning, some scattered data may not have a collocated exhaustively sampled data. Some of the locations being simulated would not have a collocated secondary data value either. If the conditioning proceeds with missing exhaustively sampled data, artifacts along the edges of the exhaustively sampled data occur in the final realization. In order to avoid this edge effect artifact, conditioning is performed with the projection of the scattered data locations and simulation location onto the exhaustively sampled data. The scattered data locations and simulation node

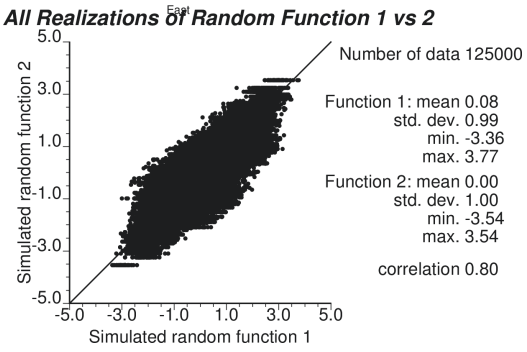


Figure 4.35: Example of the conditioning using the intrinsic cokriging with block matrix inversion – scatter plot between all realizations of the simulated random functions Y_1 and Y_2.

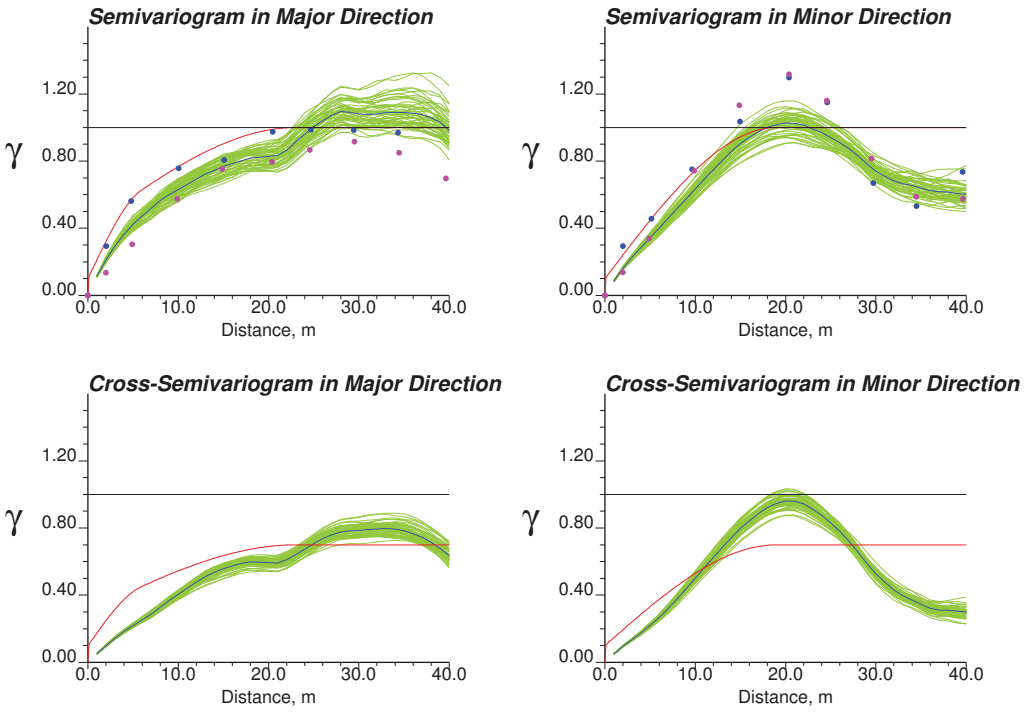


Figure 4.36: Example of the conditioning using the intrinsic cokriging with block matrix inversion – experimental variograms in green computed from 50 realizations, their average in blue, and variogram model in red of the random functions Y_1 and Y_2 in major and minor directions of the continuity. Blue dots are experimental variograms of random function Y_1 computed from data, and violet dots are computed from data of random function Y_2.

location themselves are not shifted. Instead, the exhaustively sampled data at the projected locations are used in the conditioning along with the primary data values. The projection approach reduces the correlation between the primary data value or simulation location and secondary data at the projected location according to an intrinsic model for the cross-variogram $C_{Y_1Y_2}(\boldsymbol{h})$ [4, 77]. The schematic of the conditioning using the intrinsic cokriging with the data projection is shown in Fig. 4.37. The projection approach is proposed to keep number of the secondary data constant for all simulation locations. This makes size of the data-data covariance matrix consistent in application of the block matrix inversion for all simulation locations. The alternative approach to avoid edge effect is to use several secondary data within the local data neigborhood around simulation location.

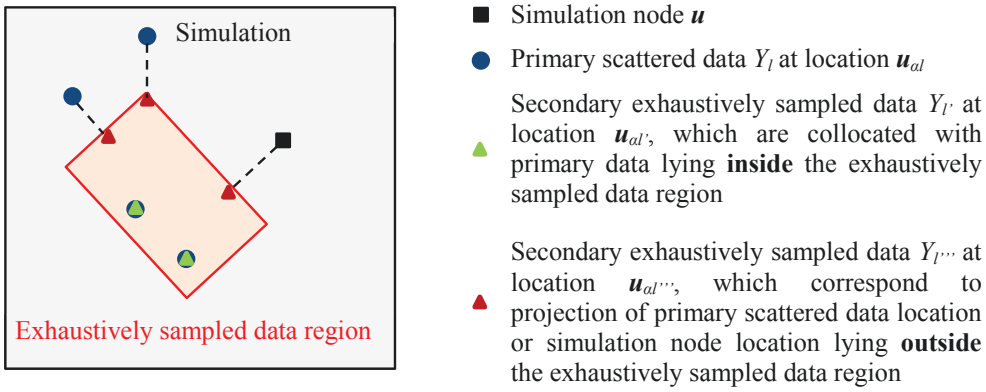


Figure 4.37: Schematic of the conditioning data in the intrinsic cokriging (ICK) with projection.

Three cases are compared to illustrate the significance of the projection approach: conditioning with conventional cokriging, intrinsic cokriging without projection, and intrinsic cokriging with projection. The examples are similar to the bivariate case studies shown in the previous two sections. The number of data is reduced to emphasize the importance of the projection and highlight

edge effects. The total number of scattered data is 25. Exhaustively sampled data set is represented by 625 (25 × 25) evenly spaced data. The data locations over a 50.0 × 50.0 m^2 simulation area are shown in Fig. 4.38. The data follow a Gaussian $N(0,1)$ distribution. The experimental variograms computed from the larger exhaustively sampled data (not shown here) and fitted variogram model are presented in Fig. 4.39, where the blue line corresponds to the direction of maximum continuity (10° azimuth), the violet line is the variogram in the direction of minimum continuity (100° azimuth), and the red lines are variogram model. The expression of the corresponding covariance model is shown in Eq. (4.48). The correlation coefficient between these two random functions is 0.70. The simulation domain is defined by 2500 (50 × 50) evenly placed simulation nodes on a grid. A total of 50 realizations are generated for each case. There are 100 lines, where each 1-D line simulation considered 1001 Fourier coefficients.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_1}(\boldsymbol{h}) = 0.6Sph_{\substack{r_1=10.0\\r_2=10.0\\\alpha=10°}}(\boldsymbol{h}) + 0.4Sph_{\substack{r_1=25.0\\r_2=10.0\\\alpha=10°}}(\boldsymbol{h}) \\ C_{Y_1Y_2}(\boldsymbol{h}) = 0.3Sph_{\substack{r_1=10.0\\r_2=10.0\\\alpha=10°}}(\boldsymbol{h}) + 0.4Sph_{\substack{r_1=25.0\\r_2=10.0\\\alpha=10°}}(\boldsymbol{h}) \\ C_{Y_2}(\boldsymbol{h}) = 0.6Sph_{\substack{r_1=10.0\\r_2=10.0\\\alpha=10°}}(\boldsymbol{h}) + 0.4Sph_{\substack{r_1=25.0\\r_2=10.0\\\alpha=10°}}(\boldsymbol{h}) \end{cases} \tag{4.48}$$

The first case consists of the simulation of both random functions Y_1 and Y_2 conditioned to all available data with the conventional cokriging formalism. It is a reference case, with which ICK with and without projection is compared. Simulation of the random function Y_1, which includes map of the first realization, histogram of all realizations, and maps of the realizations' mean and estimation error variance, are shown in Fig. 4.40. The same results for the random function Y_2 are shown in Fig. 4.41. It is clear that the data are reproduced, the realizations are normal, and the uncertainty map meets the theoretical prediction.

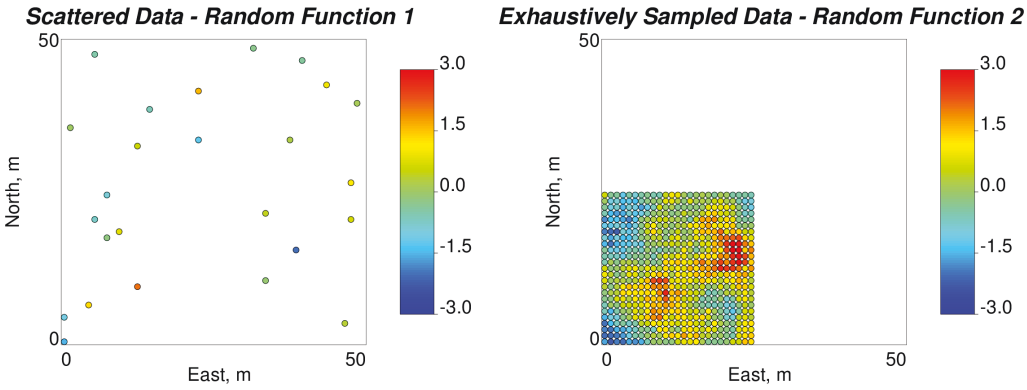


Figure 4.38: Example of the conditioning using the intrinsic cokriging without and with projection – data locations.

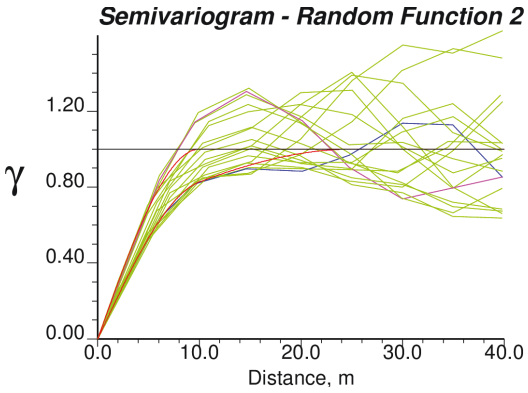


Figure 4.39: Example of the conditioning using the intrinsic cokriging projection – experimental variograms and variogram model of the random function Y_2. Green lines are experimental variograms, blue line corresponds to the direction of maximum continuity (10° azimuth), violet line is the variogram computed in the direction of minimum continuity (100° azimuth), and red lines are fitted varioram model.

The means of the realizations of both random functions are smooth outside of the exhaustively sampled data region and closely follow spatial pattern of the secondary data inside the region. There is no edge effect, but a smooth transition between the secondary data region and the rest of the simulation domain. The spread of the transition zone is determined by the spatial correlation range. The bivariate normality of the system is honored and is presented in the form of the scatter plot between simulated realizations of both random functions in Fig. 4.42. Variogram reproduction is illustrated in Fig. 4.43. The variogram is reproduced quite well for shorter ranges. There is slight deviation of experimental variograms from the model in the minor direction of continuity for the random function Y_2.

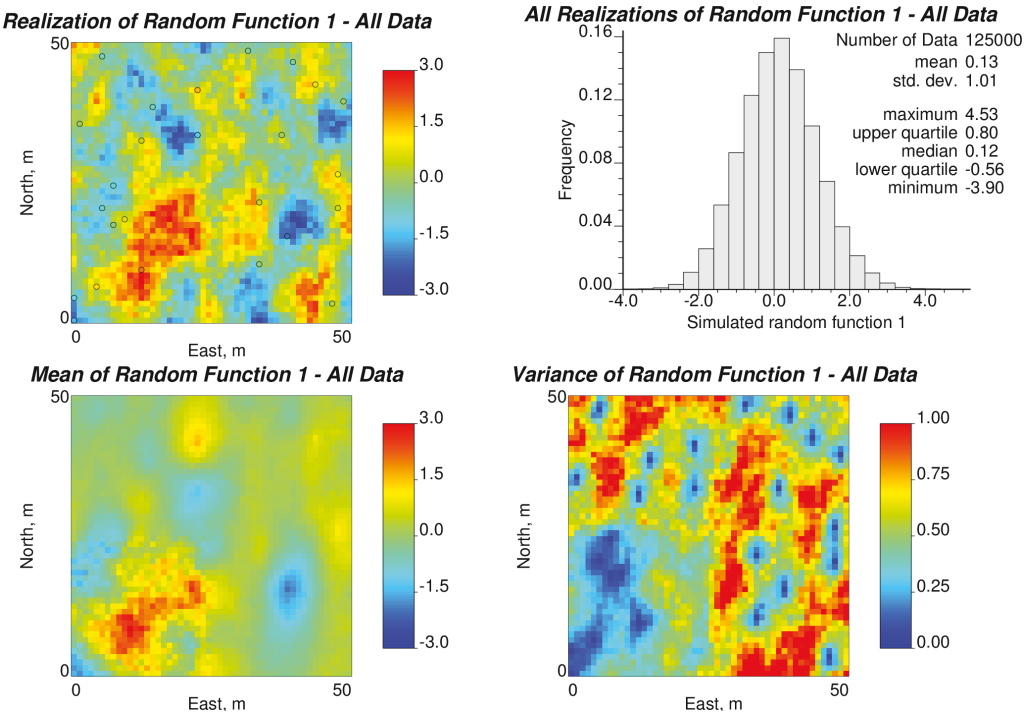


Figure 4.40: Example of the conditioning using the cokriging – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_1.

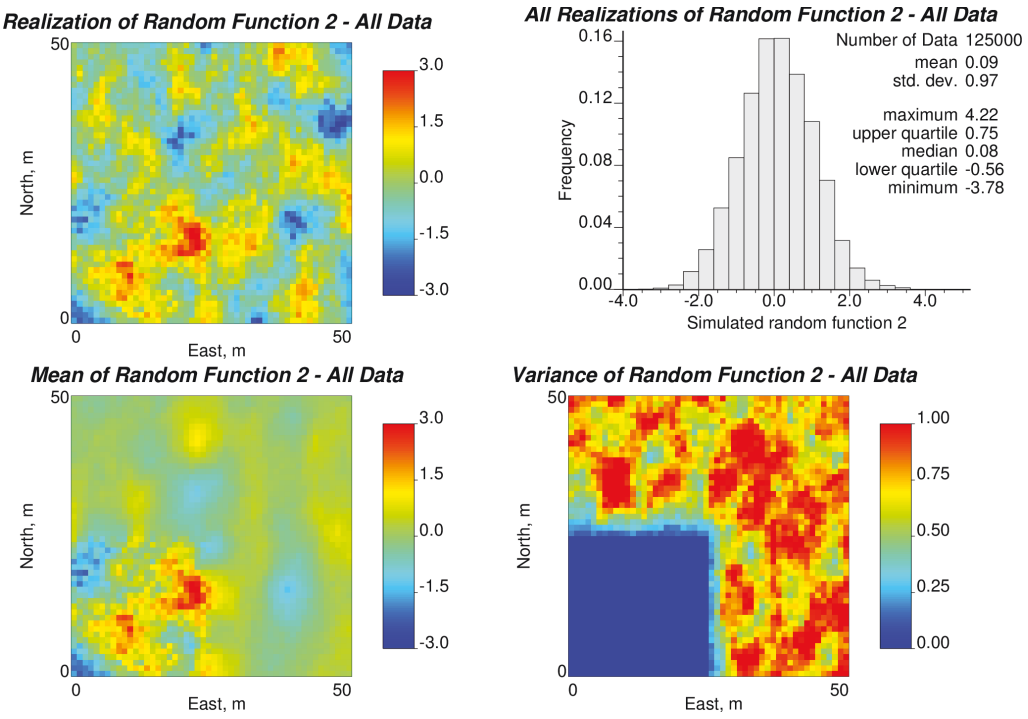


Figure 4.41: Example of the conditioning using the cokriging – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_2.

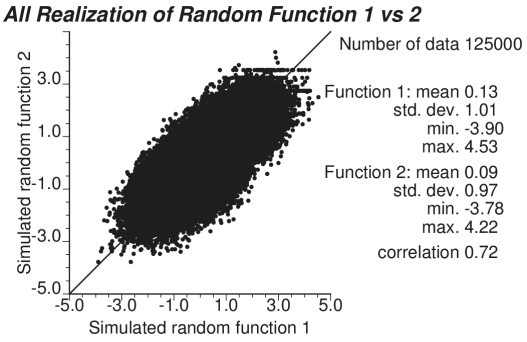


Figure 4.42: Example of the conditioning using the cokriging – scatter plot between all realizations of the simulated random functions Y_1 and Y_2.

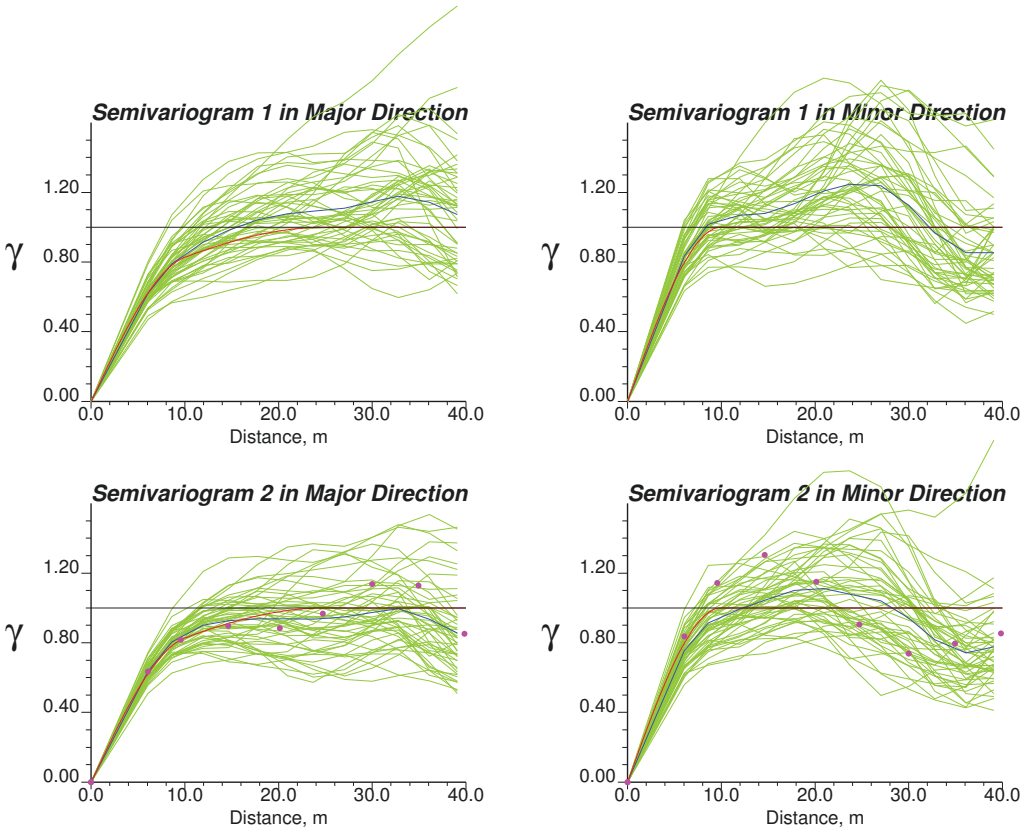


Figure 4.43: Example of the conditioning using the cokriging – experimental variograms in green computed from 50 realizations, their average in blue, and variogram model in red of the random functions Y_1 and Y_2 in major and minor directions of the continuity. Violet dots are experimental variograms of random function Y_2 computed from data.

The second case reveals the artifact that is caused by ignoring the projection approach in intrinsic cokriging conditioning. Fig. 4.44 and Fig. 4.45 show the first realization, a histogram of all realizations, their mean, and estimation error variance maps of both random functions Y_1 and Y_2. Even though the data are reproduced, the edge effect is prominent on the realization maps, especially for Y_2. Part of the simulation, which represents regions of the exhaustively sampled data, does not coincide with the rest of the simulation neither on the realization map nor on the map of mean of the realizations. In contrast, the simulated

statistics are very reasonable. Histograms of the realizations are quite normal. Mean and variance maps resemble theoretically expected ones. Estimation error variance is zero at the data locations. The scatter plot between simulated realizations is depicted in Fig. 4.46. The correlation coefficient of the simulations is 0.71, which is very close to the target one. The bivariate distribution appears to be bi-Gaussian. Fig. 4.47 shows good variogram reproduction.

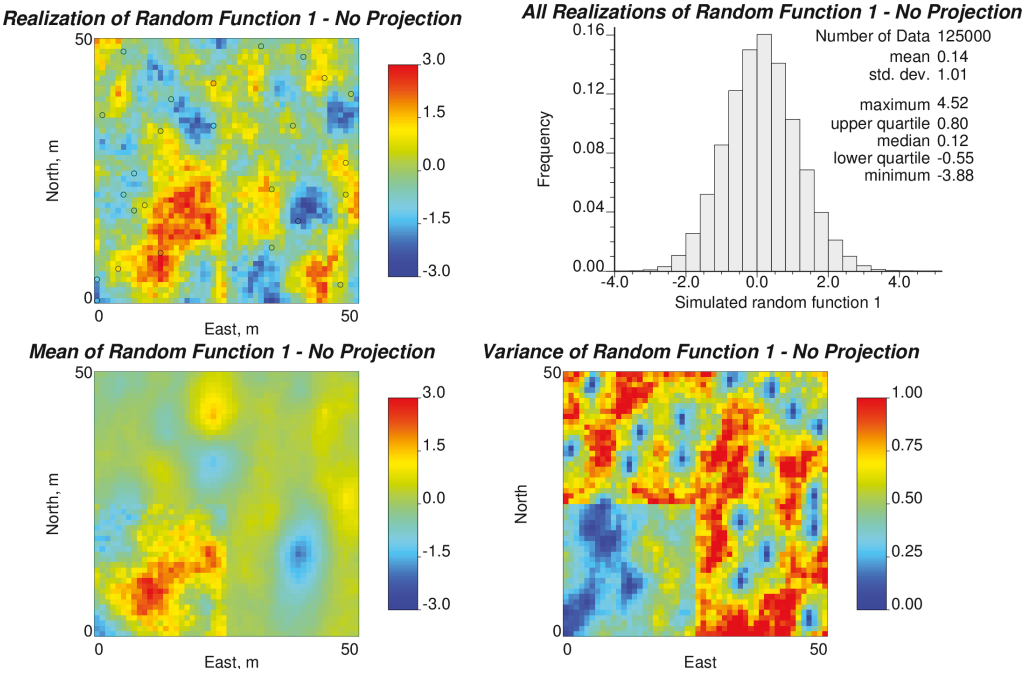


Figure 4.44: Example of the conditioning using the intrinsic cokriging without projection – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_1.

The last case consists of the same setting as before with the difference that the projection of the scattered data and simulation location onto the exhaustively sampled data region is deployed for the ICK-based conditioning. Fig. 4.48 and Fig. 4.49 show the resulting realizations of each random function Y_1 and Y_2. A map of first realization, histogram of all realizations, mean and estimation error variance maps are shown in these two figures. The edge effect is resolved on

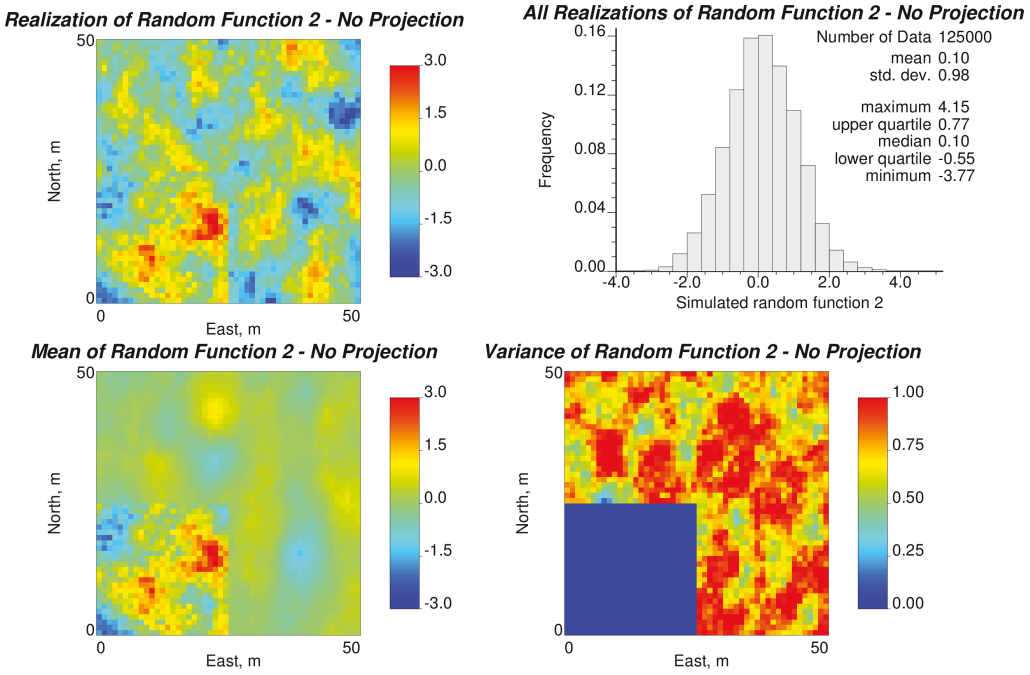


Figure 4.45: Example of the conditioning using the intrinsic cokriging without projection – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_2.

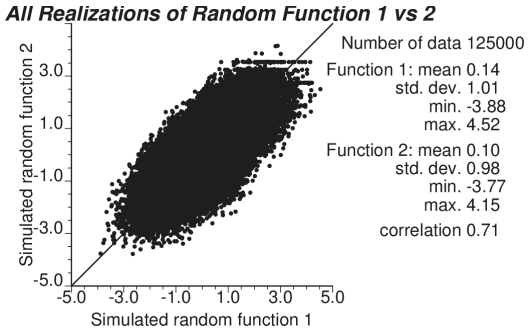


Figure 4.46: Example of the conditioning with using intrinsic cokriging without projection – scatter plot between all realizations of the simulated random functions Y_1 and Y_2.

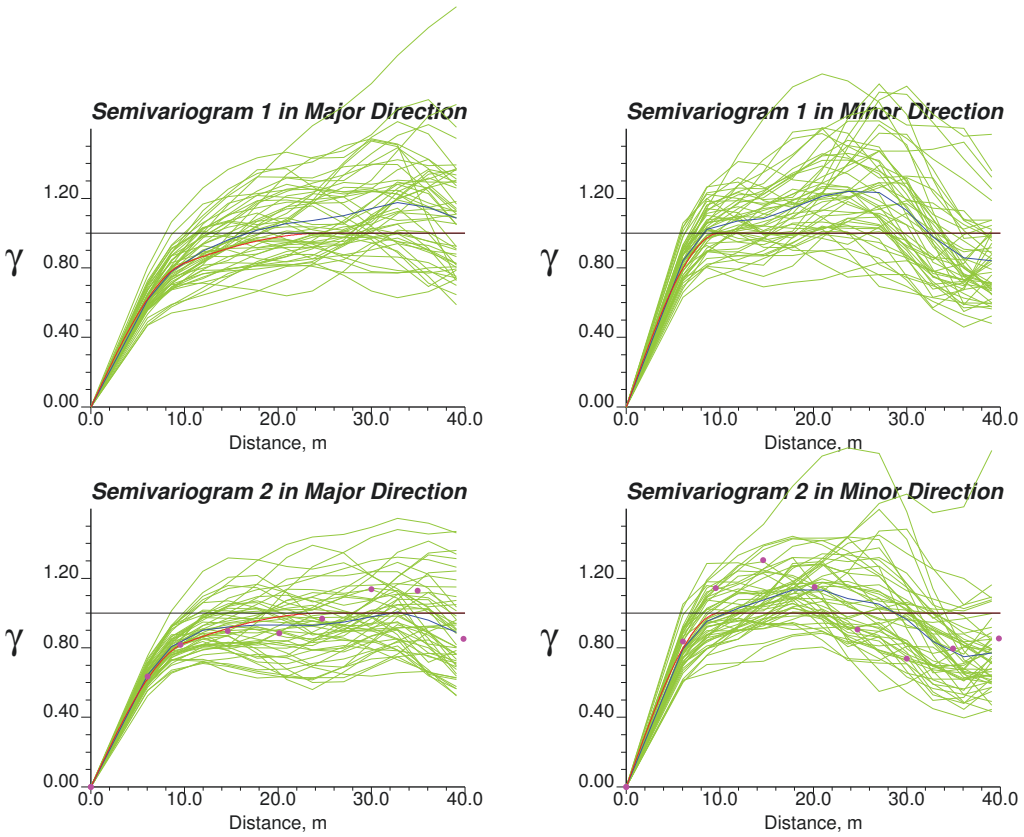


Figure 4.47: Example of the conditioning using the intrinsic cokriging without projection – experimental variograms in green computed from 50 realizations, their average in blue, and variogram model in red of the random functions Y_1 and Y_2 in major and minor directions of the continuity. Violet dots are experimental variograms of random function Y_2 computed from data.

the realization maps. A smooth transition from the exhaustively sampled data region to the rest of the simulation domain is observed on both the mean and variance maps. It is found that the transition zone is seen by some radial effect on the mean maps. The simulation results of the intrinsic cokriging approach with projection are very similar to the results of the conventional cokriging-based conditioning. Minor differences are justified by the different number of data used in

the conditioning. Both realizations of random functions Y_1 and Y_2 are individually normal. The bivariate relationship in Fig. 4.50 is binormal. Data values and correlation coefficient are honored. Fig. 4.51 contains experimental variograms computed from the realizations along with the variogram model in directions of maximum and minimum continuity. The variogram model is reproduced quite well for shorter lag distances. Some deviations of the experimental variograms from the model are observed for longer lag distances, and are deemed to be due to spatial structure inherited in the data.

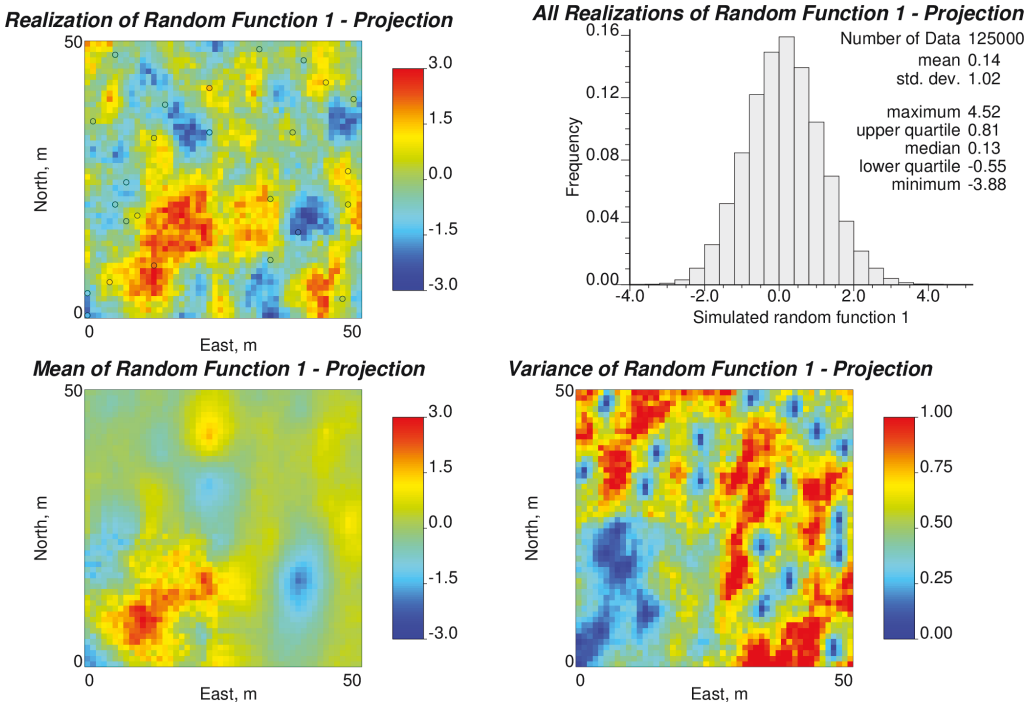


Figure 4.48: Example of the conditioning using the intrinsic cokriging with projection – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_1.

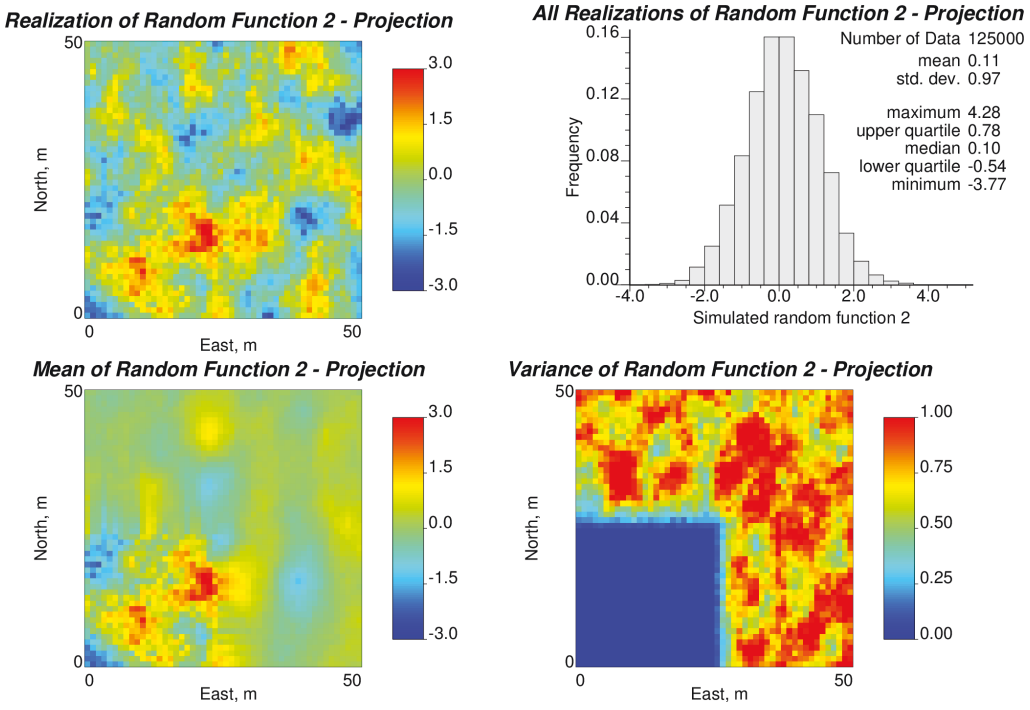


Figure 4.49: Example of the conditioning using the intrinsic cokriging with projection – map of first realization, histogram of all realizations, mean map of all realizations, and estimation error variance map of random function Y_2.

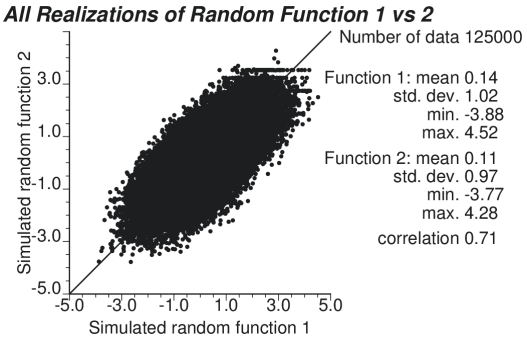


Figure 4.50: Example of the conditioning using the intrinsic cokriging with projection – scatter plot between all realizations of the simulated random functions Y_1 and Y_2.

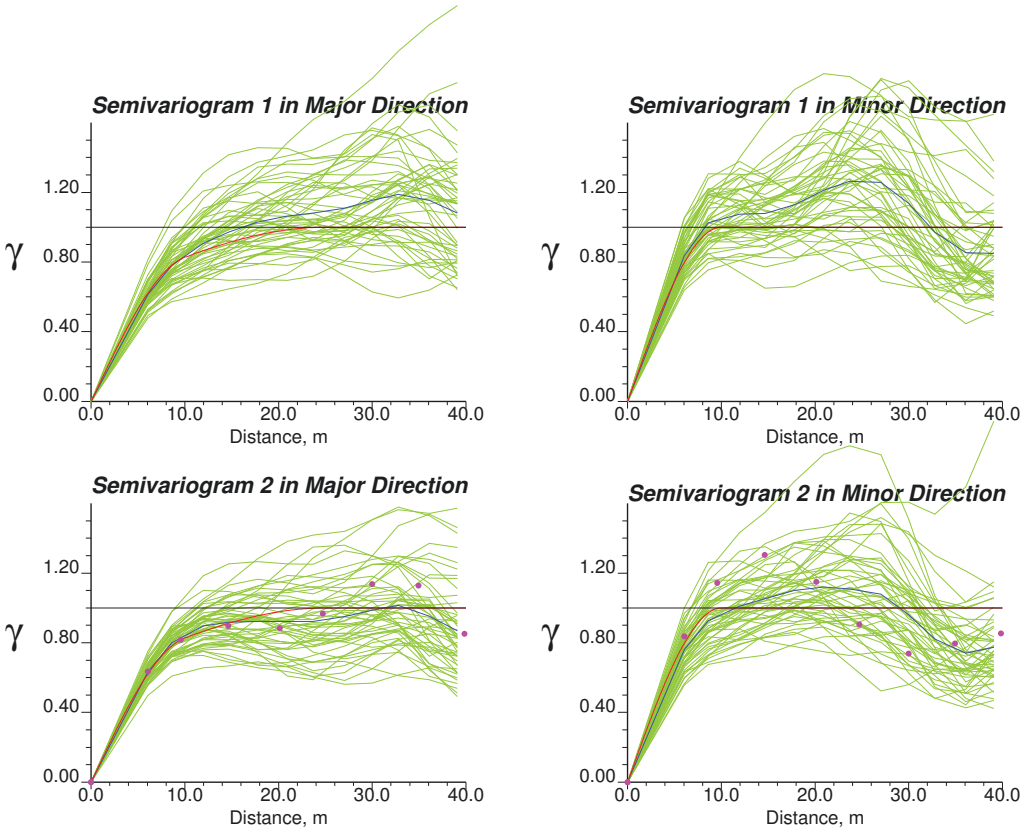


Figure 4.51: Example of the conditioning using the intrinsic cokriging with projection – experimental variograms in green computed from 50 realizations, their average in blue, and variogram model in red of the random functions Y_1 and Y_2 in major and minor directions of the continuity. Violet dots are experimental variograms of random function Y_2 computed from data.

4.2.4 Point-Scale Block Value Representation

Most of the time, the exhaustively sampled data are measured at a scale larger than a point scale. Seismic data is a good example of this. A methodology is explained next to bring block-scale data to a pseudo point scale in a grid free manner to avoid block-scale conditioning artifacts caused by abrupt changes across the adjacent block-scale data values. This block data representation with pseudo point-scale values is an alternative way to block cokriging for assimilation

block-scale data artifact-free [69]. There is no need to compute point scale-block scale covariances between data sampled at different scales. This reduces computational time for conditioning realizations to block-scale data. Thus, the advantage of the block-scale data representation with pseudo point-scale values is simplicity and complete consistency within the proposed grid free framework.

The exhaustively sampled data at a block scale are usually gridded. The grid consists of a set of data volumes in 3-D (or areas in 2-D, and intervals in 1-D). A single block-scale data value is assigned to the entire data volume. This block-scale data value represents an average of the point-scale data values contained within the volume. The average is computed arithmetically, if the studied geological property scales linearly. The location of any data volume is tied to the centre of the grid block. The schematic of 2-D gridded exhaustively sampled block-scale data is presented in Fig. 4.52. Black dots represent the centres of grid blocks or nodes of the block-scale data characterized by location indices i, j, and k, which define location order of each block in the data grid. Dashed lines define boundary between data volumes. The transition between any two adjacent volumes is not necessarily smooth, which creates artifacts in the models conditioned to this type of data, unless a conventional block cokriging approach is applied [36].

The grid-free simulation algorithm requires knowledge of the exhaustively sampled data at every location in the data region at a point scale. Therefore, the block-scale data should be transformed to a point scale for scale consistency. A point-scale block (PSB) value interpolation is proposed between centres of the block data to smooth transition between the adjacent block-scale data by representing block-scale data in a grid-free form and approximate the block-scale data to a pseudo point-scale data. This approach is not a downscaling. The smoothing algorithm does not bring any additional information, however it has a unique solution. These PSB values will be defined analytically as a function of

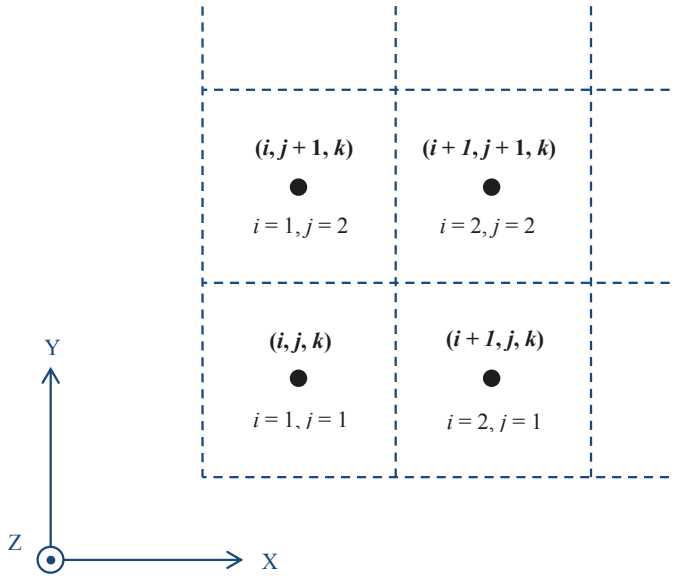


Figure 4.52: A schematic of a portion of the gridded exhaustively sampled block-scale data in plan view.

the coordinates of the location $\boldsymbol{u}$ in the space. The PSB representation of the secondary data produces more extreme values than original data. The average of the point-scale block values within the block volume should be equal to the block-scale data value. The global mean of the PSB values should match the mean of the data. The global variance of the PSB values may be larger than the variance of the block data, partly due to the introduction of more fluctuations by block-scale values regridding. The PSB should be applied before any normal score transformation.

The PSB value representation procedure is as follows. The block-scale data values $Z(V(\boldsymbol{u}_\alpha)), \alpha = 1, ..., N$, in original units are assigned to the centers of corresponding grid blocks and are called data nodal values. These data nodal values form another mesh shown as solid lines in Fig. 4.53. The dashed lines form grid blocks of original data. Then, these data nodal values $Z(V(x_i, y_j, z_k))$ presented by black dots are used as anchor points in the linear interpolation

model to obtain smooth pseudo point-scale block data values $Z(x, y, z)$, which are expressed in a grid-free manner as a function of the coordinates x, y, and z. The linear algorithm is summarized in Eqs. (4.49) – (4.55) through seven steps for 3-D data systems. Eq. (4.56) combines all previous seven equations to a single equation. Fig. 4.53 and Fig. 4.54 are provided for a visual explanation. Each equation depicts one calculation step. In the first step, a PSB value is calculated as in Eq. (4.49) between any two data nodal values at required coordinate on axis, which defines direction between two nodal values. In the second step, another PSB value is calculated between other two data nodal values, which are parallel to the first two data nodal values, as presented in Eq. (4.50). In the third step, new PSB value is interpolated between two previously computed PSB values. Similar steps are carried out for the data nodal values located in adjacent parallel plane. In the last seventh step, final PSB value is interpolated between PSB values computed at steps 3 and 6 to get pseudo point-scale block value at the required location $\boldsymbol{u}$ with coordinates x, y, and z. The order of the calculations shown in the equations and figures has been chosen arbitrarily. The indices $i = 0, ..., N_X + 1$, $j = 0, ..., N_Y + 1$, and $k = 0, ..., N_Z + 1$, imply the location of the data nodal values in the grid. The integers N_X, N_Y, and N_Z are the number of the blocks in three principal directions of the gridded data. The edge data nodes, which are shown as hollow dots in Fig. 4.53, are assumed to be equal to the neighbouring data node value.

$$\begin{aligned} &Z(x \in \{x_i : x_{i+1}\}, y_j, z_k) = Z(V(x_i, y_j, z_k)) \\ &+ (Z(V(x_{i+1}, y_j, z_k)) - Z(V(x_i, y_j, z_k))) \left(\frac{x - x_i}{x_{i+1} - x_i} \right) \end{aligned} \tag{4.49}$$

$$
\begin{aligned}
&Z(x \in \{x_i : x_{i+1}\}, y_{j+1}, z_k) = Z(V(x_i, y_{j+1}, z_k)) \\
&+ (Z(V(x_{i+1}, y_{j+1}, z_k)) - Z(V(x_i, y_{j+1}, z_k))) \left(\frac{x - x_i}{x_{i+1} - x_i} \right)
\end{aligned} \tag{4.50}
$$

$$
\begin{aligned}
&Z(x \in \{x_i : x_{i+1}\}, y \in \{y_j : y_{j+1}\}, z_k) = Z(x \in \{x_i : x_{i+1}\}, y_j, z_k) \\
&+ (Z(x \in \{x_i : x_{i+1}\}, y_{j+1}, z_k) - Z(x \in \{x_i : x_{i+1}\}, y_j, z_k)) \\
&\cdot \left(\frac{y - y_j}{y_{j+1} - y_j} \right)
\end{aligned} \tag{4.51}
$$

$$
\begin{aligned}
&Z(x \in \{x_i : x_{i+1}\}, y_j, z_{k+1}) = Z(V(x_i, y_j, z_{k+1})) \\
&+ (Z(V(x_{i+1}, y_j, z_{k+1})) - Z(V(x_i, y_j, z_{k+1}))) \left(\frac{x - x_i}{x_{i+1} - x_i} \right)
\end{aligned} \tag{4.52}
$$

$$
\begin{aligned}
&Z(x \in \{x_i : x_{i+1}\}, y_{j+1}, z_{k+1}) = Z(V(x_i, y_{j+1}, z_{k+1})) \\
&+ (Z(V(x_{i+1}, y_{j+1}, z_{k+1})) - Z(V(x_i, y_{j+1}, z_{k+1}))) \left(\frac{x - x_i}{x_{i+1} - x_i} \right)
\end{aligned} \tag{4.53}
$$

$$
\begin{aligned}
&Z(x \in \{x_i : x_{i+1}\}, y \in \{y_j : y_{j+1}\}, z_{k+1}) = Z(x \in \{x_i : x_{i+1}\}, y_j, z_{k+1}) \\
&+ (Z(x \in \{x_i : x_{i+1}\}, y_{j+1}, z_{k+1}) - Z(x \in \{x_i : x_{i+1}\}, y_j, z_{k+1})) \\
&\cdot \left(\frac{y - y_j}{y_{j+1} - y_j} \right)
\end{aligned} \tag{4.54}
$$

$$
\begin{aligned}
&Z(x \in \{x_i : x_{i+1}\}, y \in \{y_j : y_{j+1}\}, z \in \{z_k, z_{k+1}\}) \\
&= Z(x \in \{x_i : x_{i+1}\}, y \in \{y_j : y_{j+1}\}, z_k) \\
&+ (Z(x \in \{x_i : x_{i+1}\}, y \in \{y_j : y_{j+1}\}, z_{k+1}) \\
&- Z(x \in \{x_i : x_{i+1}\}, y \in \{y_j : y_{j+1}\}, z_k)) \cdot \left(\frac{z - z_k}{z_{k+1} - z_k} \right)
\end{aligned}
\tag{4.55}
$$

$$
\begin{aligned}
&Z(x \in \{x_i : x_{i+1}\}, y \in \{y_j : y_{j+1}\}, z \in \{z_k, z_{k+1}\}) = Z(V(x_i, y_j, z_k)) \\
&+ (Z(V(x_{i+1}, y_j, z_k)) - Z(V(x_i, y_j, z_k))) \left(\frac{x - x_i}{x_{i+1} - x_i} \right) \\
&+ (Z(V(x_i, y_{j+1}, z_k)) - Z(V(x_i, y_j, z_k))) \left(\frac{y - y_j}{y_{j+1} - y_j} \right) \\
&+ (Z(V(x_i, y_j, z_{k+1})) - Z(V(x_i, y_j, z_k))) \left(\frac{z - z_k}{z_{k+1} - z_k} \right) \\
&+ (Z(V(x_i, y_j, z_k)) - Z(V(x_{i+1}, y_j, z_k)) \\
&+ Z(V(x_{i+1}, y_{j+1}, z_k)) - Z(V(x_i, y_{j+1}, z_k))) \left(\frac{x - x_i}{x_{i+1} - x_i} \frac{y - y_j}{y_{j+1} - y_j} \right) \\
&+ (Z(V(x_i, y_j, z_k)) - Z(V(x_{i+1}, y_j, z_k)) \\
&- Z(V(x_i, y_j, z_{k+1})) + Z(V(x_{i+1}, y_j, z_{k+1}))) \left(\frac{x - x_i}{x_{i+1} - x_i} \frac{z - z_k}{z_{k+1} - z_k} \right) \\
&+ (Z(V(x_i, y_j, z_k)) - Z(V(x_i, y_{j+1}, z_k)) \\
&- Z(V(x_i, y_j, z_{k+1})) + Z(V(x_i, y_{j+1}, z_{k+1}))) \left(\frac{y - y_j}{y_{j+1} - y_j} \frac{z - z_k}{z_{k+1} - z_k} \right) \\
&+ (-Z(V(x_i, y_j, z_k)) + Z(V(x_{i+1}, y_j, z_k)) \\
&- Z(V(x_{i+1}, y_{j+1}, z_k)) + Z(V(x_i, y_{j+1}, z_k)) \\
&+ Z(V(x_i, y_j, z_{k+1})) - Z(V(x_{i+1}, y_j, z_{k+1})) \\
&+ Z(V(x_{i+1}, y_{j+1}, z_{k+1})) - Z(V(x_i, y_{j+1}, z_{k+1}))) \\
&\cdot \left(\frac{x - x_i}{x_{i+1} - x_i} \frac{y - y_j}{y_{j+1} - y_j} \frac{z - z_k}{z_{k+1} - z_k} \right)
\end{aligned}
\tag{4.56}
$$

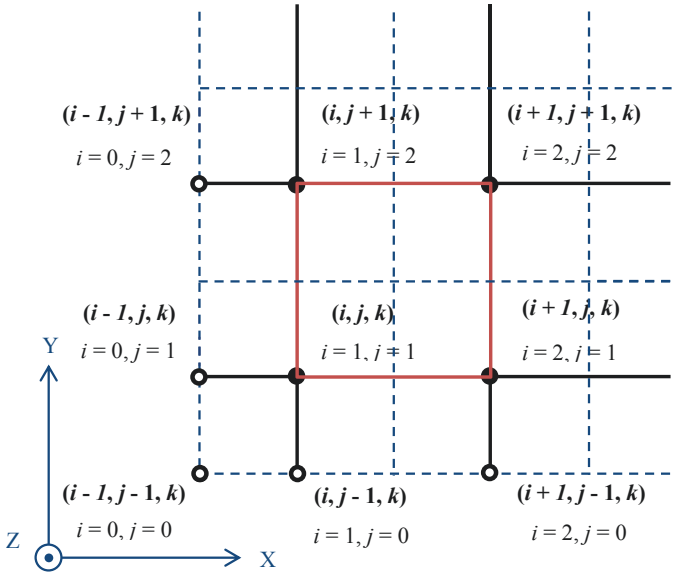


Figure 4.53: Point-scale block data representation paradigm – mesh in 2-D.

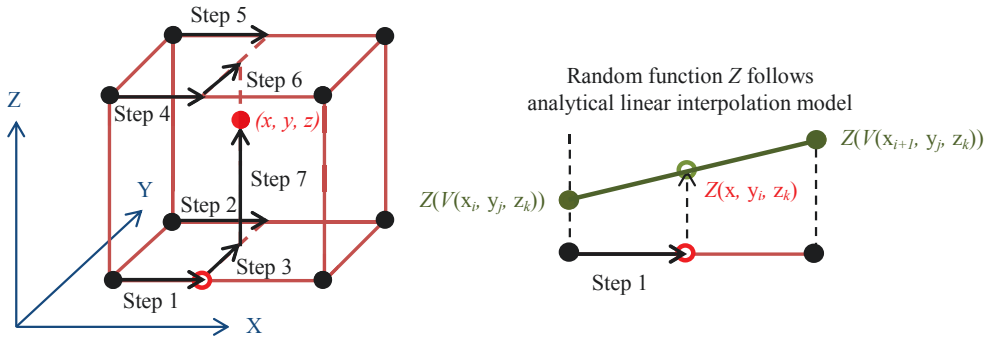


Figure 4.54: Point-scale block data representation paradigm – algorithmic steps in 3-D.

The major criteria for the interpolation is that local means of the point-scale block values within each block should match block-scale data value for each block. It can be done by discretizing block-scale data into finite number of the point-scale block values. The average of the point-scale block values is computed and compared with the original block-scale data value. The initial setting, when

data nodal values equal to the block-scale data, does not ensure reproduction of the block-scale data by local PSB averages. An iterative procedure is implemented to match local averages with corresponding original block-scale data by changing data nodal values in the interpolation equations. The data nodal values are adjusted through the Eqs. (4.57) – (4.60) by matching the averages of the PSB values with the input block-scale data. Integers ΔN_X, ΔN_Y, and ΔN_Z are the number of the discretization PSB points in X, Y, and Z directions used to compute local averages of the PSB representation model values. First, the data nodal values are assumed to be equal to the original block-scale data as shown in Eq. (4.57). Then, nodal PSB values are adjusted through coefficient f_s according to Eq. (4.58). The coefficient in Eq. (4.59) is just a ratio of the block datum to the corresponding local PSB averages computed through Eq. (4.60). The mismatch between model and actual data is presented by mean square error ϵ in Eq. (4.61). The procedure is repeated for several other iterations. The iteration number s that gives the minimal discrepancy is chosen for the final model of grid-free PSB value representation model. The iterations can be terminated after particular number or when ϵ reaches some minimal threshold value. In this way, the original block-scale data are transformed to point-scale block values with no systematic bias.

$$Z_{s=0}(x_i, y_j, z_k) = Z(V(x_i, y_j, z_k)) \tag{4.57}$$

$$Z_{s+1}(x_i, y_j, z_k) = Z_s(x_i, y_j, z_k) f_s(x_i, y_j, z_k) \tag{4.58}$$

$$f_s(x_i, y_j, z_k) = \frac{Z(V(x_i, y_j, z_k))}{\bar{Z}_s \begin{pmatrix} x \in \{x_{i-1/2} : x_{i+1/2}\}, \\ y \in \{y_{j-1/2} : y_{j+1/2}\}, \\ z \in \{z_{k-1/2} : z_{k+1/2}\} \end{pmatrix}} \tag{4.59}$$

$$\bar{Z}_s\begin{pmatrix} x\in\{x_{i-1/2}:x_{i+1/2}\},\\ y\in\{y_{j-1/2}:y_{j+1/2}\},\\ z\in\{z_{k-1/2}:z_{k+1/2}\}\end{pmatrix} = \frac{\sum\limits_{i'=1}^{\Delta N_X}\sum\limits_{j'=1}^{\Delta N_Y}\sum\limits_{k'=1}^{\Delta N_Z} Z_s\begin{pmatrix} x_{i'}\in\{x_{i-1/2}:x_{i+1/2}\},\\ y_{j'}\in\{y_{j-1/2}:y_{j+1/2}\},\\ z_{k'}\in\{z_{k-1/2}:z_{k+1/2}\}\end{pmatrix}}{\Delta N_X \Delta N_Y \Delta N_Z} \tag{4.60}$$

$$\epsilon_s = \frac{\sum\limits_{i=0}^{N_X+1}\sum\limits_{j=0}^{N_Y+1}\sum\limits_{k=0}^{N_Z+1}\left(\bar{Z}_s\begin{pmatrix} x\in\{x_{i-1/2}:x_{i+1/2}\},\\ y\in\{y_{j-1/2}:y_{j+1/2}\},\\ z\in\{z_{k-1/2}:z_{k+1/2}\}\end{pmatrix} - Z(V(x_i,y_j,z_k))\right)^2}{(N_X+2)(N_Y+2)(N_Z+2)} \tag{4.61}$$

Four examples are prepared to demonstrate implementation details and characteristics of the point-scale block value representation method for representing block-scale data as pseudo point-scale block values. The interpolated values are still at the block scale, but infinitely resolved.

The first example motivates need for the PSB value representation of the secondary exhaustively sampled data. Fig. 4.55 presents realizations of the bivariate system conditional to secondary exhaustively sampled data of random variable 2. When no PSB representation of the secondary block data is assumed (plots on the left-hand side), the resulting realization of the random variable 1 has artifacts related to the sharp boundaries between adjacent secondary data. The realization of the random variable 2 precisely depicts the conditioning data. Only values of secondary data collocated with simulation locations are used. These secondary data values are assumed constant throughout entire block region. The artifacts are removed (plots on the right-hand side), when the PSB value representation of the secondary data is applied. The realization of random variable 1 is similar to the previous realization, but without boundary affect. Secondary data dictate high and low regions of the simulated random variable 1. The realization of random variable 2 is the PSB value-based interpolation of four block data, values of which are centered in the middle of the appropriate

blocks.

The second example consists of 10 block-scale data values distributed along line axis as shown in Fig. 4.56 in red. The PSB value representation is performed on this data set to get pseudo point-scale block data values. Each block datum is discretized into 10 point data. The PSB value-based interpolation results are presented in the same Fig. 4.56, where PSB values are shown by a green line and their local averages are presented by blue line. It is clear that the initial setting of the nodal PSB values produces an interpolated model that is far from the data. After 4 iterations, the model improves significantly. The mismatch between the data and local averages of the PSB values in terms of the mean square error is presented in Fig. 4.57 for each of 20 iterations. Based on this bar chart, it can be argued that in this 1-D case $4 - 8$ iterations are enough to match PSB value model with the data quite well. The statistics in the form of a histogram is presented in Fig. 4.58 for data, PSB values, and their averages. The local averages of PSB values coincide with the data statistics for higher iteration numbers. The match is not so good for the initial setting and the first iteration. The number of iterations depends on the complexity of the block-scale data. It is recommended to run at least several iterations to get acceptable data reproduction. The mean and standard deviation of the data and local averages of last iteration are the same. While the mean of PSB values of the last iteration is the same as the data mean, the standard deviation is higher (0.56 versus 0.53) because of some higher values in high blocks and lower values in some low blocks; there is no scale change, but the point representation still calls for some more extreme values.

The third example is a small 2-D case, which consists of 16 (4×4) block-scale data values. A map of the original data is shown in Fig. 4.59. There are sharp boundaries between adjacent block-scale data. The objective is to represent part of this data set at a point-scale in the area shown by the black

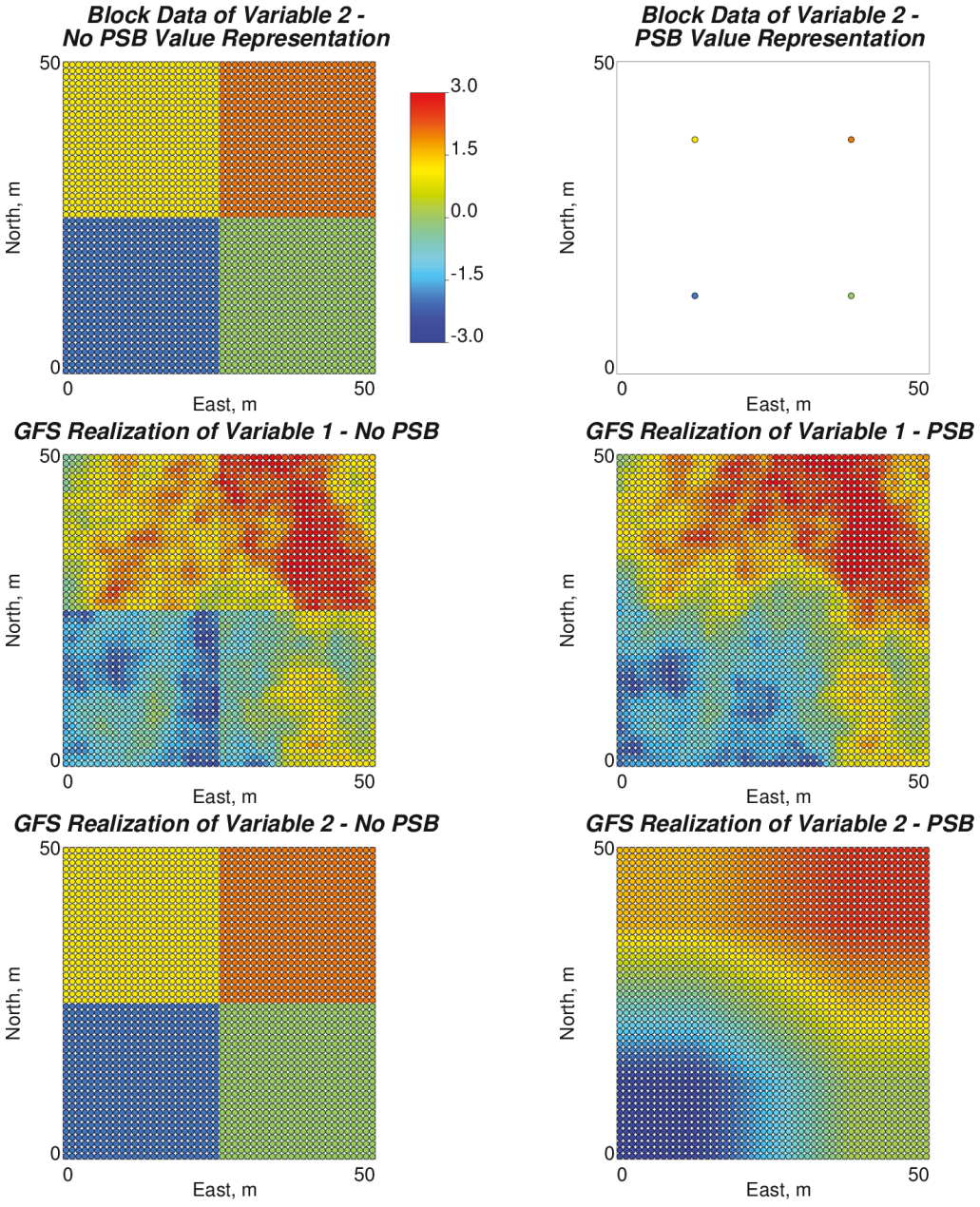


Figure 4.55: Example of 2-D point-scale block value representation – plots of conditioning secondary data and realizations of the bivariate system with and without PSB value representation of secondary data.

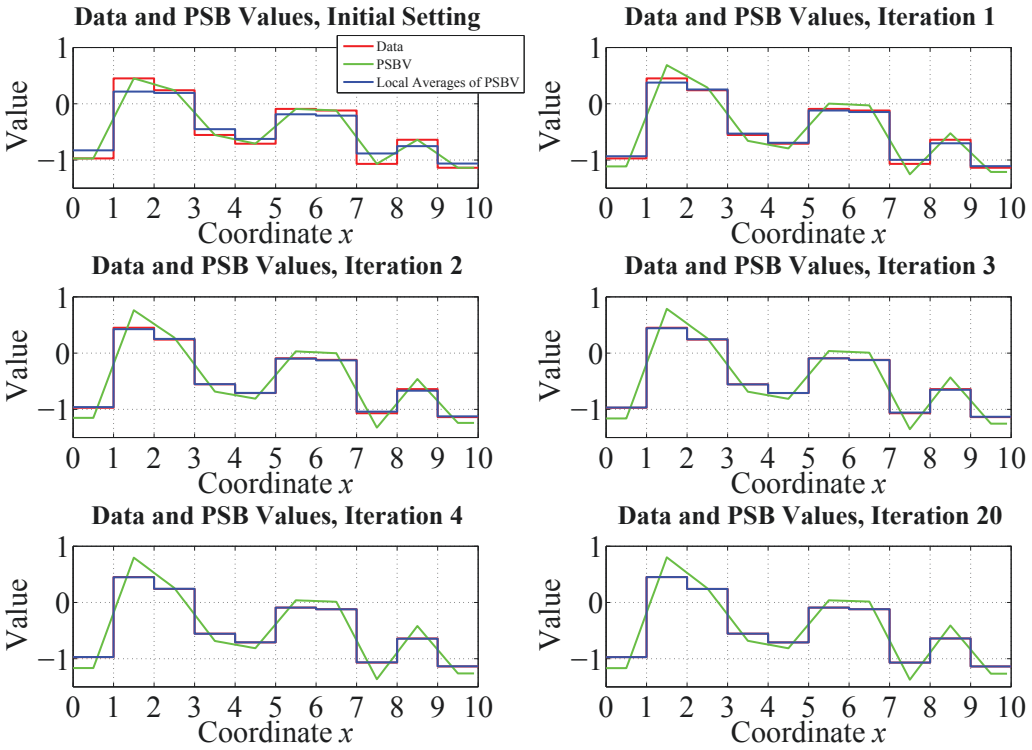


Figure 4.56: Example of 1-D point-scale block value representation – plots of original block-scale data, PSB values, and their local averages for several iterations.

box. The required number of point-scale block data is 800 (40×20). The results of the point-scale block value representation are shown in the same Fig. 4.59. It contains PSB values and their local averages for the initial setting and for iteration 30. It is clear that the block-scale data are not matched with initial PSB values, but are perfectly reproduced with the adjusted PSB values after 30 iterations. The mismatch between local averages of the tuned PSB values and data is presented as a bar chart of mean square error in Fig. 4.60 for each iteration. From this chart it can be suggested to run 10 – 15 iterations for this 2-D case to get a close match. Fig. 4.61 contains histograms of the data, PSB

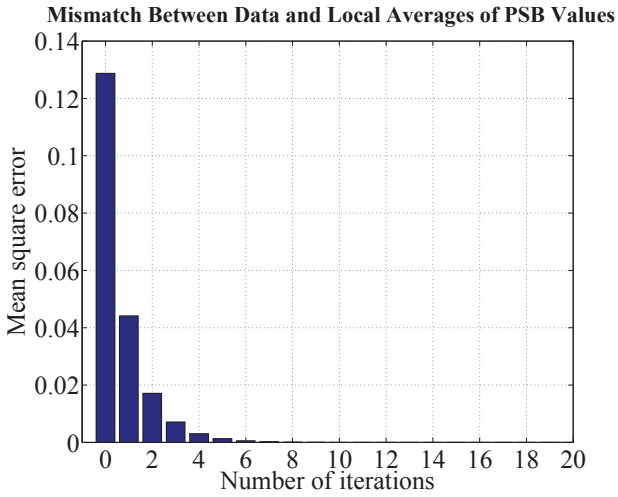


Figure 4.57: Example of 1-D point-scale block value representation – mismatch between original block-scale data and local averages of PSB values in terms of mean square error for each iteration.

values for initial setting and last iteration with corresponding local averages at block scale. It is obvious that neither global mean or variance of the PSB values match the data parameters for the initial setting. Similar observations can be said about the match between local averages of the PSB and original data for initial setting. On the other hand, global means of PSB values and their local averages match the data mean exactly for iteration 30. The global variance of the local averages are the same as of the data for iteration 30. The variance of PSB values is larger as expected.

The fourth example involves PSB representation of a large 3-D model. The sampling grid of the block-scale data consists of 62500 ($50 \times 50 \times 25$) blocks of $5.0 \times 5.0 \times 1.0$ m^3 size each. The data are generated synthetically with sequential Gaussian simulation according to the anisotropic spatial function presented in Eq. (4.62). The data are shown in Fig. 4.62 in 3-D. The bottom slice of the data model and histogram of the data are also presented in this figure. The distribution of the data is Gaussian with 0.02 mean and 0.88 variance.

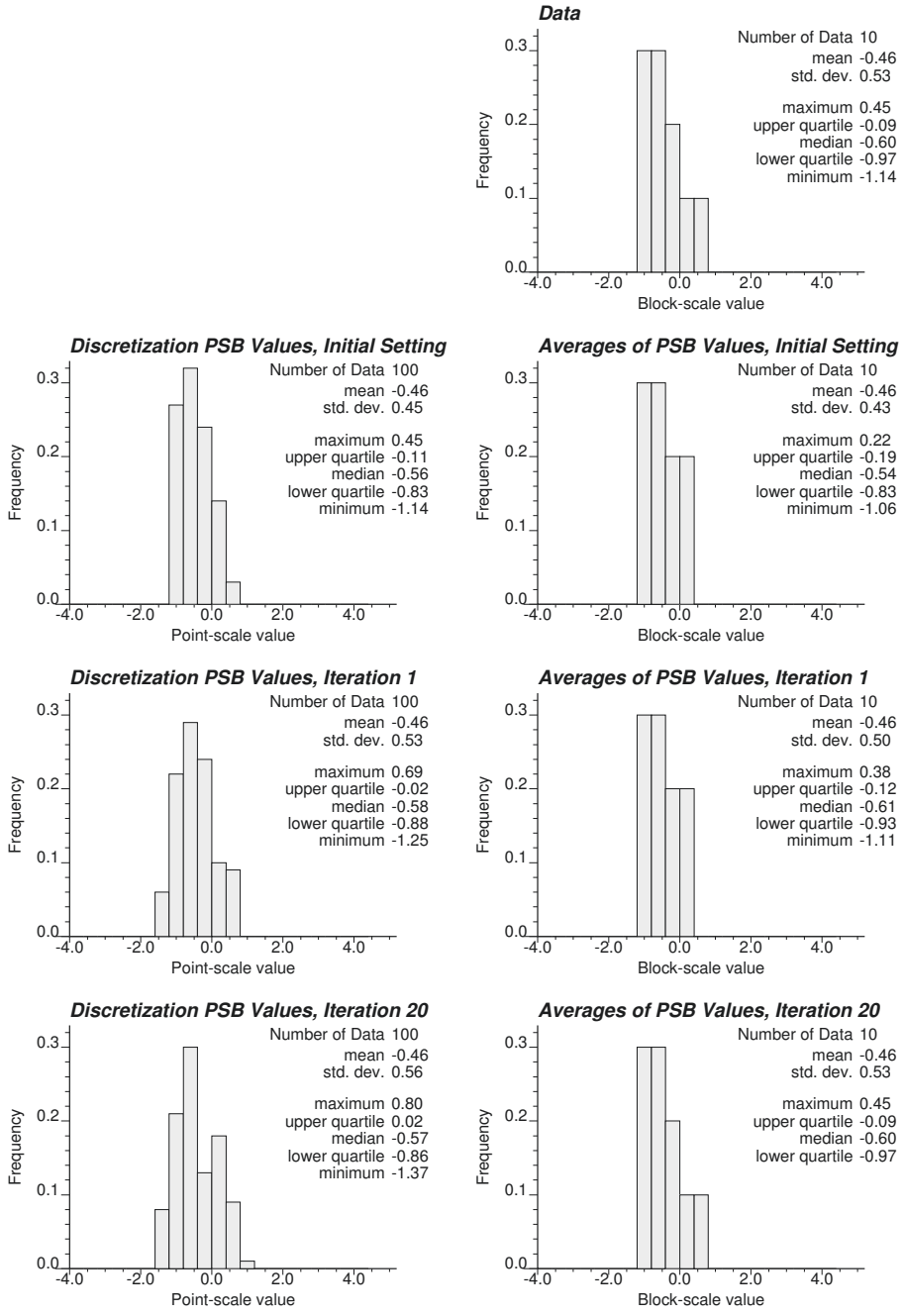


Figure 4.58: Example of 1-D point-scale block value representation – histograms of original block-scale data, PSB values, and their local averages for several iterations.

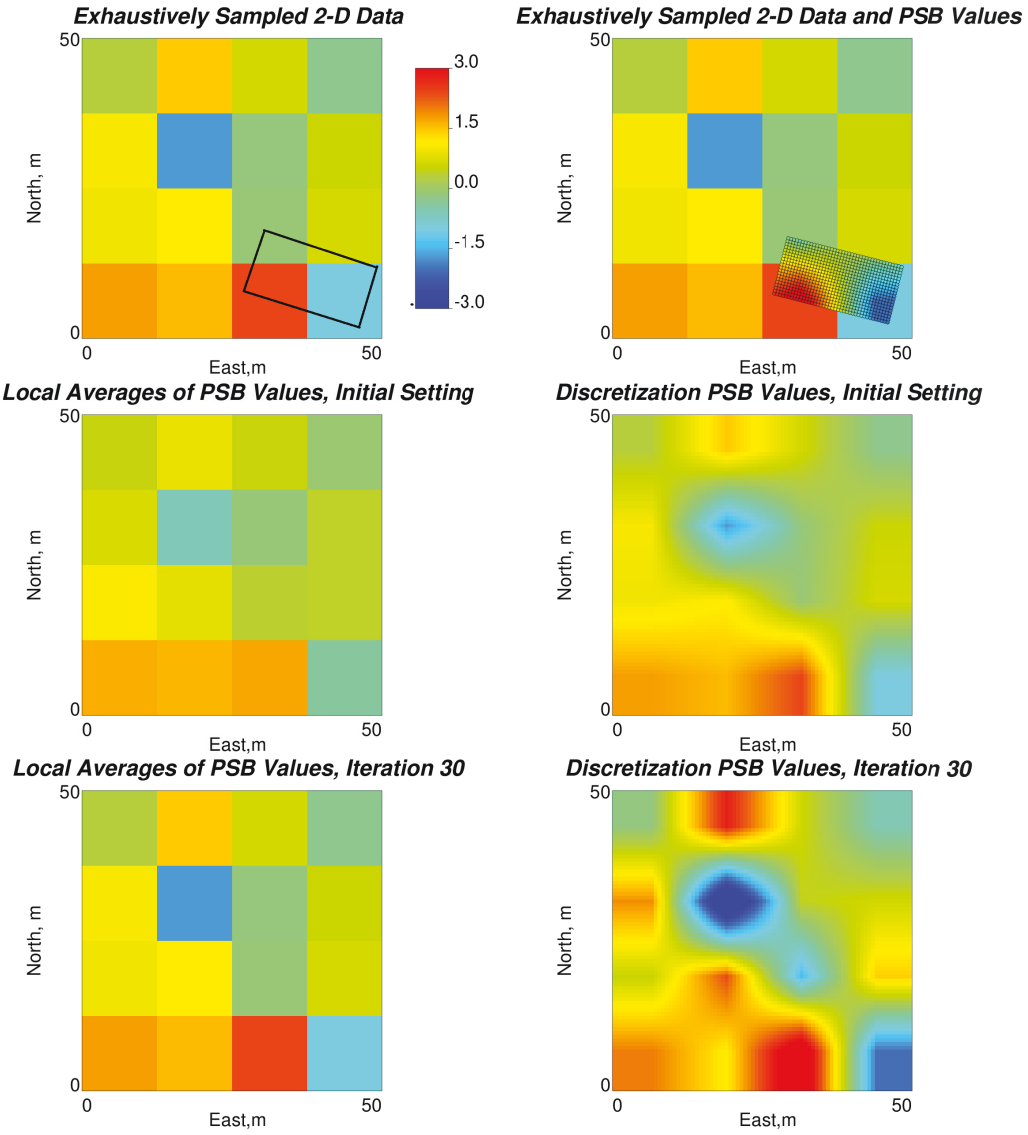


Figure 4.59: Example of 2-D point-scale block value representation – maps of original block-scale data, initial PSB values and after 30 iteration, and their local averages.

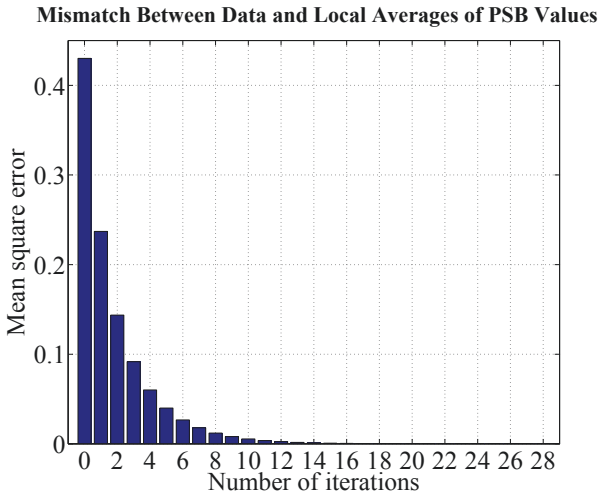


Figure 4.60: Example of 2-D point-scale block value representation – mismatch between original block-scale data and local averages of PSB values in terms of mean square error for each iteration.

Discretized PSB values and their local averages for the initial setting are shown in Fig. 4.63. The distribution of these values is Gaussian, but do not match the original data statistics precisely. Fig. 4.64 contains corrected PSB values after 30 iterations. The global mean and variance of local averages of PSB values match the data statistics exactly. The global variance of the PSB values is higher as expected. It is recommended to run more than 20 iterations for this 3-D case based on the bar chart of mean square error in Fig. 4.65. Fig. 4.66 shows the experimental variograms computed for the original data at block scale (dark blue dots), initial PSB values at point scale (light green dots), and corrected PSB values at block-scale (light blue dots) for iteration 30. The shape of all experimental variograms is similar. Variogram sills are different for all three cases as reflected in the summary statistics next to the histograms. The experimental variogram points of the corrected PSB values are higher than of the data, because of the slight increase in variance.

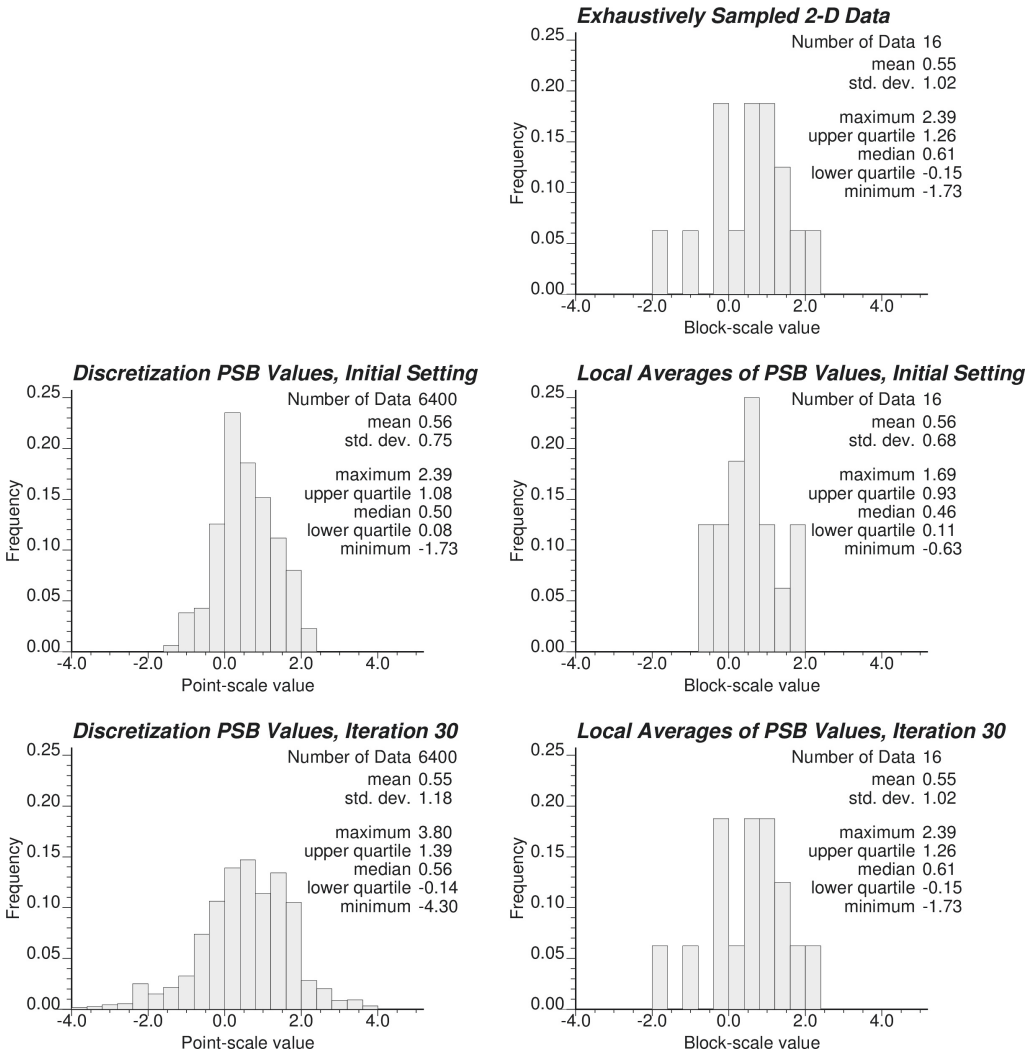
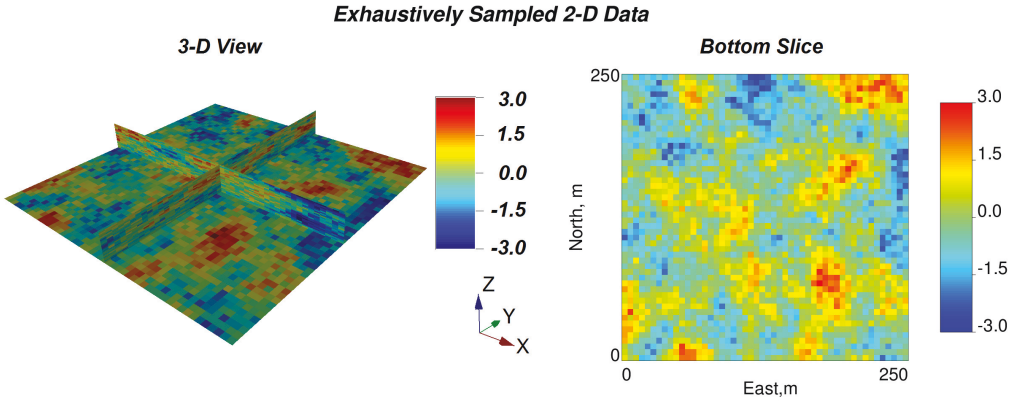
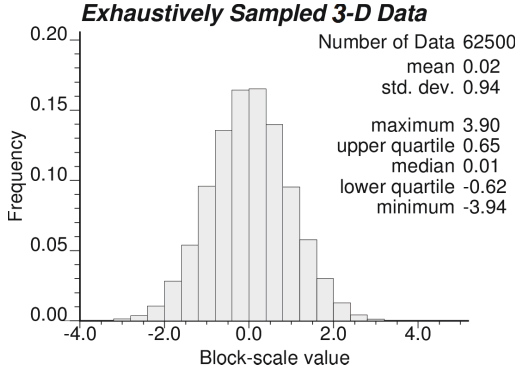


Figure 4.61: Example of 2-D point-scale block value representation – histograms of original block-scale data, initial PSB values and after 30 iteration, and their local averages.

$$C_{Y_E}(\boldsymbol{h}) = Sph_{\substack{r_1=50.0 \\ r_2=50.0 \\ r_3=10.0 \\ \alpha=0°}}(\boldsymbol{h}) \tag{4.62}$$



(a)



(b)

Figure 4.62: Example of 3-D point-scale block value representation – (a) maps of original gridded block-scale data, and (b) histogram of original data.

Note that data in all four cases are in original units. When real exhaustively sampled data are used, similar PSB value-based representation should be applied first in original units. Then, the PSB values are converted to the normal space and conditional simulation is performed. The corrected data nodal PSB values at the centers of the grid blocks can be used as reference points for the table-based normal score transformation (Chapter 5).

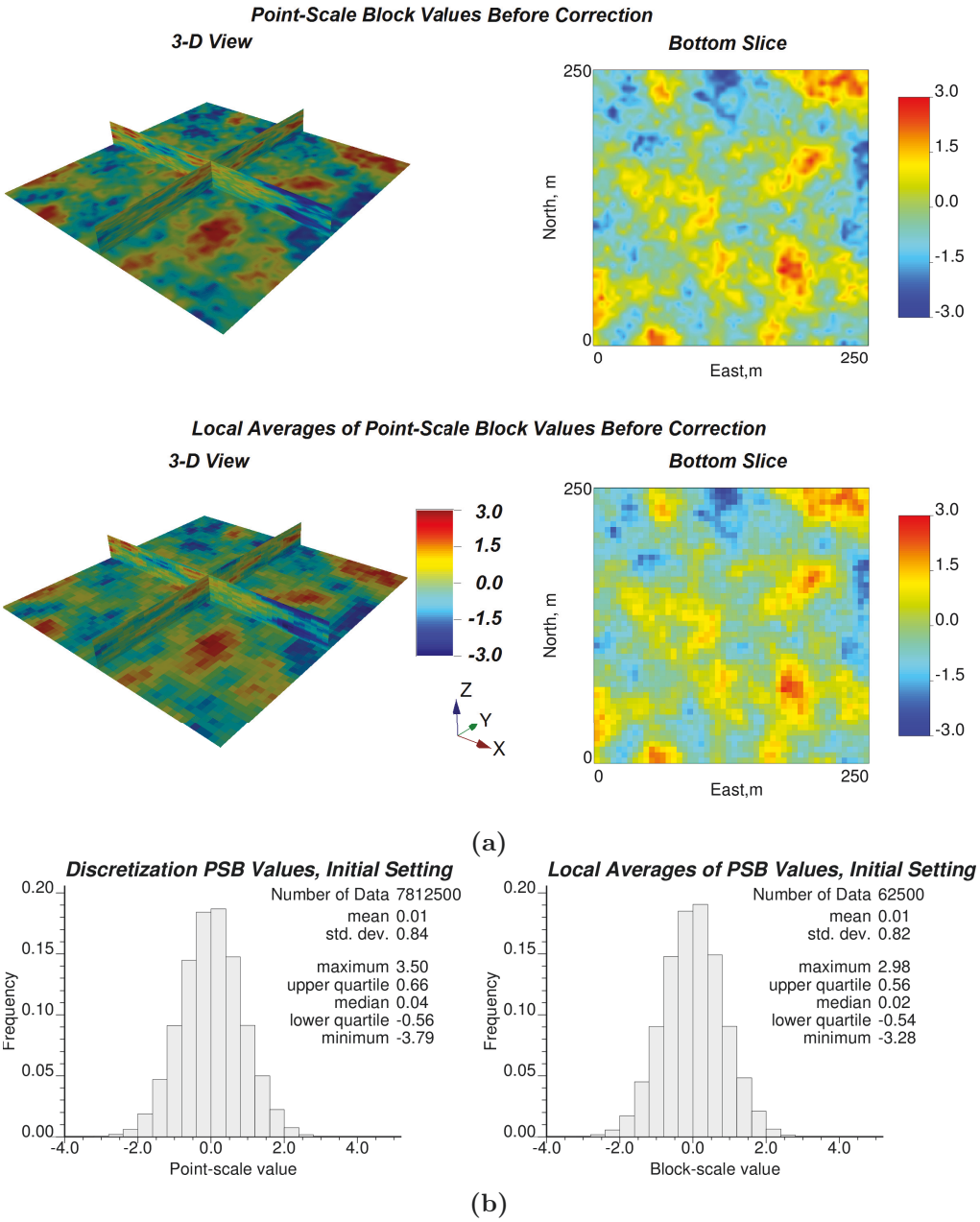


Figure 4.63: Example of 3-D point-scale block value representation – (a) maps of discretization point-scale block values and their local averages before nodal data correction, and (b) corresponding histograms.

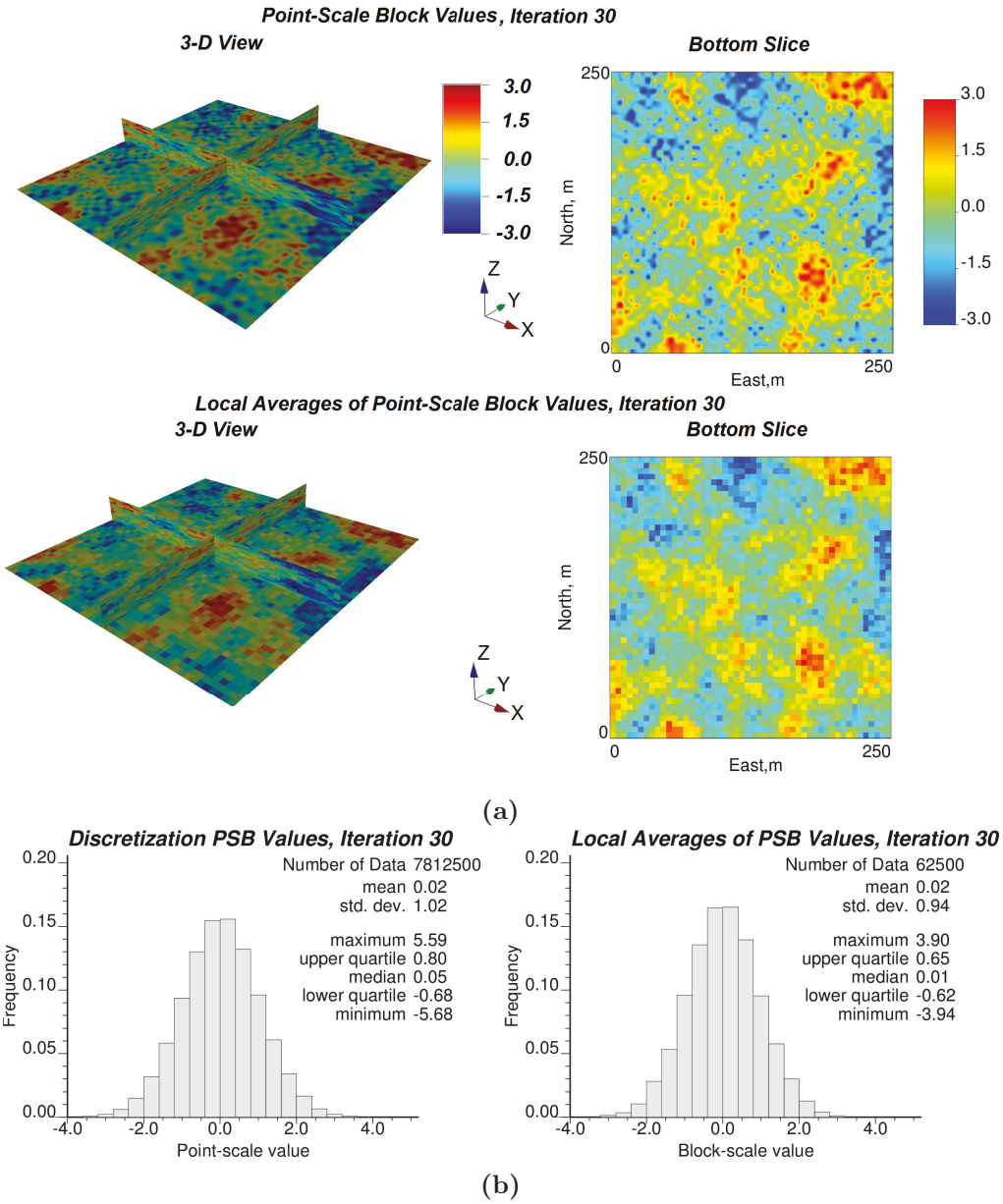


Figure 4.64: Example of 3-D point-scale block value representation – (a) maps of discretization point-scale block values and their local averages after nodal data correction, and (b) corresponding histograms.

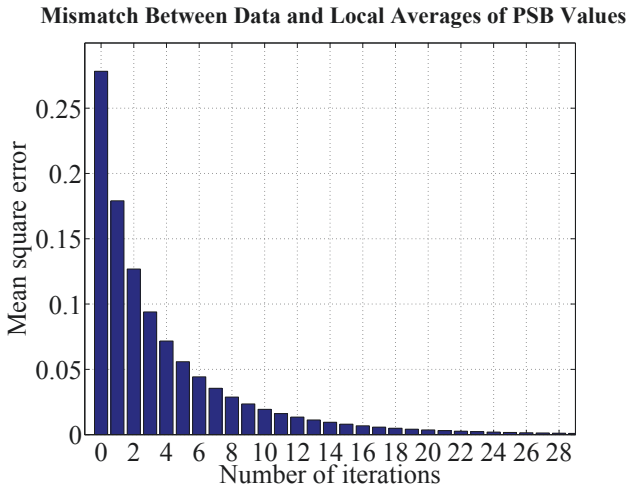


Figure 4.65: Example of 3-D point-scale block value representation – mismatch between original block-scale data and local averages of PSB values in terms of mean square error for each iteration.

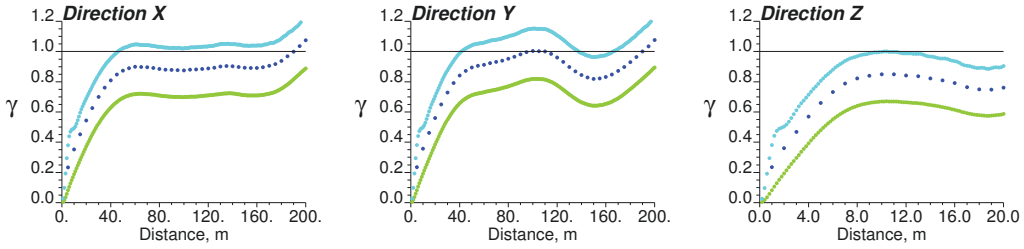


Figure 4.66: Example of 3-D point-scale block value representation – experimental variograms of block-scale data shown by dark blue line, experimental variograms of PSB values before correction shown by light green line, and experimental variograms of PSB values after correction shown by light blue line for major, minor, and vertical directions of continuity.

Chapter 5

Implementation Aspects

This chapter is devoted to various implementation aspects of the grid-free simulation, validation of the algorithm, and comparison of the proposed simulation method with other more conventional geostatistical simulation methods.

5.1 Data Pre-Processing and Simulation Post-Processing

The algorithm of grid-free simulation has been presented in normal units in previous Chapters 3 and 4. The real distributions of the geological properties are rarely normal. Therefore, a normal score transformation procedure is required to bring the conditioning data from original to normal units and back transform simulation results from normal units back to original units. Two normal score transformation procedures based on the table and kernel approximation of the CDF are explained in this section.

The grid-free simulation is performed at a point scale. But, sometimes, it is required to get realizations at a larger scale, for instance, for reserve estimation, when equipment selectivity is large. A post-processing upscaling procedure is explained to get simulation at a block scale in original units.

Simulation of non-stationary systems with the GFS, which are frequently encountered in practice, is also explained briefly.

5.1.1 Normal Score Transformation

Normal score transformation of the data values from original units to normal scores and back transformation of simulation values from normal units to original units is required for all Gaussian/normal simulation methods. The NST considers only transformation of the point-scale values or block-scale values resolved at a point-scale. The NST can be employed in different ways [17]. The NST approaches based on the transformation table and kernel approximation of the CDF of original data are described. The transformation table-based NST method is good for medium size data sets. The kernel-based NST method is more appropriate for small and large data sets. Despite the type of the approach, the NST procedure requires knowledge of CDF of original data distribution and the Gaussian distribution. It consists of a data pre-processing step and simulated values post-processing step.

NST is applied to univariate and multivariate systems, where different data types are transformed independently. An assumption about the multi-Gaussian behavior in normal space of different variables is made. For systems, where variables do not support the multi-Gaussian distribution, various decorrelation techniques may be applied. The altered distributions of the multiple random functions become normal and independent from each other. Stepwise conditional transformation allows decorrelating multivariate systems in normal units, but is limited to relatively few variables (3 to 5 at a time) [42]. Sufficiently large amount of the data is required for successful transformation. The projection pursuit multivariate transform is another option [6].

The transformation table-based NST is depicted in Fig. 5.1. The transformation proceeds as follows. First, the data values are transformed to normal

units. To do so, the unbiased CDF of the data Z^d is constructed in original units. The data values in original units $z(\boldsymbol{u}_\alpha), \alpha = 1, ..., N$, are sorted in ascending order with corresponding weights $\upsilon(\boldsymbol{u}_\alpha), \alpha = 1, ..., N$, where N is the total number of data. If the weights are not specified, the same weight $\upsilon(\boldsymbol{u}_\alpha) = 1/N$ is assigned to each datum $z(\boldsymbol{u}_\alpha)$. Then, the weights are transformed to $N+2$ CDF values $F(z)$, where the first and last CDF values are 0 and 1 (Eq. (5.1)) with corresponding data values being lower $z(\boldsymbol{u}_0)$ and upper $z(\boldsymbol{u}_{N+1})$ extrapolation tails specified by a modeler. The relationship between the rest of CDF values and corresponding weights is shown in Eq. (5.1). Finally, the CDF values $F(z)$ are mapped to the CDF $G(y)$ of normal distribution, and normal score values corresponding to the original data are found through the inversion of the normal CDF as in Eq. (5.2). The CDF $G(y)$ of normal scores can be easily computed numerically according to Eq. (5.3), where $erf()$ is the error function [62].

$$F\left(z(\boldsymbol{u}_\alpha)\right) = \begin{cases} 0.0, & \alpha = 0 \\ -0.5\upsilon(\boldsymbol{u}_1) + \sum\limits_{\alpha'=1}^{\alpha} \upsilon(\boldsymbol{u}_{\alpha'}), & \alpha = 1, ..., N \\ 1.0, & \alpha = N+1 \end{cases} \tag{5.1}$$

$$y(\boldsymbol{u}_\alpha) = G^{-1}\left(F\left(z(\boldsymbol{u}_\alpha)\right)\right), \ \alpha = 1, ..., N \tag{5.2}$$

$$G\left(y(\boldsymbol{u}_\alpha)\right) = 0.5\left(1 + erf\left(\frac{y(\boldsymbol{u}_\alpha)}{\sqrt{2}}\right)\right) \tag{5.3}$$

Second, the back transformation of the simulated value $y(\boldsymbol{u})$ or data $y(\boldsymbol{u}_\alpha)$ is performed. The CDF value $G(y(\boldsymbol{u}))$ is defined for the simulated value $y(\boldsymbol{u})$ in normal units, which is equivalent to the CDF $F(z(\boldsymbol{u}))$ in original units as in Eq. (5.4). Then, the simulated value $z(\boldsymbol{u})$ is inferred from interpolation as presented in Eq. (5.5). For the back transformation of simulated values from normal scores $y(\boldsymbol{u})$ to original units $z(\boldsymbol{u})$, the form of the CDF $F(z)$ in original units

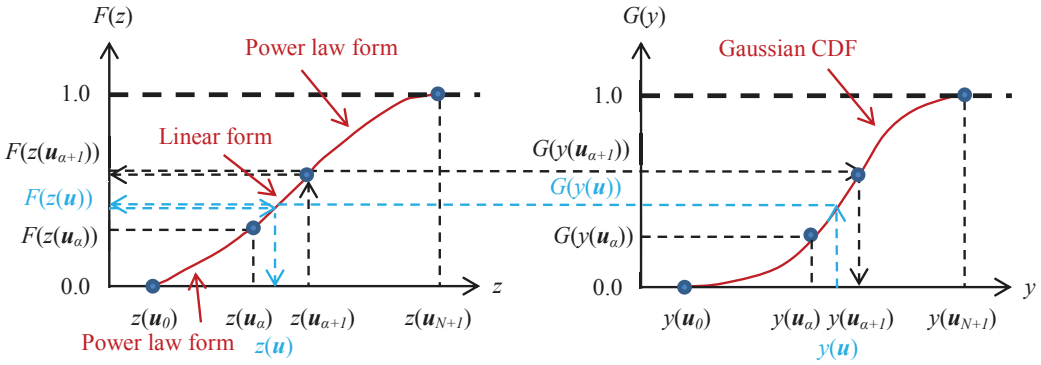


Figure 5.1: Schematic of normal score transformation procedure based on the transformation table.

is assumed linear between data values and in power law relationship between lower extrapolation tail $z(\boldsymbol{u}_0)$ and first data value $z(\boldsymbol{u}_1)$, and between last data value $z(\boldsymbol{u}_N)$ and upper extrapolation tail $z(\boldsymbol{u}_{N+1})$. The practical distribution of the tails is often non-uniform and quite important [17]. The ω_l and ω_u are the parameters that define nonlinear degree of the extrapolation functions for lower and upper tails, respectively.

$$F\left(z(\boldsymbol{u})\right) = G\left(y(\boldsymbol{u})\right) \tag{5.4}$$

$$z(\boldsymbol{u}) = \begin{cases} z(\boldsymbol{u}_0) + (z(\boldsymbol{u}_1) - z(\boldsymbol{u}_0)) \left(\dfrac{F(z(\boldsymbol{u}))}{F(z(\boldsymbol{u}_1))} \right)^{\omega_l}, \\ \qquad 0.0 \leq F(z(\boldsymbol{u})) \leq F(z(\boldsymbol{u}_1)) \\ z(\boldsymbol{u}_\alpha) + (z(\boldsymbol{u}_{\alpha+1}) - z(\boldsymbol{u}_\alpha)) \left(\dfrac{F(z(\boldsymbol{u})) - F(z(\boldsymbol{u}_\alpha))}{F(z(\boldsymbol{u}_{\alpha+1})) - F(z(\boldsymbol{u}_\alpha))} \right), \\ \qquad F(z(\boldsymbol{u}_\alpha)) \leq F(z(\boldsymbol{u})) \leq F(z(\boldsymbol{u}_{\alpha+1})), \\ \qquad \alpha = 1, ..., N-1 \\ z(\boldsymbol{u}_N) + (z(\boldsymbol{u}_{N+1}) - z(\boldsymbol{u}_N)) \left(\dfrac{1.0 - F(z(\boldsymbol{u}))}{1.0 - F(z(\boldsymbol{u}_N))} \right)^{\omega_u}, \\ \qquad F(z(\boldsymbol{u}_N)) \leq F(z(\boldsymbol{u})) \leq 1.0 \end{cases} \tag{5.5}$$

This tables-based NTS procedure should work the best for medium size data sets. It can handle between 20 to several ten thousand values. However, because of the imprecise interpolation between scarce data values, the use of the transformation table directly for very small data sets (less than 20 data values) is not reliable. On the other hand, the computational time might be an issue for very large data sets (larger than several ten thousand data values), where constructing and interpolating between large number of the tabulated values is a tediously long process. For these reasons, the kernel approximation of the data CDF is proposed for very small and very large data sets, where only limited manageable number N_A of the anchor points $\hat{z}_{\alpha'}, \alpha' = 0, ..., N_A - 1$, on the kernel curve is used in the NST instead of actual data $z(\boldsymbol{u}_\alpha), \alpha = 1, ..., N$ [27]. The number N_A of the anchor points is recommended to be around 101 to enable precise and fast transformation of the data values to normal scores and back transformation of simulated values to original units. When 101 anchor points are placed evenly on the CDF axis, the corresponding random function values become distribution percentiles. These anchor points form 100 intervals on the CDF plot. The kernel-based NST procedure also requires interpolation between

the data values $z(\boldsymbol{u}_\alpha)$ and corresponding CDF value $F\left(z(\boldsymbol{u}_\alpha)\right)$ pairs. The CDF values are computed in similar way as in Eq. (5.1) from the data weights. The local regression fitting procedure is presented in Eqs. (5.6) – (5.18), where the CDF is treated as an argument or independent variable, and data values are dependent variable. The axes are switched to ensure faster computation of the quantile values. Fig. 5.2 provides some basic principles of the local regression fitting, linear interpolation between anchor points, and back transformation of the simulated value from normal units to original units.

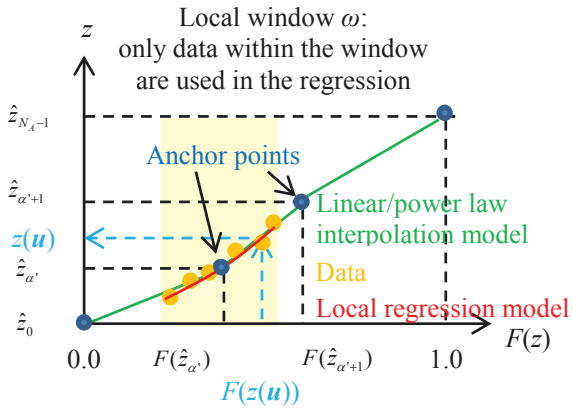


Figure 5.2: Schematic of normal score transformation procedure based on the kernel approximation of the data CDF.

The local linear regression in Eq. (5.6) states that a fitted anchor point $\hat{z}_{\alpha'}$ is the linear estimate derived from two local coefficients $a_{\alpha'}$ and $b_{\alpha'}$ and an argument in the form of CDF $F\left(\hat{z}_{\alpha'}\right)$, which is one of the evenly placed quantiles. The argument takes values shown in Eq. (5.7). The coefficients $a_{\alpha'}$ and $b_{\alpha'}$ are derived by minimizing local weighted mismatch between actual data values and their estimates from the same local regression model within some window of size ω on the CDF axis as presented in Eq. (5.8). The minimization of the error is expressed mathematically in Eq. (5.9). The locality is ensured by a kernel $K_\omega(\alpha', \boldsymbol{u}_\alpha)$, which is inversely proportional to the distance between input CDF

and its adjacent data CDF values as presented in Eq. (5.10). The parameter c is required to avoid division by zero, it should be relatively small. The local coefficients $a_{\alpha'}$ and $b_{\alpha'}$ can be easily derived in the matrix form of the local linear regression by taking derivative of the mismatch in respect to the coefficient vector as in Eqs. (5.11) – (5.18). The estimated data values are computed for $N_A - 2$ anchor values using Eq. (5.17), and remaining 2 anchor values at the tails of the distribution are specified by a user.

$$\hat{z}(\boldsymbol{u}_{\alpha'}) = a_{\alpha'} + b_{\alpha'} F\left(\hat{z}_{\alpha'}\right) \tag{5.6}$$

$$F\left(\hat{z}_{\alpha'}\right) = \frac{\alpha'}{N_A - 1},\ \alpha' = 0, ..., N_A - 1 \tag{5.7}$$

$$\hat{z}(\boldsymbol{u}_{\alpha}) = a_{\alpha'} + b_{\alpha'} F\left(z(\boldsymbol{u}_{\alpha})\right) \tag{5.8}$$

$$\varepsilon_{\alpha'} = \sum_{\alpha=1}^{N} K_{\omega}(\alpha', \boldsymbol{u}_{\alpha}) \left(\hat{z}(\boldsymbol{u}_{\alpha}) - z(\boldsymbol{u}_{\alpha})\right)^2 \tag{5.9}$$

$$K_{\omega}(\alpha', \boldsymbol{u}_{\alpha}) = \begin{cases} \frac{1}{|F(\hat{z}_{\alpha'}) - F(z(\boldsymbol{u}_{\alpha}))| + c}, & \left|F\left(\hat{z}_{\alpha'}\right) - F\left(z(\boldsymbol{u}_{\alpha})\right)\right| \leq \omega \\ 0, & \left|F\left(\hat{z}_{\alpha'}\right) - F\left(z(\boldsymbol{u}_{\alpha})\right)\right| > \omega \end{cases} \tag{5.10}$$

$$\min \varepsilon_{\alpha'} = \left[\boldsymbol{Z}^d - \boldsymbol{F}\boldsymbol{P}_{\alpha'}\right]^T \boldsymbol{K}_{\alpha'} \left[\boldsymbol{Z}^d - \boldsymbol{F}\boldsymbol{P}_{\alpha'}\right] \tag{5.11}$$

$$\boldsymbol{Z}^d = \begin{bmatrix} z(\boldsymbol{u}_1) \\ ... \\ z(\boldsymbol{u}_N) \end{bmatrix} \tag{5.12}$$

$$\boldsymbol{F} = \begin{bmatrix} 1 & F\left(z(\boldsymbol{u}_1)\right) \\ ... & ... \\ 1 & F\left(z(\boldsymbol{u}_N)\right) \end{bmatrix} \tag{5.13}$$

$$\boldsymbol{P}_{\alpha'} = \begin{bmatrix} a_{\alpha'} \\ b_{\alpha'} \end{bmatrix} \tag{5.14}$$

$$\boldsymbol{K}_{\alpha'} = [\boldsymbol{K}_{\alpha'}]^T = \begin{bmatrix} K_\omega(\alpha', \boldsymbol{u}_1) & ... & 0 \\ ... & & ... \\ 0 & & K_\omega(\alpha', \boldsymbol{u}_N) \end{bmatrix} \tag{5.15}$$

$$\frac{\partial \varepsilon_{\alpha'}}{\partial \boldsymbol{P}_{\alpha'}} = 0 \Rightarrow \hat{\boldsymbol{P}}_{\alpha'} = \left[\boldsymbol{F}^T \boldsymbol{K}_{\alpha'} \boldsymbol{F}\right]^{-1} \boldsymbol{F}^T \boldsymbol{K}_{\alpha'} \boldsymbol{Z}^d \tag{5.16}$$

$$\hat{z}_{\alpha'} = \boldsymbol{F}'_{\alpha'} \hat{\boldsymbol{P}}_{\alpha'} = \boldsymbol{F}'_{\alpha'} \left[\boldsymbol{F}^T \boldsymbol{K}_{\alpha'} \boldsymbol{F}\right]^{-1} \boldsymbol{F}^T \boldsymbol{K}_{\alpha'} \boldsymbol{Z}^d \tag{5.17}$$

$$\boldsymbol{F}'_{\alpha'} = [1 \;\; F\left(\hat{z}_{\alpha'}\right)] \tag{5.18}$$

Therefore, N_A anchor points $\hat{z}_{\alpha'}, \alpha' = 0, ..., N_A - 1$, are defined for each regularly spaced CDF values. Then, the NST procedure is performed as described above, where data values are replaced with the anchor values and CDF values are equally spaced. The normal score transformation is carried out according to Eqs. (5.19) and (5.20) taking into account linear interpolation between anchor points. First, the approximated CDF $\hat{F}\left(z(\boldsymbol{u}_\alpha)\right)$ of each data point is found through linear interpolation between anchor points computed from local regression. Second, the quantile of the Gaussian distribution is found by matching approximated CDF value of the data point with Gaussian CDF value.

$$\begin{aligned}\hat{F}\left(z(\boldsymbol{u}_\alpha)\right) = F\left(\hat{z}_{\alpha'}\right) + \left(F\left(\hat{z}_{\alpha'+1}\right) - F\left(\hat{z}_{\alpha'}\right)\right)\left(\frac{z(\boldsymbol{u}_\alpha) - \hat{z}_{\alpha'}}{\hat{z}_{\alpha'+1} - \hat{z}_{\alpha'}}\right), \\ \hat{z}_{\alpha'} \leq z(\boldsymbol{u}_\alpha) \leq \hat{z}_{\alpha'}, \\ \alpha' = 0, ..., N_A - 1,\ \alpha = 1, ..., N\end{aligned} \tag{5.19}$$

$$y(\boldsymbol{u}_\alpha) = G^{-1}\left(\hat{F}\left(z(\boldsymbol{u}_\alpha)\right)\right),\ \alpha = 1, ..., N \tag{5.20}$$

The back transformation of the simulated value $y(\boldsymbol{u})$ is carried out according to Eqs. (5.21) and (5.22), which are very similar to the equations from previous section with data replaced by anchor values.

$$F\left(z(\boldsymbol{u})\right) = G\left(y(\boldsymbol{u})\right) \tag{5.21}$$

$$\begin{aligned}z(\boldsymbol{u}) = \hat{z}_{\alpha'} + \left(\hat{z}_{\alpha'+1} - \hat{z}_{\alpha'}\right)\left(\frac{F\left(\hat{z}(\boldsymbol{u})\right) - F\left(\hat{z}_{\alpha'}\right)}{F\left(\hat{z}_{\alpha'+1}\right) - F\left(\hat{z}_{\alpha'}\right)}\right), \\ F\left(\hat{z}_{\alpha'}\right) \leq F\left(z(\boldsymbol{u})\right) \leq F\left(\hat{z}_{\alpha'+1}\right), \\ \alpha' = 0, ..., N_A - 1\end{aligned} \tag{5.22}$$

An example of kernel-based normal score transformation of the small data set is presented in Fig. 5.3. The data set consists of 23 data values. The local window is 10% of the CDF range with the c parameter of 1% of the CDF range. The data points are presented as red dots. A total of 101 anchor points are shown as blue dots as a result of the fitting. The regression fit follows the data values quite closely. The normal scores of the data do not look symmetric, because the interpolated CDF values from the local linear regression model are not equally spaced any longer. The statistics of the normal scores of the data are close to the target zero mean and unit variance.

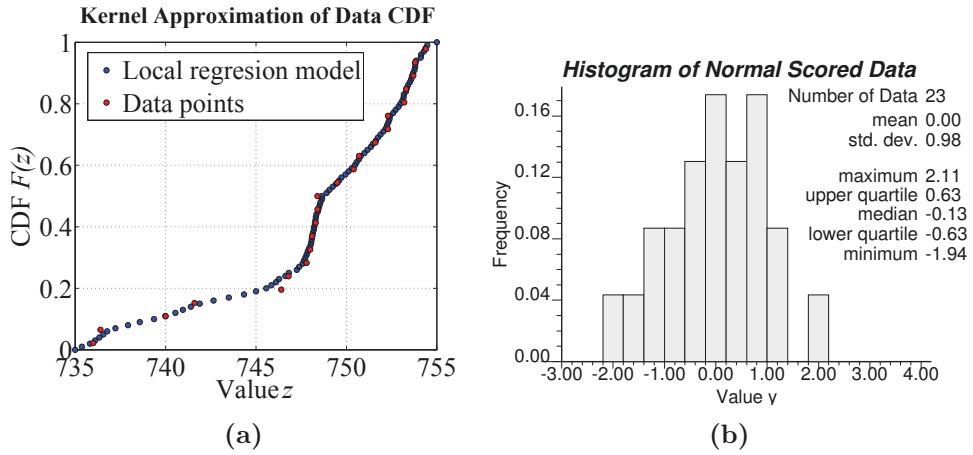


Figure 5.3: Example of kernel-based approximation of data CDF curve – (a) local regression model fitted to the CDF of small data set and (b) histogram of normal scored data.

5.1.2 Block Scale Realizations

Simulation at a block scale is performed by simulating at a point scale within the blocks, and then computing the block-scale values by averaging the point-scale values. This is straightforward when the property averages linearly with scale; Eq. (2.3) presented in Chapter 2 is used to compute effective value at a block scale from point-scale values. Other properties such as permeability would be upscaled based on appropriate physics such as the flow simulation equations [18].

Ideally, an infinitely large number of the point-scale simulation nodes should be used to compute each block-scale simulation value $Z(V(\boldsymbol{u}))$ tied to a location $\boldsymbol{u}$. In practice, the optimal number of the discretization points that could be simulated with the GFS approach at a point scale must be considered. Two cases are examined: conditional 2-D simulation with a small variogram range and unconditional 3-D simulation with a larger variogram range. The 3-D case with a variogram range much larger than the block size is frequently encountered in practice. Porosity and permeability properties are selected for the

study. Data values are transformed to normal units for the simulation, and the simulated point-scale values are mapped back to original units for the upscaling.

The spatial distribution of the conditioning data is an evenly spaced 16 locations as presented in Fig. 5.4. The size of the 2-D model, which represents a block-scale volume, is 100.0×100.0 m^2. Simulated block-scale values of porosity and permeability are found by simulating point-scale values within this model at the discretization nodes and upscaling them to the block scale. Porosity is upscaled by simply taking arithmetic average of the simulated values within the model. Permeability is upscaled by running flow simulation (solving the pressure equation based on conservation of mass and Darcy's Law) for the model and inferring a single effective value for each principal direction. The number of discretization nodes for each of two dimensions is varied between 2 and 50 with an increment of 2. An isotropic variogram with a variogram range of 10.0 m in one case and 100.0 m in another is selected for the simulation. These variogram ranges are smaller than the block-scale. A total of 10 realizations are simulated. First realizations of porosity and permeability at 1×1, 18×18, and 50×50 discretization nodes for small and medium variogram ranges are shown in Figs. 5.5 and 5.7, respectively. Data reproduction and the proper spatial structure are seen on the realization maps. The upscaled results are summarized in Figs. 5.6 and 5.8 for ten realizations for various numbers of discretization nodes. Each plot indicates the relationship between the number of point-scale discretization values and the ratio of the effective value, which corresponds to the upscaled point-scale values, to the effective block-scale value computed from 50×50 point-scale values. For the short variogram range, the effective values stabilize at around 10 discretization nodes in each dimension for both porosity and permeability. For medium variogram range, the effective values level out at around 6 discretization nodes in each dimension also for both porosity and permeability. Porosity stabilizes faster than permeability. This indicates that

somewhere from 6 × 6 to 10 × 10 point-scale values within each model block produce reasonable results for 2-D systems.

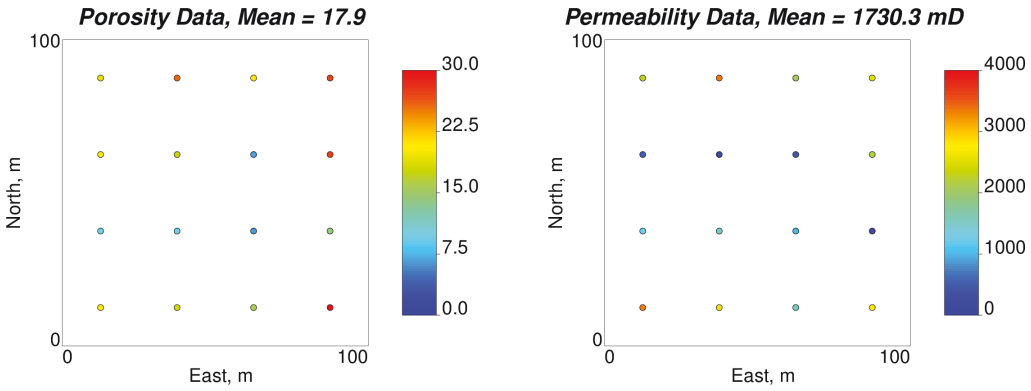


Figure 5.4: Choice of number of point-scale discretization values in 2-D – porosity and permeability data location maps.

The results are slightly different for upscaling 3-D properties. Single permeability property is generated at a point-scale and flow-based upscaled to a block scale. The model size (block volume) is selected to be 100.0 × 100.0 × 10.0 m^3. Isotropic variogram ranges are 50.0 m and 1000.0 m for two cases. The first of ten realizations is shown in Figs. 5.9 and 5.11 at 1 × 1 × 1, 6 × 6 × 6, and 20 × 20 × 20 discretization points for medium and long variogram ranges. The number of discretization points varies between 1 and 20 with an increment of 1 between 1 and 10, and with an increment of 2 between 10 and 20. The upscaled results for ten realizations through the flow simulation are shown in Figs. 5.10 and Fig. 5.12 for different variogram ranges. The upscaled values for various numbers of discretization nodes are shown relative to the effective values at 20 × 20 × 20 discretization density in the form of their ratio. In this 3-D unconditional case, the effective values become stable at about 5 discretization nodes in each dimension for medium variogram range and 4 discretization nodes in each dimension for long variogram range.

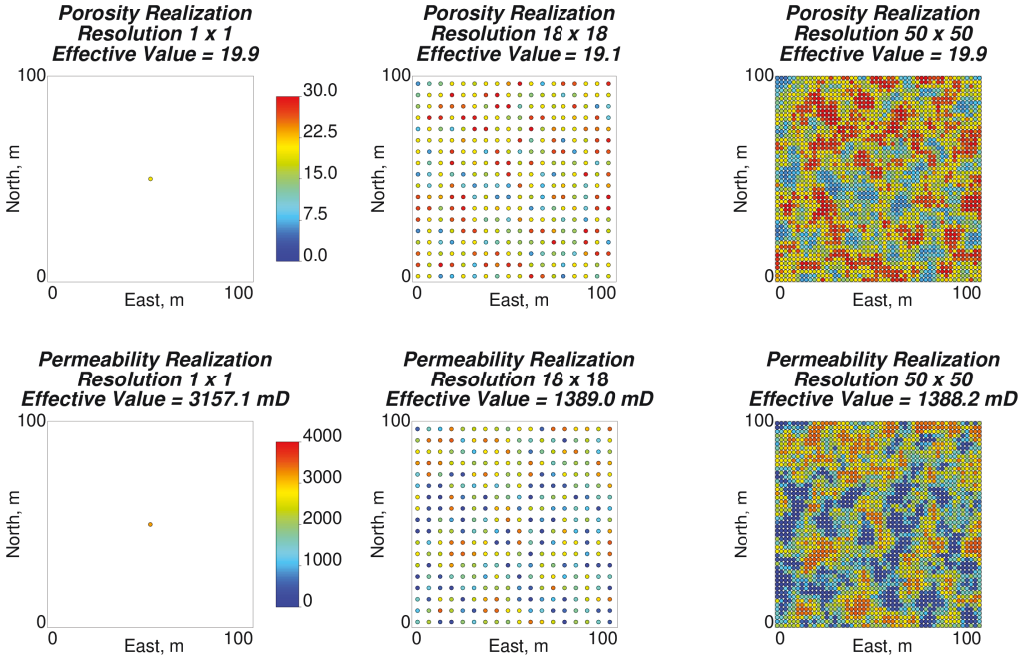


Figure 5.5: Location maps of point-scale discretization values for porosity and permeability – location map of single discretization value, location map of 18 × 18 discretization values, and location map of 50 × 50 discretization values used for block representation in 2-D. Point-scale values are simulated conditionally with 10.0 m variogram range.

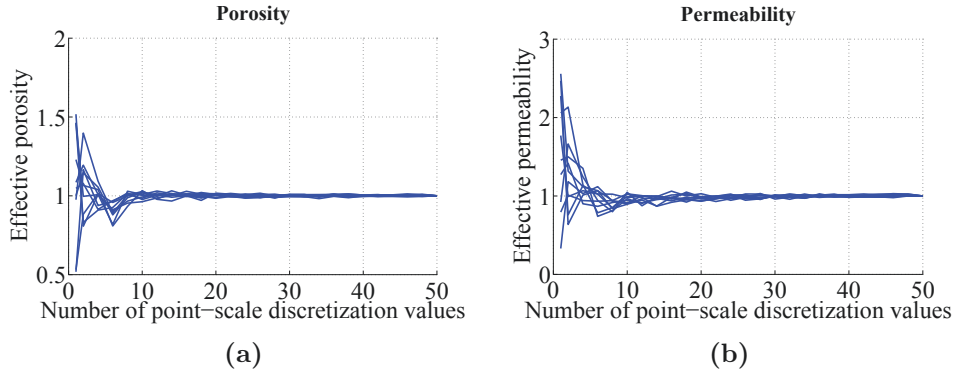


Figure 5.6: Choice of number of point-scale discretization values in 2-D – relationship between upscaled rock properties and number of point-scale discretization values: (a) porosity, and (b) permeability. Point-scale values are simulated conditionally with 10.0 m variogram range.

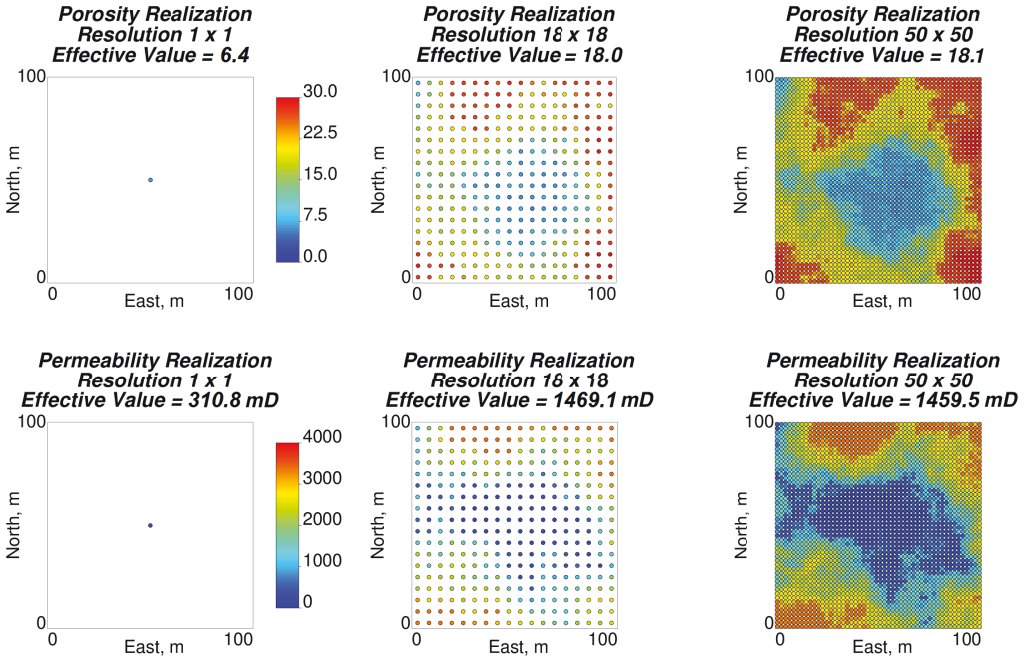


Figure 5.7: Location maps of point-scale discretization values for porosity and permeability – location map of single discretization value, location map of 18 × 18 discretization values, and location map of 50 × 50 discretization values used for block representation in 2-D. Point-scale values are simulated conditionally with 100.0 m variogram range.

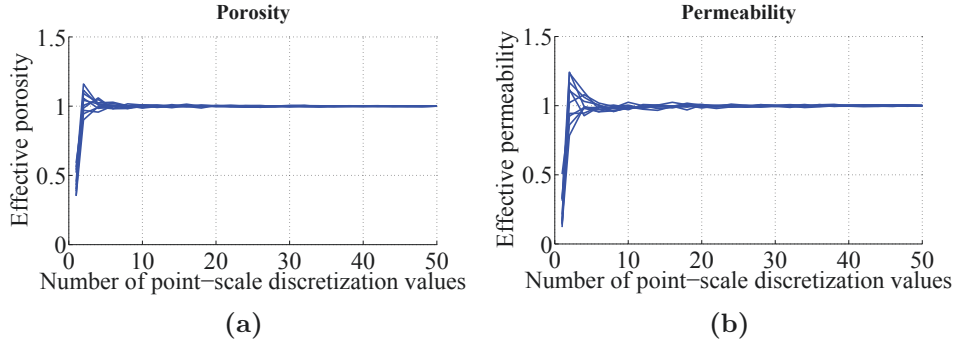


Figure 5.8: Choice of number of point-scale discretization values in 2-D – relationship between upscaled rock properties and number of point-scale discretization values: (a) porosity, and (b) permeability. Point-scale values are simulated conditionally with 100.0 m variogram range.

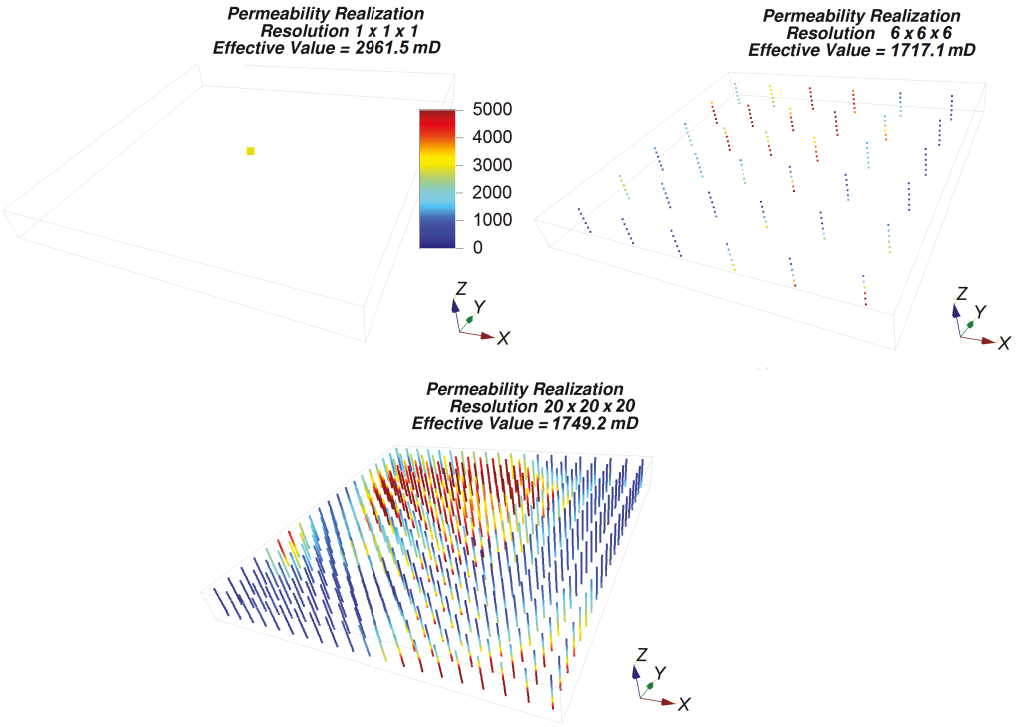


Figure 5.9: Location maps of point-scale discretization values for permeability – location map of single discretization value, location map of $6 \times 6 \times 6$ discretization values, and location map of $20 \times 20 \times 20$ discretization values used for block representation in 3-D. Point-scale values are simulated unconditionally with 50.0 m variogram range.

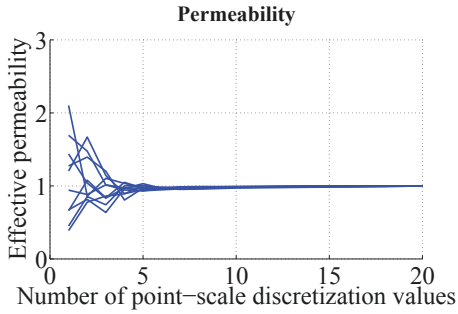


Figure 5.10: Choice of number of point-scale discretization values in 3-D – relationship between upscaled permeability and number of point-scale discretization values. Point-scale values are simulated unconditionally with 50.0 m variogram range.

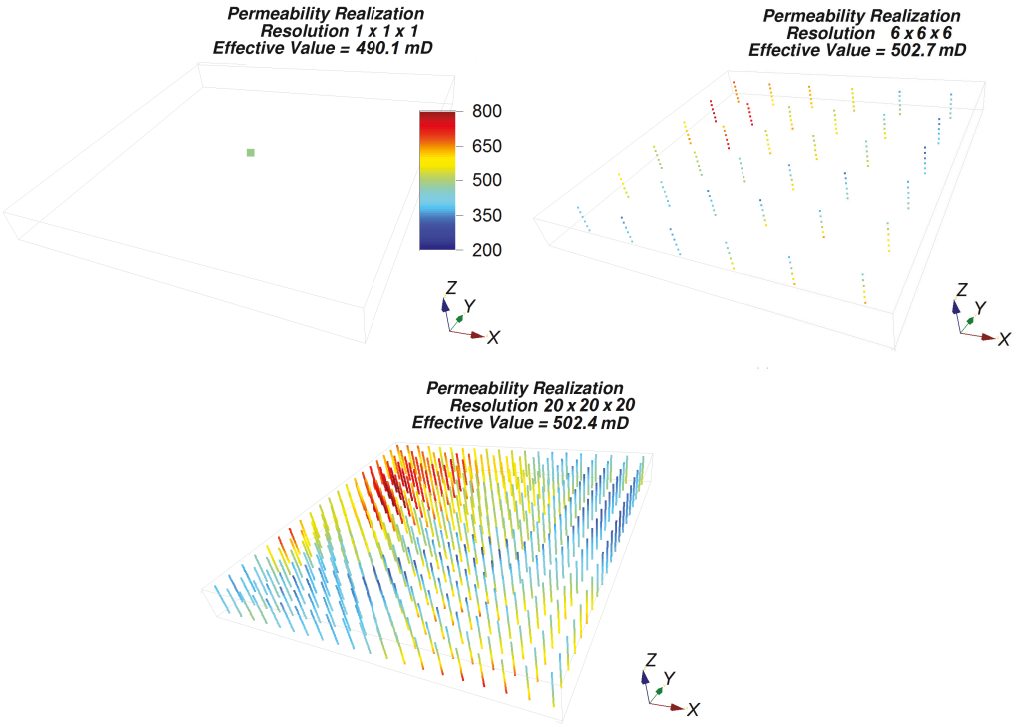


Figure 5.11: Location maps of point-scale discretization values for permeability – location map of single discretization value, location map of 6 × 6 × 6 discretization values, and location map of 20 × 20 × 20 discretization values used for block representation in 3-D. Point-scale values are simulated unconditionally with 1000.0 m variogram range.

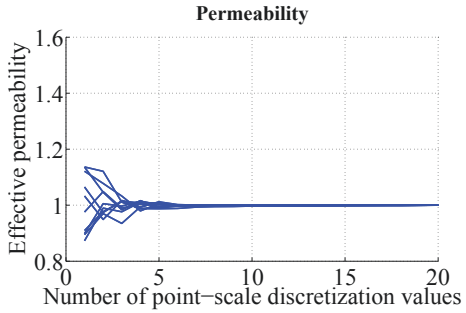


Figure 5.12: Choice of number of point-scale discretization values in 3-D – relationship between upscaled permeability and number of point-scale discretization values. Point-scale values are simulated unconditionally with 1000.0 m variogram range.

Therefore based on the conditional 2-D case study and unconditional 3-D case study, it is recommended to simulate between 4 and 10 point-scale values with the GFS in each dimension to generate block-scale values with a minimum computational effort.

5.1.3 Modeling Non-Stationary Systems

There are many forms of non-stationarity. The treatment of non-stationarity in this thesis is only partial and quite conventional; non-stationary systems are modeled with the GFS by modeling boundary between or zones of stationary domains and performing GFS within each stationary domain separately. Sequential indicator simulation, multiple point statistics simulation, and object-based modeling are some examples of modeling categorical domains [31, 64, 74].

A case study of modeling mineral grade in a 2-D cross-sectional view, which is represented by two stationary domains A and B separated by a fault, is shown below. Even though the boundary between two zones are modeled deterministically here, it can be also modeled stochastically. It is assumed that the location of the fault is known precisely. It is believed that domain A to the left of the fault is presented by horizontally stacked geologic strata, while domain B to the right of the fault is tilted. Therefore, to account for such geological deposition and study the area around fault more precisely, a particular configuration of the simulation node locations has been chosen. A 2-D slice with the simulation nodes and conditioning data are shown in Fig. 5.13 along with data values and their histograms. The data from the two populations are distinct. Domain A contains lower values than domain B. A single realization is generated for the two domains separately at point-scale locations in normal scores units and later back transformed to original units. The spatial structure of the entire system in normal units is determined by the covariance functions shown in Eqs. (5.23) and (5.24) for domains A and B, respectively. The result of the simulation is

shown in Fig. 5.14. Here, two domains are shown separately and together as a single model. Another representation of the simulated system can be found in Fig. 1.8 of Chapter 1, where polygonal declustering is performed to divide the modeled area fairly between the simulation nodes. The histogram reproduction for separated domains is accurate. The combined histogram looks a bit different from the data histogram, because there are more simulation nodes for domain B (1488 simulation nodes) than for domain A (810 simulation nodes). Therefore, it is relatively straightforward to generate a realization of the non-stationary system with the GFS method on the irregular grid taking into account several distinct stationary domains, which have to be defined beforehand by categorical variable simulation techniques.

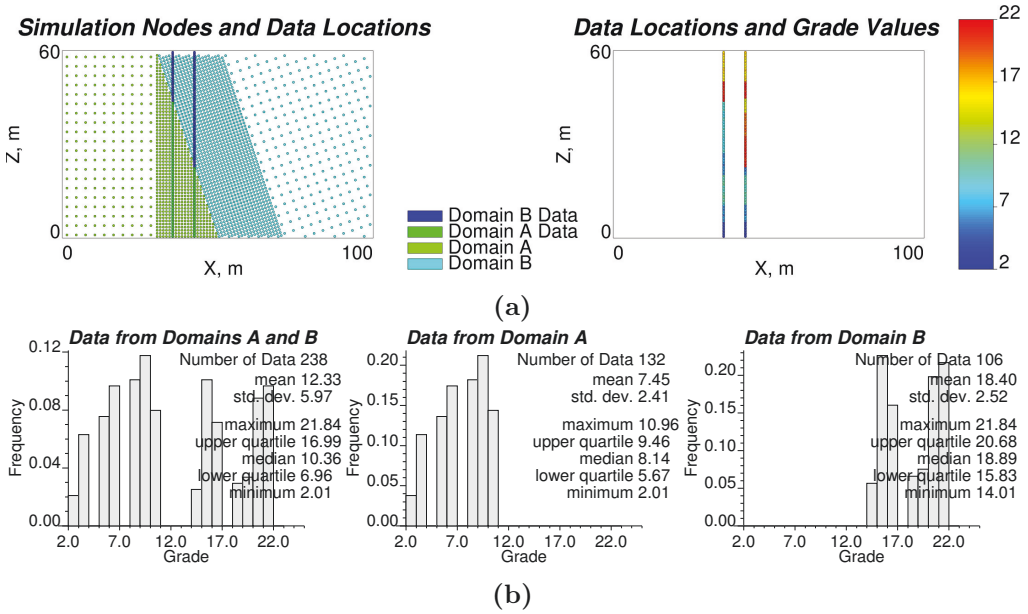


Figure 5.13: Modeling a non-stationary system – (a) locations of simulation nodes and data for stationary domains A and B, (b) histograms of the data for domains A and B separately and together.

$$C_{Y_A}(\boldsymbol{h}) = 0.1Nugget\,(\boldsymbol{h}) + 0.9Sph_{\substack{r_2=100.0\\ r_3=20.0\\ \alpha=90.0°}}\,(\boldsymbol{h}) \tag{5.23}$$

$$C_{Y_B}(\boldsymbol{h}) = 0.1Nugget\,(\boldsymbol{h}) + 0.9Sph_{\substack{r_2=100.0\\ r_3=20.0\\ \alpha=-28.4°}}\,(\boldsymbol{h}) \tag{5.24}$$

As another example, the non-stationarity could be presented by a trend $\tilde{Z}(\boldsymbol{u})$. Detrending would be performed as shown in Eq. (2.27) of Chapter 2. Then, the trend $m_Z(\boldsymbol{u})$ and local variations $Z(\boldsymbol{u})$ are modeled in a grid-free manner and combined.

5.2 Additional Implementation Details

Additional implementation aspects of the grid-free simulation and its application including simulation with zonal anisotropy, grid coordinates transformation, fast matrix inversion, and the two-level grid-free simulation code paradigm are discussed in this section.

5.2.1 Zonal Anisotropy Simulation

Natural phenomena quite often exhibit spatial structure with zonal anisotropy, which implies that variogram does not reach the sill within the size of the simulation domain in one of the principal directions of the continuity, i.e. there is a clear zonation in one or more directions not seen in the other directions [64]. Simulation with zonal anisotropy is implemented in the grid-free algorithm through the representation of the simulation Y as a linear model of (co-)regionalization, where some independent factors X are presented in a lower dimensional space than the model itself. This approach is sometimes referred to a lower dimensional decomposition of the variogram [10].

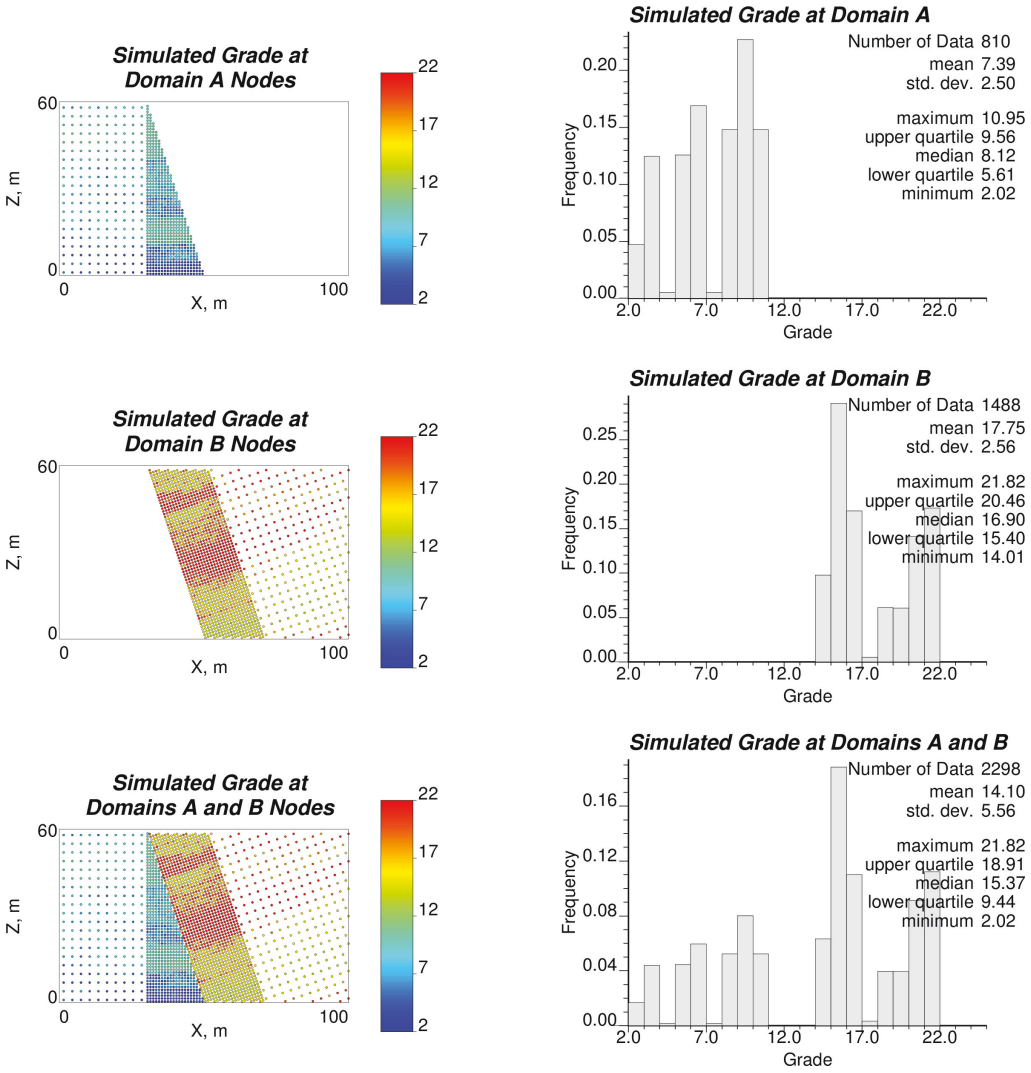


Figure 5.14: Modeling s non-stationary system – simulation results for stationary domains A and B separately and together, and corresponding histograms.

The simulation of a single random function Y with zonal anisotropy is presented and can be easily extended to a multivariate case. A target random function Y can be expressed as a weighted sum of the independent random factors $X_p, p = 0, ..., P$, as shown in Eq. (5.25). The argument of the variables, spatial coordinate $\boldsymbol{u}$, is decomposed into its projections u_1, u_2, and u_3 on three principal directions of continuity as presented in Eq. (5.26). The first two principal directions of continuity are usually defined in the horizontal plane (especially for stratabound deposits), and the third one corresponds to a vertical direction. The parameters ω_1, ω_2, and ω_3 define zonal anisotropy in these three principal directions of continuity and are stored in a matrix $\boldsymbol{\omega}$ as shown in Eq. (5.27). The parameters take binary values 0 or 1, where 0 is used when corresponding dimension of a random factor is omitted, and 1 is used if corresponding dimension of a random factor is present. Therefore, by adjusting parameters in $\boldsymbol{\omega}$ the dimensionality of the constituent random factors $X_p, p = 0, ..., P$, can be lowered as presented in Eq. (5.28). Note that only last random factors can exhibit lower dimensionality in the simulated random function Y. Covariance function of Y can be decomposed into weighted sum of covariances of random factors $X_p, p = 0, ..., P$, as shown in Eq. (5.29). The covariance weights are squares of random factor coefficients $a_p, p = 0, ..., P$, in the LMR/LMC definition. As a result, the separation lag vector $\boldsymbol{h}$ is also decomposed into three components aligned with the three principal directions of the continuity as in Eq. (5.30) and combined with zonal anisotropy parameters as in Eq. (5.31). By looking at the standardized lag distance in Eq. (5.32), which is used in the computation of the covariance values in the simulation, it can be interpreted that due to binary 0-1 nature of the parameters $\boldsymbol{\omega}$, they lower the dimension of the covariances of the random factors and target random function Y.

$$Y(\boldsymbol{u}) = \sum_{p=0}^{P} a_p X_P(\boldsymbol{u}) \tag{5.25}$$

$$\boldsymbol{u} = [u_1 \;\; u_2 \;\; u_3]^T \tag{5.26}$$

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix} \tag{5.27}$$

$$\boldsymbol{\omega u} = [\omega_1 u_1 \;\; \omega_2 u_2 \;\; \omega_3 u_3]^T \tag{5.28}$$

$$C_Y(\boldsymbol{h}) = \sum_{p=0}^{P} a_p^2 C_{X_P}(\boldsymbol{h}) \tag{5.29}$$

$$\boldsymbol{h} = [h_1 \;\; h_2 \;\; h_3]^T \tag{5.30}$$

$$\boldsymbol{\omega h} = [\omega_1 h_1 \;\; \omega_2 h_2 \;\; \omega_3 h_3]^T \tag{5.31}$$

$$\frac{\boldsymbol{h}}{\boldsymbol{r}} = \sqrt{\left(\frac{\omega_1 h_1}{r_1}\right)^2 + \left(\frac{\omega_2 h_2}{r_2}\right)^2 + \left(\frac{\omega_3 h_3}{r_3}\right)^2} \tag{5.32}$$

An example of a realization exhibiting zonal anisotropy is shown in Fig. 5.15. Fifty unconditional realizations are generated in normal units with the grid-free simulation approach, spatial structure of which is spherical with zonal anisotropy presented in Eq. (5.33). The variogram range in the major direction of continuity (red dots) is 30.0 units, which comprises only 0.5 of the total sill 1.0, and variogram range in minor direction of continuity (blue dots) is 10.0 units. The target variogram with zonal anisotropy is reproduced by average of

the individual experimental variograms computed from the realizations. The features of the zonal anisotropy also appear on the realization map.

$$C_Y(\boldsymbol{h}) = 0.5Sph_{\substack{r_1=30.0\\r_2=10.0\\\alpha=90.0°}}(\boldsymbol{h}) + 0.5Sph_{\substack{r_2=10.0\\\alpha=90.0°}}(\boldsymbol{h}) \tag{5.33}$$

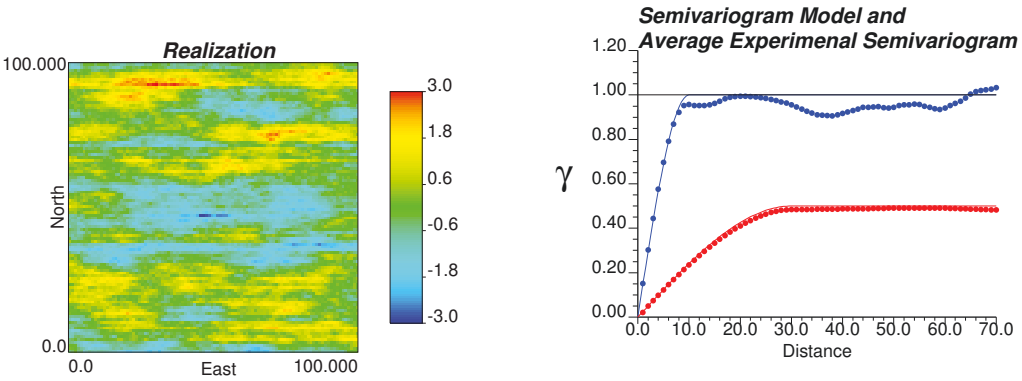


Figure 5.15: Simulation with zonal anisotropy – realization map and variogram reproduction. Red dots represent direction of maximum continuity. Blue dots represent direction of minimum continuity. Variogram model is shown by lines.

5.2.2 Grid Coordinates Transformation

Sometimes the grid of exhaustively sampled data does not align with the coordinate axes. Coordinates x, y, and z of the simulation location or conditioning data should be transformed to local coordinates within the exhaustively sampled data grid to find location of the collocated pseudo point-scale block secondary data value, which is later interpolated in between the block-scale data. The data grid is characterized by number of the blocks N_1, N_2, and N_3 for three dimensional data set, size of a regular block S_1, S_2, and S_3, Cartesian coordinates of a center of the first block x_{min}, y_{min}, and z_{min}, and rotational angles plunge α_X, dip α_Y, and azimuth α_Z, which are formed by rotating coordinate axes X, Y, and Z to align with the data grid. The blocks are indexed as shown in Fig. 4.52

of Chapter 4 starting from bottom lower left block, cycling index i first along X axis, then index j along Y axis, and finally index k along Z axis. The transformation, which defines coordinates x', y', and z' of secondary data location in the data grid coordinate system, is depicted in Eq. (5.34) [17]. The rotation matrices $\boldsymbol{M}_{R_{X'}}$, $\boldsymbol{M}_{R_{Y'}}$, and $\boldsymbol{M}_{R_{Z'}}$ are defined in Eqs. (5.35) – (5.37). The center of first data grid block ($i = 1, j = 1, k = 1$) is assumed to be the origin of the local coordinate system. Eq. (5.38) defines the position of the location (x, y, z) relative to the indices of the data block (i, j, k), which are necessarily for the PSB value representation in Eq. (4.56).

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \boldsymbol{M}_{R_{X'}} \boldsymbol{M}_{R_{Y'}} \boldsymbol{M}_{R_{Z'}} \begin{bmatrix} x - x_{min} \\ y - y_{min} \\ z - z_{min} \end{bmatrix} \tag{5.34}$$

$$\boldsymbol{M}_{R_{X'}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_X & -\sin\alpha_X \\ 0 & \sin\alpha_X & \cos\alpha_X \end{bmatrix} \tag{5.35}$$

$$\boldsymbol{M}_{R_{Y'}} = \begin{bmatrix} \cos\alpha_Y & 0 & -\sin\alpha_Y \\ 0 & 1 & 0 \\ \sin\alpha_Y & 0 & \cos\alpha_Y \end{bmatrix} \tag{5.36}$$

$$\boldsymbol{M}_{R_{Z'}} = \begin{bmatrix} \cos\alpha_Z & \sin\alpha_Z & 0 \\ -\sin\alpha_Z & \cos\alpha_Z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.37}$$

$$\begin{bmatrix} i \\ j \\ k \end{bmatrix} = int \left\{ \begin{bmatrix} \frac{1}{S_1} & 0 & 0 \\ 0 & \frac{1}{S_2} & 0 \\ 0 & 0 & \frac{1}{S_3} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \right\} \tag{5.38}$$

Similar Eq. (5.39) can be applied to define Cartesian coordinates $x_{i,j,k}$, $y_{i,j,k}$, and $z_{i,j,k}$ of a simulation node (i, j, k) of the rotated grid, when GFS algorithm is performed on a grid. Here, x_{min}, y_{min}, and z_{min} are the Cartesian coordinates of the first simulation node $(i = 1, j = 1, k = 1)$, and fixed distances between any two adjacent nodes are S_1, S_2, and S_3. The rotation matrices $\boldsymbol{M}_{R_X}$, $\boldsymbol{M}_{R_Y}$, and $\boldsymbol{M}_{R_Z}$ were defined in Eqs. (3.73) – (3.75) of Chapter 3.

$$\begin{bmatrix} x_{i,j,k} \\ y_{i,j,k} \\ z_{i,j,k} \end{bmatrix} = \begin{bmatrix} x_{min} \\ y_{min} \\ z_{min} \end{bmatrix} + \boldsymbol{M}_{R_X} \boldsymbol{M}_{R_Y} \boldsymbol{M}_{R_Z} \begin{bmatrix} (i-1)S_1 \\ (j-1)S_2 \\ (k-1)S_3 \end{bmatrix} \tag{5.39}$$

Therefore, transformation of the coordinates is an integral part of grid-free simulation conditional to the exhaustively sampled gridded data or when the simulation is performed on the rotated grid of the simulation nodes.

5.2.3 Matrix Inversion Algorithm

The matrix inversion is the computationally demanding aspect of the conditioning step of the grid-free simulation. The conditioning is performed with (co)kriging in a dual form. Robust and efficient inversion of the data covariance matrix is explained [62].

The data covariance matrix $\boldsymbol{C}$ is a square symmetric matrix of size $N \times N$, which should be inverted once to compute dual kriging weights. Therefore, square matrix inversion procedure should be implemented. The product of the matrix and its inverse is an identity matrix as shown in Eq. (5.40), where inverse $\boldsymbol{C}^{-1}$ is an unknown quantity, and the identity matrix $\boldsymbol{I}$ is a $N \times N$

diagonal matrix with zero terms except unit diagonal elements. This equality can be seen as a set of linear systems of Eqs. (5.41). By rewriting this set of linear systems of equations and looking at its parts separately as in Eq. (5.42), it becomes clear that there are N sets that have to be solved separately to get all elements of the matrix inverse $\boldsymbol{C}^{-1}$. Each system of equations in Eq. (5.42) can be stated as a separate problem and simply rewritten as in Eq. (5.43). The solution to this equation can be presented through LU decomposition or Cholesky decomposition for the symmetric positive-definite matrix of $\boldsymbol{C}$ as presented in Eq. (5.44). Matrix $\boldsymbol{L}$ is the lower triangular matrix of the Cholesky decomposition of the matrix $\boldsymbol{C}$, see Eq. (5.45). Matrix $\boldsymbol{L}^T$ is the transpose of $\boldsymbol{L}$. Linear system of equations (5.43) can be rewritten into two separate equations as shown in Eq. (5.46) for easy and fast computation of vector $\boldsymbol{x}$, which contains some elements of $\boldsymbol{C}^{-1}$. First, elements of $\boldsymbol{L}$ are computed as shown in Eq. (5.47). Then, vectors $\boldsymbol{y}$ and $\boldsymbol{x}$ can be easily found as in Eqs. (5.48) and (5.49), respectively, for each of N linear systems of equations (5.42). Finally, the inverse $\boldsymbol{C}^{-1}$ is reconstructed from $\boldsymbol{x}$ values found from each of the system of equations.

$$\boldsymbol{C}\boldsymbol{C}^{-1} = \boldsymbol{I} \tag{5.40}$$

$$\begin{bmatrix} C_{11} & \dots & C_{1N} \\ \dots & & \dots \\ C_{N1} & \dots & C_{NN} \end{bmatrix} \begin{bmatrix} C'_{11} & \dots & C'_{1N} \\ \dots & & \dots \\ C'_{N1} & \dots & C'_{NN} \end{bmatrix} = \begin{bmatrix} 1 & \dots & 0 \\ \dots & & \dots \\ 0 & \dots & 1 \end{bmatrix} \tag{5.41}$$

$$\begin{cases} \begin{bmatrix} C_{11} & ... & C_{1N} \\ ... & & ... \\ C_{N1} & ... & C_{NN} \end{bmatrix} \begin{bmatrix} C'_{11} \\ ... \\ C'_{N1} \end{bmatrix} = \begin{bmatrix} 1 \\ ... \\ 0 \end{bmatrix} \\ ... \\ \begin{bmatrix} C_{11} & ... & C_{1N} \\ ... & & ... \\ C_{N1} & ... & C_{NN} \end{bmatrix} \begin{bmatrix} C'_{1N} \\ ... \\ C'_{NN} \end{bmatrix} = \begin{bmatrix} 0 \\ ... \\ 1 \end{bmatrix} \end{cases} \tag{5.42}$$

$$\boldsymbol{Cx} = \boldsymbol{b} \tag{5.43}$$

$$\boldsymbol{C} = \boldsymbol{L}\boldsymbol{L}^T \tag{5.44}$$

$$\boldsymbol{L} = \begin{bmatrix} L_{11} & ... & 0 \\ ... & & ... \\ L_{N1} & ... & L_{NN} \end{bmatrix} \tag{5.45}$$

$$\boldsymbol{Cx} = \boldsymbol{b} \Rightarrow \begin{cases} \boldsymbol{Ly} = \boldsymbol{b} \\ \boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{y} \end{cases} \tag{5.46}$$

$$\begin{cases} L_{11} = \sqrt{C_{11}} \\ L_{ij} = \sqrt{C_{ij} - \sum\limits_{k=1}^{j-1} L_{jk}^2},\ i = j \\ L_{i1} = \frac{C_{i1}}{|L_{11}|} \\ L_{ij} = \frac{C_{ij} - \sum\limits_{k=1}^{j-1} L_{ik}L_{jk}}{|L_{jj}|},\ i > j \\ L_{ij} = 0,\ i < j \end{cases} \quad , \ i, j = 1, ..., N \tag{5.47}$$

$$\begin{cases} y_1 = \frac{b_1}{|L_{11}|} \\ y_i = \frac{b_i - \sum_{j=1}^{i-1} L_{ij} y_j}{|L_{ii}|}, \; i = 2, ..., N \end{cases} \tag{5.48}$$

$$\begin{cases} x_N = \frac{y_N}{|L_{NN}|} \\ x_i = \frac{y_i - \sum_{j=i+1}^{N_D} L_{ij} x_j}{|L_{ii}|}, \; i = N-1, ..., 1 \end{cases} \tag{5.49}$$

Such computational algorithm of the matrix inversion allows finding stable inverse of any large square symmetric matrix. This allows effective conditioning unconditional simulation to a large number of data.

5.2.4 Semi-Positive Definiteness Check

An important aspect of the multivariate modeling is that the structure of the covariance functions in the form of LMC should be checked for its plausibility. The main requirement is that the sill contribution of each structure (random factor $X_p, p = 0, ..., P$) in the LMC should be semi-positive definite. This means that the determinant of each sill contribution matrix $\boldsymbol{B}_p, p = 0, ..., P$, should be equal to or larger than zero (Eqs. (2.51) – (2.52) of Chapter 2). The size of any matrix $\boldsymbol{B}_p$ is $K \times K$, where K is the number of simulated variables. Because all sill contribution matrices are symmetric, they can be decomposed by Cholesky decomposition into triangular matrices as shown in Eq. (5.50). The determinant of a lower triangular matrix $\boldsymbol{L}_p$ equals to the product of its diagonal elements as presented in Eq. (5.51) [3, 62]. Diagonal elements of the lower triangular matrix can be computed from the corresponding sill contribution matrix as it explained in the previous section. It happens that the determinant of the product of two matrices is equal to the product of the determinants of these two matrices as shown in Eq. (5.52). Thus, the determinant of the matrix $\boldsymbol{B}_p$ is computed as shown in Eq. (5.53). Finally, for the LMC covariance/variogram model to be

licit, all determinants $det\{\boldsymbol{B}_p\}, p = 0, ..., P$, of sill contribution matrices should conform to the inequality in Eq. (5.53). Same requirement applies to the diagonal elements $b_{p,kk}$ of the matrices $\boldsymbol{B}_p, p = 0, ..., P$.

$$\boldsymbol{B}_p = \boldsymbol{L}_p \boldsymbol{L}_p^T, \ p = 0, ..., P \tag{5.50}$$

$$det\{\boldsymbol{L}_p\} = \prod_{k=1}^{K} L_{p,kk}, \ p = 0, ..., P \tag{5.51}$$

$$det\{\boldsymbol{B}_p\} = det\{\boldsymbol{L}_p \boldsymbol{L}_p^T\} = det\{\boldsymbol{L}_p\} det\{\boldsymbol{L}_p^T\}, \ p = 0, ..., P \tag{5.52}$$

$$\begin{cases} det\{\boldsymbol{B}_p\} = \prod\limits_{k=1}^{K} (L_{p,kk})^2 \geq 0 \\ b_{p,kk} \geq 0, \ k = 1, ..., K \end{cases}, \ p = 0, ..., P \tag{5.53}$$

Therefore, the semi-positive definiteness of the LMC variogram model can be checked according to the determinant and diagonal elements of the sill contribution matrices.

5.2.5 Computational Cost Reduction

Computational cost reduction can be achieved through matrix manipulation and parallel programming. Some matrix manipulations for improved computational time, such as kriging implementation in the dual form and block matrix inversion, have been explained in the previous chapters. Parallel programming would allow using several processors to perform simulation at several simulation nodes and/or of several realizations simultaneously [57]. In this case, the reduction in time would be achieved by utilizing more hardware power, but not by the optimization of the code.

Another interesting aspect for computational cost reduction is to implement the GFS algorithm in two separate programs. This would allow saving time by setting up the Fourier coefficients, defining covariance matrices, computing matrix inverse, performing PSB value interpolation, etc., in a first program only once, and using already processed data in the second program for the simulation. The concept of two-level code is depicted in Fig. 5.16. The first program would process raw data and make some preliminary calculations, while second program would simulate values of required realizations of specified variables at target locations. By doing so, the computational time taken by first program will not be wasted next time, when simulation is performed over and over again. These two programs are called `gfsim1` and `gfsim2`, parameter files of which are described in the Appendix. A one-level grid-free simulation program `gfsim` is described in the Appendix as well.

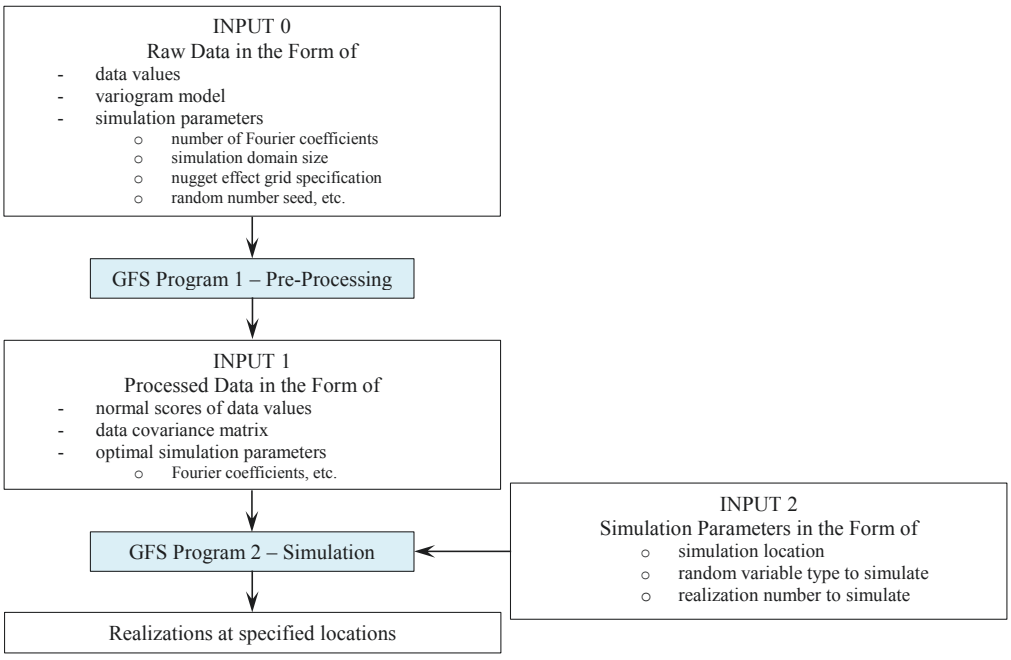


Figure 5.16: Conceptual paradigm of two-level grid-free simulation program.

5.3 Choice of Grid-Free Simulation Parameters

This section is devoted to selection of the proper grid-free simulation parameters: number of the Fourier coefficients required for the accurate 1-D covariance decomposition for turning line processes, and number of the turning lines required to get simulation without artifacts for the 2-D and 3-D models.

5.3.1 Number of Fourier Coefficients

Number of the Fourier coefficients required to properly represent a 1-D covariance function, which is used to simulate 1-D processes on turning lines, is recommended based on a 2-D case study. The number of coefficients is selected to approximate the 1-D covariance with low error, which is computed as a ratio of the smallest Fourier coefficient to the largest and is expressed in percent. Ideally, when infinitely large number of the coefficients is used, this ratio would be zero. In practice, a limited number of coefficients is used to reduce computational time. The errors of 10%, 5%, 1%, and 0.5% are selected to generate realizations with the GFS on a regular 100×100 grid with 1.0×1.0 m^2 grid node spacing. A total of 50 unconditional realizations are generated with an exponential variogram model shown in Eq. (5.54) and a total number of 100 turning lines. The corresponding numbers of Fourier coefficients that lead to such errors are 61, 81, 161, and 241, respectively. The number of the Fourier coefficients are selected with an increment of 20 starting with 21. The associated computational times to generate all 50 realizations are 172, 232, 450, and 659 seconds, respectively. The computations are performed on a 64-bit Windows machine with Intel i7 processor of 2.8 GHz and 24.0 GB RAM. When the proportions of the number of Fourier coefficients (1.00 : 1.33 : 2.67: 4.00) are compared with the proportions of the simulation time (1.00 : 1.34 : 2.61 : 3.89), the direct relationship between them can be seen as expected. The maps of the first realizations for various

error thresholds are presented in Fig. 5.17. When the threshold is large such as 10% or 5%, the spatial structure of the realizations looks more continuous than exponential. An error threshold of 1% and lower produces realizations with the expected spatial features. Note that because of the different numbers of the Fourier coefficients used in the grid-free simulation, there are a different number of random terms, resulting in realizations that look different. Fig. 5.18 presents the histograms of all 50 realizations, which look normal with distribution parameters close to the target zero mean and unit variance for all cases. Mean and variance maps of the realizations are shown in Fig. 5.19. A small number of Fourier coefficients leads to too much continuity. The reproduction of the variogram model by the experimental variograms computed from the realizations is presented in Fig. 5.20 for directions X (east) and Y (north). The exponential variogram model is shown by a red line. The individual experimental variograms are presented by green lines with their averages in blue. The target variogram model is reproduced well at longer ranges for all error thresholds. However, the variogram is not reproduced at the very short scales [0.0 – 2.0 m] for realizations with larger decomposition errors such as 10% and 5%. On the other hand, the variogram model is reproduced accurately for short ranges for small decomposition errors of 1% and 0.5%. Medium and long ranges are reproduced similarly in all four cases, with 10% producing the worst fit overall and 1% and 0.5% producing the best fit overall.

$$C_Y(\boldsymbol{h}) = Exp_{r=20.0}(\boldsymbol{h}) \tag{5.54}$$

To sum up, in order to get a simulation with target variogram model and avoid poor variogram reproduction at short ranges, a 1-D covariance functions should be decomposed to the 1% error threshold or better. The corresponding number of the Fourier coefficients would differ depending on the covariance

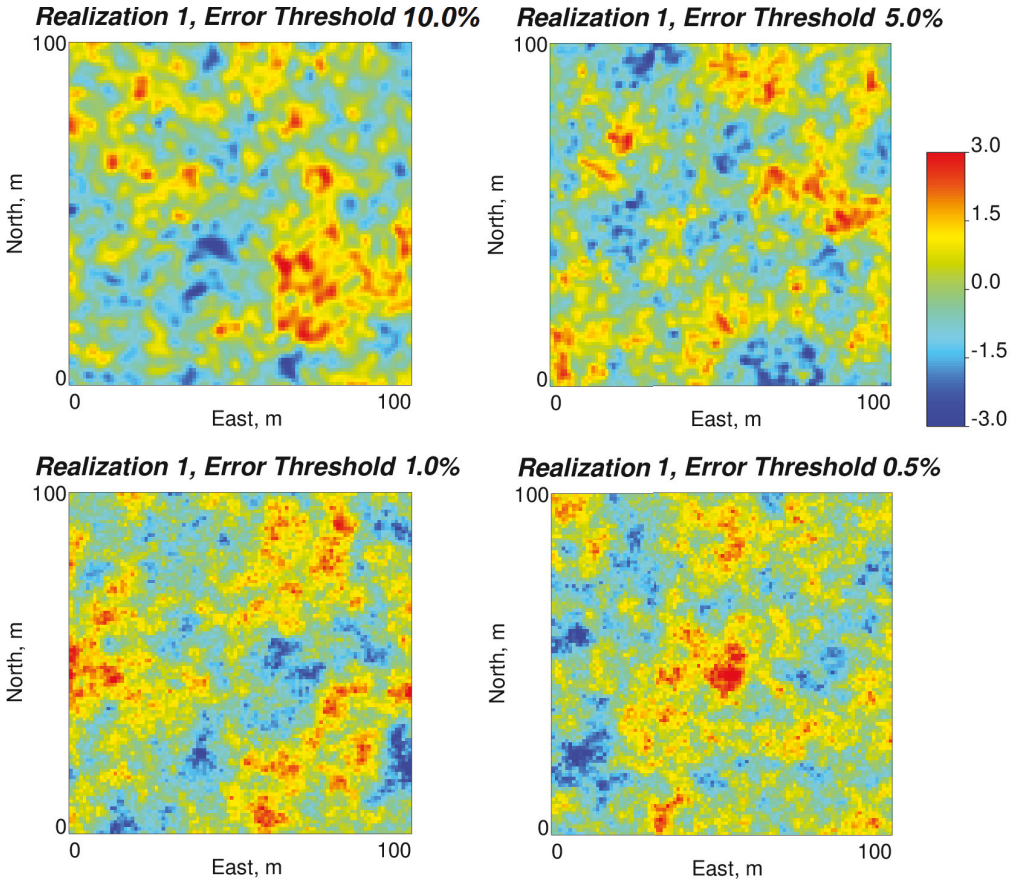


Figure 5.17: A 2-D case study to define proper number of the Fourier coefficients – realization maps with different precision degrees of covariance decomposition.

function form used as a target spatial structure model, the ratio of the variogram ranges used in the variogram model, and size of the periodic domain for target covariance decomposition in comparison with the variogram ranges. The `gfsim` code allows users to choose a degree of the error threshold in the grid-free simulation.

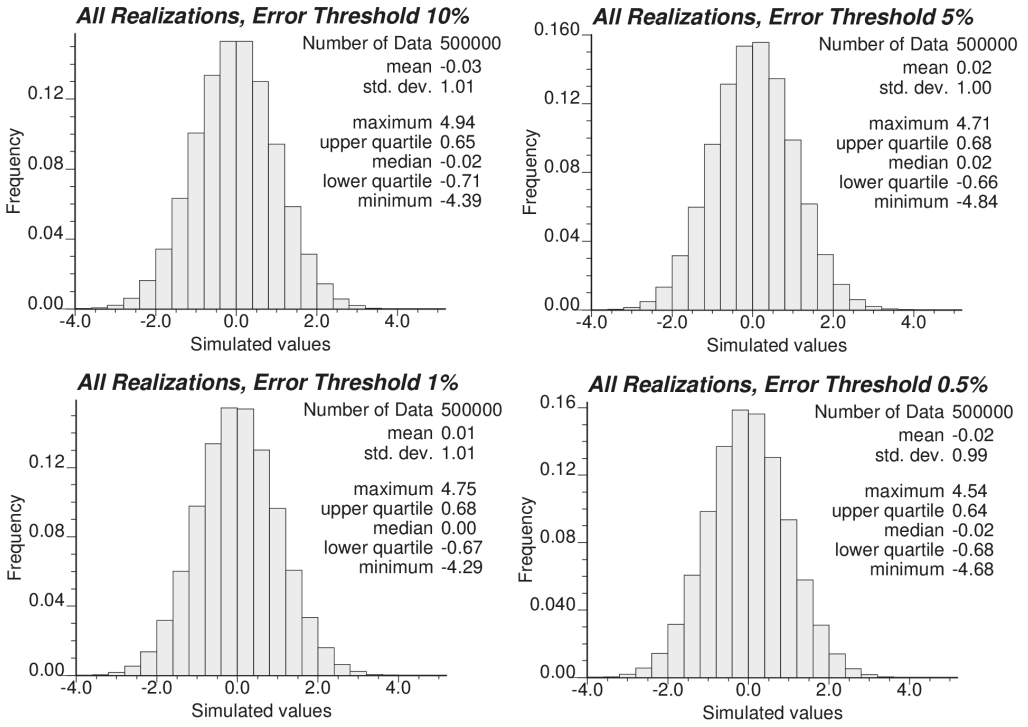


Figure 5.18: A 2-D case study to define proper number of the Fourier coefficients – histograms of all realizations with different precision degrees of covariance decomposition.

5.3.2 Number of Turning Line Processes

Two case studies are examined on 2-D and 3-D systems to choose the proper number of the turning lines to be used in the grid-free simulation. The turning lines should be placed evenly in the space and originate from one point to avoid any artifacts in the simulation. A direction of a turning line in the 2-D space is defined from the centre of the circle of unit radius to its perimeter. A direction of a turning line in the 3-D space is determined from the centre of the sphere of unit radius to its surface. Centers of both unit circle and unit sphere are placed at the coordinates system origin. The procedure of the turning lines placement has been described in Section 3.2 of Chapter 3. The number of evenly placed

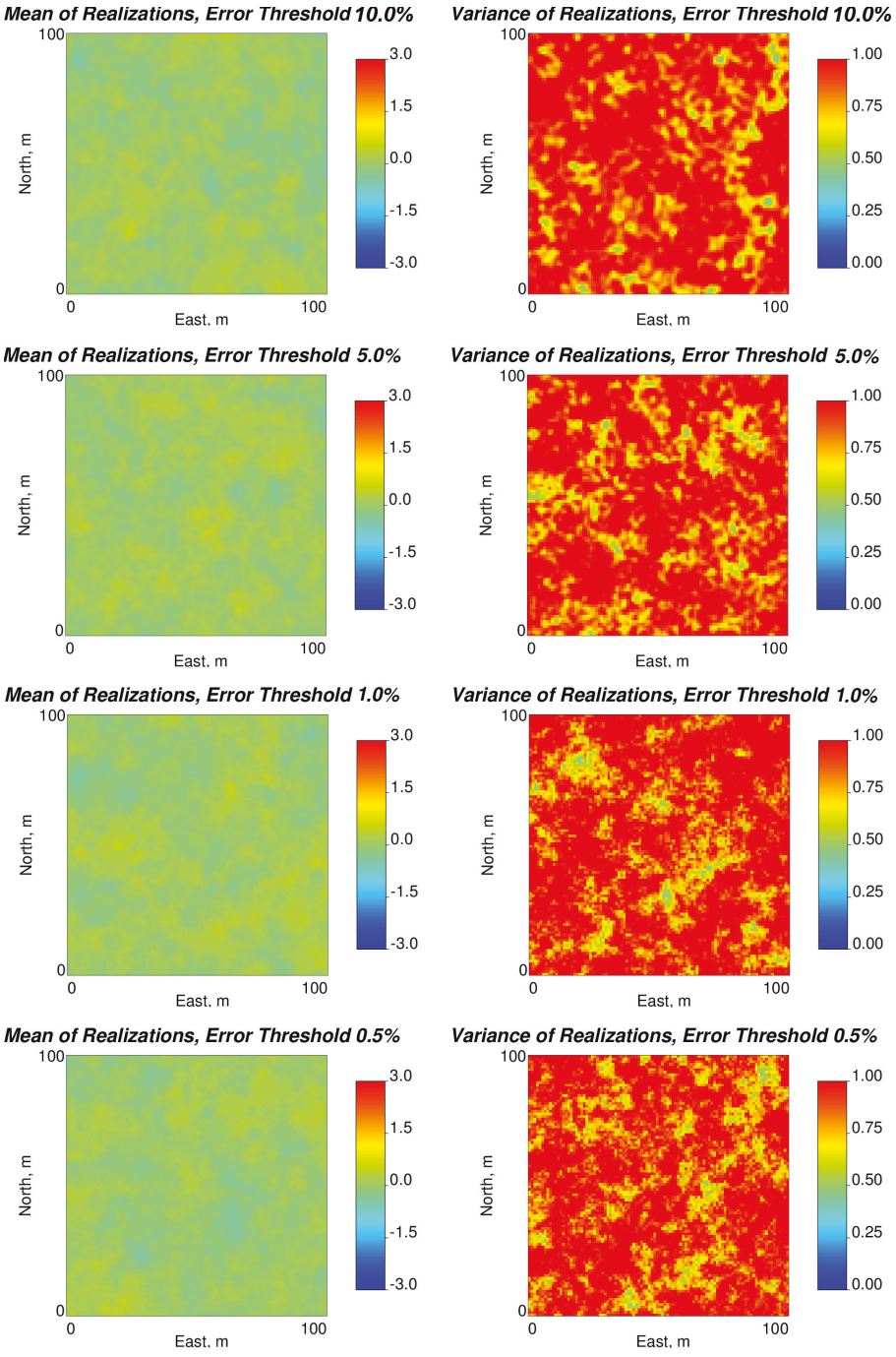


Figure 5.19: A 2-D case study to define proper number of the Fourier coefficients – mean and variance maps of realizations with different precision degrees of covariance decomposition.

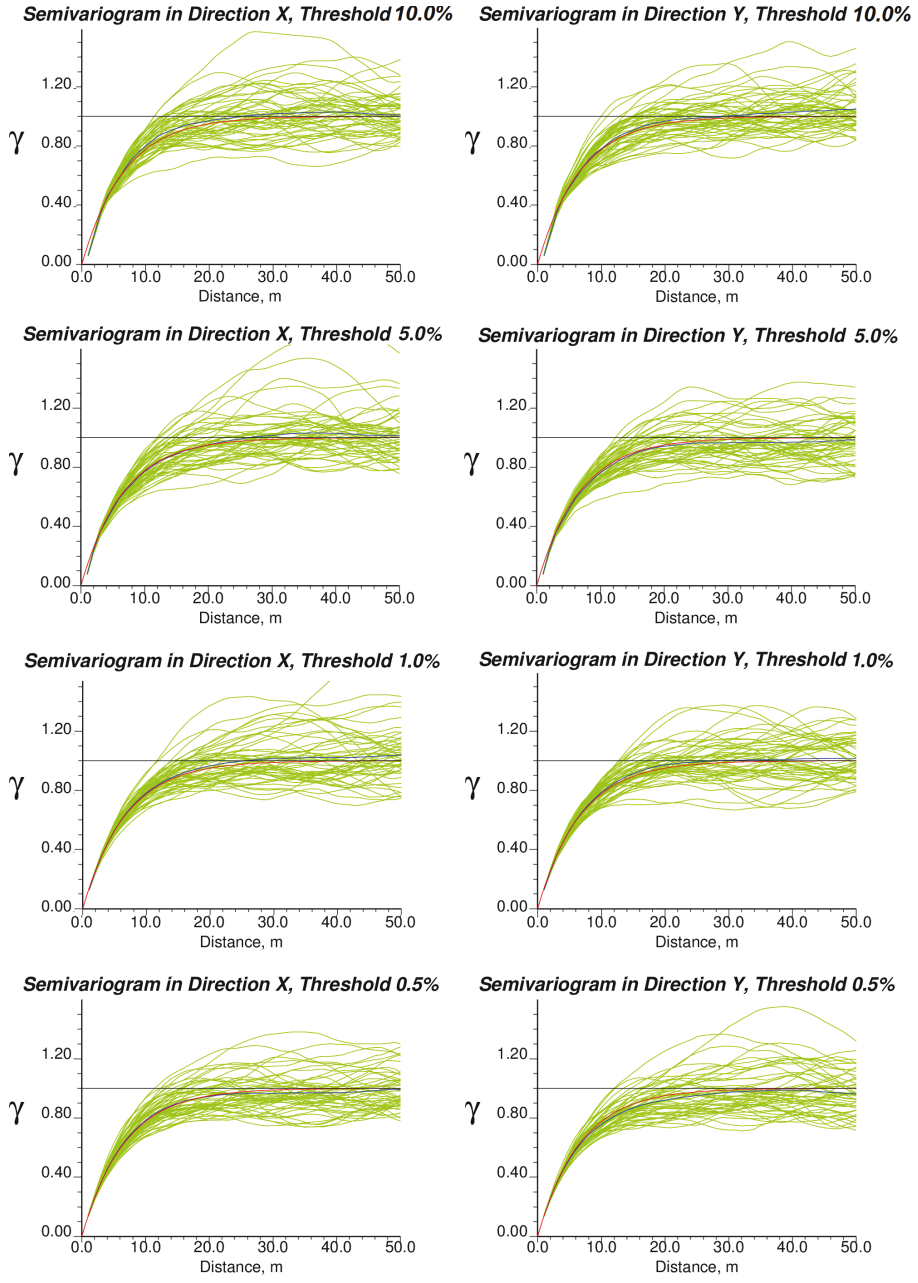


Figure 5.20: A 2-D case study to define proper number of the Fourier coefficients – variograms in X and Y directions for realizations with different precision degrees of covariance decomposition. Individual experimental variograms are shown by green lines, their average is presented by a blue line, and variogram model is a red line.

lines that are tested in the simulation are 10, 50, and 100 for 2-D models and 12, 42, and 162 for 3-D models. The optimal number of the lines is defined for 2-D and 3-D grid-free simulation based on the examination of 50 unconditional realizations generated with the exponential covariance functions.

The 2-D model is generated on a 100 $\times$ 100 grid with 1.0 $\times$ 1.0 m^2 grid node spacing. Target variogram model is presented by an isotropic covariance function in Eq. (5.55). The computational times for 10, 50, and 100 turning lines are 44, 211, and 427 seconds, respectively, which is proportional to the number of the lines used in the simulation. The computations are performed on a 64-bit Windows machine with Intel i7 processor of 2.8 GHz and 24.0 GB RAM. Maps of the first realizations for each case along with the histograms of all realizations are shown in Fig. 5.21. The generated realizations look normal for all cases. There are some artifacts in the form of the lines in the realization produced by 10 turning lines. No artifacts are observed on the other realizations. Fig. 5.22 presents the variogram reproduction, the experimental variograms computed from individual realizations are shown by green lines, their averages over the realizations are shown by blue lines, and the variogram model is presented by a red line. The variogram is approximately reproduced for 10 turning lines. The average experimental variogram is lower than the expected variogram sill. Variogram reproduction for 50 lines is very close to the target, and average of experimental variograms for 100 turning lines very accurately describes the variogram model. Therefore, it is recommended to use at least 50 turning lines for simulation of 2-D models with the GFS to avoid artifacts and get plausible target variogram reproduction. More turning lines would lead to better statistics reproduction with the downside of increased computational time.

$$C_Y(\boldsymbol{h}) = Exp_{r=20.0}(\boldsymbol{h}) \tag{5.55}$$

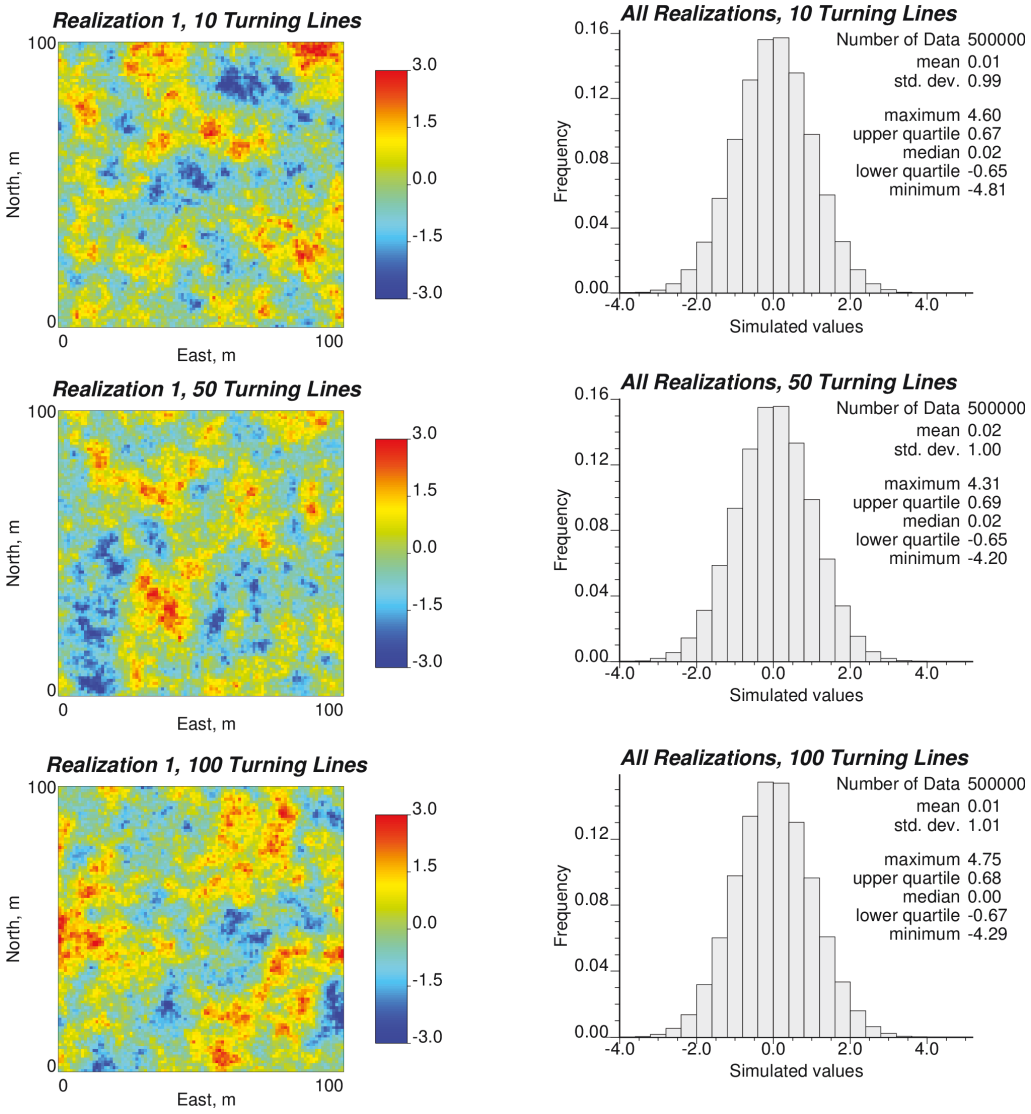


Figure 5.21: A case study to define proper number of the turning lines in 2-D – realization maps and histograms with different numbers of the turning line processes.

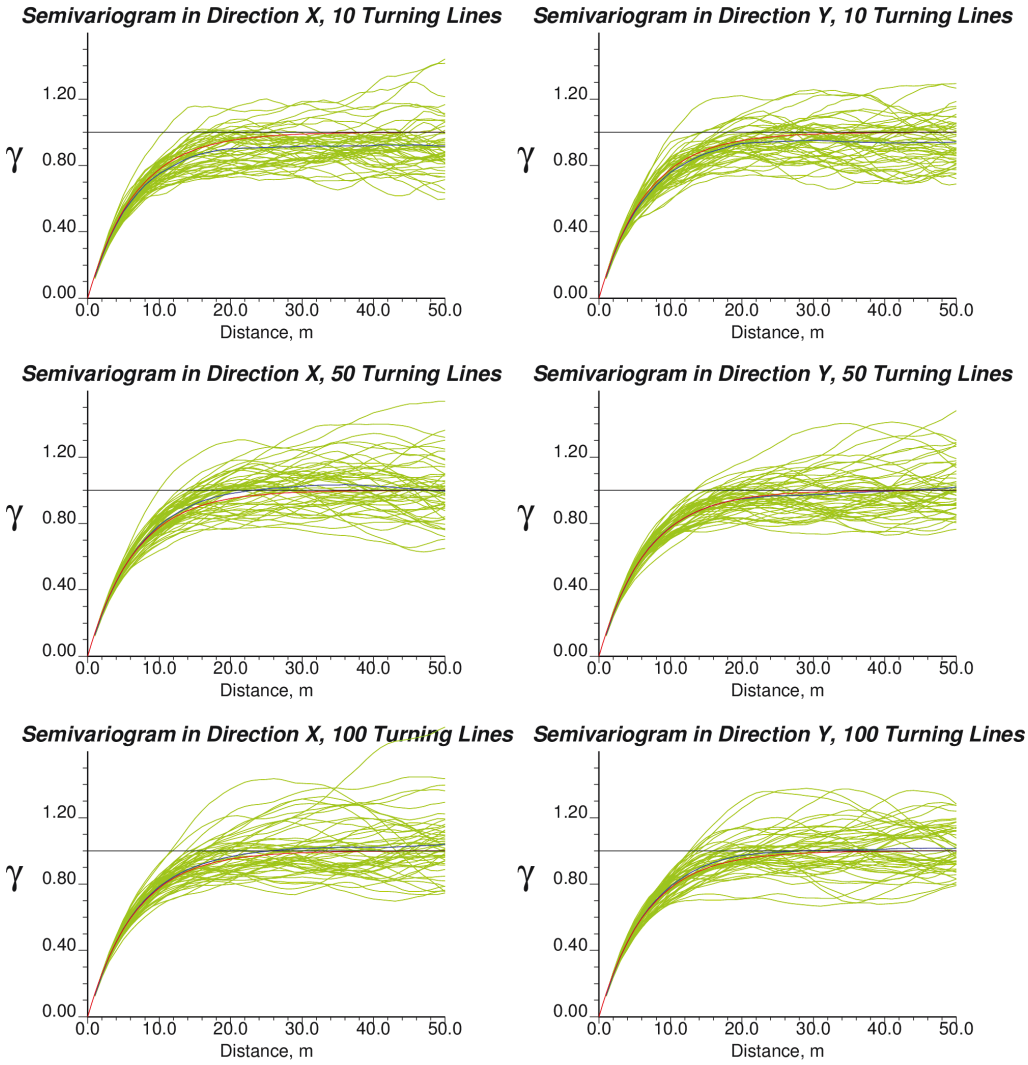


Figure 5.22: A case study to define proper number of the turning lines in 2-D – variograms in X and Y directions for realizations with different numbers of the turning line processes. Individual experimental variograms are shown by green lines, their average is presented by a blue line, and variogram model is a red line.

The 3-D model is simulated on a 100 × 100 × 50 grid with 1.0 × 1.0 × 0.1 m^3 grid node spacing. The target spatial structure used in the simulation is presented by an anisotropic covariance function in Eq. (5.56). The variogram range in the vertical direction differs from the variogram ranges on the horizontal plane. The number of turning lines used in the simulation are 12, 42, and 162, respectively. The computational time is about 2.5 hours, 10.0 hours, and 36.0 hours. Again, the proportions of the computational time are directly related to the number of the turning lines used in the simulation. A total of 50 realizations are generated for each case. The first realizations are shown in Figs. 5.23 – Fig. 5.25 in a 3-D and cross-sectional views. Obvious line artifacts are seen on the realization maps generated with 12 and 42 lines. The realization generated with 162 turning lines look artifact-free. Variogram reproduction for the three cases is depicted in Fig. 5.26. Variogram reproduction in the form of the average experimental variogram in blue of the individual variograms in green computed from each realization improves with an increasing number of turning lines, as well as the spread of the individual realizations becomes smaller. The variogram model is shown by red line.

$$C_Y(\boldsymbol{h}) = Exp_{\substack{r_1=20.0 \\ r_2=20.0 \\ r_3=1.0}}(\boldsymbol{h}) \tag{5.56}$$

Therefore, it is recommended to use at least 50 turning lines to simulate 2-D models and 162 turning lines to simulate 3-D models with exponential covariance spatial structure in a grid-free form to avoid artifacts in the realizations and to get proper variogram reproduction. Simulation models generated with spherical covariance function in 3-D space as in the 3-D example of Section 4.1.1 of Chapter 4 reproduce the imposed variogram model better with the same recommended simulation parameters discussed here. Therefore, the recommendation about the optimal number of the turning lines may change depending on the form of the

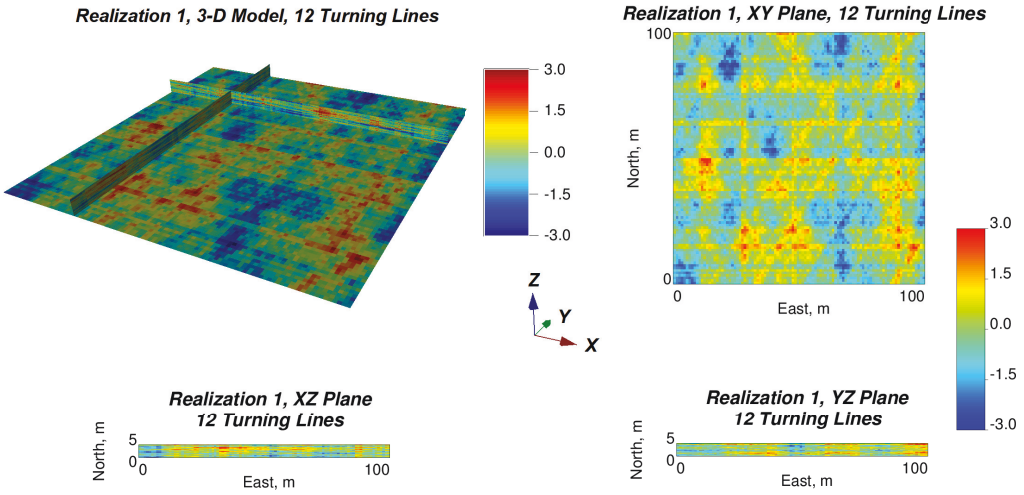


Figure 5.23: A case study to define proper number of the turning lines in 3-D – realization maps constructed with 12 lines.

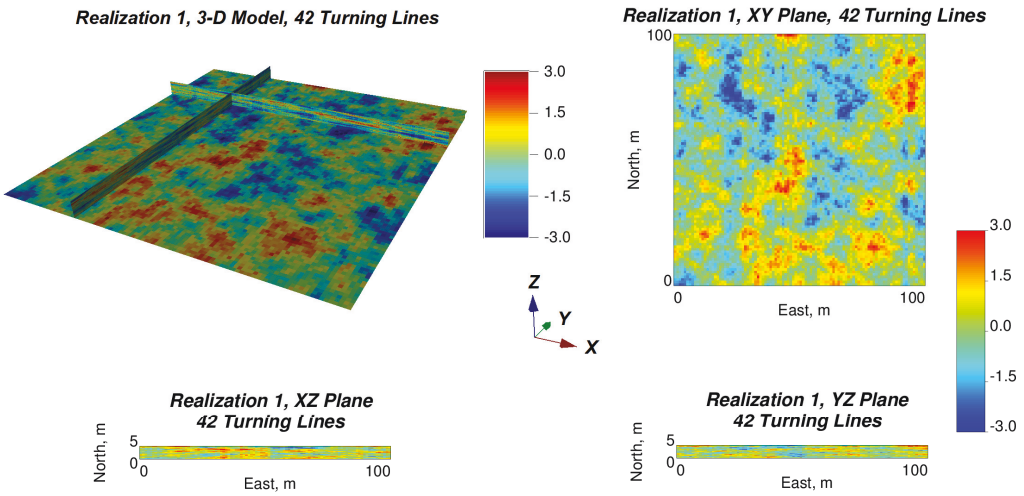


Figure 5.24: A case study to define proper number of the turning lines in 3-D – realization maps constructed with 42 lines.

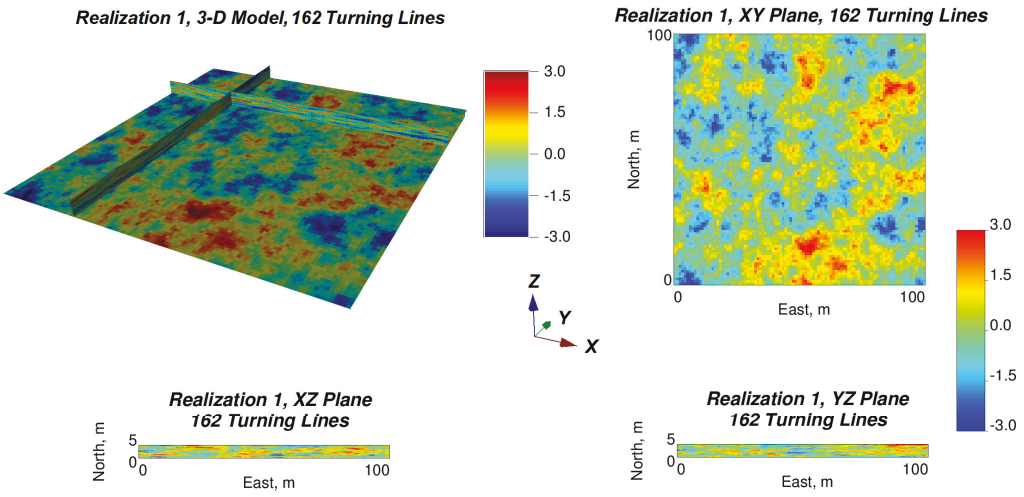


Figure 5.25: A case study to define proper number of the turning lines in 3-D – realization maps constructed with 162 lines.

input variogram model.

5.4 Validation of Grid-Free Simulation

The realizations generated by the proposed grid-free simulation algorithm are validated to conform to the criteria for plausible geostatistical simulation. Also, the GFS is compared with other conventional simulation algorithms on a small case study to define its advantages and highlight possible shortcomings.

5.4.1 Verification of Grid-Free Simulation Algorithm

The realizations generated by the GFS algorithm should be fair samples from the multivariate Gaussian spatial distribution parameterized by the transformation to Gaussian units and the covariance function in Gaussian units. Checking this is difficult, but there are four main criteria that simulated conditional realizations have to comply with to represent underlying system plausibly [43]:

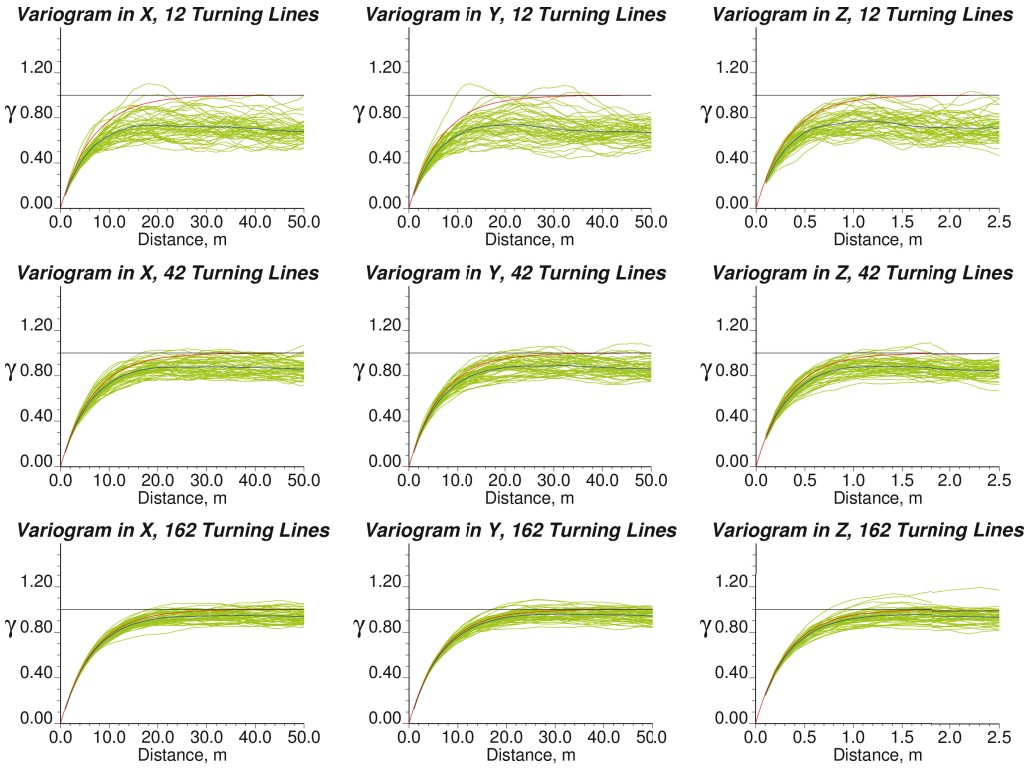


Figure 5.26: A case study to define proper number of the turning lines in 3-D – variograms in X, Y, and Z directions for realizations constructed with different numbers of the turning line processes. Individual experimental variograms are shown as green lines, their average is presented by a blue line, and variogram model is a red line.

1. the simulated values at the data locations should honour the data values

2. the distribution of the simulated values on average should be the same as distribution of the data

3. the distribution parameters such as mean and variance of the simulated values on average should match the distribution parameters of the data

4. the average experimental variogram computed from individual unconditional realizations should reproduce target variogram model in normal scores, and the average experimental variogram computed from individual

conditional realizations should be a combination of the target variogram model and experimental variogram computed from data in normal scores

A 2-D example is assembled in original units to demonstrate that simulation generated by the GFS meets all four criteria. The simulation is generated on a 100×100 regular grid with 1.0×1.0 m^2 node spacing. The spatial data distribution is shown in Fig. 5.27 along with the polygons used for the polygonal declustering of the data values and a scatter plot between the true data values versus the simulated values at the data locations for the first realization. There are 55 data values. The simulation is performed in the normal units. The transformation table-based NST procedure is implemented. The spatial structure used in the simulation of the normal scores is presented by a spherical isotropic covariance function in Eq. (5.57). The declustering is performed to get unbiased data statistics for the normal score transformation of the data to normal scores. It can be seen from the scatter plot that the simulated values at the data locations from the first realization produced by the GFS honor the data values. All the points are on a 45° line on the scatter plot of Fig. 5.27. The histograms of the data, simulated values in original units, and normal scores are presented in Fig. 5.28 with summary statistics. By comparing the data distribution with the distribution of simulated values from all 50 realizations in original units, it can be claimed that the second criteria of simulation goodness is also satisfied. Note that the simulated values in normal units look normal with statistical parameters close to the target zero mean and unit variance. The distribution parameters in original units of the simulated values are very close to the data parameters, and, thus, third criterion is met. Maps of the first realization in original units and normal scores, and of the mean and variance of the realizations are presented in Fig. 5.29. The data are honored. The mean map looks similar to map of kriged estimates as it should. The variance that describes the local uncertainty is zero at data locations and increases as the data spacing becomes larger. No

obvious visual artifacts are present in the realizations. Fig. 5.30 demonstrates reproduction of the target variogram model in red by experimental variograms in green computed from individual realizations and their average in blue. The variogram is reproduced quite well before it reaches the sill. Some slight deviations of the average experimental variogram from the model in the form of the cyclicity are seen for ranges longer than the spatial correlation range. As a result of the variogram reproduction, all presented four criteria for plausible geostatistical simulation are met. In addition to this example, numerous case studies have been shown in Chapter 4 to empirically check that the grid-free simulation algorithm generates plausible realizations of the described reality in normal units.

$$C_Y(\boldsymbol{h}) = Sph_{r=20.0}(\boldsymbol{h}) \tag{5.57}$$

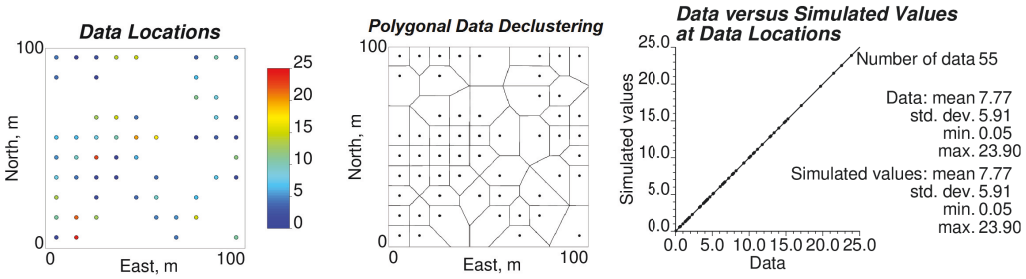


Figure 5.27: A 2-D case study to show plausible nature of the realizations generated by the GFS – data location map, polygonal data declustering, and data reproduction by the simulation.

Note that the data reproduction in the GFS is ensured, because the conditioning is performed with kriging, which is an unbiased minimum error variance estimation method [10, 17, 26, 35, 36]. Variogram reproduction is ensured by accurate Fourier series-based decomposition of the covariance function. Cyclicity can be present in the experimental variograms, when the periodicity domain

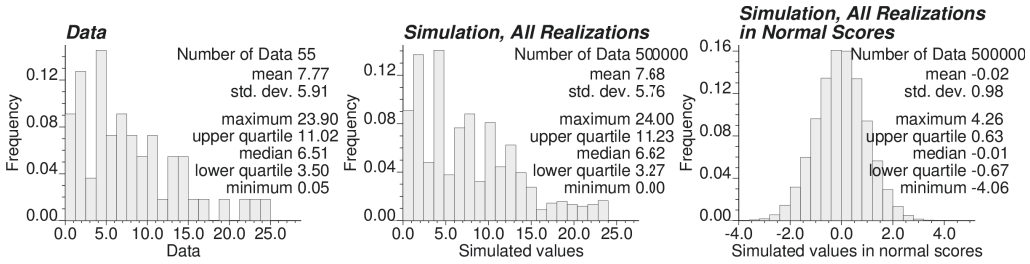


Figure 5.28: A 2-D case study to show plausible nature of the realizations generated by the GFS – histograms of data values and simulated values in original units and normal scores.

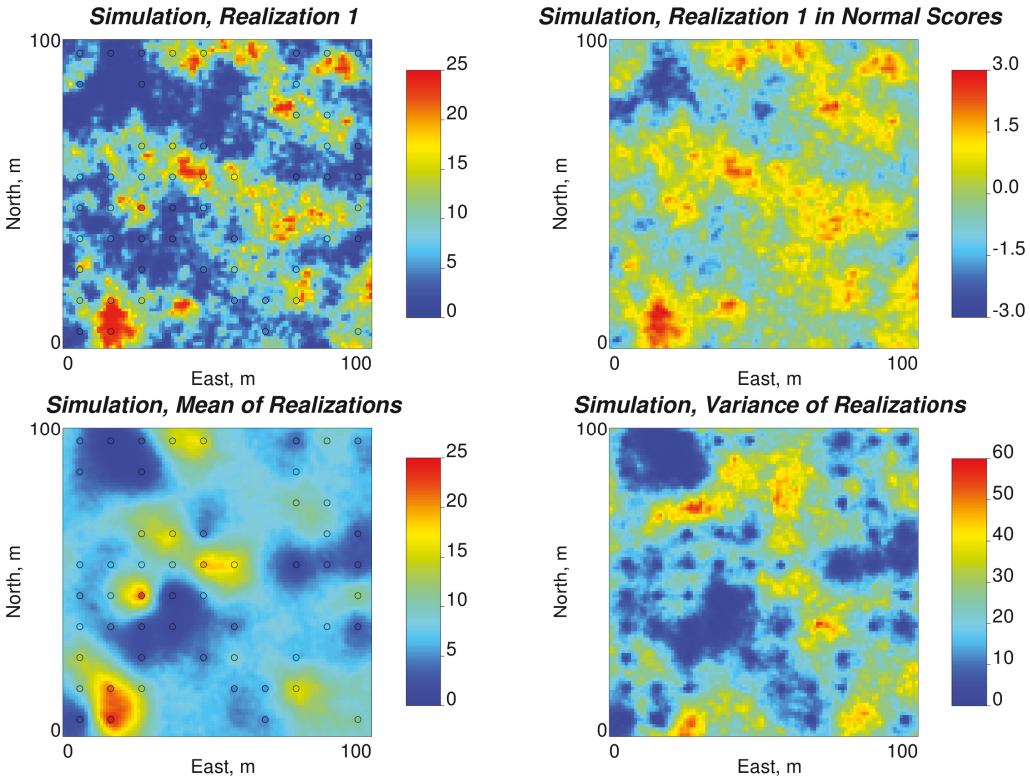


Figure 5.29: A 2-D case study to show plausible nature of the realizations generated by the GFS – map of first realization in original units and normal scores, and maps of the realization mean and variance.

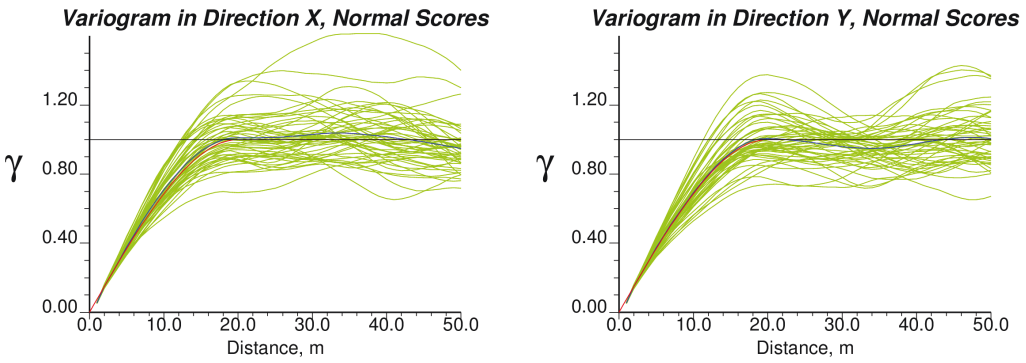


Figure 5.30: A 2-D case study to show plausible nature of the realizations generated by the GFS – variograms in X and Y directions for grid-free realizations. Individual experimental variograms are shown by green lines, their average is presented by a blue line, and variogram model is a red line.

selected to decompose the covariance function is relatively small in comparison with the simulation domain, or when the simulation domain is not large enough in comparison with the variogram range. Also, the multivariate relationship in systems with multiple variables should be honored. According to the minimum acceptance criteria for geostatistical simulation in [43], simple 3-D/2-D visualization of the model is a simple, yet effective check of the plausibility of the realizations. Simulated realizations could be also checked against geological interpretation of the studied area. Cross-validation and jack-knife can be used to check quality of the variogram model. Any small deviations can be claimed to be caused by ergodic fluctuations caused by a small simulation domain in comparison with the variogram range, limited random numbers used in the simulation, and precision errors in the computation. Additional primary or secondary data may be used to check the closeness of the simulation to the reality.

5.4.2 Comparison of Simulation Methods

The unconditional part of the GFS algorithm is compared with the widely used SGS and LUS on a small 2-D example with Gaussian variogram model in normal

units. A total of 50 realizations are generated by each method on a regular 100 $\times$ 100 grid with 1.0 $\times$ 1.0 m^2 grid spacing with isotropic Gaussian spatial structure presented in Eq. (5.58). The maps of the first realizations produced by each technique along with the histogram of all realizations are shown in Fig. 5.31. All methods generate realizations with Gaussian spatial features. The histograms look normal with statistics close to the target zero mean and unit variance. The variogram reproduction is presented in Fig. 5.32 for X and Y directions. Green lines represent experimental variograms computed from the individual realizations with their average shown in blue line. The variogram model is in red. All techniques reproduce the target variogram accurately. There are some slight deviations, which are believed to be due to ergodicity. The spread between the individual experimental variograms is about the same for the GFS and SGS, but larger for the LUS. Experimental variograms computed from the first realizations of each simulation method are compared in Fig. 5.33. The averages of the experimental variograms are presented in the same figure. The variogram model is shown in red. The green line corresponds to the experimental variogram computed from the GFS simulation, orange – from the SGS, and purple – from the LUS. There are significant deviations in the individual experimental variograms from the target model. The average of the realizations precisely follows the variogram model for each method. As a result of this comparison study, it is deemed that all three methods produce unconditional simulation of similar quality. The computational time for 50 realizations generated by GFS with 100 lines and 41 Fourier coefficients is 116 seconds. The computational time required to generate 50 realizations with SGS is 31 seconds. The computational time for 50 realizations generated by LUS is 2340 seconds. The computations are performed on a 64-bit Windows machine with Intel i7 processor of 2.8 GHz and 24.0 GB RAM. The SGS algorithm is the least computationally costly. The GFS is slightly slower than SGS. The LUS method is quite slow in comparison.

$$C_Y(\boldsymbol{h}) = Gaus_{r=20.0}(\boldsymbol{h}) \tag{5.58}$$

An advantage of GFS over SGS and LUS is a grid-free form of the resulting realizations allowing consistent zooming to any grid and easiness of the simulation of the multivariate systems. Another benefit of using GFS over SGS that comes from the conditioning through kriging is that the additionally assimilated data would not change current realizations dramatically, but would change them only locally within the correlation range domain. Small drawback lies in the computational time of GFS algorithm, which is more demanding than SGS, but faster than LUS.

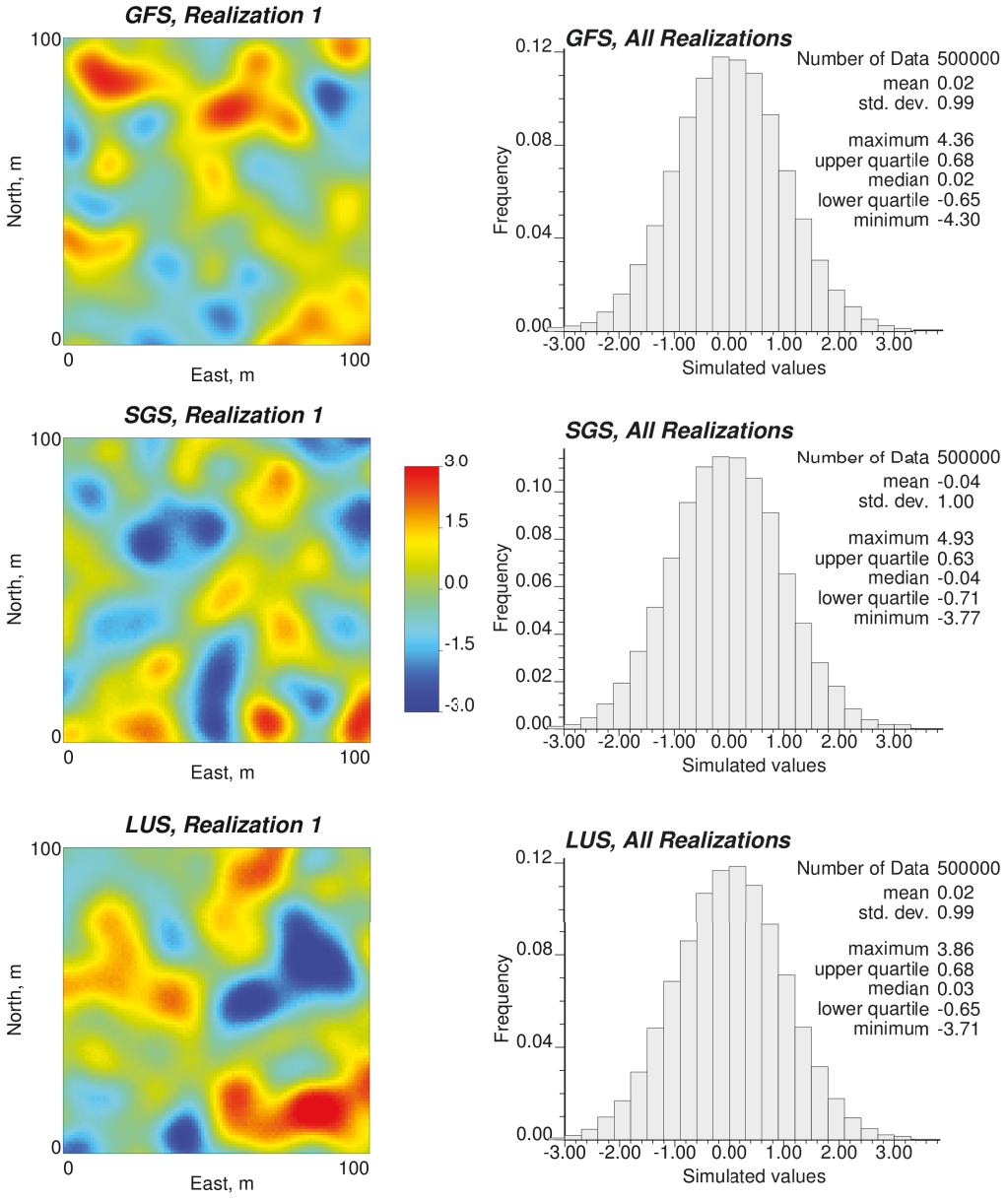


Figure 5.31: A 2-D comparison study for the GFS, SGS, and LUS – realization maps and histograms of simulated values for different simulation methods.

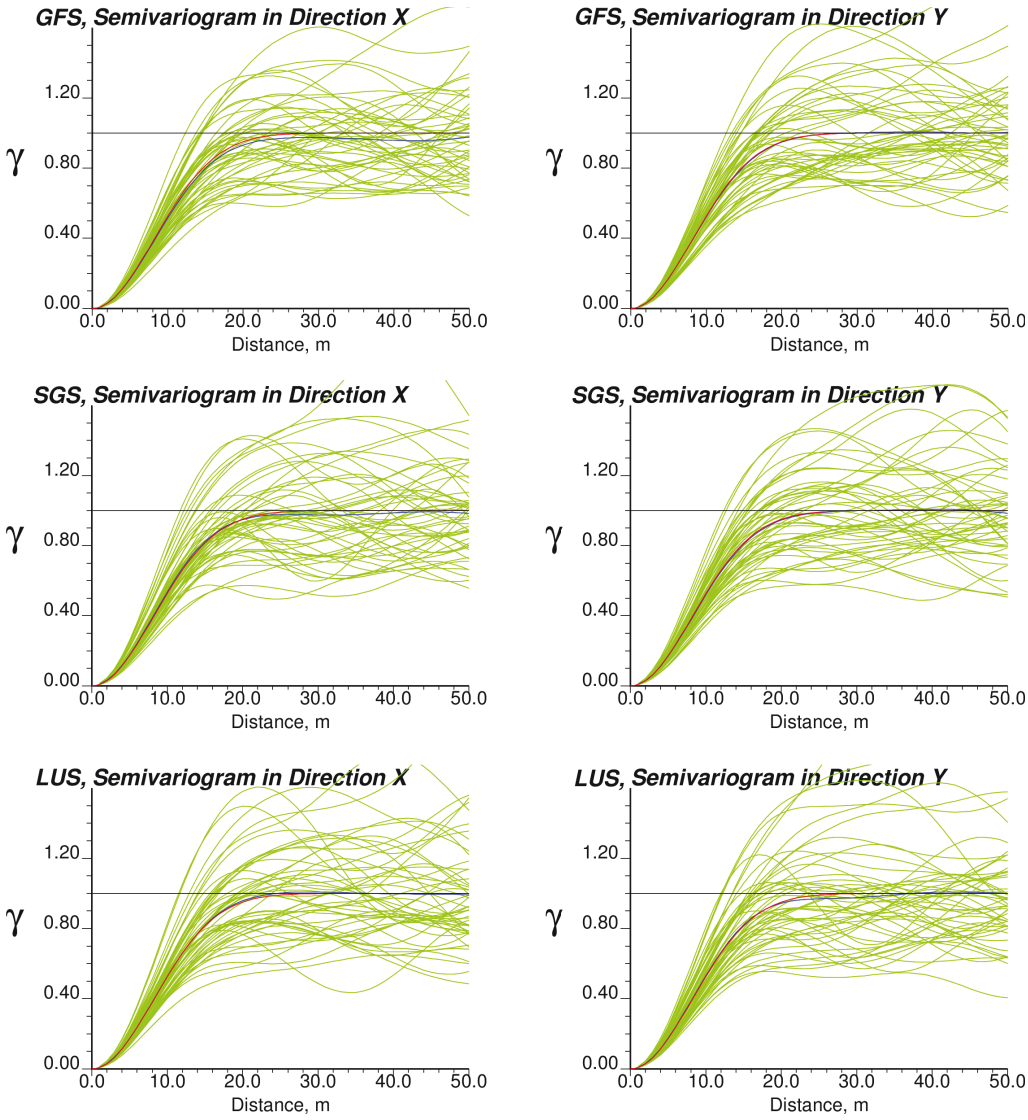


Figure 5.32: A 2-D comparison study for the GFS, SGS, and LUS – variograms in X and Y directions for realizations generated by different simulation methods. Individual experimental variograms are shown by green lines, their average is presented by a blue line, and variogram model is a red line.

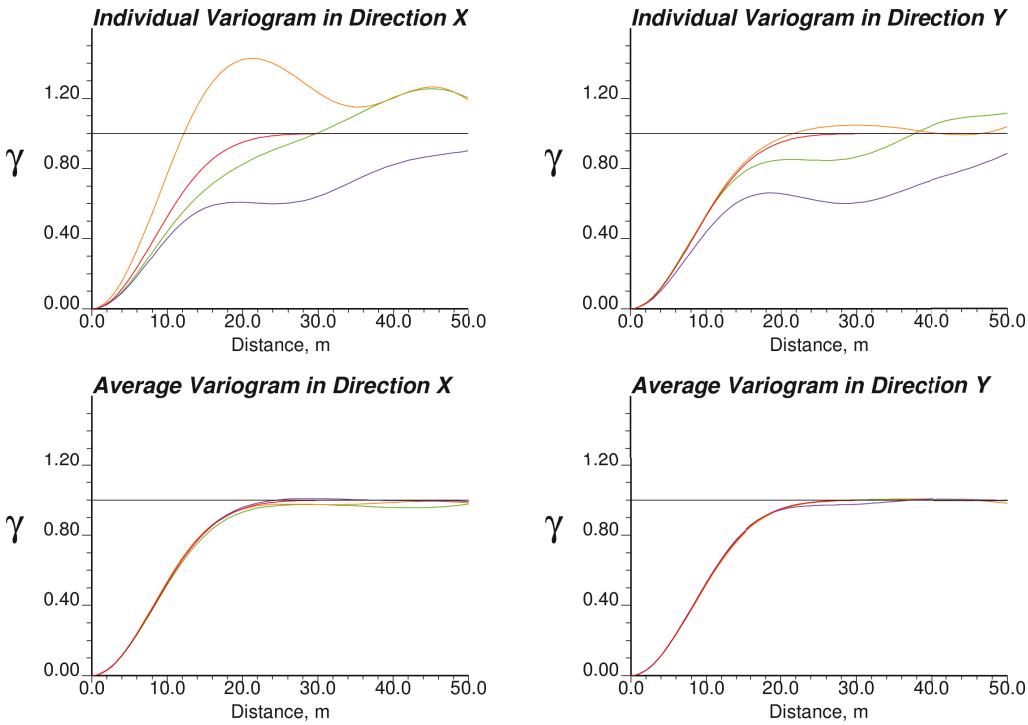


Figure 5.33: A 2-D comparison study for the GFS, SGS, and LUS – individual experimental variograms and averages of all experimental variograms in X and Y directions for realizations generated by different simulation methods. The variogram in green corresponds to the GFS, orange line is the SGS, the LUS is in purple, and variogram model is a red line.

Chapter 6

Case Study – Firebag Oil Sands Project

The developed geostatistical grid-free simulation algorithm is applied to the Firebag oil sands project located in northern Alberta for petroleum reservoir characterization. Essential petrophysical properties, such as porosity, permeability, and water saturation, are modeled by-facies at various resolutions over several domains conditional to hard point-scale core and log data and soft exhaustively sampled block-scale seismic attributes. Resulting models are consistent with each other and can be used for the oil reserve estimation and fluid flow simulation. All of the data used in this case study were taken from public records and processed as required. No data or information was provided by the operator.

6.1 Geostatistical Reservoir Characterization

The general methodology for modeling heavy oil reservoirs in a grid-free manner is explained to meet specific objectives.

6.1.1 Heavy Oil Reservoirs in Alberta

A large volume of oil sands reserves are found in northern Alberta, Canada. The steam assisted gravity drainage (SAGD) heavy oil extraction technique is widely implemented to extract the bitumen when deposits are deeper than 200.0 m below the surface [9]. A number of SAGD pads are built over the oil deposits, from which several horizontal SAGD well pairs are drilled. A SAGD well pair consists of a production well placed roughly 5.0 m below an injection well. Both wells are placed as close as possible to the reservoir base for increased oil production. Steam is injected to lower the viscosity of the accumulated bitumen and allow it to drain to the production well.

For the effective oil production, the SAGD wells should be placed in accordance with the geological setting of the reservoir. Numerical modeling of the petrophysical properties within a geological conceptual model of the reservoir is used for horizontal well placement [64]. Conventional geostatistical simulation techniques like SGS for continuous variables and SIS for categorical variables are commonly used for geomodeling and uncertainty management, where the model grid is specified beforehand [17]. Subsequent grid refinement or regridding is required, but previously simulated realizations are not necessarily reproduced with these conventional geostatistical techniques. Therefore, GFS technique can be implemented to represent a simulation as a function of the coordinates of the simulation location honoring data values, spatial structure of the system, and relationship between system's variables. The GFS has been successfully applied to several synthetic case studies in the previous chapters of this thesis. Here, the GFS method is applied to an oil sands project called Firebag, which is operated by Suncor in northern Alberta using SAGD technology [75].

6.1.2 Modeling Objectives

The objective of this realistic case study is to build three consistent petrophysical models of the Firebag subsurface geology at a point scale for oil reserve estimation at coarse resolution grid, for flow simulation around all available SAGD well pads at medium resolution grid, and for flow simulation around an individual SAGD well pad at a fine resolution conditional to point-scale core data, log data, and exhaustively sampled block-scale seismic measurements. Correlated realizations of porosity ϕ, permeability k, and water saturation s models are built for this purpose within the stratigraphic surfaces of the reservoir. The porosity defines void space in the rock, and, therefore, implies the storage capacity of the reservoir. The permeability is important, because it indicates how easily reservoir fluids can flow from the reservoir to the production wells. The water saturation is modeled to estimate hydrocarbon reserves, which are stored in the void space of the rock unoccupied by water. The water saturation is also important for thermal oil extraction processes such as SAGD, because it is a key parameter of thermal capacity of the reservoir. It is used in the computation of required energy that should be carried by heating agent (steam).

6.1.3 Modeling Procedure

A routine geomodeling workflow is presented below for petroleum reservoir characterization [64]. The workflow would change depending on the data availability and geological complexity of the reservoir. The first set of steps involves data analysis. The second set is related to simulation and post-processing including model validation.

The data analysis can be performed as follows.

1. Define reservoir stratigraphic top and base and possible compartmentalization. Core data, core pictures, and log data analysis along with seismic

surveys and a conceptual geological model can be of use. One of the surfaces is picked as a reference, relative to which the structure of the reservoir is built, that is, the vertical coordinates are adjusted to conform to the stratigraphic layers.

2. Define geological facies with distinct petrophysical properties required for the stationarity assumption in the modeling. Core data, core pictures, and log data analysis along with a conceptual geological model are used for this purpose.

3. Choose geological attributes to model and available data for the conditioning. Usually petrophysical properties such as porosity and permeability are modeled conditional to hard data (measured observations) and soft seismic attributes.

4. Group data of petrophysical properties in accordance with the facies types.

5. Clean the data and look for outliers and measurement errors, note the scale difference in the data. The core data are direct measurements of the petrophysical properties of the rock. But, the log data are indirect observations of the petrophysical properties. Therefore, because limited core data are more reliable than abundantly available log data, correct the log data with the core data for each facies, and proceed modeling with the corrected log data.

6. Transform all data values to normal scores for simulation. For multivariate data sets, consider keeping the relationship between the variables after the simulation is done. Cosimulation or decorrelation procedure are considered for dependent random variables.

7. Build variogram models of facies and all properties for the geostatistical modeling.

The simulation and model checking are carried out after the data analysis as explained below.

1. Perform simulation of geological facies, which provide the main source of the reservoir heterogeneity; object-based modeling can be an alternative to grid-free simulation of categorical variables, but it has some limitations and issues with conditioning [17, 31–33]. SIS is commonly used.

2. Simulate key petrophysical properties within corresponding facies (stationary domains) in normal space conditional to all relevant information.

3. Back transform the simulated normal scores to the original units. If required, perform upscaling to get the realizations at a block scale.

4. Validate the data reproduction, statistical and spatial properties (variogram for two-point statistics) reproduction, preservation of the variables relationship in original units, and consistency of the numerical model with the conceptual model.

5. Post-process the model to meet the study objectives, e.g. estimate reserves, evaluate economic feasibility of the area, run the model through the flow simulation, etc.

6.2 Data Analysis

The data related to the Firebag project are cleaned, analyzed, and sorted. The major parts of the data analysis are presented in this section.

6.2.1 Firebag Project Overview

The Firebag Oil Sands Project is located in the northern Alberta, about 100 km Northeast of Fort McMurray as shown in Fig. 6.1. The project is operated by

Suncor Energy [75]. The SAGD technology is used to extract the bitumen. The target reservoir is located in the McMurray Formation. The lease area covers approximately 15.0 × 15.0 km^2.

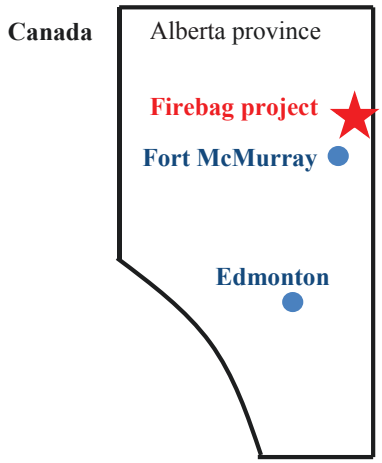


Figure 6.1: Map of the geographical location of the Firebag oil sands project, which is shown by a red star.

The stratigraphy of the lease area can be described as shown in Fig. 6.2 [75]. The major elements are:

- Muskeg Formation
- Glacial till
- Clearwater Formation, which represents the cap of the reservoir and consists mostly of shale
- McMurray Formation, which is the target reservoir with good reservoir properties. It consists of shoreface, tidal flat, channel complex, and continental parts with channel complex being the most economical feasible strata to produce from. Four facies are distinguished in the McMurray

Formation: impermeable shale, permeable sand, semipermeable inclined heterolithic strata (IHS), and permeable breccia.

- Eroded Devonian carbonate depositions below the McMurray Formation

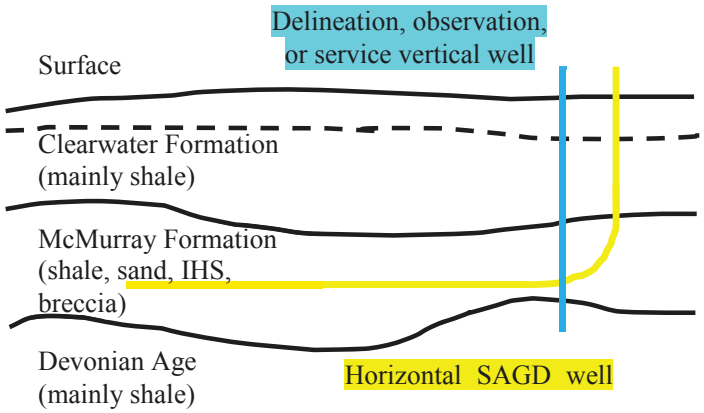


Figure 6.2: Stratigraphy of the Firebag Project area in cross-sectional view and schematic representation of wells.

The numerical data have been obtained from the `geoSCOUT` data base software in LAS format [23]. The wells within the lease area are represented by 233 SAGD well pairs (yellow), 545 delineation wells (dark blue), 180 observation wells (light blue), and 31 miscellaneous service wells (red) totaling to 1222 as shown in Figs. 6.3 and 6.4. There are 1682 wells in the Firebag region (Fig. 6.3). The topography of the study area is shown in Fig. 6.3.

There are three main sources of information available. Fig. 6.5 shows well locations with points-scale core and log data along with the block-scale seismic survey coverage. Not all wells have all data variables. The number of wells with some core data within McMurray Formation is 303: with some porosity core data – 295, with some horizontal permeability core data - 291, and with some water saturation core data – 214. The number of core data are: 19474 measurements

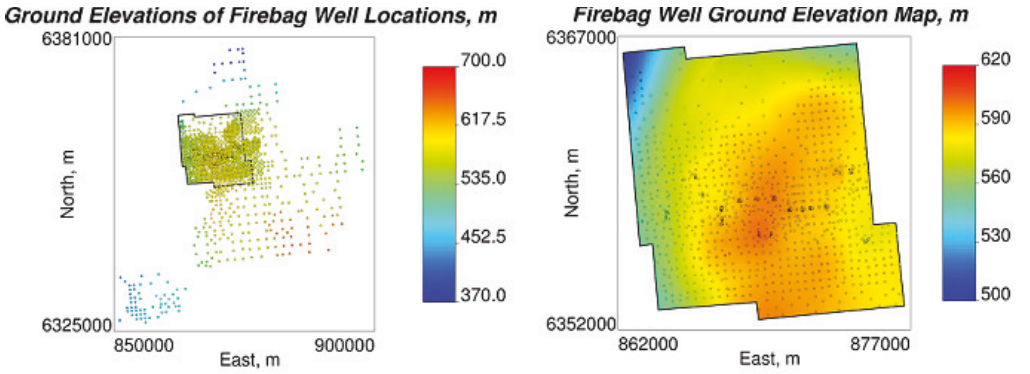


Figure 6.3: Map of well locations at the Firebag project and lease area. Total number of the wells is 1682. Number of the wells under study within lease area is 1222.

of porosity, 2857 values of horizontal permeability, and 13402 measurements of water saturation. There are 230 wells with all required logs.

The objective is geomodels for oil reserve estimation and subsequent flow simulation at three different resolutions. The key geomodeling variables are:

- Geological facies: sand, shale, breccia, and IHS
- Structural geology: reservoir top and base of the McMurray Formation
- Petrophysical properties: porosity by-facies, permeability (horizontal permeability) by-facies, fluid saturation by-facies (water saturation, where oil saturation is the remaining fraction), and the volume fraction of shale mineral (V-shale) by-facies
- Seismic acoustic impedance

Because of the unprocessed nature of the data, several assumptions and simplifications are required. Given the information above, the geostatistical data analysis for the Firebag project is as follows.

1. Perform facies classification based on the well log data

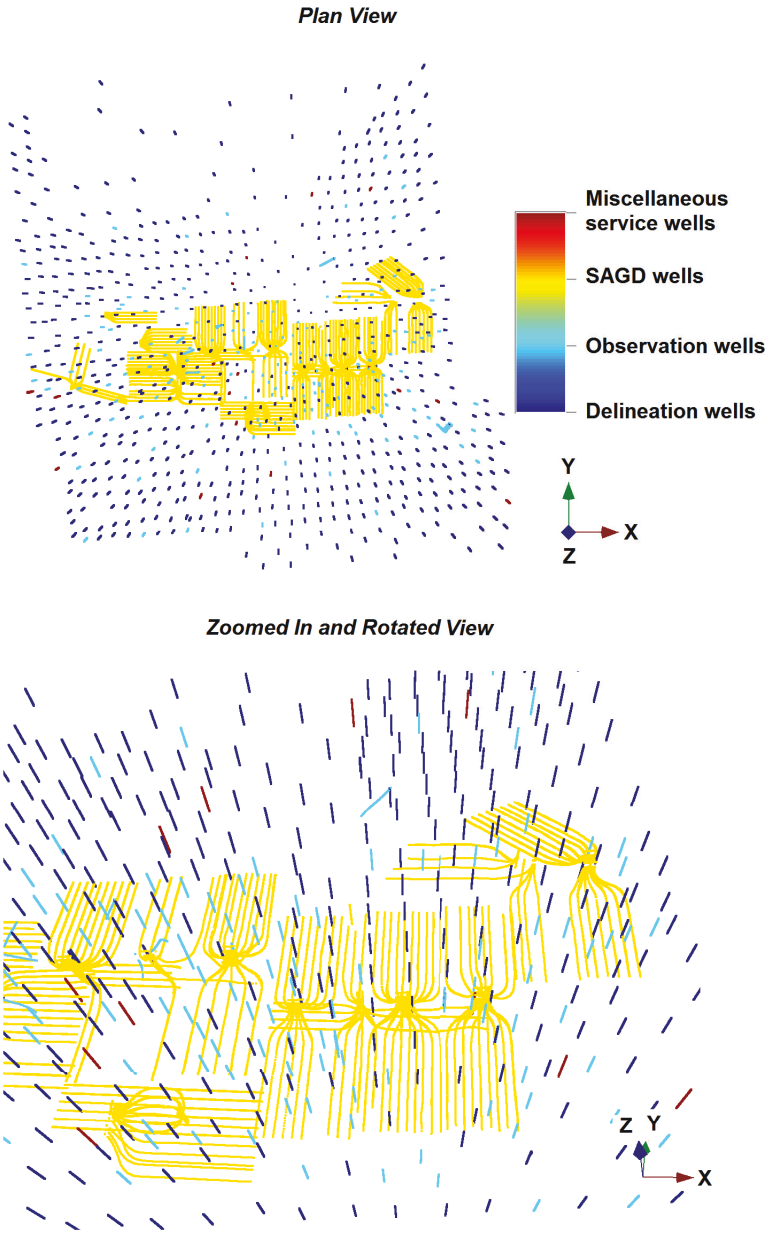


Figure 6.4: Well trajectories at the Firebag lease area. Total well number is 1222. There are 233 SAGD well pairs (233 production wells and 233 injection wells) shown in yellow, 545 delineation wells shown in dark blue, 180 observation wells shown in light blue, and 31 miscellaneous service wells shown in red.

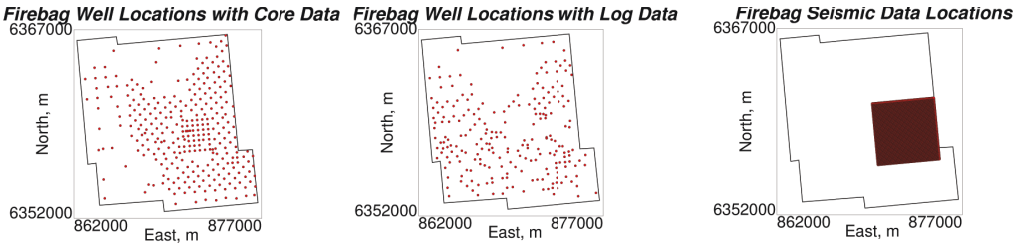


Figure 6.5: Plan view of data locations – 302 wells with core data, 224 wells with log data, and seismic coverage resolution for all facies. Not all shown wells have measurements of the entire set of modeled geologic reservoir properties.

2. Define McMurray Formation top and base elevations from facies classification and adjust vertical coordinates to the top of McMurray Formation

3. Clean core data by-facies: porosity, permeability, water saturation, and V-shale

4. Define log porosity and V-shale data for each facies

5. Infer log permeability and water saturation data from log porosity data, assuming permeability and water saturation have similar spatial structure as porosity and using core data relationship between porosity and permeability, porosity and water saturation

6. Compute variogram models of key properties in normal scores units

7. Process seismic data: get volumes from the Gassmann's fluid substitution model using log data of all facies at once

8. Finally, perform grid-free simulation conditional to all available scattered and exhaustively sampled data to get petrophysical models of various resolutions

6.2.2 Facies Classification

The gamma ray (GR) and photoelectric effect (PE) logs are used to define the facies types [20]. The sand facies is characterized by low gamma ray and low photoelectric effect values. The shale facies has high readings of gamma ray and photoelectric effect. The IHS facies is closer to sand, and breccia facies is closer to shale in their log properties. Based on these characteristics, an algorithm shown in Fig. 6.6 is applied to assign facies based on the gamma ray and photoelectric effect log values. When the proposed facies classification algorithm is applied to the wells, realistic stratigraphy is defined along the well bores. As an example, facies defined in the well 1AA102009405W400 are shown in Fig. 6.7 that appear in compliance with the stratigraphic conceptual model presented in Fig. 6.2. The facies statistical distribution and horizontal and vertical experimental variograms derived from all available data with corresponding variogram models for each facies type are shown in Figs. 6.8 and 6.9, respectively. Because of the particular geological deposition of the facies, the experimental variograms do not reach corresponding theoretical sill values for distances of 3.0 km for all facies. It may indicate zonal anisotropy. The variogram model in the form of the covariance function is depicted in Eq. (6.1). These statistics and variogram models are used later to generate stationary domains with SIS for subsequent grid-free simulation of geological properties of the petroleum reservoir in the Firebag project.

$$C(\boldsymbol{h}) = \begin{cases} C_{shale}(\boldsymbol{h}) = 0.10 Sph_{\substack{r_h=400.0\ m \\ r_v=2.6\ m}}(\boldsymbol{h}) + 0.08 Exp_{\substack{r_h=1300.0\ m \\ r_v=25.0\ m}}(\boldsymbol{h}) \\ C_{sand}(\boldsymbol{h}) = 0.12 Sph_{\substack{r_h=300.0\ m \\ r_v=2.5\ m}}(\boldsymbol{h}) + 0.10 Exp_{\substack{r_h=1000.0\ m \\ r_v=30.0\ m}}(\boldsymbol{h}) \\ C_{IHS}(\boldsymbol{h}) = 0.07 Sph_{\substack{r_h=500.0\ m \\ r_v=1.5\ m}}(\boldsymbol{h}) + 0.04 Exp_{\substack{r_h=550.0\ m \\ r_v=12.0\ m}}(\boldsymbol{h}) \\ C_{breccia}(\boldsymbol{h}) = 0.024 Exp_{\substack{r_h=300.0\ m \\ r_v=2.0\ m}}(\boldsymbol{h}) \end{cases} \tag{6.1}$$

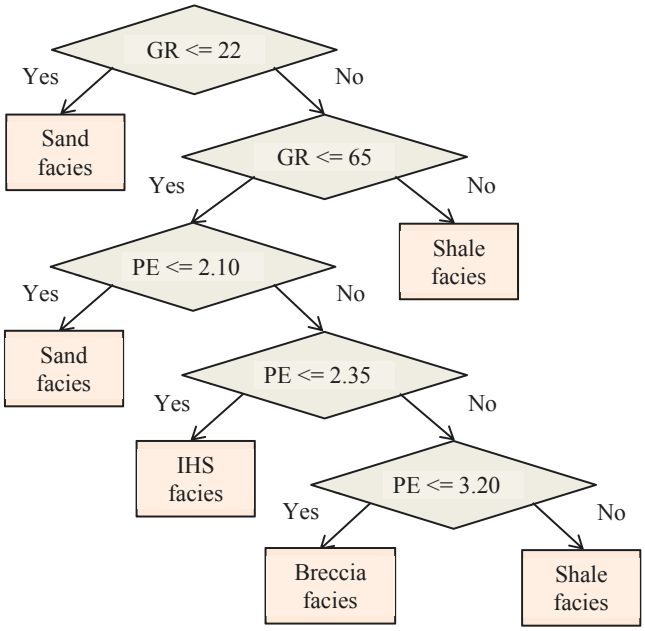


Figure 6.6: Facies classification scheme based on the gamma ray (GR) and photoelectric effect (PE) logging measurements.

6.2.3 Reservoir Surfaces Definition

The target petroleum reservoir comprises McMurray Formation rock. The stratigraphic top and base of the reservoir are found using facies definition considering that McMurray Formation is bounded by impermeable layers of shale deposition. Thus, the McMurray top and base are inferred from the long shale intervals that are typical to Wabiskaw or Clearwater Formations overlying the McMurray Formation and eroded Devonian Age rock underlying the McMurray Formation as shown in Fig. 6.7 for well 1AA102009405W400. A total of 224 wells with classified facies are found. The picks of top and base are carried out for all wells in the same way as for the well 1AA102009405W400 shown above. The target reservoir top is easier to distinguish. Once the reservoir boundaries are defined along each of 224 wells, the surfaces can be modeled with the grid-free simulation to model

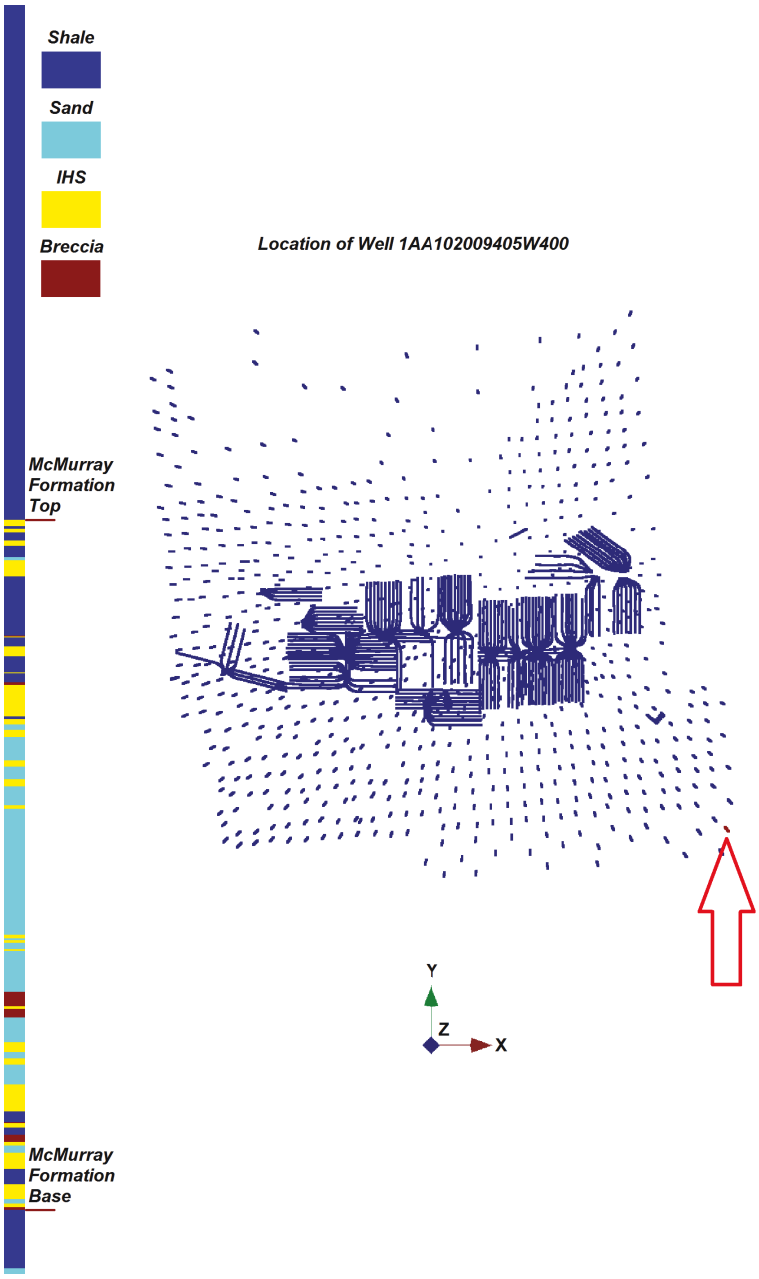


Figure 6.7: Facies classification example of well 1AA102009405W400 – gamma ray, neutron porosity, and photoelectric effect log data are used for the facies classification. Four facies are distinguished: shale (0), sand (1), IHS (2), and breccia (3). The example well is indicated by red arrow.

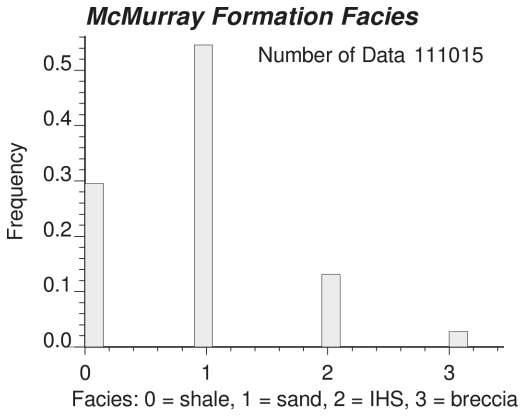


Figure 6.8: Histogram of McMurray Formation facies. Four facies are distinguished: shale (0), sand (1), IHS (2), and breccia (3).

the vertical limits to the geological model. First, the reservoir top elevation and thickness, which is the reservoir base elevation subtracted from the reservoir top elevation, are modeled to better constrain the reservoir base. Second, the reservoir base elevation is found by subtracting the modeled thickness from the modeled top elevation. The data histograms of the structural elements are presented in Fig. 6.10. The scatter plots between reservoir top and thickness in original units and normal scores are shown in Fig. 6.11. The experimental direct and cross-variograms for normal scores of reservoir top elevation and thickness with associated variogram model, which is used for the simulation, are depicted in Fig. 6.12 and expressed in Eq. (6.2). Fig. 6.13 contains the kriging maps of three structural elements of the reservoir to better understand the shape of the formation. The variogram model used in the estimation is not presented. The top of the reservoir seems very continuous. The reservoir thickness resembles the complexity of the channelized (fluvial and deltaic) nature of the Firebag deposit quite well.

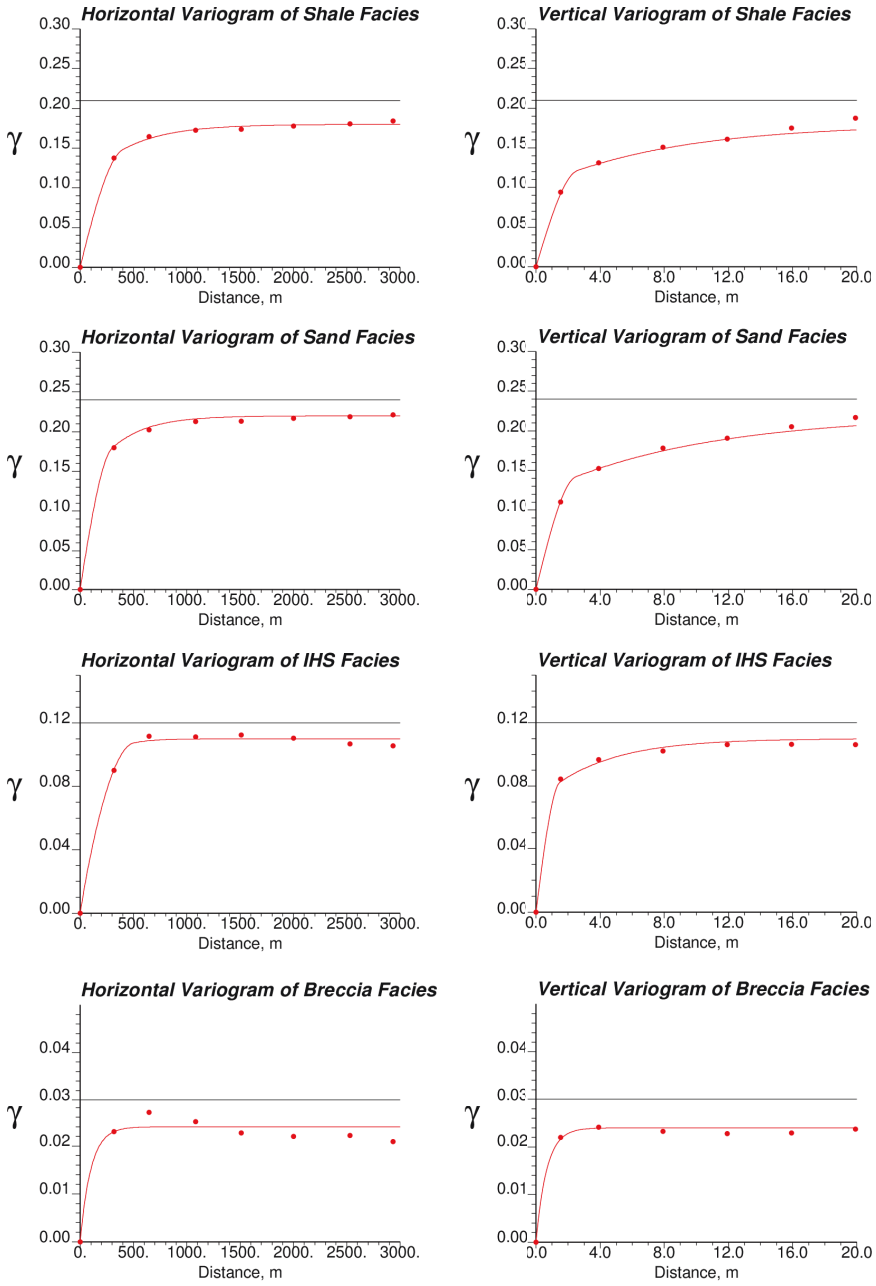


Figure 6.9: Experimental horizontal and vertical variograms and associated models for McMurray Formation facies. Experimental variograms are shown by red dots. Variogram models are presented by solid red lines.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_{top}}(\boldsymbol{h}) = 1.00 Exp_{r=3000.0\ m}(\boldsymbol{h}) \\ C_{Y_{top}Y_{thickness}}(\boldsymbol{h}) = 0.56 Exp_{r=3000.0\ m}(\boldsymbol{h}) \\ C_{Y_{thickness}}(\boldsymbol{h}) = 1.00 Exp_{r=3000.0\ m}(\boldsymbol{h}) \end{cases} \tag{6.2}$$

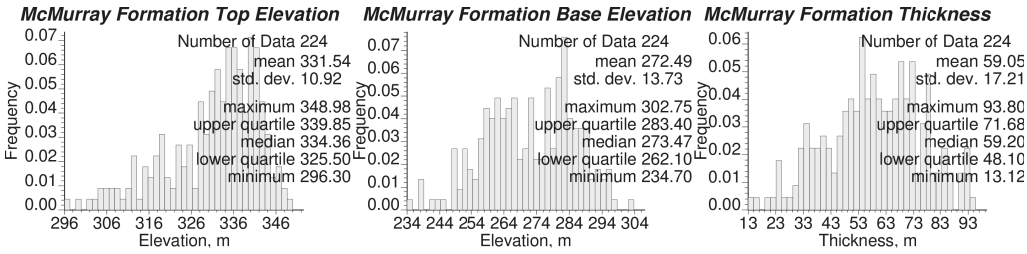


Figure 6.10: Definition of the McMurray Formation top and base elevations – histograms of the data elevation and thickness.

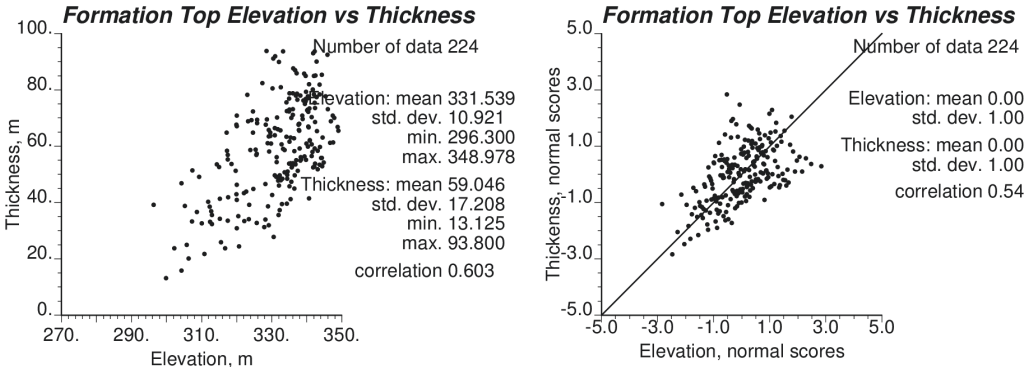


Figure 6.11: Definition of the McMurray Formation top and base elevations – scatter plots of the formation top data elevation and formation data thickness in meters and normal scores.

Because the McMurray top is relatively undisturbed and the McMurray base is deposited on an eroded surface, the adjustments to vertical coordinates are made for all subsequent calculations relative to the McMurray Formation top to make sure that data at similar adjusted elevations correspond to the same position of geological deposition within each well [68]. The vertical coordinate z of a location $\boldsymbol{u}(x, y, z)$ in the space is adjusted to coordinate z' as shown in Eq.

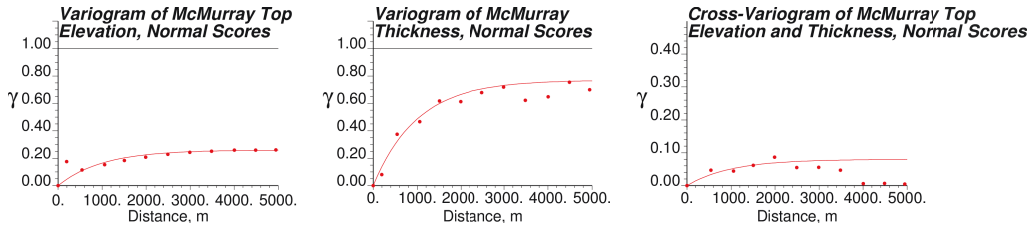


Figure 6.12: Definition of the McMurray Formation top and base elevations – experimental omnidirectional direct and cross-variograms shown by red dots and resulting variogram model for formation top elevation and formation thickness presented by red line.

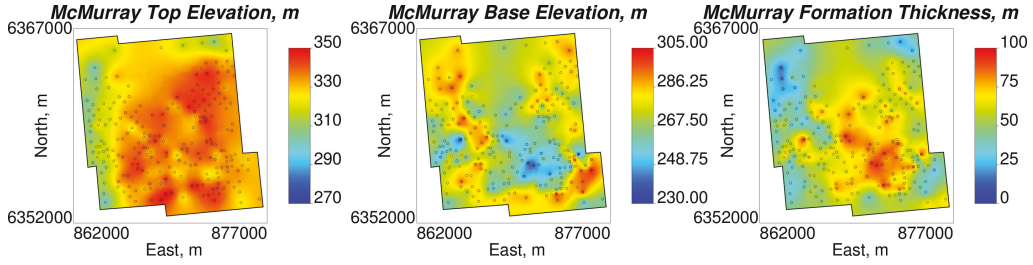


Figure 6.13: Definition of the McMurray Formation top and base elevations – kriging maps of formation top elevation, base elevation, and thickness.

(6.3), where $z_{top}(x, y)$ is the elevation of the McMurray Formation just above location $\boldsymbol{u}$. Therefore, modeling of the reservoir top in a grid-free manner is very useful to define z_{top} for deviated wells and wells without reservoir boundaries picks. Note that the relative vertical coordinates z' are negative. To get original elevations z back, Eq. (6.4) can be used.

$$z' = z - z_{top}(x, y) \tag{6.3}$$

$$z = z' + z_{top}(x, y) \tag{6.4}$$

Therefore, the vertical boundaries of the model are defined to be within McMurray Formation, and horizontal boundaries are defined by the lease area

extent. The modeling is performed relative to the McMurray Formation top.

6.2.4 Core Data Analysis

The petrophysical data analysis applied to the data is explained next. First, the core data are analyzed by-facies. Then, the log data are processed and used in the GFS-based modeling, because of wider coverage, greater data number, and denser sampling rate along the wells. The variables to be analyzed from core and log data are porosity, permeability (horizontal permeability, because there are no relibale data on vertical permeability), and water saturation. No gas is assumed to present in the reservoir, and, therefore, the oil saturation is simply the remaining of the rock porous volume after water saturation is subtracted.

The by-facies histograms of cleaned porosity, permeability, and water saturation are shown in Fig. 6.14. The cleaning involves removal of the outliers and suspiciously large or small data values. The sand facies, breccia and IHS have the highest porosity and permeability. The shale facies is lower quality relative to the other three facies. The water saturation is the highest in the shale and the lowest in the sand. The values for the IHS and breccia facies are in between shale and sand. The by-facies scatter plots between the porosity and permeability data are shown in Fig. 6.15. The by-facies scatter plots between the porosity and water saturation data are shown in Fig. 6.16.

6.2.5 Log Data Analysis

The by-facies histograms of the corrected neutron log porosity ϕ' are shown in Fig. 6.17. The neutron porosity ϕ is corrected by V-shale, as shown in Eq. (6.5). The V-shale is derived from the gamma ray as expressed in Eq. (6.6), where GR_{min} and GR_{max} are the minimum and maximum readings of the gamma ray along the well bore. The porosity log data represent more variability than the core data. The mean of the porosity neutron log data matches the mean of the

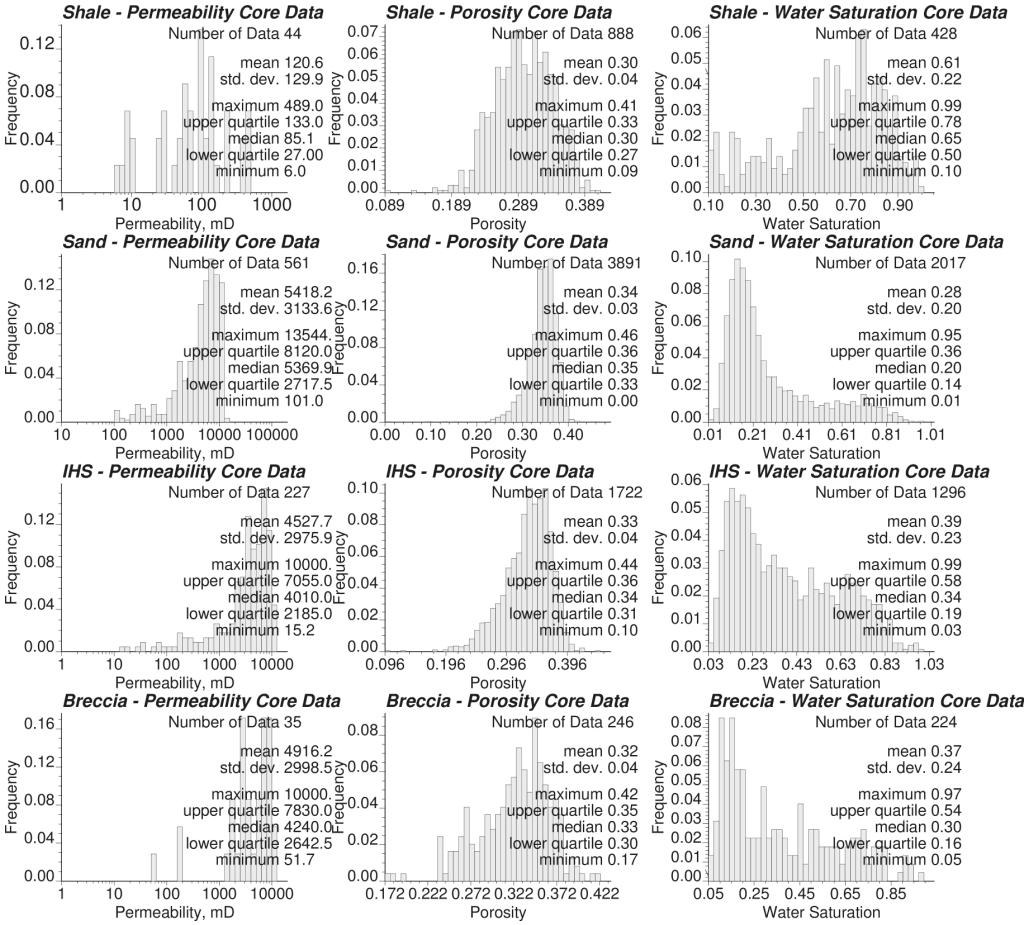


Figure 6.14: Histograms of core data of porosity, permeability, and water saturation for each facies.

porosity core data for all facies except for shale. The mean of the porosity log data for shale is intentionally lower, because it is believed that core data are preferentially sampled in better quality rock. Next, the porosity log data are sampled randomly at 5.0 m for the remainder of the case study. One sample from 5.0 m interval is selected randomly to be a representative of the entire interval. Corresponding by-facies histograms of the porosity log data are shown in Fig. 6.18, and they are in compliance with previous histograms in Fig. 6.17.

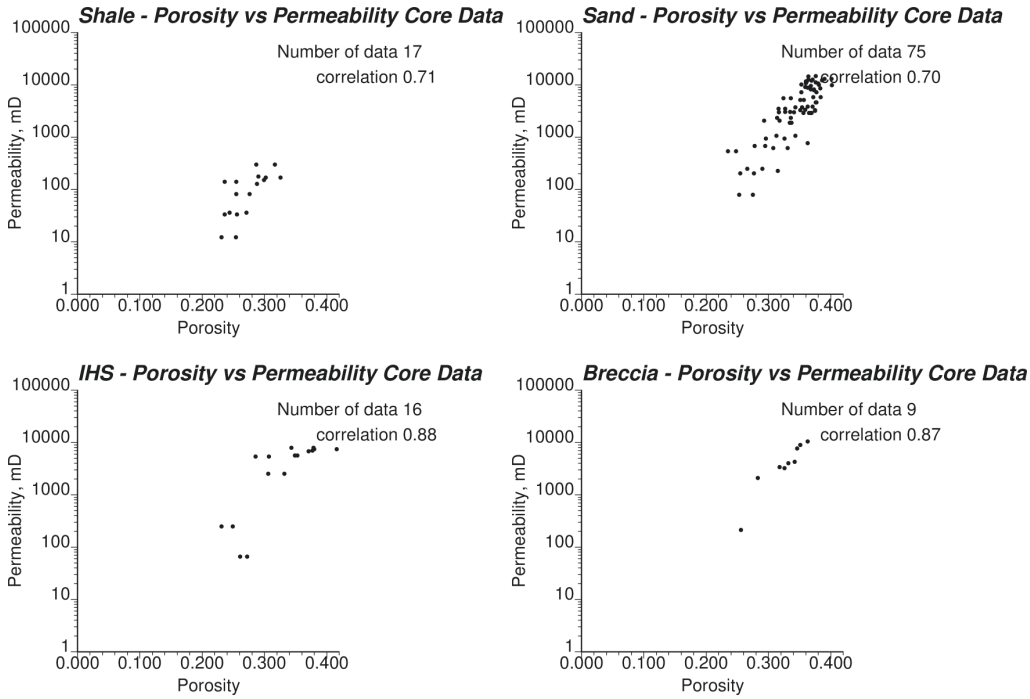


Figure 6.15: Scatter plots between porosity core data and permeability core data for each facies.

$$\phi' = \phi(1.0 - V_{shale}) \tag{6.5}$$

$$V_{shale} = \frac{GR - GR_{min}}{GR_{max} - GR_{min}} \tag{6.6}$$

Porosity is reliably measured by logging tools. The permeability and water saturation are derived from the porosity log data using cloud transformation/P-field simulation technique [73]. The direct physical relationships could be used to derive permeability and water saturation from porosity data. But it would diminish the uncertainty in the relationships between these properties. In cloud transformation, values are drawn from the conditional distributions between these two variables, which can be presented analytically or in the form of a scatter

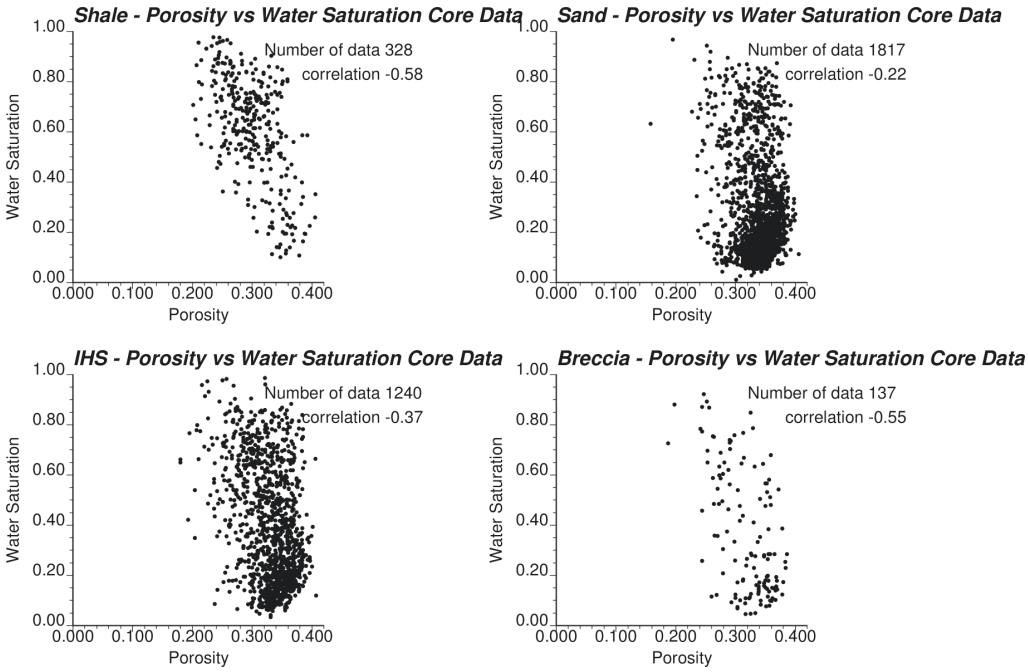


Figure 6.16: Scatter plots between porosity core data and water saturation core data for each facies.

plot. First, correlated probabilities are simulated at the well locations honoring spatial structure of the variable to be assigned. Second, the simulated values of the secondary variables are drawn from the conditional distributions using these correlated probabilities and collocated values of the primary variable. Here, the primary variable is porosity, and the secondary variables are permeability and water saturation. The probability fields for horizontal permeability and water saturation are generated with the GFS using variogram models computed from the core data of permeability and water saturation for sand facies, which are shown in Fig. 6.19. For most faces, the relationship between core porosity data and core permeability data are approximated analytically by bivariate Gaussian distribution. The relationship between core porosity data and core water saturation data for each facies are described by the conditional distribution computed

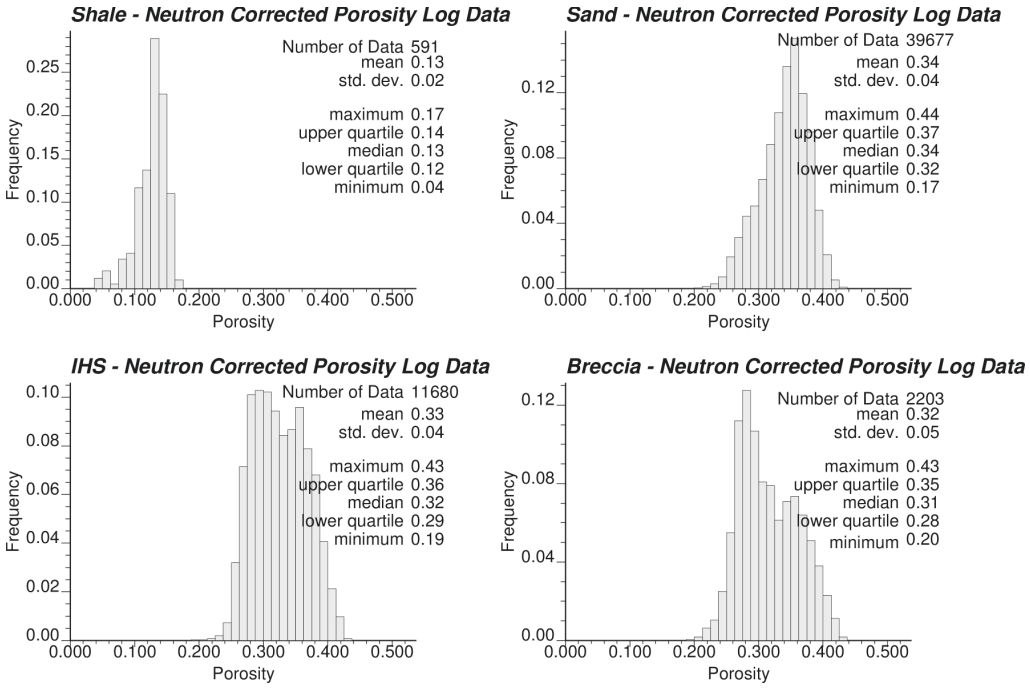


Figure 6.17: Histograms of corrected neutron porosity log data by-facies.

within local windows of porosity. The fitted relationships are shown in Figs. 6.20 and 6.21 for permeability and water saturation, respectively. The resulting simulated permeability pseudo-log data and water saturation pseudo-log data are shown on the same figures as scatter plots with the neutron porosity log data for each facies. Simulated log data relationships have been slightly adjusted by the P-field simulation from core data to meet petrophysical properties expected by each facies. Both core and log data types match each other quite well for all facies except shale, where log porosity values are lower than core porosity values, and are deemed more realistic. The pseudo-log data of permeability and water saturation are not unique. Several realizations of the pseudo-log data could be generated and used in the conditioning corresponding simulation realization. Even though the porosity-water saturation relationship is well defined by core data, the cloud transformation allows to sort out outliers and make all data types

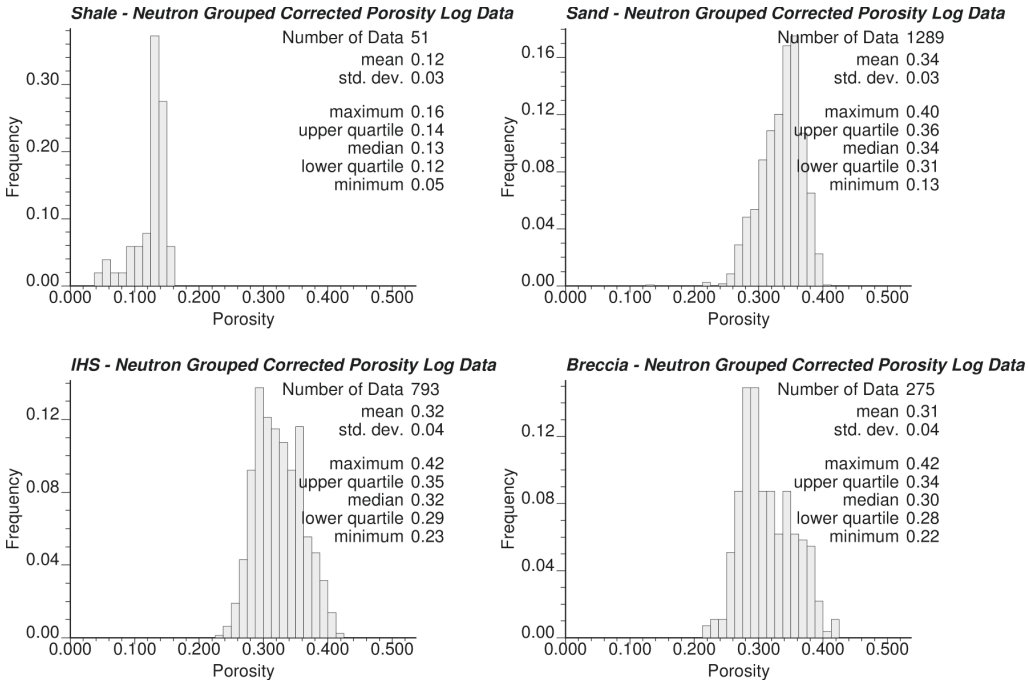


Figure 6.18: Histograms of grouped corrected neutron porosity log data by-facies.

consistent in the form of the log data. The uncertainty in the data, such as measurement errors, could be addressed through the data imputation technique [5]. The idea of imputation is based on the noise injection into the data for various realizations, and synthetic simulation of missing homotopic (collocated) data of various geological properties.

The locations of the log data of porosity, permeability, and water saturation, which are used as conditional data in the grid-free simulation, are shown in Fig. 6.22 by-facies. The experimental direct and cross-variograms for these three variables in normal scores of log data for sand facies and corresponding LMC variogram model are presented in Fig. 6.23 for the horizontal direction and in Fig. 6.24 for the vertical direction. It is deemed that the sand facies variogram model can be applied with a small error to model geological properties within

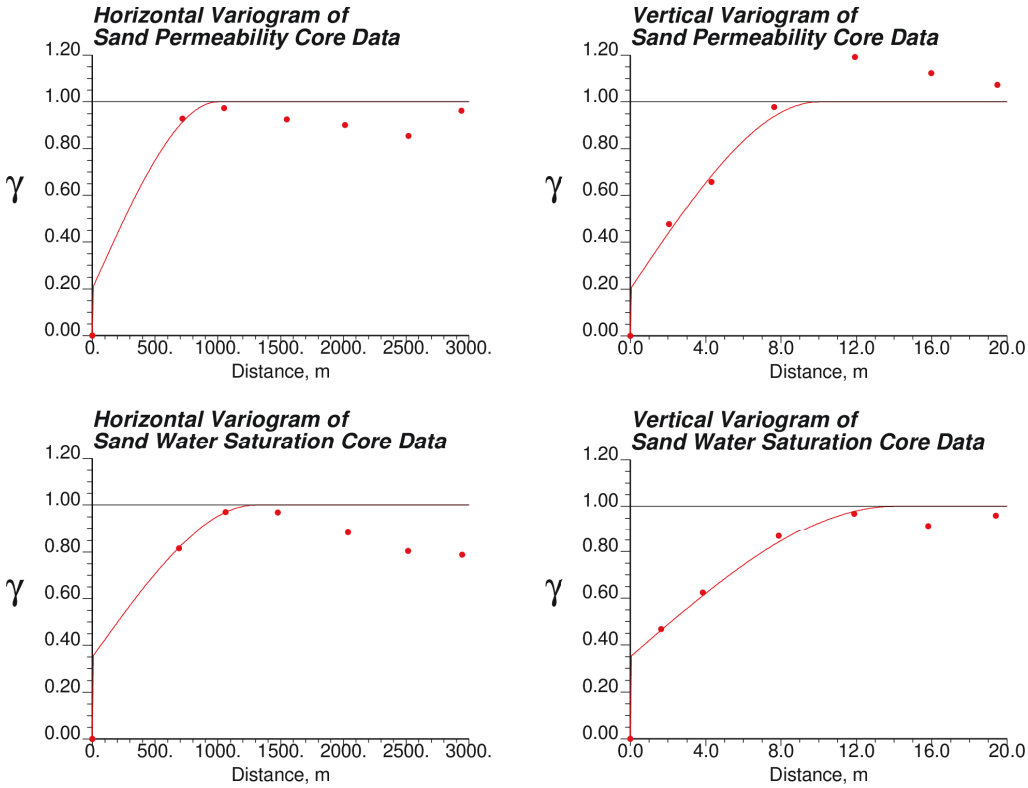


Figure 6.19: Horizontal and vertical variograms of permeability and water saturation core data for sand facies. Experimental variograms are presented by red dots. Variogram model is a red line.

other facies. The variogram model has the form shown in Eq. (6.7). It is an exponential variogram type with isotropic horizontal component with a range of 600.0 m and a vertical range of 8.0 m.

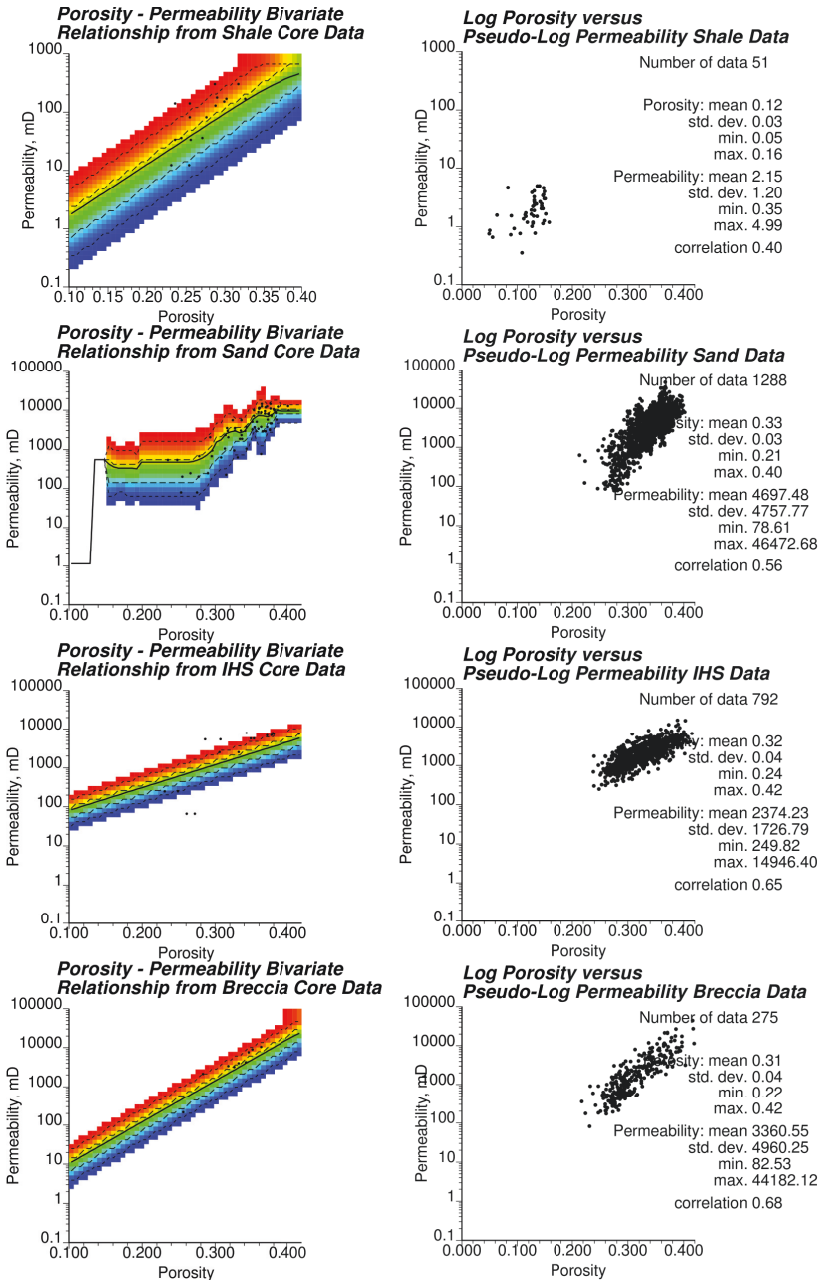


Figure 6.20: Bivariate relationship model between porosity and permeability core data and resulting permeability pseudo-log data for each facies.

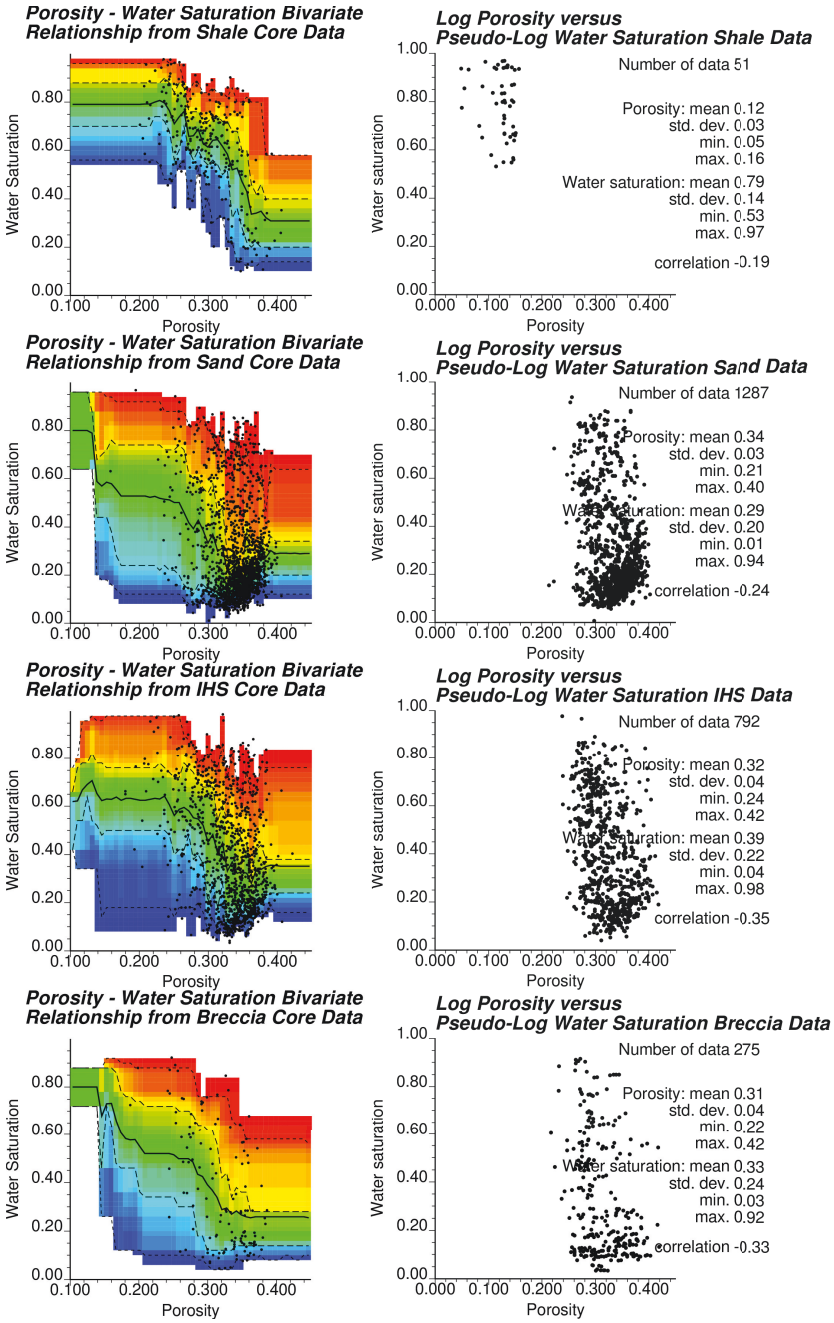


Figure 6.21: Bivariate relationship model between porosity and water saturation core data and resulting water saturation pseudo-log data for each facies.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_\phi}(\boldsymbol{h}) = 0.93 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_\phi Y_k}(\boldsymbol{h}) = 0.69 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_\phi Y_s}(\boldsymbol{h}) = -0.02 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_k}(\boldsymbol{h}) = 0.93 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_k Y_s}(\boldsymbol{h}) = 0.09 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_s}(\boldsymbol{h}) = 1.0 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \end{cases} \tag{6.7}$$

6.2.6 Seismic Data Generation

A seismic survey was conducted over part of the lease area, but these data are not available to the public. Synthetic seismic data is generated to make the case study more realistic in the use of an exhaustive secondary data. Gassmann's fluid substitution model is adopted to generate synthetic acoustic impedance based on the porosity, water saturation, and V-shale data [39, 84]. These variables were calculated for all facies, because seismic surveys measure acoustic properties independent of facies. The acoustic impedance can be computed using Eqs. (6.8) - (6.18) of Gassmann's fluid substitution model adapted for a petroelastic model, where AI is the P-wave acoustic impedance; K_b is the elastic bulk modulus; K_s is the elastic shear modulus; ϕ is the porosity; and ρ is the density. The saturated *sat* and dry *dry* rocks are distinguished. It is assumed that the rock matrix is a binary mixture of the clay and quartz minerals of varying concentrations defined by V-shale. Physical parameters of the clay and quartz minerals, water and oil fluids can be found in Table 6.1 and Table 6.2. It is also assumed that no gas is present in the reservoir. The reservoir fluid comprises brine reservoir water and heavy viscous oil.

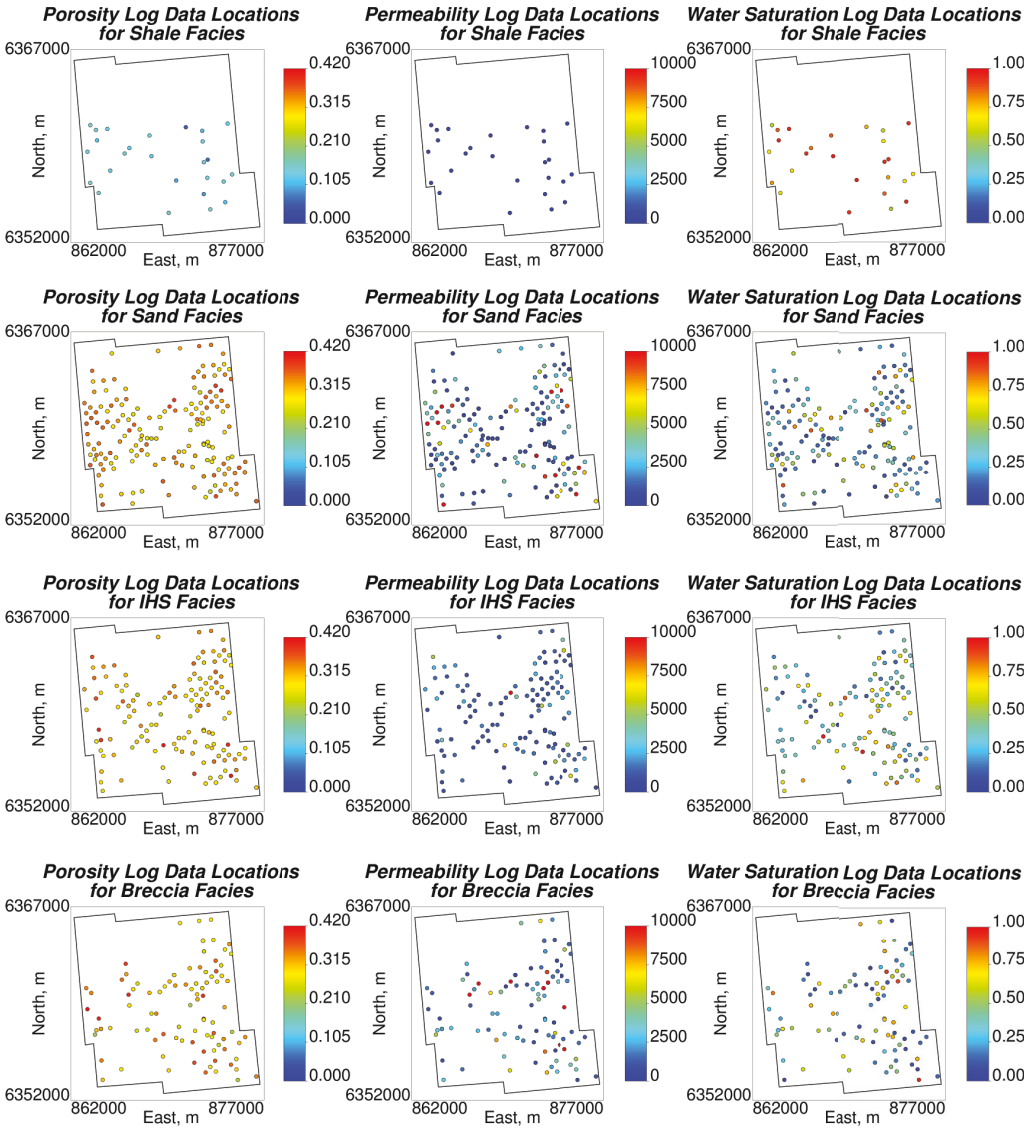


Figure 6.22: Plan view of log data locations for porosity, permeability, and water saturation.

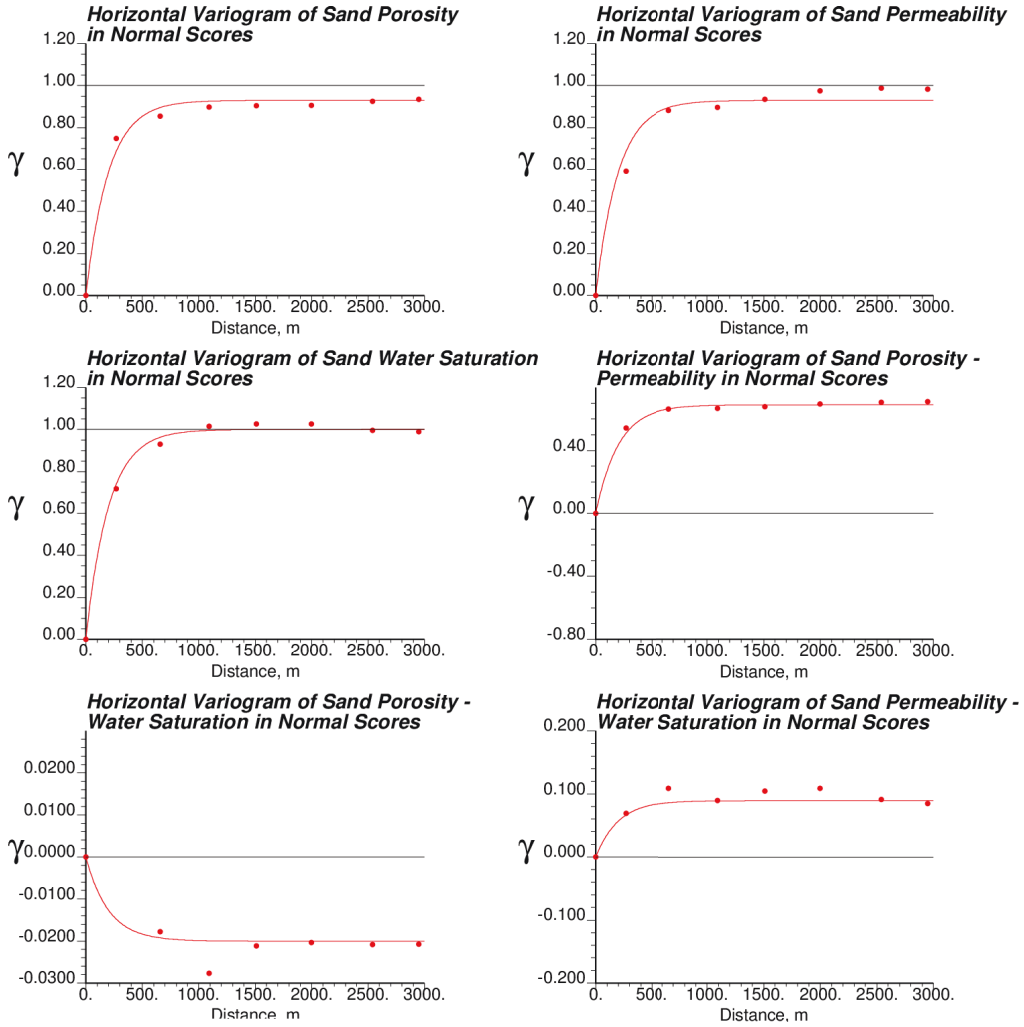


Figure 6.23: Experimental direct and cross-variograms as red dots and associated variogram model shown by red line for normal scores of porosity, permeability, water saturation of sand facies, and their interactions in horizontal direction.

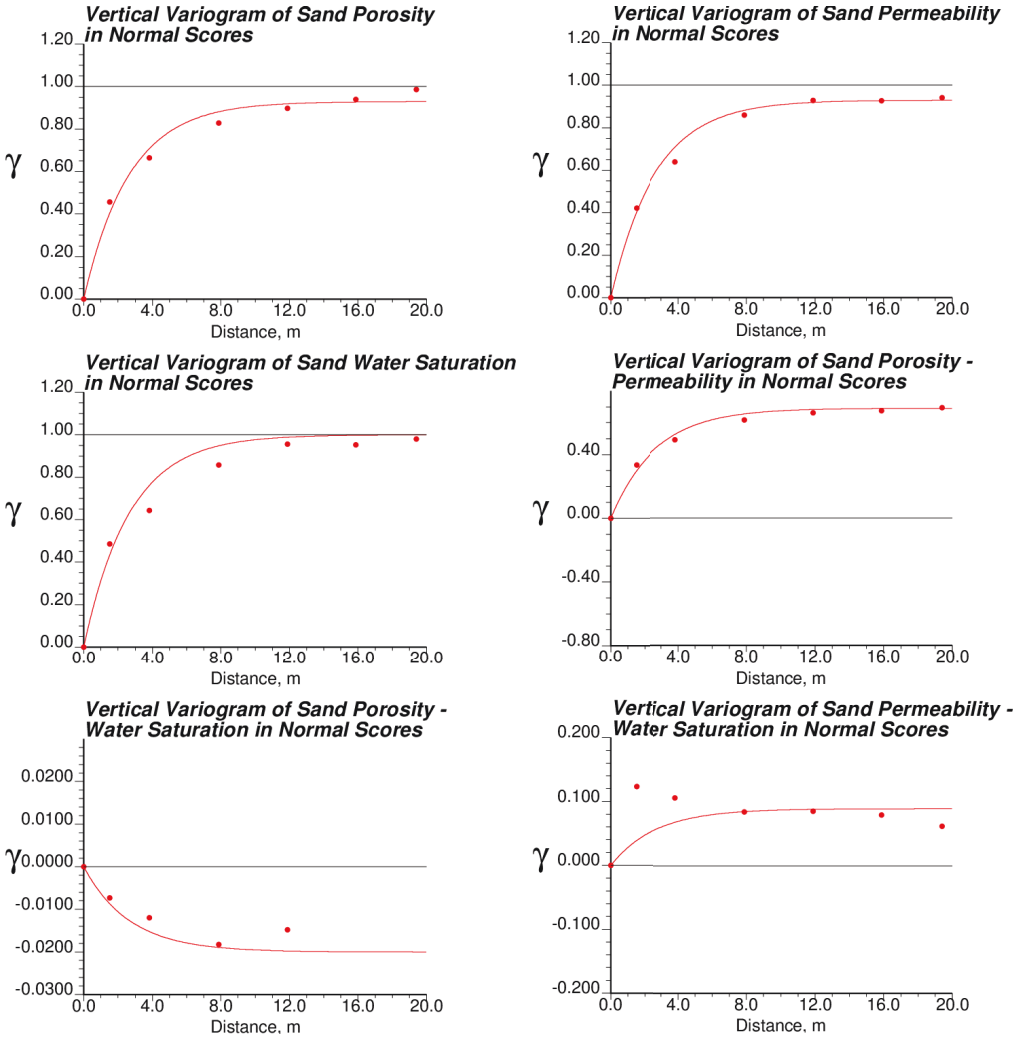


Figure 6.24: Experimental direct and cross-variograms as red dots and associated variogram model shown by red line for normal scores of porosity, permeability, water saturation of sand facies, and their interactions in vertical direction.

$$AI = \sqrt{\rho_{sat}\left(K_{b,sat} + \frac{4}{3}K_{s,sat}\right)} \tag{6.8}$$

$$\rho_{sat} = (1-\phi)\rho_{matrix} + \phi\rho_{fluid} \tag{6.9}$$

$$\rho_{matrix} = V_{shale}\rho_{clay} + (1 - V_{shale})\rho_{quartz} \tag{6.10}$$

$$\rho_{fluid} = s\rho_{water} + (1-s)\rho_{oil} \tag{6.11}$$

$$\frac{K_{b,sat}}{K_{b,matrix} - K_{b,sat}} = \frac{K_{b,dry}}{K_{b,matrix} - K_{b,dry}} + \frac{K_{b,fluid}}{\phi(K_{b,matrix} - K_{b,fluid})} \tag{6.12}$$

$$K_{s,sat} = K_{s,dry} \tag{6.13}$$

$$K_{b,dry} = K_{b,matrix}(1-\phi)^{\frac{3}{1-\phi}} \tag{6.14}$$

$$K_{s,dry} = K_{s,matrix}(1-\phi)^{\frac{3}{1-\phi}} \tag{6.15}$$

$$\begin{aligned} K_{b,matrix} = \frac{1}{2}\Bigg(& K_{b,clay} + K_{b,quartz} + \\ & \frac{1 - V_{shale}}{(K_{b,quartz} - K_{b,clay})^{-1} + V_{shale}(K_{b,clay} + \frac{4}{3}K_{s,clay})^{-1}} + \\ & \frac{V_{shale}}{(K_{b,clay} - K_{b,quartz})^{-1} + (1 - V_{shale})(K_{b,quartz} + \frac{4}{3}K_{s,quartz})^{-1}}\Bigg) \end{aligned} \tag{6.16}$$

$$K_{s,matrix} = \frac{1}{2}\left(K_{s,clay} + K_{s,quartz} + \frac{1 - V_{shale}}{(K_{s,quartz} - K_{s,clay})^{-1} + 2V_{shale}\frac{K_{b,clay}+2K_{s,clay}}{5K_{s,clay}(K_{b,clay}+\frac{4}{3}K_{s,clay})}} + \frac{V_{shale}}{(K_{s,clay} - K_{s,quartz})^{-1} + 2(1 - V_{shale})\frac{K_{b,quartz}+2K_{s,quartz}}{5K_{s,quartz}(K_{b,quartz}+\frac{4}{3}K_{s,quartz})}} \right) \tag{6.17}$$

$$K_{b,fluid} = \left(\frac{s}{K_{b,water}} + \frac{1 - s}{K_{b,oil}} \right)^{-1} \tag{6.18}$$

Table 6.1: Physical properties of clay and quartz minerals.

Mineral Units	Density ρ g/cm^3	Bulk modulus K_b GPa	Shear modulus K_s GPa
Clay (kaolinite)	1.58	15.0	6.0
Quartz	2.65	37.0	44.0

Table 6.2: Physical properties of reservoir fluids.

Fluid Units	Density ρ g/cm^3	Bulk modulus K_b GPa
Pure water	0.99	2.2
Heavy Oil	1.01	1.7

The log data used to generate the acoustic impedance is shown in Fig. 6.25 in a plan view. The histograms of log porosity, V-shale, and water saturation data along with the scatter plots can be found in Fig. 6.26. Computed horizontal and vertical experimental variograms as red dots with the variogram model in a red line are shown in Figs. 6.27 and 6.28, respectively. The expression for the variogram model is provided in Eq. (6.19) in the covariance function form.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_\phi}(\boldsymbol{h}) = 0.83 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_\phi Y_{V_{shale}}}(\boldsymbol{h}) = -0.55 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_\phi Y_s}(\boldsymbol{h}) = -0.15 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_{V_{shale}}}(\boldsymbol{h}) = 0.72 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_{V_{shale}} Y_s}(\boldsymbol{h}) = 0.20 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_s}(\boldsymbol{h}) = 1.00 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \end{cases} \tag{6.19}$$

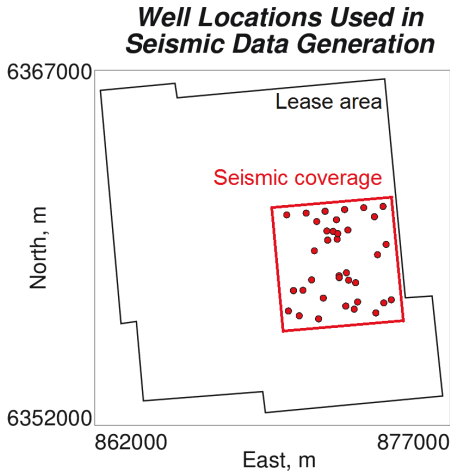


Figure 6.25: Plan view of well locations, log data from which are used for the synthetic acoustic impedance generation.

The simulation of three input variables (porosity, V-shale, and water saturation) is performed with the GFS in normal space at a point scale to provide the inputs to Gassmann's fluid substitution model. The seismic coverage area has been shown previously in Fig. 6.5. The simulation is performed on a grid with adjusted vertical coordinates to align with the formation top. The size of the seismic data grid is 100 × 100 × 20 blocks of 50.0 × 50.0 × 5.0 m^3 spacing. First, the seismic attributes are computed at a point scale from the simulated input variables according to the explained Gassmann's fluid substitution model.

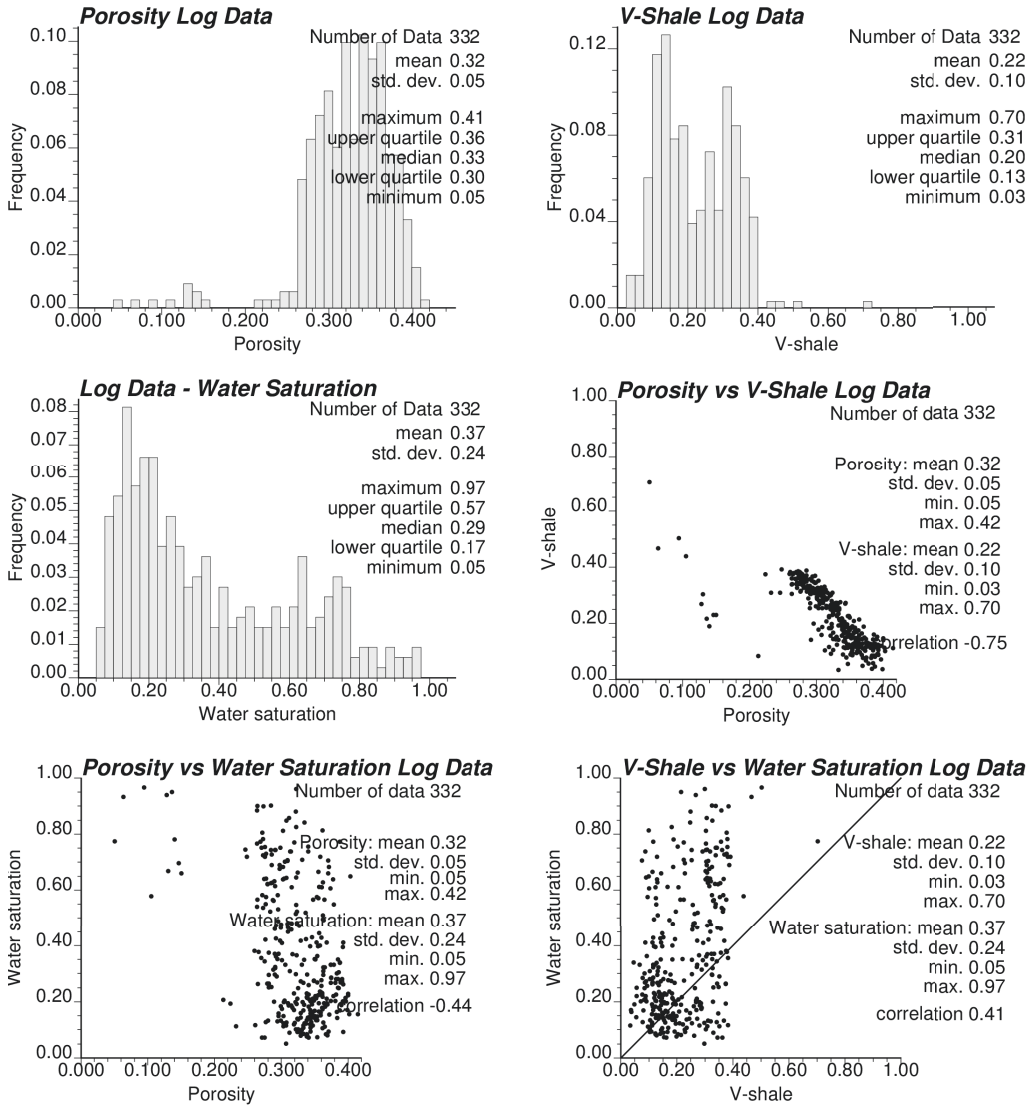


Figure 6.26: Histograms and scatter plots of log porosity, V-shale, and water saturation data used for the synthetic acoustic impedance generation.

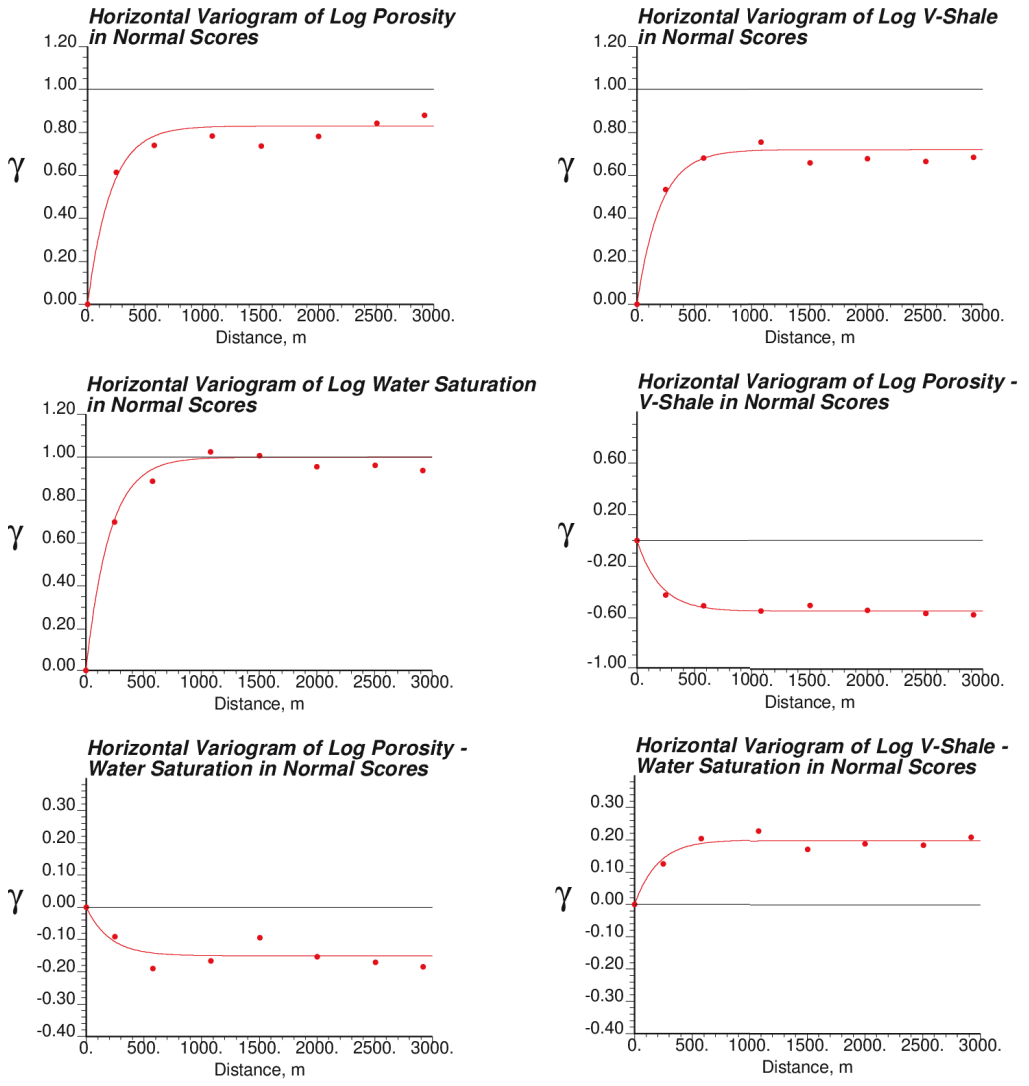


Figure 6.27: Experimental variograms shown by red dots in horizontal direction computed from the log data used for the synthetic seismic data generation and associated variogram model presented by a red line.

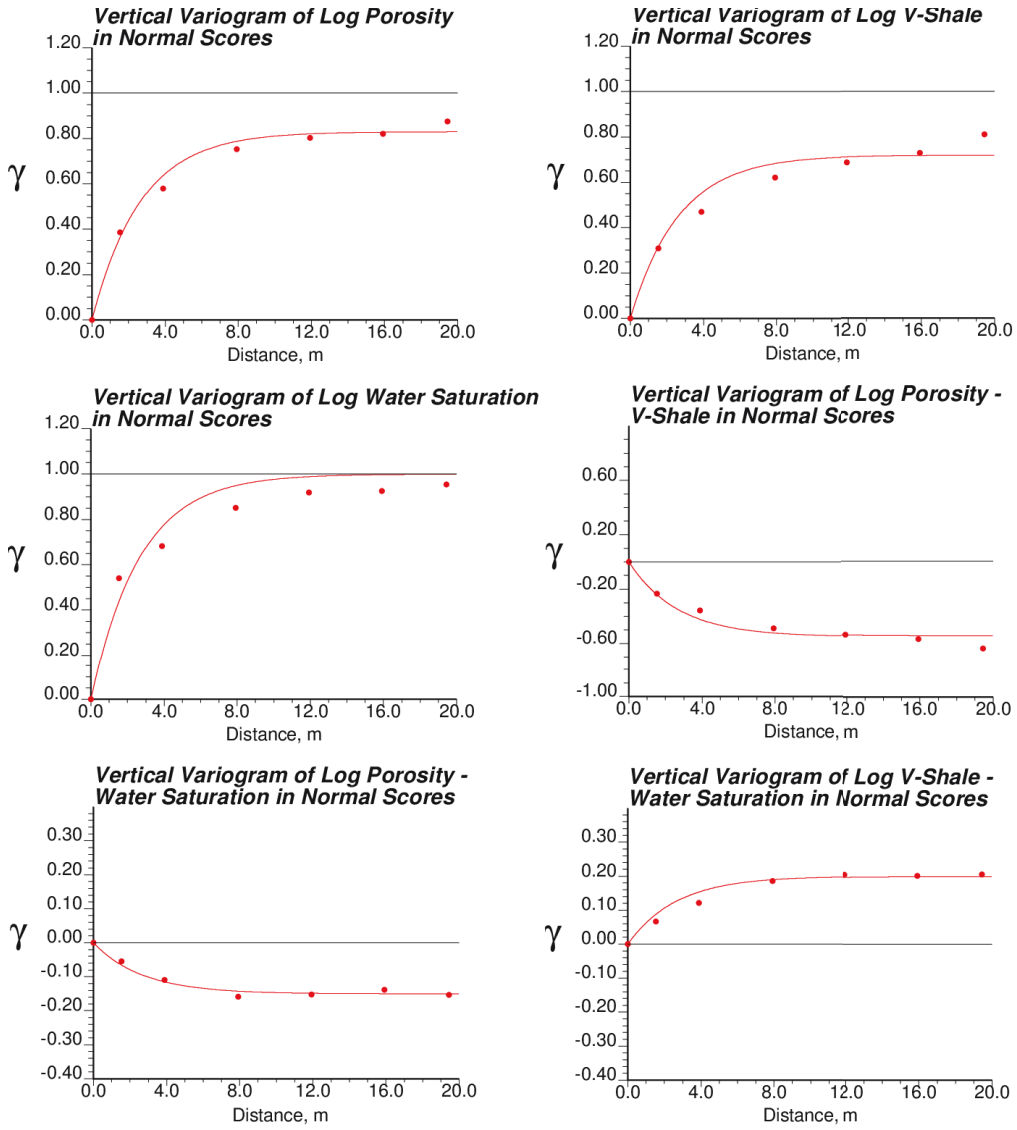


Figure 6.28: Experimental variograms shown by red dots in vertical direction computed from the log data used for the synthetic seismic data generation and associated variogram model presented by a red line.

Then, these point-scale acoustic impedance is arithmetically averaged over 200.0 $\times$ 200.0 $\times$ 20.0 m^3 cubes to get block-scale seismic data. Note that in reality acoustic impedance is not an additive attribute. Thus, it is assumed that resulting acoustic impedance is resolved at a block scale of 200.0 $\times$ 200.0 $\times$ 20.0 m^3 volume. Histograms of the simulated input variables and output acoustic impedance data at point-scale are shown in Fig. 6.29. The histogram of the acoustic impedance at a block scale is shown in Fig. 6.30 and looks to follow normal distribution. The mean has been preserved, and variance has been reduced. The scatter plots between all variables at a point scale are shown in next Fig. 6.31. The acoustic impedance is highly inversely related to porosity. The correlation between acoustic impedance and V-shale or water saturation is weak. The relationships between input variables are preserved, when compared to the data scatter plots in Fig. 6.26. The derived acoustic impedance data are shown in 3-D in original units at point and block scales in Fig. 6.32. The experimental variogram of the synthetic seismic attributes at a point scale in normal scores for vertical and horizontal directions and suggested variogram model are presented in Fig. 6.33. Surprisingly, the variogram structure derived for the key petrophysical properties also fit the acoustic impedance in normal scores. Therefore, the final LMC covariance model for the GFS simulation of the porosity, permeability, water saturation, and acoustic impedance in normal scores is presented in Eq. (6.20). The contributions of every direct- and cross-variogram are licit. The contributions of cross-varioagrams of acoustic impedance with other variables are chosen approximately based on the correlation coefficients between acoustic impedance and rest of the variables.

$$C_Y(\boldsymbol{h}) = \begin{cases} C_{Y_\phi}(\boldsymbol{h}) = 0.93 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_\phi Y_k}(\boldsymbol{h}) = 0.69 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_\phi Y_s}(\boldsymbol{h}) = -0.02 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_\phi Y_{AI}}(\boldsymbol{h}) = -0.90 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_k}(\boldsymbol{h}) = 0.93 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_k Y_s}(\boldsymbol{h}) = 0.09 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_k Y_{AI}}(\boldsymbol{h}) = -0.50 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_s}(\boldsymbol{h}) = 1.0 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_s Y_{AI}}(\boldsymbol{h}) = 0.20 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \\ C_{Y_{AI}}(\boldsymbol{h}) = 1.00 Exp_{\substack{r_h=600.0\ m \\ r_v=8.0\ m}}(\boldsymbol{h}) \end{cases} \tag{6.20}$$

6.3 Grid-Free Simulation

The key petrophysical properties including porosity, permeability, and water saturation are modeled in a grid-free manner within the McMurray Formation reservoir conditional to hard log data presented at point-scale and soft acoustic impedance resolved at a block scale. Three grids are generated to meet the objectives of the case study: a model at coarse resolution for reserve estimation, at medium resolution for flow simulation within the high quality reservoir zone, and at fine resolution around a SAGD well pad in the upper right corner. The grid resolutions are 200.0 × 200.0 × 20.0 m^3, 100.0 × 100.0 × 10.0 m^3, and 20.0 × 50.0 × 3.0 m^3, respectively. The projection of the grid nodes are shown in Fig. 6.34. The modeling is performed within each facies independently. The key point is to generate a consistent realizations at various resolutions that would be part of same random function despite the locations of the simulation nodes and resolution. One realization is generated for illustration in this case study.

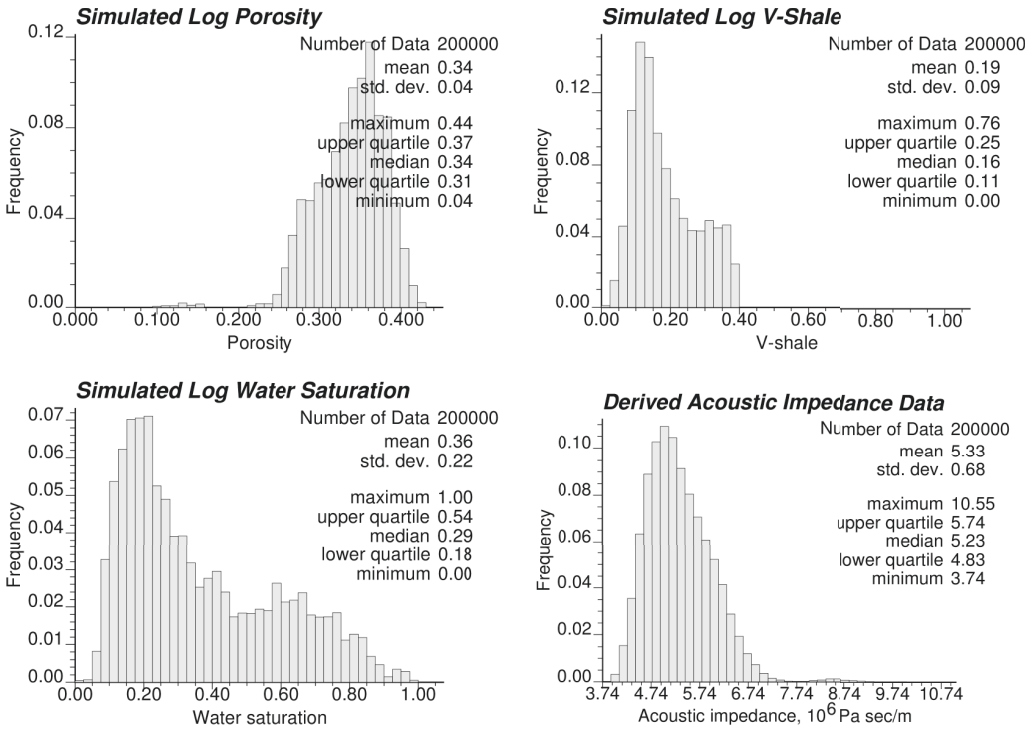


Figure 6.29: Histograms of simulated input parameters (porosity, V-shale, and water saturation) and output synthetic acoustic impedance data at a point scale.

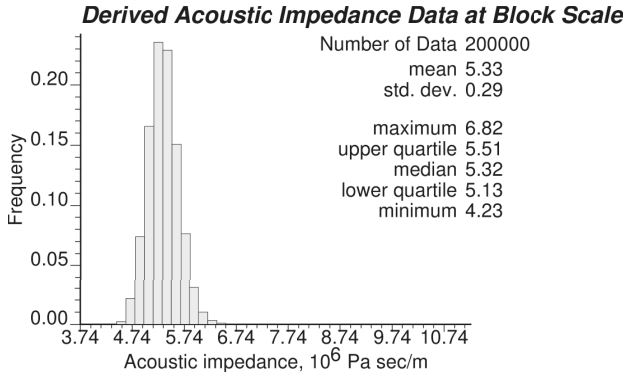


Figure 6.30: Histogram of derived acoustic impedance data at a block-scale.

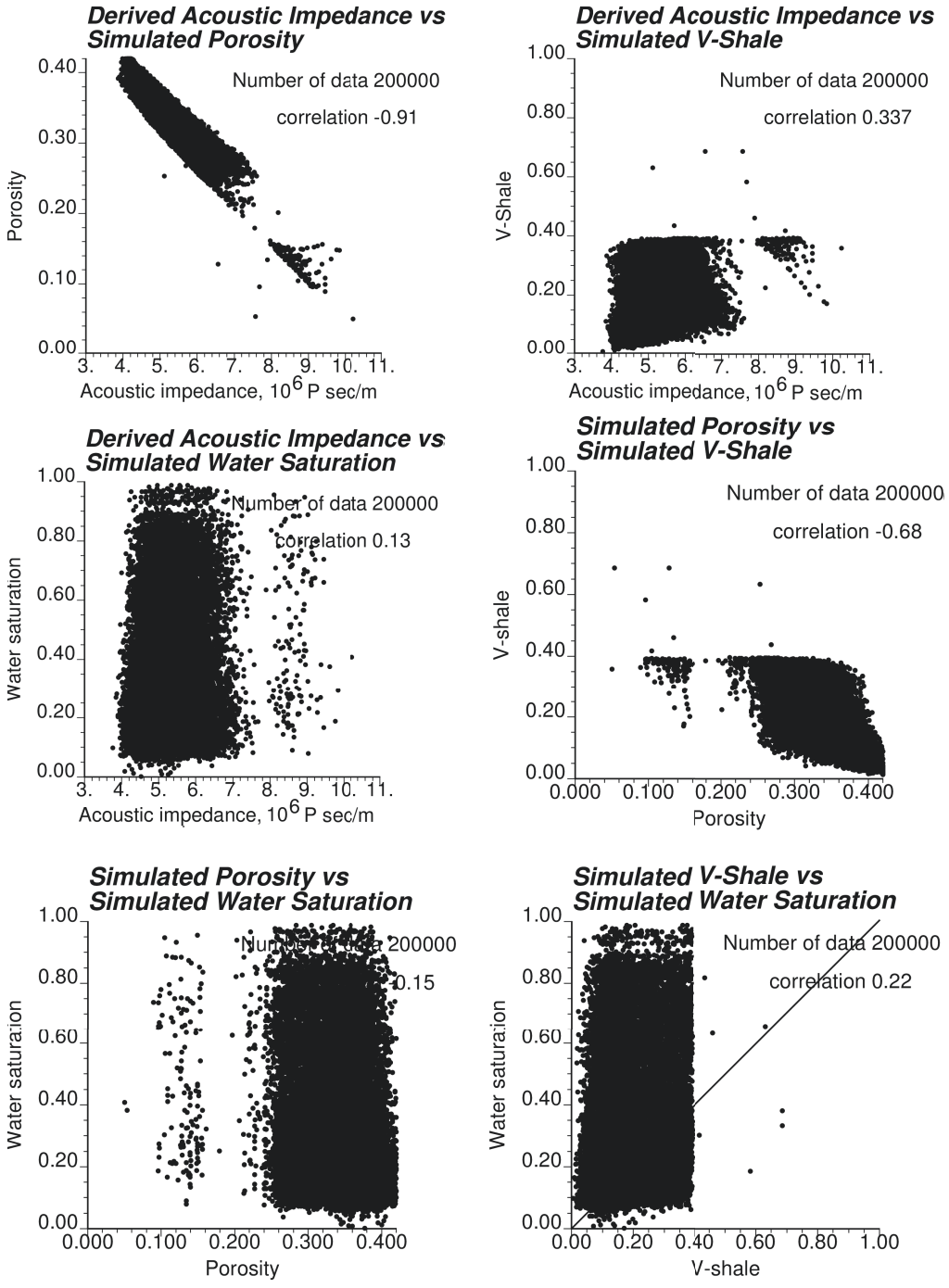


Figure 6.31: Scatter plots between simulated input (parameters porosity, V-shale, and water saturation) and output synthetic acoustic impedance data at a point scale.

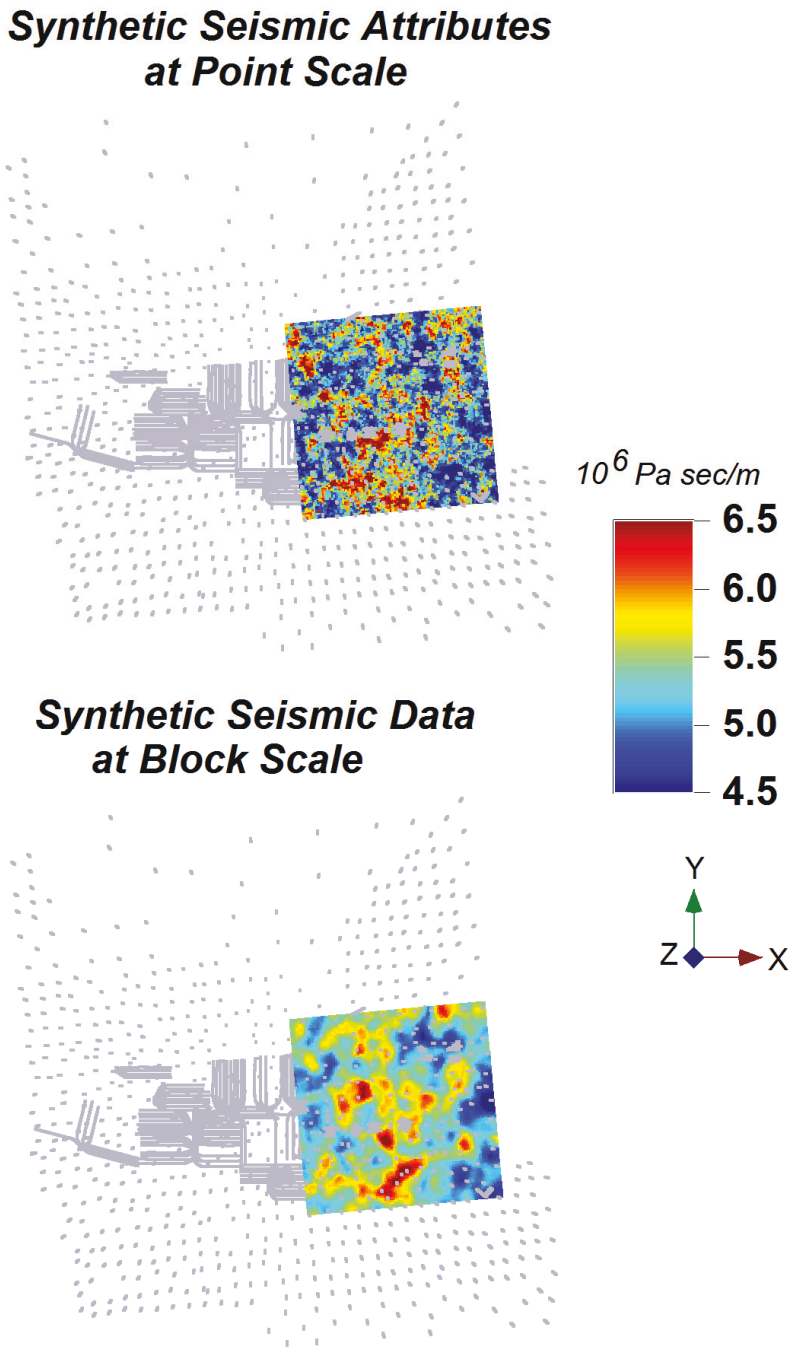


Figure 6.32: A plan view of synthetic seismic data in original units at a point and block scales.

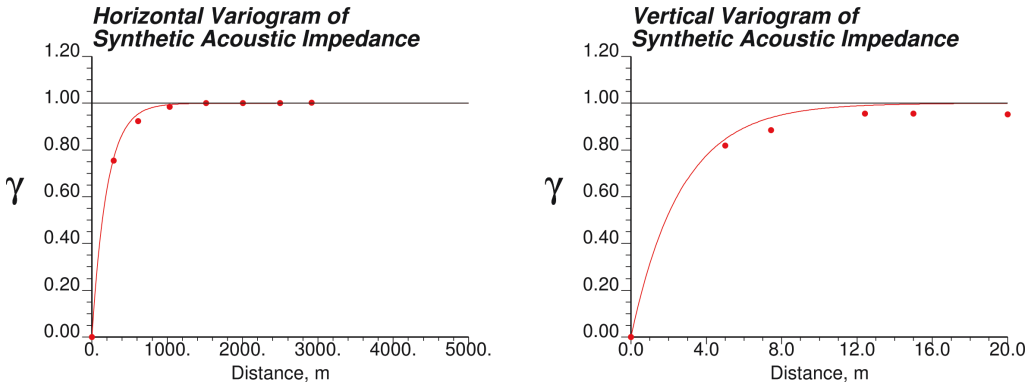


Figure 6.33: Experimental horizontal and vertical variograms shown by red dots of the synthetic acoustic impedance data at a point scale in normal scores and suggested variogram model shown by red line.

The number of the turning lines for 3-D is 161. The number 3161 of the Fourier coefficients is selected to keep the error less than 1.0 % in the 1-D covariance decomposition.

6.3.1 Reservoir Surfaces Simulation

Reservoir top elevation and reservoir thickness are generated with the GFS at the grid node projections shown in Fig. 6.34 to impose limits on the simulation domains. The variogram model of the correlated reservoir top elevation and thickness were presented above in Eq. (6.2) and Fig. 6.12. The reservoir base is derived from simulated reservoir top and thickness. The resulting elevations and thickness are shown in Fig. 6.35. These boundaries are used to limit the simulation grids in the vertical dimension. The simulation of key geological variables is performed relative to the simulated reservoir top. Later, vertical coordinates z' are adjusted back to original elevations z.

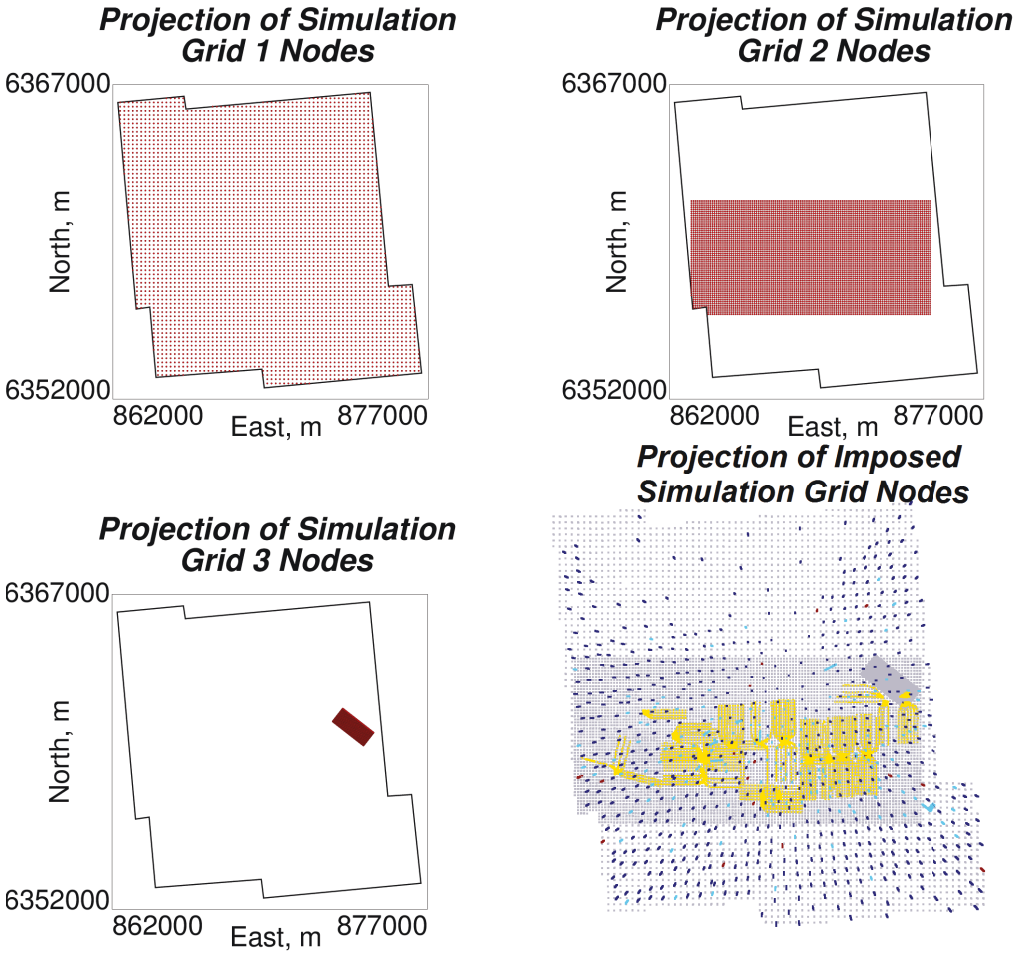


Figure 6.34: Plan view of the simulation grid nodes projections at coarse, medium, and fine resolutions, and together with well trajectories.

6.3.2 Facies Simulation

A realization of facies is generated with SIS according to the variograms in Eq. (6.1) and Figs. 6.8 and 6.9 to get four stationary domains for petrophysical properties simulation. The resulting conditional simulation is presented in Fig. 6.36. Because the SIS method considers only two-point statistics, complex curvilinear features that may be expected in a channelized formation are not prominently seen in the realization [75]. MPS or object-based modeling technique could be

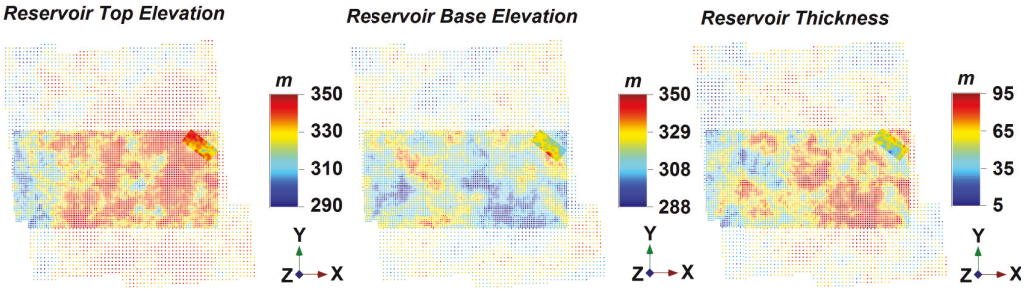


Figure 6.35: Grid-free realizations of the McMurray top elevation, base elevation, and reservoir thickness resolved on three grids.

used to inject these features in the facies model. The facies model could have been conditioned to the seismic data to honor not only the hard facies picks, but also soft data. The seismic data would be converted to the facies probabilities based on the collocated facies and acoustic impedance data, and used as the locally varying mean in the categorical model simulation with SIS [17, 64].

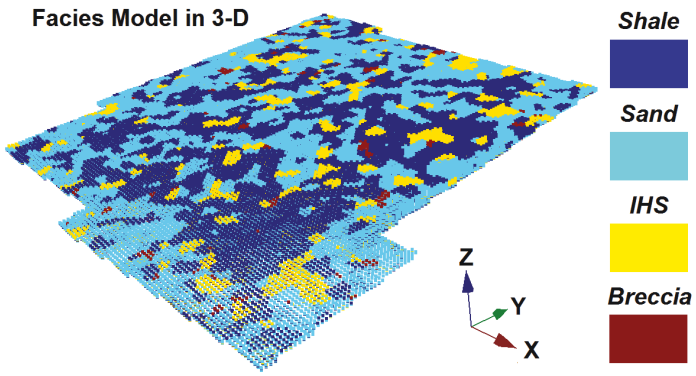


Figure 6.36: Single realization of the categorical facies model for stationary domains definition.

6.3.3 Geological Properties Simulation

The simulation of the porosity, permeability, water saturation, and complimentary acoustic impedance is performed by-facies on three grids of various resolutions at a point scale conditional to point-scale hard data and blocks-scale soft seismic data with the LMC covariance structure shown in Eq. (6.20). The difference in the scale of the seismic data is addressed through the PSB value representation method. The intrinsic co-kriging with data and simulation locations projection is used to assimilate exhaustively sampled acoustic impedance. The simulation is performed in the normal space, prior to which the data values are transformed to the normal space using normal score transformation. As an example, the simulation results for the sand facies over the sparsest simulation domain (grid 1) are shown in Figs. 6.37 and 6.38 in normal scores and in Figs. 6.40 and 6.41 in original units. The univariate and bivariate distributions of the variables are honored in normal space and original units. The variogram model in red line and experimental variograms computed from simulated normal scores resolved on grid 1 in horizontal plane are shown in Fig. 6.39 for this single realization. The variogram reproduction looks well, except the cross-variogram between porosity and water saturation, which could be addressed by very low correlation coefficient (small sill contribution in comparison with sill contributions of other direct- and cross-variograms), what infused computational precision error. Resulting models of the key petrophysical properties are shown in Figs. 6.42 - 6.45 by-facies and for all facies at once. Note the seismic data reproduction in the last figure when all facies are combined together. Because the PSB values of the acoustic impedance are represented at the larger block scale than other data type, and, hence, more continuous, the simulated acoustic impedance in the data region looks also more continuous in comparison with the rest of the model. Similar observation is applied to the rest of the variables: simulated values within the seismic data look more continuous than in the other parts of

the simulation domain, because of the continuous nature of the soft block-scale seismic data. The hard data have been honored.

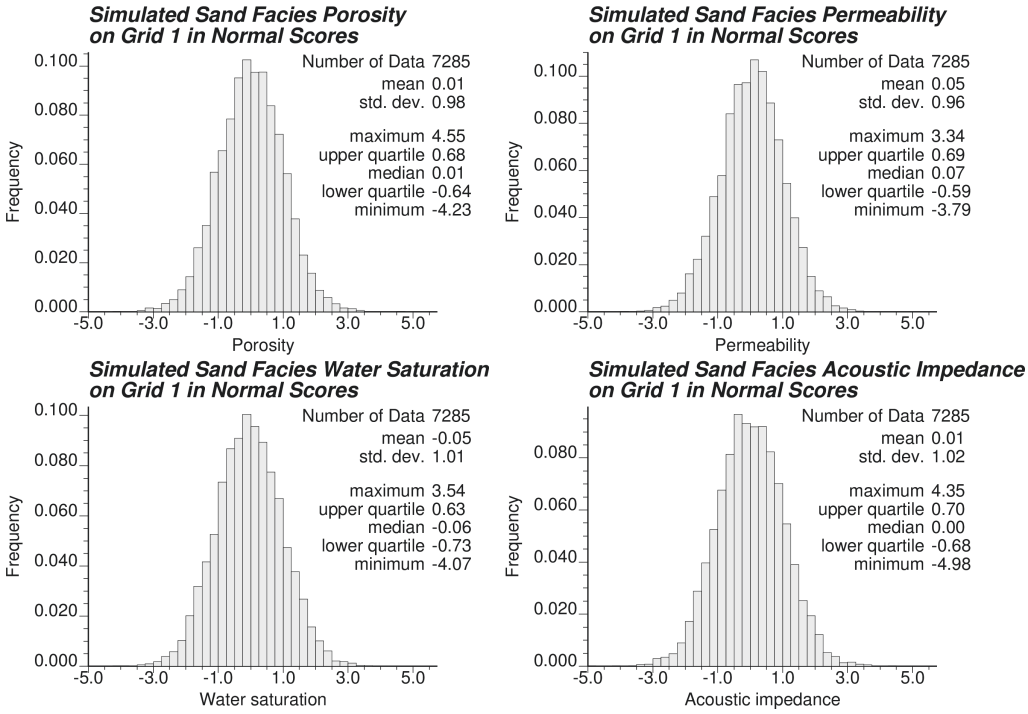


Figure 6.37: Histograms of simulated porosity, permeability, water saturation, and acoustic impedance in normals scores for sand facies over simulation grid 1.

6.3.4 Discussion

The grid-free simulation entitles for being a practical method for petroleum reservoir characterization. In this case study, the developed simulation method was applied in three different circumstances: the probabilities within the wells for calculation of pseudo-log permeability and water saturation values, target reservoir top and base elevations, and geological properties resolved at various resolutions were computed in a grid-free manner with great deal of flexibility of resolving realizations at any simulation node configuration. Therefore, the modeling results

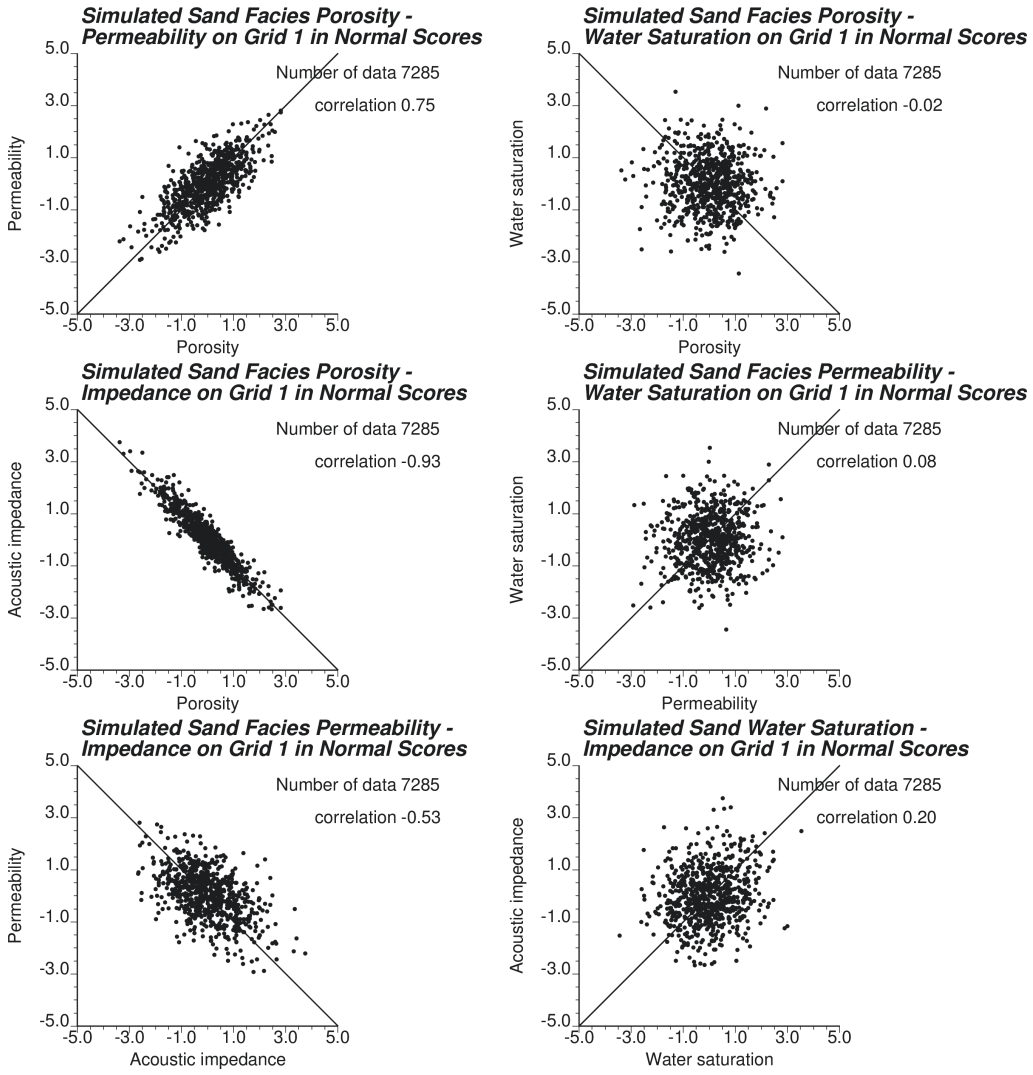


Figure 6.38: Scatter plots of simulated porosity, permeability, water saturation, and acoustic impedance in normals scores for sand facies over simulation grid 1.

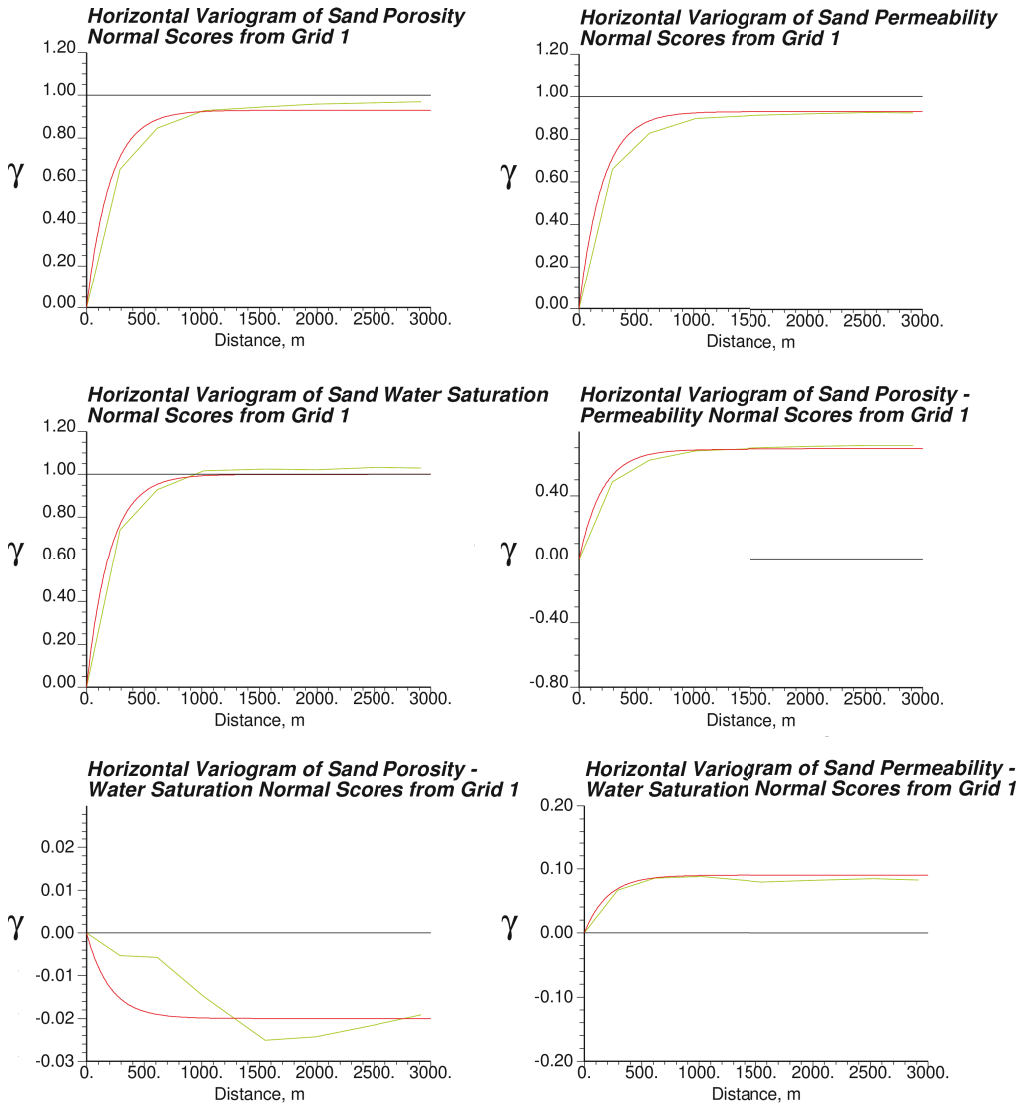


Figure 6.39: Horizontal experimental variograms shown by green lines of simulated geological properties in normal scores for sand facies resolved on grid 1 and target variogram model presented by a red line. The reproduction of cross-variogram between porosity and water saturation is poor, because their correlation is weak in comparison with sill contributions of other variograms.

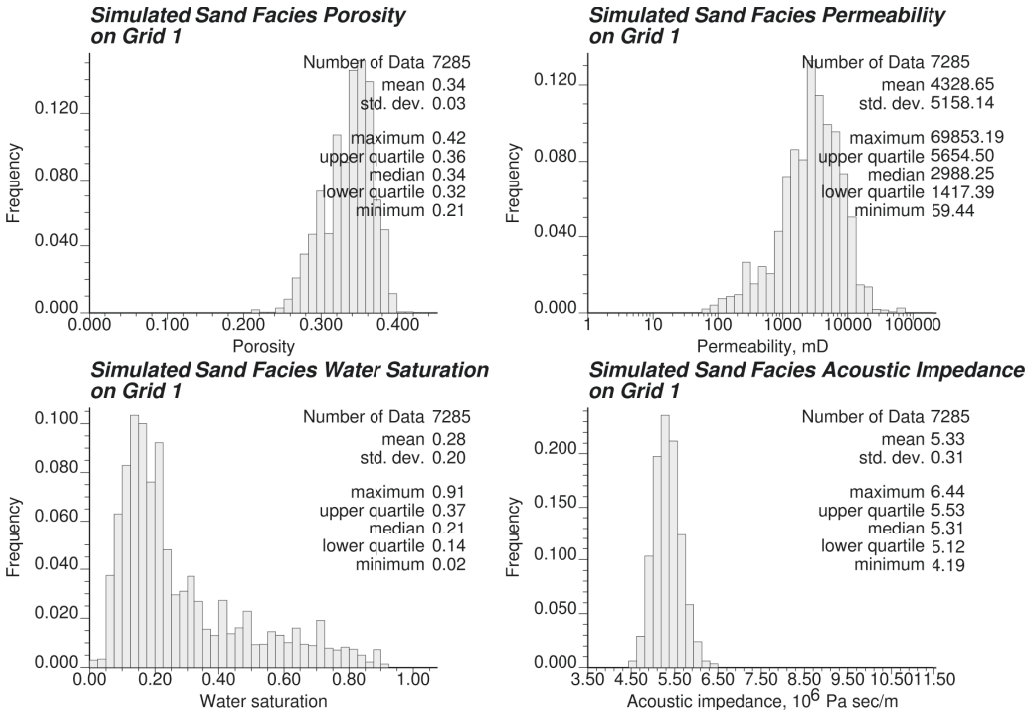


Figure 6.40: Histograms of simulated porosity, permeability, water saturation, and acoustic impedance in original units for sand facies over simulation grid 1.

presented in this chapter proves the potential of the GFS technique for the application to real case studies and one in particular – Firebag Oil Sands Project. The simulation remains consistent regardless of the simulation grid density and orientation, what enables three models built for various purposes being a part of the infinitely resolvable model presented in the polynomial grid-free form. The data values and data distribution are honored in final realizations. The variogram is reproduced quite accurately. Note that most geological variation comes from the difference between facies properties, and not the variation within the facies. The uncertainty could have been assessed by generating multiple realizations and subsequent post-processing.

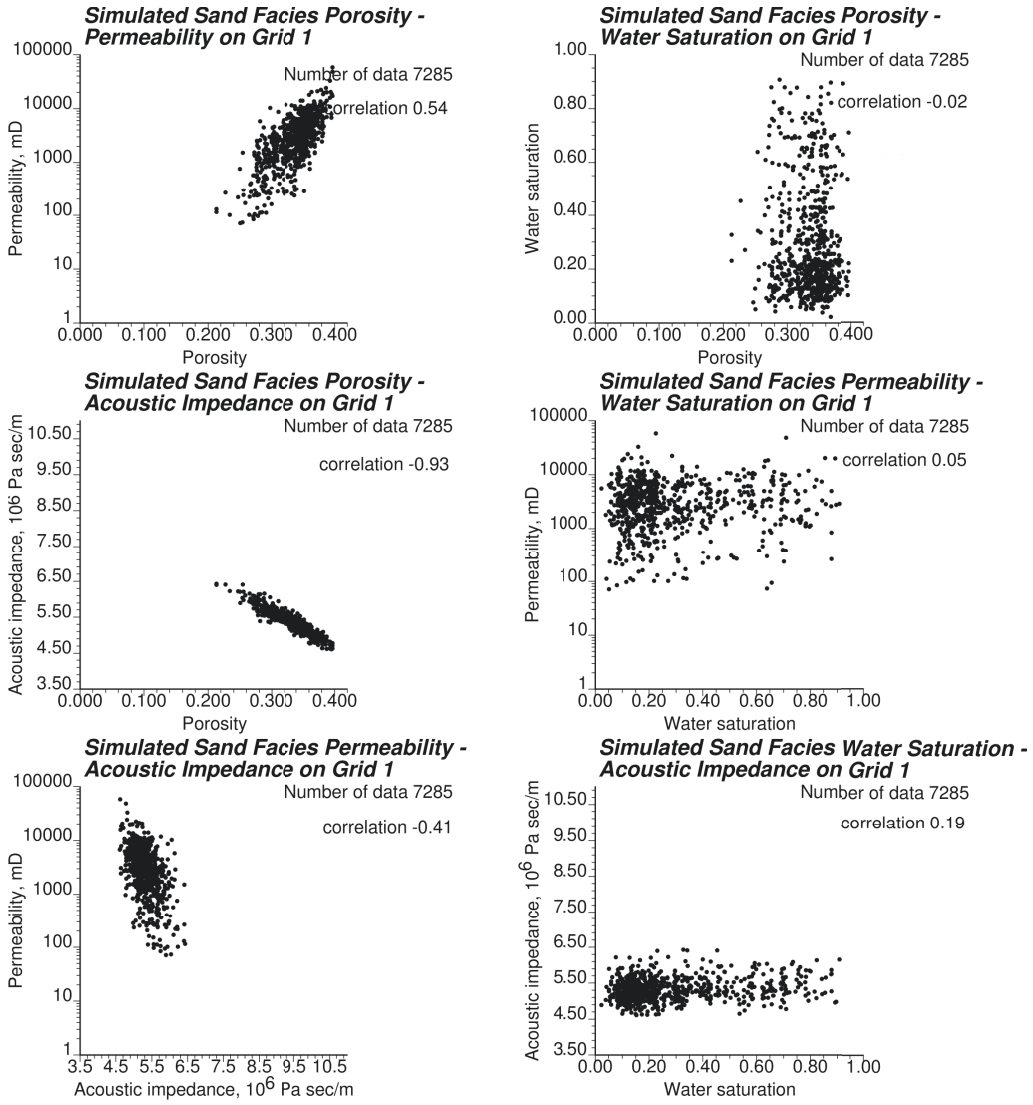


Figure 6.41: Scatter plots of simulated porosity, permeability, water saturation, and acoustic impedance in original units for sand facies over simulation grid 1.

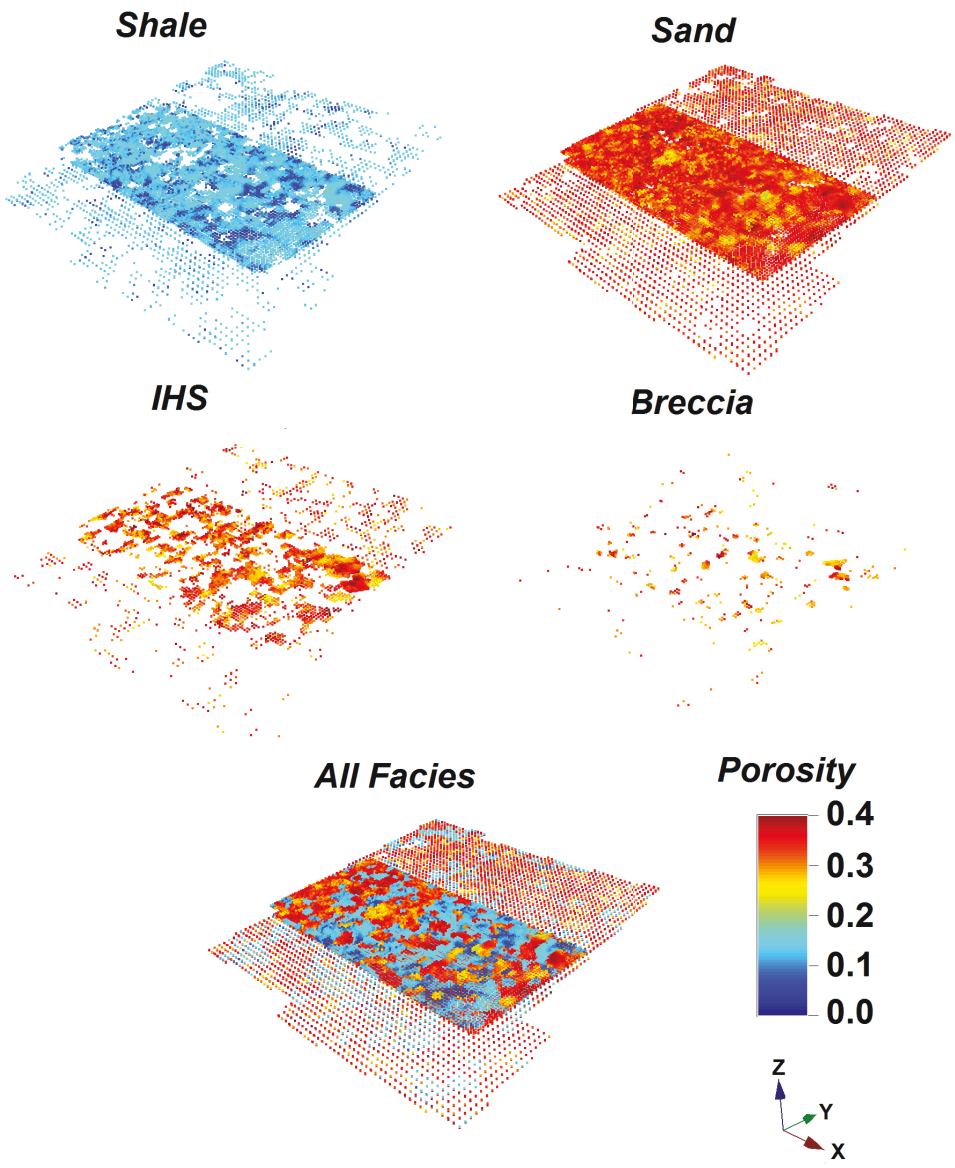


Figure 6.42: Porosity realization for each facies domain separately and together at a point scale.

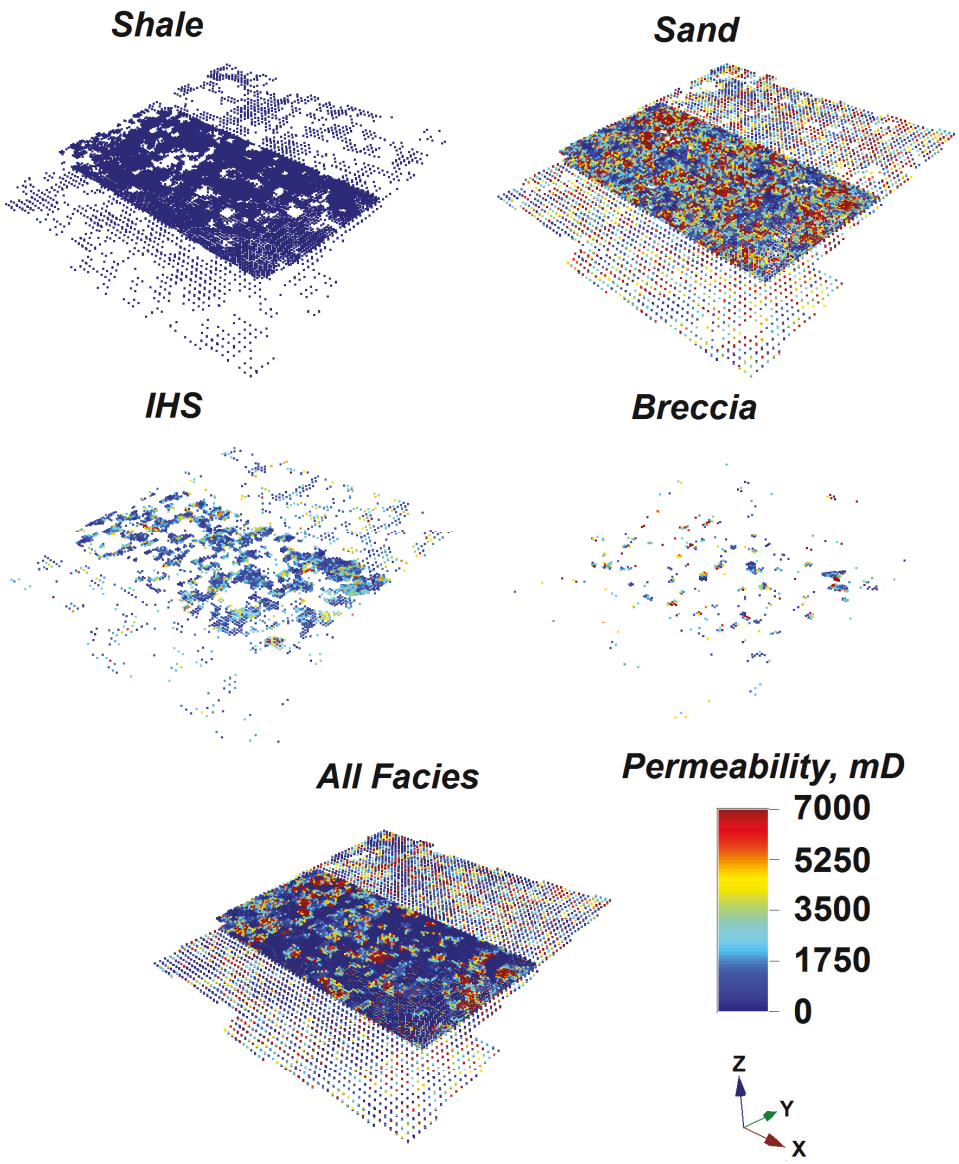


Figure 6.43: Permeability realization for each facies domain separately and together at a point scale.

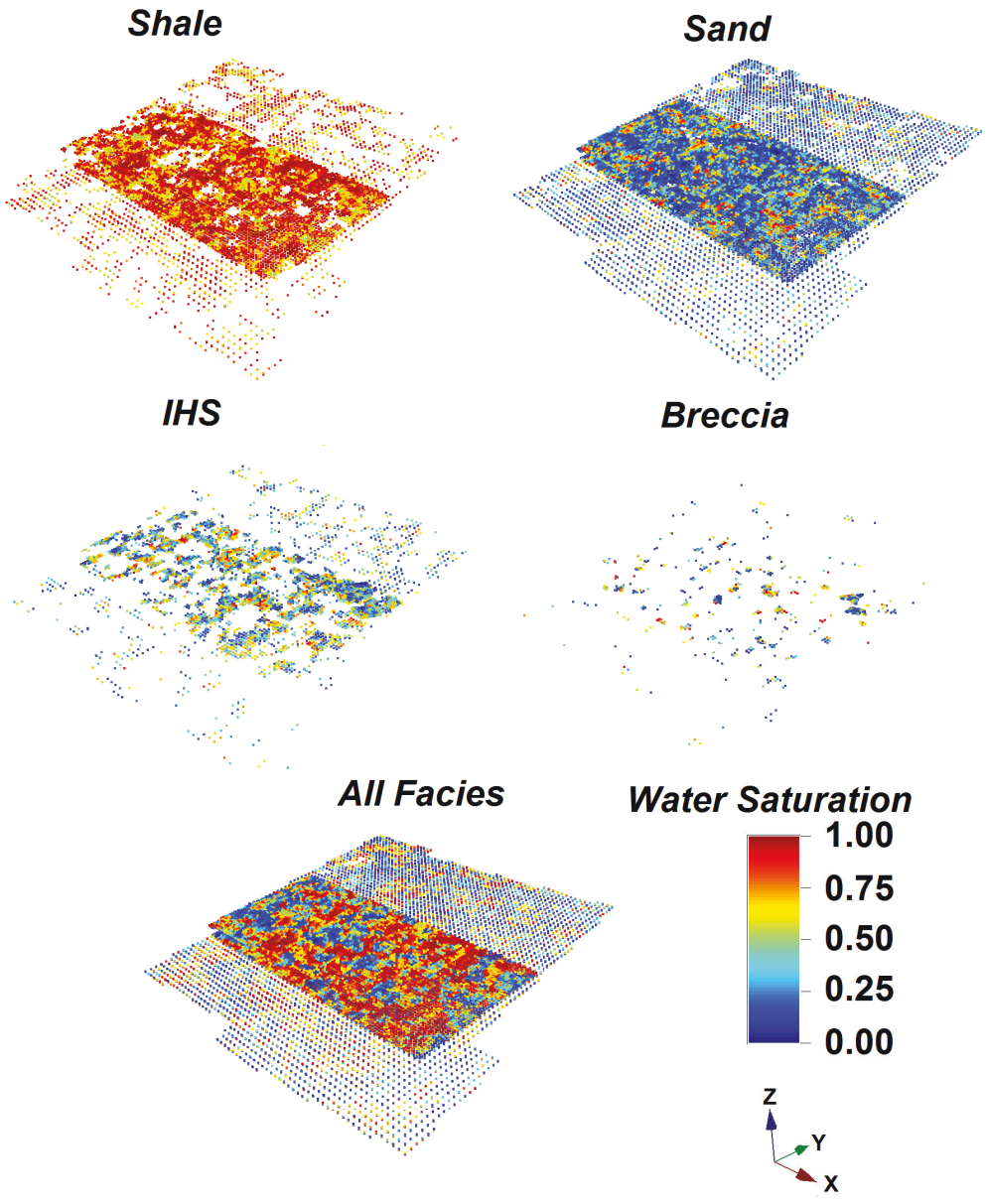


Figure 6.44: Water saturation realization for each facies domain separately and together at a point scale.

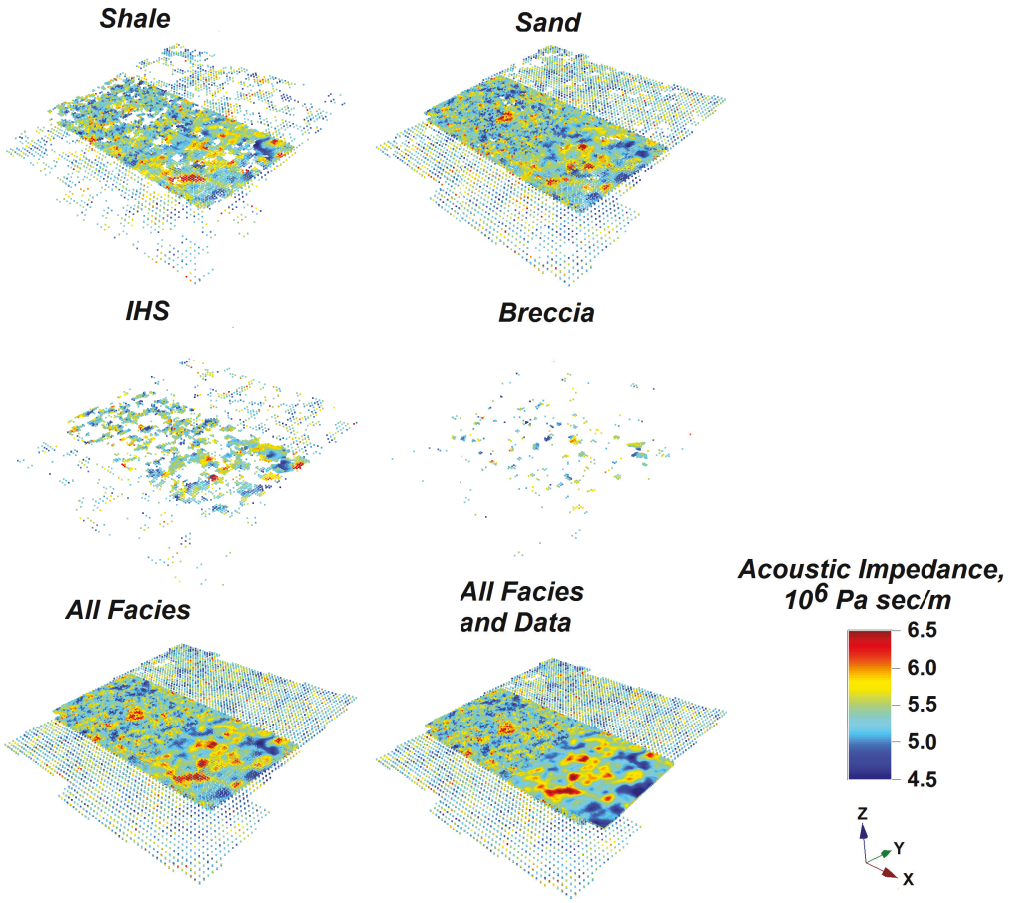


Figure 6.45: Acoustic impedance realization for each facies domain separately and together at a point scale, and conditioning gridded block-scale data.

Chapter 7

Conclusions

The framework of grid-free geostatistical simulation of multivariate properties of a natural phenomena in presence of multiscale data has been developed and demonstrated with numerous 2-D and 3-D synthetic small examples and a real case study. Final remarks, conclusions, and future work of the GFS implementation for research and industrial purposes are summarized in this chapter.

7.1 Contributions

Simulation of the coregionalized variables is an important task for the petroleum, mining, and other industrial projects. Even though most conventional geostatistical methods are inherently grid-free, they are not implemented in this way. When the simulation is performed on a new set of simulation node locations, the results depend on the ordering of simulation or are not consistent with previously simulated values. The presented grid-free simulation approach is a novel idea to express geostatistical simulation as a function of the coordinates of the simulation locations. The regridding of the simulation model, modeling with locally refined areas, simulation on unstructured grids, and the assimilation of additional data is straightforward with the proposed approach.

Therefore, the multivariate geostatistical simulation of properties of natural phenomena in a grid-free manner is the main contribution of this research work. The simulation is conditioned to multiscale data in the form of the point-scale hard data and exhaustively sampled block-scale measurements. The Fourier series decomposition of target input variogram model is performed to preserve spatial structure of the system. The high dimensional simulation is derived from the modified turning bands concept by combining several line processes resolved with the newly developed grid-free Fourier series simulation. The multiple correlated attributes are easily cosimulated with the linear model of coregionalization concept. Thus, the conditional realization of grid-free simulation is expressed as a finite-length series of the weighted cosine functions. The two-step conditional grid-free simulation also ensures that integration of new data preserves previously resolved realizations and changes model locally. The gradual changes would affect regions of the simulation domain only within correlation range from the new data locations. The resulting simulation is infinitely resolvable in space.

Other essential contributions of this thesis can be highlighted as follows.

- The assimilation of gridded block-scale data with point-scale block value representation method is proposed and developed. Frequently, an exhaustively sampled data at a block scale in the form of seismic attributes are available for the conditioning of the geological properties of the natural deposit. However, block-scale data cannot be assimilated into the geostatistical model as provided, because the difference between block values at their boundaries will create artifacts in the GFS realizations. Thus, a linear interpolation algorithm is developed to represent the block values continuously in space.

- The GFS implementation algorithm considers intrinsic cokriging approach for conditioning model to exhaustively sampled data with the intrinsic

variogram formalism. The well established screening assumption allows efficient assimilation of large data sets. Also, the implemented projection approach in the intrinsic cokriging removes edge effects in the realizations caused by the exhaustively sampled data that partially covers the simulation domain.

- The computational cost of the GFS is reduced by several enhancements to the original algorithm. The combination of multiple 1-D line simulations generated by the Fourier series simulation within the turning bands framework for higher dimensional simulation, conditioning based on the cokriging in a dual form, and effective matrix manipulation including block matrix inversion for the exhaustively sampled data assimilation, where only part of the matrix is inverted, provide improvement in the computational time.

- Inherently random white noise with nugget effect variogram is represented as a function of the approximate coordinates of the simulation location. The simulation space is gridded at fine resolution. An index is assigned to each block of the mesh and represents the random number seed in the pseudo-random number generator. Therefore, resulting random numbers are independent of each other, but tied to the particular grid block location in the space.

- A study has been conducted to assess optimal number of regularly spaced point-scale discretization nodes for the accurate representation of the block scale value after their upscaling. It is recommended to use between 4 and 8 discretization nodes for each dimension of a block volume to properly represent associated block-scale value.

The extensively examined GFS method, a successful real 3-D case study, and readily available code `gfsim` allow to simulate properties of the natural

phenomena at industrial scale. The proposed method would improve managerial decisions and make geomodels prepared for multiple purposes consistent with each other and accurate. Regridding and grid refinement of coarse scale models would no longer be issues.

7.2 Limitations

The GFS method is very effective for the simulation of multivariate geological attributes in presence of multiscale data. There are, however, a series of conceptual and computational limitations.

The proposed method considers only stationary domains, i.e., the global mean and covariance structure of the variables under study are assumed constant throughout the simulation domain. Some geological attributes may show a trend, which should be removed before the GFS is applied. The boundaries between stationary domains should be defined clearly by deterministic or stochastic techniques. The multivariate system is modeled with the LMC, which also assumes multivariate stationarity.

A large number of Fourier series coefficients and turning lines have to be used to generate realizations with the required spatial structure and free off artifacts. When a small number of Fourier coefficients are used, the realizations appear too continuous. When a small number of turning lines are used, linear artifacts are present in the realizations. GFS is approximately 5 times slower than conventional SGS.

Another computationally demanding step is the simulation at block scale. The simulation at point-scale nodes and subsequent upscaling to block scale is costly in practice. This drawback is not particularly related to GFS, but would be relevant to any geostatistical simulation method that produces simulation at a point-scale.

7.3 Future Work

The GFS has a great potential to be applied in practice. There are a number of suggestions that could bring the performance of the proposed grid-free simulation to a new level.

The computational time could be improved by a parallel programming, another more efficient decomposition method of the covariance function that requires fewer coefficients, and replacement of turning lines with other technique that allows combining lower dimensional simulations to a larger dimensional simulation. The relationship between number of the Fourier coefficients, correlation range of the modeled variables, and size of the simulation domain could be studied further to understand better the nature of the Fourier coefficients for keeping fewer the most significant ones in the simulation.

The wavelet representation of the simulation would allow to perform upscaling and downscaling in a straightforward manner and condition realizations to multiscale data sets [49]. The wavelet decomposition ensures proper relationship between realizations resolved at different scales and their covariances. Unfortunately, the wavelets are gridded in the current implementation and should be expressed in a grid-free manner to be applied within GFS framework. The GFS could be expanded for simulation of non-stationary systems.

Parallel programming would allow using multiple processors to simulate several realizations at multiple locations simultaneously. The GFS algorithm is inherently parallel in that multiple locations and realizations can be processed independently; unlike sequential simulation algorithms. The two-level code is a first step to this approach. It processes raw data and sets up all required information to be directly used for the grid-free simulation.

The GFS method could be coupled with various geostatistical techniques, such as grid-free simulation of categorical variables, and all reservoir components,

such as structural elements, compartmentalization, geological attributes, trends, etc., could be simulated at once to perform simulation and visualization of the entire reservoir on the fly. The grid-free simulation of categorical variables can be performed with truncated Gaussian simulation (TGS) within GFS paradigm, where simulated continuous random functions are truncated at some thresholds to obtain distribution of categories in space [26].

Decorrelation techniques including principal component analysis and other for multivariate systems could be implemented to better preserve the relationship of the simulated variables in the original units.

Bibliography

[1] Alabert, F., 1987. The Practice of Fast Conditional Simulations Through the LU Decomposition of the Covariance Matrix. *Mathematical Geology*, 19(5), pp. 369 – 386.

[2] Allard, D., D'Or, D. and Froidevaux, R., 2011. An Efficient Maximum Entropy Approach for Categorical Variable Prediction. *European Journal of Soil Science*, 62, pp. 381 – 393.

[3] Anton, H., 1994. *Elementary Linear Algebra. 7th Edition.* New York: John Wiley and Sons, Inc., 650 p.

[4] Babak, O., 2008. *Theoretical and Practical Improvements to Geostatistical Estimation and Simulation. PhD Thesis.* Edmonton: University of Alberta, 231 p.

[5] Barnett, R.M. and Deutsch, C.V., 2013. Imputation of Geologic Data. *Centre for Computational Geostatistics (CCG)*, 15(102), pp. 102-1 – 102-16.

[6] Barnett, R.M., Manchuk, J.G. and Deutsch, C.V., 2014. Projection Pursuit Multivariate Transform. *Mathematical Geosciences*, 46(3), pp. 337 – 359.

[7] Bochner, S. and Chandrasekharan, K., 1949. *Fourier Transforms.* Princeton: Princeton University Press, 219 p.

[8] Bracewell, R.N., 2000. *The Fourier Transform and Its Applications. 3rd Edition.* New York: McGraw Hill, 616 p.

[9] Butler, R.M., 1991. *Thermal Recovery of Oil and Bitumen.* New Jersey: Prentice Hall, 528 p.

[10] Chiles, J.-P. and Delfiner, P., 2012. *Geostatistics. Modeling Spatial Uncertainty. 2nd Edition.* New York: Wiley, 734 p.

[11] Cooley, J.W., Lewis P.A.W. and Welch, P.D., 1969. The Fast Fourier Transform and Its Applications. *IEEE Transactions on Education*, 12(1), pp. 27 – 34.

[12] Coxeter, H.S.M., 1940. Regular and Semi-Regular Polytopes. I. *Mathematische Zeitschrift*, 46(1), pp. 380 – 407.

[13] Davis, J.C., 2002. *Statistics and Data Analysis in Geology. 3rd Edition.* New Jersey: John Wiley and Sons Inc., 656 p.

[14] Davis, M.W., 1987. Production of Conditional Simulations via the LU Triangular Decomposition of the Covariance Matrix. *Mathematical Geology*, 19(2), pp. 91 – 98.

[15] Dehdari, V., 2011. *Review of Geostatistical Simulation Methods for Large Problems. Vol. 11.* Edmonton: Centre for Computational Geostatistics (CCG), 34 p.

[16] Deutsch, C.V., 1992. *Annealing Techniques Applied to Reservoir Modeling and The Integration of Geological and Engineering (Well Test) Data. PhD Thesis.* Stanford: Stanford University, 306 p.

[17] Deutsch, C.V. and Journel, A.G., 1998. *GSLIB: Geostatistical Software Library and User's Guide. 2nd Edition.* New York: Oxford University Press, 369 p.

[18] Deutsch, C.V., 2010. Estimation of Vertical Permeability in the McMurray Formation. *Journal of Canadian Petroleum Technology*, 49(12), pp. 10 – 18.

[19] Devore, J. and Peck, R., 2005. *Statistics: The Exploration and Analysis of Data. 5th Edition*. Belmont: Brooks/Cole, 763 p.

[20] Ellis, D.V. and Singer, J.M., 2008. *Well Logging for Earth Scientists. 2nd Edition*. Dordrecht: Springer, 708 p.

[21] Emery, X., 2004. Testing The Correctness of the Sequential Algorithm for Simulating Gaussian Random Fields. *Stochastic Environmental Research and Risk Assessment*, 18(6), pp. 401 – 413.

[22] Feder, J., 1988. *Fractals*. New York: Plenum Press, 284 p.

[23] http://www.geologic.com/products-services/geoscout [Accessed on August 28, 2014].

[24] Gneiting, T., 1998. Closed Form Solutions of the Two-Dimensional Turning Bands Equation. *Mathematical Geology*, 30(4), pp. 379 – 389.

[25] Goldberger, A.S., 1964. *Econometric Theory*. New York: J. Wiley, 399 p.

[26] Goovaerts, P., 1997. *Geostatistics for Natural Resources Evaluation*. New York: Oxford University Press, 496 p.

[27] Gupta, M.R., Garcia, E.K. and Chin, E., 2008. Adaptive Local Linear Regression With Application to Printer Color Management. *IEEE Transactions on Image Processing*, 17(6), pp. 936 – 945.

[28] Hardy, H.H. and Beier, R.A., 1994. *Fractals in Reservoir Engineering*. New Jersey: World Scientific, 359 p.

[29] Harris, J.W. and Stocker, H., 1998. *Handbook of Mathematics and Computational Science*. New York: Springer, 1028 p.

[30] Hartman, H.L. and Mutmansky, J.M., 2002. *Introductory Mining Engineering. 2nd Edition*. Hoboken, New Jersey: John Wiley and Sons Inc., 584 p.

[31] Hassanpour, R.M. and Deutsch, C.V., 2010. An Introduction to Grid-Free Object-based Facies Modeling. *Centre for Computational Geostatistics (CCG)*, 12(107), pp. 107-1 – 107-6.

[32] Hassanpour, R.M. and Deutsch, C.V., 2010. Grid-Free Object-based Modeling of IHS Facies in the McMurray Formation. *Centre for Computational Geostatistics (CCG)*, 12(203), pp. 203-1 – 203-10.

[33] Hassanpour, R.M. and Deutsch, C.V., 2010. Programs for Grid-Free Object-based Facies Modeling of IHS Sets. *Centre for Computational Geostatistics (CCG)*, 12(413), pp. 413-1 – 413-6.

[34] Huang, S.P., Quek, S.T. and Phoon, K.K., 2001. Convergence Study of the Truncated Karhunen-Loeve Expansion for Simulation of Stochastic Processes. *International Journal for Numerical Methods in Engineering*, 52(9), pp. 1029 – 1043.

[35] Isaaks, E.H. and Srivastava, R.M., 1989. *An Introduction to Applied Geostatistics*. New York: Oxford University Press, 592 p.

[36] Journel, A.G. and Huijbregts, C.J., 1978. *Mining Geostatistics*. New York: Academic Press, 600 p.

[37] Journel, A.G., 1974. Geostatistics for Conditional Simulation of Ore Bodies. *Economic Geology*, 69(50), pp. 673 – 687.

[38] Kenner, H., 2003. *Geodesic Math and How to Use It. 2nd Edition.* Berkeley: University of California Press, 183 p.

[39] Kumar, D., 2006. A Tutorial on Gassmann Fluid Substitution: Formulation, Algorithm and Matlab Code. *Geohorizons*, 1, pp. 4 – 12.

[40] Lantuejoul, C., 2002. *Geostatistical Simulation: Models and Algorithms*. Verlag: Springer, 256 p.

[41] Le Ravalec, M., Noetinger, B. and Hu, L.Y., 2000. The FFT Moving Average (FFT-MA) Generator: An Efficient Numerical Method for Generating and Conditioning Gaussian Simulations. *Mathematical Geology*, 32(6), pp. 701 – 723.

[42] Leuangthong, O. and Deutsch, C.V., 2003. Stepwise Conditional Transformation for Simulation of Multiple Variables. *Mathematical Geology*, 35(2), pp. 155 – 173.

[43] Leuangthong, O., McLennan, J.A. and Deutsch, C.V., 2004. Minimum Acceptance Criteria for Geostatistical Realizations. *Natural Resources Research*, 13(3), pp. 131 – 141.

[44] Luster, G. R., 1985. *Raw Materials For Portland Cement: Applications of Conditional Simulation of Coregionalization. PhD Thesis*. Stanford: Stanford University, 531 p.

[45] Manchuk, J.G., 2010. *Geostatistical Modeling of Unstructured Grids for Flow Simulation. PhD Thesis*. Edmonton: University of Alberta, 198 p.

[46] Mandelbrot, B.B., 1977. *Fractals: Form, Chance, and Dimension*. San Francisco: W.H. Freeman and Company Ltd, 365 p.

[47] Mantoglou, A. and Wilson, J.L., 1982. The Turning Bands Method for Simulation of Random Fields Using Line Generation by a Spectral Method. *Water Resources Research*, 18(5), pp. 1379 – 1394.

[48] Marcotte, D., 2012. Revisiting the Linear Model of Coregionalization. *9th International Geostats Congress - Geostatistics Oslo 2012*, pp. 67 – 78.

[49] Marcotte, D. and Gloaguen, E., 2014. Exact Conditioning of Gaussian Fields on Wavelet Coefficients. *Mathematical Geosciences*, pp. 1 – 24.

[50] Matheron, G., 1971. *The Theory of Regionalized Variables and Its Applications*. Paris: Ecole Nationale Superieure des Mines de Paris, 211 p.

[51] Metropolis, N. and Ulam, S., 1949. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247), pp. 335 – 341.

[52] Myers, D.E., 1982. Matrix Formulation of Co-Kriging. *Mathematical Geology*, 14(3), pp. 249 – 257.

[53] Myers, D.E., 1989. *Vector Conditional Simulation. In: M. Armstrong, ed. Geostatistics: Proceedings of the Third International Geostatistics Congress*. Avignon, France: Kluwer Academic Publishers, pp. 283 – 293.

[54] Neufeld, C., Ortiz, J.M. and Deutsch, C.V., 2005. A Non-Stationary Correction of the Probability Field Covariance Bias. *Centre for Computational Geostatistics (CCG)*, 7(103), pp. 103-1 – 103-11.

[55] Oliver, D.S., 1995. Moving Averages for Gaussian Simulation in Two and Three Dimensions. *Mathematical Geology*, 27(8), pp. 939 – 960.

[56] Oliver, D.S., Reynolds, A.C. and Liu, N., 2008. *Inverse Theory for Petroleum Reservoir Characterization and History Matching*. New York: Cambridge University Press, 394 p.

[57] Pacheco, P., 1996. *Parallel Programming with MPI*. San Francisco: Morgan Kaufmann, 500 p.

[58] Pardo-Iguzquiza, E. and Chica-Olmo, M., 1993. The Fourier Integral Method: An Efficient Spectral Method for Simulation of Random Fields. *Mathematical Geology*, 25(2), pp. 177 – 217.

[59] Pardo-Iguzquiza, E. and Chica-Olmo, M., 1994. Spectral Simulation of Multivariable Stationary Random Functions Using Covariance Fourier Transforms. *Mathematical Geology*, 26(3), pp. 277 – 299.

[60] Pardo-Iguzquiza, E. and Chica-Olmo, M., 1994. SPECSIM: A Program for Simulating Random Fields by An Improved Spectral Approach. *Computers and Geosciences*, 20(4), pp. 597 – 613.

[61] Pardo-Iguzquiza, E. and Dowd, P.A., 2003. SPECSIM2: A Program for Spectral Simulation of Anisotropic Two-Dimensional Random Fields. *Computers and Geosciences*, 29(8), pp. 1013 – 1020.

[62] Press, W.H. Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., 1997. *Numerical Recipes in Fortran 77. 2nd Edition.* Volume 1, Cambridge: Press Syndicate, 448 p.

[63] Pyrcz, M.J., 2004. *Integration of Geologic Information into Geostatistical Models. PhD Thesis.* Edmonton: University of Alberta, 296 p.

[64] Pyrcz, M.J. and Deutsch, C.V., 2014. *Geostatistical Reservoir Modeling. 2nd Edition.* New York: Oxford University Press, 448 p.

[65] Rosenblatt, M., 1952. Remarks on a Multivariate Transformation. *The Annals of Mathematical Statistics*, 23(3), pp. 470 – 472.

[66] Ruan, F. and McLaughlin, D., 1998. An Efficient Multivariate Random Field Generator using The Fast Fourier Transform. *Advances in Water Resources*, 21(5), pp. 385 – 399.

[67] Sarma, P., Durlofsky, L.J. and Aziz, K., 2008. Kernel Principal Component Analysis for Efficient, Differentiable Parameterization of Multipoint Geostatistics. *Mathematical Geosciences*, 40(1), pp. 3 – 32.

[68] Schneider, C.L. and Grobe, M., 2013. Regional Cross-Sections of Devonian Stratigraphy in Northeastern Alberta (NTS 74D, E). *AER/AGS Open File Report 2013-05*, Edmonton: Alberta Energy Regulator, 31 p.

[69] Shamsipour, P., Marcotte, D. and Chouteau, M., 2012. Integrating Multi-scale Parameters Information Into 3-D Stochastic Magnetic Anomaly Inversion. *Geophysics*, 77(4), pp. D85 – D93.

[70] Shinozuka, M. and Deodatis, G., 1991. Simulation of Stochastic Processes by Spectral Representation. *Applied Mechanics Reviews*, 44(4), pp. 191 – 204.

[71] Shinozuka, M. and Jan, C.-M., 1972. Digital Simulation of Random Processes and Its Applications. *Journal of Sounds and Vibration*, 25(1), pp. 111 – 128.

[72] Sneddon, I.N., 1951. *Fourier Transforms*. New York: McGraw-Hill Book Company, Inc., 542 p.

[73] Srivastava, R.M., 1992. Reservoir Characterization With Probability Field Simulation. *SPE Annual Technical Conference and Exhibition*, Washington, DC, US.

[74] Strebelle, S.B. and Journel, A.G., 2001. Reservoir Modeling Using Multiple-Point Statistics. *SPE Annual Technical Conference and Exhibition*, Orleans, Louisiana, US.

[75] Suncor, 2013. Commercial Scheme Approval No 8870 for Firebag. *ERCB Performance Presentation*, Calgary.

[76] Voss, R.F., 1986. Characterization and Measurement of Random Fractals. *Physica Scripta*, T(13), pp. 27 – 32.

[77] Wackernagel, H., 2003. *Multivariate Geostatistics. 3rd Edition.* Berlin: Springer, 388 p.

[78] Warren, J.E. and Price, H.S., 1961. Flow in Heterogeneous Porous Media. *Society of Petroleum Engineers Journal*, 1(3), pp. 153 – 169.

[79] Wenninger, M.J., 1975. *Polyhedron Models for the Classroom. 2nd Edition.* Virginia: The National Council of Teachers of Mathematics, Inc., 43 p.

[80] Wikramaratna, R.S., 1989. ACORN – A New Method for Generating Sequences of Uniformly Distributed Pseudo-Random Numbers. *Journal of Computational Physics*, 83(1), pp. 16 – 31.

[81] Wikramaratna, R.S., 2010. Theoretical and Empirical Convergence Results for Additive Congruential Random Number Generators. *Journal of Computational and Applied Mathematics*, 233(9), pp. 2302 – 2311.

[82] Yao, T., 1998. *Automatic Covariance Modeling and Conditional Spectral Simulation with Fast Fourier Transform. PhD Thesis.* Stanford: Stanford University, 192 p.

[83] Yao, T., 1998. SPECSIM: A FORTRAN-77 Program for Conditional Spectral Simulation in 3D. *Computers and Geosciences*, 24(10), pp. 911 – 921.

[84] Zagayevskiy, Y., 2012. *Constraining 3D Petroleum Reservoir Models to Petrophysical Data, Local Temperature Observations, and Gridded Seismic Attributes with the Ensemble Kalman Filter (EnKF). MSc Thesis.* Edmonton: University of Alberta, 151 p.

Appendix A

Appendix – Description of Fortran Programs

A description and documentation of the Fortran programs prepared specifically for the grid-free simulation paradigm is provided in this Appendix. The presented programs are `ffsim` for grid-free Fourier series simulation, `gfsim` for grid-free simulation with turning lines, `gfsim1` and `gfsim2` for two-level grid-free simulation with turning lines, and `psbv` for point-scale block value representation of the gridded block-scale data.

A.1 Program `ffsim`

A Fortran program called `ffsim` is prepared to perform grid-free Fourier series simulation, which is presented in Section 3.1 of Chapter 3. A default parameter file of the program `ffsim` is shown in Table A.1. The first four lines are the header of the parameter file. The number of the variables to simulate is entered on line 5. Next block is devoted for the data specification. Line 6 is reserved for input data file name of first variable to simulate. The data can be stored in a scattered (option 0) or gridded (option 1) format in the data file. Columns with x, y, and z

coordinates are specified for the scattered data sets. Grid specification is required for the regularly gridded data sets, where number of the data nodes in X, Y, and Z directions, x, y, and z coordinates of the first ordered datum, and spacing between the data nodes in X, Y, and Z directions are needed. Same parameters are entered for any other remaining data files. If the data files are missing, unconditional simulation is performed. The number of the realizations must be specified along with the random number seed on lines 18 and 19, respectively. The grid-free simulation can be performed at scattered simulation nodes (option 0) or on regularly structured grid of simulation nodes (option 1). This option is indicated on line 20. In order to represent realizations at the scattered simulation nodes, the input data file with Cartesian x, y, and z coordinates is required (lines 21 and 22). When realizations are resolved on regularly placed simulation nodes, the grid orientation along with the grid specification are entered on lines 23 – 26. Option for precision of target covariance decomposition with the Fourier series is specified on line 27: option 1 implies 10% precision error, option 2 – 5%, option 3 – 1%, and option 4 would approximate covariance function with 0.1% error. The size of the periodic domain is defined on line 28. Lines 29 – 31 are reserved for the nugget effect mesh specification. The output file name for the realizations of grid-free simulation is specified on line 32. The resulting simulation is presented in normal scores at a point scale. The target variogram model of normal scores of the simulated system in the LMR/LMC form are entered last in the parameter file. The direct- and cross-variogram should be specified for multivariate case. The number of the nested structures, types of the structures, their sill contributions, principal directions of the anisotropy, and variogram ranges are all needed to set up the variogram model in normal units.

Table A.1: Default parameter file of program `ffsim`.

```
                         Parameters for FFSIM
                         ********************

START OF PARAMETERS:
2                        -number of variables
data1.dat                -input file with variable 1 data
1 0                      -   data, scatter/grid (0/1)
2 3 4                    -   if (0):  columns for x, y, z
50 0.5 1.0               -   if (1):  nx, xmin, xsize
50 0.5 1.0               -            ny, ymin, ysize
50 0.5 1.0               -            nz, zmin, zsize
data2.dat                -input file with variable 2 data
1 1                      -   data, scatter/grid (0/1)
2 3 4                    -   if (0):  columns for x, y, z
50 0.5 1.0               -   if (1):  nx, xmin, xsize
50 0.5 1.0               -            ny, ymin, ysize
50 0.5 1.0               -            nz, zmin, zsize
100                      -number of realizations to generate
69069                    -random number seed
0                        -simulation:  grid-free/gridded (0/1)
simnodes.dat             -   if (0):  file with coordinates
1 2 3                    -            columns for x, y, z
0.0 0.0 0.0              -   if (1):  azimuth, plunge, dip
50 0.5 1.0               -            nx, xmin, xsize
50 0.5 1.0               -            ny, ymin, ysize
50 0.5 1.0               -            nz, zmin, zsize
3                        -option for covariance approximation
200.0                    -covariance discretization domain size
5000 0.005 0.01          -nugget effect grid:  nx, xmin, xsize
5000 0.005 0.01          -nugget effect grid:  ny, ymin, ysize
5000 0.005 0.01          -nugget effect grid:  nz, zmin, zsize
gfsim.out                -output file for simulated realizations
1  0.1                   -variogram 1:  nst, nugget
1  0.9  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
        30.0 30.0 30.0   -              a1, a1, a3
1  0.1                   -variogram 12:  nst, nugget effect
1 -0.5  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
        30.0 30.0 30.0   -              a1, a1, a3
1  0.1                   -variogram 2:  nst, nugget
1  0.9  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
        30.0 30.0 30.0   -              a1, a1, a3
```

A.2 Program `gfsim`

A grid-free simulation method GFS based on the Fourier series simulation and turning bands is implemented in the program `gfsim`; default parameter file is shown in Tables A.2 and A.3. This parameter file is similar to the parameter file of program `fssim` with some changes as follows. The weights for the debiased data distribution can be specified in a parameter file for each data file. The column for weight within the data file for simulation of bivariate systems are entered on lines 7 and 21 for variable 1 and 2, respectively. Trimming limits for the data are specified on lines 8 and 22 for the different data sets. The weights are required for accurate normal score transformation, which is also implemented in this code. The option for the NST is specified on line 16 for first data set, and on line 30 for second date set. There are two options how the transformation can be performed. It is either based on the transformation table or kernel approximation of the debiased data CDF as explained in Section 5.1.1 of Chapter 5. The lower and upper tail values (or possible minimum and maximum data values) are specified next along with the power used for the interpolation between lower tail – minimum data and maximum data – upper tail in the data CDF curve. An additional parameter is required for the kernel-based NST: the size of the window used to fit local liner regression model to the CDF – data value pairs. The number of realizations to consider and simulate in this run are specified on line 34. Additional realizations can be generated with other runs of the program as required. The particular realizations to generate are defined on line 35. The option for the number of the turning lines to use in the simulation is also defined. For 2-D modeling, option 1 implies 10 lines, option 2 – 50 lines, option 3 – 100 lines, and option 4 – 200 line processes are used in the simulation. For 3-D modeling, number of the lines is slightly different: option 1 – 12 lines, option 2 – 42 lines, and option 3 – 162 equally spaced lines on a unit 3-D sphere are

used in the simulation. The rest of the parameters remain the same as explained in Section A.1 of Appendix for program `ffsim`. The simulation is performed in normal scores at point scale, but output is reported in original units, when normal score transformation option is specified. Otherwise, the realizations are stored in the output file in the normal scores.

A.3 Program `gfsim1`

The program `gfsim1` performs pre-processing of the data, simulation parameters, and input variogram model, output of which is used in `gfsim2` for grid-free simulation of the realizations at required locations in the space. The `gfsim1` incorporates a big initial portion of the `gfsim`, but no simulation is performed. A default parameter file of `gfsim1` is presented in Tables A.4 and A.5. It is identical to most of the `gfsim` parameter file as explained in Section A.2 of Appendix, except that the dimension of the simulated system should be specified on line 34. The output file name for the pre-processed data is entered on line 43.

A.4 Program `gfsim2`

The program `gfsim2`, a default parameter file of which is presented in Table A.6, performs grid-free simulation using the pre-processed data in the form of output file of program `gfsim1`. Some parts of the parameter file are similar to the parameter file of `gfsim` as explained in Section A.2 of Appendix. The input file generated by the `gfsim1` is specified on line 5. The number of the realizations to generate and indices of these realizations are entered on lines 6 and 7, respectively. The total number of the realizations is stored in the input data file. The simulation is performed either at scattered locations or on a grid as explained in Section A.1 of Appendix for program `ffsim`. The resulting realizations are stored in the output file specified on line 15. This two-level

Table A.2: Default parameter file of program `gfsim`.

```
1-                          Parameters for GFSIM
2-                          ********************
3-
4-   START OF PARAMETERS:
5-   2                      -number of variables
6-   data1.dat              -input file with variable 1 data
7-   1 0 0                  -   data, weight, scatter/grid (0/1)
8-   0.0 1.0e21             -   trimming limits for data
9-   2 3 4                  -   if (0):  columns for x, y, z
10-  0.0                    -   if (1):  angle of data grid
11-  50 0.5 1.0            -            nx, xmin, xsize
12-  50 0.5 1.0            -            ny, ymin, ysize
13-  50 0.5 1.0            -            nz, zmin, zsize
14-  10 10 10              -            discretization
15-  30                     -            number of iterations
16-  1 0                    -   no/yes NST (0/1), table/kernel (0/1)
17-  0 1.0                 -      if (1):  lower tail, power option
18-  1 2.0                 -               upper tail, power option
19-  10.0                   -      if (1) and (1):  averaging window
20-  data2.dat             -input file with variable 2 data
21-  1 0 1                  -   data, weight, scatter/grid (0/1)
22-  0.0 1.0e21             -   trimming limits for data
23-  2 3 4                  -   if (0):  columns for x, y, z
24-  0.0                    -   if (1):  angle of data grid
25-  50 0.5 1.0            -            nx, xmin, xsize
26-  50 0.5 1.0            -            ny, ymin, ysize
27-  50 0.5 1.0            -            nz, zmin, zsize
28-  10 10 10              -            discretization
29-  30                     -            number of iterations
30-  1 0                    -   no/yes NST (0/1), table/kernel (0/1)
31-  0 1.0                 -   if (1):  lower tail, power option
32-  1 2.0                 -            upper tail, power option
33-  10.0                   -   if (1) and (1):  averaging window
34-  10 3                  -number of realizations:  all, simulate
35-  1 5 7                  -   realizations to generate
36-  69069                  -random number seed
```

Table A.3: Default parameter file of program `gfsim`. *Continuation of Table A.2.*

```
37-  0                        -simulation:  grid-free/gridded (0/1)
38-  simnodes.dat             -   if (0):  file with coordinates
39-  1 2 3                    -            columns for x, y, z
40-  0.0 0.0 0.0              -   if (1):  azimuth, plunge, dip
41-  50 0.5 1.0               -            nx, xmin, xsize
42-  50 0.5 1.0               -            ny, ymin, ysize
43-  50 0.5 1.0               -            nz, zmin, zsize
44-  3                        -option for covariance approximation
45-  200.0                    -covariance discretization domain size
46-  3                        -option for number of turning lines
47-  5000 0.005 0.01          -nugget effect grid:  nx, xmin, xsize
48-  5000 0.005 0.01          -nugget effect grid:  ny, ymin, ysize
49-  5000 0.005 0.01          -nugget effect grid:  nz, zmin, zsize
50-  gfsim.out                -output file for simulated realizations
51-  1  0.1                   -variogram 1:  nst, nugget
52-  1  0.9  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
53-         30.0 30.0 30.0    -              a1, a1, a3
54-            0    0    0    -              zonal:  no/yes (0/1)
55-  1  0.1                   -variogram 12:  nst, nugget effect
56-  1 -0.5  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
57-         30.0 30.0 30.0    -              a1, a1, a3
58-            0    0    0    -              zonal:  no/yes (0/1)
59-  1  0.1                   -variogram 2:  nst, nugget
60-  1  0.9  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
61-         30.0 30.0 30.0    -              a1, a1, a3
62-            0    0    0    -              zonal:  no/yes (0/1)
```

implementation of the grid-free simulation algorithm allows resolving simulation at any locations and of any realizations easily and efficiently.

A.5 Program `psbv`

A default parameter file of the program `psbv` for the points-scale block value representation of the regularly gridded block-scale data (Section 4.2.4 of Chapter 4) is presented in Table A.7. The name of the input data file with the gridded

Table A.4: Default parameter file of program `gfsim1`.

```
1-                          Parameters for GFSIM1
2-                          *********************
3-
4-   START OF PARAMETERS:
5-   2                      -number of variables
6-   data1.dat              -input file with variable 1 data
7-   1 0 0                  -   data, weight, scatter/grid (0/1)
8-   0.0 1.0e21             -   trimming limits for data
9-   2 3 4                  -   if (0):  columns for x, y, z
10-  0.0                    -   if (1):  angle of data grid
11-  50 0.5 1.0             -            nx, xmin, xsize
12-  50 0.5 1.0             -            ny, ymin, ysize
13-  50 0.5 1.0             -            nz, zmin, zsize
14-  10 10 10               -            discretization
15-  30                     -            number of iterations
16-  1 0                    -   no/yes NST (0/1), table/kernel (0/1)
17-  0 1.0                  -      if (1):  lower tail, power option
18-  1 2.0                  -               upper tail, power option
19-  10.0                   -      if (1) and (1):  averaging window
20-  data2.dat              -input file with variable 2 data
21-  1 0 1                  -   data, weight, scatter/grid (0/1)
22-  0.0 1.0e21             -   trimming limits for data
23-  2 3 4                  -   if (0):  columns for x, y, z
24-  0.0                    -   if (1):  angle of data grid
25-  50 0.5 1.0             -            nx, xmin, xsize
26-  50 0.5 1.0             -            ny, ymin, ysize
27-  50 0.5 1.0             -            nz, zmin, zsize
28-  10 10 10               -            discretization
29-  30                     -            number of iterations
30-  1 0                    -   no/yes NST (0/1), table/kernel (0/1)
31-  0 1.0                  -   if (1):  lower tail, power option
32-  1 2.0                  -            upper tail, power option
33-  10.0                   -   if (1) and (1):  averaging window
34-  3                      -dimensional size of the simulated system
35-  10                     -number of realizations to consider
36-  69069                  -random number seed
37-  3                      -option for covariance approximation
38-  200.0                  -covariance discretization domain size
39-  3                      -option for number of turning lines
```

Table A.5: Default parameter file of program `gfsim1`. *Continuation of Table A.4.*

```
40-  5000 0.005 0.01          -nugget effect grid:  nx, xmin, xsize
41-  5000 0.005 0.01          -nugget effect grid:  ny, ymin, ysize
42-  5000 0.005 0.01          -nugget effect grid:  nz, zmin, zsize
43-  gfsim.dat                -output file for pre-processed data
44-  1  0.1                   -variogram 1:  nst, nugget
45-  1  0.9  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
46-         30.0 30.0 30.0    -              a1, a1, a3
47-            0    0    0    -              zonal:  no/yes (0/1)
48-  1  0.1                   -variogram 12:  nst, nugget effect
49-  1 -0.5  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
50-         30.0 30.0 30.0    -              a1, a1, a3
51-            0    0    0    -              zonal:  no/yes (0/1)
52-  1  0.1                   -variogram 2:  nst, nugget
53-  1  0.9  0.0  0.0  0.0    -              it, cc, ang1, ang2, ang3
54-         30.0 30.0 30.0    -              a1, a1, a3
55-            0    0    0    -              zonal:  no/yes (0/1)
```

Table A.6: Default parameter file of program `gfsim2`.

```
1-                          Parameters for GFSIM2
2-                          *********************
3-
4-   START OF PARAMETERS:
5-   gfsim.dat              -input file with pre-processed data
6-   3                      -number of realizations to simulate
7-   1 5 7                  -   realizations to generate
8-   0                      -simulation:  grid-free/gridded (0/1)
9-   simnodes.dat           -   if (0):  file with coordinates
10-  1 2 3                  -             columns for x, y, z
11-  0.0 0.0 0.0            -   if (1):  azimuth, plunge, dip
12-  50 0.5 1.0             -             nx, xmin, xsize
13-  50 0.5 1.0             -             ny, ymin, ysize
14-  50 0.5 1.0             -             nz, zmin, zsize
15-  gfsim.out              -output file for simulated realizations
```

block-scale data is defined on line 5. Column for the attribute and grid specification with number of the realizations are entered on next lines 6 – 10. The numbers of the discretization nodes for each dimension are specified on line 11. These discretization nodes are used to discretize the block-scale data in order to find optimal nodal values of the point-scale block values used in the interpolation. The number of the iterations implemented in the nodal values calculation is defined on line 12. Next, line 13 is reserved for the output file with the reported point-scale block values, which are presented either at scattered locations (option 0 on line 14) or on a structured set of nodes (option 1 on line 14). The parameters of the output nodes are presented on lines 15 – 20 for both options.

Table A.7: Default parameter file of program `psbv`.

```
1-                          Parameters for PSBV
2-                          ******************
3-
4-   START OF PARAMETERS:
5-   data.dat               -input file with block data
6-   1                      -   column for attribute
7-   50 0.5 1.0             -   nx, xmin, xsize
8-   50 0.5 1.0             -   ny, ymin, ysize
9-   50 0.5 1.0             -   nz, zmin, zsize
10-  1                      -   number of realizations
11-  10 10 10               -number of discretization nodes
12-  30                     -number of iterations
13-  psbv.out               -output file for point-scale block data
14-  0                      -   output:  grid-free/gridded (0/1)
15-  simnodes.dat           -   if (0):  file with coordinates
16-  1 2 3                  -            columns for x, y, z
17-  0.0 0.0 0.0            -   if (1):  azimuth, plunge, dip
18-  50 0.5 1.0             -            nx, xmin, xsize
19-  50 0.5 1.0             -            ny, ymin, ysize
20-  50 0.5 1.0             -            nz, zmin, zsize
```