

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ v Praze  
Fakulta stavební  
K132 - Katedra mechaniky



## **Popis morfologie heterogenních materiálů pomocí náhodných polí odvozených z obrazové analýzy**

*David Šilhánek*

Studijní program: Stavební inženýrství  
Studijní obor: Pozemní stavitelství  
Vedoucí práce: Doc. Ing. Anna Kučerová, Ph.D.

Plzeň, Duben 2021

**Vedoucí práce:**

Doc. Ing. Anna Kučerová, Ph.D.

K132 - Katedra mechaniky

Fakulta stavební

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ v Praze

Thákurova 7

166 29 Prague 6 - Dejvice

Česká Republika

# Prohlášení

Čestně prohlašuji, že jsem tuto práci vypracoval samostatně pouze za poskytnutých konzultací osob uvedených v této práci. Veškeré zdroje, ze kterých jsem čerpal, jsem uvedl v části bibliografie.

V Plzni, Duben 2021

.....  
David Šilhánek

# Abstrakt

Předmětem této práce je studie popisu konstrukce náhodných polí odvozených z obrazové analýzy. Konkrétně se zabývá především autokovariančními funkcemi z literatury běžně používanými pro tyto účely. Pro tvorbu náhodného pole je zde využit Karhunen-Loèveho rozvoj. Práce je napsána v českém jazyce.

**Klíčová slova:** Náhodné pole, dvoubodová pravděpodobnost, kovarianční jádro, Karhunenovo-Loèveho rozvoj

# Abstract

The purpose of this thesis is a study of constructing random fields based on image analysis. More specifically, there are studied autocorrelation functions known from literature which are widely used for this purpose. Random fields are generated using the Karhunen-Loève expansion. Unfortunately, this thesis is written in Czech, so this whole abstract is pointless.

**Keywords:** Random fields, two-point probability, covariance function, Karhunen-Loève expansion

# Seznam tabulek

2.1	Koeficienty aproximace pro Gaussovskou $L_{opt}$ . $L$ - čtvercová inkluze. $L_x$ - obdélníková inkluze ve směru konstantní velikosti, $L_y$ - obdélníková inkluze ve směru proměnné velikosti . . . . .	16
2.2	Koeficienty aproximace pro exponenciální $L_{opt}$ . $L$ - čtvercová inkluze. $L_x$ - obdélníková inkluze ve směru konstantní velikosti, $L_y$ - obdélníková inkluze ve směru proměnné velikosti . . . . .	16
2.3	Přehled variant zpětného generování náhodného pole . . . . .	23

# Seznam obrázků

2.1	(a) Dvoubodová pravděpodobnost pro objemové zastoupení inkluze $c^{(1)} = 0.1$ a velikosti inkluze 5x5 viz (d), (b) dvoubodová pravděpodobnost pro $c^{(1)} = 0.1$ , inkluze 10x10, struktury (e), (c) Dvoubodová pravděpodobnost ekvivalentní s (b)	5
2.2	Průběh gama funkce $\Gamma(x)$ a její převrácené hodnoty $\frac{1}{\Gamma(x)}$ , $x \in \mathbb{R}$	8
2.3	Reálná část Besselových modifikovaných funkcí (a) prvního druhu a (b) druhého druhu	10
2.4	Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze $c^{(1)} = 0, 1$ (a) inkluze 5x5, (b) inkluze 10x10	11
2.5	Průběh absolutní chyby pro Gaussovské (a) a exponenciální (b) jádro pro inkluze 10x10 - viz obr. 2.1	12
2.6	Růst průměrné optimální korelační délky $L_x$ v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussovské (a) a exponenciální (b) jádro	12
2.7	Růst průměrné minimální chyby (sumy čtverců chyb) v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussovské (a) a exponenciální (b) jádro	13
2.8	(a) Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze $c^{(1)} = 0, 3$ inkluze 5x5, (b) příslušná doména	13
2.9	Optimální korelační délka pro Gaussovské jádro s obdélníkovou inkluzí v závislosti na její velikosti a objemovém zastoupení - (a) $L_{x,opt}$ (b) $L_{y,opt}$	14
2.10	Aproximace hodnot $L_{opt}$ pro exponenciální jádro (u Gaussovského jádra je trend podobný). Aproximace (a) rovinou (2.22), (b) hyperbolickým paraboloidem (2.23), (c) hyperbolickým paraboloidem (2.24).	14
2.12	Konvergence vlastních čísel (a) a odhadnuté chyby (b) na doméně 50x50 pro velikost inkluze 5x5	19
2.13	Vlastní tvary exponenciálního Jádra, doména 50x50 px, inkluze 3x12 px	19
2.14	Vlastní tvary z PCA, doména 50x50 px, inkluze 3x12 px	20
2.15	Vlastní tvary zjištěné pomocí $S_2$ , doména 50x50 px, inkluze 3x12 px	20
2.16	Porovnání konvergence odhadnu chyby v závislosti na velikosti inkluze pro $\lambda_i$ zjištěné první popsanou metodou	20
2.17	Vykreslení $\xi$ vůči sobě a distribuční funkce (a,b,c) pro PCA, (d,e,f) pro posun. obr.	21
2.18	Porovnání generovaných obrázků 50x50 px, pro čtvercové inkluze, (a) originální obr. (b) Gaussovské jádro 50 $\phi$ (c) Gaussovské jádro 150 $\phi$ , (d) Gaussovské jádro 500 $\phi$ , (e) jádro z $S_2$ 50 $\phi$ , (f) jádro z $S_2$ 150 $\phi$ , (g) jádro z $S_2$ 500 $\phi$	23

2.19	Porovnání generovaných obrázků 50x50 px, pro obdélníkové inkluze 3x12 px (a) originální obrázek (b) exp. jádro, 40 $\phi$ , (c) exp. jádro, 250 $\phi$ , (d) varianta SHIFT-H, 40 $\phi$ , (e) varianta SHIFT-H, 250 $\phi$ , (f) varianta PCA, 40 $\phi$ , (g) varianta PCA, 250 $\phi$ , (g) varianta PCA, 500 $\phi$ . . . . .	24
2.20	Konvergence chyby při porovnání vůči vzorku pro $c^{(1)} = 0, 1$ na doméně 50x50 px, velikost inkluze (a) 3x3 px (b) 3x12 px . . . . .	25
2.21	Konvergence chyby při porovnání vůči souboru pro $c^{(1)} = 0, 1$ na doméně 50x50 px, velikost inkluze (a) 3x3 px (b) 3x12 px . . . . .	26



# Obsah

<b>Abstrakt</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>Seznam tabulek</b>	<b>vi</b>
<b>Seznam obrázků</b>	<b>vii</b>
<b>1 Úvod</b>	<b>1</b>
<b>2 Popis nehomogenního materiálu</b>	<b>2</b>
2.1 Reprezentace mikrostruktury materiálu . . . . .	3
2.2 Dvoubodová pravděpodobnost . . . . .	4
2.3 Kovarianční jádra . . . . .	6
2.3.1 Gama funkce . . . . .	8
2.3.2 Besselova funkce . . . . .	8
2.3.3 Numerická studie optimálních korelačních délek . . . . .	10
2.4 Karhunenovo–Loèveho rozvoj . . . . .	17
2.4.1 Vlastní čísla a vlastní tvary kovariančních jader . . . . .	18
2.4.2 Volba náhodných proměnných $\xi(\omega)_i$ v KLR . . . . .	21
2.4.3 Hodnocení výsledků náhodného generování . . . . .	22
<b>3 Závěr</b>	<b>27</b>
<b>A Použité kódy a komentář</b>	<b>28</b>
A.1 Náhodné generování binárního obrázku . . . . .	28
A.2 Dvoubodová pravděpodobnost . . . . .	30
A.3 KLR . . . . .	31
A.3.1 Hlavní skript . . . . .	33
<b>Bibliografie</b>	<b>40</b>

# Kapitola 1

## Úvod

V technické praxi se velice často setkáváme s nestejnorodými materiály, ať se už jedná o kompozity či pouze více fází stejného média. V mikroskopickém měřítku lze téměř veškeré materiály prohlásit za nehomogenní a je to právě mikrostruktura zodpovědná za makroskopické chování každého vzorku.

Dosud nejběžnějším způsobem modelování heterogenních materiálů v inženýrské praxi je homogenizace a využití bezpečnostních součinitelů, které mají pokrýt případné náhodné výkyvy v chování materiálu. V posledních letech se však prosazuje i jiný směr, který se snaží o modelování nahodilého chování materiálu pomocí náhodných polí. Snahou je tedy modelovat přímo stochastické vlastnosti materiálu a následným fyzikálním výpočtem<sup>1</sup> (vedení tepla, mechanické modely, transport vlhkosti etc.) získat pravděpodobnostní rozdělení odezvy.

Předem je zřejmé, že pro některé praktické úlohy je tento postup zbytečně náročný a složitý. Uplatnění tedy nenajde pro aplikace, jakými jsou například masivní ocelové stavební konstrukce nebo kvalitně vypálený keramický střep. Naopak u kompozitních materiálů s více fázemi, rozptýlenou výztuží či například u oprav historických staveb, kde můžeme najít znační rozptyl vlastností stavebních materiálů, stejně jako mnohdy nepravidelné zdění, statistický model skýtá značný potenciál oproti odhadu shodných vlastností pro celou konstrukci. Náhodná pole mohou reprezentovat jak mikrostrukturu (např.: pevná fáze, pór), tak makroskopické složení materiálu (např.: kamenivo, výztužné vlákno, cementový tmel). Je však potřeba mít na paměti, že při nesprávné volbě vstupních parametrů tento komplexnější postup může vést k mnohem většímu odchýlení, než by vedlo klasické zprůměrování vlastností s předpokladem homogenity.

Jako možnost, která by usnadnila získání vstupních údajů, se jeví právě obrazová analýza. Předpokladem je, že máme k dispozici nějaký reprezentativní vzorek (či sérii mnoha vzorků), ze kterého můžeme vyčíst zákonitost, jíž se řídí stochastické rozmístění fází v materiálu (např. pro náhodné generování nových vzorků).

Obsahem této práce je studium několika možných postupů pro tvorbu náhodného pole s využitím obrazové analýzy. Pro psaní nezbytných funkcí a skriptů bylo zvoleno prostředí MATLAB R2020b, neboť má implementovanou rozsáhlou řadu funkcí, je optimalizováno pro práci s maticemi a vektory a umožňuje snadno generovat grafický výstup a tedy i rychlou kontrolu výsledků<sup>2</sup>.

---

<sup>1</sup>Těmito modely se v práci nebudeme zabývat. Jejich následná aplikace je nezbytná, aby mělo celé počínání smysl. Tyto listy se věnují především hodnocení samotného náhodného pole.

<sup>2</sup>Jak bude zmíněno i později, některé části tohoto postupu jsou velmi výpočetně náročné, proto by v případě praktického použití bylo vhodné užít programovací jazyk umožňující větší kontrolu práce s pamětí, jakým je např. C/C++.

# Kapitola 2

## Popis nehomogenního materiálu

Pro postihnutí a následnou reprezentaci stochastického rozložení fází v materiálu lze využít řadu funkcí, označovaných jako *statistické deskriptory*. Ty nám umožní kvantifikovat náhodný proces výskytu zvolené fáze v daném místě struktury. Některé deskriptory prezentované v [1] si zde blíže uvedeme.

Prvním zástupcem je ***n*-bodová funkce pravděpodobnosti (n-point probability function)**. Jedná se o skupinu deskriptorů, které vyjadřují pravděpodobnost, že *n* libovolně zvolených bodů bude ležet ve stejné fázi<sup>1</sup>. Tato práce se blíže věnuje právě speciálnímu případu dvoubodové pravděpodobnosti v sekci 2.2, která je výhodná pro popis binárního média, které v této práci uvažujeme. N-bodová pravděpodobnost je definována následující rovnicí:

$$S_n^{i_1, i_2, \dots, i_n}(X_1, X_2, \dots, X_n) = \left\langle \prod_{j=1}^n \chi^{i_j}(x_j, \alpha) \right\rangle, \quad (2.1)$$

- kde:

$\chi^{i_j}(x_r, \alpha)$  je operátor, který přiřadí:

- \* 1 - pokud  $x_j \in D^{i_j}(\alpha)$ , kde  $D^{i_j}(\alpha)$  je oblast zaplněná fází *i*,
- \* 0 - pokud  $x_j \notin D^{i_j}(\alpha)$ ,

$\langle \dots \rangle$  značí průměr výše popsaných operací,  
 $i_j$  značí *j*-tou fázi,  
 $x_j$  zastupuje *j*-tý bod v uspořádané *n*-tici bodů,  
*n* je počet bodů.

Zjednodušeně tyto funkce udávají, jaká je pravděpodobnost, že uspořádaná *n*-tice bodů splní předepsané podmínky (každý bod padne do požadované fáze) kdekoli na doméně.

Dalším statistickým deskriptorem je kupříkladu ***úsečková funkce (linear-path function)***. Na rozdíl od *n*-bodové (resp. dvoubodové) pravděpodobnosti nesleduje splnění

---

<sup>1</sup>Respektive obecněji v konkrétní uspořádané *n*-tici fází.

podmínky předepsané fáze pouze pro příslušnou část úsečky. Zapsáno následovně:

$$L^{i_1, i_2, \dots, i_n}(\mathbf{r}) = \left\langle \prod_{j=1}^n \zeta^{i_j}(\mathbf{r}) \right\rangle, \quad (2.2)$$

• kde:

$\zeta^{i_j}$  je operátor, který přiřadí:

- \* 1 - pokud  $\exists \delta r \subset r : \delta r \subset D^{i_j}(\alpha)$ , kde  $D^{i_j}(\alpha)$  je oblast zaplněná fází  $i_j$ ,
- \* 0 - pokud  $\nexists \delta r \subset r : \delta r \subset D^{i_j}(\alpha) \vee \exists \delta r \subset r : \delta r \subset D^{i_k}(\alpha)$ , kde fáze  $i_k \notin \{i_1, i_2, \dots, i_n\}$ ,

$\langle \dots \rangle$  značí průměr výše popsaných operací,

$\mathbf{r}$  je vektor určený libovolnými dvěma body média,

$i_1, i_2, \dots, i_n$  jsou požadované fáze.

Nejčastěji je používána ve formě  $L^i(r)$ <sup>2</sup>, kdy vzhledem k (2.2) vyjadřuje pravděpodobnost, že zvolená úsečka celá leží ve fázi  $i$ , čímž obsahuje velmi hrubou informaci o spojitosti fáze.

Jak jistě každého napadne, vlastnosti těchto funkcí budou záviset na vlastnostech domény, kterou popisujeme. Přesněji, velmi záleží na tom, zda je daná doména *statisticky homogenní, izotropní*. U statisticky homogenního média jsou fáze rovnoměrně distribuované po celé (nekonečné) doméně. Jinými slovy, objemové zastoupení fáze výrazně nezávisí na zvoleném výřezu a proto i dosud zmíněné deskriptory nebudou závislé na konkrétní poloze vektoru (resp. bodů). Je-li médium statisticky izotropní, funkce nezávisí ani orientaci zvolených vektorů.

Existuje mnoho dalších statistických (a fyzikálních) deskriptorů, jakými jsou např. "tětivová funkce" (chord-length density function), funkce korelace povrchů (surface correlation functions) nebo třeba zobecnění *linear-path function* na libovolnou křivku. Tyto jsou mnohdy složité na implementaci, případně neposkytují informace potřebné pro účel této práce, proto se jimi více nebudeme zabývat.

Jak bylo prezentováno v [2],  $L^i(r)$  na rozdíl od  $S_2^i(x_1, x_2)$  klade důraz na tvar inkluze.  $S_2^i(x_1, x_2)$  na druhou stranu krom této informace nese i údaje o tom, jak jsou fáze rozmístěny vůči sobě na větších vzdálenostech. Proto tento deskriptor použijeme pro další práci. Před popisem samotné dvoubodové pravděpodobnosti v sek. 2.2 se ještě krátce zmíníme o tom, jak je struktura reprezentována v této práci.

## 2.1 Reprezentace mikrostruktury materiálu

Pro účely této práce je struktura reprezentována pomocí dvou fází. Pracujeme tedy na binárním médiu, kde 0 reprezentuje černou a 1 bílou barvu. Ačkoliv v některých velmi specifických případech není nezbytně nutné, je přirozené doménu nějakým způsobem diskretizovat. Velmi oblíbený způsob je FEM (*final element method - metoda konečných*

<sup>2</sup> $L^{i,j}(r)$  by vyjadřovala pravděpodobnost, že úsečka kříží právě fáze  $i, j$ ,  $L^{i,j,k}(r)$ , že kříží právě fáze  $i, j, k$  etc. v libovolném pořadí.

*prvků*), který je v kombinaci s náhodnými poli prezentován např. v [3], avšak vzhledem k prostředí MATLAB, kde je obrázek reprezentován maticí pixelů, je pro nás mnohem jednodušší ponechat toto rozdělení.

Inkluze (bílá barva) zde uvažujeme pouze dvojího tvaru, sice čtvercového či obdélníkového s jednotnou velikostí v rámci daného vzorku, shodně orientované. Důvodem je, že takto jsou jednoduché na implementaci, lze jednoduše upravit jejich rozměry a přitom aproximují "skutečné tvary" (např. rozptýlené ho plniva) bez ztráty kontrolovatelnosti výsledků z důvodu složité vnitřní geometrie.

Nutno uznat, že reálnou mikrostrukturu (popř. makrostrukturu) toto zjednodušení nepopisuje, nicméně cílem v tuto chvíli není test na reálných datech, nýbrž studium chování a vhodnosti algoritmu pro takové užití v budoucnosti.

V případě práce s reálnými vzorky je potřeba nejprve obrázek převést (v našem případě), do binární podoby. Nejjednodušším způsobem je využití prahových metod (*Thresholding*). Obrázek nejprve převedeme do černo-bílé podoby (z tříkanálového RGB tak získáme pouze jednokanálový) a nyní nastavíme hodnotu, nad níž vrátíme 1 a pod níž 0. Tyto metody samozřejmě mohou být sofistikovanější a nastavovat práh automaticky (Otsuova metoda) či dělit doménu více prahy. Možností vhodnou především pro více než dvě domény by bylo též "manuální" přiřazení hodnot či za pomoci externího preprocesoru. Tento postup zde nahradíme náhodným generováním obrázku, které, jak už bylo zmíněno, zatím pro naše účely bohatě postačí. Ve chvíli, kdy máme k dispozici takto reprezentovanou strukturu, lze začít se statistickým popisem rozložení inkluzí.

## 2.2 Dvoubodová pravděpodobnost

Funkcí dvoubodové pravděpodobnosti (*two-point probability function*) rozumíme předpis, kterým charakterizujeme míru pravděpodobnosti výskytu stejné fáze od jistého bodu v daném směru a vzdálenosti. Takto jsme schopni poměrně dobře charakterizovat náhodnou strukturu o dvou fázích. Médium o více než dvou fázích můžeme popisovat buď volbou konkrétních dvojic fází, do nichž mají body padnout, nebo lépe, pomocí více dvoubodových pravděpodobností, kdy pro každou fázi zavedeme samostatnou funkci  $S_2^i(x_1, x_2)$  [1]. Jak celá dvoubodová pravděpodobnost funguje je jasnější z její formální definice [2], kterou dostaneme zjednodušením (2.1):

$$S_2^{i,j}(x_1, x_2) = \langle \chi^i(x_1, \alpha) \chi^j(x_2, \alpha) \rangle, \quad (2.3)$$

• kde:

$\chi^{i,j}(x_r, \alpha)$  je operátor viz (2.1),  
 $\langle \dots \rangle$  značí průměr výše popsanych operací,  
 $x_1$  a  $x_2$  jsou dva libovolné body.

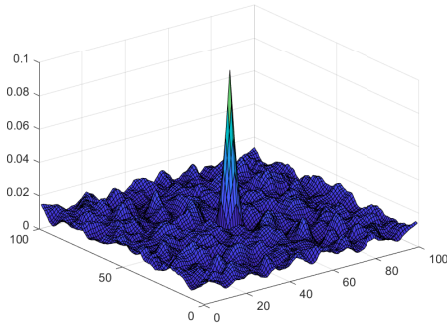
Už zde je vidět, že dvoubodovou pravděpodobnost bude se zachováním diskretizace pomocí pixelů výhodné konstruovat pomocí matice. Funkci dvoubodové pravděpodobnosti, pro periodickou doménu lze také definovat následující rovnicí [2], zde již v podobě upravené

pro náš případ diskrétního výpočtu:

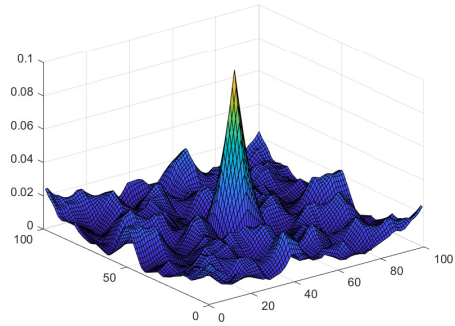
$$S_2^i(x, y) = \frac{1}{n \cdot m} \sum_{x_1=0}^{n-1} \sum_{y_1=1}^{m-1} \chi^i(x_1, y_1) \chi^i((x_1 + x) \% n, (y_1 + y) \% m), \quad (2.4)$$

• kde:

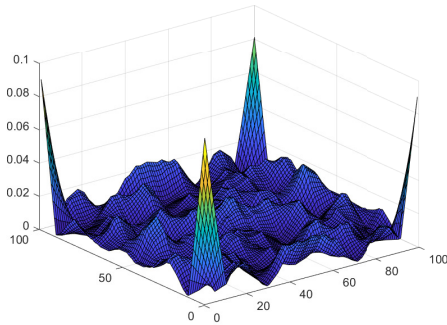
$\%$  je modulo (zbytek po dělení),  
 $n$  a  $m$  jsou rozměry obdélníkové oblasti, respektive počet řádků a sloupců matice,  
 $x_1$  a  $y_1$  jsou souřadnice počátečního bodu,  
 $\chi^i$  značí operátor viz rovnice (2.3) .



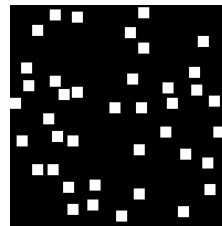
(a)



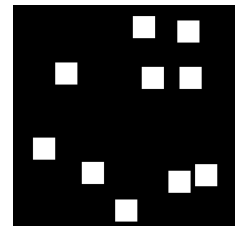
(b)



(c)



(d)



(e)

Obrázek 2.1: (a) Dvoubodová pravděpodobnost pro objemové zastoupení inkluze  $c^{(1)} = 0.1$  a velikosti inkluze 5x5 viz (d), (b) dvoubodová pravděpodobnost pro  $c^{(1)} = 0.1$ , inkluze 10x10, struktury (e), (c) Dvoubodová pravděpodobnost ekvivalentní s (b)

Tento způsob je sice transparentní a přímočarý, ale poměrně výpočetně náročný. Ovšem za předpokladu periodické domény, lze využít výrazně efektivnější výpočet pomocí Fourierových transformací [2]:

$$S_2^{i,j}(x_1, x_2) = \frac{1}{n \cdot m} IDFT\{DFT\{\chi^i(x_1, x_2)\} \overline{DFT\{\chi^j(x_1, x_2)\}}\}, \quad (2.5)$$

- kde:

$n$ a $m$	jsou rozměry obdélníkové oblasti, respektive počet řádků a sloupců matice,
$DFT$	značí diskrétní dopřednou Fourierovu transformaci,
$IDFT$	značí diskrétní inverzní Fourierovu transformaci,
$\dots$	symbolizuje komplexně sdruženou hodnotu,
$\chi^i$	značí operátor viz rovnice (2.1) .

Na obrázku 2.1 jsou graficky prezentovány dvoubodové pravděpodobnosti pro konkrétní struktury. Jak si můžeme povšimnout v obrázku 2.1c, zajímavá místa jsou koncentrována v rozích zájmové oblasti. Neboť ale uvažujeme periodické médium, lze matice přeuspořádat (viz podkapitola 2.3.3) a dostaneme tak rovnocennou informaci v přehlednějším tvaru. Zároveň je to jeden ze způsobů, jak si poradit s tím, že je doména ve skutečnosti neperiodická, často se s těmito problémy vyrovnává i pomocí váhových tlumících funkcí [4] u zpracování pomocí série výřezů.

## 2.3 Kovarianční jádra

Kovarianční jádra nebo též kovarianční funkce jsou zásadní pro popis a generování náhodných polí.

V literatuře [5], [6] se běžně uvádí následující kovarianční funkce:

- Gaussovske kovarianční jádro

$$C_G(X, X') = \sigma_\lambda^2 \cdot e^{-\left(\frac{x-x'}{\sqrt{2}L_x}\right)^2 - \left(\frac{y-y'}{\sqrt{2}L_y}\right)^2}, \quad (2.6)$$

- exponenciální kovarianční jádro

$$C_E(X, X') = \sigma_\lambda^2 \cdot e^{-\left|\frac{x-x'}{L_x}\right| - \left|\frac{y-y'}{L_y}\right|}, \quad (2.7)$$

- kde:

$\sigma_\lambda^2$  je rozptyl příslušné náhodné veličiny,  
 $L_x$  a  $L_y$  jsou příslušné volené korelační délky,

\* *Pod korelační délkou rozumíme vzdálenost dvou bodů, v níž už nejsou vzájemně korelovány. To lze definovat následovně [3], [6]:*

$$\forall |X - X'| \geq L : cov(X, X') \leq \frac{\sigma_\lambda^2}{e} \quad (2.8)$$

$X$  a  $X'$  jsou dva libovolně zvolené body o souřadnicích  $X = [x, y]$  a  $X' = [x', y']$ .

S těmito jádry se i nejčastěji pracuje [3], [5]. Avšak je potřeba si uvědomit, že se jedná o uměle zvolené funkční předpisy, které nemusí dostatečně přesně reflektovat skutečnou

strukturu a navíc jejich přesnost velmi závisí na korelačních délkách, které v tomto případě hrají roli voleného parametru bez jednoznačného fyzikálního významu, který by ulehčil jejich odhad. Numerická studie a diskuze k volbě těchto parametrů se nachází v podkapitole 2.3.3.

Výrazně přesnější je kovarianční funkce založená na dvoubodové pravděpodobnosti [1], [5], kterou lze pohodlně pro binární případ získat jako:

$$C_1(X, X') = (S_2^{(1)}(X, X') - (c^{(1)})^2) \cdot (\kappa^{(1)} - \kappa^{(2)})^2, \quad (2.9)$$

• kde:

$S_2^{(1)}(X, X')$	je hodnota funkce dvoubodové pravděpodobnosti,
$c^{(1)}$	je objemové zastoupení sledované fáze,
$\kappa^{(1)}$ a $\kappa^{(2)}$	jsou hodnoty případné fyzikální veličiny pro každou fázi.

Na rozdíl od výše uvedených jader má tato funkce přímou návaznost na strukturu zkoumaného obrazu a je tak možné její použití pro validaci funkcí (2.6) a (2.7) výše zmíněných spolu s optimalizací příslušných korelačních délek. Porovnání výše uvedených funkcí je (pro optimalizované korelační délky) je vyobrazeno na obr. 2.4.

Ovšem obě používané funkce, jak Gaussova, tak exponenciální, jsou pouze speciálním případem výrazně obecnějšího Matérnova modelu (správněji zřejmě Whittleova-Matérnova modelu viz [7], ale běžně se používá pouze označení Matérnoův model/funkce, kterého se zde podržím).

Matérnovu kovarianční funkci lze definovat následujícím předpisem [8]:

$$C_M(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \sqrt{2\nu} \frac{d}{\rho} \right)^\nu K_\nu \left( \sqrt{2\nu} \frac{d}{\rho} \right), \quad (2.10)$$

• kde:

$\sigma^2$	je rozptyl fyzikální veličiny,
$\Gamma(\nu)$	je <i>gamma funkce</i> viz podkap. 2.3.1,
$K_\nu$	je <i>modifikovaná Besselova funkce druhého druhu</i> viz podkap. 2.3.2,
$d$	je vzdálenost (proměnná),
$\nu$ a $\rho$	jsou volené parametry modelu.

Na rozdíl od rovnic (2.6) a (2.7) se zde vyskytuje pouze jedna proměnná vzdálenosti, avšak i Matérnoův model samozřejmě lze rozšířit do více dimenzí. Pozorný čtenář již zřejmě vytušil, že parametr  $\rho$  je analogický ke korelační délce  $L$ . Ovšem stupeň Besselovy funkce  $\nu$  již na první pohled tak explicitní funkci nemá. V tuto chvíli přenechme k uvěření, že pro  $\nu = \frac{1}{2}$  Matérnoův model odpovídá exponenciálnímu kovariančnímu jádru, pro  $\nu \rightarrow \infty$  pak přejde na Gaussovské jádro [7].

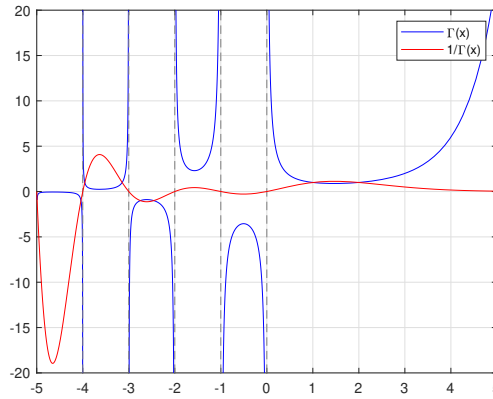


### 2.3.1 Gama funkce

Pod pojmem (*Eulerova*) *gama funkce* rozumíme zobecnění faktoriálu do množiny  $\mathbb{C}$  a tedy i do  $\mathbb{R}$ . Definována je integrálním vztahem:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt, \quad (2.11)$$

Rovnici (2.11) lze definovat pro  $z \in \mathbb{R}^+$ , nebo zavést její analytické prodloužení do  $\mathbb{C} - \mathbb{Z}_\nu^-$ , kde pak i funkční hodnoty mají reálnou i imaginární složku. Na obrázku 2.2 je pro lepší představu vykreslena gama funkce pro reálnou osu a i průběh její převrácené hodnoty, neboť s ní pracuje Matérnovo kovarianční jádro.



Obrázek 2.2: Průběh gama funkce  $\Gamma(x)$  a její převrácené hodnoty  $\frac{1}{\Gamma(x)}$ ,  $x \in \mathbb{R}$

Občas se lze rovněž setkat s *pí funkcí*, která je oproti *gama funkci* pouze posunutá o hodnotu 1 na reálné ose.

$$\Gamma(z+1) = \Pi(z) = \int_0^\infty t^z e^{-t} dt, \quad (2.12)$$

Pro  $n \in \mathbb{N}$  tedy platí:

$$\begin{aligned} \Gamma(n) &= (n-1)! \\ \Pi(n) &= n! \end{aligned}$$

### 2.3.2 Besselova funkce

Besselovými funkcemi nazýváme takové funkce, které jsou řešením diferenciálních rovnic (2.13) a (2.14) a definované na  $\mathbb{C}$ . Besselovy funkce jsou prvního až třetího druhu, "běžné" a modifikované. Zde se zaměřím pouze na popis používané *Besselovy modifikované funkce druhého druhu* a potřebných souvislostí, detailnější informace lze nalézt např. v [9]–[12].

Besselova diferenciální rovnice:

$$z^2 \frac{\delta^2 w(z)}{\delta z^2} + z \frac{\delta w(z)}{\delta z} + (z^2 - \nu^2) w(z) = 0, \quad (2.13)$$

modifikovaná Besselova diferenciální rovnice:

$$z^2 \frac{\delta^2 w(z)}{\delta z^2} + z \frac{\delta w(z)}{\delta z} - (z^2 + \nu^2) w(z) = 0, \quad (2.14)$$

- kde:

$z$             proměnná  $z \in \mathbb{C}$ ,  
 $w(z)$         je hledaná funkce a  
 $\nu$             nazýváme trochu nešťastně řád (lépe stupeň) Besselovy rovnice,  $\nu \in \mathbb{C}$ .

Řád Besselovy rovnice je velmi důležitý parametr, který ovlivňuje výslednou podobu řešení. Obecně lze zapsat řešení ať už (2.13) či (2.13) jako lineární kombinaci dvou *Besselových funkcí*. Dále uvedu pouze řešení pro modifikovanou rovnici, řešení pro (2.13) jsou analogická a konkrétní funkce lze najít ve výše citované literatuře.

- Pokud je  $\nu \in \mathbb{C} - \mathbb{N}$ :

$$w(z) = c_1 I_\nu(z) + c_2 I_{-\nu}(z), \quad (2.15)$$

- pokud je  $\nu \in \mathbb{N}$ :

$$w(z) = c_1 I_\nu(z) + c_2 K_{-\nu}(z), \quad (2.16)$$

- kde:

$c_1$  a  $c_2$         jsou libovolné konstanty,  
 $\nu$             je řád Besselovy rovnice (který je voleným parametrem Matérnova modelu),  
 $I_\nu$  a  $I_{-\nu}$         jsou *Besselovy modifikované funkce prvního druhu* vypočtené jako:

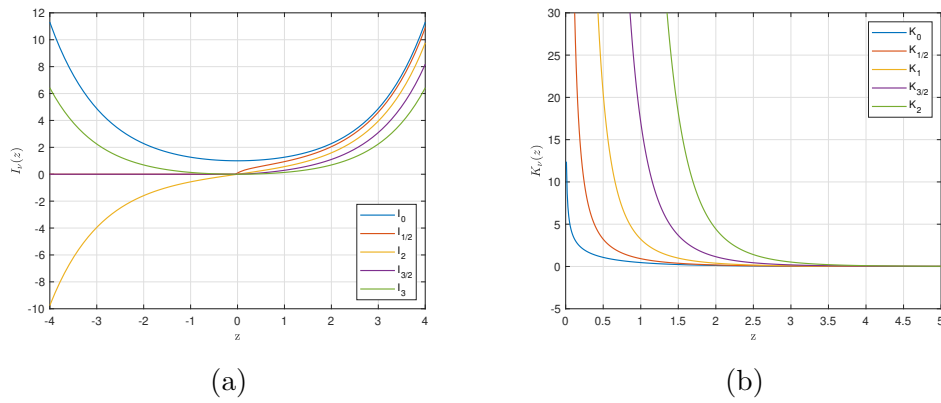
$$I_\nu(z) = \left(\frac{z}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{1}{k! \Gamma(\nu + k + 1)} \left(\frac{z^2}{4}\right)^k, \quad (2.17)$$

$K_\nu$         je *Besselova modifikovaná funkce druhého druhu* vypočtená dle:

$$K_\nu(z) = \frac{\pi}{2} \frac{I_{-\nu}(z) - I_\nu(z)}{\sin(\nu\pi)}, \quad \nu \in \mathbb{C} - \mathbb{N} \quad (2.18)$$

$$K_\nu(z) = \lim_{\nu \rightarrow n} \frac{\pi}{2} \frac{I_{-\nu}(z) - I_\nu(z)}{\sin(\nu\pi)}, \quad \nu = n, n \in \mathbb{N}. \quad (2.19)$$

V prvním případě jsou  $I_\nu$  a  $I_{-\nu}$  lineárně nezávislé, což však neplatí v okamžiku, kdy  $\nu \in \mathbb{N}$ . Proto přichází do hry právě *Besselova modifikovaná funkce druhého druhu*: I samotná *Besselova modifikovaná funkce druhého druhu* je závislá na tom, zda řád rovnice  $\nu$  je celé číslo. Vypočte se z *Besselovy modifikované funkce prvního druhu* dle vztahů (2.17) a (2.18), resp. (2.19). Na obr. 2.3 je vyobrazena reálná část Besselových modifikovaných funkcí pro lepší představu o jejich chování.



Obrázek 2.3: Reálná část Besselových modifikovaných funkcí (a) prvního druhu a (b) druhého druhu

### 2.3.3 Numerická studie optimálních korelačních délek

V této části je provedena a okomentována numerická studie optimálních korelačních délek pro Gaussovskou a exponenciální (dle formulí (2.6) a (2.7)) kovarianční funkci, přesnost modelů a vliv na chybu pro dva základní případy. Kromě optimálních korelačních délek byla sledovaným parametrem i chyba, počítaná jako suma čtverců odchylek umělých kovariančních jader vůči kovarianční funkci počítané z dvoubodové pravděpodobnosti dle rovnice (2.9).

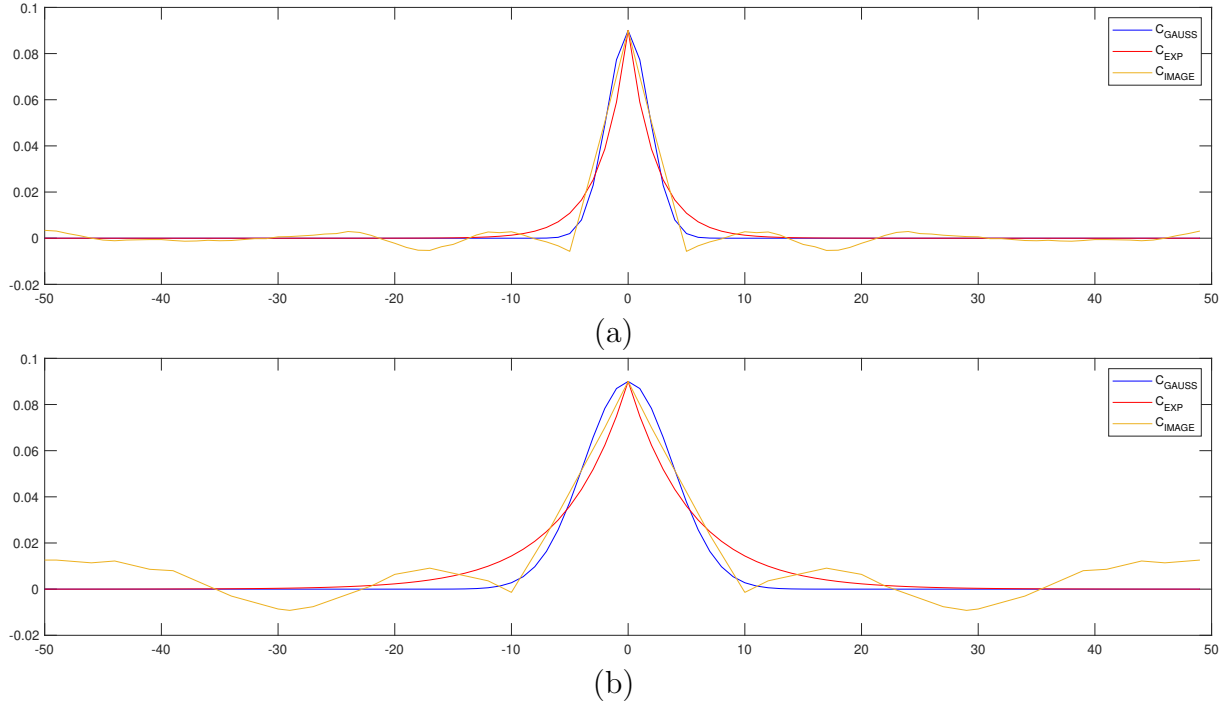
Proměnnými byly kromě objemového zastoupení inkluze (značeno  $c^{(1)}$ ) také rozměr inkluze v px. Rozměr domény, na které byly tyto parametry sledovány, byl pro účely studie zvolen  $100 \times 100$  px. V prvním případě mají inkluze čtvercový tvar, v druhém obdélníkový.

Ještě než si zde uvedeme výsledky samotné studie, zaměřím se na stručný popis chování kovariančních funkcí.

V tomto případě byla předpokládána periodická struktura, což má za následek koncentrace vyšších hodnot pravděpodobnosti (a tedy i z ní počítané kovariance) v rozích matice. Ovšem Gaussovské ani exponenciální jádro ze své povahy periodické není a snaha napodobit jím skutečnou kovarianční matici zjištěnou z obrázku je v tomto případě nevhodná. Vliv periodického chování lze lehce potlačit přeuspořádáním periodické matice dvoubodové pravděpodobnosti tak, že se rozdělí na čtvrtiny a její rohové prvky se sesadí k sobě (díky periodicitě přenáší stále stejnou informaci). Vznikne tak hlavní hrot ve středu oblasti, který rychle klesá a na zbytku domény vzniknou již výrazně nižší fluktuace.

Nadále se využije sudosti Gaussovské a exponenciální funkce (viz obr 2.4), které se optimalizují již na takto přeuspořádané matici, čímž jsou tyto funkce s kovarianční funkcí výrazně lépe porovnatelné. Obě tato jádra vesměs vykryjí větší či menší část středového vrcholu, pak ale rychle konvergují k nule. Z toho plyne, krom faktu, že přeuspořádání nemá významný vliv na výpočet samotné korelační délky, že uměle volená funkce vždy bude zatížena jistou mírou chyby, neboť není schopna pokrýt fluktuaci mimo maximum.

Korelační délka pak udává, jak moc bude v příslušném směru protažen Gaussovský nebo exponenciální klobouk, tedy šířku maximálního vrcholu. Nyní je již zřejmá důležitost správné volby korelační délky, zvolíme-li korelační délku příliš malou, funkce dostatečně nevykryje hlavní vrchol, pokud ji ale naopak nadhodnotíme, funkce bude klesat příliš pomalu oproti realitě. Jak je patrné z obrázku 2.5, chyba s vyšší volenou korelační délkou oproti optimální roste velmi rychle.



Obrázek 2.4: Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze  $c^{(1)} = 0, 1$  (a) inkluze 5x5, (b) inkluze 10x10

Pro účely této studie je chybou míněn rozdíl kovariančních jader vztažených ke kovariančnímu jádru zjištěnému z dvoubodové pravděpodobnosti a sice absolutní, popř. jako chyba nejmenších čtverců:

$$ERR_{abs} = \sum_{i=1}^n \sum_{j=1}^m |C_{G/E} - C_I|, \quad (2.20)$$

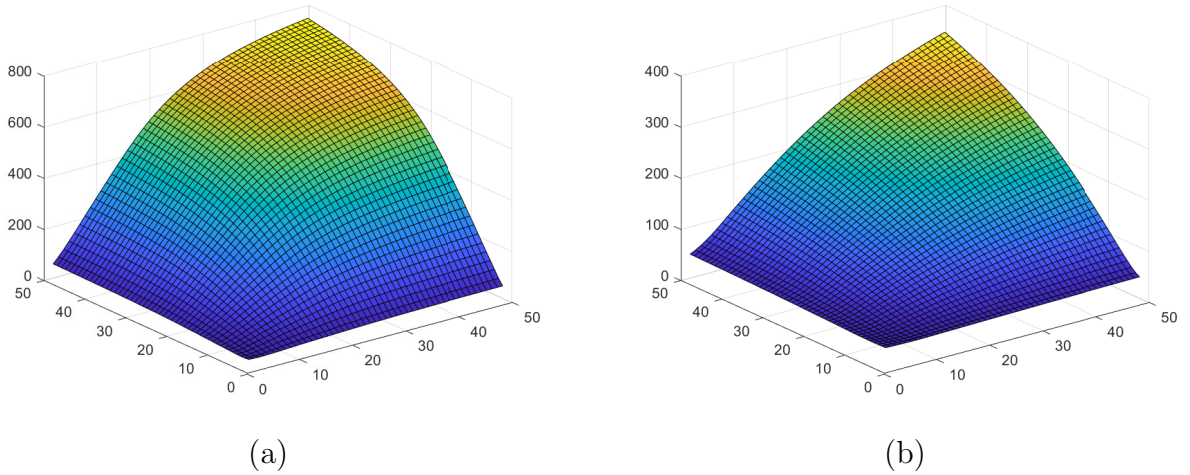
$$ERR_{sqr} = \sum_{i=1}^n \sum_{j=1}^m (C_{G/E} - C_I)^2, \quad (2.21)$$

• kde:

$m$  a  $n$  jsou rozměry domény,  
 $C_G, C_E, C_I$  jsou kovarianční jádra viz (2.6), (2.7), (2.9).

Rozhodně se tak vyplatí volit délky spíše nižší. Jako méně náchylný na špatný odhad se rovněž ukazuje exponenciální model oproti Gassovskému. Jeho minimální absolutní chyba je sice o malinko větší, ale při volbě korelační délky mimo optimum výrazně pomaleji roste, neboť exponenciální funkce klesá rychleji a při špatném (příliš vysokém) odhadu tak rozdíl mezi fluktuacemi kovariance z obrázku a modelovou funkcí není tak velký (i když chyba stále roste velmi rychle).

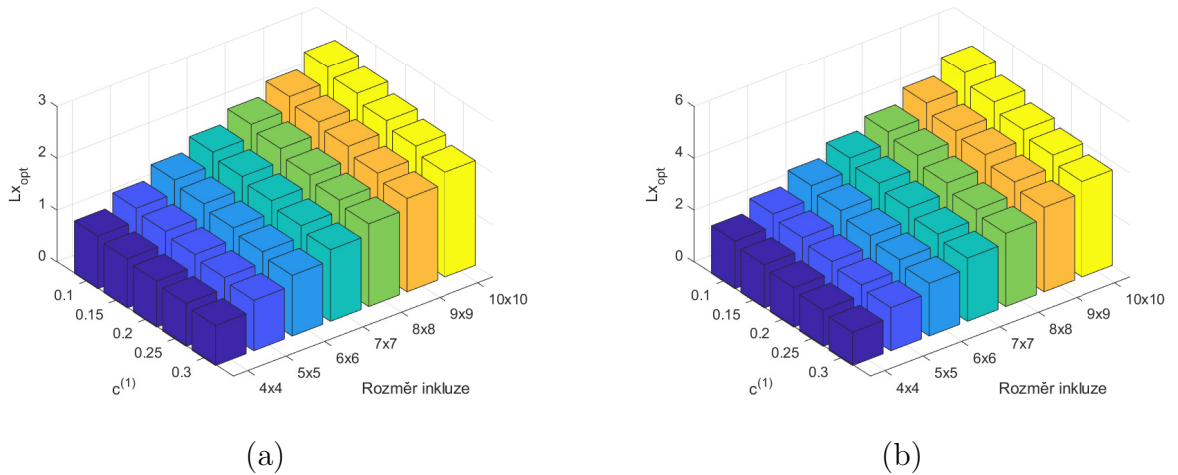
Pro potřeby samotné studie pak byl ještě potřeba nastavit rozptyl hodnot  $\sigma^2$  pro Gaussovské a exponenciální jádro a  $(\kappa^{(1)} - \kappa^{(2)})^2$  pro jádro počítané z obrázku dle rovnice (2.9). Za normálních okolností by tyto rozptyly byly vypočteny na základě sledovaných fyzikálních parametrů, avšak pro potřeby této práce sledující pouze vliv korelačních délek bez vazby na konkrétní fyzikální aplikaci, byly hodnoty nastaveny následovně:



Obrázek 2.5: Průběh absolutní chyby pro Gaussové (a) a exponenciální (b) jádro pro inkluze 10x10 - viz obr. 2.1

$$\begin{aligned}
 (\kappa^{(1)} - \kappa^{(2)})^2 &= 1, & \text{kde } \kappa^{(1)} &= 1 \text{ a } \kappa^{(2)} = 0. \\
 \sigma^2 &= (c^{(1)} - c^{(1)^2}), & \text{kde } c^{(1)} &\text{ je objemové zastoupení fáze 1.}
 \end{aligned}$$

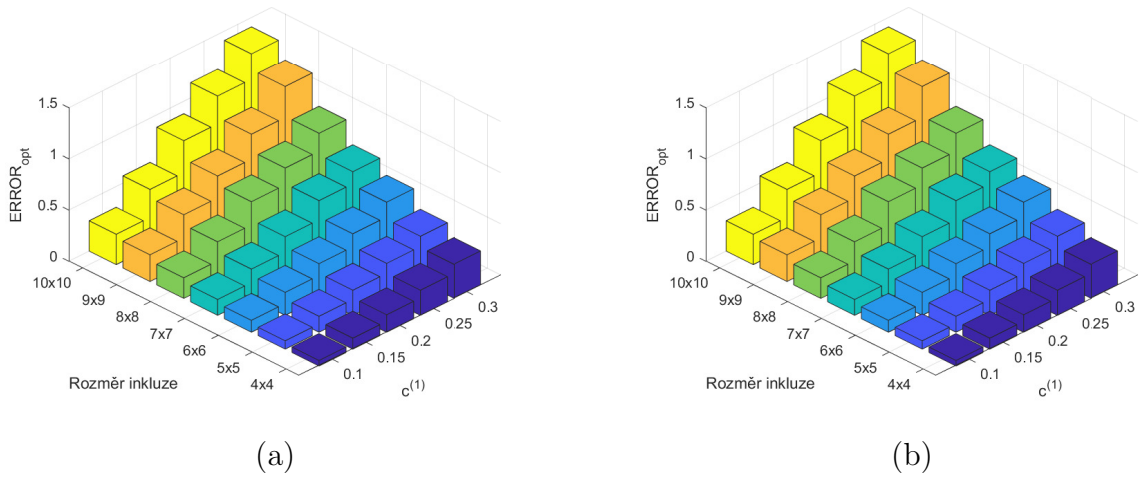
Takto se funkce shodují v počátku a mohou tak být optimalizovány. Obě studie, pro čtvercový i obdélníkový tvar inkluze, byly provedeny na 100 náhodně generovaných vzorcích, z nichž pak byl následně spočten aritmetický průměr. Rozměry inkluzí se pohybují v rozsahu 4x4 px až 10x10 px a 3x4 až 3x10 a objemové zastoupení odstupňované po 0,05 v intervalu  $< 0, 1; 0, 3 >$ <sup>3</sup>.



Obrázek 2.6: Růst průměrné optimální korelační délky  $L_x$  v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussové (a) a exponenciální (b) jádro

Z výsledků prezentovaných v na obrázku 2.6 je zřejmé, že korelační délka závisí na obou sledovaných parametrech. To, že s rostoucí velikostí inkluze roste i korelační délka,

<sup>3</sup>POZN: V následujících grafech jsou uváděny pouze orientační objemová zastoupení inkluzí. Toto zastoupení je vždy buď rovno zobrazenému, je-li možno umístit přesný počet inkluzí, nebo nejvyšší vyšší vzhledem k náhodnému generování obrazu (viz A). Odchylka je však vždy  $< 4 \cdot 10^{-3}$ .

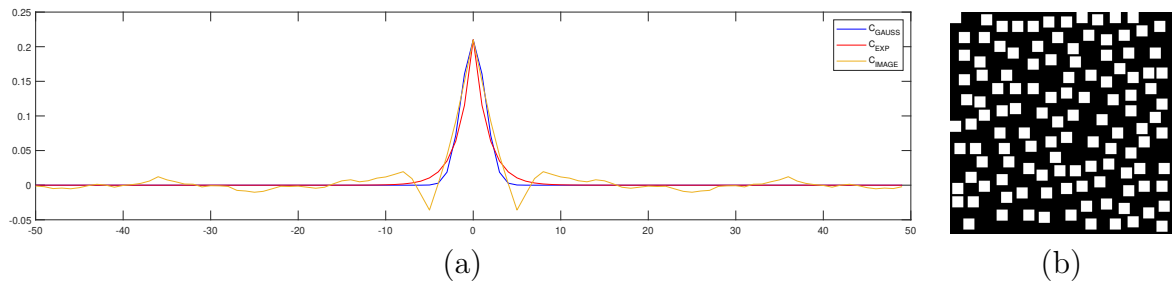


Obrázek 2.7: Růst průměrné minimální chyby (sumy čtverců chyb) v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussovské (a) a exponenciální (b) jádro

překvapivé není, neboť větší inkluze umožní i delším než jednotkovým vektorům padnout svým počátkem i koncem do stejné inkluze. Vzroste pro ně tedy hodnota dvoubodové pravděpodobnosti a hrot se tak rozšíří. U čtvercové inkluze se délky ve směru  $x$  a  $y$  významně neliší, délku ale velmi ovlivňuje právě rozměr inkluze, v případě obdélníkových inkluzí je samozřejmě větší v delším směru viz obrázek 2.9.

Druhý projevený aspekt je, že při vyšším objemovém zastoupení stejné inkluze korelační délka klesá. Vysvětlení zde není tak jednoduché, jako v předchozím případě. U kovariančního jádra vypočteného z dvoubodové pravděpodobnosti vzniká za hlavním vrcholem "podkmit" do záporných hodnot. Jelikož funkce z literatury jsou vždy nezáporné, toto místo nemohou nikdy dobře popsat.

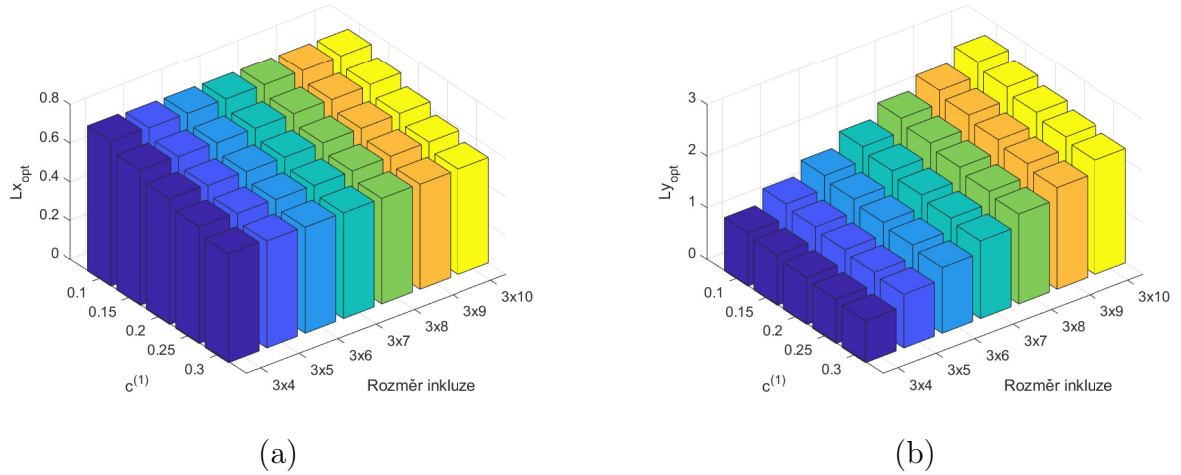
Při rostoucím objemovém zastoupení dochází k zahuštění domény a tak tedy vektory, které mají délku jen o trochu větší než aby se vešly do inkluze jsou více znevýhodněny oproti ostatním, "podkmit" za hrotem je tedy hlubší. V důsledku tohoto vrcholu v záporných hodnotách při minimalizaci čtverce chyby dojde ke snížení korelační délky, aby se umělá funkce nenacházela zbytečně vysoko. Graficky je tento fenomén zřejmý při porovnání obrázků 2.4 a 2.8, kde jsou vyobrazena kovarianční jádra pro vysoké objemové zastoupení.



Obrázek 2.8: (a) Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze  $c^{(1)} = 0,3$  inkluze  $5 \times 5$ , (b) příslušná doména

Zajímavější je však pohled na to, jak různá velikost inkluze či změna objemového zastoupení ovlivňuje samotnou minimální chybu při aproximaci pomocí kovariančních funkcí z literatury. Jak je patrné z prezentovaných výsledků na obrázku 2.7, chyba významně

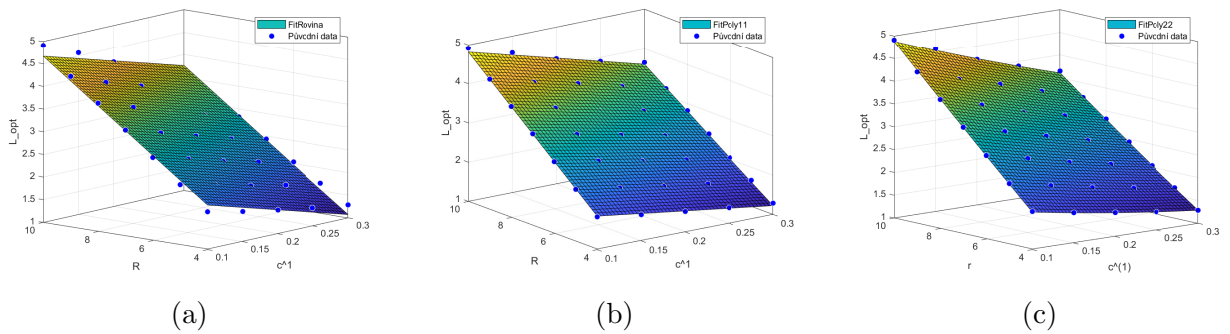




Obrázek 2.9: Optimální korelační délka pro Gaussovské jádro s obdélníkovou inkluzí v závislosti na její velikosti a objemovém zastoupení - (a)  $L_{x,opt}$  (b)  $L_{y,opt}$

roste s velikostí inkluze i objemovým zastoupením. Vezmeme-li v úvahu dosud nabyté poznatky, je zřejmé, že i tento fakt plyne ze stejných příčin. Větší velikost inkluze způsobí větší fluktuace v oblasti, kde jsou umělá kovarianční jádra blízka nule a nemohou je tedy postihnout (viz obrázek 2.4 a 2.1), zatímco rostoucí objemové zastoupení navyšuje amplitudu především blízko za vrcholem, viz obrázek 2.8.

Poslední částí této numerické studie je aproximace funkcí a pokus o formulaci jednoduchého vztahu pro odhad korelačních délek v závislosti na velikosti inkluze a objemovém zastoupení. Jako nejjednodušší se jeví aproximace rovinou plochou, zde je ale potřeba zmínit, že zatímco poměr mezi korelační délkou a objemovým zastoupením je přibližně lineární, podíl korelační délky a velikosti inkluze nikoliv. Že rovina není vhodnou aproximační plochou je vidět z obr. 2.10.



Obrázek 2.10: Aproximace hodnot  $L_{opt}$  pro exponenciální jádro (u Gaussovského jádra je trend podobný). Aproximace (a) rovinou (2.22), (b) hyperbolickým paraboloidem (2.23), (c) hyperbolickým paraboloidem (2.24).

Dále je vhodné poznamenat, že pro lineární regresi jsou data pro  $L_{x,opt}$  a  $L_{y,opt}$  u čtvercových inkluzí zprůměrována, neboť odchylka mezi těmito hodnotami je malá a především způsobena vlivem náhody, tvorba více aproximačních ploch by proto nedávala smysl.

V rámci studie jsou v tabulkách 2.1 a 2.2 prezentovány výsledky (včetně chyby) tří

aproximačních polynomiálních funkcí:

$$L_{opt} \approx A + B.r + C.c^{(1)}, \quad (2.22)$$

$$L_{opt} \approx B.r + C.c^{(1)} + D.r.c^{(1)}, \quad (2.23)$$

$$L_{opt} \approx B.r + C.c^{(1)} + D.r.c^{(1)} + E.r^2 + F.c^{(1)2}, \quad (2.24)$$

• kde:

$c^{(1)}$  je objemové zastoupení inkluze,  
 $r$  značí rozměr inkluze v daném směru a  
 $A$  až  $F$  jsou hledané konstantní koeficienty.

V tab. 2.2 a tab. 2.1 jsou uvedeny dvě měřítka chyby. První, běžně používaná, míra chyby *RMSE* (*z anglického root mean square error*) a také průměrná absolutní chyba pro lepší představu.

$$RMSE = \sqrt{\frac{\sum_{i=1}^d (x_i - x_{i,aprox})^2}{d}}, \quad (2.25)$$

$$MAE = \frac{\sum_{i=1}^d |x_i - x_{i,aprox}|}{d}, \quad (2.26)$$

• kde:

$d$  je množství dat,  
 $x_i$  je hodnota v  $i$ -tém bodě,  
 $x_{i,aprox}$  je aproximovaná hodnota v  $i$ -tém bodě.

Odchyłka u aproximace rovinou je poměrně značná, navíc pokud bychom zakázali konstantu  $A$ , ještě významně naroste. Naopak zavedení tohoto členu do rovnic (2.23) či (2.24) se výrazně na snížení chyby neprojeví a aproximaci zbytečně komplikuje. Že formule (2.22) dobře netrefuje trend je vidět i z obrázku 2.10. Povšimněme si, že největší pokles chyby způsobí u Gaussovských i exponenciálních délek uvážení součinu objemového zastoupení a rozměru inkluze (cca 3-4 násobný), tedy přechod od roviny k hyperbolickému paraboloidu.

Zahrnutí dalších kvadratických členů sice chybu dále snižuje, pokles ale již není tak významný, především, vezmeme-li v úvahu, že nepotřebujeme při výpočtu trefit přesné optimum (což se u náhodné struktury ani nemůže podařit), ale postačí rozumné okolí, v němž chyba neroste příliš rychle (viz obr. 2.5). Pro Gaussovské jádro tedy bohatě stačí vztah (2.23). Zahrnutí polynomu vyšších stupňů by stálo za úvahu u exponenciálního jádra, kde pokles chyby mezi rovnicí (2.23) a (2.24) je zhruba o  $\frac{1}{3}$ ). Opět zde ale narážíme na zkomplikování rovnice, která by měla usnadnit odhad, a především pak malé množství dat pro aproximaci vyššími stupni polynomů.

S uvážením výše zmíněných výsledků tak lze finální odhad korelačních délek formulovat:



Tabulka 2.1: Koeficienty aproximace pro Gaussovskou  $L_{opt}$ .  $L$  - čtvercová inkluze.  $L_x$  - obdélníková inkluze ve směru konstantní velikosti,  $L_y$  - obdélníková inkluze ve směru proměnné velikosti

Rce		A	B	C	D	E ( $\cdot 10^{-4}$ )	F	RMSE	MAE
(2.22)	L	0.3596	0.2326	-2.024	0	0	0	0.0407	0.0290
	$L_x$	0.3512	0.2333	-1.998	0	0	0	0.0386	0.0280
	$L_y$	0.3680	0.2319	-2.050	0	0	0	0.0433	0.0311
(2.23)	L	0	0.2842	-0.2169	-0.259	0	0	0.0086	0.0060
	$L_x$	0	0.2835	-0.2403	-0.2513	0	0	0.0093	0.0069
	$L_y$	0	0.2848	-0.1934	-0.2667	0	0	0.0106	0.0073
(2.24)	L	0	0.2792	-0.1216	-0.2829	7.389	0.2146	0.0074	0.0051
	$L_x$	0	0.2778	-0.0912	-0.2672	6.589	-0.0709	0.0089	0.0066
	$L_y$	0	0.2805	-0.1519	-0.2986	8.189	0.5001	0.0087	0.0059

Tabulka 2.2: Koeficienty aproximace pro exponenciální  $L_{opt}$ .  $L$  - čtvercová inkluze.  $L_x$  - obdélníková inkluze ve směru konstantní velikosti,  $L_y$  - obdélníková inkluze ve směru proměnné velikosti

Rce		A	B	C	D	E ( $\cdot 10^{-4}$ )	F	RMSE	MAE
(2.22)	L	0.6474	0.4475	-4.548	0	0	0	0.0924	0.0668
	$L_x$	0.6046	0.4520	-4.444	0	0	0	0.0873	0.0618
	$L_y$	0.6901	0.4430	-4.652	0	0	0	0.1037	0.0770
(2.23)	L	0	0.542	-1.209	-0.486	0	0	0.0335	0.0241
	$L_x$	0	0.540	-1.337	-0.4512	0	0	0.0380	0.0268
	$L_y$	0	0.544	-1.080	-0.5208	0	0	0.0443	0.0342
(2.24)	L	0	0.530	-1.576	-0.6176	23.30	3.411	0.0198	0.0135
	$L_x$	0	0.5294	-1.374	-0.5687	26.48	2.317	0.0306	0.0211
	$L_y$	0	0.5486	-1.779	-0.6665	20.12	4.505	0.0319	0.0206

- pro Gaussovské jádro:

$$L_{opt,G} \approx 0,3.r - 0,2.c^{(1)} - 0,25.c^{(1)}.r, \quad (2.27)$$

- pro exponenciální jádro:

$$L_{opt,e} \approx 0,55.r - 1,2.c^{(1)} - 0,5.c^{(1)}.r, \quad (2.28)$$

- kde:

$c^{(1)}$  je objemové zastoupení inkluze,  
 $r$  značí rozměr inkluze v daném směru.

Nutno podotknout, že odhady (2.27) a (2.28) mají velmi omezenou platnost. Dalo by se říci, že platí pouze pro druh (tedy inkluze o stejné velikosti, náhodně rozložené s minimální mezerou 1) vzorků a rozsah hodnot, na nichž byly testovány.

## 2.4 Karhunenovo–Loèveho rozvoj

Po té, co jsme získali informaci o struktuře pomocí kovariančních funkcí, přichází chvíle pro generování samotného náhodného pole. V této práci budeme náhodné pole generovat pomocí Karhunenovo–Loèveho rozvoje, zkráceně KLR (*KLE* - *Karhunen–Loève Expansion*), v literatuře se však setkat i s odlišnými přístupy, jako například vylepšenými per-turbačními metodami či Hashin-Shtrikmanovým variačním přístupem [3]. Těmito se zde však nebudeme zabývat.

KLR je vlastně nekonečný součet sloužící pro popis náhodného pole. V rovnici (2.29) je zapsán v čisté podobě pro spojitou doménu ve tvaru nekonečného součtu, rovnice (2.30) pak je pouhou úpravou pro výpočet na diskretizované doméně [3], [5], [6]:

$$\kappa(X, \omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} \phi_i(X) \xi_i(\omega), \quad (2.29)$$

$$\kappa(X, \omega) \approx \mu_{\kappa}(X) + \sum_{i=1}^n \sqrt{\lambda_i} \phi_i(X) \xi_i(\omega), \quad (2.30)$$

- kde:

$\kappa(X, \omega)$  značí veličinu jako funkci souřadnice  $X$  a náhodné proměnné  $\omega$ ,  
 $\mu_{\kappa}(X)$  je střední hodnota veličiny  $\kappa$ ,  
 $\lambda_i$  je  $i$ -té vlastní číslo kovarianční funkce,  
 $\phi(X)_i$  zastupuje  $i$ -tou vlastní funkci kovarianční funkce,  
 $\xi_i(\omega)$  je náhodná proměnná (viz podkapitola 2.4.2) a  
 $n$  je počet členů uvažovaných v konečném rozvoji pro diskretizovaný výpočet.

Neboť pracujeme na diskrétní doméně, tak v našem případě přejde vlastní funkce  $\phi_i(x)$  na vlastní vektor kovarianční matice, který se dá snadno určit. Vlastní číslo nám pak hraje roli váhy, kterou vstupuje příslušná vlastní funkce do rozvoje.

### 2.4.1 Vlastní čísla a vlastní tvary kovariančních jader

Před vlastní dekompozicí do vlastních tvarů  $\phi_i$  a čísel  $\lambda_i$ , je potřeba přeuspořádat kovarianční jádra. Dosud jsme vždy vztahovali kovariance ke konkrétnímu bodu, avšak ze vztahů (2.6) a (2.7), kovarianční funkce lze zapsat jako funkce vzdáleností dvou bodů. Původní doména je  $m.n$ , kovarianční matice tak bude čtvercová rozměru  $(m.n)^2$ , z definice symetrická, pozitivně semidefinitní a vyjadřuje právě kovarianci mezi jednotlivými body.

Prvním způsobem, jak dostat tuto matici, je napočítat kovarianční jádra pro všechny body, rozvinout kovarianční jádro do vektoru a ten vložit do příslušného řádku matice. Toto lze provést pro rovnice (2.6) a (2.7), avšak kovarianční jádro zjištěné z dvoubodové pravděpodobnosti (rovnice (2.9)) nelze znovu počítat k jinému bodu, ale opět se jedná o funkci vzdálenosti, proto je možné matici posouvat. Následným řešením charakteristické rovnice dostaneme vlastní čísla a vypočteme vlastní vektory.

Tuto metodu sestavení kovarianční matice použijeme vždy pro Gaussovské a exponenciální jádro, neboť se jedná o umělé funkce vesměs nezávislé na konkrétní realizaci (opomineme-li optimalizaci korelační délky), nemáme tedy jinou variantu postupu. V případě korelačního jádra zjištěného z dvoubodové pravděpodobnosti se nám však nabízí další možnost. Máme-li k dispozici dostatečné množství vzorků, lze vlastní tvary získat pomocí *analýzy hlavních komponent* (dále též PCA - z ang. *Principal Component Analysis*). Nejedná se pak tedy o vlastní tvary konkrétní realizace, ale statistického souboru, který byl generován shodným algoritmem.

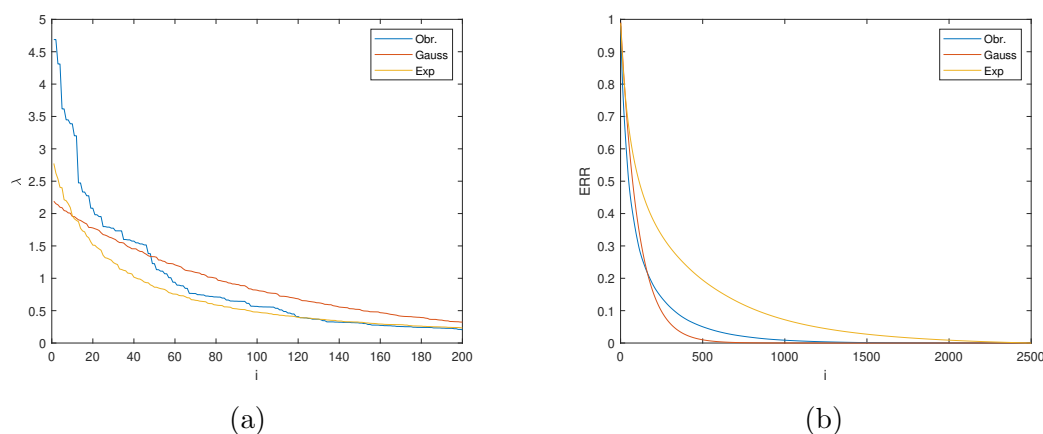
První vlastní tvary popisují zkoumanou doménu nejvíce, poslední pak nejméně. Váhu, jaký význam mají dílčí vlastní tvary  $\phi_i$ , určují vlastní čísla  $\lambda_i$ . Neboť množství i délka vlastních vektorů kvadraticky roste s velikostí domény, nebudeme chtít při zpětném generování náhodného pole zahrnovat všechny vektory. Pro odhad chyby, které se dopustíme zanedbáním daného množství vlastních tvarů, použijeme právě postup z PCA, zapsaný následující formulí:

$$\varepsilon = 1 - \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^n \lambda_i}, \quad (2.31)$$

• kde:

- $\varepsilon$  je odhadovaná chyba,
- $\lambda_i$  je  $i$ -té vlastní číslo příslušící  $\phi_i$ ,
- $k$  značí počet zahrnutých vlastních tvarů  $\phi_i$ ,
- $n$  jest počet všech vlastních tvarů.

Jak lze pozorovat z obr. 2.12b, nejpomaleji klesá chyba exponenciálního jádra, rychlost konvergence je srovnatelná pro Gaussovské jádro a jádro zjištěné z dvoubodové pravděpodobnosti. Rychlost konvergence chyby lze vysvětlit právě konvergencí vlastních čísel patrnou z obr. 2.12a. Exponenciální jádro má v porovnání z jádrem z obrázku malá vlastní čísla, která navíc rychle klesají. Strmý sestup sice vykazují i vlastní čísla jádra z obrázku, ale tato čísla jsou na počátku mnohem větší, první tvary tedy popíší většinu domény. Gaussovská vlastní čísla jsou sice nízká, na svém počátku však nejpomaleji klesají a proto chyba konverguje nejrychleji k 0. Jak však již bylo diskutováno dříve, u Gaussovského (i exponenciálního) jádra je zanesena chyba v samotném popisu



Obrázek 2.12: Konvergence vlastních čísel (a) a odhadnuté chyby (b) na doméně 50x50 pro velikost inkluze 5x5

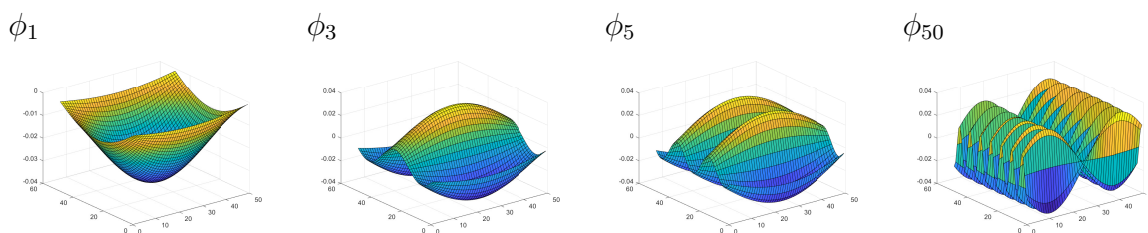
konkrétní domény.

Avšak rychlost konvergence je velmi odvislá od toho, jak velké jsou inkluze (vůči doméně). Tato závislost je bohužel nepříznivá pro kompresi dat, neboť čím větší jsou inkluze, tím rychleji konverguje chyba k 0 viz obr. 2.16. Vlastní čísla a konvergence chyby se také liší v případě, že je určujeme pomocí PCA.

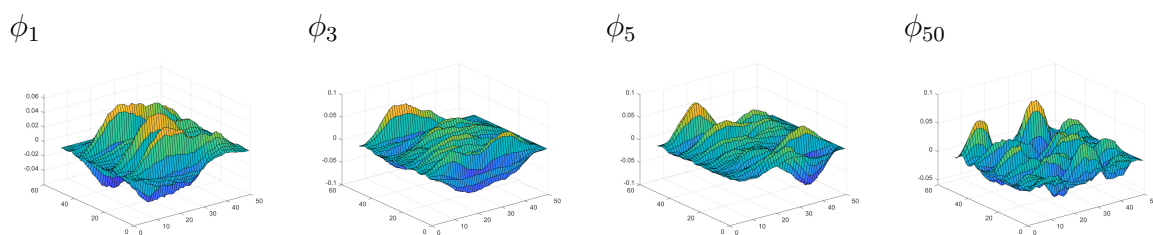
Další komentář si zaslouží podoba samotných vlastních tvarů  $\phi$ . První tvary v sobě nesou nejvíce informace, jsou zároveň nejméně rozkmitané. Frekvence fluktuací se zvyšuje s číslem vlastního tvaru.

Několik vlastních tvarů pro exponenciální kovarianční jádro je vyobrazeno na obr. 2.13. Tyto funkce jsou však známy z literatury a především jsou nezávislé na studované doméně. (Vliv má pouze korelační délka  $L$ , s rostoucí  $L$  se zvyšuje i perioda fluktuací, první tvar by například mohl tedy obsahovat pouze část Gaussovského klobouku. Toto by samozřejmě vedlo k dalšímu zkreslení výsledků.)

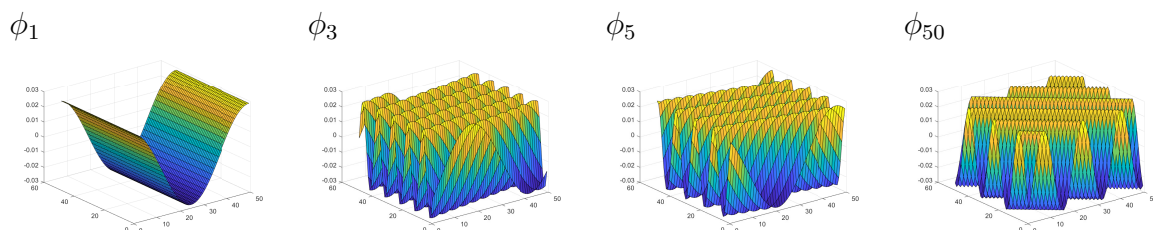
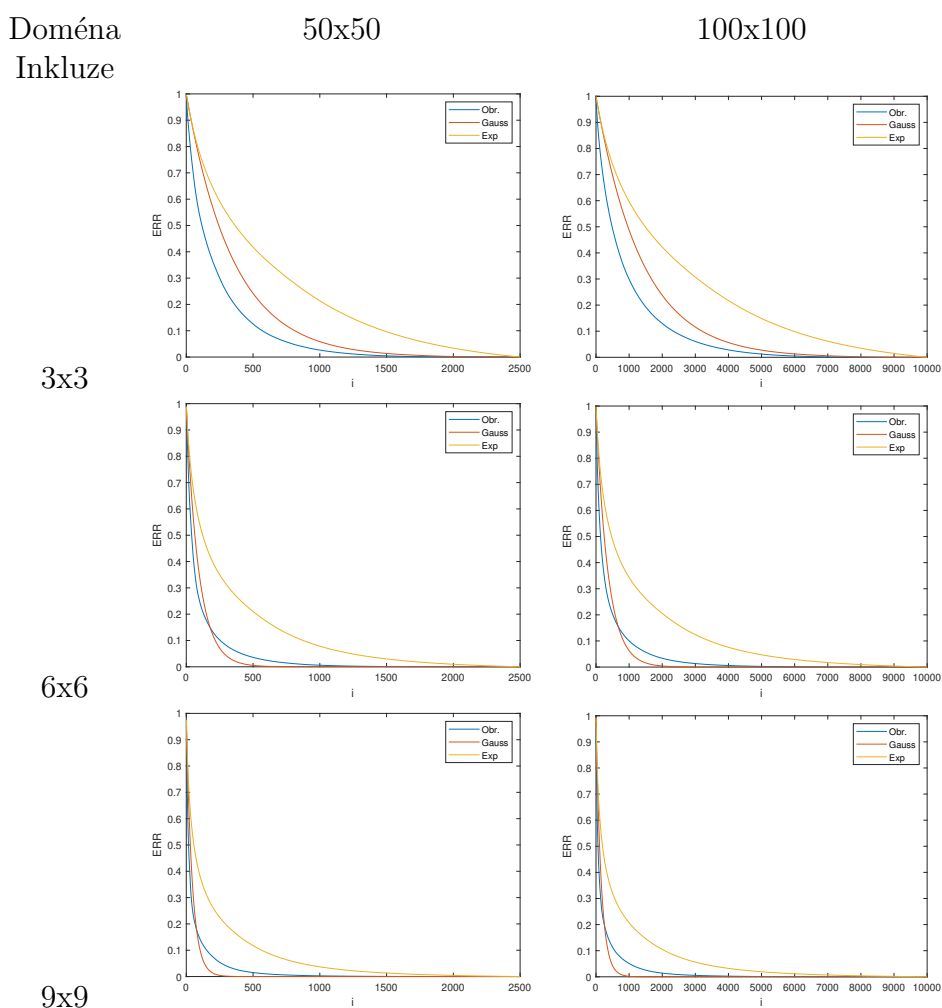
Naopak vlastní tvary odvozené z PCA (viz obr. 2.14) již závisí na studované doméně, stejně jako vlastní tvary odvozené z dvoubodové pravděpodobnosti (viz obr. 2.15).



Obrázek 2.13: Vlastní tvary exponenciálního Jádra, doména 50x50 px, inkluze 3x12 px



Obrázek 2.14: Vlastní tvary z PCA, doména 50x50 px, inkluze 3x12 px

Obrázek 2.15: Vlastní tvary zjištěné pomocí  $S_2$ , doména 50x50 px, inkluze 3x12 pxObrázek 2.16: Porovnání konvergence odhadu chyby v závislosti na velikosti inkluze pro  $\lambda_i$  zjištěné první popsanou metodou

### 2.4.2 Volba náhodných proměnných $\xi(\omega)_i$ v KLR

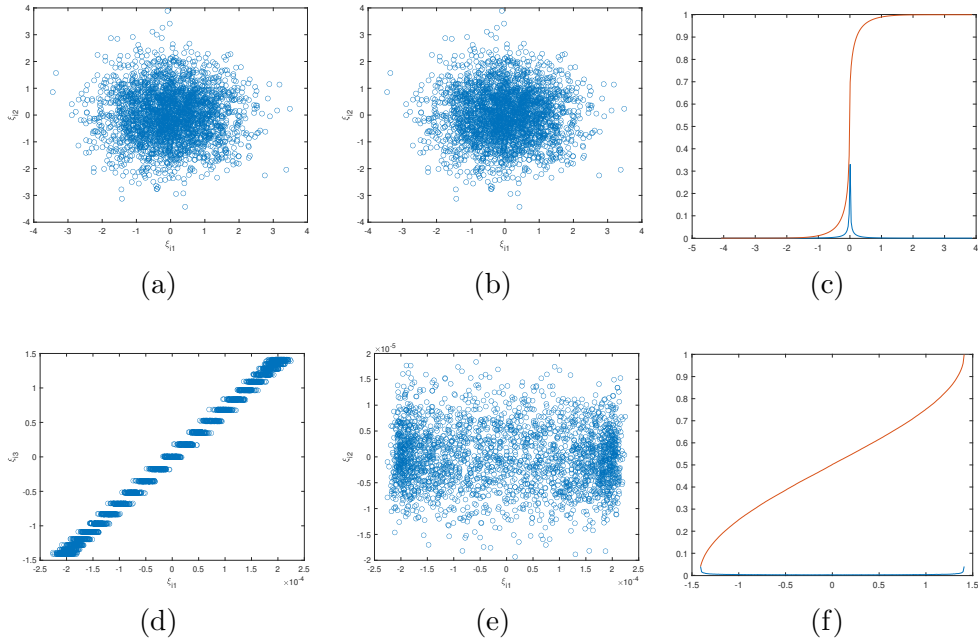
Nejproblematictější částí KLR je právě vypořádání se s náhodnými proměnnými  $\xi$ . Jejich pravděpodobnostní rozdělení není známé, proto je běžně volen přístup, kdy jsou náhodné proměnné vybírány z normálního spojitého rozdělení. Toto ale předpokládá, že  $\xi$  jsou nezávislá. Navíc v našem případě, kdy se snažíme charakterizovat binární náhodné pole toto povede na generování reálného náhodného pole<sup>4</sup>.

Pokusíme se nyní několika způsoby získat odpovídající  $\xi$  z referenčního řešení, které máme k dispozici. Uvažujme dva případy, kdy máme k dispozici pouze jediný obrázek, který považujeme za reprezentativní a druhý, kdy máme větší množství vzorků studovaného média. V prvním případě již máme k dispozici vypočtená  $\phi$  a  $\lambda$ .  $\xi$  tedy dostaneme z rovnice (2.30), zde v maticovém zápisu jako:

$$\xi(\omega) = \phi_\lambda^{-1}(\kappa(X, \omega) - \mu_\kappa(X)), \quad (2.32)$$

- kde:

$\phi_\lambda$	značí matici vlastních vektorů přeškálovaných $\lambda_i$ ,
$\kappa(X, \omega)$	matice realizací obrázku,
$\mu_\kappa(X)$	je matice středních hodnot veličiny $\kappa$ .



Obrázek 2.17: Vykreslení  $\xi$  vůči sobě a distribuční funkce (a,b,c) pro PCA, (d,e,f) pro posun. obr.

V našem případě jsou prvky  $\mu_\kappa(X)$  rovny objemovému zastoupení inkluze  $c^{(1)}$ . V matici  $\kappa$  sloupce odpovídají realizacím<sup>5</sup>. Jelikož máme k dispozici jediný vzorek média,

<sup>4</sup>Toto je vyřešeno pomocí prahu, který rozdělí reálné hodnoty na binární (0,1) tak, aby odpovídalo objemové zastoupení inkluze.

<sup>5</sup>Dvourozměrné binární náhodné pole rozvinuté do vektoru. Každý prvek matice tedy koresponduje s konkrétním pixelem v obrázku. Máme-li tedy např. doménu 20x20 px, matice  $\kappa$  bude mít 400 řádek.

budeme prozatím uvažovat jeho periodicitu a další vzorky získáme posouváním obrázku rozvinutého do vektoru vždy o jeden px<sup>6</sup>. V druhém případě, kdy máme k dispozici velké množství vzorků, dostaneme náhodné proměnné  $\xi$  přímo z PCA jako souřadnice realizací  $\kappa$  v rotovaném souřadném systému. Pro další postup si stanovíme distribuční funkce pro obě varianty. Ty jsou společně s několika  $\xi$  vykresleny na obr. 2.17). Pro variantu PCA se zdají  $\xi$  být nezávislá, na rozdíl od varianty posouvaného obr.

### 2.4.3 Hodnocení výsledků náhodného generování

Nyní je potřeba posoudit kvalitu vygenerovaného náhodného pole. Určení vhodného měřítka chyby však není zcela přímočaré.

Stanovme si tedy nyní dva požadavky, *kvalitativní* a *kvantitativní*. Pod kvalitativním požadavkem rozumějme, že pokud jsme vycházeli z anizotropního (resp. izotropního) média, budeme požadovat, aby i vygenerované náhodné pole bylo anizotropní (resp. izotropní)<sup>7</sup>. U kvantitativního měření chyby budeme rozlišovat dva případy. A sice situaci, že máme k dispozici pouze jeden vzorek struktury, který považujeme za reprezentativní. Pak podobnost budeme měřit pomocí rozdílu kovariančních matic následovně:

$$\varepsilon_1 = \frac{1}{q} \sum_{k=1}^q \sum_{i=1}^n \sum_{j=1}^m |C_{I,new}(i, j) - C_{I,org}(i, j)|, \quad (2.33)$$

• kde:

- $m$  a  $n$  jsou rozměry domény,
- $q$  je počet nově vygenerovaných náhodných realizací,
- $C_{I,org}$  je původní kovarianční matice (vzorku) vypočtená dle (2.9),
- $C_{I,new}$  je nová kovarianční matice (vzorku) vypočtené dle (2.9).

Druhou variantu uvažujeme tak, že máme k dispozici informace o "nekonečném médiu" zde reprezentovaném velkým množstvím vzorků, respektive že máme k dispozici reprezentativní soubor, s nímž můžeme porovnávat generované vzorky.

$$\varepsilon_q = \sum_{i=1}^n \sum_{j=1}^m |K_{new}(i, j) - K_{org}(i, j)|, \quad (2.34)$$

• kde:

- $m$  a  $n$  jsou rozměry domény,
- $K_{org/new}$  je kovarianční matice původního, resp. nového souboru, jejíž  $i$ - $j$ -tý prvek je určen jako:

$$K_n(i, j) = \langle (\mathbf{x}_i - \mu_{\mathbf{x}_i})(\mathbf{x}_j - \mu_{\mathbf{x}_j}) \rangle, \quad (2.35)$$

kde  $\langle \dots \rangle$  značí aritmetický průměr,  $\mathbf{x}_i$  je  $i$ -tý vektor pozorování a  $\mu_{\mathbf{x}_i}$  je střední hodnota vektoru  $\mathbf{x}_i$ .

<sup>6</sup>Takto získaná  $\xi$  rozhodně nebudou nezávislá.

<sup>7</sup>I tento požadavek by bylo možné kvantifikovat, prozatím jej však budeme vyhodnocovat "opticky".



Hodnota této chyby samozřejmě bude záviset na velikosti nově vygenerovaného souboru, avšak pro porovnání postupů generování náhodného pole absolutní hodnotu nemusíme řešit. Porovnávány varianty jsou přehledně shrnuty v tabulce tab. 2.3.

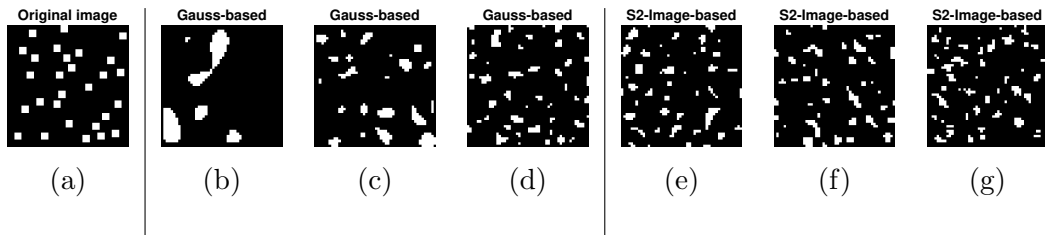
Tabulka 2.3: Přehled variant zpětného generování náhodného pole

Označení	Vstupní data	Kovarianční jádro	Rozdělení $\xi$
Gauss	$L_{opt}$ z 1 obr.	Gauss	Gauss
Exp	$L_{opt}$ z 1 obr.	Exp	Gauss
S2-Image	$S_2$ z 1 obr.	$S_2$	Gauss
SHIFT-H	$S_2$ z 1 obr. + hist $\xi$	$S_2$	hist $\xi$
SHIFT-U	$S_2$ z 1 obr. + hist $\xi$	$S_2$	rovnoměrná $\xi$
PCA-G	$K_{org}$ z víc obr.	neperiodická $K_{org}$	Gauss
PCA-H	$K_{org}$ z víc obr. + hist $\xi$	neperiodická $K_{org}$	hist $\xi$

Konvergenci průměrné chyby v závislosti na počtu zahrnutých vlastních tvarů prezentují obrázky 2.20 a 2.21. Výsledky na nich jsou v souladu s (2.21) a (2.20) pro množství generovaných vzorků  $q = 100$ . Množství vzorků referenčního řešení souboru je 25 000, z toho 2 750 bylo zpřístupněno pro výpočty pomocí PCA

Prvně je dobré si povšimnout, že tato konvergence se liší od odhadu přesnosti pomocí vlastních čísel z rovnice (2.31). To může být částečně způsobeno nastavením prahu, který nám generovanou náhodnou reálnou proměnnou převádí na binární. Konvergence chyby je ilustrována i na generovaných vzorcích viz obr. 2.18. Neb chyba konverguje poměrně rychle, není rozhodně potřeba do výpočtu zahrnovat velké množství  $\phi$ . Jak je nejlépe patrné z obr. 2.21b, v některých případech to ani nemusí být vhodné.

Z grafů je dále vidět, že vypočtená  $\xi$  za předpokladu periodického obrázku (SHIFT-H) lze v našem případě dobře aproximovat rovnoměrným rozdělením. Veškeré varianty zjištěné z obrázku si vedou lépe při porovnání z referenčním obrázkem, hůře pak proti souboru.

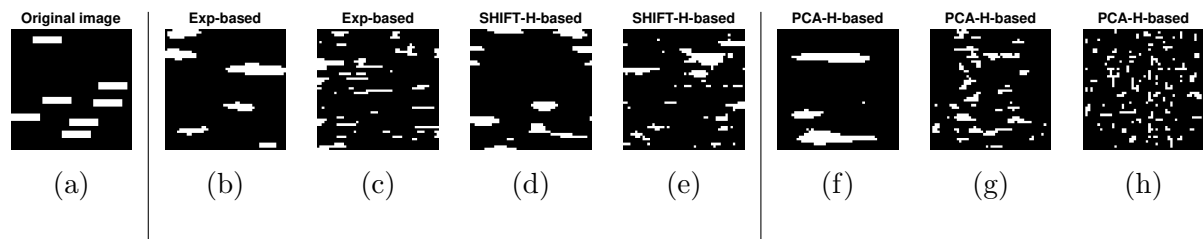


Obrázek 2.18: Porovnání generovaných obrázků 50x50 px, pro čtvercové inkluze, (a) originální obr. (b) Gaussovské jádro 50  $\phi$  (c) Gaussovské jádro 150  $\phi$ , (d) Gaussovské jádro 500  $\phi$ , (e) jádro z  $S_2$  50  $\phi$ , (f) jádro z  $S_2$  150  $\phi$ , (g) jádro z  $S_2$  500  $\phi$

Varianta PCA chyba za svým lokálním minimem roste i s vyšším počtem zahrnutých  $\phi$ , obdobný trend vykazují i ostatní křivky. Z obr. 2.21 a obr. 2.20 by se mohlo zdát, že PCA je nejlepší metodou poisu média, ale jak prezentuje obr. 2.19, zatímco ostatní varianty se při zahrnutí většího množství  $\phi$  blíží "kvalitnějšímu" řešení a zachovávají si správnou informaci o anizotropii materiálu, PCA nikoliv (pravděpodobně stejným způsobem selhává pro izotropní doménu, kde se tato chyba však neprojeví).

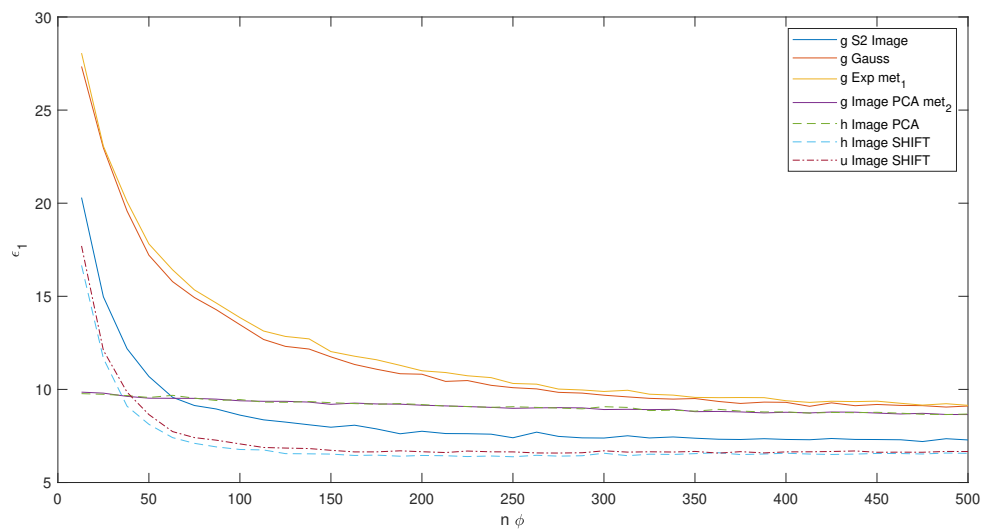
Tento jev může být způsoben tím, že  $\xi$  jsou ve skutečnosti závislá a toto rozdělení neumíme dobře popsat. Ani aproximace vypočteným histogramem v tomto případě řešení výrazně nezlepší.



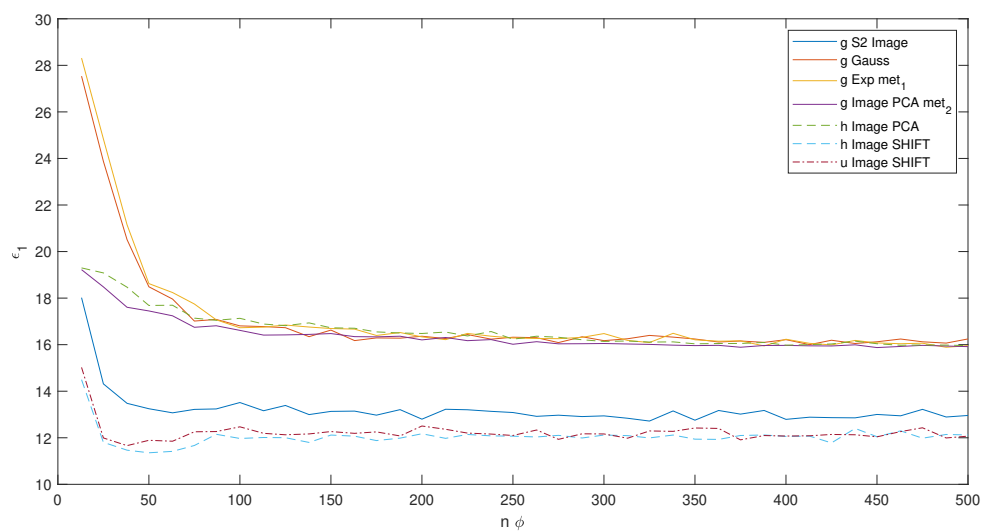


Obrázek 2.19: Porovnání generovaných obrázků 50x50 px, pro obdélníkové inkluze 3x12 px (a) originální obrázek (b) exp. jádro, 40  $\phi$ , (c) exp. jádro, 250  $\phi$ , (d) varianta SHIFT-H, 40  $\phi$ , (e) varianta SHIFT-H, 250  $\phi$ , (f) varianta PCA, 40  $\phi$ , (g) varianta PCA, 250  $\phi$ , (h) varianta PCA, 500  $\phi$

Z těchto poznatků plyne, že chyba změřená pouze pomocí rozdílu kovariancí může pomoci pro utvoření představy o kvalitě pole, ale sama o sobě je nedostatečný ukazatel. Nabízí se tedy hodnotit pole jinak, například pomocí odezvy jednoduchého a přísného fyzikálního (např. mechanického) modelu, což je želbohu mimo rozsah této práce.

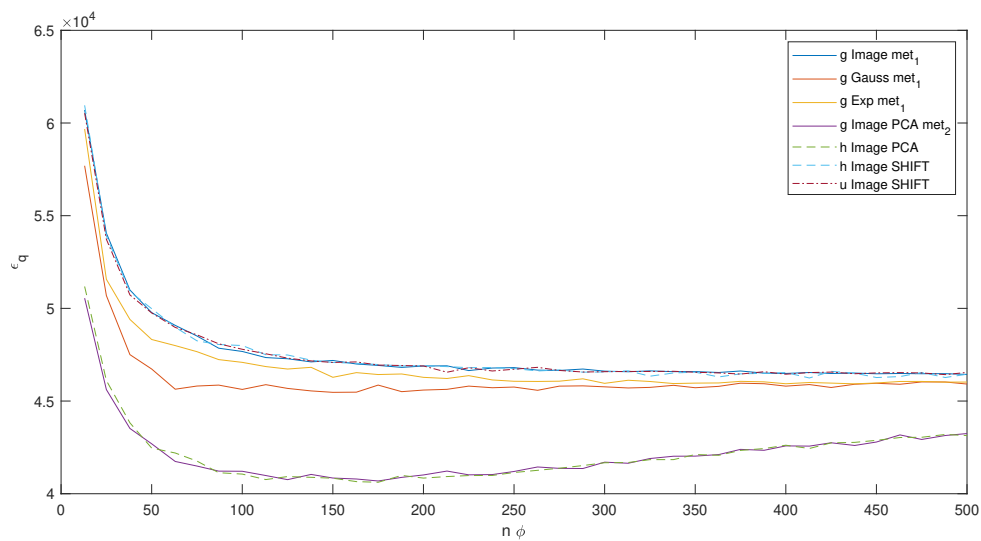


(a)

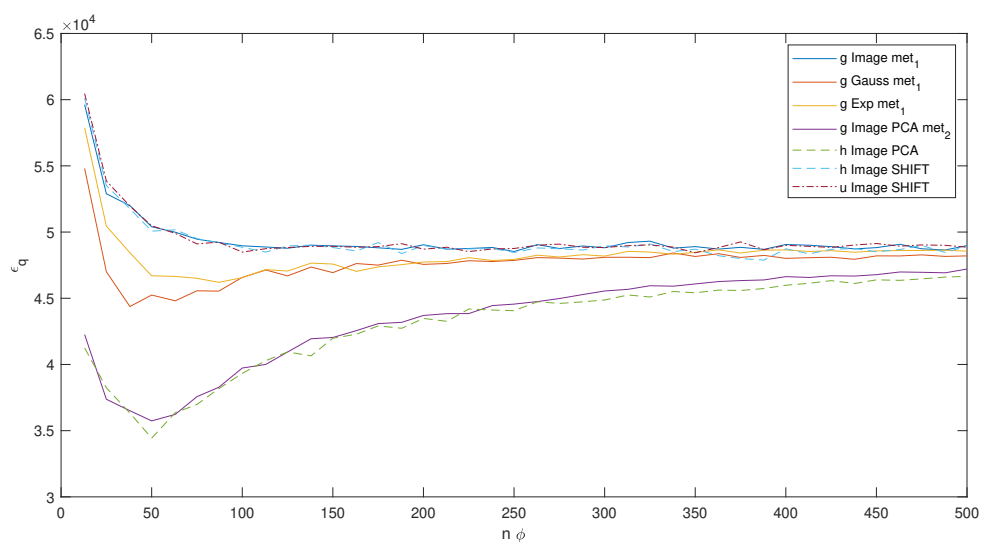


(b)

Obrázek 2.20: Konvergence chyby při porovnání vůči vzorku pro  $c^{(1)} = 0,1$  na doméně  $50 \times 50$  px, velikost inkluze (a)  $3 \times 3$  px (b)  $3 \times 12$  px



(a)



(b)

Obrázek 2.21: Konvergence chyby při porovnání vůči souboru pro  $c^{(1)} = 0, 1$  na doméně  $50 \times 50$  px, velikost inkluze (a)  $3 \times 3$  px (b)  $3 \times 12$  px

# Kapitola 3

## Závěr

V rámci této práce je předvedeno generování binárního náhodného pole pomocí obrazové analýzy s využitím dvoubodové pravděpodobnosti a analýzy hlavních komponent. Tyto postupy jsou srovnávány s funkcemi z literatury, pro jejichž volené (korelační délky) parametry je provedena numerická studie. Při špatně zvolené korelační délce chyba aproximace média velmi prudce roste, proto je potřeba ji buď velmi dobře odhadnout, nebo lépe při použití umělých kovariančních jader tento parametr optimalizovat.

V porovnání s běžnými postupy z literatury se obrazová analýza ukazuje jako mnohem vhodnější metoda pro kompresi dat. Analýza hlavních komponent je citlivá na zahrnutí velkého množství vlastních tvarů, kdy není schopna udržet informace o anizotropii materiálu.

Zdá se, že náhodné proměnné  $\xi$  v KLR, nejsou nezávislé a zanedbání tohoto aspektu může zhoršení výsledků.

Analýza hlavních komponent i po selhání při vyšším množství zahrnutých vlastních tvarů vykazuje chybu měřenou pomocí rozdílů kovariancí menší nebo srovnatelnou s ostatními metodami, které správnou informaci o anizotropii zachovají výrazně lépe.

Proto se zdá, že pro posouzení kvality náhodného pole odvozeného z obrazové analýzy tento postup nestačí a nabízí se kvalitu postupů ohodnotit odezvou fyzikálních modelů. Budiž toto předmětem další práce.

# Příloha A

## Použité kódy a komentář

V této příloze jsou uvedeny sepsané algoritmy<sup>1</sup> a je k nim uveden drobný komentář tak, aby jejich fungování pochopil i čtenář, který není v programování zblhlý.

Veškeré kódy by samozřejmě šly napsat efektivněji, především pro velké matice se vyplácí v prostředí MATLAB využívat vektorizace. Kódy jsou psány tak, aby bylo dosaženo kompromisu mezi přehledností (srozumitelností) algoritmu a výpočetní rychlostí pro běžné užití.

### A.1 Náhodné generování binárního obrázku

Tento skript byl napsán aby generoval náhodné struktury pro potřeby této práce. Zakomentovaná časová podmínka byla užita v rámci numerické studie, neboť u vyšších objemových zastoupení s menším rozměrem inkluze může snáze nastat situace, že algoritmus rozhodí předchozí inkluze tím způsobem, že není schopen umístit další. Tímto je zajištěné opětovné spuštění algoritmu v takovém případě.

```
1     function [OBR,C1]=RandomPictureGenerator(V_inkluze, DIST, x_ink, ...
        y_ink, nx, ny)
2
3     % ZAKLADNI PARAMETRY POZADOVANEHO OBRAZKU
4     % V_inkluze -- OBJEMOVE ZASTOUPENI INKLUZE (BILA)
5     % DIST -- MINIMALNI ROZESTUP INKLUZE (BILA) [px]
6     % x_ink; [px] y_ink; [px] -- ROZMER INKLUZE
7     % nx; %[px] %radky, ny; %[px] Sloupce -- ROZMER OBRAZKU
8     % VRACI -- MATICI NAHODNEHO OBRAZKU OBR A SKUTECNE OBJEMOVE ...
        ZASTOUPENI C1
9
10    %% VYPOCET
11    % ALOKACE PAMETI PRO MATICI (CERNA)
12    OBR=zeros(nx,ny);
13    % DEFINICE INKLUZE
14    INKLUZE=ones(x_ink, y_ink);
15
16    % uprava pro 1D
17    hx=0;
18    hy=0;
19    if x_ink==nx
```

---

<sup>1</sup>Uvedeny zde jsou pouze funkce a algoritmy sloužící pro výpočet práce. Drobné obměny daných funkcí (např. z důvodu urychlení pro numerickou studii) a funkce pro export výsledků zde uvedeny nejsou.

```
20     hx=1;
21 end
22 if y_ink==ny
23     hy=1;
24 end
25
26 % VEKTOR POCATECNI POLOHY
27 rx1=randi([1,nx-x_ink+hx]);
28 ry1=randi([1,ny-y_ink+hy]);
29 R=[rx1,ry1];
30
31 C1=0; SEMITEST=true; SEMITEST0=true;
32 V_fill=0;
33 V=nx*ny;
34
35 % T2=0; % CASOVA PODMINKA
36
37 while C1<V_inkluzi
38     % PRIRAZENI INKLUZE
39     OBR(R(1):R(1)+x_ink-1,R(2):R(2)+y_ink-1)=INKLUZE;
40
41     SEMITEST0=true;
42     while SEMITEST0
43
44         %         tic % CASOVA PODMINKA
45
46         % vyber novych souradnic
47         rx=randi([1,nx-x_ink+hx]);
48         ry=randi([1,ny-y_ink+hy]);
49
50         DMX=max(rx-DIST,1);
51         HMX=min(rx+x_ink+DIST,nx);
52         DMY=max(ry-DIST,1);
53         HMY=min(ry+y_ink+DIST,ny);
54
55         if sum(OBR(DMX:HMX,DMY:HMY),'all')==0
56             SEMITEST0=false;
57         end
58
59         %         % CASOVA PODMINKA
60         %         T1=toc;
61         %         T2=T2+T1;
62         %         if T2>0.5
63         %             OBR=zeros(nx,ny);
64         %             T2=0;
65         %             V_fill=0;
66         %         end
67
68     end
69
70     % AKTUALIZACE POZICE
71     R=[rx,ry];
72
73     % KONTROLA ZAPLNENI PROSTORU
74     V_fill=V_fill+x_ink*y_ink;
75     C1=V_fill/V;
76
77 end
```

## A.2 Dvoubodová pravděpodobnost

Tento kód vypočte dvoubodovou pravděpodobnost pro binární doménu pomocí definice. Nejprve zjistí velikost domény a pak pro každý prvek matice definuje vektor  $v = (i, j)$ , který nechá proběhnout všechny prvky matice a počítá počet úspěšných pokusů, kolikrát počátek i konec padl do bílé fáze odpovídající číslu 1. Neboť předpokládáme periodicitu, je zde navíc ověřovací podmínka, která případný konec vektoru mimo doménu přesune na odpovídající pozici v matici.

```

1      function [DP] = TwoPointProbability(OBR)
2      % TwoPointProbability FUNCTION
3      % Vypocet funkce z definice
4      % OBR ... matice obrazku
5
6      [nROW,nCOLUMN]=size(OBR);
7
8      for i=1:nROW
9          for j=1:nCOLUMN
10             nANO=0;
11             for ki=1:nROW
12                 for kj=1:nCOLUMN
13
14                     if OBR(ki,kj)==1
15                         indexR=ki+i-1;
16                         indexC=kj+j-1;
17
18                         if indexR>nROW
19                             indexR=indexR-nROW;
20                         end
21
22                         if indexC>nCOLUMN
23                             indexC=indexC-nCOLUMN;
24                         end
25
26                         if OBR(indexR,indexC)==1
27                             nANO=nANO+1;
28                         end
29                     end
30                 end
31             end
32             DP(i,j)=nANO/A;
33         end
34     end
35
36 end

```

Dlužno poznamenat, že tento kód je extrémně výpočetně náročný a zvláště pro větší matice počítané mnohokrát (numerická studie v podkapitole 2.3.3) nepoužitelný. Proto zde byl použit kód pracující s diskretní Fourierovou transformací dle rovnice 2.5.

```

1      function [S2] = TwoPointProbability_FT(OBR)
2      %TWOPOINTPROBABILITY_FT
3      % Vypocet dvoubodovou pravdepodobnost pomoci Fourierovych transformaci
4      % Predpoklada periodicitu
5      %

```

```

6 %   OBR... vstupni matice obrazku
7
8 [nROW,nCOLUMN]=size(OBR);
9 % fftn - rychla Fourierova transformace
10 S2=1/(nROW*nCOLUMN)*ifftn(fftn(OBR).*conj(fftn(OBR)));
11 end

```

### A.3 KLR

Zde je předvedena implementace KLR do kódu se třemi možnými rozděleními proměnných  $\xi$ , normálním, rovnoměrným a založeným na dříve vypočteném histogramu. V hlavním kódu se pak oběhuje funkce "KLR3". Ta slouží pro PCA a liší se pouze tím, že se již neškálují její vektory vlastními čísly, neb je tak už obdržíme z PCA.

```

1 function [obrKLR,out2]=KLR(randgen,cdf,hx,Q1,Q2,c1,fi,lambda,nEigV,
2 precision, varargin)
3 %POMOCI KLR ZPETNE VYGENERUJE OBRAZEK
4 % randgen... metoda generovani nahodnych cisel
5 % cdf... vektor hodnot cdf
6 %c1 - objemove zastoupeni inkluze
7 %fi - matice vlastnich tvaru
8 %lambda - matice vlastnich cisel
9 %nEigV - pocet vlastnich tvaru, ktere chceme v KLR zahrnut, nevime-li ...
   = [],
10 %pak je nutno udat dalsi parametr
11 %precision - pozadovana presnost \in (0,1), neudava se, pokud
12
13 %FCE vraci matici obrazku
14
15 Q1=round(Q1);
16 Q2=round(Q2);
17 % nargin
18 lambda=abs(sort(lambda));
19 % lambda(end)
20 switch nargin
21     case 9
22         precision=sum(lambda(end-nEigV+1:end))/sum(lambda(1:end));
23     %         disp('case nEigV')
24     case 10
25         test=0;
26         Asum=sum(lambda);
27         size(lambda)
28         size(Asum)
29         nEigV=0;
30         Lsum=0;
31         while test<precision
32             nEigV=nEigV+1;
33             Lsum=Lsum+lambda(end-nEigV+1);
34             test=Lsum/Asum;
35         end
36     %         disp('case precision')
37 end
38
39 lambda=sqrt(lambda);

```



```

40 %skalovani vlastnich vektoru
41 for i=1:length(lambda)
42     fi(:,i)=lambda(i)*fi(:,i);
43 end
44
45 %KRL soucet
46 obrKLR=zeros(sqrt(length(fi(:,end)))));
47
48 % Vyber rozdeleni
49 switch randgen
50     case 'original'
51         % vrati puvodni obr... misto cdf zadat \xi
52         xi=cdf(end-nEigV+1:end);
53     case 'uniform'
54         xi=randi([Q1 Q2],length(fi(:,end)),1);
55     case 'gauss-normal'
56         % misto Q1 zadej \mu
57         % misto Q2 zadej \sigma
58         mu=Q1;
59         sigma=Q2;
60         xi=normrnd(mu,sigma,[length(fi(:,end)),1]);
61     case 'hist-based'
62         xi=rand(length(obrKLR(:)),1);
63         xi=interp1(hx,cdf,xi);
64 end
65
66 obrKLR(:)=c1+fi(:,end-nEigV+1:end)*xi(end-nEigV+1:end);
67
68 obrKLR=SetTreshold(c1,obrKLR);
69
70 %Vystup 2 fce
71 switch nargin
72     case 9
73         out2=precision;
74     case 10
75         out2=nEigV;
76 end
77
78 end

```

Vzhledem k náhodné úpravě vektorů dostáváme z původně binární domény, která nabývala pouze hodnot 0/1 pole reálných čísel. Proto v KLR voláme funkci *SetTreshold*, která určí horní kvantil dle požadovaného objemového zastoupení a vrátí logické hodnoty (1/0) podle toho, zda jsou hodnoty v generovaném poli větší nebo menší.

```

1 function [bool] = SetTreshold (c1, obrKLR)
2 % Nastavi prah (threshold) pro dany vysledek KLR tak, aby odpovidalo
3 % objemove zastoupeni inkluze
4 % c1 - objemove zastoupeni inkluze
5 % obrKLR - matice obrazku ziskana z KLR
6
7 Q=quantile(obrKLR(:),1-c1);
8
9 % bool=obrKLR(:, :) ≤ Q;
10 bool=obrKLR(:, :) ≥ Q;
11 end

```

### A.3.1 Hlavní skript

Propojující kód. Významné funkce byly uvedeny výše. U některých funkcí s mnoha argumenty jsou tyto z důvodu omezeného místa v této příloze naznačeny jako "...".

```

1  clc;
2  clear variables;
3  close all;
4
5  %% ZAKLADNI PARAMETRY POZADOVANEHO OBRAZKU
6
7  % OBJEMOVE ZASTOUPENI INKLUZE (BILa)
8  V_inkluze = 0.1;
9  %ZESTUP INKLUZE (BILa)
10 DIST=1; %[px]
11
12 % CTVERCOVY ROZMER INKLUZE
13 x_ink=3; %[px]
14 y_ink=12; %[px]
15 % ROZMER OBRaZKU
16 nROW=50; %[px] %Radky
17 nCOLUMN=50; %[px] %Sloupce
18 % Nasobek presnosti vykreslovani chyb
19 % p=2 %[-]
20
21 %% VYPOCET NAHODNEHO OBRaZKU
22 OBR=RandomPictureGenerator(V_inkluze, DIST, x_ink, y_ink, 2*nROW, ...
    2*nCOLUMN);
23
24 %% DVOUBODOVA PRAVDEPODOBNST (Predpokladana periodicka struktura, pro ...
    bilou fazi 1 )
25 %Vypocet pomoci Fourierovy Transformace
26 OBR=RandomPictureGenerator(V_inkluze, DIST, x_ink, y_ink, nROW, ...
    nCOLUMN);
27 DP=TwoPointProbability_FT(OBR);
28 DP1=DP;
29 %VYPOCET Z DEFINICE
30 % DP=TwoPointProbability(OBR);
31 %% KOVARIACNI JaDRA A HODNOCENI JEJICH KVALITY
32
33 % SMERODATNa ODCHYLKA A ROZPTYL
34 SIG=(DP(1,1)-DP(1,1)^2)^0.5; SIG_2=SIG^2;
35 lambda_1=0; lambda_2=1; DELTA_lam=lambda_1-lambda_2;
36
37 % JaDRO ZJISTENE Z DVOUBODOVE PRAVDEPODOBNOSTI
38 C_dp=Cf_image(DP, DELTA_lam);
39 C_dp1=Cf_image(DP1, DELTA_lam);
40 C_dp2=Cf_image(DP2, DELTA_lam);
41
42 %% OTIMALIZACE Lx a Ly
43 % % L=[Lx, Ly]
44 % Pocatecni bod
45 x0=[1 1];
46
47 % --- FUNKCE PRO VYPOCET CHYB
48 % Pro Gaussovo kovariancni jadro
49 fmin_g=@(L) sum((C_dp-Cf_gauss(DP,L,SIG_2)).^2,'all');
```

```

50 fmin_g_err=@(L) (abs(C_dp-Cf_gauss(DP,L,SIG_2)));
51 % Pro exponencialni kovariancni jadro
52 fmin_e=@(L) sum((C_dp-Cf_exp(DP,L,SIG_2)).^2,'all');
53 fmin_e_err=@(L) (abs(C_dp-Cf_exp(DP,L,SIG_2)));
54
55 % --- SAMOTNa OPTIMALIZACE
56 % Pro Gaussovo kovariancni jadro
57 % [L_gauss_opt,ERR_gauss]=fminsearch(fmin_g,x0);
58 [L_gauss_opt,ERR_gauss]=fmincon(fmin_g,x0,[],[],[],[],[],[0,0],[nROW,nCOLUMN]);
59 C_gauss_opt=Cf_gauss(DP,L_gauss_opt,SIG_2);
60 C_gauss_opt_err=fmin_g_err(L_gauss_opt);
61
62 % Pro exponencialni kovariancni jadro
63 % [L_exp_opt,ERR_exp]=fminsearch(fmin_e,x0);
64 [L_exp_opt,ERR_exp]=fmincon(fmin_e,x0,[],[],[],[],[],[0,0],[nROW,nCOLUMN]);
65 C_exp_opt=Cf_exp(DP,L_exp_opt,SIG_2);
66 C_exp_opt_err=fmin_e_err(L_exp_opt);
67
68 %% VYPOCET CHYBY V ZaVIsLOSTI NA DELE
69
70 CHL_gauss=zeros(nROW,nCOLUMN);
71 CHL_exp=zeros(nROW,nCOLUMN);
72 %
73 % % parpool(4);
74 for j=1:nCOLUMN
75     for i=1:nROW
76         %CHYBA GAUSSOVSKÉHO JaDRA
77         CHL_gauss(i,j)=fmin_g([i,j]);
78         %CHYBA EXPONENCIÁLNÍHO JaDRA
79         CHL_exp(i,j)=fmin_e([i,j]);
80     end
81 end
82 % delete(gcf('nocreate'))
83
84 %% UMELE POTLACENI PERIODICITY FUNKCE PRO LEPSI POROVNANI S ...
    KOVARIANCNIMI JaDRY
85
86 % NOVY SOURADNY SYSTEM
87 if rem(nROW,2)>0
88     ind_R=(nROW+1)/2;
89 else
90     ind_R=nROW/2;
91 end
92
93 if rem(nCOLUMN,2)>0
94     ind_C=(nCOLUMN+1)/2;
95 else
96     ind_C=nCOLUMN/2;
97 end
98
99 [Xs,Ys]=meshgrid(-(nCOLUMN-ind_C):ind_C-1,-(nROW-ind_R):ind_R-1);
100
101 % REORGANIZACE MATICE DVOUBODOVE PRAVDEPODOBNOSTI
102 NDP=zeros(nROW,nCOLUMN);
103 NDP(1:ind_R,1:ind_C)=DP((nROW-ind_R+1):nROW,(nCOLUMN-ind_C+1):nCOLUMN);
104 NDP(1:ind_R,(nCOLUMN-ind_C+1):nCOLUMN)=DP((nROW-ind_R+1):nROW,1:ind_C);
105 NDP((nROW-ind_R+1):nROW,1:ind_C)=DP(1:ind_R,(nCOLUMN-ind_C+1):nCOLUMN);
106 NDP((nROW-ind_R+1):nROW,(nCOLUMN-ind_C+1):nCOLUMN)=DP(1:ind_R,1:ind_C);

```

```

107
108 %VYPOCET NOVE KORELACE
109 NC_dp=zeros(nROW,nCOLUMN);
110
111 for i=1:nROW
112     for j=1:nCOLUMN
113         % JaDRO ZJISTENE Z DVOUBODOVE PRAVDEPODOBNOSTI
114         NC_dp(i,j)=(NDP(i,j)-DP(1,1)^2)*DELTA_lam^2;
115     end
116 end
117
118
119 %% OTIMAIZACE A VYPOCET CHYBY RESTRUKTURALIYZOVANE MATICE
120 % % L=[Lx, Ly]
121 % Pocatecni bod
122 x0=[1 1];
123
124 % ... FUNKCE PRO VYPOCET CHYB
125 % Pro Gaussovo kovariancni jadro
126 fmin_gN=@(L) sum((NC_dp-NCf_gauss(NDP,L,SIG_2,ind_R,ind_C)).^2,'all');
127 fmin_gN_abs=@(L) ...
128     sum(abs(NC_dp-NCf_gauss(NDP,L,SIG_2,ind_R,ind_C)),'all');
129 % Pro exponencialni kovariancni jadro
130 fmin_eN=@(L) sum((NC_dp-NCf_exp(NDP,L,SIG_2,ind_R,ind_C)).^2,'all');
131 fmin_eN_abs=@(L) sum(abs(NC_dp-NCf_exp(NDP,L,SIG_2,ind_R,ind_C)),'all');
132 fmin_eN_err=@(L) (abs(NC_dp-NCf_exp(NDP,L,SIG_2,ind_R,ind_C)));
133
134 % ... OPTIMALIZACE
135 % Pro Gaussovo kovariancni jadro
136 % [N-L-gauss-opt,N-ERR-gauss]=fminsearch(fmin_gN,x0);
137 [N-L-gauss-opt,N-ERR-gauss]=fmincon(fmin_gN,x0,[],[],[],[], \
138 [-nROW,-nCOLUMN],[nROW,nCOLUMN]);
139 NC_gauss_opt=NCf_gauss(NDP,N-L-gauss_opt,SIG_2,ind_R,ind_C);
140 NC_gauss_opt_err=fmin_gN_err(N-L-gauss_opt);
141 % Pro exponencialni kovariancni jadro
142 % [N-L-exp-opt,N-ERR-exp]=fminsearch(fmin_eN,x0);
143 [N-L-exp-opt,N-ERR-exp]=fmincon(fmin_eN,x0,[],[],[],[], \
144 [-nROW,-nCOLUMN],[nROW,nCOLUMN]);
145 NC_exp_opt=NCf_exp(NDP,N-L-exp_opt,SIG_2,ind_R,ind_C);
146 NC_exp_opt_err=fmin_eN_err(N-L-exp_opt);
147
148 %% VYPOCET CHYBY RESTRUKTURIZOVANYCH MATIC V ZAVISLOSTI NA DELCE
149
150 NxLX=linspace(-(ind_R+nROW),ind_R-1,nROW);
151 NyLY=linspace(-(ind_C+nCOLUMN),ind_C-1,nCOLUMN);
152 %
153 NCHL_gauss=zeros(nROW,nCOLUMN);
154 NCHL_exp=zeros(nROW,nCOLUMN);
155 NCHL_gauss_abs=zeros(nROW,nCOLUMN);
156 NCHL_exp_abs=zeros(nROW,nCOLUMN);
157
158
159 for i=(-ind_R+nROW):ind_R-1
160     for j=(-ind_C+nCOLUMN):ind_C-1
161         %CHYBA GAUSSOVSKÉHO JADRA
162         NCHL_gauss(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
163         =fmin_gN([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);

```

```

164     NCHL_gauss_abs(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
165     =fmin_gN_abs([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);
166     %CHYBA EXPONENCIaLNIHO JaDRA
167     NCHL_exp(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
168     =fmin_eN([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);
169     NCHL_exp_abs(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
170     =fmin_eN_abs([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);
171     end
172 end
173
174 %% PREUSPORADANI KOVARIANCNI MATICE PRO KRL
175 C_gauss=Cf_gauss(DP,N_L_gauss_opt, SIG_2);
176 C_exp=Cf_exp(DP,N_L_exp_opt, SIG_2);
177
178 %KOVARIANCI MATICE VZNIKLE POSUMEN
179 RC_gauss=PointCovMatrix('gauss',N_L_gauss_opt, DP, SIG_2);
180 RC_exp=PointCovMatrix('exp',N_L_exp_opt, DP, SIG_2);
181 RC_image=PointCovMatrix('image',[], C_dp, DELTA_lam);
182 RC_image_q=PointCovMatrix('image',[], C_dp2, DELTA_lam);
183 %REALIZACE KAPPA PRO PCA
184 SC_dp=PointSamMatrix(DP(1,1),x_ink,y_ink,'image',N_L_exp_opt, OBR, ...
185     DELTA_lam);
186
187 %% ROZKLAD DO VLASTNICH VEKTORU
188 % close all;
189
189 % MET 1 - ROZKLAD KOVAR MCE DO VL. TVARU
190 [Eig_V_image,Eig_D_image]=eig(RC_image);
191 [Eig_V_gauss,Eig_D_gauss]=eig(RC_gauss);
192 [Eig_V_exp,Eig_D_exp]=eig(RC_exp);
193
194 % MET 2 - PCA
195 [Eig_V_image_pca,Xi_image_pca,Eig_D_image_pca]=
196 =pca(SC_dp(:,1:1.1*nROW*nROW)');
197
198 C_ensemble=cov(SC_dp');
199 size(C_ensemble)
200
201 Xi_image_pca=Xi_image_pca';
202 % MET 3
203 [Eig_V_image_shift,Eig_D_image_shift]=eig(RC_image);
204
205
206 %% SERAZENI VL. CISEL A TVARU
207 Eig_D_image=diag(Eig_D_image);
208 Eig_D_image_q=diag(Eig_D_image_q);
209 Eig_D_gauss=diag(Eig_D_gauss);
210 Eig_D_exp=diag(Eig_D_exp);
211 Eig_D_image_shift=diag(Eig_D_image_shift);
212 Eig_D_gauss_shift=diag(Eig_D_gauss_shift);
213 Eig_D_exp_shift=diag(Eig_D_exp_shift);
214
215 %preusporadani do vzestupneho poradí
216 Eig_V_image_pca= SortEigVec(Eig_V_image_pca);
217 Eig_D_image_pca=sort(Eig_D_image_pca,'ascend');
218
219 disp('Zornhaw')
220

```

```

221 %% XI POMOCI KLR (KARHUNEN-LOEVE ROZVOJ)
222 % MET 1
223 Xi_image=FindXi(DP(1,1),x_ink,y_ink, nROW, nCOLUMN, Eig_D_image, ...
    Eig_V_image);
224 Xi_gauss=FindXi(DP(1,1),x_ink,y_ink, nROW, nCOLUMN, Eig_D_gauss, ...
    Eig_V_gauss);
225 Xi_exp=FindXi(DP(1,1),x_ink,y_ink, nROW, nCOLUMN, Eig_D_exp, Eig_V_exp);
226 % MET 2 - \xi z pca drive ziskany
227 % MET 3 - shift
228 Xi_image_shift=FindXi3(OBR, DP(1,1),Eig_D_image, Eig_V_image);
229
230 %% Marginalni hustoty pravdepodobnosti histogramy
231 r=5; n=10;
232 % size(Xi_image)
233 [HyperHist_image,hxi,MIN_image,MAX_image]=xi_hist(Xi_image,...);
234 [HyperHist_gauss,hxg,MIN_gauss,MAX_gauss]=xi_hist(Xi_gauss,...);
235 [HyperHist_exp,hxe,MIN_exp,MAX_exp]=xi_hist(Xi_exp,...);
236 % MET 2 - pca
237 [HyperHist_image_pca,hxip,MIN_image_pca,MAX_image_pca]=
238 =xi_hist(Xi_image_pca,...);
239 % MET 3 - SHIFT
240 [HyperHist_image_shift,hxis,MIN_image_shift,MAX_image_shift]=
241 =xi_hist(Xi_image_shift,...);
242 image_shift_q,hxisq,MIN_image_shift_q,MAX_image_shift_q]=
243 =xi_hist(Xi_image_shift_q,...);
244
245 %% Distribucni fce
246 Dist_image=cdf_databased('Image-S2-rnd',hxi,HyperHist_image,...);
247 Dist_gauss=cdf_databased('Gauss-rnd',hxg,HyperHist_gauss,...);
248 Dist_exp=cdf_databased('Exp-rnd',hxe,HyperHist_exp,...);
249 % MET 2 - pca
250 Dist_image_pca=cdf_databased('PCA',hxip,HyperHist_image_pca,...);
251 % MET 3 - SHIFT
252 Dist_image_shift=
253 =cdf_databased('Image-SHIFT',hxis,HyperHist_image_shift,...);
254
255 %% Zpetne generovani obrazku KLR
256 close all;
257
258 % Nastaveni vysledneho porovnani
259 nOut=10; nTest=1;
260 step=0.025; ending=0.05;
261 A=zeros(ending/step,1);
262 % Chyba vuci autokovarianci funkci
263 errmat=zeros(nTest,nOut);
264 ERRmat=zeros(length(A),nOut);
265 errmatq=zeros(nTest,nOut);
266 ERRmatq=zeros(length(A),nOut);
267 precmat=zeros(nTest,nOut);
268 PRECmat=zeros(length(A),nOut);
269
270 % Chyba vuci kovarianci souboru
271 errensemble=zeros(length(A),nOut);
272 storesample=zeros(nTest, nROW*nCOLUMN, nOut);
273
274 k=0;
275
276 for j=step:step:ending

```

```

277
278     k=k+1;
279     a=round(j*nROW*nCOLUMN);
280     A(k)=a;
281
282     for i=1:nTest
283
284         % % % POMOCI POCTU ZAHRNUTYCH VL. TVARU
285         % % Uniform
286         [OBR_image_shift_u,pimage_shift_u]=KLR('uniform',hxis,...);
287
288         % % Gauss/normal
289         sig=abs(DELTA_lam);
290         [OBR_image,pimage]=KLR('gauss-normal',...);
291         [OBR_gauss,pgauss]=KLR('gauss-normal',...);
292         [OBR_exp,pexp]=KLR('gauss-normal',...);
293         [OBR_image_pca,pimage_pca]=KLR3('gauss-normal',...);
294         [OBR_image_shift,pimage_shift]=KLR('gauss-normal',...);
295         [OBR_image_shift_q,pimage_shift_q]=KLR('gauss-normal',...);
296         % % hist-based
297         [OBR_image_h,pimage_h]=KLR('hist-based',...);
298         [OBR_image_pca_h,pimage_pca_h]=KLR3('hist-based',...);
299         [OBR_image_shift_h,pimage_shift_h]=KLR('hist-based',...);
300         [OBR_image_shift_q_h,pimage_shift_q_h]=KLR('hist-based',...);
301
302     %         DATA GAUSS
303     precmat(i,1)=pimage;
304     precmat(i,2)=pgauss;
305     precmat(i,3)=pexp;
306     precmat(i,4)=pimage_pca;
307     %         precmat(i,5)=pimage_h;
308     precmat(i,6)=pimage_pca_h;
309     precmat(i,7)=pimage_shift_h;
310     precmat(i,8)=pimage_shift_u;
311     precmat(i,9)=pimage_shift_q;
312     precmat(i,10)=pimage_shift_q_h;
313
314     % % HODNOCENI
315     %kovariance vuci periodickemu mediu
316     errmat(i,1)=err_cov_one(OBR_image,DELTA_lam,C_dp);
317     errmat(i,2)=err_cov_one(OBR_gauss,DELTA_lam,C_dp);
318     errmat(i,3)=err_cov_one(OBR_exp,DELTA_lam,C_dp);
319     errmat(i,4)=err_cov_one(OBR_image_pca,DELTA_lam,C_dp);
320     %         errmat(i,5)=err_cov_one(OBR_image_h,DELTA_lam,C_dp);
321     errmat(i,6)=err_cov_one(OBR_image_pca_h,DELTA_lam,C_dp);
322     errmat(i,7)=err_cov_one(OBR_image_shift_h,DELTA_lam,C_dp);
323     errmat(i,8)=err_cov_one(OBR_image_shift_u,DELTA_lam,C_dp);
324     errmat(i,9)=err_cov_one(OBR_image_shift_q,DELTA_lam,C_dp);
325     errmat(i,10)=err_cov_one(OBR_image_shift_q_h,DELTA_lam,C_dp);
326
327     % ulozeni pro porovnani se souborem
328     storesample(i,:,1)=OBR_image(:);
329     storesample(i,:,2)=OBR_gauss(:);
330     storesample(i,:,3)=OBR_exp(:);
331     storesample(i,:,4)=OBR_image_pca(:);
332     %         storesample(i,:,5)=OBR_image_h(:);
333     storesample(i,:,6)=OBR_image_pca_h(:);
334     storesample(i,:,7)=OBR_image_shift_h(:);

```

```
335     storesample(i,:,8)=OBR_image_shift_u(:);
336     storesample(i,:,9)=OBR_image_shift_q(:);
337     storesample(i,:,10)=OBR_image_shift_q_h(:);
338
339     end
340
341     % VYPOCET CHYBY VUCI KOVARIANCI SOUBORU
342     for i=1:nOut
343         CC=cov(storesample(:, :, i));
344         errensemble(k,i)=sum(abs(C_ensemble-CC),'all');
345     end
346
347     ERRmat(k,:)=mean(errmat,1);
348     PRECmat(k,:)=mean(precmat,1);
349
350 end
```



# Bibliografie

- [1] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer Science & Business Media, 2013, sv. 16.
- [2] J. Havelka, A. Kučerová a J. Sýkora, “Compression and reconstruction of random microstructures using accelerated lineal path function”, *Computational Materials Science*, roč. 122, s. 102–117, 2016.
- [3] M. Lombardo, J. Zeman, M. Sejnoha a G. Falsone, “Stochastic modeling of chaotic masonry via mesostructural characterization”, *International Journal for Multiscale Computational Engineering*, roč. 7, č. 2, 2009.
- [4] V. Hlaváč, “Fourierova transformace v 1D a 2D”, *VYSOKÉ UČENÍ TECHNICKÉ V PRAZE, České, Sv. Přednáška*, [cit. 1.12. 2013]. Dostupné z: <http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/12FourierTxCz.pdf>, 2012.
- [5] A. Kučerová, J. Sýkora a J. Zeman, *STOCHASTIC MODELLING OF HETEROGENEOUS MATERIALS BASED ON IMAGE ANALYSIS*, květ. 2014. WWW: <https://www.engmech.cz/improc/2014/73-Kucerova-CD.pdf>.
- [6] J. Havelka, “Efektivní metody pro propagování nejistot v popisu proudění podzemní vody”, 2014.
- [7] B. Minasny a A. B. McBratney, “The Matérn function as a general model for soil variograms”, *Geoderma*, roč. 128, č. 3-4, s. 192–207, 2005.
- [8] H. Jan, “Aplikace hraničních inverzních metod ve stavebním inženýrství”, dis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2019.
- [9] K. Rektorys, *Přehled užití Matematiky*, (7. vydání), 2000.
- [10] F. W. J. Olver a L. C. Maximon, *Chapter 10 Bessel Functions*, pros. 2020. WWW: <https://dlmf.nist.gov/10>.
- [11] *Bessel Function of the First Kind*. WWW: <https://mathworld.wolfram.com/BesselFunctionoftheFirstKind.html>.
- [12] E. M. Society, *Bessel functions*. WWW: [https://encyclopediaofmath.org/wiki/Bessel\\_functions](https://encyclopediaofmath.org/wiki/Bessel_functions).