

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



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

Bakalářská práce

David Šilhánek

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

Plzeň, Květen 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, květen 2021

.....
David Šilhánek

Poděkování

V první řadě bych chtěl poděkovat vedoucí mé práce, doc. Ing. Anně Kučerové, Ph.D., za skutečně aktivní přístup k vedení, bez něhož bych se v pro mě jinak zcela novém tématu orientoval jen stěží. Po celou dobu mne směřovala, nikdy nenechala tápat ani kráčet slepými uličkami.

Nemenší vděk si zaslouží i prof. RNDr. Daniela Jarušková, CSc., za její poskytnuté konzultace ohledně matematické formulace problému, pravděpodobnosti a statistiky. Její připomínky mi velmi pomohly dát si některé poznatky do souvislostí.

Obě výše jmenované osoby pak sklízí můj obdiv především za své neutuchající odhodlání a snahu pomoci mi s pochopením proniknout alespoň na okraj velmi rozsáhlé problematiky. A musím uznat, že jejich úkol nebyl vždy snadný.

Zadání

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta stavební
Thákurova 7, 166 29 Praha 6

Shadyon



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Šilhánek	Jméno: David	Osobní číslo: 477228
Zadávatel: katedra: K132 - Katedra mechaniky		
Studijní program: Stavební inženýrství		
Studijní obor: Konstrukce pozemních staveb		

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce: Popis morfologie heterogenních materiálů pomocí náhodných polí odvozených z obrazové analýzy

Název bakalářské práce anglicky: Image-based Random Fields in Description of Heterogeneous Morphology

Pokyny pro vypracování:
Student se v bakalářské práci bude věnovat následujícím tématům: i) dvoubodová pravděpodobnostní funkce [1], ii) konstrukce náhodných polí za pomoci Karhunen-Loèveho rozvoje [2] a iii) využití obrazového podkladu při konstrukci kovariančních jader náhodných polí [3]. Cílem bakalářské práce je syntetizovat výše uvedené body a představit porovnání konstrukcí náhodných polí, které se začínají v současné době těšit oblibě v inženýrské komunitě.

Seznam doporučené literatury:
[1] Havelka, Jan, Anna Kučerová, and Jan Sýkora. (2016) Compression and reconstruction of random microstructures using accelerated lineal path function." Computational Materials Science 122, pp. 102-117.
[2] Adler, R. J. & Taylor, J. E. (2007) Random Fields and Geometry. Springer, 2007.
[3] Lombardo, M., Zeman, J., Sejnoha, M. & Falsone, G. (2009) Stochastic modeling of chaotic masonry via mesostructural characterization. Int. Journal for Multiscale Computational Engineering, 7, 2, pp.171-185.

Jméno vedoucího bakalářské práce: Doc., Ing. Anna Kučerová, Ph.D.

Datum zadání bakalářské práce: 12. 2. 2021 Termín odevzdání bakalářské práce: 16. 5. 2021
Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku

Anna Kučerová
Podpis vedoucího práce

a
Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v bakalářské práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

13. 2. 2021
Datum převzetí zadání

David Šilhánek
Podpis studenta(ky)

Skenováno pomocí CamScanner

Abstrakt

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

Klíčová slova: Náhodná 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 correlation functions known from literature which are widely used for this purpose. Samples of random fields are generated using the Karhunen-Loève expansion. This thesis is written in Czech.

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

Seznam tabulek

2.1	Koeficienty aproximace pro Gaussovskou optimální korelační délku L pro čtvercovou inkluzi, L_x pro obdélníkovou inkluzi ve směru konstantní velikosti, L_y pro obdélníkovou inkluzi ve směru proměnné velikosti	20
2.2	Koeficienty aproximace pro exponenciální optimální korelační délku L pro čtvercovou inkluzi, L_x pro obdélníkovou inkluzi ve směru konstantní velikosti, L_y pro obdélníkovou inkluzi ve směru proměnné velikosti	21
3.1	Přehled variant zpětného modelování náhodného pole	30

Seznam obrázků

2.1	Ukázka vláknového kompozitu - příklad dvoufázového média [1]	2
2.2	(a) Vizualizace n-bodové pravděpodobnosti u třífázového média. úsečky - dvoubodová pravděpodobnost (nebo L_2), trojúhelníky - třibodová pravděpo- dobnost, (b) vizualizace nejednoznačnosti určení úsečky pro výpočet L_2 na síťově (pixelově) diskretizovaném médiu.	5
2.3	(a) Dvoubodová pravděpodobnost pro objemové zastoupení inkluze $c^{(1)} =$ 0.1 a velikosti inkluze 5x5 px viz (d), (b) dvoubodová pravděpodobnost pro $c^{(1)} = 0.1$ a inkluze 10x10 px struktury (e), (c) dvoubodová pravděpodobnost ekvivalentní s (b)	8
2.4	Průběh gama funkce $\Gamma(x)$ a její převrácené hodnoty $\frac{1}{\Gamma(x)}$ pro $x \in \mathbb{R}$	12
2.5	Reálná část Besselových modifikovaných funkcí (a) prvního druhu a (b) druhého druhu	14
2.6	Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze $c^{(1)} = 0, 1$ (a) inkluze 5x5 px, (b) inkluze 10x10 px	15
2.7	Průběh absolutní chyby dle rovnice (2.30) pro Gaussovské (a) a expo- nenciální (b) jádro pro inkluze 10x10 - viz obr. 2.3	16
2.8	Růst průměrné optimální korelační délky L_x v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussovské (a) a exponenciální (b) jádro	17
2.9	Růst průměrné minimální chyby (sumy čtverců chyb dle (2.31)) v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussovské (a) a expo- nenciální (b) jádro	17
2.10	(a) Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze $c^{(1)} = 0, 3$ inkluze 5x5 px, (b) příslušná doména	18
2.11	Optimální korelační délka pro Gaussovské jádro s obdélníkovou inkluzí v závislosti na její velikosti a objemovém zastoupení - (a) $L_{x,opt}$ (b) $L_{y,opt}$	18
2.12	Aproximace hodnot L_{opt} pro exponenciální jádro (u Gaussovského jádra je trend podobný). Aproximace (a) rovinou (2.32), (b) hyperbolickým para- boloidem (2.33), (c) hyperbolickým paraboloidem (2.34).	19
3.1	Konvergence vlastních čísel (a) a odhadnuté chyby (b) na doméně 50x50 px pro velikost inkluze 5x5 px, $c^{(1)} = 0.1$	25
3.2	Porovnání konvergence odhadnu chyby v závislosti na velikosti inkluze pro λ_i , $c^{(1)} = 0.1$	26
3.3	Vlastní tvary zjištěné pro doménu 50x50 px, inkluze 3x12 px, $c^{(1)} = 0.1$	27
3.4	Vykreslení realizací v průmětu do roviny vybraných ξ_i a distribuční funkce získané pro (a,b,c) soubor vzorků (PCA), (d,e,f) posouvaný obrázek (SHIFT), u (c,f) oranžová barva značí distribuční funkci $\varphi(\xi)$, modrá příslušný his- togram	29

3.5	Porovnání generovaných obrázků 50x50 px, pro čtvercové inkluze, (a) originální obr. (b) Gaussovské jádro 50 n_ϕ (c) Gaussovské jádro 150 n_ϕ , (d) Gaussovské jádro 500 n_ϕ , (e) jádro z S_2 50 n_ϕ , (f) jádro z S_2 150 n_ϕ , (g) jádro z S_2 500 n_ϕ	31
3.6	Porovnání generovaných obrázků 50x50 px, pro obdélníkové inkluze 3x12 px (a) originální obrázek (b) exp. jádro, 40 n_ϕ , (c) exp. jádro, 500 n_ϕ , (d) varianta SHIFT-H, 40 n_ϕ , (e) varianta SHIFT-H, 500 n_ϕ , (f) varianta PCA, 40 n_ϕ , (g) varianta PCA, 250 n_ϕ , (g) varianta PCA, 500 n_ϕ	31
3.7	Přehled vzorků náhodného pole pro dílčí varianty a obdélníkový tvar inkluze	32
3.8	Přehled vzorků náhodného pole pro dílčí varianty a čtvercový tvar inkluze	33
3.9	Konvergence chyby při porovnání vůči vzorku pro $c^{(1)} = 0,1$ na doméně 50x50 px, velikost inkluze (a) 3x3 px (b) 3x12 px	34
3.10	Konvergence chyby při porovnání vůči souboru pro $c^{(1)} = 0,1$ na doméně 50x50 px, velikost inkluze (a) 3x3 px (b) 3x12 px	35
3.11	Detail konvergence chyby při porovnání vůči vzorku pro $c^{(1)} = 0,1$ na doméně 50x50 px, velikost inkluze (a) 3x3 px (b) 3x12 px	36
3.12	Detail konvergence chyby při porovnání vůči souboru pro $c^{(1)} = 0,1$ na doméně 50x50 px, velikost inkluze (a) 3x3 px (b) 3x12 px	37

Obsah

Poděkování	iv
Zadání	v
Abstrakt	vi
Abstract	vii
Seznam tabulek	viii
Seznam obrázků	ix
1 Úvod	1
2 Popis nehomogenního materiálu	2
2.1 Statistické deskriptory	3
2.2 Materiál s inkluzemi	6
2.3 Dvoubodová pravděpodobnost	7
2.3.1 Fourierova transformace	9
2.4 Kovarianční jádra	10
2.4.1 Gama funkce	12
2.4.2 Besselova funkce	13
2.4.3 Numerická studie optimálních korelačních délek	14
3 Generování vzorků náhodného pole	22
3.1 Karhunenovo–Loèveho rozvoj	22
3.1.1 Kovarianční matice	23
3.1.2 Vlastní čísla a vlastní vektory kovariančních matic	24
3.1.3 Volba náhodných proměnných $\xi_i(\omega)$ v KLR	28
3.1.4 Hodnocení modelů pro generování náhodného pole	29
4 Závěr	38
A Použité kódy a komentář	39
A.1 Náhodné generování binárního obrázku	39
A.2 Dvoubodová pravděpodobnost	41
A.3 KLR	42
A.4 Hlavní skript	44
Bibliografie	51

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 určitém měřítku (mikroskopickém, mesoskopickém etc.) lze většinu materiálů prohlásit za nehomogenní a často je to právě mikrostruktura zodpovědná za makroskopické chování 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 pravděpodobnostní rozdělení vlastností materiálu a následným fyzikálním výpočtem¹ (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 střepe. Naopak u kompozitních materiálů s více fázemi, rozptýlenou výztuží či například u oprav historických staveb, kde můžeme pozorovat výrazné kolísání 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). Abychom se za použití tohoto komplexnějšího přístupu neodchýlili od modelované skutečnosti více, než za pomoci homogenizace (idealizace) vzorku, je vhodné využít veškeré dostupné informace pro co nepřesnější popis.

Jako jedna možnost, která by umožnila získání vstupních údajů, se jeví právě obrazová analýza. Předpokladem je, že máme k dispozici nějaký reprezentativní vzorek (či sérii mnoha vzorků), ze kterého se pokoušíme 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 výsledků².

¹Těmito 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.

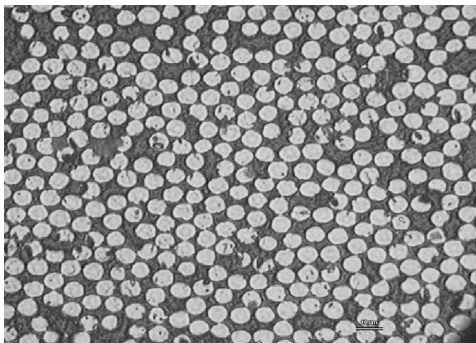
²Jak bude zmíněno i později, některé části tohoto postupu jsou velmi výpočetně náročné, proto by v případě praktického použití bylo vhodné užít programovací jazyk umožňující větší kontrolu práce s pamětí, jakým je např. C/C++.

Kapitola 2

Popis nehomogenního materiálu

Uvažujeme nekonečný heterogenní materiál, který se v tuto chvíli skládá pouze ze dvou fází, jejichž rozložení v objemu je nahodilé a předem nevíme, ve kterém místě se nachází která fáze, viz např. obr. 2.1. Libovolný řez tímto materiálem bude mít rovněž dvě složky, které budeme reprezentovat hodnotami 0 (černá barva) a 1 (bílá barva). Na celý problém se lze dívat i tak, že jsme třetí rozměr zanedbali a vybíráme pouze výřezy z rovinného média. Různé výřezy se tedy budou lišit podle toho, kde provedeme řez daným materiálem, v čemž pro nás spočívá náhodnost celého procesu.

V této práci se pokusíme vytvořit pravděpodobnostní model odvozený z realizace takového výřezu, resp. souboru výřezů. Tyto realizace nám pomohou vystihnout statistické charakteristiky střídání fází.



Obrázek 2.1: Ukázka vláknového kompozitu - příklad dvoufázového média [1]

Ačkoliv to obecně není nezbytně nutné, je výhodnější jak pro matematickou formulaci, tak praktickou aplikaci, doménu výřezu nějakým způsobem diskretizovat. Velmi oblíbený způsob je FEM (*final element method - metoda konečných prvků*), který je v kombinaci s náhodnými poli prezentován např. v [2], avšak vzhledem k prostředí MATLAB, kde je obrázek reprezentován maticí pixelů, je pro nás mnohem jednodušší ponechat tuto diskretizaci.

Každý takovýto řez, pro nás velikosti $n \times m$ px, lze považovat za jednu nezávislou realizaci diskrétního náhodného pole $\{X_{(i,j)}; i = 1, \dots, i = m; j = 1, \dots, j = n\}$, tj. souboru náhodných proměnných $\{X_{(i,j)}\}$, které nabývají v našem případě pouze hodnot z $H = \{0; 1\}$ ¹.

¹Náhodné proměnné $X_{(i,j)}$ určují barvu (výskyt fáze v) pixelu na pozici (i, j) . V obecném případě samozřejmě mohou nabývat více hodnot (pokud by materiál obsahoval více fází), zde však řešíme případ binárního náhodného pole.

Rozdělení náhodného pole je určeno všemi:

- 1-rozměrnými rozděleními $\{X_{(i,j)}\}$,
- 2-rozměrnými rozděleními $\{X_{(i_1,j_1)}, X_{(i_2,j_2)}\}$,
- ...
- $n \cdot m$ -rozměrnými rozděleními $\{X_{(i_1,j_1)}, \dots, X_{(i_n,j_m)}\}$.

V této práci budeme dále předpokládat, že celé náhodné pole je takzvaně *statisticky homogenní*. Pod tím rozumíme, že $\forall k, l, h \in \mathbb{N}$: rozdělení vektoru $(X_{(i_1,j_1)}, \dots, X_{(i_k,j_k)})$ je shodné s rozdělením vektoru $(X_{(i_1+l,j_1+h)}, \dots, X_{(i_k+l,j_k+h)})$. Jinými slovy, že konkrétní místo výběru řezu nebo jeho velikost nemá vliv na jeho vlastnosti. Veškeré $\{X_{(i,j)}\}$, $\{X_{(i_1,j_1)}, X_{(i_2,j_2)}\}, \dots, \{X_{(i_1,j_1)}, \dots, X_{(i_k,j_k)}\}$ tedy mají stejné rozdělení. Pro náš případ tedy platí:

$$P(X_{(i,j)} = 1) = p, \quad \text{kde } p \text{ značí objemové zastoupení fáze 1,}$$

$$P(X_{(i,j)} = 0) = 1 - p,$$

pak střední hodnota $\mathbb{E}(X_{(i,j)}) = p$ a rozptyl $\sigma^2(X_{(i,j)}) = (1 - p) \cdot p$. Dále z homogenity náhodného pole plyne, že dvourozměrné rozdělení:

$$P(X_{(i_1,j_1)} = 1 \cap X_{(i_2,j_2)} = 1) = P(X_{(i_1+l,j_1+h)} = 1 \cap X_{(i_2+l,j_2+h)} = 1), \quad \forall l, h \in \mathbb{N}. \quad (2.1)$$

Tato informace je pro nás ekvivalentní s informací o vzájemné rovnosti korelace dvojice náhodných proměnných:

$$\text{corr}(X_{(i_1,j_1)}, X_{(i_2,j_2)}) = \text{corr}(X_{(i_1+l,j_1+h)}, X_{(i_2+l,j_2+h)}) = \varrho(l, h), \quad \forall l, h \in \mathbb{N}, \quad (2.2)$$

kde $\varrho(l, h)$ je korelační funkce závislá na vzájemné poloze dvou bodů (tj. vzdálenosti l a h). V některých případech korelační funkci budeme uvažovat v součinném tvaru:

$$\varrho(l, h) = \varrho_1(l) \cdot \varrho_2(h), \quad (2.3)$$

kde l , resp. h , je vzdálenost dvou bodů ve svislém, resp. vodorovném, směru. Výhodou funkce (2.3) je, že pokud $\varrho_1(l)$ i $\varrho_2(h)$ jsou pozitivně semidefinitní a nezávislé, pak totéž platí i pro $\varrho(l, h)$ a i z ní odvozené autokovarianční matice.

2.1 Statistické deskriptory

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. Ačkoliv tyto funkce vyjadřují pravděpodobnost, je potřeba mít na paměti, že zde budeme jejich hodnoty vždy odhadovat pomocí relativních četností na pozorovaném výřezu či souboru výřezů. Dva nejpoužívanější deskriptory prezentované v [3] si zde blíže uvedeme. Tyto funkce jsou

obecnější a lze je tak použít i pro popis média o více fázích, *statisticky homogenního i nehomogenního*. Jsou navíc definovány i pro spojité náhodné pole.

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 požadované fázi. Tj. můžeme například při výpočtu tříbodové pravděpodobnosti požadovat, aby pro trojúhelník $x_1x_2x_3$ bylo splněno, že vrchol x_1 bude prvkem černé fáze, x_2 prvkem modré fáze a x_3 prvkem bílé fáze. Toto je znázorněno na obr. 2.2a, kde je daná podmínka splněna pouze v jednom případě (zvýrazněném oranžovou barvou) z vyobrazených pokusů (záměna bodů totiž není možná, jedná se o jinou pravděpodobnost). Zjednodušeně tyto funkce udávají, jaká je pravděpodobnost, že uspořádaná *n*-tice bodů splní předepsané podmínky (každý bod padne do požadované fáze) kdekoli na doméně. Tato práce se blíže věnuje právě speciálnímu případu dvoubodové pravděpodobnosti v sekci 2.3, která je výhodná pro popis binárního média, které v této práci uvažujeme.

Vzhledem k uvažované *statistické homogenitě* náhodného pole lze tuto pravděpodobnost odhadovat dvojím způsobem. Pro konkrétní délku a orientaci vektoru (případně trojice, čtveřice, ..., *n*-tice bodů) vybereme náhodně několik jejích umístění, pro které předpis (2.4) vyhodnotíme, nebo (2.4) vyhodnotíme pro všechna možná umístění dané *n*-tice bodů na konkrétním výřezu, který máme k dispozici. První způsob se hodí především pro velmi velké domény z důvodu snížení výpočetní náročnosti, my se však podržíme v této práci druhého, který nám poskytne méně zkreslenou informaci vlivem výběru. Předpoklad statistické homogenity na reálném vzorku nemusí být splněn. Především pro velmi malé výřezy se mohou vlastnosti u dvou vzorků významně lišit. Proto si při práci s daty je nutno dát pozor, abychom pracovali s dostatečně velkým statisticky reprezentativním vzorkem.

V případě, že by doména *nebyla statisticky homogenní* (tj. zastoupení jedné fáze by například bylo výrazně odlišné u krajů než uvnitř domény), tyto funkce by byly závislé i na konkrétní poloze a pro odhad pravděpodobnosti bychom museli mít k dispozici více vzorků.

Definujme *n*-bodovou pravděpodobnost následující rovnicí:

$$S_n^{f_1, f_2, \dots, f_n}(x_1, x_2, \dots, x_n) = \left\langle \prod_{j=1}^n \chi^{f_j}(x_j) \right\rangle, \quad (2.4)$$

• kde:

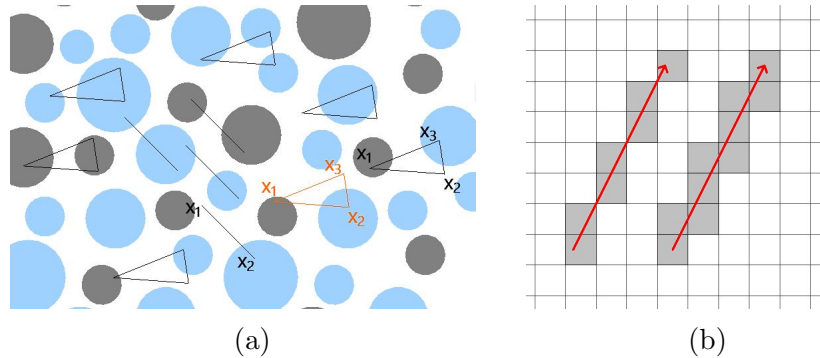
f_j značí předepsanou (sledovanou) fázi pro bod x_j ,
 x_j zastupuje *j*-tý bod v uspořádané *n*-tici bodů,
 $\chi^{f_j}(x_j)$ je charakteristická funkce, která x_j přiřadí hodnotu:

- * 1 - pokud $x_j \in D^{f_j}$, kde D^{f_j} je oblast zaplněná fází f_j ,
- * 0 - pokud $x_j \notin D^{f_j}$,

$\langle \cdot \rangle$ značí aritmetický průměr součinů charakteristických funkcí χ^{f_j} všech možných (resp. v rámci odhadu všech vybraných) umístění (pokusů) dané uspořádané *n*-tice bodů na referenčním výřezu náhodného pole,

n je počet bodů.

Pokud bychom tedy chtěli odhadnout pravděpodobnost $P(X(x_1) = 1 \cap X(x_2) = 1)$, označenou jako $S_2^{(1)}(x_1, x_2)$ ², kde x_1 je od x_2 vzdálen o l ve vodorovném a h ve svislém směru, budeme u *statisticky homogenní domény* v souladu s (2.4) postupovat tak, že vybereme náhodně několik (co nejvíce) dvojic bodů vzdálených o l , resp. h , ve vodorovném, resp. svislém, směru a vyhodnotíme, v kolika pokusech oba body ve dvojici ležely ve fázi 1. Pro *statisticky nehomogenní doménu* bychom tento pokus vyhodnocovali v absolutní pozici bodů x_1 a x_2 avšak na různých realizacích náhodného pole. Obdobně bychom postupovali u dalších odhadů $P(X(x_1) = 1 \cap X(x_2) = 0)$ ³, $P(X(x_1) = 1 \cap X(x_2) = 1 \cap X(x_3) = 1)$ etc.



Obrázek 2.2: (a) Vizualizace n -bodové pravděpodobnosti u třífázového média. úsečky - dvoubodová pravděpodobnost (nebo L_2), trojúhelníky - třibodová pravděpodobnost, (b) vizualizace nejednoznačnosti určení úsečky pro výpočet L_2 na síťově (pixelově) diskretizovaném médiu.

Dalším statistickým deskriptorem je kupříkladu **”úsečková funkce”** (*lineal-path function*). Na rozdíl od n -bodové (resp. dvoubodové) pravděpodobnosti sleduje splnění podmínky předepsaných fází na celé úsečce, nikoliv jen v krajních bodech. $L^{(1)}(\mathbf{r})$ by tak značilo pravděpodobnost, že (vázaný) vektor \mathbf{r} leží celý ve fázi 1, $L^{(1)(2)}(\mathbf{r})$, že leží částečně ve fázi 1, částečně ve fázi 2 atd. Odtud je zřejmé, že i tento deskriptor lze zobecnit pro libovolné množství fází, nejčastěji se však používá právě ve formě $L^f(\mathbf{r})$ právě proto, že obsahuje (sic velmi hrubou) informaci o spojitosti dané fáze f .

$$L^{f_1, f_2, \dots, f_n}(\mathbf{r}) = \langle \chi_L(\mathbf{r}) \rangle, \quad (2.5)$$

- kde:

- \mathbf{r} je (vázaný) vektor určený libovolnými dvěma body média,
- f_j značí j -tou požadovanou fázi (zde na rozdíl od S_2 nezáleží na pořadí, může se vyskytovat kdekoliv na úsečce, viz dále),
- n značí počet různých fází, kterými má úsečka procházet,
- $\chi_L(\mathbf{r})$ je charakteristická funkce, která \mathbf{r} přiřadí hodnotu:

²V případě, že n -bodovou pravděpodobnost počítáme pro stejnou fázi u všech bodů zjednodušíme značení následujícím způsobem $S_2^{(1),(1)}(x_1, x_2) \rightarrow S_2^{(1)}(x_1, x_2)$.

³Zde bychom sledovali, kdy první bod padl do fáze 1 a druhý do fáze 0.

- * 1 - pokud $\forall j, j = 1, \dots, n : \exists \delta \mathbf{r} \subset \mathbf{r} : \delta \mathbf{r} \subset D^{f_j}$, kde D^{f_j} je oblast zaplněná fází f_j , $\wedge \nexists \delta \mathbf{r} \subset \mathbf{r} : \delta \mathbf{r} \subset D^{f_k}$, kde fáze $f_k \notin \{f_1, f_2, \dots, f_n\}$,
- * 0 - v ostatních případech

$\langle \cdot \rangle$ značí aritmetický průměr součinů charakteristických funkcí χ_L všech možných (resp. v rámci odhadu všech vybraných) umístění (pokusů) vektoru \mathbf{r} na referenčním výřezu náhodného pole,

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.

2.2 Materiál s inkluzemi

Neboť bílou fází zde budeme modelovat jako izolované shluky, nikoliv kontinuum, hovoříme zde o bílé barvě, resp. konkrétním jednom shluku, jako o inkluzi. Toto může reprezentovat například pórovitý materiál (kde by pór představoval inkluzi), výše uvedený vláknový kompozit (viz obr. 2.1), polystyrenbeton atp.

Jak bylo prezentováno v [4], $L^f(\mathbf{r})$ na rozdíl od $S_2^f(x_1, x_2)$ klade důraz na tvar inkluze. $S_2^f(x_1, x_2)$ 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. Dalším důvodem, proč byl zvolen právě tento deskriptor je fakt, že při diskretizaci pomocí čtvercové sítě, kterou aplikujeme, je S_2 resp. S_n , jednoznačně aplikovatelná pro všechny směry. Při užití L_2 je nutno zvolit algoritmus pro aproximaci "úsečky" na čtvercové síti, neboť ta zde není jednoznačně určena, jak prezentuje obr. 2.2b. Před popisem samotné dvoubodové pravděpodobnosti v sek. 2.3 se ještě krátce zmíníme o tom, jak je struktura reprezentována v této práci.

Inkluze 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 snadno upravit jejich rozměry a přitom aproximují "skutečné tvary" (např. rozptýleného 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 (výřez) převést do binární podoby. Nejjednodušším způsobem je využití prahových metod (*thresholding*). Obrázek nejprve převedeme do černo-bílé podoby (z tříkanálového RGB tak získáme pouze jednonábový) 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 zatím pro naše účely bohatě postačí.

2.3 Dvoubodová pravděpodobnost

Funkcí dvoubodové pravděpodobnosti (*two-point probability function*) rozumíme předpis, kterým charakterizujeme odhad pravděpodobnosti výskytu jisté fáze u dvojice vybraných bodů. Jedná se o odhad dvourozměrného rozdělení (2.1), proto zde budeme sledovat situaci, kdy oba body padnou do stejné fáze. Takto se snažíme charakterizovat médium 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 $S_2^f(x_1, x_2)$ [3]. Jak bylo zmíněno výše, pro plnohodnotnou charakterizaci domény informace o jednobodové a dvoubodové pravděpodobnosti není z matematického pohledu dostatečná. Nicméně, poskytuje alespoň nějakou informaci o rozložení fází v materiálu a umožňuje sestavit kovarianční matici, kterou budeme dále potřebovat.

Jak celá dvoubodová pravděpodobnost funguje je jasnější z její formální definice [4], kterou dostaneme zjednodušením (2.4):

$$S_2^{f_1, f_2}(x_1, x_2) = \langle \chi^{f_1}(x_1) \chi^{f_2}(x_2) \rangle, \quad (2.6)$$

- kde:

$\chi^{f_j}(x_j)$ je charakteristická funkce viz (2.4),
 $\langle \cdot \rangle$ je aritmetický průměr z různých umístění vektoru $x_1 x_2$ viz (2.4),
 x_1 a x_2 je libovolná dvojice bodů, pro níž pravděpodobnost odhadujeme.

Už zde je vidět, že dvoubodovou pravděpodobnost bude se zachováním diskretizace pomocí pixelů výhodné konstruovat jakožto matici. Funkci dvoubodové pravděpodobnosti, pro periodickou doménu lze také definovat následující rovnicí [4], zde již v podobě upravené pro náš případ diskrétního výpočtu:

$$S_2^f(x_1, x_2) = \frac{1}{n \cdot m} \sum_{x_{1x}=0}^{n-1} \sum_{x_{1y}=0}^{m-1} \chi^f(x_{1x}, x_{1y}) \chi^f((x_{1x} + x_{2x}) \% n, (x_{1y} + x_{2y}) \% m), \quad (2.7)$$

- kde:

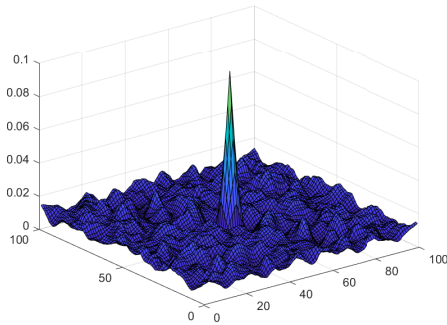
$\%$ je modulo (zbytek po dělení),
 n a m jsou rozměry obdélníkové oblasti, respektive počet řádků a sloupců matice,
 x_{1x} a x_{1y} jsou souřadnice počátečního bodu,
 x_{2x} a x_{2y} jsou souřadnice koncového bodu,
 χ^f značí zobrazení viz rovnice (2.4) .

Tento způsob je sice transparentní a přímočarý, ale poměrně výpočetně náročný. Ovšem za předpokladu periodické domény, lze využít výrazně efektivnější výpočet pomocí Fourierových transformací [4]:

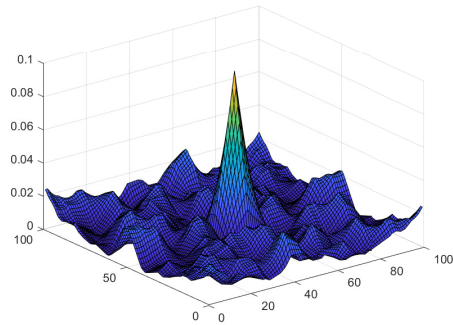
$$S_2^f(x_1, x_2) = \frac{1}{n \cdot m} IDFT \{ DFT \{ \chi^f(x_1, x_2) \} \overline{DFT \{ \chi^f(x_1, x_2) \}} \}, \quad (2.8)$$

- kde:

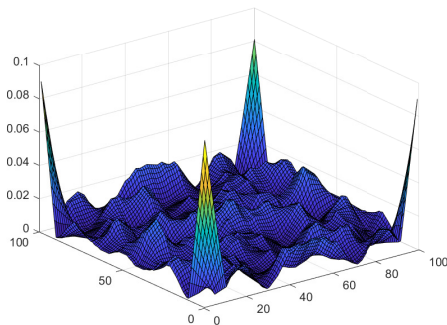
- n a m jsou rozměry obdélníkové oblasti, respektive počet řádků a sloupců matice,
 DFT značí diskrétní dopřednou Fourierovu transformaci,
 $IDFT$ značí diskrétní inverzní Fourierovu transformaci,
 $\overline{\cdot}$ symbolizuje komplexně sdruženou hodnotu,
 x_1 a x_2 je libovolná dvojice bodů, pro níž pravděpodobnost odhadujeme,
 χ^f značí charakteristickou funkci viz rovnice (2.4) .



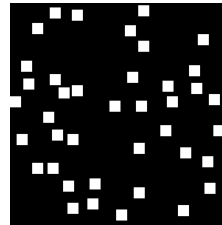
(a)



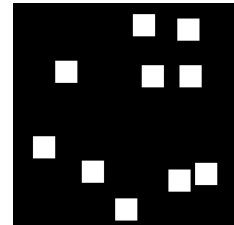
(b)



(c)



(d)



(e)

Obrázek 2.3: (a) Dvoubodová pravděpodobnost pro objemové zastoupení inkluze $c^{(1)} = 0.1$ a velikosti inkluze 5×5 px viz (d), (b) dvoubodová pravděpodobnost pro $c^{(1)} = 0.1$ a inkluze 10×10 px struktury (e), (c) dvoubodová pravděpodobnost ekvivalentní s (b)

Výše zmiňované postupy platí pro výpočet na periodické doméně. Toto zjednodušení jsme si mohli dovolit aplikovat, neboť předpokládáme *statisticky homogenní* náhodné pole (materiál), vedlejší výřezy by se tedy neměly významně lišit. V této práci se ale pokusíme i o několik postupů, kterými se předpokladu periodicity vyhneme, nebo se její vliv omezíme a pokusíme se vyhodnotit, jaké dopady zavedení předpokladu periodicity má.

V této části půjde o výpočet neperiodické S_2 , resp. S_2 s výrazným potlačením vlivu periodicity. Předpoklad *statistické homogeneity* je zde však v obou případech zachován. Aby měly tyto postupy dostatek informací, tj. získané četnosti pro všechny dvojice bodů měly statistickou vypovídající hodnotu pro odhad korelace, v obou případech bude funkce dvoubodové pravděpodobnosti počítána na 4x větším médiu než v případě (2.7). Tzn.

pokud byl původní obraz o velikosti $m \times n$, nyní předpokládáme, že máme k dispozici médium o velikosti $2m \times 2n$.

V prvním případě (který označíme QUARTER) tedy budeme počítat pouze vektory svislé, resp. vodorovné, délky nejvýše m , resp. n . Nejdelší vektory tak mají nejméně realizací, ale stále stejně jako v případě uvažování periodického média. Rovnice pro výpočet tedy bude vypadat následovně:

$$S_2^f(x_1, x_2) = \frac{\sum_{x_{1x}=0}^{(2n-h-1)} \sum_{x_{1y}=0}^{(2m-l-1)} \chi^f(x_{1x}, x_{1y}) \chi^f(x_{1x} + h, x_{1y} + l)}{(2n - h) \cdot (2m - l)}, \quad (2.9)$$

• kde:

- n a m jsou rozměry oblasti, kterou chceme reprezentovat,
- x_1 je bod o souřadnicích $[x_{1x}, x_{1y}]$,
- x_2 je bod o souřadnicích $[x_{1x} + h, x_{1y} + l]$,
- χ^f značí charakteristickou funkci viz rovnice (2.4) .

Druhou možností (označenou dále jako QUARTER-CROP), kterou zde uplatníme, bude výpočet S_2 z obrázku o rozměrech $2n \cdot 2m$ s uvažováním periodicity dle (2.7), resp. (2.8) a následně z takto spočtené S_2 vyřízneme její první čtvrtinu o velikost $n \cdot m$. Tímto oříznutím neztratíme žádnou informaci, neb hodnoty v první čtvrtině matice S_2 spočítané za předpokladu periodicity se v ostatních čtvrtinách pouze opakují, neboť dlouhé vektory přesahující tuto čtvrtinu se chovají jako vektory kratší opačného směru, jak je podrobněji ukázáno v [4].

Na obrázku 2.3 jsou graficky prezentovány funkce dvoubodových pravděpodobností pro konkrétní struktury za předpokladu periodicity. Jak si můžeme povšimnout v obrázku 2.3c, zajímavá místa jsou koncentrována v rozích zájmové oblasti. Neboť ale uvažujeme periodické médium, lze matice přeuspořádat (viz podkapitola 2.4.3) 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. Častý přístup, jak se s těmito problémy vyrovnat, je použití váhových tlumících funkcí [5] aplikovaných na data získaná ze série výřezů za předpokladu jejich periodicity.

2.3.1 Fourierova transformace

Tato podkapitola stručně přibližuje princip fungování Fourierových transformací. Jedná se vlastně o zobrazení funkce na jinou funkci, resp. převedení funkce jedné proměnné na funkci jiné proměnné. Tato operace nám ušetří mnoho výpočetního času především pro větší domény, neboť výpočet S_2 má z definice náročnost $O(n^2m^2)$ ⁴, viz příloha A, kdežto výpočet pomocí *rychlých Fourierových transformací* z rovnice (2.8) vykazuje náročnost pouze $O(mn \log(mn) + mn)$ [4].

Konečně Fourierova transformace je pro funkci jedné proměnné na spojitě doméně definována následovně [5], [6]:

⁴Toto značení udává náročnost programu určenou jako "množství operací, které počítač musí provést na jeho dokončení". U polynomu se udává pouze nejvyšší řád.

- přímá Fourierova transformace:

$$F(v) = \int_{-\infty}^{\infty} f(n) e^{-2\pi i v n} dn, \quad (2.10)$$

- inverzní Fourierova transformace:

$$f(n) = \int_{-\infty}^{\infty} F(v) e^{2\pi i v n} dv. \quad (2.11)$$

- kde:

$F(v)$ je funkce proměnné v ,

$f(n)$ je funkce proměnné n .

V našem případě však pracujeme s diskrétními hodnotami, proto určitý integrál ve Fourierově transformaci přejde na konečný součet. Zároveň zobrazení rovnou rozšíříme pro funkci dvou proměnných tak, aby již byla aplikovatelná pro náš případ:

- přímá diskrétní Fourierova transformace:

$$F(u, v) = \frac{1}{m \cdot n} \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} f(k, j) e^{-2\pi i \left(\frac{j u}{m} + \frac{v k}{n} \right)}, \quad (2.12)$$

- inverzní diskrétní Fourierova transformace:

$$f(k, j) = \sum_{j=0}^{m-1} \sum_{k=0}^{n-1} F(u, v) e^{2\pi i \left(\frac{j u}{m} + \frac{v k}{n} \right)}, \quad (2.13)$$

- kde:

$D(u, v)$ je posloupnost proměnných u a v ,

$d(k, j)$ je posloupnost proměnných k a j ,

m, n jsou rozměry domény,

Nyní přichází na řadu samotný pojem *rychlá Fourierova transformace*, pod kterým rozumíme efektivní algoritmus sloužící k výpočtu *diskrétní Fourierovy transformace* v 1D. Pouhý výpočet pomocí (2.12) by ke zrychlení nevedl, neb vykazuje výpočetní náročnost $O(m^2 n^2)$. Existuje řada algoritmů, které tuto problematiku řeší, nicméně základní myšlenka je založena na rozdělení vstupních dat do více částí a počítat tak sice více transformací, ale pro méně členů, čímž je potlačen vliv kvadratické náročnosti výpočtu [5].

2.4 Kovarianční jádra

Kovarianční jádra nebo též kovarianční funkce jsou zásadní pro popis a generování náhodných polí. V literatuře [7], [8] se běžně uvádí následující kovarianční funkce součinného tvaru (viz (2.3)):

- Gaussovské kovarianční jádro

$$C_G(x_1, x_2) = \sigma_\kappa^2 \cdot e^{-\left(\frac{x_{1x}-x_{2x}}{\sqrt{2}L_x}\right)^2 - \left(\frac{x_{1y}-x_{2y}}{\sqrt{2}L_y}\right)^2}, \quad (2.14)$$

- exponenciální kovarianční jádro

$$C_E(x_1, x_2) = \sigma_\kappa^2 \cdot e^{-\left|\frac{x_{1x}-x_{2x}}{L_x}\right| - \left|\frac{x_{1y}-x_{2y}}{L_y}\right|}, \quad (2.15)$$

- kde:

σ_κ^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 dvě náhodné proměnné nejsou korelovány. To lze definovat následovně [2], [8]:

$$\forall |x_1 - x_2| \geq L : \text{cov}(X(x_1), X(x_2)) \leq \frac{\sigma_\kappa^2}{e} \quad (2.16)$$

x_1 a x_2 jsou zvolené body o souřadnicích $x_1 = [x_{1x}, x_{1y}]$ a $x_2 = [x_{2x}, x_{2y}]$.

S těmito jádry se i nejčastěji pracuje [2], [7]. 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 2.4.3.

Výrazně přesnější je kovarianční funkce založená na dvoubodové pravděpodobnosti [3], [7], kterou lze pohodlně pro binární případ získat jako:

$$C_I(x_1, x_2) = (S_2^{(1)}(x_1, x_2) - (c^{(1)})^2) \cdot (\kappa^{(1)} - \kappa^{(2)})^2, \quad (2.17)$$

- kde:

$S_2^{(1)}(x_1, x_2)$ je hodnota funkce dvoubodové pravděpodobnosti pro body x_1, x_2 ,
 $c^{(1)}$ je objemové zastoupení sledované fáze,
 $\kappa^{(1)}$ a $\kappa^{(2)}$ jsou hodnoty modelované fyzikální veličiny pro každou fázi.

Na rozdíl od výše uvedených jader má tato funkce přímou návaznost na strukturu zkoumaného obrazu a je tak možné ji použít pro validaci funkcí (2.14) a (2.15) a pro optimalizaci jejich korelačních délek. Porovnání výše uvedených funkcí je (pro optimalizované korelační délky) je vyobrazeno na obr. 2.6.

Obě používané funkce, jak Gaussovská, tak exponenciální, jsou ovšem pouze speciálním případem výrazně obecnějšího Matérnova modelu (správněji zřejmě Whittleova-Matérnova modelu viz [9], ale běžně se používá pouze označení Matérnoův model/funkce, kterého se zde podržím).

Matérovu kovarianční funkci lze definovat následujícím předpisem [10]:

$$C_M(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{d}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{d}{\rho} \right), \quad (2.18)$$

• kde:

- σ^2 je rozptyl fyzikální veličiny,
- $\Gamma(\nu)$ je *gama funkce* viz podkap. 2.4.1,
- K_ν je *modifikovaná Besselova funkce druhého druhu* viz podkap. 2.4.2,
- d je vzdálenost (proměnná),
- ν a ρ jsou volené parametry modelu.

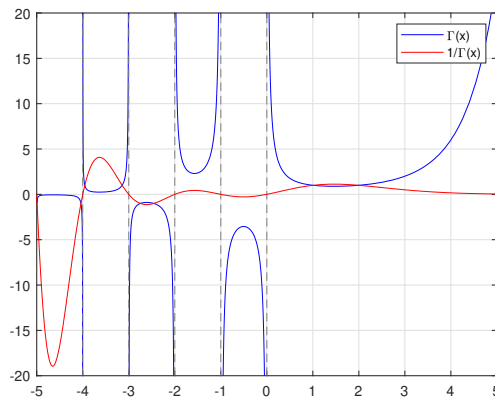
Na rozdíl od rovnic (2.14) a (2.15) se zde vyskytuje pouze jedna proměnná vzdálenosti, avšak s uvážením součinného tvaru (2.3) i Matérovův model samozřejmě lze rozšířit do více dimenzí. Parametr ρ je analogický ke korelační délce L . Ovšem stupeň Besselovy funkce ν již na první pohled tak explicitní význam nemá. Pak vezme, že pro $\nu = \frac{1}{2}$ Matérovův model odpovídá exponenciálnímu kovariančnímu jádru, pro $\nu \rightarrow \infty$ pak přejde na Gaussovské jádro [9].

2.4.1 Gama funkce

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

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt, \quad (2.19)$$

který lze definovat pro $z \in \mathbb{R}^+$, nebo zavést její analytické prodloužení do $\mathbb{C} - \mathbb{Z}_0^-$, kde pak i funkční hodnoty mají reálnou a imaginární složku. Na obrázku 2.4 je pro lepší představu vykreslena gama funkce pro reálnou osu a průběh její převrácené hodnoty, neboť s ní pracuje Matérovu kovarianční jádro.



Obrázek 2.4: Průběh gama funkce $\Gamma(x)$ a její převrácené hodnoty $\frac{1}{\Gamma(x)}$ pro $x \in \mathbb{R}$

Občas se lze rovněž setkat s *pí funkcí* $\Pi(z)$, která je oproti *gama funkci* $\Gamma(z)$ pouze

posunuta o hodnotu 1 na reálné ose:

$$\Gamma(z + 1) = \Pi(z) = \int_0^\infty t^z e^{-t} dt. \quad (2.20)$$

Pro $n \in \mathbb{N}$ tedy platí:

$$\Gamma(n) = (n - 1)!, \quad (2.21)$$

$$\Pi(n) = n!. \quad (2.22)$$

2.4.2 Besselova funkce

Besselovými funkcemi nazýváme takové funkce, které jsou řešením diferenciálních rovnic (2.23) a (2.24) a definované na \mathbb{C} . Besselovy funkce jsou prvního až třetího druhu, "běžné" a modifikované. Zde se zaměříme pouze na popis používané *Besselovy modifikované funkce druhého druhu* a potřebných souvislostí, detailnější informace lze nalézt např. v [11]–[14].

Besselova diferenciální rovnice je:

$$z^2 \frac{\delta^2 w(z)}{\delta z^2} + z \frac{\delta w(z)}{\delta z} + (z^2 - \nu^2)w(z) = 0, \quad (2.23)$$

modifikovaná Besselova diferenciální rovnice:

$$z^2 \frac{\delta^2 w(z)}{\delta z^2} + z \frac{\delta w(z)}{\delta z} - (z^2 + \nu^2)w(z) = 0, \quad (2.24)$$

• kde:

- z proměnná $z \in \mathbb{C}$,
- $w(z)$ je hledaná funkce a
- ν nazýváme řád (či 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 (2.23) či (2.24) jako lineární kombinaci dvou *Besselových funkcí*. Dále uvedu pouze řešení pro modifikovanou rovnici, řešení pro (2.23) jsou analogická a konkrétní funkce lze najít ve výše citované literatuře.

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

$$w(z) = c_1 I_\nu(z) + c_2 I_{-\nu}(z), \quad (2.25)$$

• pokud je $\nu \in \mathbb{N}$:

$$w(z) = c_1 I_\nu(z) + c_2 K_{-\nu}(z), \quad (2.26)$$

• kde:

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

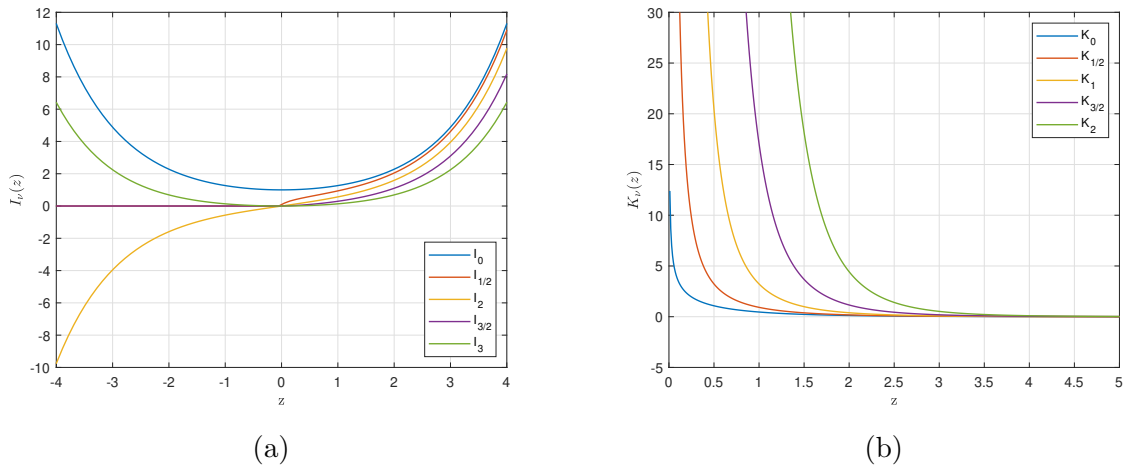
$$I_\nu(z) = \left(\frac{z}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{1}{k! \Gamma(\nu + k + 1)} \left(\frac{z^2}{4}\right)^k, \quad (2.27)$$

K_ν je Besselova modifikovaná funkce druhého druhu vypočtená dle:

$$K_\nu(z) = \frac{\pi}{2} \frac{I_{-\nu}(z) - I_\nu(z)}{\sin(\nu\pi)}, \quad \nu \in \mathbb{C} - \mathbb{N} \quad (2.28)$$

$$K_\nu(z) = \lim_{\nu \rightarrow n} \frac{\pi}{2} \frac{I_{-\nu}(z) - I_\nu(z)}{\sin(\nu\pi)}, \quad \nu = n, n \in \mathbb{N}. \quad (2.29)$$

V prvním případě jsou I_ν 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 ν je celé číslo. Vypočte se z Besselovy modifikované funkce prvního druhu dle vztahů (2.27) a (2.28), resp. (2.29). Na obr. 2.5, je vyobrazena reálná část Besselových modifikovaných funkcí pro lepší představu o jejich chování.



Obrázek 2.5: Reálná část Besselových modifikovaných funkcí (a) prvního druhu a (b) druhého druhu

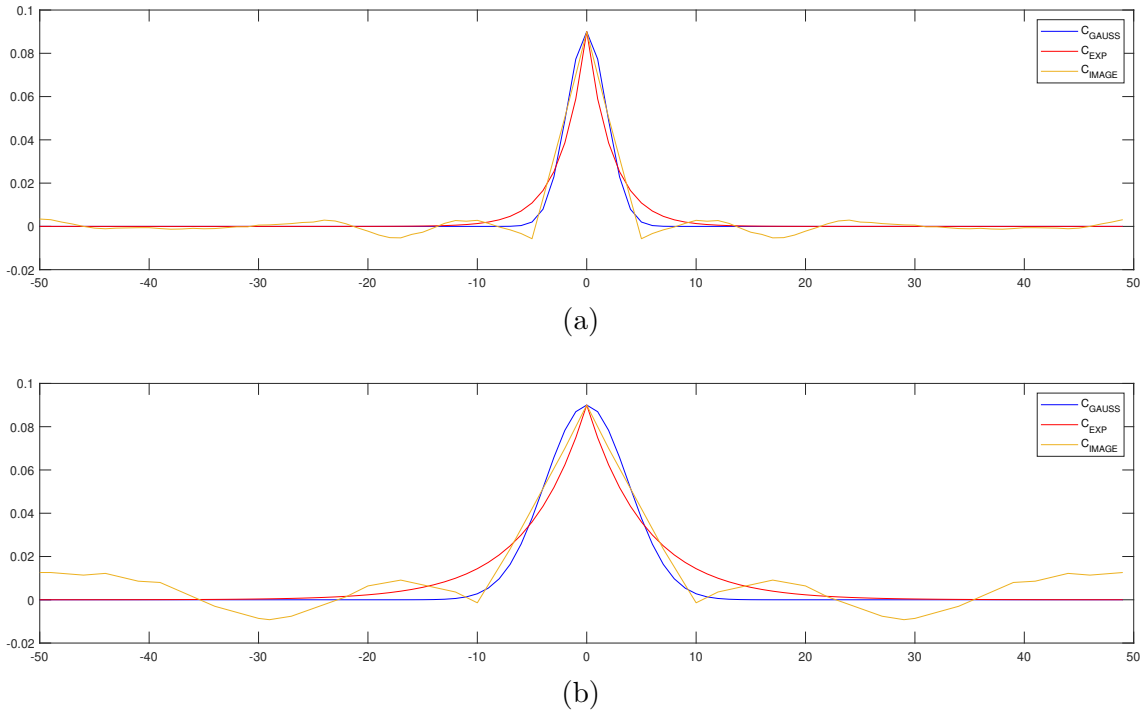
2.4.3 Numerická studie optimálních korelačních délek

V této části je provedena a okomentována numerická studie optimálních korelačních délek pro Gaussovskou a exponenciální (dle formulí (2.14) a (2.15)) kovarianční funkci, přesnost modelů a vliv na chybu pro dva základní případy. Kromě optimálních korelačních délek byla sledovaným parametrem i chyba odchylek umělých kovariančních jader vůči kovarianční funkci počítané z dvoubodové pravděpodobnosti dle rovnice (2.17).

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

Za předpokladu periodické struktury jsou 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 přináší stále stejnou informaci [4]). Vznikne tak hlavní vrchol ve středu oblasti, od kterého hodnota funkce rychle klesá a na zbytku domény se vyskytují již pouze výrazně nižší fluktuační.



Obrázek 2.6: Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze $c^{(1)} = 0, 1$ (a) inkluze 5x5 px, (b) inkluze 10x10 px

Nadále se využije sudosti Gaussovské a exponenciální funkce (viz obr 2.6), které se optimalizují již na takto přeuspořádané matici, čímž jsou tyto funkce s kovarianční funkcí výrazně lépe porovnatelné. Obě tato jádra vesměs dobře popíší 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 fluktuační mimo maximum.

Korelační délka pak udává, jak moc bude v příslušném směru protažen Gaussovský nebo exponenciální ”klobouk”. Nyní je již zřejmá důležitost správné volby korelační délky. Zvolíme-li korelační délku příliš malou, funkce dostatečně nepopíše hlavní vrchol, pokud ji ale naopak nadhodnotíme, funkce bude klesat příliš pomalu oproti realitě. Jak je patrné z obrázku 2.7, chyba s vyšší volenou korelační délkou oproti optimální roste velmi rychle. Rozhodně se tak vyplatí volit délky spíše nižší.

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

$$\varepsilon_{\text{abs}} = \sum_{i=1}^n \sum_{j=1}^m |C_{G/E} - C_I|, \quad (2.30)$$

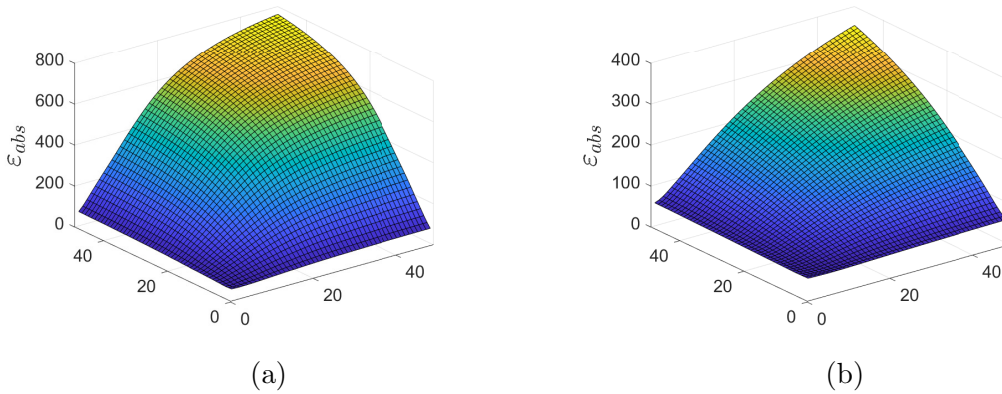
$$\varepsilon_{\text{square}} = \sum_{i=1}^n \sum_{j=1}^m (C_{G/E} - C_I)^2, \quad (2.31)$$

- kde:

m a n jsou rozměry domény,
 C_G, C_E, C_I jsou kovarianční jádra viz (2.14), (2.15), (2.17).

Optimální korelační délka je stanovena Newtonovou metodou hledající minimum této sumy čtverců chyb dle rovnice (2.31).

Jako méně náchylný na špatný odhad se ukazuje exponenciální model oproti Gaussovskému. 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).



Obrázek 2.7: Průběh absolutní chyby dle rovnice (2.30) pro Gaussovské (a) a exponenciální (b) jádro pro inkluze 10x10 - viz obr. 2.3

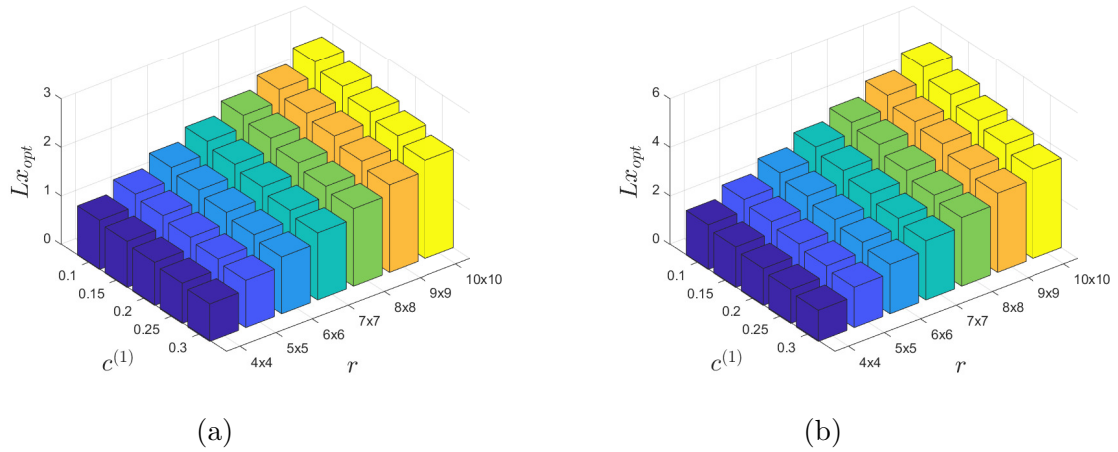
Pro potřeby samotné studie pak bylo ještě nezbytné nastavit rozptyl hodnot σ_κ^2 pro Gaussovské a exponenciální jádro a $(\kappa^{(1)} - \kappa^{(2)})^2$ pro jádro počítané z obrázku dle rovnice (2.17). Za normálních okolností by tyto rozptyly byly vypočteny na základě sledovaných fyzikálních parametrů, avšak pro potřeby této práce sledující pouze vliv korelačních délek bez vazby na konkrétní fyzikální aplikaci, byly hodnoty nastaveny následovně:

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

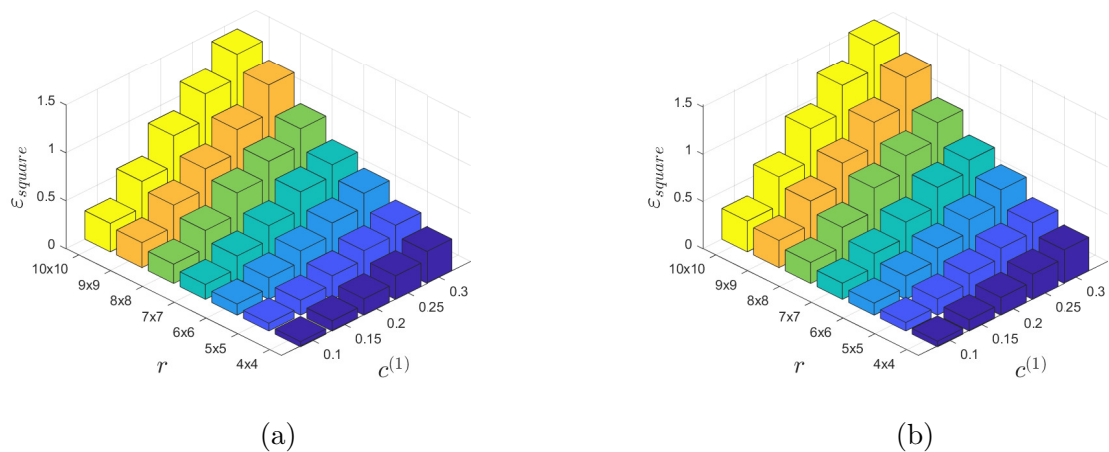
Takto se funkce shodují v počátku a mohou tak být optimalizovány. Obě studie, pro čtvercový i obdélníkový tvar inkluze, byly provedeny na 100 náhodně generovaných vzorcích, z nichž pak byl následně spočten aritmetický průměr. Rozměry čtvercových inkluzí se pohybují v rozsahu 4x4 px až 10x10 px a obdélníkových 3x4 px až 3x10 px a objemové zastoupení odstupňované po 0,05 v intervalu $\langle 0, 1; 0, 3 \rangle$ ⁵.

Z výsledků prezentovaných na obrázku 2.8 je zřejmé, že korelační délka závisí na obou sledovaných parametrech. To, že s rostoucí velikostí inkluze roste i korelační délka,

⁵POZN: V následujících grafech jsou uváděny pouze orientační objemové 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). Odchyłka je však vždy $< 4 \cdot 10^{-3}$.



Obrázek 2.8: Růst průměrné optimální korelační délky L_x v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussovské (a) a exponenciální (b) jádro

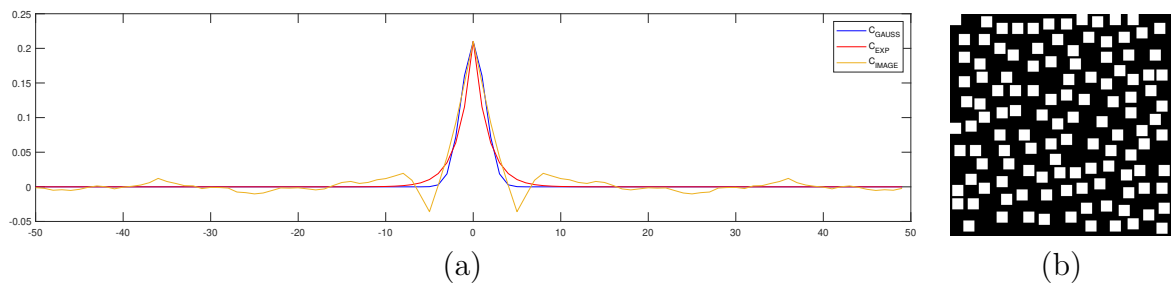


Obrázek 2.9: Růst průměrné minimální chyby (sumy čtverců chyb dle (2.31)) v závislosti na velikosti inkluze a objemovém zastoupení pro Gaussovské (a) a exponenciální (b) jádro

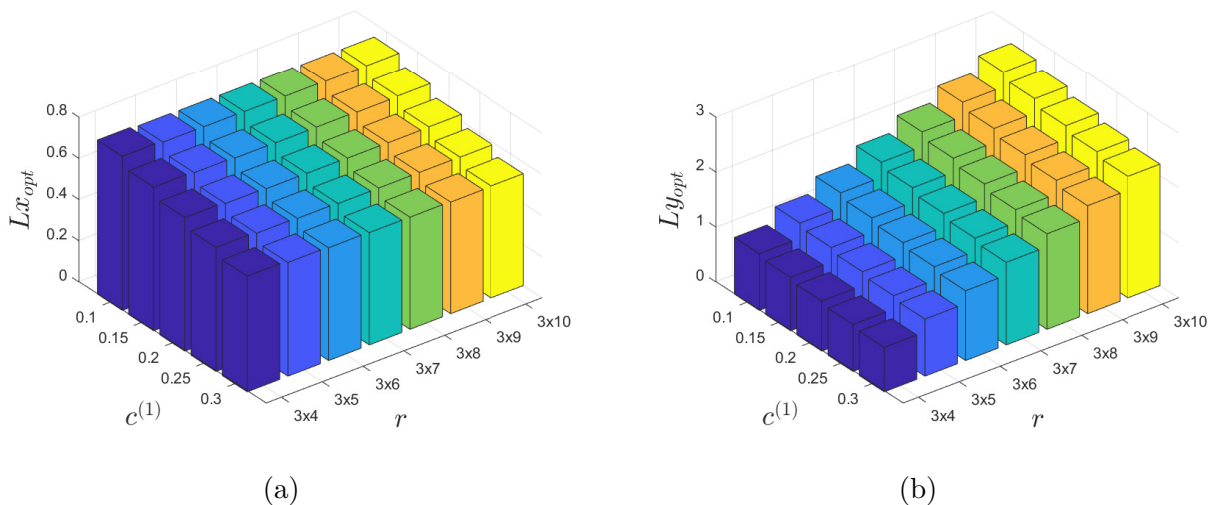
překvapivé není, neboť větší inkluze umožní i delším než jednotkovým vektorům padnout svým počátkem i koncem do stejné inkluze. Vzroste tedy hodnota jejich dvoubodové pravděpodobnosti a "klobouk" se tak rozšíří. U čtvercové inkluze se korelační délky ve směru x a y významně neliší a velmi je ovlivňuje právě rozměr inkluze. V případě obdélníkových inkluzí vychází samozřejmě větší korelační délka ve směru delšího rozměru inkluzí viz obrázek 2.11.

Druhý projevený aspekt je, že s vyšším objemovým zastoupením inkluzí 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 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 2.6 a 2.10, kde jsou vyobrazena kovarianční jádra pro vysoké objemové zastoupení.



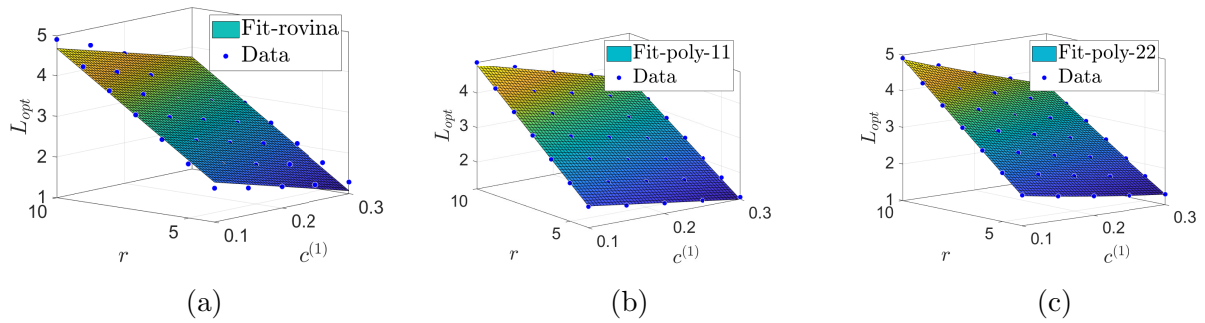
Obrázek 2.10: (a) Řez pro srovnání kovariančních jader při objemovém zastoupení inkluze $c^{(1)} = 0,3$ inkluze 5×5 px, (b) příslušná doména



Obrázek 2.11: Optimální korelační délka pro Gaussové jádro s obdélníkovou inkluzí v závislosti na její velikosti a objemovém zastoupení - (a) $L_{x,opt}$ (b) $L_{y,opt}$

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 2.9, chyba významně 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 inkluze způsobí větší fluktuace v oblasti, kde jsou umělá kovarianční jádra blízka nule a nemohou je tedy postihnout (viz obrázek 2.6 a 2.3), zatímco rostoucí objemové zastoupení navyšuje amplitudu především blízko za vrcholem, viz obrázek 2.10.

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 (viz (2.32)) není vhodnou aproximační plochou je vidět z obr. 2.12.



Obrázek 2.12: Aproximace hodnot L_{opt} pro exponenciální jádro (u Gaussovského jádra je trend podobný). Aproximace (a) rovinou (2.32), (b) hyperbolickým paraboloidem (2.33), (c) hyperbolickým paraboloidem (2.34).

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:

$$L_{opt} \approx A + B.r + C.c^{(1)}, \quad (2.32)$$

$$L_{opt} \approx B.r + C.c^{(1)} + D.r.c^{(1)}, \quad (2.33)$$

$$L_{opt} \approx B.r + C.c^{(1)} + D.r.c^{(1)} + E.r^2 + F.c^{(1)2}, \quad (2.34)$$

- kde:

L_{opt} je optimální korelační délka,

$c^{(1)}$ 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 (zde pod zkratkou MAE - *mean absolute error*) pro lepší představu.

$$\varepsilon_{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_d} (f(y_i) - f_{aprox}(y_i))^2}{n_d}}, \quad (2.35)$$

$$\varepsilon_{MAE} = \frac{\sum_{i=1}^{n_d} |f(y_i) - f_{aprox}(y_i)|}{n_d}, \quad (2.36)$$

- kde:

n_d je množství dat,
 $f(y_i)$ je původní (zjištěná) hodnota v i -tém bodě,
 $f_{aprox}(y_i)$ je aproximovaná hodnota v i -tém bodě.

Odchylka u aproximace rovinou je poměrně značná, navíc pokud bychom zakázali konstantu A , ještě významně naroste. Naopak zavedení tohoto členu do rovnic (2.33) či (2.34) se výrazně na snížení chyby neprojeví a aproximaci zbytečně komplikuje. Povšimněme si, že největší pokles chyby (cca 3-4 násobný) způsobí u Gaussovských i exponenciálních délek uvážení součinu objemového zastoupení a rozměru inkluze, tedy přechod od roviny k hyperbolickému paraboloidu.

Tabulka 2.1: Koefficienty aproximace pro Gaussovskou optimální korelační délku L pro čtvercovou inkluzi, L_x pro obdélníkovou inkluzi ve směru konstantní velikosti, L_y pro obdélníkovou inkluzi ve směru proměnné velikosti

Rce		a	B	C	D	E ($\cdot 10^{-4}$)	F	ε_{RMSE}	ε_{MAE}
(2.32)	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
(2.33)	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
(2.34)	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

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 zřejmě ani nepodaří), ale postačí rozumné okolí, v němž chyba neroste příliš rychle (viz obr. 2.7). Pro Gaussovské jádro tedy bohatě postačí vztah (2.33). Zahrnutí polynomu vyšších stupňů by stálo za úvahu u exponenciálního jádra, kde pokles chyby mezi rovnicí (2.33) a (2.34) 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 2.2: Koeficienty aproximace pro exponenciální optimální korelační délku L pro čtvercovou inkluzi, L_x pro obdélníkovou inkluzi ve směru konstantní velikosti, L_y pro obdélníkovou inkluzi ve směru proměnné velikosti

Rce		a	B	C	D	E ($\cdot 10^{-4}$)	F	ε_{RMSE}	ε_{MAE}
(2.32)	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
(2.33)	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
(2.34)	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:

$$L_{opt,G} \approx 0,3.r - 0,2.c^{(1)} - 0,25.c^{(1)}.r, \quad (2.37)$$

- pro exponenciální jádro:

$$L_{opt,e} \approx 0,55.r - 1,2.c^{(1)} - 0,5.c^{(1)}.r, \quad (2.38)$$

- kde:

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

Nutno podotknout, že odhady (2.37) a (2.38) 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.

Kapitola 3

Generování vzorků náhodného pole

Po té, co jsme získali informaci o struktuře pomocí kovariančních funkcí, přichází chvíle pro generování realizací samotného náhodného pole. V této práci budeme náhodné pole generovat pomocí Karhunenovo–Loèveho rozvoje, zkráceně KLR (*KLE - Karhunen–Loève Expansion*). V literatuře se lze setkat i s odlišnými přístupy, jako například vylepšenými perturbačními metodami či Hashin-Shtrikmanovým variačním přístupem [2]. Těmito se zde však nebudeme zabývat.

3.1 Karhunenovo–Loèveho rozvoj

KLR je vlastně nekonečný součet sloužící pro popis náhodného pole. V rovnici (3.1) je zapsán v čisté podobě pro spojitou doménu ve tvaru nekonečného součtu, rovnice (3.2) pak je pouhou úpravou pro výpočet na diskretizované doméně [2], [7], [8]:

$$\kappa(x, \omega) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} \phi_i(x) \xi_i(\omega), \quad (3.1)$$

$$\kappa(\omega) \approx \boldsymbol{\mu}_\kappa + \sum_{i=1}^{n_\phi} \sqrt{\lambda_i} \boldsymbol{\phi}_i \xi_i(\omega), \quad (3.2)$$

- kde:

$\kappa(x, \omega), \boldsymbol{\kappa}(\omega)$ značí veličinu jako funkci souřadnice x , resp. v prostoru diskretizovaný vektor, obé závislé na elementárním jevu ω ,

$\boldsymbol{\mu}_\kappa$ je vektor středních hodnot veličiny κ v diskretních bodech prostoru, resp. pro náš případ dílčích pixelech obrázku, které jsou seřazeny do jednorozměrného vektoru,

λ_i je i -té vlastní číslo kovarianční funkce, resp. matice,

$\phi_i(x), \boldsymbol{\phi}_i$ zastupuje i -tou vlastní funkci, resp. vektor, autokovarianční funkce, resp. matice,

$\xi_i(\omega)$ je náhodná proměnná (viz podkapitola 3.1.3) a

n_ϕ je počet členů uvažovaných v konečném rozvoji pro diskretizovaný výpočet.

Neboť pracujeme na diskretní doméně, tak v našem případě přejde vlastní funkce $\phi_i(x)$

na vlastní vektor ϕ_i kovarianční matice, který se dá snadno určit. Vlastní číslo nám pak hraje roli váhy, kterou vstupuje příslušná vlastní funkce do rozvoje.

3.1.1 Kovarianční matice

Před samotnou dekompozicí do vlastních vektorů ϕ_i a čísel λ_i , je potřeba přeuspořádat kovarianční jádra. Dosud jsme vždy vztahovali kovariance ke konkrétnímu bodu, avšak ze vztahů (2.14) a (2.15), 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 (2.14) a (2.15), jak naznačuje schéma (3.3), avšak kovarianční jádro zjištěné z dvoubodové pravděpodobnosti (rovnice (2.17)) nelze znovu počítat k jinému bodu, ale opět se jedná o funkci vzdálenosti. Proto budeme pro sestavení auto-kovarianční matice prvky C_I kruhově posouvat. Tedy například při posunu sloupců o γ prvků v matici, která má n sloupců přejde sloupec j -tý sloupec na pozici $(j + \gamma) \% n$, u řádků analogicky. Matici sestavíme dle stejného schématu. Následným řešením charakteristické rovnice dostaneme vlastní čísla a vypočteme vlastní vektory.

$$\tilde{C}_\alpha = \begin{pmatrix} C_{\alpha, x_{1,1}}(1, 1) & C_{\alpha, x_{1,1}}(1, 2) & \dots & C_{\alpha, x_{1,1}}(n, m) \\ C_{\alpha, x_{1,2}}(1, 1) & C_{\alpha, x_{1,2}}(1, 2) & \dots & C_{\alpha, x_{1,2}}(n, m) \\ \vdots & \vdots & \ddots & \vdots \\ C_{\alpha, x_{n,m}}(1, 1) & C_{\alpha, x_{n,m}}(1, 2) & \dots & C_{\alpha, x_{n,m}}(n, m) \end{pmatrix}, \quad (3.3)$$

- kde:

- α zastupuje indexy $G/E/I$ podle toho, zda se jedná o Gaussovskou, Exponenciální či z dvoubodové pravděpodobnosti odvozenou kovarianční matici,
- $x_{i,j}$ značí, ke kterému bodu jsou data spočtena (Gaussovská a exponenciální funkce), resp. posunuta (kovarianční funkce odvozená z S_2), viz text výše.

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 odvozeného z výřezu mikrostruktury se nám však nabízí další možnost. Máme-li k dispozici dostatečné množství vzorků/výřezů, 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 o n_{pca} vzorcích, který byl generován shodným algoritmem. Zmíněná kovarianční matice je sestavena následovně:

$$\tilde{C}_P(i, j) = \langle (\kappa_i - \mu_{\kappa_i})(\kappa_j - \mu_{\kappa_j}) \rangle, \quad (3.4)$$

- kde:

- $\langle \cdot \rangle$ značí aritmetický průměr přes soubor výřezů,

κ_i, κ_j jsou hodnoty pozorované veličiny v i -tém, resp. j -tém bodu (pixelu) a
 $\mu_{\kappa_i}, \mu_{\kappa_j}$ je střední hodnota veličiny κ_i v i -tém, resp. j -tém pixelu,

neboli v maticovém zápisu:

$$\tilde{\mathbf{C}}_P = \frac{1}{n_{\text{pca}} - 1} (\mathbf{K}\mathbf{K}^T - n_{\text{pca}}\boldsymbol{\mu}_{\kappa}\boldsymbol{\mu}_{\kappa}^T), \quad (3.5)$$

• kde:

$\boldsymbol{\mu}_{\kappa}$ je řádkový vektor středních hodnot veličiny κ v jednotlivých pixelech,
 \mathbf{K} je matice vzorků $\mathbf{K} \in \mathbb{R}^{m \cdot n \times n_{\text{pca}}}$ uspořádaných tak, že sloupce představují jednotlivé výřezy uspořádané do jednorozměrných vektorů a řádky odpovídají jednotlivým pixelům ve výřezech.

V následující části budeme porovnávat dosud zmíněné postupy získání kovarianční matice s následujícím značením:

S₂-Image: odvozením z dvoubodové pravděpodobnosti S_2 dle (2.17) získané pro jeden zvolený výřez o rozměrech $m \times n$ za předpokladu periodicity,

Gauss: vyhodnocením Gaussovské kovarianční funkce (2.14) s optimální korelační délkou stanovenou pro tentýž zvolený výřez,

Exp: vyhodnocením exponenciální kovarianční funkce (2.15) s optimální korelační délkou stanovenou pro tentýž zvolený výřez,

PCA: výpočtem kovariancí mezi fázemi v jednotlivých pixelech na souboru $n_{\text{pca}} = m \cdot n + 1$ výřezů o velikosti $m \times n$ dle (3.5), kdy je vliv periodicity zcela eliminován,

QUARTER: výpočtem dvoubodové pravděpodobnosti S_2 odvozené pro jeden zvolený výřez o rozměrech $2m \times 2n$ bez předpokladu periodicity dle (2.9),

QUARTER-CROP: výpočtem dvoubodové pravděpodobnosti S_2 odvozené pro jeden zvolený výřez o rozměrech $2m \times 2n$ za předpokladu periodicity dle (2.7) a následně oříznutím získané matice na rozměr $m \times n$.

Všechny výsledné autokovarianční matice mají shodný rozměr $(m \cdot n) \times (m \cdot n)$, jejich vlastní vektory $\boldsymbol{\phi}_i$ pak mají rozměr $m \cdot n$ a umožňují tak vytvořit náhodné pole o těchto rozměrech, ze kterého můžeme následně generovat vzorky opět o velikosti $m \times n$.

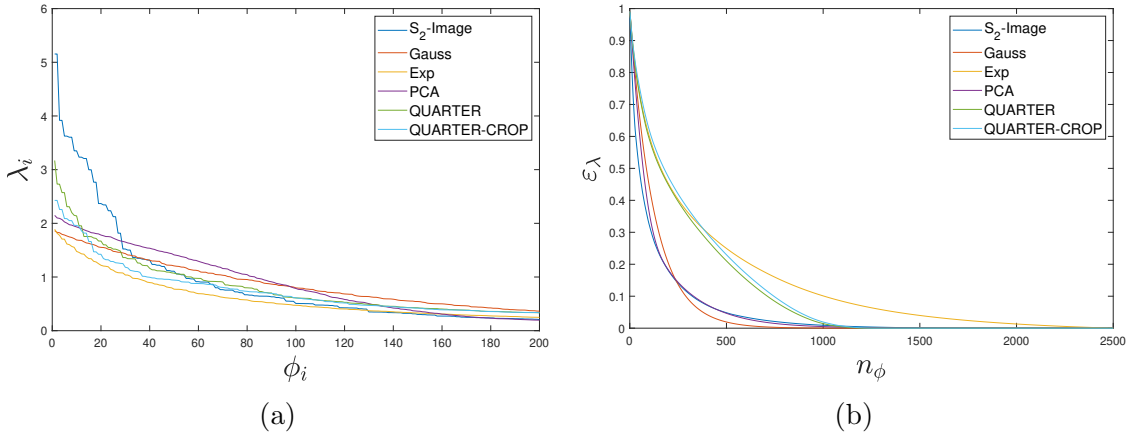
3.1.2 Vlastní čísla a vlastní vektory kovariančních matic

První vlastní tvary popisují zkoumanou doménu nejvíce, poslední pak nejméně. Váhu, jaký význam mají dílčí vlastní tvary $\boldsymbol{\phi}_i$, určují vlastní čísla λ_i . Neboť množství i délka vlastních vektorů kvadraticky roste s velikostí domény, nebudeme chtít při zpětném generování náhodného pole zahrnovat všechny vektory. Jako odhad chyby, které se dopustíme zanedbáním daného množství vlastních tvarů, použijeme:

$$\varepsilon_{\lambda} = 1 - \frac{\sum_{i=1}^{n_{\phi}} \lambda_i}{\sum_{i=1}^{n_a} \lambda_i}, \quad (3.6)$$

- kde:

λ_i je i -té vlastní číslo příslušící ϕ_i ,
 n_ϕ značí počet zahrnutých vlastních vektorů ϕ_i ,
 n_a je počet všech vlastních vektorů.



Obrázek 3.1: Konvergence vlastních čísel (a) a odhadnuté chyby (b) na doméně 50x50 px pro velikost inkluze 5x5 px, $c^{(1)} = 0.1$

Jak lze pozorovat z obr. 3.1b, 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. 3.1a. 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 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. 3.2. Vlastní čísla a konvergence chyby se také liší v případech, kdy je určujeme pomocí PCA.

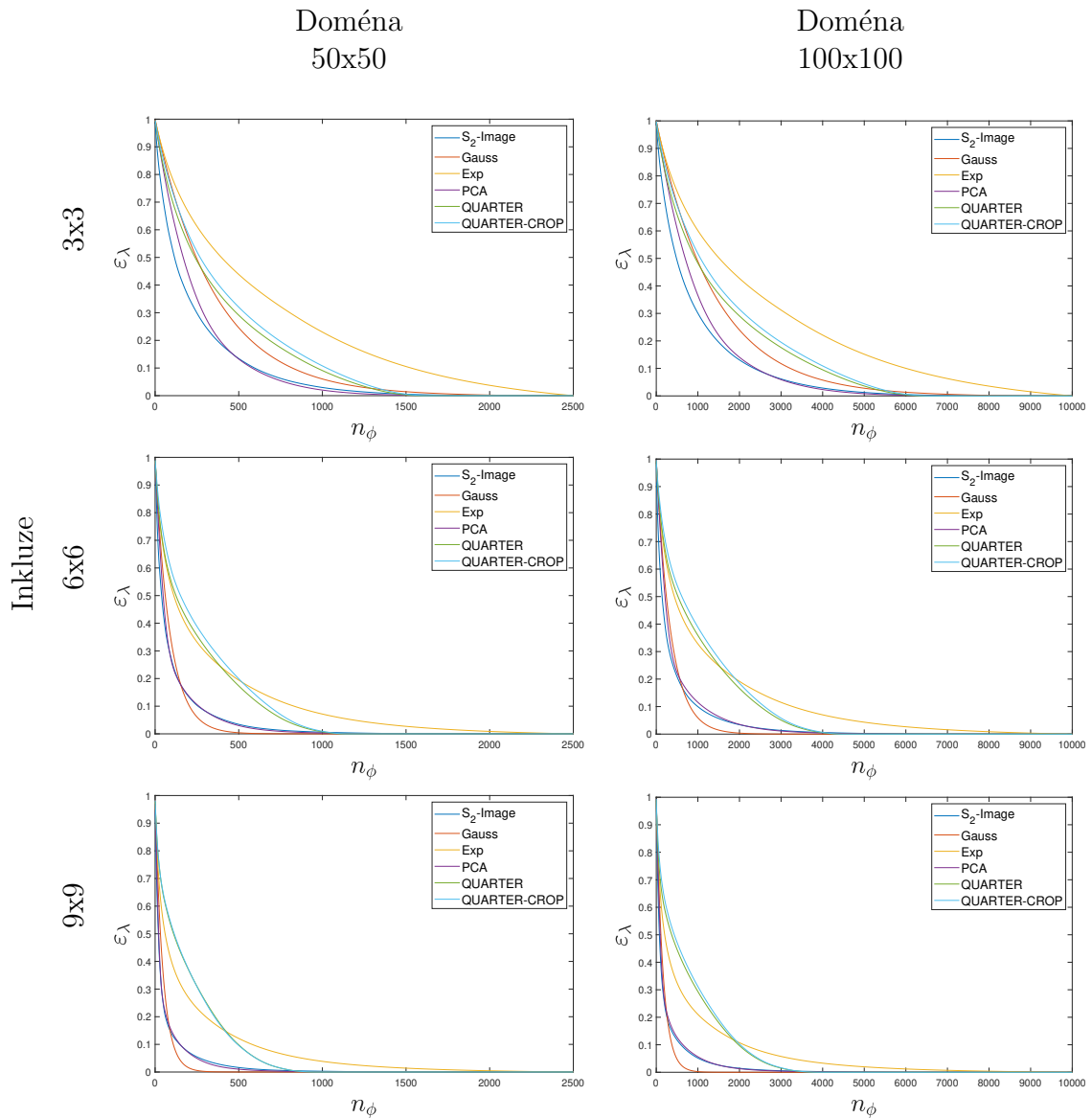
U neperiodických variant odvozených z S_2 (QUARTER a QUARTER-CROP) není kovarianční matice pozitivně semidefinitní. Záporná vlastní čísla jsou oříznuta, neboť v KLR pracujeme s odmocninou z vlastního čísla, dostávali bychom tak komplexní část řešení, která pro nás nemá význam. Tyto poslední vlastní tvary nerepresentují modelované médium. Podobný problém nastává i u ostatních kovariančních matic, kde se však jedná o velmi málo vlastních čísel, jejichž velikost je navíc velmi malá a jsou důsledkem numerického výpočtu vlastních čísel v prostředí MATLAB.

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

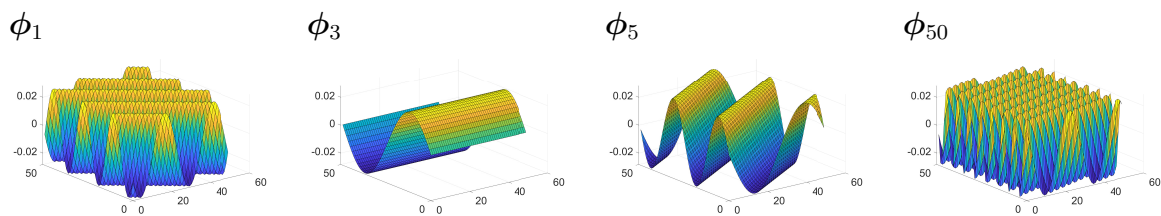
Několik vlastních tvarů pro exponenciální kovarianční jádro je vyobrazeno na obr. 3.3b. Tyto funkce jsou však známé z literatury a především jsou nezávislé na studované doméně.

(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.)

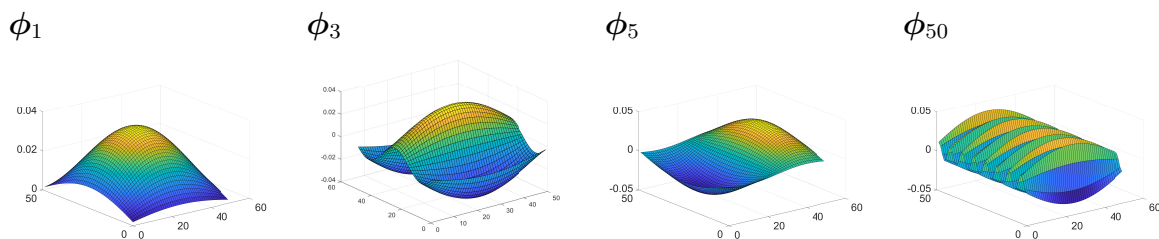
Naopak vlastní tvary odvozené z PCA (viz obr. 3.3c) již závisí na studované doméně, stejně jako vlastní tvary odvozené z dvoubodové pravděpodobnosti (viz obr. 3.3a).



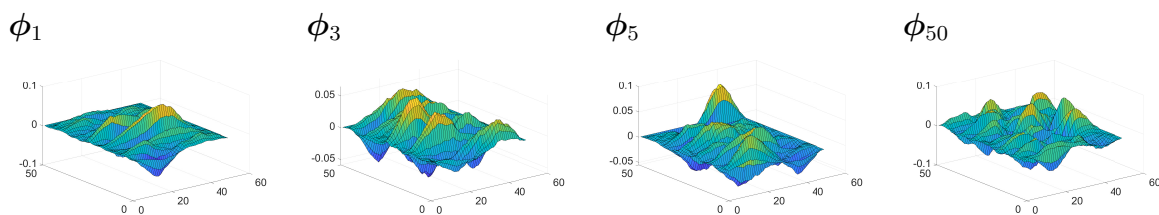
Obrázek 3.2: Porovnání konvergence odhadu chyby v závislosti na velikosti inkluze pro $\lambda_i, c^{(1)} = 0.1$



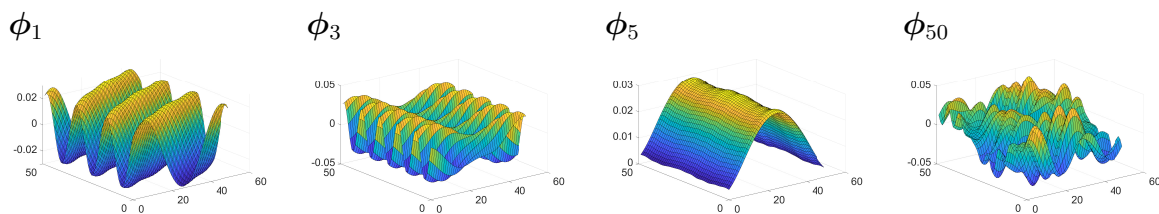
(a) S_2 -Image - S_2 z jednoho vzorku



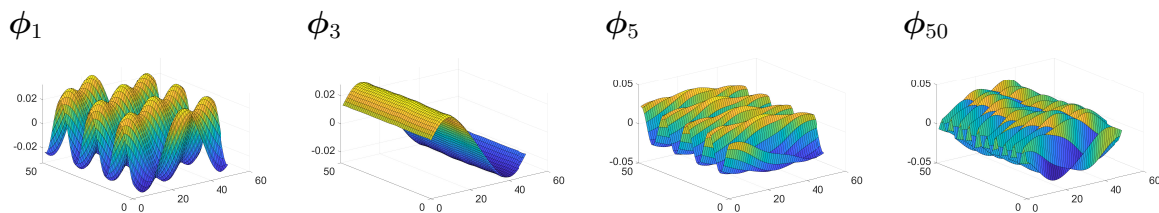
(b) Exponenciální, u Gaussovské jsou velmi podobné



(c) PCA - ze souboru $n_{pca} = 2\,501$ vzorků



(d) QUARTER - Neperiodická S_2 ze čtyřnásobného vzorku



(e) QUARTER-CROP - Periodická oříznutá S_2 ze čtyřnásobného vzorku

Obrázek 3.3: Vlastní tvary zjištěné pro doménu 50x50 px, inkluze 3x12 px, $c^{(1)} = 0.1$

3.1.3 Volba náhodných proměnných $\xi_i(\omega)$ v KLR

Nejproblematictější částí KLR je právě vypořádání se s náhodnými proměnnými $\xi_i(\omega)$. Jejich pravděpodobnostní rozdělení není známé, proto je běžně volen přístup, kdy jsou náhodné proměnné vybírány z nezávislých standardních normálních rozdělení. Navíc v našem případě, kdy se snažíme charakterizovat binární náhodné pole, tímto obdržíme vzorky reálného náhodného pole. Toto je vyřešeno pomocí prahu, který rozdělí reálné hodnoty na binární (0,1) tak, aby odpovídalo objemové zastoupení inkluze.

Pokusíme se nyní několika způsoby získat soubor vzorků proměnných $\xi_i(\omega)$ z výřezu, který máme k dispozici. Uvažujme dva případy. Prvně mějme k dispozici větší množství vzorků studovaného média, podruhé pouze jediný obrázek, který však považujeme za reprezentativní.

Máme-li soubor vzorků, uspořádáme je do matice \mathbf{K} tak, že jednotlivé výřezy seřadíme do jednorozměrných vektorů, které umístíme do sloupců matice \mathbf{K} , řádky pak odpovídají jednotlivých pixelům výřezu. Ve variantě PCA můžeme analýzou hlavních komponent z matice \mathbf{K} určit přímo vlastní tvary a čísla. Odpovídající realizace náhodných proměnných $\xi_i(\omega)$ pak získáme rotací souřadného systému do směrů hlavních komponent dle:

$$\Xi = \Phi_\lambda^{-1}(\mathbf{K} - \mathbf{M}_\kappa), \quad (3.7)$$

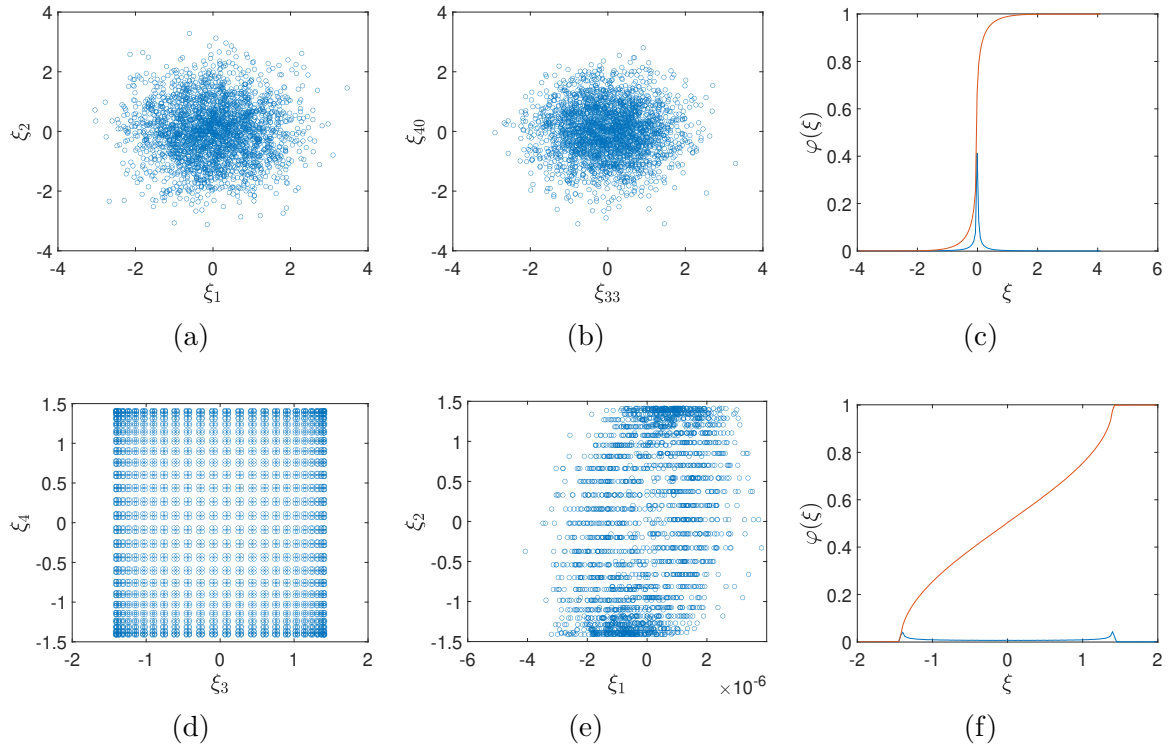
- kde:

Φ_λ je matice vlastních vektorů škálovaných vlastními čísly λ_i , $\Phi_\lambda \in \mathbb{R}^{m \cdot n \times m \cdot n}$,
 \mathbf{K} je matice realizací obrázků $\mathbf{K} \in \mathbb{R}^{m \cdot n \times n_{\text{pca}}}$,
 \mathbf{M}_κ představuje matici s identickými sloupci tvořenými sloupcovým vektorem $\boldsymbol{\mu}_\kappa^T$ středních hodnot veličiny κ v jednotlivých pixelech výřezu¹, $\mathbf{M}_\kappa \in \mathbb{R}^{m \cdot n \times n_{\text{pca}}}$.

V případě, kdy máme pouze jeden vzorek (výřez) média (ať už o velikost $m \times n$ nebo $2m \times 2n$), vytvoříme si soubor vzorků uměle tak, že si obrázek za předpokladu periodicity rozkopírujeme čtyřikrát (dva vedle sebe a dva pod sebou) a nové vzorky získáme jako výřezy posunutím postupně o jeden pixel v obou směrech. Vzorků tedy získáme právě tolik, kolik pixelů obrázek má. Vzniklé vzorky pak už opět uspořádáme do matice \mathbf{K} výše popsaným způsobem a vzorky Ξ získáme dle rovnice (3.7) s tím rozdílem, že matici vlastních tvarů zde odvodíme z kovarianční matice získané libovolným postupem popsaným v kapitole 3.1.1.

Ze získaných vzorků Ξ si následně vytvoříme histogramy jednotlivých proměnných $\xi(\omega)$, z nichž posléze vypočteme aproximaci distribuční funkce $\varphi(\xi)$. Ty jsou společně s několika soubory Ξ vykresleny na obr. 3.4). V případě souboru získaného Ξ ze souboru nezávislých výřezů se i výsledné realizace proměnných $\xi_i(\omega)$ zdají být nezávislé, na rozdíl od realizací získaných posouváním pouze jednoho obrázku.

¹V našem případě jsou všechny prvky vektoru $\boldsymbol{\mu}_\kappa$ rovny objemovému zastoupení inkluze $c^{(1)}$.



Obrázek 3.4: Vykreslení realizací v průmětu do roviny vybraných ξ_i a distribuční funkce získané pro (a,b,c) soubor vzorků (PCA), (d,e,f) posouvaný obrázek (SHIFT), u (c,f) oranžová barva značí distribuční funkci $\varphi(\xi)$, modrá příslušný histogram

3.1.4 Hodnocení modelů pro generování náhodného pole

Nyní je potřeba posoudit kvalitu vytvořeného 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)². 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:

$$\varepsilon_1 = \frac{1}{q} \sum_{k=1}^q \sum_{i=1}^n \sum_{j=1}^m |C_{I,new}(i,j) - C_{I,org}(i,j)|, \quad (3.8)$$

• kde:

m a n jsou rozměry domény,

i, j značí i, j -tý prvek,

q je počet nově vygenerovaných náhodných realizací,

$C_{I,org}$ je původní kovarianční matice (vzorku) vypočtená dle (2.17),

$C_{I,new}$ je nová kovarianční matice (vzorku) vypočtené dle (2.17).

²I tento požadavek by bylo možné kvantifikovat, prozatím jej však budeme vyhodnocovat "opticky".

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:

$$\varepsilon_q = \sum_{i=1}^n \sum_{j=1}^m |\tilde{C}_{P,new}(i, j) - \tilde{C}_{P,org}(i, j)|, \quad (3.9)$$

• kde:

m a n jsou rozměry domény,
 i, j značí i, j -tý prvek,
 $\tilde{C}_{P,org/new}$ je kovarianční matice původního, resp. nového, souboru vypočtená dle (3.5).

Hodnota této chyby samozřejmě bude záviset na velikosti nově vygenerovaného souboru, avšak pro porovnání postupů generování náhodného pole absolutní hodnotu nemusíme řešit. Porovnávané varianty jsou přehledně shrnuty v tabulce tab. 3.1.

Tabulka 3.1: Přehled variant zpětného modelování náhodného pole

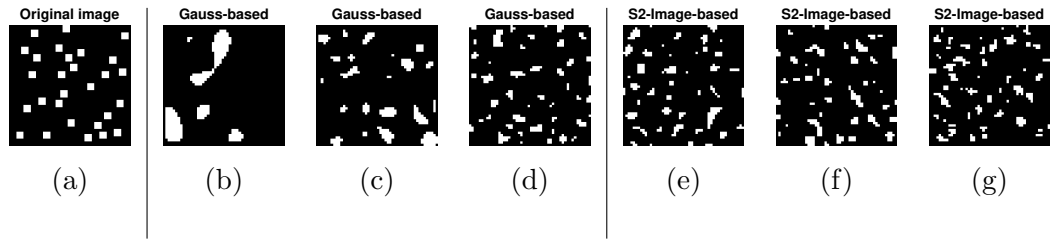
Označení	Vstupní data	Kovarianční jádro	Rozdělení ξ
S_2 -Image	S_2 z 1 obr.	S_2	Gauss
Gauss	L_{opt} z 1 obr.	Gauss	Gauss
Exp	L_{opt} z 1 obr.	Exp	Gauss
PCA-G	K_{org} z víc obr.	neperiodická K_{org}	Gauss
PCA-H	K_{org} z více obr. + hist ξ	neperiodická K_{org}	hist ξ
SHIFT-H	S_2 z 1 obr. + hist ξ	S_2	hist ξ
SHIFT-U	S_2 z 1 obr. + hist ξ	S_2	rovnoměrná ξ
QUARTER-CROP-G	S_2 z 4x většího obr.	S_2 , oříznuta	Gauss
QUARTER-CROP-H	S_2 z 4x většího obr. + hist ξ	S_2 , oříznuta	hist ξ
QUARTER-G	S_2 z 4x většího obr.	neperiodická S_2	Gauss
QUARTER-H	S_2 z 4x většího obr. + hist ξ	neperiodická S_2	hist ξ

”Konvergenci” průměrné chyby v závislosti na počtu zahrnutých vlastních vektorů prezentují obrázky 3.9 a 3.10, v detailu pak 3.11 a 3.12. Výsledky na nich jsou v souladu s (3.8) a (3.9) pro množství generovaných vzorků $q = 100$. Množství vzorků referenčního řešení souboru je 25 000, z toho 2 501 bylo zpřístupněno pro výpočty pomocí PCA.

Prvně je dobré si povšimnout, že tato konvergence se liší od odhadu přesnosti pomocí vlastních čísel z rovnice (3.6). 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. 3.5. Neb chyba konverguje poměrně rychle, není rozhodně potřeba do výpočtu zahrnovat velké množství vlastních vektorů ϕ_i . Jak je nejlépe patrné z obr. 3.12a, v některých případech to ani nemusí být vhodné.

Z grafů (3.9) a (3.11) je dále vidět, že vypočtená ξ_i za předpokladu periodického obrázku (SHIFT-H) lze v našem případě dobře aproximovat rovnoměrným rozdělením. Veškeré varianty zjištěné z obrázku si vedou lépe při porovnání s referenčním obrázkem, hůře pak proti souboru ((3.10) a (3.12)).

Ve variantě PCA chyba za svým lokálním minimem roste s vyšším počtem zahrnutých vlastních vektorů ϕ_i , obdobný trend vykazují i ostatní křivky. Z obr. 3.12 a obr. 3.11 by



Obrázek 3.5: Porovnání generovaných obrázků 50x50 px, pro čtvercové inkluze, (a) originální obr. (b) Gaussovské jádro 50 n_ϕ (c) Gaussovské jádro 150 n_ϕ , (d) Gaussovské jádro 500 n_ϕ , (e) jádro z S_2 50 n_ϕ , (f) jádro z S_2 150 n_ϕ , (g) jádro z S_2 500 n_ϕ



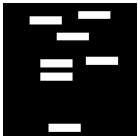
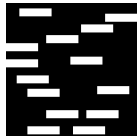



























































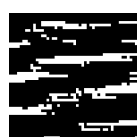






Obrázek 3.6: Porovnání generovaných obrázků 50x50 px, pro obdélníkové inkluze 3x12 px (a) originální obrázek (b) exp. jádro, 40 n_ϕ , (c) exp. jádro, 500 n_ϕ , (d) varianta SHIFT-H, 40 n_ϕ , (e) varianta SHIFT-H, 500 n_ϕ , (f) varianta PCA, 40 n_ϕ , (g) varianta PCA, 250 n_ϕ , (h) varianta PCA, 500 n_ϕ

se mohlo zdát, že PCA je nejlepší metodou popisu média, ale jak prezentuje obr. 3.6, zatímco ostatní varianty se při zahrnutí většího množství ϕ_i blíží "kvalitnějšímu" řešení nebo si alespoň zachovávají správnou (sic velmi hrubou) informaci o anizotropii materiálu, popř. velikosti inkluzí), PCA nikoliv. Toto je patrné z přehledu generovaných polí pro obdelníkový tvar inkluze na obr. 3.7 a jak si lze z přehledu prezentující náhodná pole pro čtvercové inkluze (obr. 3.8), PCA rychle ztrácí i informaci o velikosti inkluze.

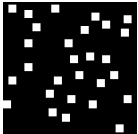
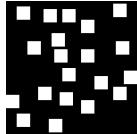


































































Velmi slušné výsledky vykazují metody QUARTER a QUARTER-CROP, tedy se snahou o potlačení periodicity. Zdá se tedy, že uvažování neperiodické domény skutečně může ovlivnit výsledky. Metoda QUARTER-CROP se zahrnutými histogramy dokonce popisuje soubor vzorků při zahrnutí většího množství tvarů nejlépe a pro nižší počty se přesností blíží variantě PCA. Jejich chyba sice konverguje pomaleji, ale hodí se pro popis jak jediného vzorku (viz obr. 3.11), tak souboru, čehož si lze povšimnout na obr. 3.10

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

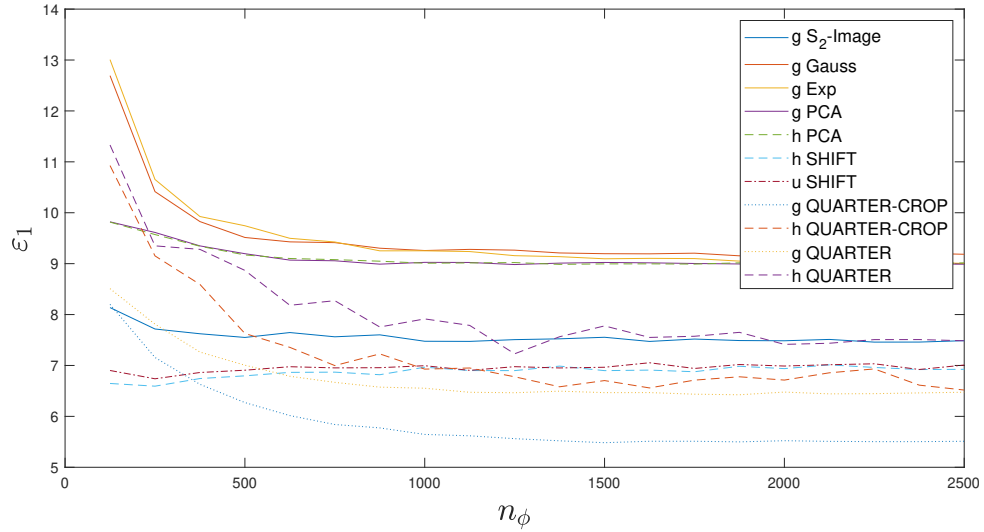
Z těchto poznatků plyne, že chyba změřená pouze pomocí rozdílu kovariancí může pomoci pro utvoření představy o kvalitě pole, ale sama o sobě je nedostatečný ukazatel. Nabízí se tedy hodnotit pole jinak, například pomocí odezvy jednoduchého a přísného fyzikálního (např. mechanického) modelu, což je želbohu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

Originální obrázek		Doména 50x50 $r = 3 \times 12px$ $c^{(1)} = 0, 1$				Doména 50x50 $r = 3 \times 12px$ $c^{(1)} = 0, 2$	
	$n_\phi = 50$	$n_\phi = 250$	$n_\phi = 500$	$n_\phi = 50$	$n_\phi = 250$	$n_\phi = 500$	
S_2 -Image							
Gauss							
Exp							
PCA-G							
PCA-H							
SHIFT-H							
SHIFT-U							
QUARTER- -CROP-H							
QUARTER- -CROP-G							
QUARTER-H							
QUARTER-G							

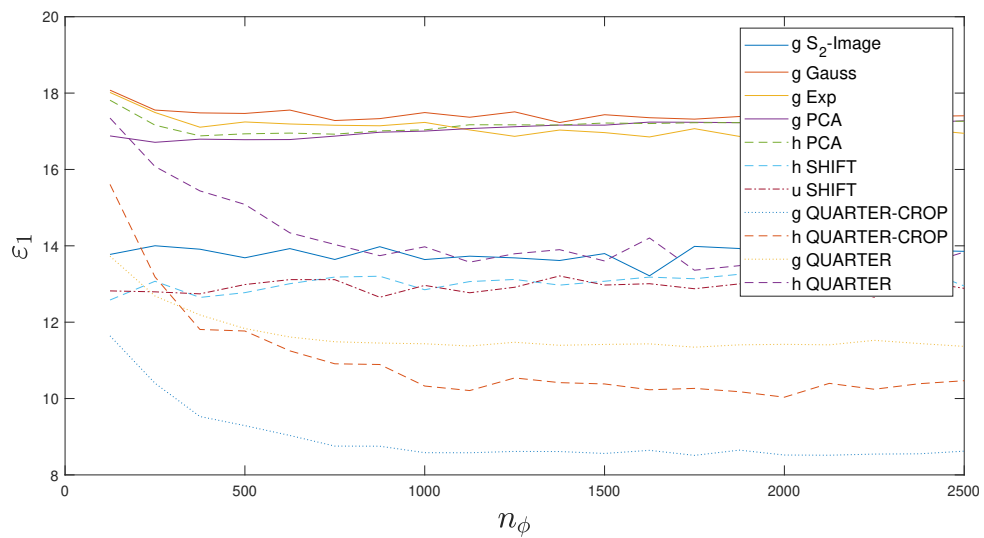
Obrázek 3.7: Přehled vzorků náhodného pole pro dílčí varianty a obdélníkový tvar inkluze

Originální obrázek		Doména 50x50 $r = 3 \times 3px$ $c^{(1)} = 0, 1$				Doména 50x50 $r = 5 \times 5px$ $c^{(1)} = 0, 2$	
	$n_\phi = 50$	$n_\phi = 250$	$n_\phi = 500$	$n_\phi = 50$	$n_\phi = 250$	$n_\phi = 500$	
S_2 -Image							
Gauss							
Exp							
PCA-G							
PCA-H							
SHIFT-H							
SHIFT-U							
QUARTER- -CROP-H							
QUARTER- -CROP-G							
QUARTER-H							
QUARTER-G							

Obrázek 3.8: Přehled vzorků náhodného pole pro dílčí varianty a čtvercový tvar inkluze

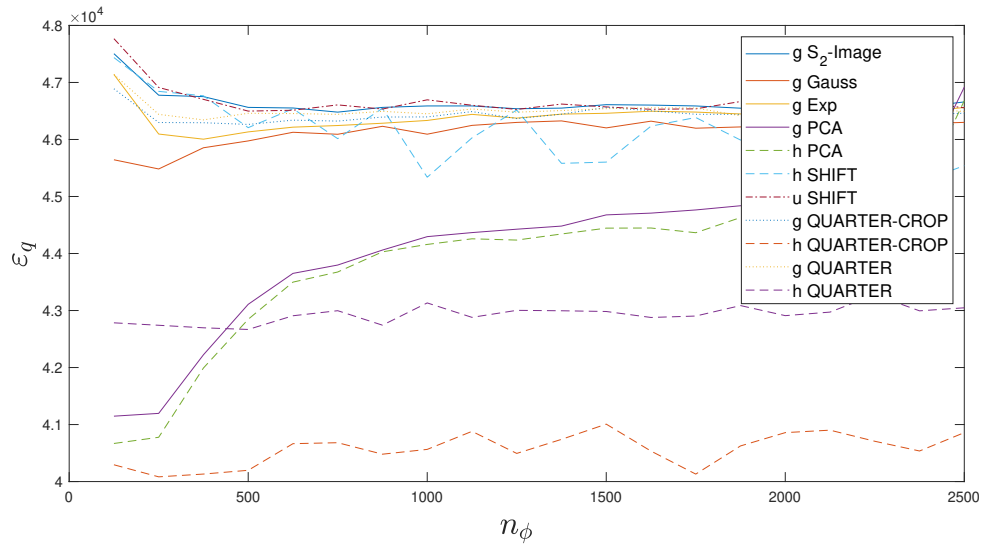


(a)

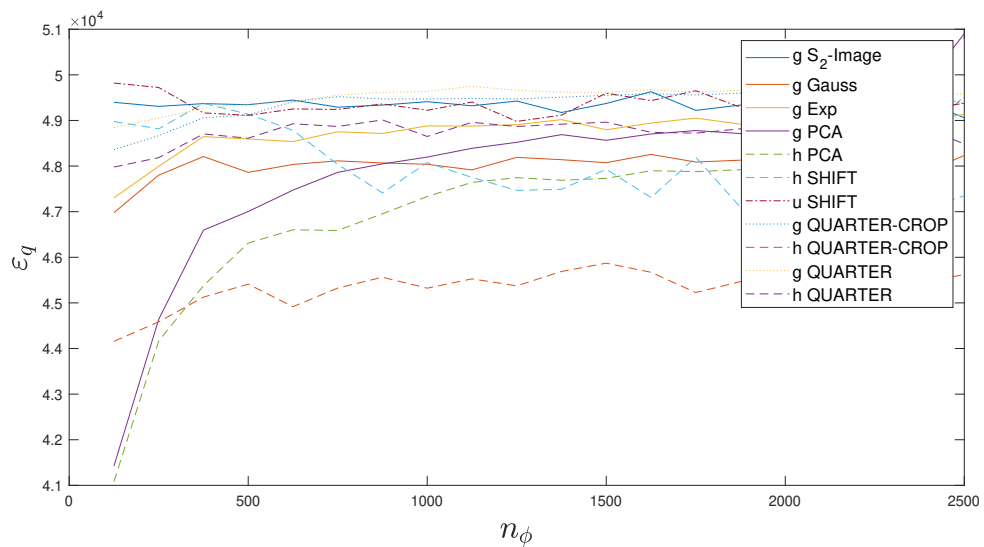


(b)

Obrázek 3.9: Konvergence chyby při porovnání vůči vzorku pro $c^{(1)} = 0,1$ na doméně 50×50 px, velikost inkluze (a) 3×3 px (b) 3×12 px

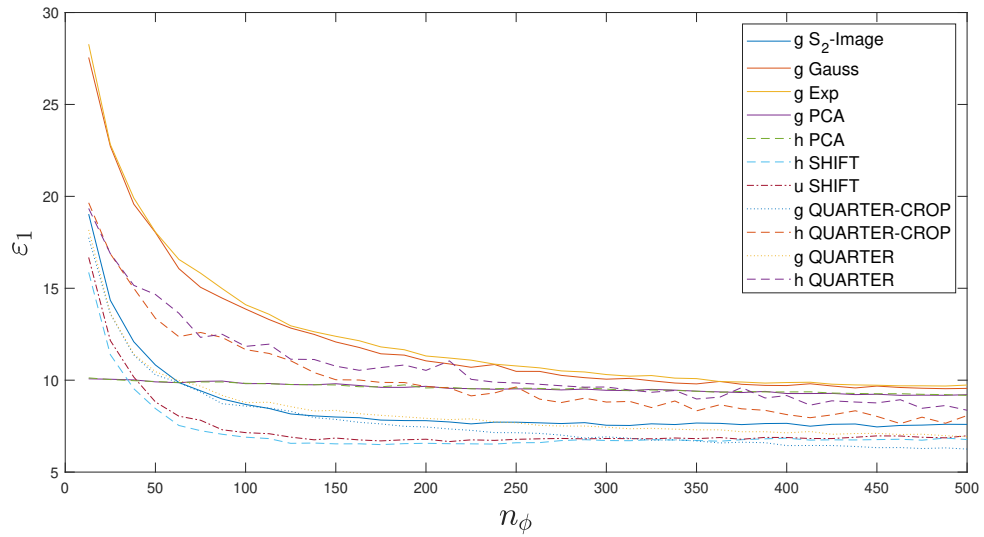


(a)

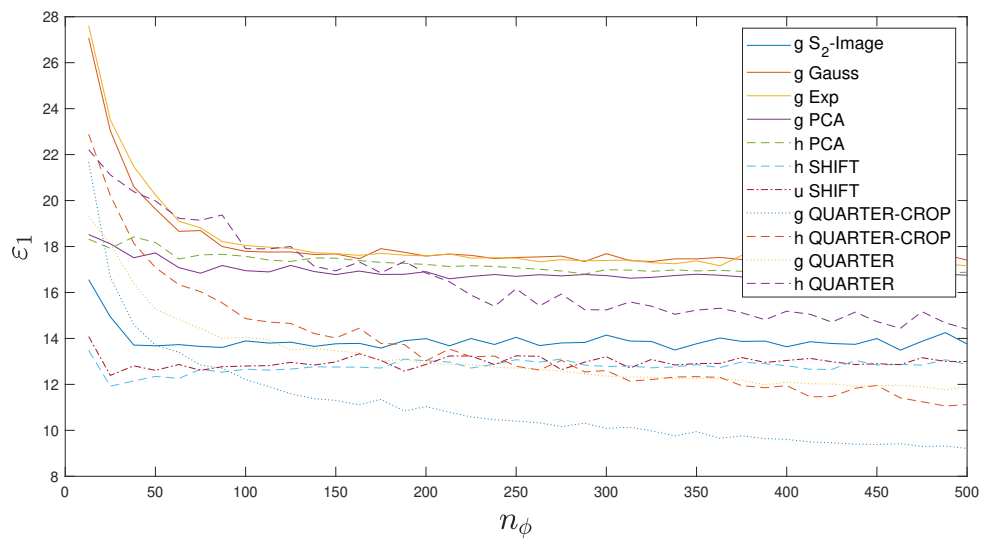


(b)

Obrázek 3.10: Konvergence chyby při porovnání vůči souboru pro $c^{(1)} = 0, 1$ na doméně 50×50 px, velikost inkluze (a) 3×3 px (b) 3×12 px

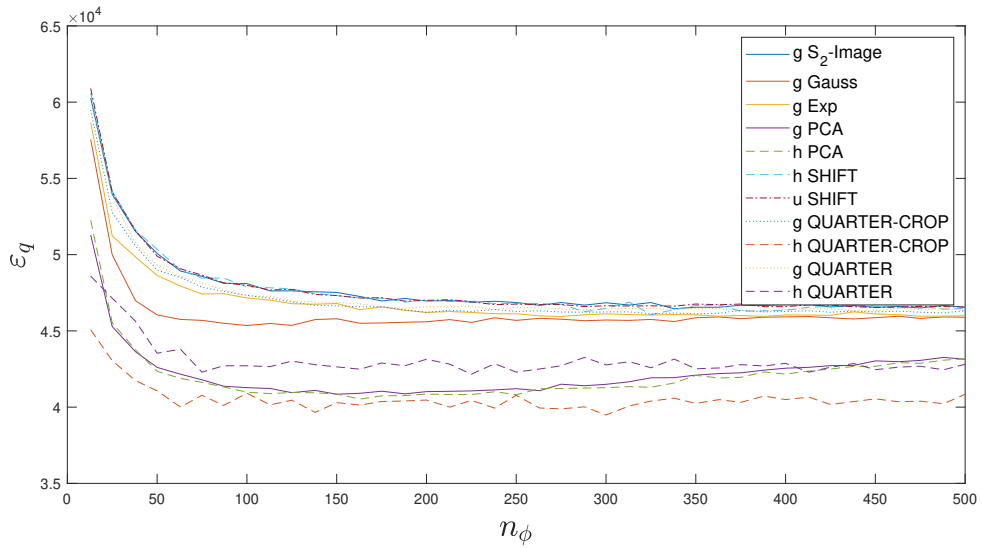


(a)

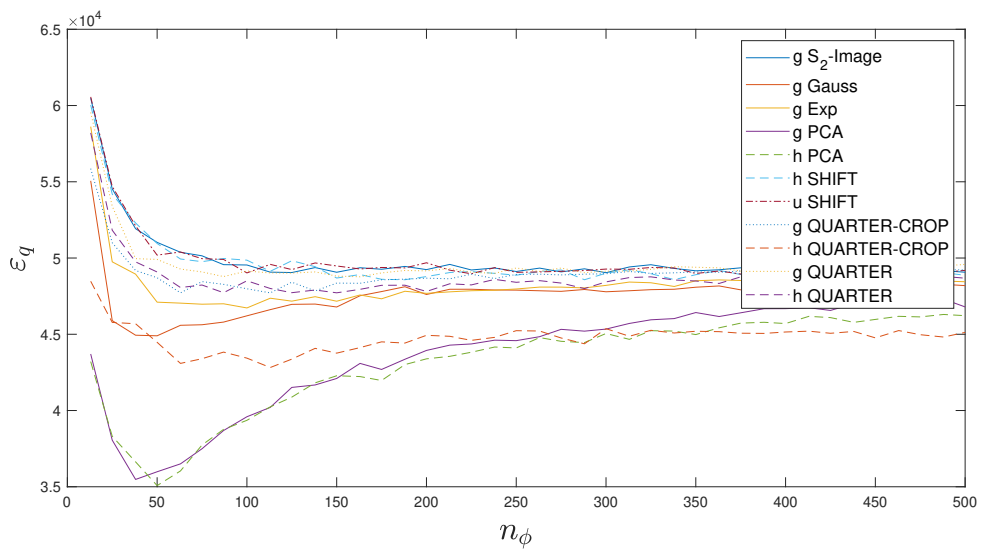


(b)

Obrázek 3.11: Detail konvergence chyby při porovnání vůči vzorku pro $c^{(1)} = 0,1$ na doméně 50×50 px, velikost inkluze (a) 3×3 px (b) 3×12 px



(a)



(b)

Obrázek 3.12: Detail konvergence chyby při porovnání vůči souboru pro $c^{(1)} = 0,1$ na doméně 50×50 px, velikost inkluze (a) 3×3 px (b) 3×12 px

Kapitola 4

Závěr

V rámci této práce je předvedeno modelování 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 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 pro náš případ 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. V případě, že si však předem určíme průběh konvergence chyby a podle ní bychom zvolili množství vlastních vektorů, tato metoda může dobře charakterizovat právě soubor více vzorků. Její nespornou nevýhodou je však potřeba velkého množství vstupních dat.

Příjemnou alternativou k PCA, která pro nižší množství tvarů vykazuje jedny z nejlepších výsledků, se v tomto ohledu jeví varianty výpočtu kovariančních funkcí na čtyřnásobném výřezu, než který se snažíme reprezentovat. Toto sice stále znamená potřebu více vstupních dat, ale výrazně menšího množství než u PCA. Obdržené výsledky jsou srovnatelné, nebo alespoň stále lepší než při použití kupříkladu kovariančních funkcí z literatury.

Zdá se, že náhodné proměnné ξ_i v KLR, nejsou nezávislé a zanedbání tohoto aspektu může zhoršení výsledků. Zvláště pokud se zahrne větší množství vlastních vektorů ve snaze o vyšší přesnost.

Metoda založená na analýze 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 není zcela uspokojivý 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 algoritmy¹ 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. Zako-mentovaná časová podmínka byla užitá v rámci numerické studie, neboť u vyšších obje-mových 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
```

¹Uvedeny 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_inkluz
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 ZAPLNE NI 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] = TwoPointProbabilty(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 2.4.3) nepoužitelný. Proto zde byl použit kód pracující s diskretní Fourierovou transformací dle rovnice 2.8.

```

1     function [S2] = TwoPointProbability_FT(OBR)
2     %TWOPOINTPROBABILITY_FT
3     % Vypocte 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 ξ , 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 (treshold) 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.4 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 OBR2=RandomPictureGenerator(V_inkluze, DIST, x_ink, y_ink, 2*nROW, ...
    2*nCOLUMN);
24
25 %% DVOUBODOVA PRAVDEPODOBNST (bro bilou fazi 1)
26 %Vypocet pomoci Fourierovy Transformace
27 DP=TwoPointProbability_FT(OBR);
28 DP1=TwoPointProbability_FT(OBR2);
29 DP1=DP1(1:nROW,1:nCOLUMN);
30 %VYPOCET Z DEFINICE
31 DP2=TwoPointProbability_quarter(OBR2); %neperiodicka
32 % DP=TwoPointProbability(OBR);
33
34
35 %% KOVARIACNI JaDRA A HODNOCENI JEJICH KVALITY
36
37 % SMERODATNa ODCHYLKA A ROZPTYL
38 SIG=(DP(1,1)-DP(1,1)^2)^0.5; SIG_2=SIG^2;
39 lambda_1=0; lambda_2=1; DELTA_lam=lambda_1-lambda_2;
40
41 % JaDRO ZJISTENE Z DVOUBODOVE PRAVDEPODOBNOSTI
42 C_dp=Cf_image(DP, DELTA_lam);
43 C_dp1=Cf_image(DP1, DELTA_lam);
44 C_dp2=Cf_image(DP2, DELTA_lam);
45
46 %% OTIMALIZACE Lx a Ly
47 % % L=[Lx, Ly]
48 % Pocatecni bod
49 x0=[1 1];
50

```

```

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

```



```

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

```

```

165     %CHYBA GAUSSOVSKÉHO JaDRA
166     NCHL_gauss(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
167     =fmin_gN([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);
168     NCHL_gauss_abs(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
169     =fmin_gN_abs([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);
170     %CHYBA EXPONENCIÁLNÍHO JaDRA
171     NCHL_exp(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
172     =fmin_eN([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);
173     NCHL_exp_abs(i+(-ind_R+nROW)+1,j+(-ind_C+nCOLUMN)+1)=
174     =fmin_eN_abs([NxLX(i+(-ind_R+nROW)+1),NyLY(j+(-ind_C+nCOLUMN)+1)]);
175     end
176 end
177
178 %% PREUSPORADANI KOVARIANČNI MATICE PRO KRL
179 C_gauss=Cf_gauss(DP,N_L_gauss_opt, SIG_2);
180 C_exp=Cf_exp(DP,N_L_exp_opt, SIG_2);
181
182 %KOVARIANČNI MATICE VZNIKLE POSUMEN
183 RC_gauss=PointCovMatrix('gauss',N_L_gauss_opt, DP, SIG_2);
184 RC_exp=PointCovMatrix('exp',N_L_exp_opt, DP, SIG_2);
185 RC_image=PointCovMatrix('image',[], C_dp, DELTA_lam);
186 RC_image_q1=PointCovMatrix('image',[], C_dp1, DELTA_lam);
187 RC_image_q2=PointCovMatrix('image',[], C_dp2, DELTA_lam);
188 %REALIZACE KAPPA PRO PCA
189 SC_dp=PointSamMatrix(DP(1,1),x_ink,y_ink,'image',N_L_exp_opt, OBR, ...
190     DELTA_lam);
191
192 %% ROZKLAD DO VLASTNÍCH VEKTORŮ
193 % close all;
194
195 % MET 1 - ROZKLAD KOVAR MCE DO VL. TVARU
196 [Eig_V_image,Eig_D_image]=eig(RC_image);
197 [Eig_V_gauss,Eig_D_gauss]=eig(RC_gauss);
198 [Eig_V_exp,Eig_D_exp]=eig(RC_exp);
199 [Eig_V_image_q1,Eig_D_image_q1]=eig(RC_image_q1);
200 [Eig_V_image_q2,Eig_D_image_q2]=eig(RC_image_q2);
201
202 % MET 2 - PCA
203 [Eig_V_image_pca,Xi_image_pca,Eig_D_image_pca]=
204 =pca(SC_dp(:,1:1.1*nROW*nROW));
205
206 C_ensemble=cov(SC_dp');
207 size(C_ensemble)
208
209 Xi_image_pca=Xi_image_pca';
210
211 % MET 3
212 [Eig_V_image_shift,Eig_D_image_shift]=eig(RC_image);
213
214 %% SERAZENÍ VL. CÍSEL A TVARU
215 Eig_D_image=diag(Eig_D_image);
216 Eig_D_image_q=diag(Eig_D_image_q);
217 Eig_D_gauss=diag(Eig_D_gauss);
218 Eig_D_exp=diag(Eig_D_exp);
219 Eig_D_image_shift=diag(Eig_D_image_shift);
220 Eig_D_gauss_shift=diag(Eig_D_gauss_shift);
221 Eig_D_exp_shift=diag(Eig_D_exp_shift);
222 Eig_D_image_q1=semidefit(Eig_D_image_q1);

```

```

222 Eig_D_image_q2=semidefit(Eig_D_image_q2);
223
224 %preusporadani do vzestupneho poradi
225 Eig_V_image_pca= SortEigVec(Eig_V_image_pca);
226 Eig_D_image_pca=sort(Eig_D_image_pca, 'ascend');
227
228 disp('Zornhaw')
229
230 %% XI POMOCI KLR (KARHUNEN-LOEVE ROZVOJ)
231 % MET 1
232 Xi_image=FindXi(DP(1,1),x_ink,y_ink, nROW, nCOLUMN, Eig_D_image, ...
    Eig_V_image);
233 Xi_gauss=FindXi(DP(1,1),x_ink,y_ink, nROW, nCOLUMN, Eig_D_gauss, ...
    Eig_V_gauss);
234 Xi_exp=FindXi(DP(1,1),x_ink,y_ink, nROW, nCOLUMN, Eig_D_exp, Eig_V_exp);
235 % MET 2 - \xi z pca drive ziskany
236 % MET 3 - shift
237 Xi_image_shift=FindXi3(OBR, DP(1,1),Eig_D_image, Eig_V_image);
238 Xi_image_shift_q1=FindXi3q(OBR2,DP2(1,1),Eig_D_image_q1, Eig_V_image_q1);
239 Xi_image_shift_q2=FindXi3q(OBR2,DP2(1,1),Eig_D_image_q2, Eig_V_image_q2);
240
241 %% Marginalni hustoty pravdepodobnosti histogramy
242 r=5; n=10;
243 % size(Xi_image)
244 [HyperHist_image,hxi,MIN_image,MAX_image]=xi_hist(Xi_image,...);
245 [HyperHist_gauss,hxg,MIN_gauss,MAX_gauss]=xi_hist(Xi_gauss,...);
246 [HyperHist_exp,hxe,MIN_exp,MAX_exp]=xi_hist(Xi_exp,...);
247 % MET 2 - pca
248 [HyperHist_image_pca,hxip,MIN_image_pca,MAX_image_pca]=
249 =xi_hist(Xi_image_pca,...);
250 % MET 3 - SHIFT
251 [HyperHist_image_shift,hxis,MIN_image_shift,MAX_image_shift]=
252 =xi_hist(Xi_image_shift,...);
253 image_shift_q,hxisq,MIN_image_shift_q,MAX_image_shift_q]=
254 =xi_hist(Xi_image_shift_q,...);
255 [HyperHist_image_shift_q1,hxisq1,MIN_image_shift_q1,MAX_image_shift_q1]=
256 =xi_hist(Xi_image_shift_q1,...);
257 [HyperHist_image_shift_q2,hxisq2,MIN_image_shift_q2,MAX_image_shift_q2]=
258 =xi_hist(Xi_image_shift_q2,...);
259
260 %% Distribucni fce
261 Dist_image=cdf_databased('Image-S2-rnd',hxi,HyperHist_image,...);
262 Dist_gauss=cdf_databased('Gauss-rnd',hxg,HyperHist_gauss,...);
263 Dist_exp=cdf_databased('Exp-rnd',hxe,HyperHist_exp,...);
264 % MET 2 - pca
265 Dist_image_pca=cdf_databased('PCA',hxip,HyperHist_image_pca,...);
266 % MET 3 - SHIFT
267 Dist_image_shift=
268 =cdf_databased('Image-SHIFT',hxis,HyperHist_image_shift,...);
269 Dist_image_shift_q1=
270 =cdf_databased('QUARTER',hxisq1,HyperHist_image_shift_q1,...);
271 Dist_image_shift_q2=
272 =cdf_databased('QUARTER-CROP',hxisq2,HyperHist_image_shift_q2,...);
273
274
275 %% Zpetne generovani obrazku KLR
276 close all;
277

```

```

278 % Nastaveni vysledneho porovnaní
279 nOut=10; nTest=1;
280 step=0.025; ending=0.05;
281 A=zeros(ending/step,1);
282 % Chyba vuci autokovarianci funkci
283 errmat=zeros(nTest,nOut);
284 ERRmat=zeros(length(A),nOut);
285 errmatq=zeros(nTest,nOut);
286 ERRmatq=zeros(length(A),nOut);
287 precmat=zeros(nTest,nOut);
288 PRECmat=zeros(length(A),nOut);
289
290 % Chyba vuci kovarianci souboru
291 errensemble=zeros(length(A),nOut);
292 storesample=zeros(nTest, nROW*nCOLUMN, nOut);
293
294 k=0;
295
296 for j=step:step:ending
297
298     k=k+1;
299     a=round(j*nROW*nCOLUMN);
300     A(k)=a;
301
302     for i=1:nTest
303
304         % % % POMOCI POCTU ZAHRNUTYCH VL. TVARU
305         % % Uniform
306         [OBR_image_shift_u,pimage_shift_u]=KLR('uniform',hxis,...);
307
308         % % Gauss/normal
309         sig=abs(DELTA_lam);
310         [OBR_image,pimage]=KLR('gauss-normal',...);
311         [OBR_gauss,pgauss]=KLR('gauss-normal',...);
312         [OBR_exp,pexp]=KLR('gauss-normal',...);
313         [OBR_image_pca,pimage_pca]=KLR3('gauss-normal',...);
314         [OBR_image_shift,pimage_shift]=KLR('gauss-normal',...);
315         [OBR_image_shift_q1,pimage_shift_q2]=KLR('gauss-normal',...);
316         [OBR_image_shift_q2,pimage_shift_q2]=KLR('gauss-normal',...);
317         % % hist-based
318         [OBR_image_h,pimage_h]=KLR('hist-based',...);
319         [OBR_image_pca_h,pimage_pca_h]=KLR3('hist-based',...);
320         [OBR_image_shift_h,pimage_shift_h]=KLR('hist-based',...);
321         [OBR_image_shift_q1_h,pimage_shift_q1_h]=KLR('hist-based',...);
322         [OBR_image_shift_q2_h,pimage_shift_q2_h]=KLR('hist-based',...);
323
324     %         DATA GAUSS
325     precmat(i,1)=pimage;
326     precmat(i,2)=pgauss;
327     precmat(i,3)=pexp;
328     precmat(i,4)=pimage_pca;
329     %         precmat(i,5)=pimage_h;
330     precmat(i,6)=pimage_pca_h;
331     precmat(i,7)=pimage_shift_h;
332     precmat(i,8)=pimage_shift_u;
333     precmat(i,9)=pimage_shift_q1;
334     precmat(i,9)=pimage_shift_q1_h;
335     precmat(i,11)=pimage_shift_q2;

```

```
336     precmat(i,12)=pimage_shift_q2_h;
337
338     % % HODNOCENI
339     %kovariance vuci periodickemu mediu
340     errmat(i,1)=err_cov_one(OBR_image,DELTA_lam,C_dp);
341     errmat(i,2)=err_cov_one(OBR_gauss,DELTA_lam,C_dp);
342     errmat(i,3)=err_cov_one(OBR_exp,DELTA_lam,C_dp);
343     errmat(i,4)=err_cov_one(OBR_image_pca,DELTA_lam,C_dp);
344     %     errmat(i,5)=err_cov_one(OBR_image_h,DELTA_lam,C_dp);
345     errmat(i,6)=err_cov_one(OBR_image_pca_h,DELTA_lam,C_dp);
346     errmat(i,7)=err_cov_one(OBR_image_shift_h,DELTA_lam,C_dp);
347     errmat(i,8)=err_cov_one(OBR_image_shift_u,DELTA_lam,C_dp);
348     errmat(i,9)=err_cov_one(OBR_image_shift_q,DELTA_lam,C_dp);
349     errmat(i,10)=err_cov_one(OBR_image_shift_q_h,DELTA_lam,C_dp);
350     errmat(i,11)=err_cov_one(OBR_image_shift_q_h,DELTA_lam,C_dp);
351     errmat(i,12)=err_cov_one(OBR_image_shift_q2,DELTA_lam,C_dp);
352
353     % ulozeni pro porovnani se souborem
354     storesample(i,:,1)=OBR_image(:);
355     storesample(i,:,2)=OBR_gauss(:);
356     storesample(i,:,3)=OBR_exp(:);
357     storesample(i,:,4)=OBR_image_pca(:);
358     % storesample(i,:,5)=OBR_image_h(:);
359     storesample(i,:,6)=OBR_image_pca_h(:);
360     storesample(i,:,7)=OBR_image_shift_h(:);
361     storesample(i,:,8)=OBR_image_shift_u(:);
362     storesample(i,:,9)=OBR_image_shift_q1(:);
363     storesample(i,:,10)=OBR_image_shift_q1_h(:);
364     storesample(i,:,11)=OBR_image_shift_q2(:);
365     storesample(i,:,12)=OBR_image_shift_q2_h(:);
366
367     end
368
369     % VYPOCET CHYBY VUCI KOVARIANCI SOUBORU
370     for i=1:nOut
371         CC=cov(storesample(:, :, i));
372         errensemble(k,i)=sum(abs(C_ensemble-CC), 'all');
373     end
374
375     ERRmat(k,:)=mean(errmat,1);
376     PRECmat(k,:)=mean(precmat,1);
377
378     end
```

Bibliografie

- [1] J. Zeman, “Analysis of mechanical properties of fiber-reinforced composites with random microstructure”, dis, Citeseer, 2000.
- [2] 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.
- [3] S. Torquato, *Random Heterogeneous Materials: Microstructure and Macroscopic Properties*. Springer Science & Business Media, 2013, sv. 16.
- [4] 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.
- [5] V. Hlaváč, “Fourierova transformace v 1D a 2D”, *VYSOKÉ UČENÍ TECHNICKÉ V PRAZE, České, Sv. Přednáška, [cit. 1.12. 2013].*, 2012. WWW: <http://people.ciiirc.cvut.cz/~hlavac/TeachPresCz/11DigZpr0br/12FourierTxCz.pdf>.
- [6] A. Klíč, K. Volka a M. Dubcová, *Fourierova transformace s příklady z infračervené spektroskopie*. Vydavatelství VŠCHT v Praze, 2002. WWW: <http://old.vscht.cz/mat/Pavel.Pokorny/students/ft/skripta/Four.pdf>.
- [7] 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>.
- [8] J. Havelka, “Efektivní metody pro propagování nejistot v popisu proudění podzemní vody”, 2014.
- [9] 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.
- [10] H. Jan, “Aplikace hraničních inverzních metod ve stavebním inženýrství”, dis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2019.
- [11] K. Rektorys, *Přehled užití Matematiky, (7. vydání)*, 2000.
- [12] F. W. J. Olver a L. C. Maximon, *Chapter 10 Bessel Functions*, pros. 2020. WWW: <https://dlmf.nist.gov/10>.
- [13] *Bessel Function of the First Kind*. WWW: <https://mathworld.wolfram.com/BesselFunctionoftheFirstKind.html>.
- [14] E. M. Society, *Bessel functions*. WWW: https://encyclopediaofmath.org/wiki/Bessel_functions.