

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta – Katedra matematiky

Diplomová práce

Simulace bodových procesů

Simulation of point processes

Autor: Jiří Kopecký
Vedoucí práce: doc. RNDr. Tomáš Mrkvička, Ph.D.
České Budějovice 2012

Prohlašuji, že jsem diplomovou práci vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu použité literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě, elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce.

Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

v Českých Budějovicích 26. 4. 2012

Jiří Kopecký

Anotace

Cílem diplomové práce je vytvoření balíku funkcí v programu Wolfram Mathematica pro simulaci náhodných množin vybraných hard-core a Gibbsových procesů. Aplikovat tyto modely na reálná data, odhadnout parametry modelů pro reálná data a otestovat validitu modelů. Práce může sloužit jako úvod do teorie prostorových bodových procesů.

Abstract

The aim of the thesis is to create a pack of functions in Wolfram Mathematica software to simulate spatial point patterns of chosen hard-core and Gibbs processes. Then it tries to apply these models to real data, estimate the parameters of models for real data and test the validity of models. The thesis can serve as introduction into the theory of spatial point processes.

Děkuji doc. RNDr. Tomáši Mrkvičkovi, Ph.D. za odborné vedení a za trpělivost.

Děkuji svým rodičům za celoživotní podporu.

Obsah

1. Úvod.....	7
1.1 Základní použité pojmy.....	8
1.1.1 Binomický bodový proces.....	11
1.1.2 Poissonův proces.....	12
2. Hard-core procesy.....	14
2.1 Matérn hard-core proces.....	14
2.2 Dead leaves model.....	17
3 Konečné Gibbsovy procesy.....	18
3.1 Gibbsovy procesy s fixním počtem bodů.....	19
3.1.1 Gibbsův hard-core proces s fixním počtem bodů.....	20
3.1.2 Straussův proces s fixním počtem bodů.....	20
3.1.3 Straussův hard-core proces.....	21
3.2 Gibbsovy procesy s náhodným počtem bodů.....	23
4. Simulace.....	26
4.1 Matérn hard-core proces.....	26
4.2 Dead leaves model.....	28
4.3 Gibbsovy procesy s fixním počtem bodů.....	31
4.3.1 Gibbsův hard-core proces s n body.....	32
4.3.2 Obecný algoritmus zrození a zániku (birth-and-death).....	35
4.3.3 Straussův hard-core proces.....	36
4.4 Gibbsovy procesy s náhodným počtem bodů.....	37
4.4.1 Metropolis-Hastings algoritmus.....	38
4.4.2 Straussův proces s náhodným počtem bodů.....	40
5 Odhady.....	45
5.1 Získání a prvotní zpracování dat.....	45
5.2 Funkce K a L.....	46
5.3 Odhady.....	47
6 Závěr.....	52

1. Úvod

Tato diplomová práce volně navazuje na práci Radky Pantlíkové z roku 2009, která se zabývá statistickou analýzou některých prostorových bodových procesů, jejich vlastnostmi a simulací. Tato problematika je poměrně mladá a hravým způsobem spojuje obě větve mého dosavadního studia na JČU – matematiku a výpočetní techniku.

Mezi hlavní cíle mé práce patří vytvoření komplexního balíku funkcí v programu Wolfram Mathematica 8.0 (WM), které simulují modely vybraných hard-core a Gibbsových procesů, jsou obohaceny o komentáře v problémových místech a názorné příklady. Základní úkol tedy spočívá v návrhu a implementaci kvalitních algoritmů, které efektivně a elegantně simulují vybrané procesy s využitím generátorů pseudonáhodných čísel ve WM.

Výsledný balík v kombinaci s vyloženou teorií a doprovodnými texty může sloužit jako pomůcka pro úvod do studia bodových procesů. Práce by proto měla poskytovat jednoduché, ale zároveň formálně správné základy oboru, bližší přiblížení zmíněných procesů a motivovat čtenáře k využití dalších pramenů. Věřím, že k pochopení pojmů a principů, které je potřeba jinak složitě popsat a definovat, stačí člověku mnohdy jen pár vhodných simulací s komentářem a toho se také pokusím využít.

Chtěl bych srozumitelně objasnit danou problematiku a pojmy i pro další zájemce o proniknutí do tohoto oboru, v němž je hledání informací, vzhledem k jeho aktuálnosti, možné téměř výhradně v anglickém jazyce. To s sebou přináší i značné problémy v terminologii, protože některé základní pojmy nemají v češtině svůj ekvivalent.

Budu se snažit takové případy vhodně opsat nebo budu používat jazykově přizpůsobený originální termín. Snahou je vybrat taková slova, jaká by s největší pravděpodobností (při větší frekvenci užití daného termínu) vznikla sama přirozeným přejímáním do českého jazyka. Zažité slangové výrazy, které vznikly mutací původních

anglických slov, a využily tak ohebnosti obou jazyků, můžeme v češtině nalézt nejen v oboru informatiky. V závorce uvádím i původní (anglické) názvy pojmů, které jednoduše mohou klíčovou myšlenku problému lépe vystihovat, a pomoci tak ke snadnějšímu pochopení nebo zapamatování.

Obecně není příliš známo, že statistická analýza a simulace prostorových bodových procesů našla v moderním světě významné uplatnění. Okruh aplikace zahrnuje mimo matematiky i obory, jako je např. meteorologie, biologie, geografie, astronomie nebo lesnictví. Proto za neméně důležitou považuji i demonstraci využití dosažených výsledků v praxi. Druhá část práce se proto snaží aplikovat vytvořené modely na množinu reálných dat, odhadnout jejich parametry a otestovat jejich validitu.

1.1 Základní použité pojmy

Při studiu hard-core a Gibbsových procesů se neobejdeme bez elementárních znalostí teorie bodových procesů. Tato kapitola si neklade za cíl podrobně popsat všechny základní pojmy, ale spíše se snaží o stručný souhrn tak, aby byl co nejsnáze a intuitivně uchopitelný i pro nováčky v oboru prostorových bodových procesů, kteří hodlají číst další kapitoly textu. Pro přesné znění definic doporučuji především konkrétní kapitoly knihy Illianové a kol. (2008), která je výchozím zdrojem informací celé práce.

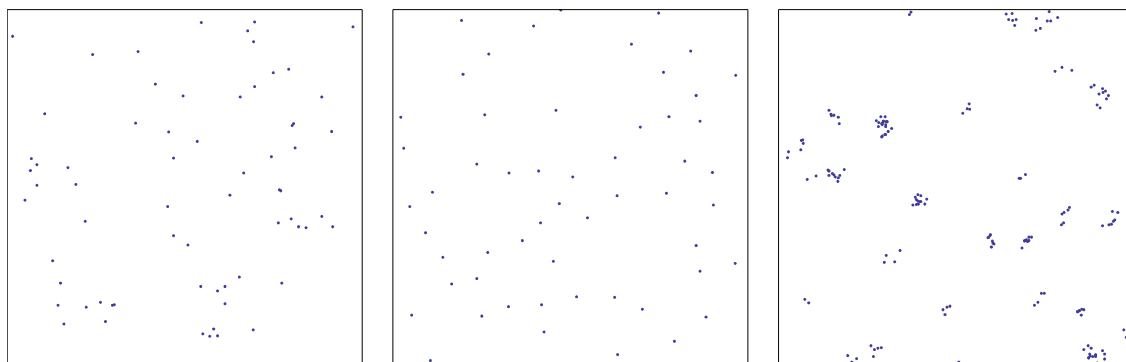
Statistika bodových procesů se snaží analyzovat geometrické struktury množin, které byly vytvořeny objekty náhodně rozmístěnými v jedno-, dvou- či třírozměrném prostoru. Například pozice stromů v lese, vodoměrek na hladině rybníka nebo galaxií ve vesmíru. Tyto objekty jsou přirozeným způsobem reprezentovány pomocí bodů a kót. *Body* popisují umístění objektů a *kóty (marks)* poskytují přídavné informace jako velikost, typ, tvar apod.

Jednoduchým přístupem k analýze prostorových bodových procesů by byla pouhá reprezentace množin objektů jako zobrazení bodů. Vizuelní prozkoumání mnohdy nabízí rychlou kvalitativní charakterizaci typu množiny a může naznačovat i korelace společně s kótami nebo vlivy různých struktur mezi sebou. Obrázek 1.1

ukazuje příklady tří procesů, u kterých lze celkem intuitivně odhadnout, o jaký typ se jedná.

Pro přesnější kvantifikaci, standardizaci a jemnější rozlišení mezi typy prostorového chování je však potřeba sáhnout po vhodných metodách statistiky. Tyto metody poskytují mnohem podrobnější informace o prostorových strukturách, než lze rozeznat pouhým okem.

Statistika bodových procesů čelí různým typům korelací ve zkoumané množině bodů. Relativní vzdálenosti mezi body korelují stejně dobře jako počty bodů v přilehlých oblastech. K tomu mohou být charakteristiky objektů (reprezentovaných body) dále prostorově vázány. Proto se statistická analýza velmi snaží o odhalení a popis těchto korelací.



Obrázek 1.1: Tři různé množiny bodů: (vlevo) náhodná (uprostřed) pravidelná (vpravo) se shluky (*clustery*).

Použitím příslušných statistických metod můžeme náhodné množiny popsat z různých aspektů. Nejjednodušším z nich je *intenzita*, tj. střední počet objektů na jednotku obsahu či objemu. Můžeme si všimnout podobnosti s výběrovým průměrem \bar{x} z klasické statistiky. Složitější charakteristiky typické pro statistiku bodových procesů popisují korelace mezi body vzhledem k jejich vzdálenostem. Např. vzdálenosti nejbližších bodů (funkce F, G, J) nebo počet sousedních bodů do určité vzdálenosti (funkce K a L, o nichž se blíže zmíním v kapitole 5.2).

Jinými slovy se statistika bodových procesů snaží porozumět a popsat interakce mezi body na krátké vzdálenosti, které vysvětlují společnou polohu bodů. Často se může jednat o určitý typ shlukování nebo odpuzování. Analýza náhodných množin tedy poskytuje informace o procesech, které stojí za jejich výslednou podobou, stejně jako o geometrických vlastnostech struktury reprezentované pomocí bodů. Statistika bodových procesů nám pomáhá při modelování těchto struktur a nalezení odpovídajících parametrů těchto modelů, které mohou být použity při klasifikaci a určení strukturálních změn v množinách závisle na čase nebo fyzikálních vlastnostech.

Bodový proces je *stochastický* (pravděpodobnostní) model nesouměrných množin bodů (*point patterns*). Při zkoumání bodového procesu se tedy snažíme o jeho popis v rovině nebo prostoru (rozuměj trojrozměrném, obecně však můžeme myslet všechny abstraktní prostory) tak, že bereme v úvahu náhodné i plánované události.

Point pattern, česky „bodový vzor“, je množina bodů v nějaké oblasti, kterou typicky chápeme jako konkrétní vzorek nebo realizaci bodového procesu. V textu se budu snažit termín *pattern* dle nejlepšího uvážení a v rámci kontextu, opsat slovy „množina“, „konfigurace“, „realizace“ nebo „model“. Ústředním problémem mé práce je *simulace* náhodných množin bodů, resp. vytvoření algoritmů, které je budou simulovat. Tzn. vytvoření algoritmů, které budou produkovat množiny bodů popsané daným procesem. Někdy pro tuto simulaci množin užívám i výraz *simulace procesu*. Zejména při zobrazení (tzn. v popisech obrázků) takto vzniklých množin budu pro zjednodušení užívat slova *simulace* pro daný konkrétní výsledek celého procesu simulace. Neboli pro jednu danou realizaci daného bodového procesu, kterou chápeme jako vzorek (*sample*) tohoto bodového procesu.

Bodový proces označíme N . Budeme tím myslet náhodnou množinu bodů, tj. množinu všech bodů x_1, x_2, \dots procesu. Jinými slovy

$$N = \{x_i\} \quad \text{nebo} \quad N = \{x_1, x_2, \dots\};$$

$x \in N$ znamená, že bod x patří do množiny N . Množina N může být konečná, nebo nekonečná. Pokud je konečná, celkový počet bodů může být deterministický, nebo

náhodný. Dále označme $N(B)$ celkový počet bodů náhodné veličiny procesu N pro omezenou množinu B .

Nejjednodušším a zároveň nejdůležitějším nekonečným bodovým procesem je homogenní Poissonův proces. Tvoří základní kámen pro výstavbu mnoha komplikovanějších modelů.

1.1.1 Binomický bodový proces

Binomický proces můžeme považovat za speciální případ Poissonova procesu a je nejjednodušším příkladem prostorového bodového procesu. Binomický proces se skládá z n bodů, které jsou náhodně rozptýleny v množině W . Výraz „náhodně“ zde znamená, že body x_1, \dots, x_n jsou rovnoměrně a nezávisle rozmístěny ve W . Předpokládejme, že množina W je omezená. Její obsah budeme značit $v(W)$, kde neutrální symbol v může znamenat také objem v kontextu trojrozměrných prostorových modelů.

Představme si nejdříve jeden jediný bod náhodně umístěný v prostoru. To je velmi jednoduchý příklad triviální jednobodové množiny, která nemá praktické využití. Spojením několika takových bodů vznikne binomický bodový proces. Jeho intenzitu můžeme odhadnout jako

$$\lambda = \frac{n}{v(W)}.$$

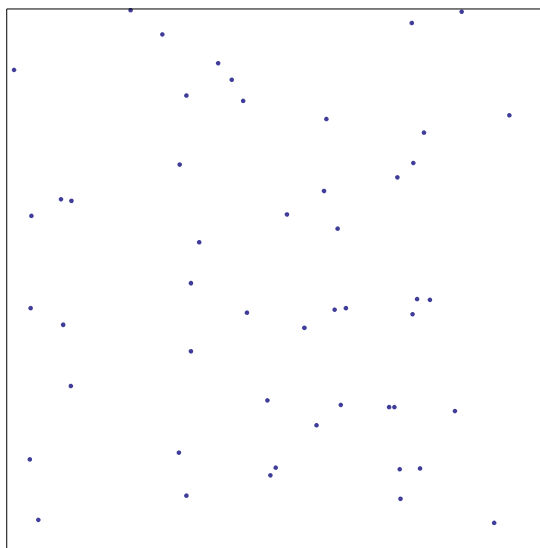
Důležitou metodou statistiky bodových procesů je simulace. Často je nezbytná pro dospění k určitým závěrům. Simulace binomického procesu je nenáročným příkladem a úvodem obecných principů. Jednoduše jde o nezávislé umístění náhodných bodů v požadované oblasti.

Simulace jednoho náhodného bodu s rovnoměrným rozdělením v jednotkovém okně není komplikovaná. Pokud $\{u_n\}$ je posloupnost nezávislých náhodných čísel s rovnoměrným rozdělením na $\langle 0, 1 \rangle$, pak body

$$x_i = (u_{2i-1}, u_{2i}) \text{ pro } i = 1, 2, \dots$$

tvoří posloupnost nezávislých náhodných bodů s rovnoměrným rozdělením v jednotkovém okně. Obecně řečeno, jestliže uvažujeme d -dimenzionální prostor, generuje se posloupnost náhodných bodů s rovnoměrným rozdělením v d -dimenzionální krychli $\langle 0, 1 \rangle^d$ stejným mechanismem, tedy

$$x_i = (u_{(i-1)d+1}, \dots, u_{id}) \text{ pro } i = 1, 2, \dots$$



Obrázek 1.2: Simulace 50 bodů náhodně rozmístěných v jednotkovém okně.

1.1.2 Poissonův proces

V této práci budeme hovořit pouze o *homogenním* případě Poissonova procesu (tzn. jeho vlastnosti jsou ve všech částech W stejné), a proto můžeme slovo „homogenní“ vynechávat. Jednoduchou ukázkou *nehomogenního* Poissonova procesu lze však najít na obrázku 1.3. Homogenní Poissonův proces N je charakterizován dvěma základními vlastnostmi:

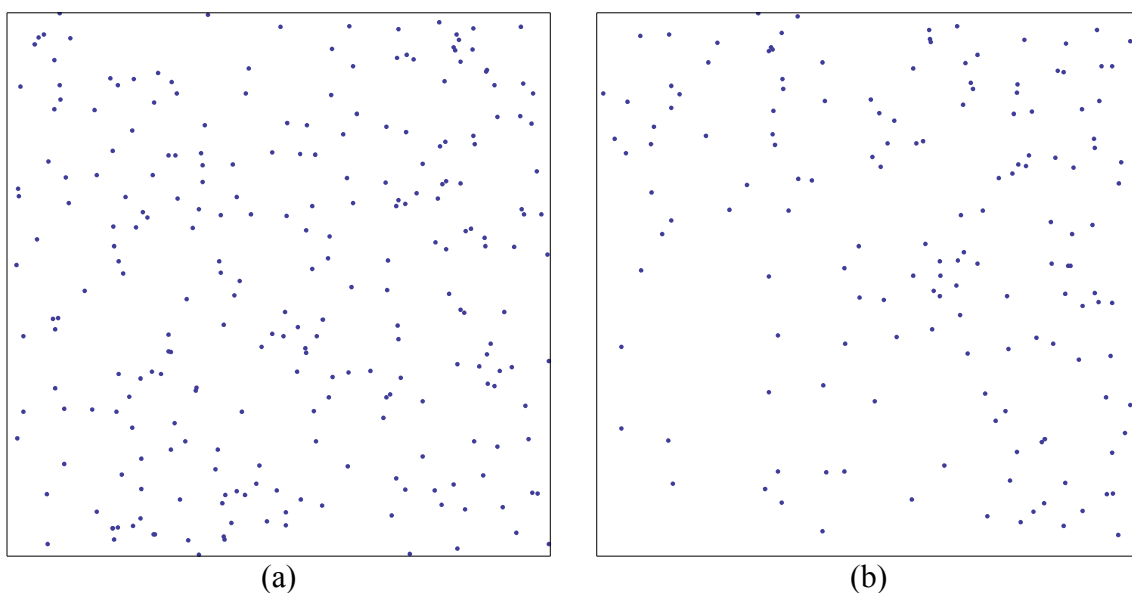
- *Poissonovo rozdělení četností bodů.* Počet bodů N každé omezené množiny B sleduje Poissonovo rozdělení se střední hodnotou $\lambda \cdot \nu(B)$ pro nějakou konstantu λ .
- *Nezávislost skládání.* Počty bodů N v k disjunktních množinách tvoří k nezávislých náhodných veličin pro libovolné k .

Druhá uvedená vlastnost se značí jako „kompletní nahodilost“ (*completely* nebo *purely random*). Poznamenejme, že tato vlastnost neplatí pro středy pevných objektů, které jsou náhodně rozmístěny v prostoru. Modely vhodné pro takové množiny jsou rozvinuty, ale přesahují rámec teorie Poissonova procesu. Takovými modely jsou třeba dále uvedené hard-core procesy.

Charakteristickou vlastností Poissonova procesu je číslo λ , kterému říkáme *intenzita* nebo *hustota bodů*. Udává střední hodnotu počtu bodů, které se vyskytují v jednotce obsahu a je dána vztahem

$$\lambda \cdot \nu(B) = \mathbf{E}(N(B)) \quad \text{pro všechny omezené množiny } B.$$

Simulace Poissonova procesu se skládá ze dvou kroků. Prvním je určení počtu bodů uvnitř W pomocí náhodného čísla s Poissonovým rozdělením, v druhém kroku jsou pak simulací určeny pozice bodů uvnitř W binomického procesu ve W s počtem bodů určeným v prvním kroku.



Obrázek 1.3: Simulace (a) homogenního Poissonova procesu (b) nehomogenního Poissonova procesu. U nehomogenního případu si můžeme všimnout, že s narůstající vzdáleností od počátku pozorovacího okna roste i intenzita procesu. Je to jen jeden z mnoha příkladů nehomogenity.

2. Hard-core procesy

Hard-core bodové procesy jsou takové bodové procesy, ve kterých se nevyskytují žádné dva body vzdálené od sebe dále, než je daná minimální vzdálenost r_0 . Popisují modely reprezentující umístění středů nepřekrývajících se objektů, obvykle kružnic nebo koulí s poloměrem $R \leq r_0 / 2$.

Budeme zkoumat dva hlavní typy hard-core procesů:

Procesy vytvořené *operací ředění* (Pantlíková, 2009, s. 10). Tyto operace mohou odstranit body, které jsou příliš blízko k ostatním, nebo eliminovat body v clusterech tak, že vznikne samostatný model izolovaných bodů.

Procesy vytvořené *interakcí pevných objektů*. Tím myslíme modely reprezentující takové objekty, které jsou pevné a nemůžou se vzájemně prolínat. Takže pokud jsou náhodně rozmístěny v prostoru, nemohou být k sobě blíže, než připouští jejich velikost. Takové objekty se mohou vyskytovat buď současně od samého začátku nebo se objevovat v průběhu času.

2.1 Matérn hard-core proces

Matérn předložil dva návrhy hard-core modelů. Popisujeme jeho druhý typ, který podléhá vyšší výsledné hustotě bodů.

Model je založen na závislém ředění Poissonova procesu N_b s mírou intenzity λ_b . Body v N_b jsou nezávisle okótovány náhodnými čísly s rovnoměrným rozdělením na intervalu $(0, 1)$. Závislé ředění ponechá ty body x procesu N_b s kótami $m(x)$, jestliže sféra $b(x, r_0)$ neobsahuje žádné body z N_b s kótami menšími než $m(x)$.

V aplikacích pak body N_b mohou být třeba pozice semen, zatímco kóty odpovídají bodům v čase, ve kterém vyklíčí. Potom výsledný proces N obsahuje pozice x těch rostlin, jejichž semena vyklíčila jako první uvnitř kruhu $b(x, r_0)$. Tj. oblasti kolem rostliny potřebné pro absorpci živin.

Intenzita λ výsledného procesu N je dána vzorcem

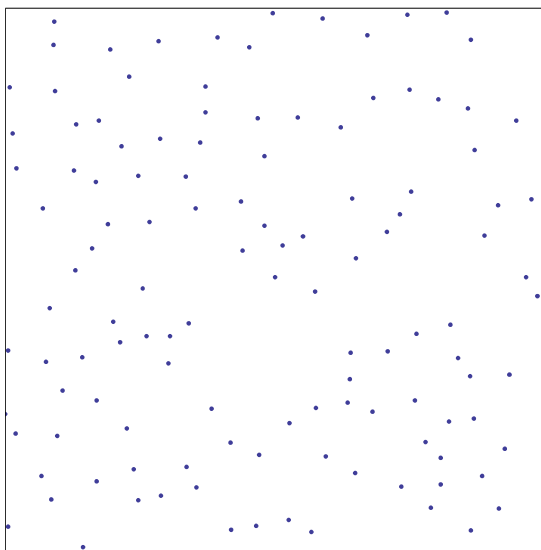
$$\lambda = p \lambda_b.$$

Neznámá p je tzv. Palmova retenční pravděpodobnost (Illian et. al., s. 177) „typického bodu“ v N_b určená jako

$$p = \int_0^1 r(t) dt = (1 - \exp(-\lambda_b V)) / (\lambda_b V) \quad \text{pro } V = b_d r_0^d,$$

kde $r(t) = \exp(-\lambda_b V t)$ je pravděpodobnost zachování bodu s kótou t a $0 < t < 1$. Vzorec vyplývá z pozorování, že podproces procesu N_b obsahující body s kótou menší než t je vlastně t -ředění Poissonova procesu, které je samo Poissonovým procesem s intenzitou $\lambda_b t$. Po dosazení

$$\lambda = (1 - \exp(-\lambda_b V)) / V.$$



Obrázek 2.1: Znáornění simulace rovinného Matérn hard-core procesu v jednotkovém čtverci pro $\lambda_b = 200$ a $r_0 = 0,039$. Výsledná intenzita (počet bodů v grafu) konkrétní simulace je $\lambda = 123$.

Příklad 1: Dosazením do vzorce určíme pravděpodobnou intenzitu λ Matérn hard-core procesu s minimální vzdáleností mezi body $r_0 = 0.039$ a intenzitou $\lambda_b = 150$. Díky nasimulování dostatečného množství Matérn hard-core procesů vytvořených

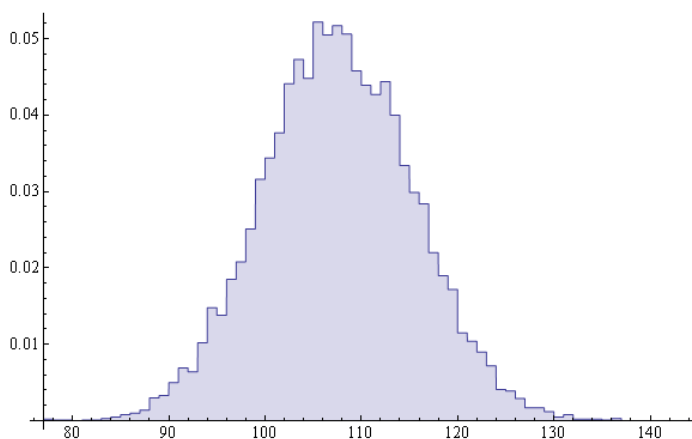
v kapitole (4.1) potom budeme moci pozorovat, zda skutečně střední hodnota počtů bodů nasimulovaných procesů odpovídá určené hodnotě λ .

Protože pracujeme v rovině, uijeme $V = \pi r_0^2$, v prostoru bychom dosadili $V = 4/3 \pi r_0^3$. Ve WM si pro dané hodnoty vyčíslíme $\lambda = 107,08$.

Nyní necháme program simulovat co největší počet Matérn hard-core procesů. Pro naše účely budeme zjišťovat pouze jejich výsledný počet bodů, který budeme ukládat do seznamu.

Z tohoto seznamu si poté necháme spočítat střední hodnotu získanou pomocí funkce *Mean*. Výsledek 107,102 se blíží očekávané hodnotě λ .

Pomocí zabudované funkce *HistogramDistribution* ze seznamu snadno získáme rozdělení pravděpodobnosti odpovídající rozdělení četností získaných dat. Hustotu tohoto rozdělení nakonec necháme vykreslit do grafu, který omezíme zleva (resp. zprava) nejmenším (resp. největším) prvkem našeho seznamu.



Obrázek 2.2: Hustota pravděpodobnosti intenzity pro 10 000 nasimulovaných Matérn hard-core procesů (graf s výplní).

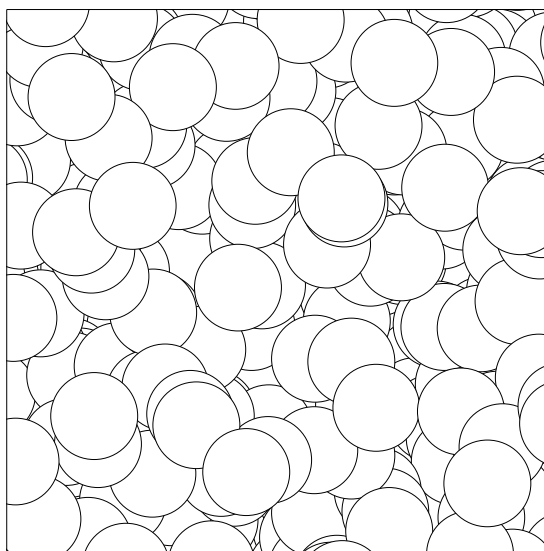
Celý příklad včetně výpočtů a vykreslení grafu ve WM je uložen v sešitě *MaternHC.nb*, který je součástí balíku přiloženého k diplomové práci.

2.2 Dead leaves model

Matheron představil v roce 1968 zajímavý model (česky *model spadaneho listí*) nepřekrývajících se kruhů (disků) nebo koulí (sfér), jehož rovinným případem se budeme zabývat. Pokládáme disky (s konstantním nebo náhodným poloměrem) náhodně a rovnoměrně do roviny po stanovenou dobu v časovém intervalu $(-\infty, 0)$ tak, že

- (1) nový disk může zakrýt disky umístěné do roviny už dříve a
- (2) do libovolné podmnožiny roviny lze umístit nekonečně mnoho disků.

Tímto způsobem vygenerujeme obraz podobný Obrázku 2.3, který můžeme chápat jako rozvrstvení kruhových listů viděné shora (obdobně též zespuď).



Obrázek 2.3: Simulace dead leaves modelu pro disky s konstantním poloměrem.

Nyní uvažujeme pouze disky z nejvyšší vrstvy, které nejsou překryty ani se navzájem nedotýkají. Jejich středy tvoří hard-core proces se stejným rozdělením jako Matérn hard-core proces uvedený výše v limitním případě $\lambda_b \rightarrow \infty$.

Pro rovinný případ s konstantním poloměrem $R = r_0/2$ platí pro výslednou intenzitu procesu vztah $\lambda = 1/(\pi r_0^2)$.

3 Konečné Gibbsovy procesy

V mnoha případech se analýza prostorových bodových procesů zaměřuje na detekci a charakterizaci interakcí mezi body. Takovou interakcí je často vzájemné odpuzování vedoucí k modelům s nějakou pravidelností. V některých aplikacích se body mohou vzájemně ovlivňovat i pozitivně, tj. jeden druhého přitahovat, což se odráží ve tvorbě shluků (clusterů).

Takové modely se často objevují v různých oblastech biologie, ale i ve vědních oborech fyziky zabývajících se studiem materiálů. Jazyk fyziky popisuje interakce mezi body jako síly, které mohou být odpudivé či přitažlivé a jejichž působení tvoří výslednou podobu modelu. Ve fyzikálních modelech jsou parametry popisující tyto energie, tzv. *párový potenciál*, většinou dané a zkoumá se výsledný model. Z hlediska statistického přístupu jsou takové modely, tj. vzory bodových procesů, zkoumány a odpovídající parametry se hledají.

Stejně jako mohou tyto síly vycházet z různých předpokladů, lze sestavit podobným způsobem mnoho různých bodových procesů, které spadají do třídy Gibbsových procesů. Je to skupina univerzálních modelů, použitelných zejména pro bodové procesy s odpuzováním. Velké množství jich lze vhodně a elegantně definovat pomocí hustoty rozdělení pravděpodobnosti (*density function*). Navzdory některým obtížím není tvorba bodových procesů založených na této funkci nezbytně komplikovaná. V souvislosti s Gibbsovými procesy se moderní přístup ukazuje jako velmi mocný a celkem intuitivní.

V následující kapitole se pokusím krok za krokem postupně přiblížit některé Gibbsovy procesy od těch nejjednodušších až po složitější. Nejdříve se zaměřím na procesy s pevně daným počtem bodů v okně W , což je oblast výskytu procesu. Poté bude možno přistoupit k procesům s náhodným počtem bodů ve W .

3.1 Gibbsovy procesy s fixním počtem bodů

Nechť je n bodů náhodně rozložených ve W , n je pevně dané. Předpokládejme, že jejich pozice jsou dány hustotou pravděpodobnosti s více proměnnými

$$f_n(x_1, \dots, x_n) \text{ pro } x_1, \dots, x_n \in W,$$

kteřá nezáleží na pořadí prvků. Tato hustota určuje bodový proces s právě n body ve W . Pro další pochopení předpokládejme následující interpretaci této funkce. Vytvoříme realizaci $\{x_1, \dots, x_n\}$ z binomického procesu s n body pro W . Tuto realizaci přijmeme s pravděpodobností úměrnou $f_n(x_1, \dots, x_n)$.

Pro binomické procesy, nebo ekvivalentně Poissonovy procesy s n body, které také spadají do této třídy bodových procesů, tedy zřejmě platí $f_n(x_1, \dots, x_n) = 1/v(W)^n$. Zaměřím se ale na procesy, které mohou být použity ke konstrukci modelů, v nichž se projevuje vzájemná interakce mezi body. Proto bude zapotřebí složitějších typů hustoty pravděpodobnosti s více proměnnými.

Elegantním a užitečným příkladem takového procesu je Gibbsův proces s fixním počtem bodů s párovým potenciálem $\phi(r)$. Předkládá hustotu

$$f_n(x_1, \dots, x_n) = \exp\left(-\sum_{i=1}^{n-1} \sum_{j=i+1}^n \phi(\|x_i - x_j\|)\right) / Z_n \text{ pro } x_1, \dots, x_n \in W.$$

Funkce $\phi(r)$ ve vzorci se často nazývá *párový potenciál*, původem z fyziky. Určuje „potenciální energii“ způsobenou interakcí dvojice bodů (x_i, x_j) jako funkci jejich vzdálenosti $\|x_i - x_j\|$. Může nabývat hodnot $-\infty < \phi(r) < \infty$ a podle dohody platí $\exp(-\infty) = 0$. Výraz Z_n je normující činitel, který zaručuje, aby funkce byla skutečně hustotou pravděpodobnosti, tzn. že její integrál je roven jedné. Výpočet Z_n či jeho přijatelná aproximace jsou ve většině případů extrémně náročné.

Dvojice bodů se vzdáleností r , pro které $\phi(r) > 0$, přispívají k exponentu ve vzorci jen málo, a proto se vyskytují v konečné konfiguraci s menší pravděpodobností než dvojice, pro něž $\phi(r) = 0$, .

To se v modelu projevuje jako „odpuzování“ takového páru. Naopak následkem $\phi(r) < 0$ jsou k sobě body „přitahovány“. V případě $\phi(r) = 0$ není mezi body žádná interakce. Pokud existuje r_{max} takové, že $\phi(r) = 0$ pro všechna $r > r_{max}$, označujeme r_{max} jako *rozpětí interakce*. Je to vzdálenost, za kterou už žádná interakce není.

Uvažovat můžeme několik různých typů funkcí párového potenciálu. Nejdříve uvedu příklady, které jsou použity při simulacích v další kapitole a vedou k modelům značného praktického zájmu.

3.1.1 Gibbsův hard-core proces s fixním počtem bodů

Proces je definován funkcí párového potenciálu

$$\phi(r) = \begin{cases} \infty & \text{pro } r \leq r_0 \\ 0 & \text{pro } r > r_0 \end{cases}$$

s pevnou *hard-core* vzdáleností r_0 . Nevyskytují se zde žádné dvojice bodů, které jsou blíže než r_0 . Výsledkem je „pravidelný“ bodový proces, pokud n není příliš malé a W příliš velké. Naopak pro malé W , velké n a r_0 je možné, že proces není proveditelný.

Tento model je užitečný například v situacích, kde body jsou středy kulových či kruhových nepružných částic stejné velikosti a r_0 jejich poloměr, který je pro všechny částice stejný.

3.1.2 Straussův proces s fixním počtem bodů

Definujme

$$\phi(r) = \begin{cases} \beta & \text{pro } r \leq r_1 \\ 0 & \text{pro } r > r_1 \end{cases}$$

kde $0 < \beta < \infty$. Parametr r_1 je stejný jako r_{max} . Hustota pravděpodobnosti zde má tvar

$$f_n(x_1, \dots, x_n) = \exp(-\beta n_2(r_1)) / Z_n \text{ pro } x_1, \dots, x_n \in W,$$

kde

$$n_2(r) = \frac{1}{2} \sum_{x_i, x_j \in N} 1(\|x_i - x_j\| \leq r) \text{ pro } r > 0, i \neq j$$

udává počet dvojic bodů, které jsou od sebe vzdáleny r nebo blíže. Funkce $n_2(r)$ je postačující statistika pro β , tzn. celá informace dostupná o β bodového procesu se odráží v počtu r_1 -blízkých párů.

3.1.3 Straussův hard-core proces

Někdy také označován jako „square well“ čili čtvercový zásobník. Jde v podstatě o kombinaci dvou předchozích. Párový potenciál je určen funkcí

$$\phi(r) = \phi_{r_0, \beta, r_{max}}(r) = \begin{cases} \infty & \text{pro } r \leq r_0 \\ \beta & \text{pro } r_0 < r \leq r_{max} \\ 0 & \text{pro } r > r_{max} \end{cases}$$

kde r_0 je opět hard-core vzdálenost. Tzn. že se zde nevyskytují body blíže než r_0 .

Nalézt se dá mnoho dalších modelů a každý si může pro svou aplikaci vytvořit vlastní párový potenciál. Zmíním ještě několik příkladů:

- *overlap model*

$$\phi(r) = \begin{cases} \theta \left(\frac{r_1^2}{2} \arccos\left(\frac{r}{r_1}\right) - \frac{r}{2} \sqrt{r_1^2 - r^2} \right) & \text{pro } 0 < r < r_1 \\ 0 & \text{v ostatních případech} \end{cases}$$

kde θ je nezáporný parametr modelu. Jedná se o vylepšený Straussův proces.

- *multiscale*

$$\phi(r) = \begin{cases} \infty & \text{pro } r \leq r_0 \\ \beta_1 & \text{pro } r_0 < r \leq r_1 \\ \beta_2 & \text{pro } r_1 < r \leq r_2 \\ \vdots & \\ \beta_k & \text{pro } r_{k-1} < r \leq r_{max} \\ 0 & \text{pro } r > r_{max} \end{cases}$$

- *very-soft-core*

$$\phi(r) = -\ln\left(1 - \exp\left(\frac{-r^2}{\sigma^2}\right)\right) \text{ pro } 0 < r < \infty$$

- *Lennart-Jonesův potenciál*

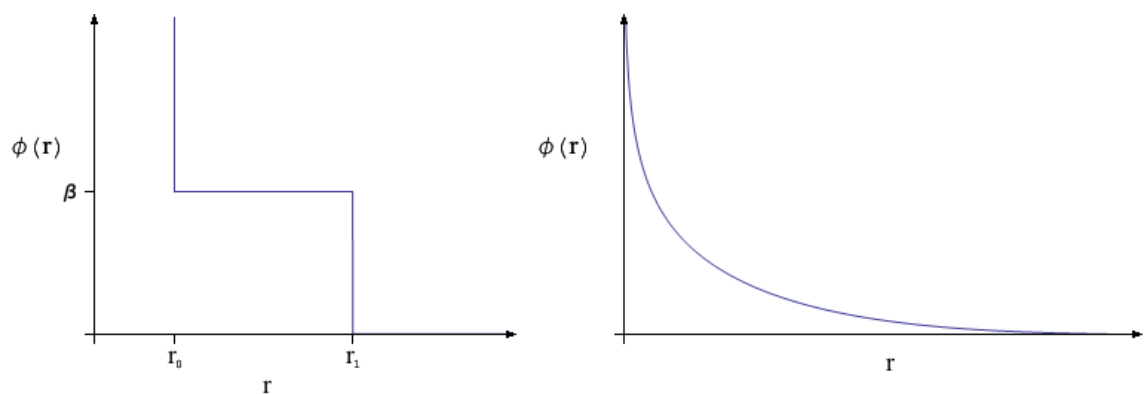
$$\phi(r) = \alpha_1 \left(\frac{\sigma}{r}\right)^{n_1} - \alpha_2 \left(\frac{\sigma}{r}\right)^{n_2}$$

pro $0 < r < \infty$ a $\alpha_1, \sigma > 0, \alpha_2 \geq 0$ a $n_1 > n_2$

Jak je napsáno v úvodu kapitoly, funkce párového potenciálu $\phi(r)$ má svůj původ ve statistické fyzice. V tomto kontextu ji lze chápat jako míru energie: určuje příspěvek dvojice bodů (x_i, x_j) vzdálených r do energie celkového systému bodů. Toto vzájemné působení bodů mezi sebou se dá popsat také pomocí funkce *párové interakce*

$$h(r) = e^{-\phi(r)}.$$

Tato funkce vyhovuje rovnici $h(r) = 1$, pokud spolu dvojice bodů (x_i, x_j) se vzdáleností r neinteraguje. Páry bodů (x_i, x_j) se vzdálenostmi r vykazují odpuzování pro $h(r) < 1$ a shlukování pro $h(r) > 1$.



Obrázek 3.1: Funkce pro Straussův hard-core (vlevo) a very-soft-core (vpravo) párový potenciál.

3.2 Gibbsovy procesy s náhodným počtem bodů

Předchozí kapitola by měla být spolu s praktickou ukázkou v kapitole 4.3 dostatečnou průpravou a motivací pro bližší seznámení se s Gibbsovými procesy s náhodným počtem bodů. Tyto procesy jsou rovněž popsány hustotou pravděpodobnosti, ale využijeme ještě symboliky \mathbf{x} z teorie množin, kde \mathbf{x} je konfigurace libovolného počtu bodů W z prostoru konfigurací N_{fin} , který obsahuje prázdnou množinu, všechny jednoprvkové množiny $\{x_1\}$ pro $x_1 \in W$, všechny dvojice $\{x_1, x_2\}$ bodů $x_1, x_2 \in W$, podobně pro všechny trojice atd. $\{x_1, \dots, x_n\}$ je tedy konfigurace složená z prvků x_1, \dots, x_n .

Rozdělení procesu N je dáno hustotou pravděpodobnosti $p(\mathbf{x})$ vzhledem k rozdělení Poissonova procesu jednotkové intenzity zúženého do W , zjednodušeně

$$\int_{N_{fin}} p(\mathbf{x}) \Pi(d\mathbf{x}) = 1,$$

neboli

$$\mathbf{P}(N \in A) = \int_A p(\mathbf{x}) \Pi(d\mathbf{x}),$$

kde A je podmnožina N_{fin} nebo událost v bodovém procesu, Π vyjadřuje rozdělení Poissonova procesu zúženého na W .

Rozdělení $\mathbf{P}(N)$ lze nejnázne pochopit s přihlédnutím na simulaci. Nejprve je vygenerován vzorek \mathbf{x} Poissonova procesu. Ten je přijat jako realizace s pravděpodobností úměrnou $p(\mathbf{x})$. Ta může být zachycena *vylučovací metodou* (tj. určíme maximum M funkce p . Vygenerujeme náhodné číslo u rovnoměrně na $\langle 0, M \rangle$ a srovnáme ho s hodnotou $p(\mathbf{x})$. Pokud $u < p(\mathbf{x})$, potom je \mathbf{x} přijato. Jinak je zamítnuto a je navržen nový vzorek). Pokud je například hustota p nulová pro konfigurace s minimálními vzdálenostmi mezi body menšími než r_0 , budou generované výsledky ukázkami hard-core procesů.

Gibbsův proces s náhodným počtem bodů je definován následující hustotou vzhledem k Poissonovu procesu jednotkové intenzity:

$$p(\{x_1, \dots, x_n\}) = \exp\left(-\left(\alpha n + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \phi(\|x_i - x_j\|)\right)\right) / Z$$

kde $x_1, \dots, x_n \in W$ pro $n = 0, 1, \dots$. V této rovnici se objevuje self-potenciál α , který je důležitým členem totální energie

$$U(x_1, \dots, x_n) = \alpha n + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \phi(\|x_i - x_j\|).$$

Podobně jako v případě fixního počtu bodů funkce párového potenciálu $\phi(r)$ zde popisuje interakce. Self-potenciál α mezitím společně s párovým potenciálem určují $p_n = \mathbf{P}(N(W) = n)$, tzn. rozdělení počtu bodů. Stejně funkce pro párový potenciál jako pro Gibbsovy procesy s fixním počtem bodů (3.1) mohou být použity i pro Gibbsovy procesy s náhodným počtem bodů. Normalizující faktor Z závisí na W , α a funkcí párového potenciálu $\phi(r)$. Obvykle je analyticky neřešitelný.

Podmíněnou intenzitou $\lambda(x|X)$ budeme myslet podmíněnou pravděpodobnost výskytu nekonečně malé sféry $b(x)$ obsahující x při dané realizaci X z N mimo $b(x)$.

Konečně zavedeme podmíněnou intenzitu $\lambda(x|X)$ pro umístění daného bodu x a bodový proces X . Je to náhodná veličina se střední hodnotou $\mathbf{E}(\lambda(x|N)) = \lambda(x)$. Pomocí podmíněné intenzity můžeme model explicitně vyjádřit výrazem

$$\lambda(x | \{x_1, \dots, x_n\}) = \frac{p(\{x_1, \dots, x_n, x\})}{p(\{x_1, \dots, x_n\})} = \exp\left(-\alpha - \sum_{i=1}^n \phi(\|x - x_i\|)\right),$$

který neobsahuje normalizující faktor. Pouze ty body x_i , které interagují s x , tj. body pro které $\phi(\|x - x_i\|) \neq 0$, přispívají do sumy v exponentu. Často je počet těchto bodů velmi malý ve srovnání s celkovým počtem bodů. Poslední uvedený vzorec je základním poznatkem pro simulaci Gibbsových procesů.

4. Simulace

4.1 Matérn hard-core proces

Při simulaci Matérn hard-core procesu budeme jednoduše postupovat podle definice. Vstupními hodnotami jsou intenzita λ_b nezřetězeného procesu N_b , rozměry A, B pozorovacího okna W a nejmenší možná vzdálenost dvou bodů r_0 . Pokud chceme zadat proces s výslednou intenzitou λ , necháme si odpovídající vstupní parametr λ_b našeho algoritmu najít jako kořen rovnice $\lambda = (1 - \exp(-\lambda_b V)) / V$.

Protože mohou být body uvnitř okna W ovlivněny i body za jeho hranicemi, rozšíříme si ho do všech stran o vzdálenost r_0 . Do takto rozšířeného okna, označme jej $W \oplus r_0$, si vytvoříme výchozí Poissonův proces s intenzitou λ_b přímo úměrnou rozměrům okna. Na konci simulace nesmíme z procesu body za hranicí W zapomenout zase smazat.

```
PoissonSubProcess[ $\lambda_b$ _, A_, B_,  $r_0$ _] := Module[ {NumbPoint},  
  NumbPoint = Random[PoissonDistribution[ $\lambda_b$ *(A+2 $r_0$ )*(B+2 $r_0$ )]];  
  Table[{Random[UniformDistribution[{- $r_0$ , A+ $r_0$ }]],  
    Random[UniformDistribution[{- $r_0$ , B+ $r_0$ }] ]}, {NumbPoint}]  
];
```

Všem bodům procesu přiřadíme nezávisle na sobě náhodné kóty s rovnoměrným rozdělením na $(0,1)$. Pro uložení známek vytvoříme zvláštní seznam, ve kterém bude pořadí známek odpovídat pořadí bodů v seznamu procesu.

Poté můžeme přistoupit k samotnému ředění procesu. Postupně porovnáváme všechny dvojice bodů a v případě, že jsou k sobě blíže než r_0 , ten s vyšší kótou smažeme.

Poznamenejme ještě, že body, které chceme smazat stále ovlivňují ne/smazání ostatních bodů. Proto si body, které nebudou ve výsledném procesu obsaženy, označíme obdobně jako v seznamu známek hodnotou 0 nebo 1 a odstraníme je až po porovnání všech dvojic.

```

MaternHCProcess[ $\lambda$ _, r0_, A_, B_] := Module[{proces, znamky, delka, smazat},
  proces = PoissonSubProcess[ $\lambda$ , A, B, r0];
  delka = Length[proces];

  znamky = Table[NahodZnamka, {delka}];
  (*na zacatku není oznacen ke smazani zadny bod, tzn. same nuly*)
  smazat = Table[0, {delka}];

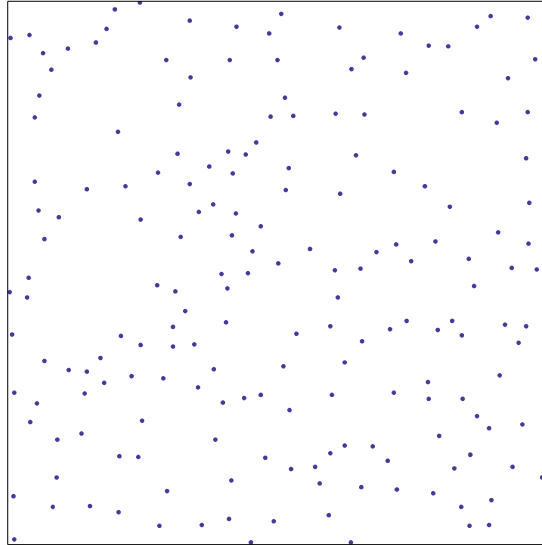
  Do[
    If[Vzdalenost[proces[[i]], proces[[j]]] < r0,
      If[znamky[[j]] < znamky[[i]],
        smazat[[i]] = 1,
        smazat[[j]] = 1];
    ],
    {i, delka}, {j, i+1, delka}
  ];

  (*ke smazani oznacime take vsechny body procesu za hranici okna W*)
  If[!VokneQ[A, B, proces[[#]]],
    smazat[[#]] = 1 ]&/@Range[delka];

  (*vymaz vsechny body, ktere jsou oznaceny ke smazani*)
  proces = Delete[proces, Position[smazat, 1]];
  proces
];

```

Celý zdrojový kód včetně pomocných funkcí *NahodZnamka*, *Vzdalenost*, *VokneQ*, spuštění simulace a jejího zobrazení ve WM je uložen v sešitě *MaternHC.nb* přiloženém k diplomové práci.



Obrázek 4.1: Ukázka výsledku Matérn hard-core procesu pro hodnoty $r_0 = 0,029$; $\lambda = 180$; $A = 1$; $B = 1$. Výsledný proces tvoří ve skutečnosti 184 bodů.

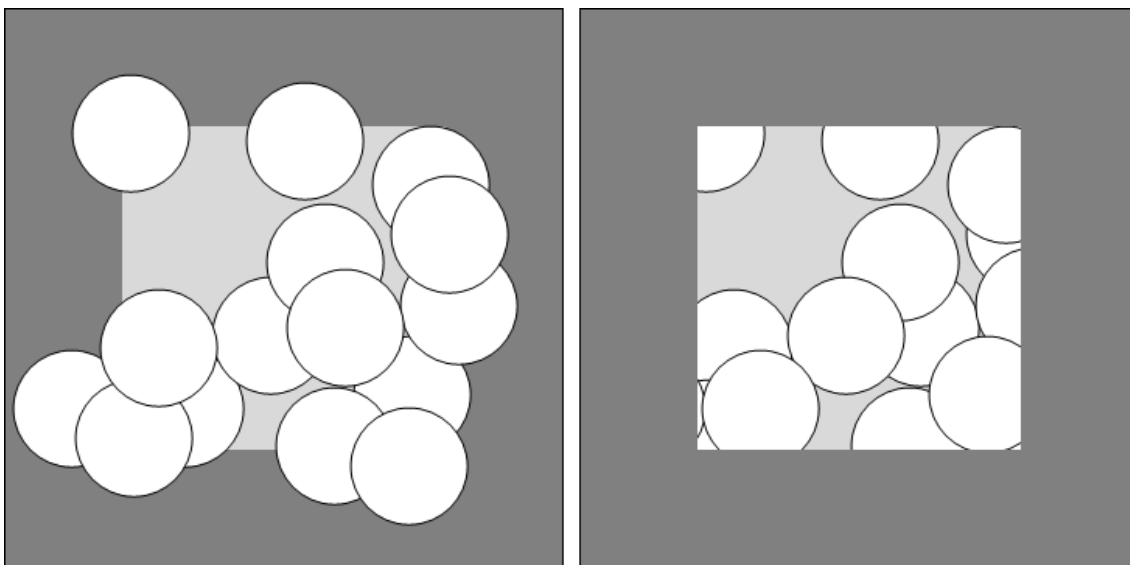
4.2 *Dead leaves model*

Jak jsem uvedl v kapitole 2.2, můžeme si model představit jako pohled na oblast plochy pokryté listy. Pokud proces padání listů začal v čase $-\infty$, na konci v čase 0 musí být pozorovací okno W listy zcela zakryto.

Protože algoritmus nemůže z podstaty vykonat nekonečně mnoho kroků, potřebujeme pro přesnou simulaci takový model, který se v určité fázi ustálí a v dalších krocích už se nebude měnit. Takový stav nazýváme Nashovo ekvilibrium (popř. Nashova rovnováha).

Uvažujeme-li model, kdy se díváme na pokládané disky shora, tohoto stavu však nedosáhneme. S každým novým listem se nám výsledný obraz vždy mění. Pokud bychom zastavili algoritmus ve fixním čase, existuje reálná pravděpodobnost, že pozorovací okno nebude zcela zakryté. Kdybychom přijali stav v prvním čase, kdy je okno disky zcela zakryto, mohou být výsledky ovlivněny směrem ke konfiguracím, které W pokrývají efektivně.

Proto změňme úhel pohledu a postupovat budeme opačně. Na disky se nyní díváme zespodu – jako bychom v jediném bodě času překlopili pohled pod skleněnou desku stolu s rozloženými kartami, nebo chcete-li, jako malý živočich v noře, jejíž otvor je zasypáván listím. Každý nový list se nám teď bude jevit, jako by ležel *pod* listy, které spadly dříve. Jakmile je W poprvé zcela zakryto, výsledný obraz už se s přibývajícím listím nebude dále měnit a dosáhli jsme ekvilibria.



Obrázek 4.2: Dva různé pohledy na stejnou situaci při simulaci dead leaves modelu. Vlevo pohled shora, vpravo pohled zespodu – listy, které spadly v čase dříve, se jeví blíže. Při podobném grafickém zobrazení se tedy musí vykreslovat v opačném pořadí. Lze dokázat, že oba přístupy mají v libovolném čase stejné rozdělení pravděpodobnosti ([3], s. 15).

Vstupními parametry našeho procesu tedy budou rozměry okna A, B a konstantní poloměr disků $R = r_0/2$. Vzhledem ke vzájemnému ovlivňování prvků procesu provádíme simulaci pro rozšířené okno $W \oplus R$.

Střediska disků umístíme náhodně a rovnoměrně. Protože ověřování pokrytí okna je podstatně náročnější na výpočetní výkon než tvorba takových bodů, nebudeme je přidávat jednotlivě, ale po „dávkách“. Intenzitu každého nově přidaného procesu odhadují parametrem $A \cdot B / R^2$.

```

DavkaListu[R_,A_,B_] := Module[ {pocet},
  pocet = Round[((A+2R)*(B+2R))/R^2];
  Table[{Random[UniformDistribution[{-R,A+R}]],
    Random[UniformDistribution[{-R,B+R}]]}, {pocet}]
];

```

Jako nejefektivnější metoda ověření, zda je zakryto celé okno W , se mi osvědčilo zobrazení všech disků jako kruhů s černou výplní na bílém pozadí. Takto vzniklou grafiku rozložím pomocí funkce *ImageData* na seznam barevných hodnot jednotlivých pixelů. Pokud je součet všech prvků seznamu roven nule, obsahuje grafika pouze černé body, a okno je tedy celé pokryto listy.

Pro zjednodušení algoritmu považujeme pořadí listů v seznamu *vsechnoListi* za jednoznačné určení pořadí, v jakém spadly. Proto je ponechán tento původní seznam beze změny po celou dobu běhu procesu a úpravy jsou prováděny do jeho kopie *proces*.

Poté už stačí pouze projít celý seznam od začátku do konce a pro každý disk zkontrolovat, zda se překrývá s nějakým blíže začátku a jestli střed leží uvnitř okna W .

```

DeadLeaves[R_,A_,B_] := Module[{vsechnoListi, kontrast, bilaMista, proces},
  vsechnoListi = DavkaListu[R,A,B];

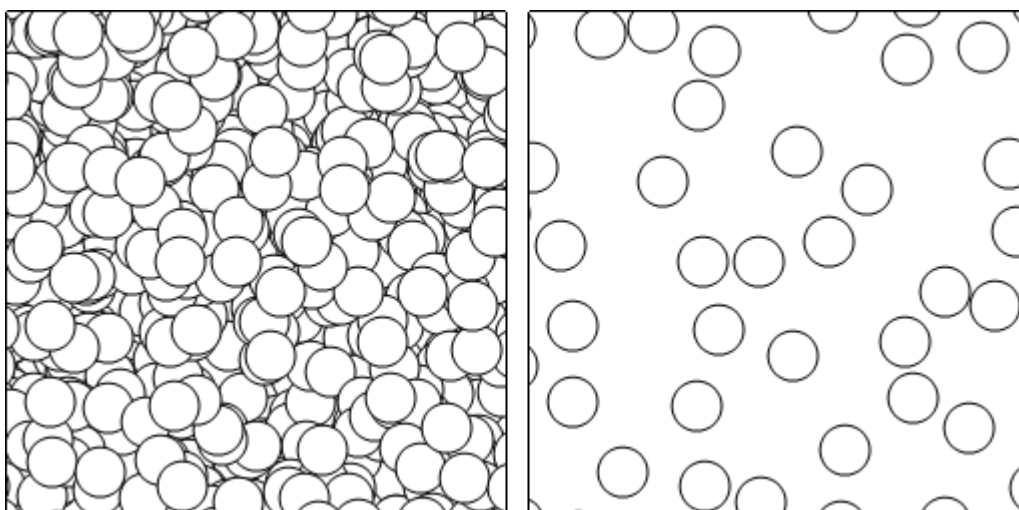
  (*dokud najdeme nepokryta mista opakujeme proces pridavani listu*)
  bilaMista = 1;
  While[bilaMista > 0,
    vsechnoListi = Join[vsechnoListi, DavkaListu[R,A,B]];
  (*vytvorime sjednoceni vseh listu v pozorovacim okne jako grafiku*)
  kontrast = Binarize[Graphics[{Black, Disk[#,R]&/@vsechnoListi}, PlotRange
    {{A,0},{0,B}}, PlotRangeClipping -> True, FrameTicks -> None]];
  (*rozebereme ji na jednotlivy pixely jako hodnoty 0,1 a ty pak secteme, pokud
  je soucet vetsi nez nula, jsou tam bila (nepokryta) mista*)
  bilaMista = Total[Flatten[ImageData[kontrast, "Bit"]]];
  ];
  proces = vsechnoListi;

```

```

Do[
  If[!(0 < First[vsechnoListi[[i]]] < A&& 0 < Last[vsechnoListi[[i]]] < B),
    proces[[i]] = 0,
    Do[
      If[Vzdalenost[vsechnoListi[[i]],vsechnoListi[[j]]] < 2R,
        proces[[i]] = 0;
        Break[]
      ],
      {j,1,i-1}
    ],
    {i,Length[vsechnoListi]}
];
(*smazeme ze seznamu body, které jsme nahradili nulou a vratime vysledek*)
proces = DeleteCases[proces,0];
proces
];

```



Obr 4.3: Simulace dead leaves modelu před a po zředění

4.3 Gibbsovy procesy s fixním počtem bodů

Při simulaci Gibbsových procesů použijeme metody zvané Markov chain Monte Carlo (MCMC). Spočívá ve vytvoření Markovova řetězce. To je proces na množině stavů, který splňuje vlastnost, že v každém stavu procesu je pravděpodobnost navštívení

dalších stavů nezávislá na dříve navštívených stavech. To znamená, že chování v Markovových řetězcích je „bezpaměťové“: V každém konkrétním stavu je možno zapomenout historii (posloupnost stavů předcházející stavu současnému).

Metodu MCMC implementujeme jako obecný algoritmus zrození a zániku (birth-and-death). Pro důkladné pochopení mechanismu čtenářem algoritmus do detailu popíšu. Nejprve rozebereme jednoduchý, ale netriviální příklad a až posléze popíšeme obecný algoritmus zrození a zániku.

Simulace spočívá v opakované modifikaci výchozího bodového procesu, který se mění mazáním (death) a vytvářením nových (birth) bodů. Proceduru opakujeme tak dlouho, dokud algoritmus nakonec nekonverguje a generuje množiny, které můžeme považovat za realizace procesu specifické hustoty.

V našem případě je výchozí konfigurací libovolný bodový proces ve W s n body, pro který hustota dosahuje nezáporné hodnoty. Vlastnosti výchozí konfigurace nejsou podstatné vzhledem k tomu, že se její vliv rychle vytrácí, často přibližně po $10n$ opakováních.

4.3.1 Gibbsův hard-core proces s n body

Nejprve uvažujme speciální případ Gibbsova hard-core procesu s fixním počtem bodů a hard-core vzdáleností r_0 . Počáteční konfigurací může být jakákoliv množina n bodů, ve které nejkratší vzdálenost mezi body není menší než r_0 . Použijeme třeba pravidelnou mřížku n bodů.

Nechť $\{x_1, \dots, x_n\}$ je aktuální stav Markovského řetězce po zopakování l kroků. Potom krok $(l + 1)$ vypadá následovně:

- (1) Náhodně smažeme bod množiny $\{x_1, \dots, x_n\}$ tak, že všechny body z $\{x_1, \dots, x_n\}$ mají stejnou pravděpodobnost smazání $1/n$. Označme smazaný bod třeba x_k .
- (2) Náhodně s rovnoměrným rozdělením uvnitř W vybereme nový bod x . Pokud

$$\min \{\|x - x_i\| : i = 1, 2, \dots, n, i \neq k\} \geq r_0,$$

potom bod x přijmeme a nahradíme jím smazaný bod, takže vznikne nová množina n bodů $\{x_1, \dots, x_{k-1}, x, x_{k+1}, \dots, x_n\}$. Nebo pokud

$$\min \{\|x - x_i\| : i=1, 2, \dots, n, i \neq k\} < r_0,$$

je bod x odmítnut a vybíráme nový bod tak dlouho, dokud není navrhovaný bod přijat. Pokud jsou n a r_0 vysoká čísla, často zabere hledání takového bodu dlouhý čas.

Vytvoříme nejprve univerzální funkci vzhledem k rozměrům okna pro navržení nového bodu, která se nám bude opakovaně hodit.

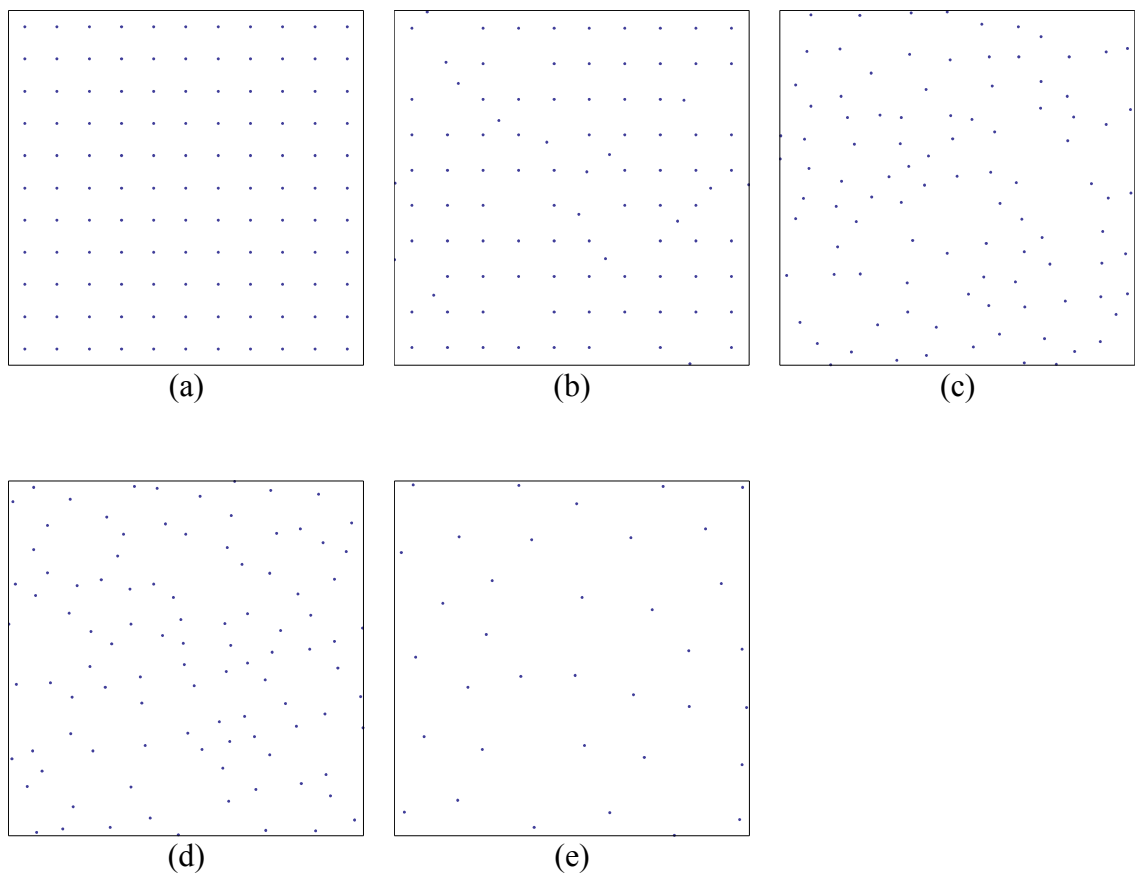
```
NovyBod[A_,B_] := {Random[UniformDistribution[{0,A}]],
                  Random[UniformDistribution[{0,B}]]};
```

Poté můžeme přistoupit k samotnému algoritmu. Jeho vstupními parametry bude výchozí bodový proces x_0 , počet opakování kroků (mohli bychom přednastavit třeba zmíněných $10n$), rozměry okna A , B a r_0 , tedy nejmenší možná vzdálenost mezi dvěma body procesu.

```
GibbsHardCore[x0_,opak_,r0_,{A_,B_}] := Module[{delka,x=x0,pozice,novy,
                                                zamitnut=True,vzdalenosti},
  delka = Length[x];
  Do[
    pozice = RandomInteger[{1,delka}];
    zamitnut = True;
    While[zamitnut,
      novy = NovyBod[A,B];
      vzdalenosti = Table[
        If[i != pozice, Vzdalenost[novy,x[[i]]], 1],
        {i,1,delka}];
      If[Min[vzdalenosti] ≥ r0,
        zamitnut = False;];
    ];
  x[[pozice]] = novy,
  {opak} ]; x];
```


Obrázek 4.2 ukazuje různé bodové procesy získané algoritmem pro $n = 100$ bodů v jednotkovém okně. Hard-core vzdálenost $r_0 = 0,06$. (a) Výchozí stav. Konfigurace po (b) 20 krocích, (c) 1000 krocích a (d) 2000 krocích. Bodové procesy (c) a (d) jsou statisticky podobné a mohou být považovány za vzorky hard-core procesu. (e) Odlišná simulace 2000 kroků pro $n = 36$ bodů s $r_0 = 0,15$.

Podotkněme, že není možné do jednotkového okna vygenerovat model se 100 body a tak velkou hard-core vzdáleností, jako je 0,15. Model se 36 body na obrázku vykazuje okrajový efekt „drift towards the boundary“ (unášení ke kraji): hustota bodů při okrajích okna je nepatrně vyšší než v jeho vnitřku .



Obrázek 4.4: Ukázky simulací Gibbsova hard-core procesu

4.3.2 Obecný algoritmus zrození a zániku (birth-and-death)

Obecný algoritmus použitý při simulování ukázek Gibbsova procesu s konečným počtem bodů má podobnou strukturu jako algoritmus popsany pro speciální případ Gibbsova hard-core procesu výše.

Náhodný bod procesu je smazán a nový bod je vygenerován podle odpovídající hustoty pravděpodobnosti. Simulace začíná výchozím stavem o nezáporné hustotě, který není „v rozporu“ s párovým potenciálem.

Předpokládejme bodový proces po prvních l krocích $\{x_1, \dots, x_n\}$.

- (1) Náhodně smažeme bod množiny $\{x_1, \dots, x_n\}$ tak, že všechny body z $\{x_1, \dots, x_n\}$ mají stejnou pravděpodobnost smazání $1/n$. Označme smazaný bod třeba x_k .
- (2) Nasimulujeme nový bod na základě podmíněné pravděpodobnosti hustoty

$$f_n(x | x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n) = \frac{f_n(x_1, \dots, x_{k-1}, x, x_{k+1}, \dots, x_n)}{\int_W f_n(x_1, \dots, x_n) dx_k}.$$

Nový stav $\{x_1, \dots, x_{k-1}, x, x_{k+1}, \dots, x_n\}$ je výsledkem po $l + 1$ krocích. Tak jako v předešlém Gibbsově hard-core procesu, bod x v části (2) může být nasimulován metodou zamítání: necht' M je horní mez nenormalizované podmíněné hustoty

$$\varphi(x) = \exp\left(-\sum_{j=1, j \neq k}^n \phi(\|x - x_j\|)\right).$$

Nový bod x je vygenerován rovnoměrně ve W společně s nezávislým náhodným číslem u na $\langle 0, 1 \rangle$. Bod x je přijat, jestliže $\varphi(x) \geq Mu$. V opačném případě je navržen nový bod.

Poznamenejme ještě, že pro algoritmus Gibbsova hard-core procesu podle této definice platí $\varphi(x) = 1$ pokud $\|x - x_j\| > r_0$ pro všechna $j = 1, \dots, n, j \neq k$, nula v ostatních případech.

4.3.3 Straussův hard-core proces

Podle birth-and-death algoritmu nyní vytvoříme simulaci *hard-core Straussova procesu*. Ten je určen párovým potenciálem v kapitole 3.1.3. Po úpravě bude nenormalizovaná podmíněná hustota

$\varphi(x)=0$ pokud $\min\|x-x_j\|\leq r_0$, pro ostatní x potom

$$\varphi(x)=\exp\left(-\beta\sum_{j=1,j\neq k}^n\mathbf{1}(r_0<\|x-x_j\|\leq r_{max})\right),$$

kde suma v exponentu udává počet bodů řady $x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n$ se vzdáleností menší než r_{max} od x . Podle definice předpokládáme $\beta > 0$, potom horní mez M můžeme položit rovnu 1. V kroku zrození je bod volený rovnoměrně uvnitř W přijat, pokud $\varphi \geq u$ pro u náhodně zvolené s rovnoměrným rozdělením na $\langle 0, 1 \rangle$.

Nejprve definujeme funkci $\phi(r)$. Všechny parametry se budou nastavovat při volání jako argumenty funkce, *vzdal* je vzdálenost r dvojice bodů.

```
 $\phi[vzdal_, r0_, \beta_, rmax_] := \text{Which}[vzdal \leq r0, \text{Infinity}, vzdal \leq rmax, \beta, vzdal > rmax, 0];$ 
```

Jako výchozí stav našeho procesu můžeme zvolit např. jednoduchý binomický proces pro okno W . Pozice bodů v něm je dána rovnoměrným rozdělením a počet bodů je dán pevně na rozdíl od Poissonova procesu, kde se generuje náhodně. Další parametry, s výjimkou počtu opakování kroků, jsou identické s funkcí ϕ .

```
StraussFixProcess[x0_, beta_, r0_, rmax_, {A_, B_}, opak_] := Module[
```

```
{delka, x = x0, pozice, novy, zamitnut = True, phi, u},
delka = Length[x];
```

```
Do[
```

```
pozice = RandomInteger[{1, delka}];
u = Random[UniformDistribution[{0, 1}]]];
```

```
zमितnut = True;
```

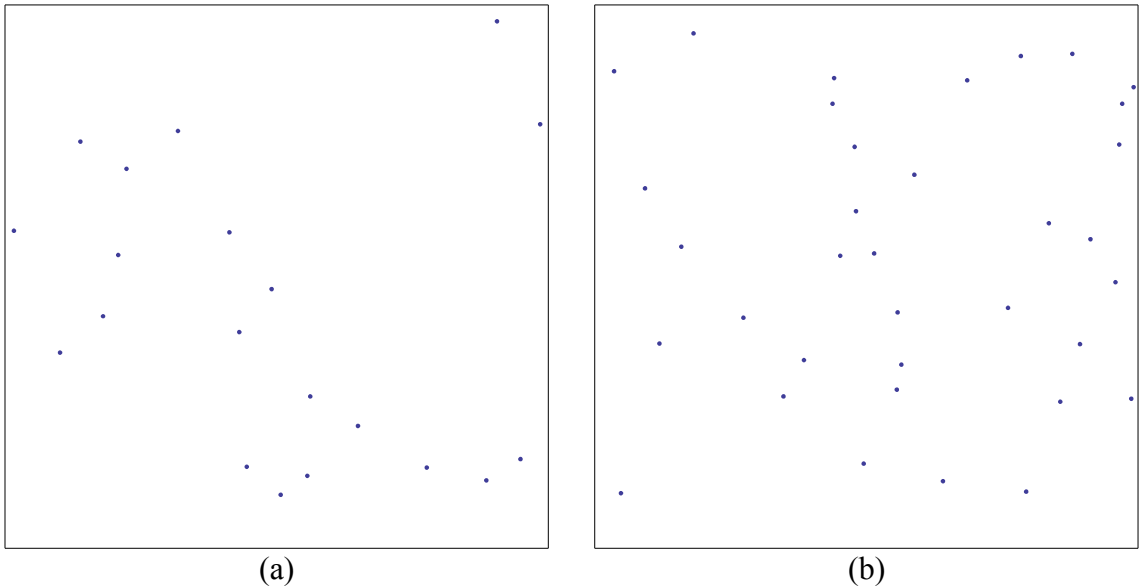
```
While[zमितnut,
```

```

novy = NovyBod[A,B];
φ = Exp[-Sum[If[i ≠ pozice, φ[Vzdalenost[novy, x[[i]]], r0, β, rmax], 0],
        {i, 1, delka}]];
If[ φ ≥ u,
    zamitnut = False];
];

x[[pozice]] = novy,
{opak}
];
x
];

```



Obrázek 4.5: Ukázka simulace Straussova procesu do jednotkového okna pro (a) $n = 20$; $r_0 = 0$; $r_{max} = 0,08$; $\beta = 0,4$. Jedná se vlastně o Straussův proces s fixním počtem bodů (b) hard-core algoritmus $n = 35$; $r_0 = 0,02$; $r_{max} = 0,06$; $\beta = 0,8$.

4.4 Gibbsovy procesy s náhodným počtem bodů

Simulace Gibbsových procesů s náhodným počtem bodů je opět založena na postupu MCMC, stejně jako Gibbsovy procesy s fixním počtem bodů probírané výše. Tudíž simulace opět začíná nějakou výchozí konfigurací spojenou s nezápornou

hodnotou hustoty, která se v průběhu simulace mění. V každém kroku je nějaký bod buď přidán, nebo smazán, nebo se nestane nic.

Úspěšným MCMC algoritmem je algoritmus birth-and-death, modifikovaná verze algoritmu použitého při simulaci Gibbsova procesu s fixním počtem bodů (kapitola 4.3.2). Jinou častou metodou je *Metropolis-Hastings algoritmus* (upouštím od použití „Metropolisův-Hastingsův“), který nyní podrobněji přiblížím.

4.4.1 Metropolis-Hastings algoritmus

Hlavní myšlenka Metropolis-Hastings algoritmu tkví v tom, že při každém kroku je učiněna náhodná nabídka ke změně aktuální konfigurace zrozením či zánikem, která buď může, nebo nemusí být přijata. Pro Gibbsův proces je postup založen na nenormalizované funkci hustoty

$$\varphi(\{x_1, \dots, x_n\}) = \exp\left(-\left(\alpha n + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \phi(\|x_i - x_j\|)\right)\right)$$

pro $x_1, \dots, x_n \in W$, $n = 0, 1, \dots$. Nabídky jsou řízeny následujícími parametry simulace:

- $b(\{x_1, \dots, x_n\})$, pravděpodobnost, že bude navrženo zrození nového bodu, zatímco pravděpodobnost navržení zániku bodu je $1 - b(\{x_1, \dots, x_n\})$
- $q_{\text{birth}}(\{x_1, \dots, x_n\}; x)$, funkce návrhu na umístění nového bodu
- $q_{\text{death}}(\{x_1, \dots, x_n\}; x_k)$, pravděpodobnost navrnutí bodu x_k množiny $\{x_1, \dots, x_n\}$ ke smazání

Simulace začíná bodovým procesem $\{z_1, \dots, z_m\}$ pro který

$$\varphi(\{z_1, \dots, z_m\}) > 0,$$

tzn. který není v rozporu se základními vlastnostmi procesu (např. je to množina bodů s hard-core vzdáleností, pokud požadujeme hard-core proces).

Předpokládejme nyní, že po l krocích je konfigurace $\{x_l, \dots, x_n\}$. V kroku $l+1$ nejprve učiníme rozhodnutí, zda bude do množiny nový bod přidán (*birth*), nebo smazán (*death*), s pravděpodobností $b(\{x_l, \dots, x_n\})$, resp. $1 - b(\{x_l, \dots, x_n\})$.

Pokud je krok $l+1$ zrození, pozice nového bodu $x \in W$ je navržena hustotou $q_{\text{birth}}(\{x_l, \dots, x_n\}; x)$ a návrh je přijat s pravděpodobností

$$\alpha_{\text{birth}} = \min\{1, \rho_{\text{birth}}\},$$

kde

$$\rho_{\text{birth}} = \frac{\varphi(\{x_1, \dots, x_n, x\})(1 - b(\{x_1, \dots, x_n, x\}))q_{\text{death}}(\{x_1, \dots, x_n, x\}; x)}{\varphi(\{x_1, \dots, x_n\})b(\{x_1, \dots, x_n\})q_{\text{birth}}(\{x_1, \dots, x_n\}; x)}$$

je tzv. Metropolis-Hastings birth poměr. Tímto poměrem společně s odpovídajícím poměrem zániku je řízen Markovův řetězec tak, že jeho stacionární rozdělení odpovídá rozdělení požadovaného bodového procesu. Pokud je nabídka přijata, nový bod je přidán do konfigurace. Pokud nabídka není akceptována, konfigurace se v tomto kroku nemění.

Pokud je krok $l+1$ zánik a $\{x_l, \dots, x_n\}$ není prázdná množina, bod $x_k \in \{x_l, \dots, x_n\}$ je navržen ke smazání s pravděpodobností $q_{\text{death}}(\{x_l, \dots, x_n\}; x_k)$. Toto smazání je přijato s pravděpodobností

$$\alpha_{\text{death}} = \min\{1, \rho_{\text{death}}\},$$

kde

$$\rho_{\text{death}} = \frac{\varphi(\{x_1, \dots, x_n\} \setminus \{x_k\})b(\{x_1, \dots, x_n\} \setminus \{x_k\})q_{\text{birth}}(\{x_1, \dots, x_n\} \setminus \{x_k\}; x_k)}{\varphi(\{x_1, \dots, x_n\})(1 - b(\{x_1, \dots, x_n\}))q_{\text{death}}(\{x_1, \dots, x_n\}; x_k)}$$

je Metropolis-Hastings death poměr, symbol \setminus značí rozdíl množin, $k \in \{1, \dots, n\}$. Je-li nabídka přijata, bod x_k je z konfigurace odstraněn. Pokud nabídka přijata není, konfigurace se v tomto kroku nemění.

Poznamenejme, že ve vzorci ρ_{birth} platí

$$\frac{\varphi(\{x_1, \dots, x_n, x\})}{\varphi(\{x_1, \dots, x_n\})} = \lambda(x | \{x_1, \dots, x_n\}),$$

a pro ρ_{death}

$$\frac{\varphi(\{x_1, \dots, x_n\} \setminus \{x_k\})}{\varphi(\{x_1, \dots, x_n\})} = \frac{1}{\lambda(x_k | \{x_1, \dots, x_n\} \setminus \{x_k\})}$$

Jinými slovy, podmíněná intenzita formuje integrační část Metropolis-Hastings algoritmu.

Volba prostých konkrétních parametrů algoritmu může vypadat následovně:

$$b(\{x_1, \dots, x_n\}) \equiv \frac{1}{2}$$

$$q_{\text{birth}}(\{x_1, \dots, x_n\}; x) = \frac{1}{v(W)} \text{ pro } x \in W$$

$$q_{\text{death}}(\{x_1, \dots, x_n\}; x_k) = \frac{1}{n} \text{ pro } x_k \in \{x_1, \dots, x_n\}$$

Zkušenosti ukazují, že ladění algoritmu zvyšuje jeho efektivitu, tj. konverguje ke stacionárnímu stavu dříve a korelace mezi následnými vzorky klesá rychleji.

Jakýkoliv MCMC algoritmus, a tedy i birth-and-death a Metropolis-Hastings algoritmus, generuje modely požadovaného procesu pouze až po jistém počtu opakování, po nějaké periodě zvané *burn-in*, jejíž délka není předem jasná. To znamená, že uživatel musí simulaci procesu sledovat a rozhodnout, zda algoritmus „dokonvergoval“. Možnými indikátory jsou stávající počet bodů nebo energie $U(\{x_1, \dots, x_n\})$. Objasnění pojmů lze vyčíst z obrázku 4.8.

4.4.2 Straussův proces s náhodným počtem bodů

Straussův proces s parametry $\alpha = -8,0$; $\beta = \exp(0,3) = 1,35$; a $r_l = 0,08$ simulujeme do jednotkového okna. Parametry Metropolis-Hastings algoritmu jsou zvoleny jako výše uvedené.

Předpokládejme, že konfigurace po l krocích je $\{x_1, \dots, x_n\}$. Nejprve uděláme rozhodnutí, zda přidávat, nebo mazat bod, obojí s pravděpodobností $\frac{1}{2}$. Pokud má být přidán bod ke konfiguraci, kandidát je vybrán rovnoměrně uvnitř W . Hodnotu Metropolis-Hastings poměru zrození po zjednodušení spočteme jako

$$\rho_{\text{birth}} = \frac{1}{n+1} \exp\left(-\alpha - \beta \sum_{i=1}^n \mathbf{1}(0 < \|x - x_i\| \leq r_1)\right).$$

Navrhovaný bod x je přijat s pravděpodobností

$$\alpha_{\text{birth}} = \min\{1, \rho_{\text{birth}}\}.$$

Zvolíme tedy náhodné číslo u na $\langle 0, 1 \rangle$ a pokud $u \leq \alpha_{\text{birth}}$, navržený bod x je přijat a nová konfigurace je $\{x_1, \dots, x_n, x\}$. Pokud $u > \alpha_{\text{birth}}$, původní konfigurace $\{x_1, \dots, x_n\}$ se nezmění. Podobný mechanismus uplatníme i u rozhodování, zda další krok bude zrození, nebo zánik.

Pokud bod mažeme, vybereme náhodně číslo k z množiny $\{1, \dots, n\}$ a navrhneme bod x_k k odebrání. Po zkrácení Metropolis-Hastings dea th poloměr

$$\rho_{\text{death}} = n \exp\left(\alpha + \beta \sum_{i=1, i \neq k}^n \mathbf{1}(0 < \|x_k - x_i\| \leq r_1)\right)$$

a $\alpha_{\text{death}} = \min\{1, \rho_{\text{death}}\}$. Nabídka je přijata, tzn. bod x_k se smaže s pravděpodobností α_{death} stejně jako výše. Pokud je přijata, nová konfigurace je $\{x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n\}$, jinak se nezmění.

V našem příkladu nastavíme výchozí konfiguraci jako realizaci binomického procesu se 120 body. Pro další zkoumání si budeme po každém kroku ukládat i aktuální počet bodů do seznamu *hist*.

```
BinP[n_, {A_, B_}] := Table[{Random[UniformDistribution[{0, A}]],
                             Random[UniformDistribution[{0, B}]]}], {n}];
```

```
phi[vzdal_, r0_] := If[0 < vzdal <= r0, 1, 0];
```

```
StraussRandom[alpha_, beta_, r1_, {A_, B_}, opak_] := Module[
  {akce, x, delka, pozice, novy, rhoBirth, rhoDeath, u, hist},
  x = BinP[n, {A, B}];
  hist = {Length[x]};
```

```
Do[
```



```

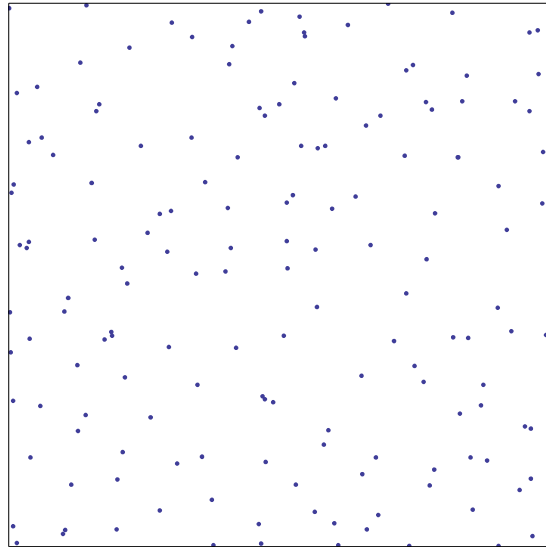
(*rozhodujeme o typu akce - 1 birth, 0 death*)
akce = RandomInteger[];
delka = Length[x];
u = Random[UniformDistribution[{0,1}]];
If[ akce > 0,
  novy = NovyBod[A,B];
   $\rho_{\text{birth}} = \text{Exp}[-\alpha-\beta*\text{Sum}[\phi[\text{Vzdalenost}[\text{novy},x[[i]]],r1],\{i,1,\text{delka}\}]]$ 
                                                                    /(\text{delka}+1);

  If[  $\rho_{\text{birth}} \geq u$ ,
    x = Append[x,novy]
  ],
  pozice = RandomInteger[{1,\text{delka}}];
   $\rho_{\text{death}} = \text{delka}*\text{Exp}[\alpha+\beta*\text{Sum}[\text{If}[i\neq\text{pozice},\phi[\text{Vzdalenost}[x[[\text{pozice}]],x[[i]]],r1],0],\{i,1,\text{delka}\}]]$ ;
  If[  $\rho_{\text{death}} \geq u$ ,
    x = Delete[x,pozice]];
  ];
hist = Append[hist,Length[x]],
{opak}
];
{x,hist}
];

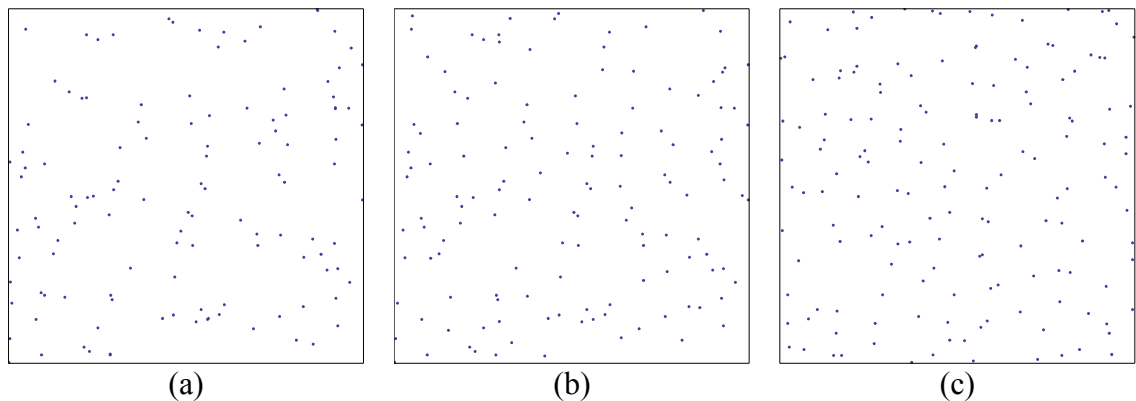
```

Obrázek 4.7 ukazuje simulaci ve výchozím stavu, po 100 a 5000 opakování, kde náhodné počty bodů jsou 120, 132 a 154. Na obrázku 4.6 je stejná simulaci po 10 000 opakování.

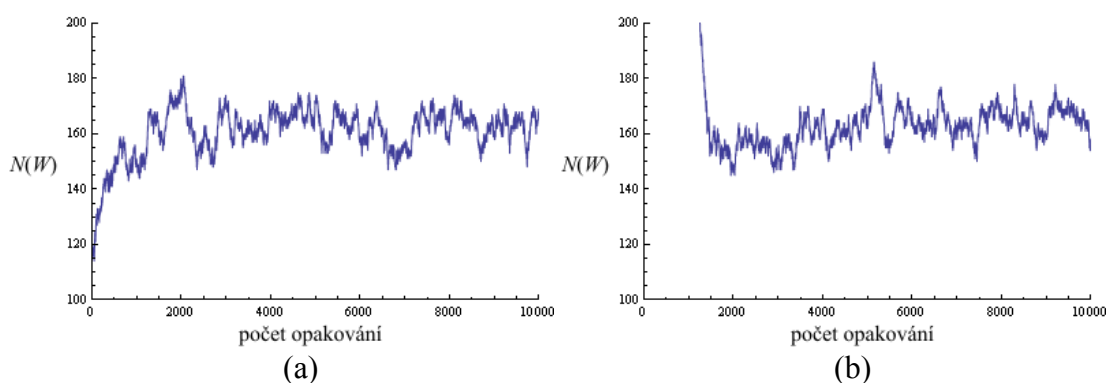
Obrázek 4.8 znázorňuje vývoj počtu bodů $N(W)$ algoritmu během 10 000 kroků s počátečními hodnotami $n = 120$ a $n = 800$. V tomto konkrétním případě *burn-in* trvá zhruba 2000 opakování, po nichž mohou být vygenerované množiny bodů považovány za realizace Straussova procesu. Počty bodů množin mezi kroky 2001 a 10 000 mohou být použity k vyšetřování rozdělení $N(W)$. Samozřejmě je to náhodná veličina. Histogram takového rozdělení je vidět na obrázku 4.9.



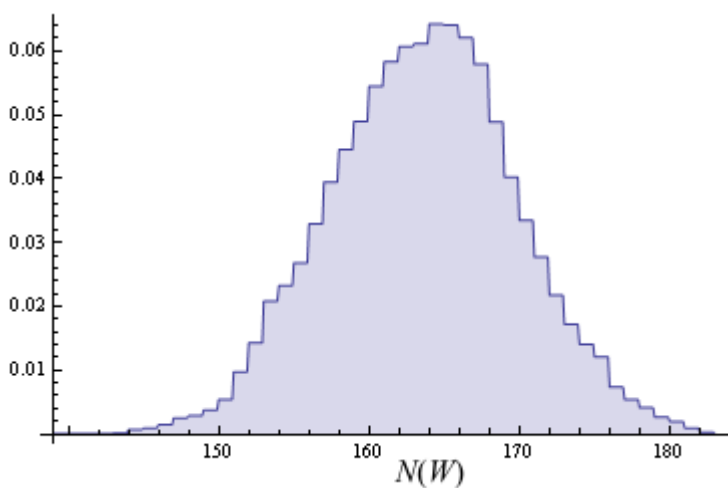
Obrázek 4.6: Výsledek simulace Straussova procesu s náhodným počtem bodů pomocí Metropolis-Hastings algoritmem do jednotkového okna po 10 000 opakování ($n = 169$). Parametry procesu a implementace algoritmu jsou popsány v textu.



Obrázek 4.7: Průběžné stavy simulace z obrázku 4.6 (a) Výchozí konfigurace ($n = 120$). Konfigurace po (b) 100 krocích ($n = 132$) a (c) 5000 krocích ($n = 154$). Množina (c) spolu s množinou na obrázku 4.6 jsou statisticky podobné a mohou být považovány za *sample* Straussova procesu. Můžeme odhalit tendenci k pravidelnosti, ale blízké páry jsou také dovoleny.



Obrázek 4.8: Grafické znázornění vývoje počtu bodů $N(W)$ v 10 000 krocích Metropolis-Hastings algoritmu (a) popsaného v textu ($n = 120$) (b) se stejnými parametry mimo počtu bodů počátečního stavu ($n = 800$). Můžeme pozorovat, že se „hladiny obou případu srovnají“.



Obrázek 4.9: Histogram rozdělení náhodného počtu bodů $N(W)$ popsaného algoritmu odvozený ze simulací po 2000 krocích. Vyčíslená střední hodnota $\mathbf{E}(N(W))$ konkrétního příkladu je 163,15.

5 Odhady

Počítačové simulace dnes ve statistice hrají důležitou roli. Ve statistice bodových procesů mají dlouhou tradici a už jednoduché otázky často vyžadují jejich využití. Vizualizace náhodných vzorků nabízí lepší pochopení prostorových struktur, které mohou být generovány z jednotlivých modelů. K analýze dat slouží několik známých metod, které charakterizují vztahy uvnitř množin a dovolují srovnávat data naměřená v reálném světě s pravděpodobnostními modely. Ke splnění daného cíle je většinou potřeba nalézt odpovídající model.

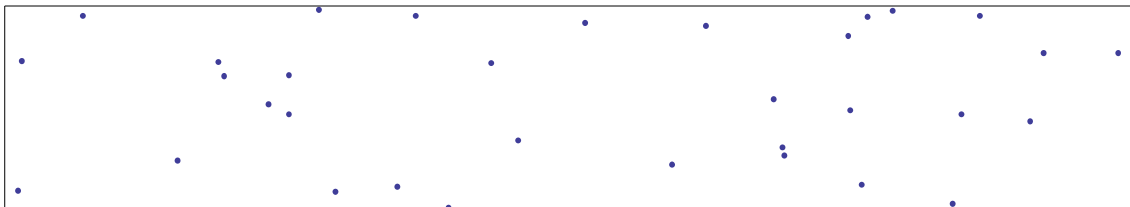
V této kapitole zmíním jen okrajově některé souhrnné statistiky pro bodový proces a postupy pro odhad jeho parametrů. Zkusím analyzovat vhodná reálná data z terénu a odhadnout parametry odpovídajícího Straussova modelu. Problém zpracování a analýzy naměřených hodnot je velmi široký a s rostoucí náročností na přesnost také velmi obtížný. V rámci přijatelného rozsahu práce se pokusím nabídnout aspoň zjednodušený postup bez přesných definic, jehož cílem je nalézt k daným datům odpovídající Straussův proces.

5.1 Získání a prvotní zpracování dat

Základním předpokladem každé analýzy jsou potřebná data. Ta většinou představují odborníkem v daném oboru zakreslené pozice objektů, které jsou předmětem zkoumání. Mohou být pořízena z leteckých snímků, mikroskopem, měřením atp. V našem případě máme množinu bodů, které představují pozice ryb zachycené sonarem z lodi jedoucí po hladině nádrže. Šířka pásma je 10m a délka 58m.

Nemáme k dispozici žádné doplňující informace, takže se omezíme pouze na nejběžnější statistiky bodových procesů, nebudeme brát v úvahu žádné okolnosti, které by mohly nějak ovlivňovat modely (kóty jako velikost, hloubka aj.) a budeme předpokládat, že se jedná o *homogenní* bodový proces. To zjednodušeně znamená, že pozorovací okno W považujeme za malou část mnohem větší množiny se stejnými vlastnostmi. Jinak: za hranicemi W vypadá rozmístění bodů obdobně, jako uvnitř.

Na obrázku 5.1 je vybrané okno W z dostupných dat. Pouhým okem můžeme vidět mezi rybami hard-core vzdálenosti a můžeme se domnívat, že jde o realizaci Straussova procesu s náhodným počtem bodů. Nejprve se zaměříme se na odhad funkce párového potenciálu.



Obrázek 5.1: Pás ryb zaznamenaný sonarem.

5.2 Funkce K a L

Pro diagnostiku bodového procesu se využívají tzv. souhrnné charakteristiky. Ty pomocí čísel, funkcí a schémat různým způsobem popisují vzdálenosti mezi body a jejich relativní pozice. Zaměříme se na charakteristiky druhého řádu, které hrají v analýze bodových procesů hlavní roli. Použití těchto charakteristik nám pak poskytne porovnání původní množiny se simulací.

K -funkce udává průměrný počet bodů do vzdálenosti r od typického bodu v poměru s intenzitou procesu. Neboli do každého bodu procesu se umístí střed kružnice o poloměru r a spočítá se počet sousedních bodů uvnitř této kružnice. Odhad funkce K upravené pro Gibbsovy procesy zavedeme jako

$$\hat{K}(r) = \frac{2 \cdot v(W)}{n(n-1)} n_2(r),$$

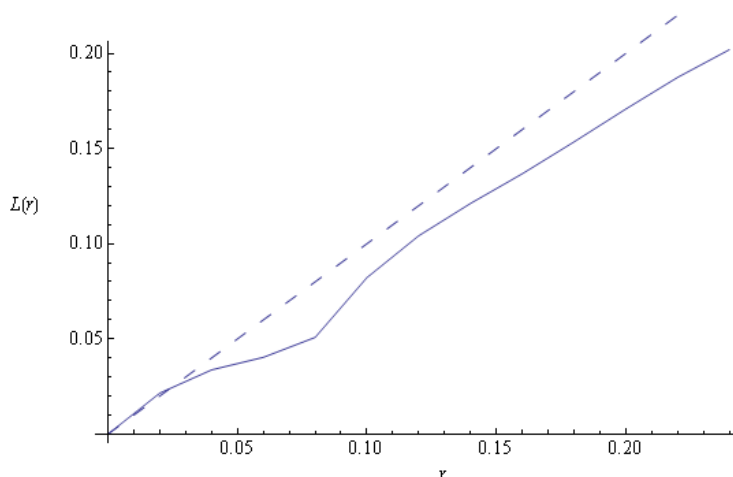
kde

$$n_2(r) = \frac{1}{2} \sum_{i=1}^n \sum_{j=i+1}^n \mathbf{1}(\|x_i - x_j\| \leq r) \quad \text{pro } r > 0.$$

Stejně informace jako K -funkce podává její analogická funkce L definovaná jako

$$\hat{L}(r) = \sqrt{\frac{\hat{K}(r)}{\pi}}.$$

Toto vyjádření je pro nás snadněji interpretovatelné, protože v případě ideálu Poissonova procesu (který je pro nás základním srovnávacím vzorem) tvoří přímou úměru, zatímco funkce K parabolou. Obrázek 5.2 ukazuje průběh L -funkce pro Straussův proces z obrázku 4.6. V tomto procesu není hard-core vzdálenost, takže začíná růst od $r = 0$. Můžeme si také všimnout jejího vrcholu (*cusp point*) v $r = 0,08$.



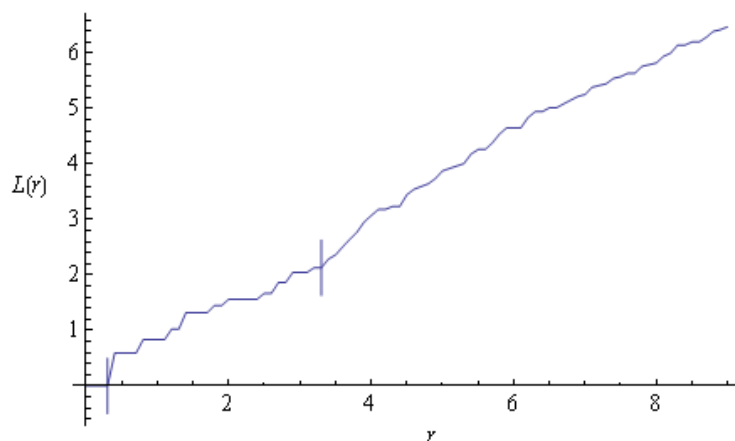
Obrázek 5.2: L -funkce pro Straussův proces s parametry $\alpha = -8,0$; $\beta = 1,35$; a $r_l = 0,08$ pro $r < R_{max} = 0,25$. (plná čára) v porovnání s grafem přímé úměry (přerušovaná).

Zjednodušeně se dá říct, že když na nějakém intervalu graf funkce L roste rychleji než graf přímé úměry, vytváří v těchto vzdálenostech shluky, pokud roste pomaleji, body se spíše odpuzují.

5.3 Odhady

Když umíme vytvořit graf L -funkce, necháme si ho zobrazit pro naše okno W . Pokud připustíme, že se jedná o ukázkou Straussova hard-core procesu, hledáme nejprve dva parametry funkce párového potenciálu r_0 a r_{max} .

Jejich odhad ukazuje obrázek 5.3. Hard-core vzdálenost $r_0 = 0,3m$ je jednoduše nejmenší naměřená vzdálenost mezi body. Zlomový vrchol, vzdálenost r_{max} do níž jsou body od sebe částečně odpuzovány, odhadneme z grafu na hodnotu 3,3m. Protože na delší vzdálenosti spolu už body neinteragují, není potřeba sledovat hodnoty L -funkce pro mnoho větších poloměrů.



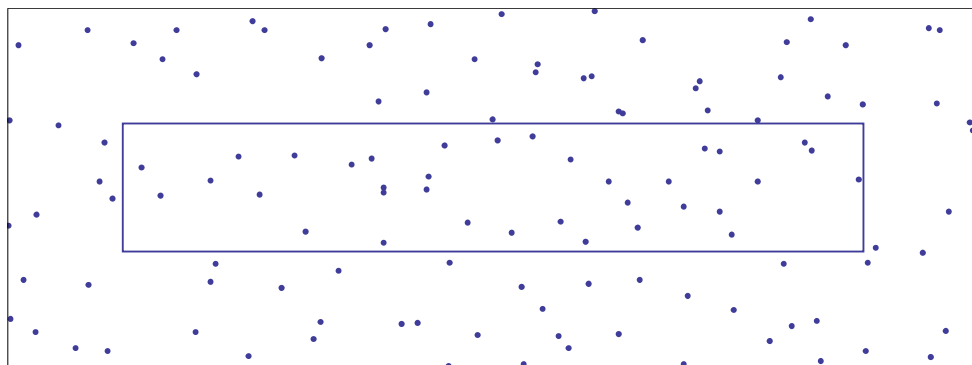
Obrázek 5.3: L -funkce pro zadanou množinu ryb. Hard-core vzdálenost a *cusp point* jsou označeny svislou úsečkou.

Když máme tyto dva parametry, necháme si hodnotu β aproximovat dosazením do následujícího vzorce (Ogata a Tamura, 1981). Pro naše zadání vychází 0,900745.

$$\hat{\beta}_{aprox} = -\ln \left(\frac{n_2(r_{max})(v(W) - \pi(r_{max}^2 - r_0^2))}{\pi(r_{max}^2 - r_0^2) \left(\frac{1}{2}n(n-1) - n_2(r_{max}) \right)} \right).$$

Poslední parametr našeho Straussova procesu α , který ovlivňuje hustotu bodů, není snadné předem určit a odhadneme ho až na základě pozdějších simulací. Než začneme se samotnou simulací, musíme mít na paměti, že náš model může být ovlivněn „okrajovým efektem“ (viz obrázek 4.2). Existuje mnoho metod (*edge-correction methods*), jak se s tímto problémem vyrovnat a vystačily by na samostatnou práci.

Použijeme jedno ze snadnějších řešení, kterým je simulace do rozšířeného okna. Necháme probíhat celý proces do okna rozšířeného o vzdálenost, která je větší než vzdálenost interakcí r_{max} . Zvolíme třeba stejnou hodnotu, pro kterou sledujeme průběh L -funkce. Za modely procesu pak budeme považovat množiny v okně s rozměry původního W ve středu rozšířeného okna, jak je znázorněno na obrázku 5.4.



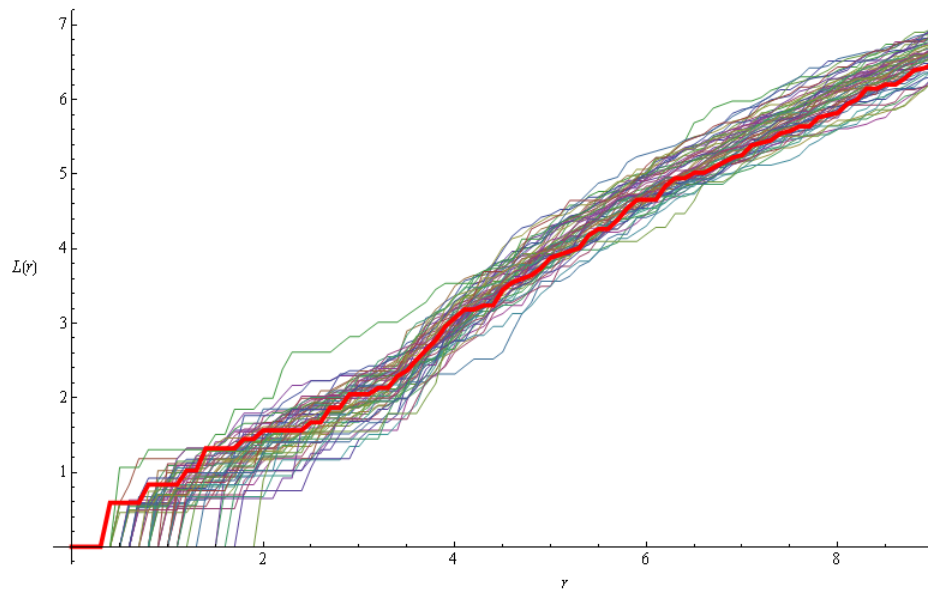
Obrázek 5.4: Znázornění metody simulace do rozšířeného okna.

Obecně čím nižší parametr α pro Straussův proces volíme, tím hustěji se nám W zaplní. Zdůrazněme, že může nabývat také záporných hodnot. Můžeme tedy spustit první simulace s libovolnou hodnotou α a vyčíslíme si jejich průměrnou intenzitu. Postupně pak α upravujeme, dokud se neblíží průměrná intenzita simulací k intenzitě zadané množiny. Touto metodou odhadneme zbývající parametr $\alpha = -6,15$.

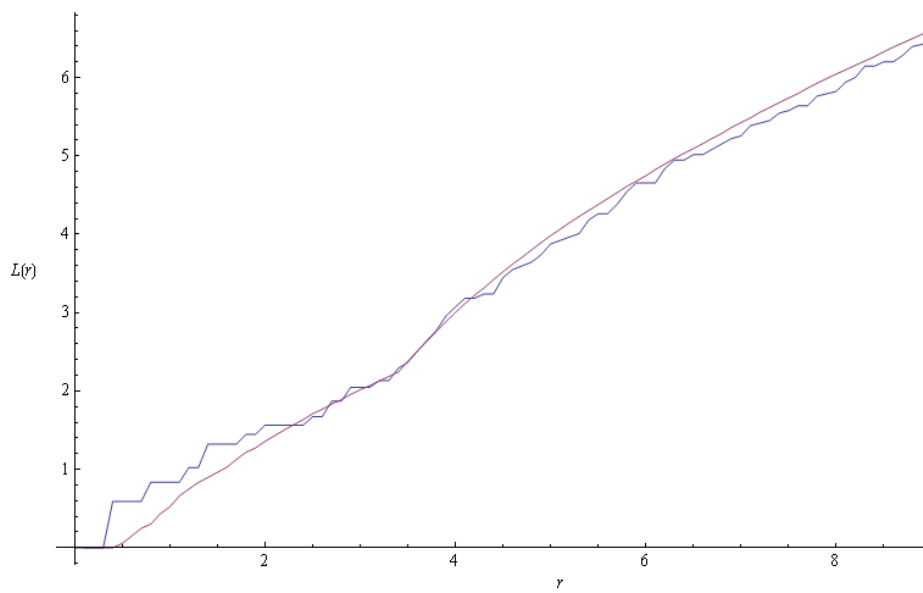
Nyní můžeme nasimulovat mnoho vzorků našeho procesu a porovnáme jejich výsledky s původní množinou. Na obrázku 5.5 je zobrazeno barevně 60 L -funkcí pro různé množiny procesu společně s původní. Můžeme na něm pozorovat, v jakém rozmezí se náš model vychyluje od původní množiny. Podobnou informaci poskytne také histogram hustoty všech bodů těchto funkcí.

Vytvoříme si ještě graf L -funkce, jejíž hodnoty budou určeny jako střední hodnota všech nasimulovaných množin. Porovnáme-li ji důkladně s původní množinou, všimneme si u původní množiny ryb většího množství shluků do vzdálenosti $2m$ oproti naší zprůměrované funkci.

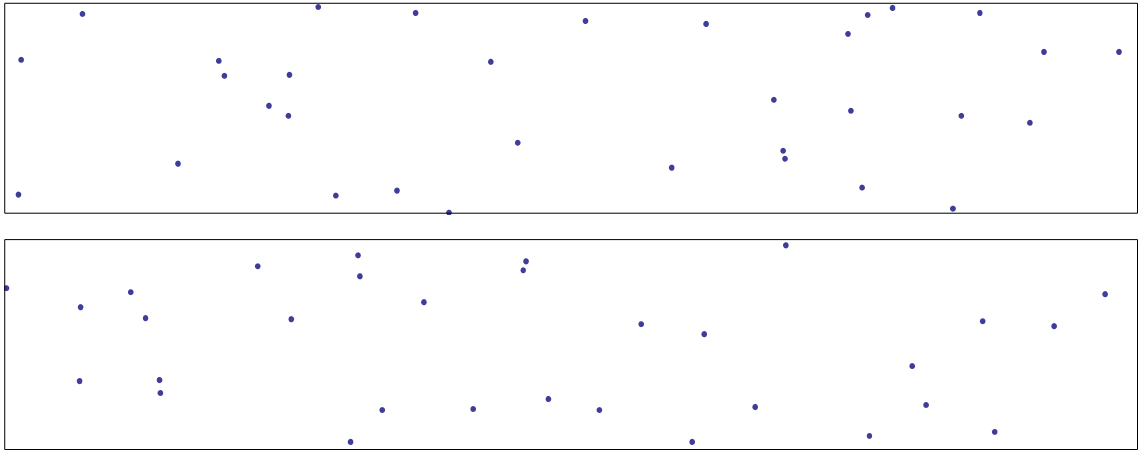
Spokojíme se s pozorováním, že průměrná realizace odhadnutého Straussova procesu se s jistými odchylkami podobá původní množině. Patrné je zejména společné zalomení kolem hodnoty r_{max} . Dodejme, že by bylo pro srovnání vhodné použít i další charakteristiky ($g, F, J...$).



Obrázek 5.5: Srovnání 60 simulací s původní množinou ryb (silnější čarou).



Obrázek 5.6: Graf funkce L pro množinu ryb (modrá barva) a funkce složená ze středních hodnot grafů pro 100 simulací (fialová).



Obrázek 5.7: Porovnání konfigurace náhodně vybrané simulace (nahore) s původní množinou ryb (dole).

6 Závěr

Na tomto místě bych rád shrnul, nakolik se podařilo naplnit cíle práce, kde jsou její rezervy a některé postřehy nabyté při její tvorbě.

Hlavním cílem bylo vytvoření balíku funkcí pro simulaci vybraných procesů. Tento úkol se podařilo bez výjimky naplnit. S balíkem je skutečně možno vytvářet realizace všech uvedených procesů, libovolně nastavovat všechny jejich parametry a k dispozici je několik užitečných funkcí pro zobrazení výsledků, sledování „chování“ některých vytvářených procesů a analýzu bodových procesů v rovině pro obdélníková pozorovací okna.

Do budoucna by šlo s využitím nově nabytých znalostí a zkušeností některé algoritmy dále modifikovat, a zvýšit tak jejich efektivitu, např. lepším zpracováním seznamů nebo uchováváním různých hodnot pro snížení počtu kalkulací.

Druhým stěžejním cílem bylo, aby text mohl co nejlépe sloužit jako odrazový můstek pro další zájemce o studium této oblasti matematiky. Snažil jsem proto psát text co nejjednodušeji a srozumitelně, komentovat klíčová místa ve zdrojových kódech a vkládat názorné příklady a obrázky. Pokusil jsem se do výkladu také vložit postupy a pojmy, které mi pomohly pochopit některé myšlenky a vztahy, na které jsem musel sám přijít.

V průběhu psaní jsem se rovněž nemohl ubránit úvahám, jak by se daly nově nabyté poznatky využít z profesního hlediska učitele na SŠ. Program WM si své místo v praxi jistě najde. Nechci pouštět uzdu někdy příliš naivním fantaziím, jak by se dala matematika moderně úspěšně vyučovat. Ani nemyslím, že bych přišel na nový zaručený způsob jak donutit žáky myslet. Sám na sobě jsem ale pozoroval, jak i jednoduchá simulace dokáže vyvolat napětí a očekávání. Tedy pocity vnitřní motivace k utváření pojmů potřebných k dosažení cíle.

Literatura

ILLIAN, Janine, Antti PENTINEN, Helga STOYAN a Dietrich STOYAN. *Statistical analysis and modelling of spatial point patterns*. Hoboken, NJ: John Wiley, c2008, 534 s. ISBN 04-700-1491-1.

OGATA, Y. a M. TANEMURA. *Estimation of interaction potentials of spatial point patterns through the maximum likelihood procedure*. Ann. Inst. Statist. Math., 1981

PANTLÍKOVÁ, Radka. *Simulace náhodných množin pomocí počítače*. České Budějovice, 2009. Diplomová práce. JČU.

STOYAN, Dietrich a Antti PENTTINEN. INSTITUTE OF MATHEMATICAL STATISTICS. *Recent Applications of Point Process Methods in Forestry Statistics* [online]. No. 1. Statistical Science, 2000 [cit. 2012-04-25]. Dostupné z: <http://www.jstor.org/stable/2676677>