

Posluchači: **Daniel KARLÍK, Michaela MAŠKOVÁ**Zadání: **Floor-field model se statickým polem**

Odevzdáno:

Získané body:

Finální: ANO/NE

## 1 Celulární modely

Nejdříve si popíšeme proč je výhodné se *celulárními modely* zabývat a jaké jsou jejich možné aplikace. Následně na to navážeme jejich formálním matematickým zavedením.

Celulární modely nám umožňují popsat a studovat situace, kdy máme systém několika částic, které mají tendenci pohybovat se nějakým daným směrem a zároveň spolu interagují. Celulární modely se tedy sestávají z konfigurace poloh částic (pracujme např. s  $N$  částicemi)  $x = (x_1, \dots, x_N)$  a interakcí, které jsou popsány potenciálem  $U(x)$ .

Tento typ modelů má široké spektrum využití, ať už ve studiu dopravních proudů nebo třeba jiných fyzikálních modelech. V historii se objevili i tací badatelé, kteří se zabývali filozofickými otázkami typu zda není i náš vesmír jistým celulárním systémem. Tyto otázky však přenecháme povolanějším a budeme se zabývat jednoduchými systémy.

Důležitou součástí celulárních modelů je definice okolí buněk, jelikož se budeme zabývat buňkami ve 2D mřížce, tak si zadefinujeme okolí právě pro tyto případy.

**Okolím buňky**  $x \in \mathbb{L}$ , kde  $\mathbb{L}$  značí množinu buněk, budeme rozumět

$$\mathbb{N}_x = \{y \in \mathbb{L} : \text{dist}(x, y) \leq d\} \quad (1)$$

Ve 2D tedy definujeme pro  $\forall x, y \in \mathbb{L}, x = (x_1, x_2), y = (y_1, y_2)$

**Von Neumannovo okolí** jako výše zmíněné okolí v případě, kdy bereme

$$\text{dist}(x, y) = |x_1 - y_1| + |x_2 - y_2| \wedge d = 1 \quad (2)$$

**Moorovo okolí** bude okolí splňující následující

$$\text{dist}(x, y) = \max\{|x_1 - y_1|, |x_2 - y_2|\} \wedge d = 1 \quad (3)$$

Obecně však může pohyb částic probíhat na nějaké množině buněk, reprezentované nějakým polem libovolné dimenze. Od tvaru této množiny se pak odvíjí možná volba okolí buněk. Zároveň zde vyvstává otázka, jak samotný pohyb částic probíhá.

Podívejme se na to, jakým způsobem mohou být částice v množině buněk rozptýleny. Neboť chceme simulovat modely ve kterých neuvažujeme nehmotnost částic (uvažujeme např. chodce), pak se v jeden čas může v jedné buňce vyskytovat maximálně jedna částice. Tedy dostáváme množinu možných stavů  $\mathbf{S} = \{0, 1\}$  a stav buňky  $x \in \mathbb{L}$  splňuje následující vztah

$$\tau(x) = \begin{cases} 0, & x \text{ je prázdné} \\ 1, & x \text{ je obsazené} \end{cases} \quad (4)$$

Částice v modelu se vždy mohou pohybovat pouze do neobsazených buněk svého okolí nebo setrvat ve stávající buňce. Neboť budeme pracovat v diskrétním čase, pak připadá pohyb částice na jednu časovou jednotku. Pro naše účely budeme předpokládat, že  $v_{\max} \leq 1$ , tzn. že se částice může pohnout za jednu časovou jednotku maximálně do okolní buňky, kdybychom ji volili vyšší pak by se náš model značně zkomplikoval.

## ZMÍNIT TASEP?

Pohyb částic v modelu představuje změnu stavu modelu. Takovou změnu modelu budeme nazývat *update*. Obecně je možné *update* stavu systému rozdělit na dva typy. První představuje, tzv. *paralelní update*, který představuje, že pohyb všech částic probíhá najednou. Druhý typ můžeme nazývat např. *postupný update* a označuje update ve kterém probíhá pohyb částic jednu po druhé. Takový pohyb probíhá podle nějak zvoleného pořadí. Užitím paralelního updatu může dojít ke konfliktu mezi částicemi a to takovému, že se více částic se chce přesunout do stejné buňky. To lze vyřešit tím, že náhodně vybereme jednu z částic, která se na onu buňku přesune.

## 2 Floor-field model

V této kapitole se blíže podíváme na tzv. *Floor-field model*, jedná se o konkrétní typ modelu spadající do skupiny *celulárních modelů*. Vyznačuje se tím, že se jedná o model na dvou-dimenzionální mřížce. Tuto dvoudimenzionální mřížku překryjeme polem, pomocí kterého budeme moci daný model dále studovat. Pro *Floor-field model* je obvyklé, že se pracuje s dvěma různými poli. První je tzv. *statické* a druhé *dynamické*. Statické pole představuje pole, které se vyčíslí na počátku daného pokusu a nadále nedochází k jeho změnám. Může se například jednat o vzdálenost od atraktoru, např. únikový východ pro evakuující. Oproti tomu dynamické pole se po každém proběhnutém kroku musí aktualizovat, jako příklad můžeme uvést stopy označující cestu, kterou se vydal nějaký jedinec v minulosti. V takovém případě většinou platí jednoduchý vztah, čím čerstvější stopa tím silnější.

Pravděpodobnost přechodu ze stavu systému  $x$  do stavu  $y$  označíme intuitivně  $P(x \rightarrow y)$ . Bude pro ně platit vztah

$$P(x \rightarrow y) \propto \exp \left\{ - \sum_F k_F F(y) \right\} \quad (5)$$

kde  $F$  značí pole a  $k_F \in \langle 0, +\infty \rangle$  je parametr.

Statické pole budeme obecně značit  $S$ . Hodnoty obsažené v polích závisí na zvoleném tvaru okolí buněk, pro naše účely postačí dvojice již dvě definovaná okolí **Von Neumannovo** (2) a **Moorovo** (3).

Máme-li floor-field model, ve kterém parametr  $k_F = k_S \rightarrow 0$ , pak pole statické pole  $S$  nehraje v modelu roli.

## 3 Kód

Pro běh simulace bylo třeba vytvořit několik pomocných funkcí, které umožnily například najít souřadnice buněk v Neumannově okolí referenční buňky, vypočítat statické pole apod. Byla použita mřížková metrika  $\text{dist}(x, y) = |x_1 - x_2| + |y_1 - y_2|$ .

Pro simulaci byla vytvořena dvě pole jako matice. Matice `pole` v sobě uchovává informaci o překážkách (-1), částicích (+1) a volných buňkách (0). Matice `staticke_pole` pak uchovává informaci o překážkách a ve zbylých políčkách jsou hodnoty představující vzdálenost daného políčka od atraktoru.

### 3.1 Pomocné funkce

Funkce `dist` definuje vzdálenost mezi dvěma buňkami. Funkce `neumann_okoli` pak vrací hodnotu `true` v případě, že se buňka nachází v Neumannově okolí referenční buňky a `false` v opačném případě.

```

1 function dist(b1, b2)
2     x1, y1 = b1
3     x2, y2 = b2
4     abs(x1 - x2) + abs(y1 - y2)
5 end
6
7 function neumann_okoli(b1, b2)
8     dist(b1, b2) == 1
9 end

```

Následují funkce pro vyplnění statického pole vzdálenostmi od atraktoru. Funkce `dist_pole(pole,b)` vytvoří „masku“, která nám označí hodnotou `true` ty prvky matice, které jsou v Neumannově okolí buňky `b`. Funkce `udpate_pole(pole,b,max_value)` pak tuto masku použije k přičtení 1 do statického pole na správném místě. Obě funkce jsou iterativně použity v poslední funkci `spocitej_S`, která pomocí `while` cyklu iterativně vyplňuje statické pole, dokud není zcela zaplněno.

```

1 function dist_pole(pole, b)
2     max_value = maximum(pole)
3     tmp = zeros(n, n)
4     for x in 1:n
5         for y in 1:n
6             if neumann_okoli((x, y), b) && (pole[x,y] == 0)
7                 tmp[x,y] = 1
8             end
9         end
10    end
11    mask = tmp .== 1
12 end
13
14 function update_pole(pole, b, max_value)
15     mask = dist_pole(pole, b)
16     pole[mask] .= max_value + 1
17     return pole
18 end
19
20 function spocitej_S(pole, atraktor)
21     max_value = maximum(pole)
22     update_pole(pole, atraktor, max_value)
23     while sum(pole .== 0) > 0
24         max_value = maximum(pole)
25         indexes = findall(x -> x == max_value, pole)
26
27         for b in Tuple.(indexes)
28             update_pole(pole, b, max_value)
29         end
30     end
31 end

```

Tím jsme schopní spočítat statické pole. Teď už potřebujeme jen novou pozici částic v

našem systému. Částice přeskochí do políčka ve svém okolí s pravděpodobností definovanou vztahem (2). Funkce `preskok(pole,ks)` nejdříve najde souřadnici, kde se nachází částice. Pak vytvoří masku neumannova okolí této částice. Zjistí si hodnotu statického pole v těchto buňkách a přidá hodnotu statického pole i buňky, ve které se částice právě nachází. Podle vzorce (2) jsou spočítány pravděpodobnosti přechodu do daných buněk. Následně je náhodně vybrán index z kategorického rozdělení se získaným pravděpodobnostním vektorem. Posledním krokem je získat souřadnice vybrané buňky pro přeskok a provést update pole.

```

1 function preskok(pole,ks)
2     b_idx = findall(x -> x == 1, pole)[1]
3     mask = dist_pole(pole, Tuple(b_idx))
4     vec_S = staticke_pole[mask]
5     idx_mask = findall(x -> x == 1, mask)
6     idx_mask = vcat(idx_mask, b_idx)
7     vec_S = vcat(vec_S, staticke_pole[b_idx])
8
9     jmenovatel = sum(exp.(.- vec_S .* ks))
10    P_ks = exp.(.- vec_S * ks) ./ jmenovatel
11    idx = rand(Categorical(P_ks))
12
13    idx_skok = idx_mask[idx]
14    pole[b_idx] = 0
15    pole[idx_skok] = 1
16 end

```

Pro více částic v poli fungují funkce velmi podobně. Možné konflikty při skoku více částic, které mohou skočit do stejného pole, byla situace vyřešena jednoduše. Update byl proto proveden iterativně postupně pro každou částici zvlášť. Není tak nutno konflikty řešit. Jedná se o samozřejmě jisté zjednodušení podobné například situaci, kdy funguje tzv. přednost zprava. Nicméně domníváme se, že toho zjednodušení nevnese do řešeného případu velký problém.

Další funkce už jsou jen pomocné funkce pro grafické znázornění výsledků a provedení samotné simulace a není třeba je zde zmiňovat.

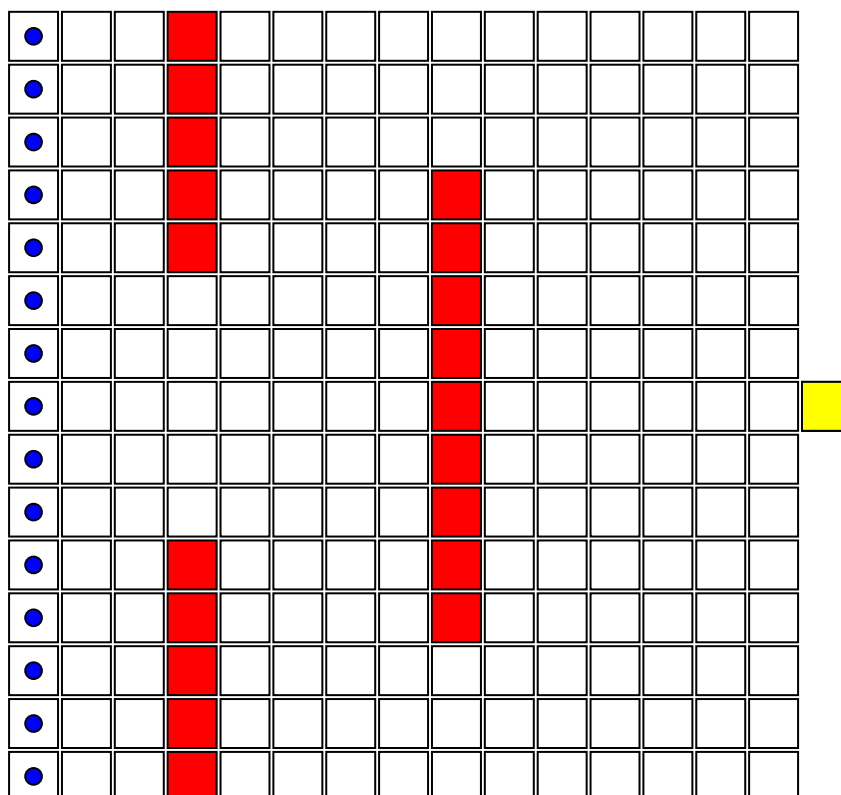
## 4 Simulace

Pro získání názorných výsledků byly vytvořeny celkem čtyři simulace, které lze rozdělit do dvou kategorií. Jednalo se o simulaci v místnosti bez překážek a s překážkami. Dále bylo zkoumáno chování pro jednu částici a pro více částic.

Pro jednu částici byl zkoumán vývoj doby do pohlcení v závislosti na konstantě  $k_S$  a přítomnosti překážek v místnosti. Pro více částic se pak jednalo o hustotu v  $(i, j)$ -té buňce pole, tedy bylo zkoumáno, kde se nejčastěji částice pohybovaly. Tato hustota byla následně vykreslena jako *heatmap*, tedy mřížka, kde je barvou vyznačena hustota v dané buňce. Zkoumána byla závislost hustoty i na hodnotě konstanty  $k_S$ , která významně ovlivňuje chování částic.

Zkoumaná mřížka měla rozměry  $15 \times 15$  a simulace byla provedena pro 15 částic seřazených v řadě na jednom konci „místnosti“. Ukázku mřížky s částicemi a překážkami lze vidět na Obr. 1.

Mřížka s překážkami, částicemi a atraktorem

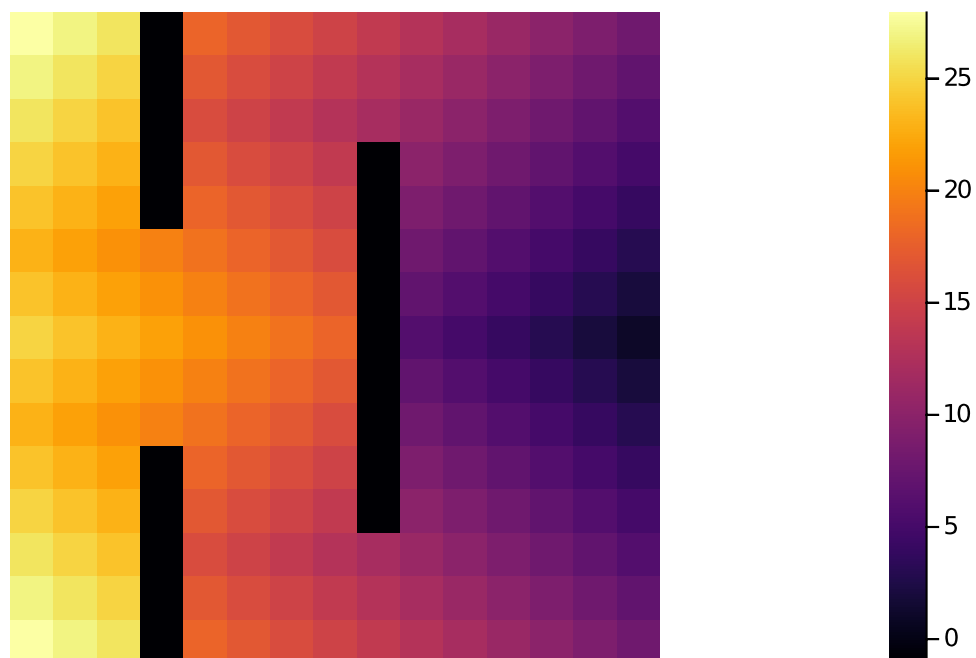


Obrázek 1: Mřížka použitá pro simulaci.

Je vidět, že mřížka je symetrická. Na obrázku 2 lze dále vidět vykreslené statické pole, tedy jednotlivá políčka jsou znázorněná barevně podle toho, jak daleko jsou od atraktoru. Černou barvou jsou zvýrazněné i překážky.

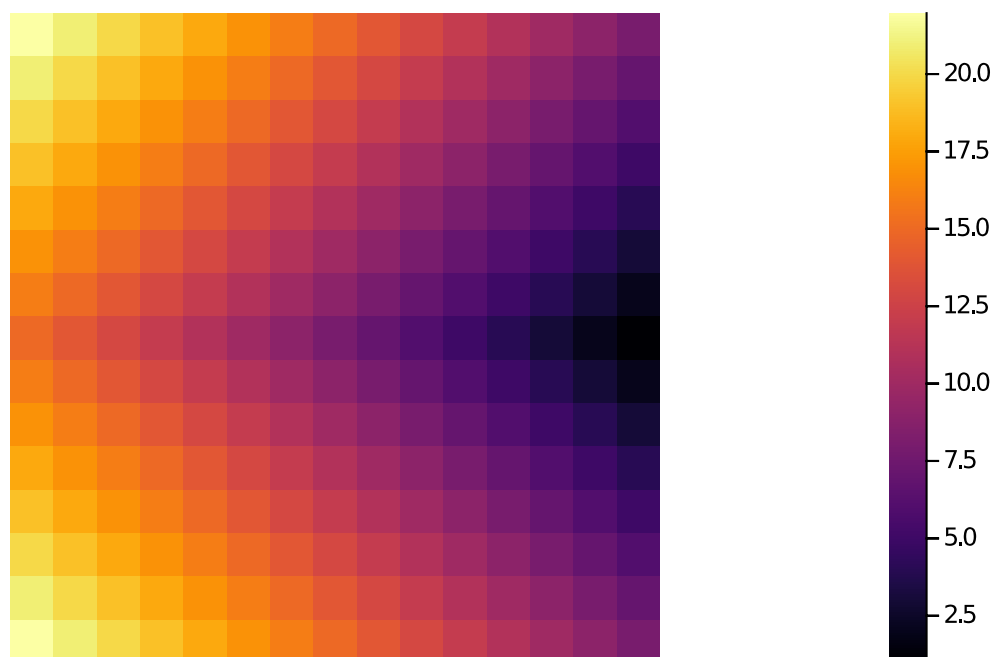
Stejnou heatmapu můžeme vidět i pro místnost bez překážek na Obr. 3.

Heatmapa vzdálenosti od atraktoru



Obrázek 2: Heatmap vzdáleností od atraktoru – vizualizace statického pole.

Heatmapa vzdálenosti od atraktoru



Obrázek 3: Heatmap vzdáleností od atraktoru – vizualizace statického pole.

## 5 Výsledky

Tato sekce shrnuje do několika podsekcí výsledky získané v průběhu experimentu. Zaměřujeme se na střední dobu do pohlcení a průměrnou hustotu buněk v mřížce. Porovnáváme pro mřížku s překážkami i bez překážek, uvažujeme vliv konstanty  $k_S$ .

Doporučujeme podívat se i na .gif soubory, které vizualizují jednotlivé simulace jako animace.

### 5.1 Výsledky pro samostatnou částici

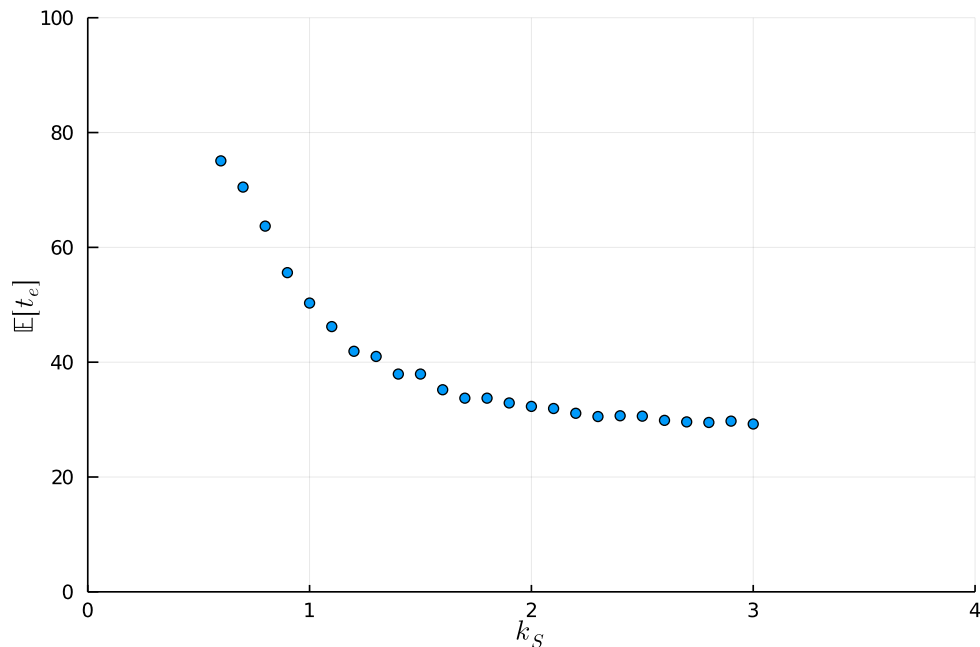
Pro samotnou částici byla spočítána střední doba do pohlcení pro různé hodnoty konstanty  $k_S$ .

#### 5.1.1 Střední doba do pohlcení

Prvním z úkolů bylo zjistit, jak střední doba do pohlcení  $t_e$  (evakuační čas) závisí na volbě konstanty  $k_S$ . Už ze vzorce (2) můžeme odhadovat, že se bude jednat o exponenciálu. Simulace byla provedena pro hodnoty  $k_S = 0.5:0.1:3$ . Pro každou hodnotu  $k_S$  byla simulace provedena 30krát, následně byl spočítán průměr.

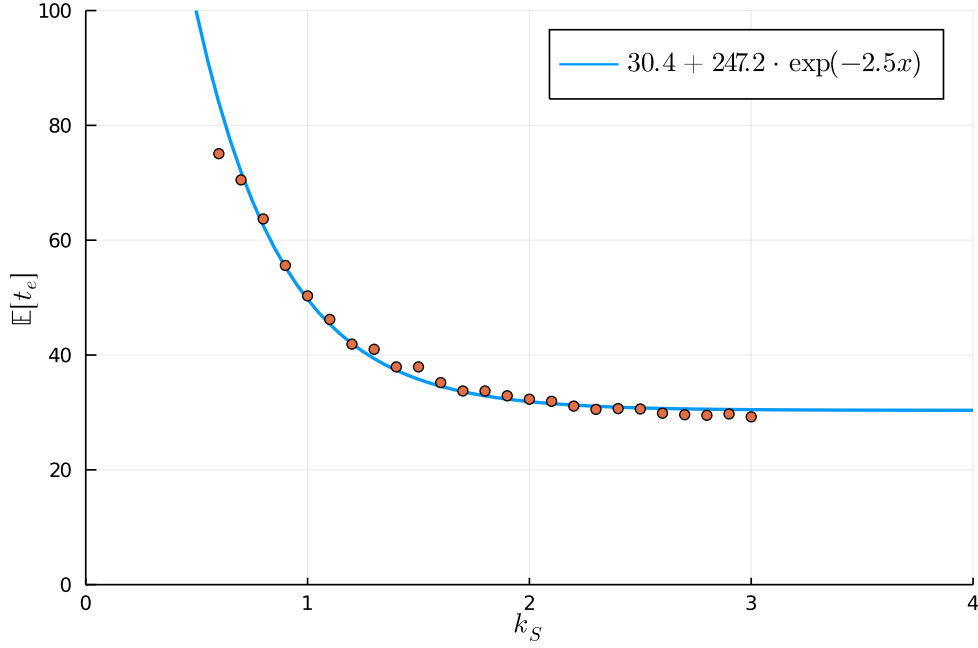
Už předem můžeme o průběhu  $k_S$  říct například to, že hodnota  $\mathbb{E}[t_e]$  bude zdola omezena hodnotou statického pole v místě startu částice. Pokud totiž v každém čase může částice skočit o jedno políčko, bude minimální čas potřebný k dosažení atraktoru roven vzdálenosti od atraktoru. Pro  $k_S \rightarrow \infty$  bude pravděpodobnost přechodu do buňky s nižší hodnotou statického pole  $\rightarrow 1$ .

Na Obr. 4 můžeme vidět výsledek simulace. S rostoucí hodnotou  $k_S$  se snižuje průměrná evakuační doba.



Obrázek 4: Průměrná doba do pohlcení závisící na  $k_S$ .

Data je možné nafitovat exponenciální funkcí, což můžeme vidět na Obr. 5.



Obrázek 5: Průměrná doba do pohlcení závisující na  $k_S$  s nafitovanou exponenciální funkcí.

## 5.2 Výsledky pro více částic

V modelu s více částicemi lze předpokládat, že interakce jednotlivých částic (zde fakt, že dvě částice nemohou být v jednu chvíli ve stejném poli mřížky) bude mít vliv na různé sledované veličiny. Například střední doba do pohlcení bude jistě delší, pokud budeme počítat dobu do pohlcení všech částic.

### 5.2.1 Hustota buněk

Nejdříve se podíváme na situaci, kdy neuvažujeme v místnosti žádné překážky.

Průměrná hustota v buňkách byla sledována pro tři hodnoty konstanty  $k_S$ : (0, 0.5, 1, 3). V prvním případě částice netuší, kterým směrem by se měly pohybovat a vzdálenost od atraktoru je pro ně bezvýznamná. To vidíme krásně i na hustotách pravděpodobnosti na Obr. 6. Tato konkrétní simulace byla provedena pro 1000 překoků, kdy ani poté nedosáhly všechny částice atraktoru a východu. Částice mají stejnou pravděpodobnost překoku do všech buněk ve svém okolí. Ve výsledku tak velmi často oscilují a nesledují žádný směr.

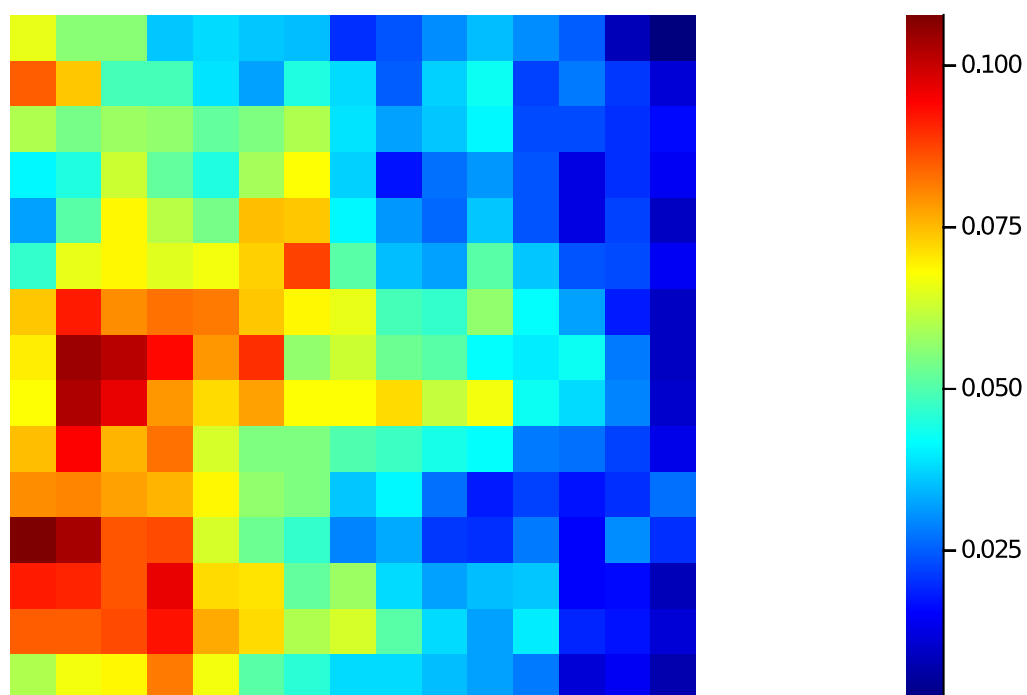
Pro zbylé hodnoty konstanty  $k_S$  už je situace lepší. Všechny částice se k atraktoru dostaly dříve než za  $T = 100$  sekund. Na obrázcích 7, 8 a 9 můžeme postupně sledovat, že zvyšování konstanty  $k_S$  zvyšuje hustotu pravděpodobnosti, že se částice budou nacházet ve středu mřížky a budou se pohybovat co nejefektivněji směrem k atraktoru.

Teď se přesuneme k případu, kdy už uvažujeme překážky jako v Obr. 1. Opět uvažujeme stejné hodnoty konstanty  $k_S$  jako v předchozím případě: (0, 0.5, 1, 3). Na heatmapách jsou překážky znázorněny hodnotou  $-0.05$ , aby bylo barevně jasné, kde se nachází.

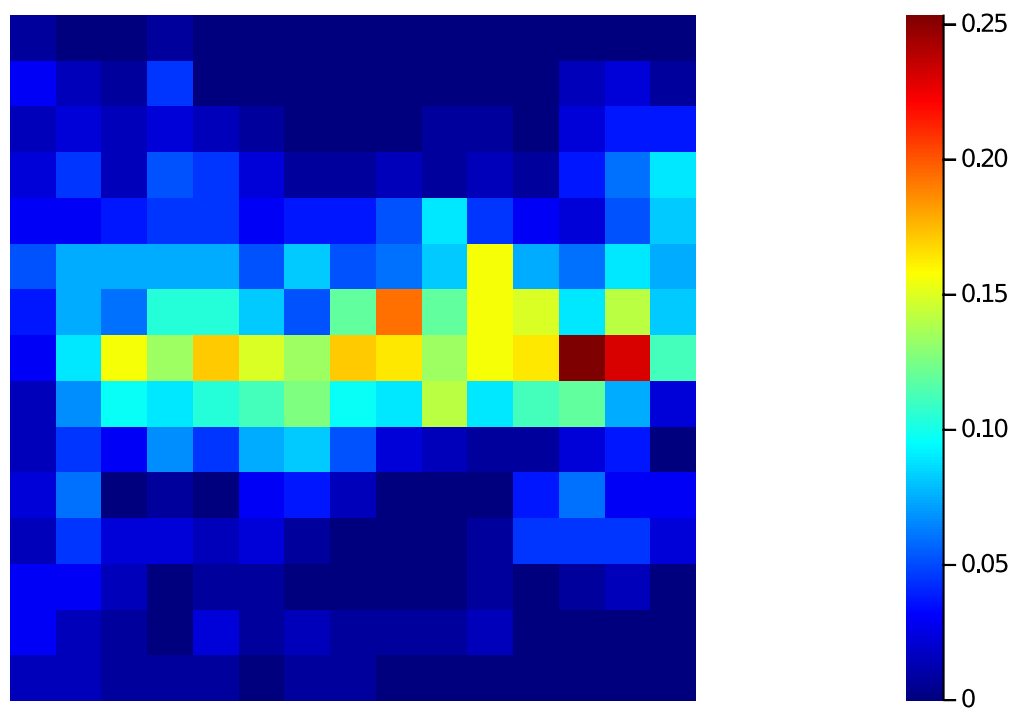
Simulace pro  $k_S = 0$  byla opět provedena pouze pro 1000 iterací, jelikož ani po takovém čase se částice nedokázaly dostat k atraktoru.

Na dalších obrázcích vidíme, jak se s rostoucí hodnotou  $k_S$  měnil průběh hustoty. Částice si postupně hledají lepší a lepší cesty, už tolik neutíkají špatným směrem a rychleji se dokáží dostat k atraktoru.

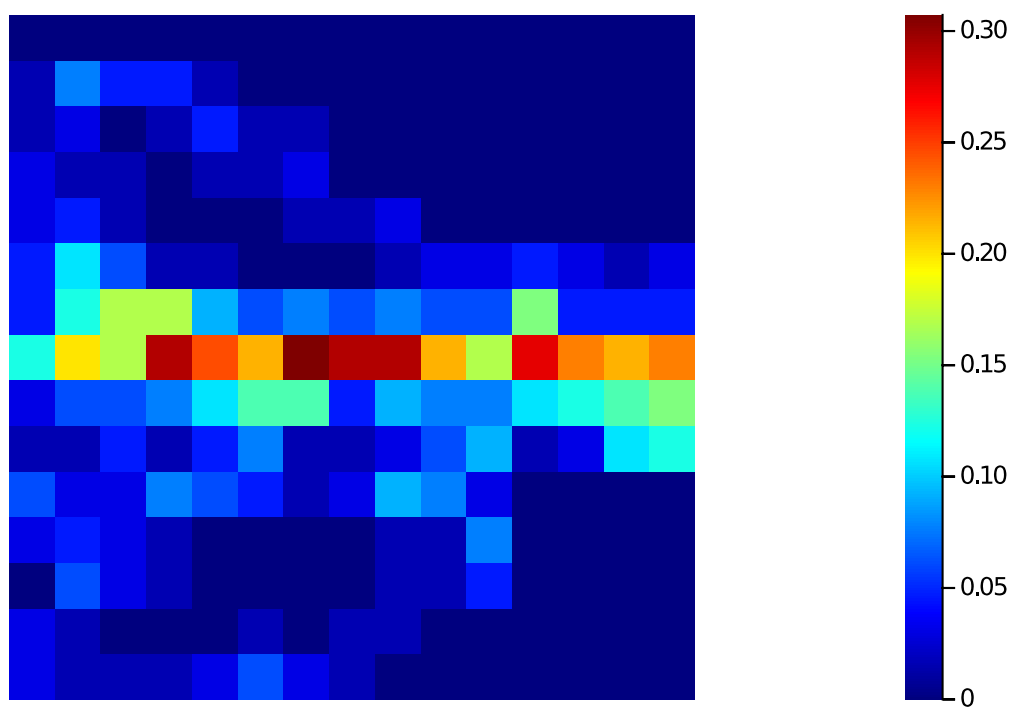




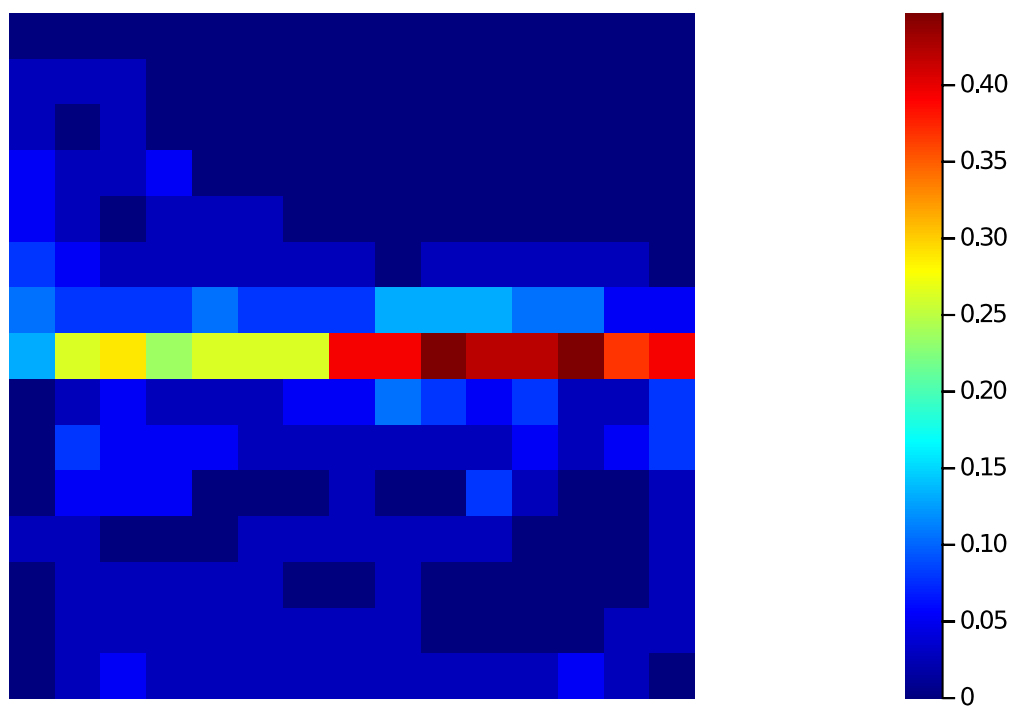
Obrázek 6: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 0$ , mřížka bez překážek.



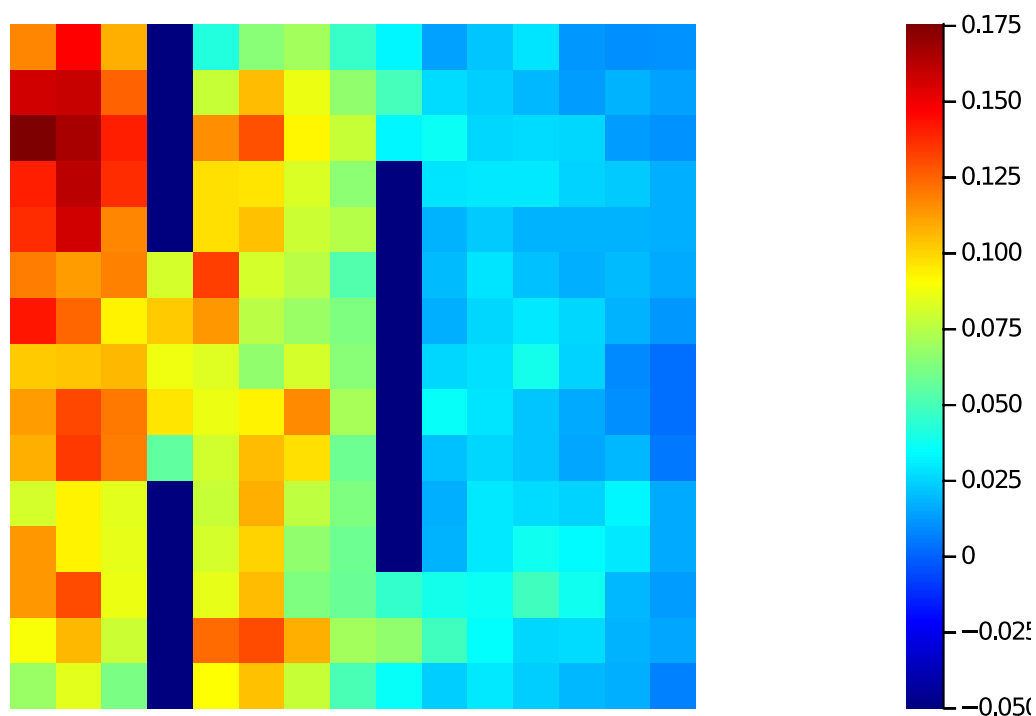
Obrázek 7: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 0.5$ , mřížka bez překážek.



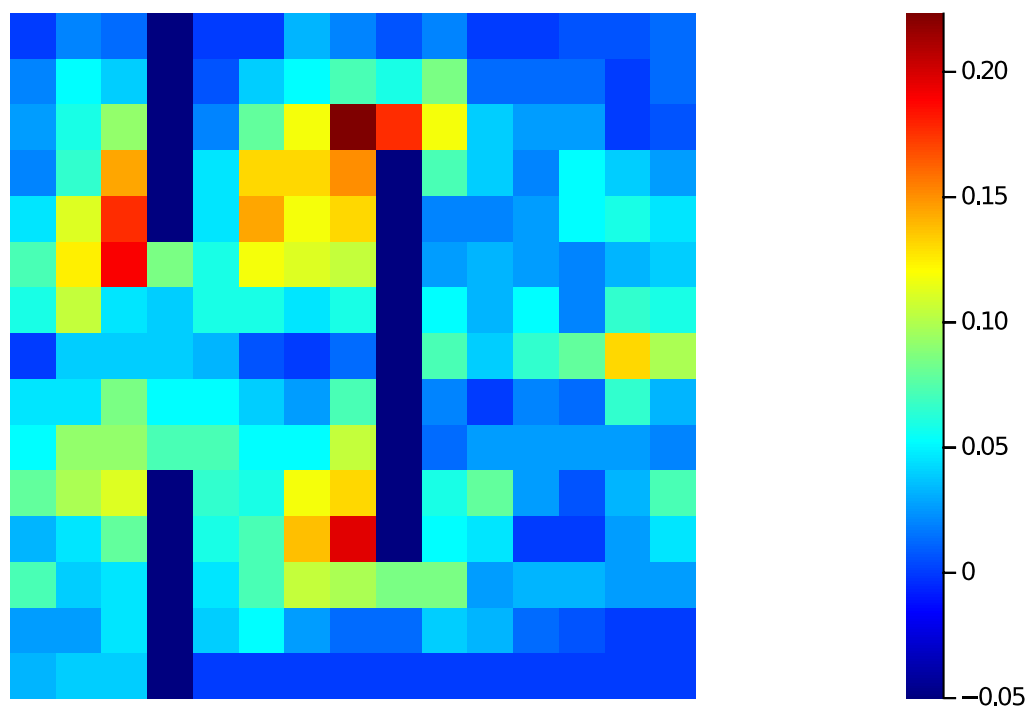
Obrázek 8: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 1$ , mřížka bez překážek.



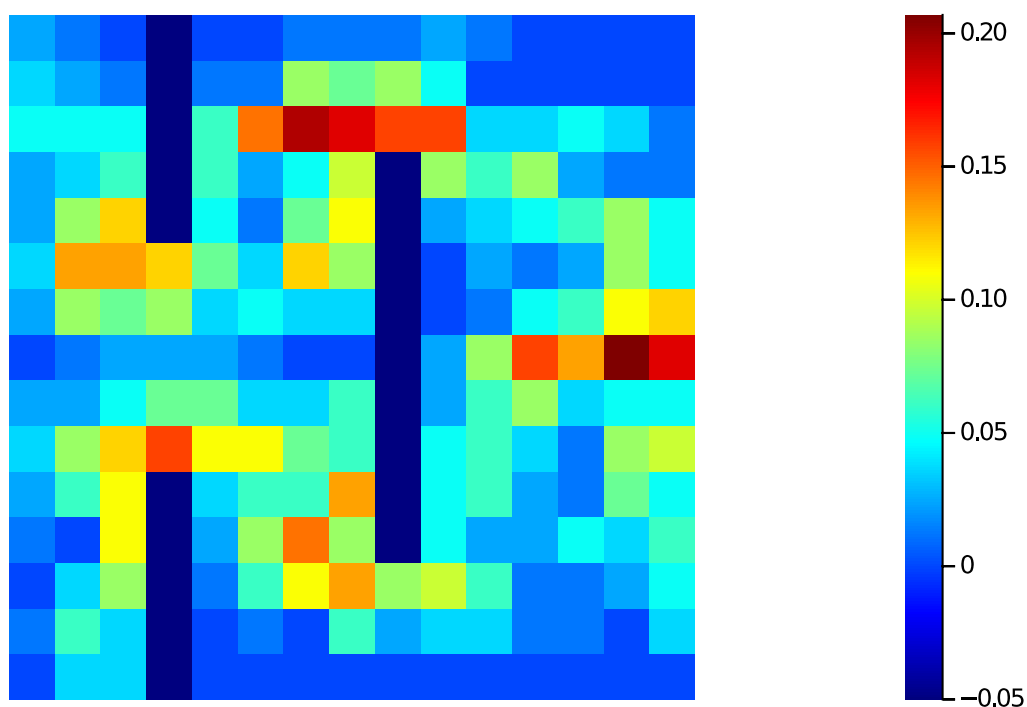
Obrázek 9: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 3$ , mřížka bez překážek.



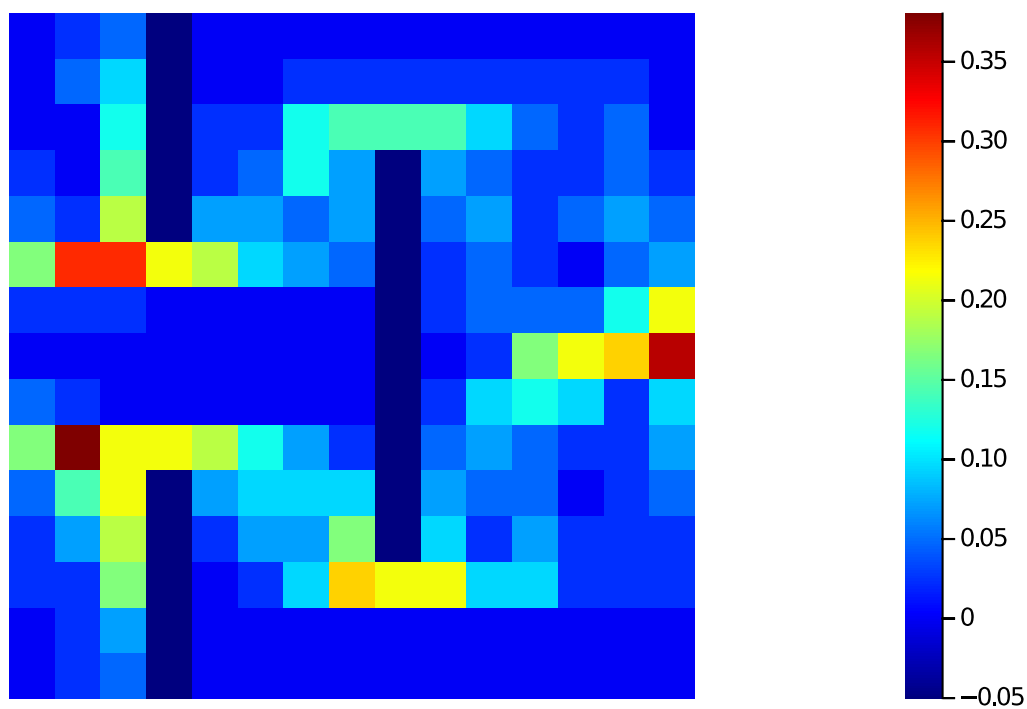
Obrázek 10: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 0$ , mřížka s překážkami.



Obrázek 11: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 0.5$ , mřížka s překážkami.



Obrázek 12: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 1$ , mřížka s překážkami.



Obrázek 13: Heatmap hustoty v jednotlivých buňkách pro  $k_S = 3$ , mřížka s překážkami.

## 6 Závěr

V protokolu byl zkoumán floor-field model se statickým polem. Byla zkoumána hlavně závislost veličin jako je střední doba do pohlcení a hustota buněk v mřížce na hodnotě konstanty  $k_S$ . Experimentálně bylo potvrzeno, že zvyšující hodnota  $k_S$  má za následek zrychlení pohybu částic a jejich rychlejší pohlcení.