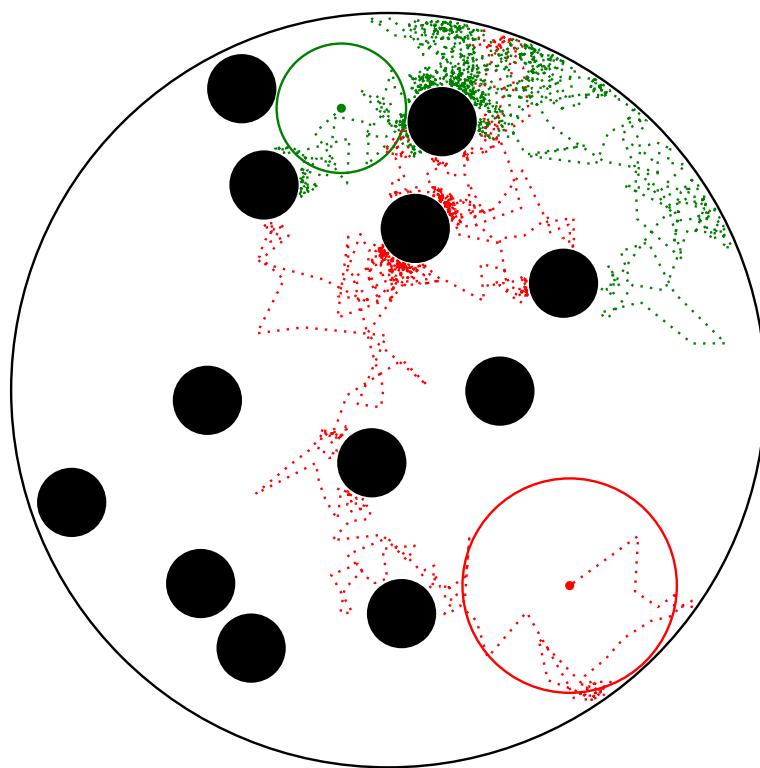


# Master 1 Internship

## Kinetic Monte Carlo studies of Reaction-Diffusion systems

Two dimensional studies on the search efficiency in the presence of obstacles

MANGEAT Matthieu - matthieu.mangeat@ens.fr



Laboratory : Theoretisch Physik - Universität des Saarlandes - Saarbrücken - Deutschland

Internship responsible : RIEGER Heiko - h.rieger@mx.uni-saarland.de

ENS correspondant : ALLEMAND Jean-François - jean-francois.allemand@ens.fr

Internship dates : from 17 February to 25 July 2014



# Table of Contents

<b>1 Methods of simulation</b>	<b>2</b>
1.1 Monte-Carlo method . . . . .	2
1.2 Gillespie's method [1] [2] [3] . . . . .	2
1.3 First passage kinetic Monte-Carlo (FPKMC) method [4] . . . . .	3
1.4 Intermittent ballistic transport algorithm . . . . .	3
1.5 Reaction-Diffusion problem with an annihilation rate [5] . . . . .	4
1.6 Sampling of random number according to an arbitrary probability density . . . . .	4
<b>2 Continuous domain</b>	<b>5</b>
2.1 Collision process . . . . .	5
2.1.1 Immobile obstacles . . . . .	6
2.1.1.1 First initialization . . . . .	6
2.1.1.2 Second initialization . . . . .	8
2.1.2 Mobile obstacles . . . . .	9
2.1.3 Intermittent ballistic transport . . . . .	10
2.2 Reaction process . . . . .	10
<b>3 Annulus</b>	<b>12</b>
3.1 Probability density $\rho_b(t)$ . . . . .	12
3.2 Probability density $\rho_n(\vec{r} t)$ . . . . .	13
3.3 Probability density $\rho_f(\varphi t)$ . . . . .	14
<b>4 Discrete domain</b>	<b>15</b>
4.1 Without bystanders . . . . .	15
4.1.1 Particles on lattice sites . . . . .	15
4.1.2 Particles with radius moving on discrete positions . . . . .	15
4.2 With bystanders . . . . .	16
4.2.1 With bystanders on lattice sites . . . . .	16
4.2.2 With bystanders out of lattice sites . . . . .	18
4.2.2.1 Asymmetric transition probabilities . . . . .	18
4.2.2.2 Symmetric transition probabilities : independent on the number of bystanders . . . . .	19
4.2.2.3 Symmetric transition probabilities : dependent on the number of bystanders . . . . .	19
4.2.3 Comparison of $t_{1/2}$ for different concentrations . . . . .	20

## Introduction

The project of this internship was the Kinetic Monte Carlo studies of some Reaction-Diffusion systems in presence of obstacles to find some efficient strategies. The domain where researchers look for targets is two dimensional with reflecting boundaries (Sec. 2 and Sec. 4.1) or periodic boundaries (Sec. 4.2). This domain, to simplify simulations, is a square or a circle. The obstacles inside the two dimensional domain can be mobile or immobile. In the section 1, algorithms and methods used all along the internship will be presented. In the section 2, results obtained in a continuous domain will be shown for different reaction conditions. In the section 3, the construction of functions which permit to have the different probability densities according to a protective domain will be explained.

And in the last section 4, results obtained in a discrete domain will be shown with, in particular, a new particle : the bystander which accelerates the reaction.

## 1 Methods of simulation

### 1.1 Monte-Carlo method

One of the most popular examples for a Monte-Carlo method is the calculation of  $\pi$ . For this, a large number of points are randomly chosen inside the square  $[-1, 1]^2$  (x and y coordinates are chosen according to an uniformly distributed probability density between -1 and 1) and we count how many points are inside the circle  $(x^2 + y^2 < 1)$  (Fig. 1). We obtain with the ratio of this number and the total number of points, the value of  $\frac{\pi}{4}$  as the area of the circle divided by the area of the square.

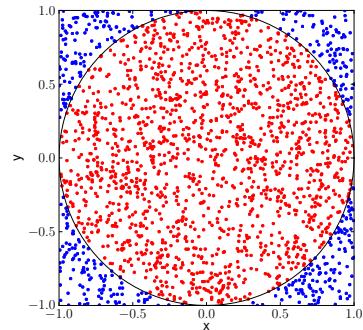


FIGURE 1 – 2000 points uniformly distributed

The precision of the measure of  $\pi$  depends on the total number of points  $N$  : the relative error scales like  $\frac{1}{\sqrt{N}}$ . In consequence, the precision of the estimation of  $\pi$  becomes better with a large  $N$ .

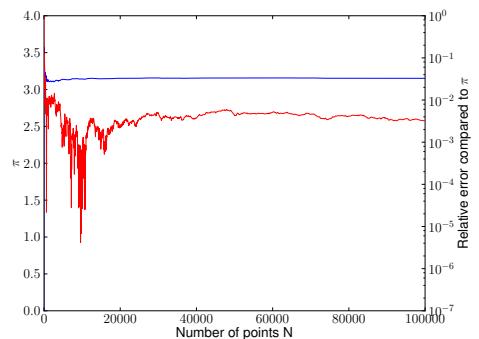


FIGURE 2 –  $\pi$  value obtain by Monte-Carlo method (blue) and the relative error (red) with the number of points

To have a good precision, we need also a good random number generator. Within my studies I always used the random number generator "mersenne twister", an algorithm for excellent pseudo random numbers, which passes the "Die Hard tests".

## Kinetic Monte-Carlo method

The kinetic Monte-Carlo method is a Monte-Carlo method to simulate the time evolution of some stochastic processes with a given known rate. The update mechanism consists to find the next transition thanks to the rates and the time when this transition happens. To do this, we iterate the processus over a large number  $N$  of samples where random numbers are chosen according to the known rates. This algorithm gives the correct solution with a relative error of  $\frac{1}{\sqrt{N}}$  if the transitions are not correlated and if the processes associated with the rates are of the Poisson process type.

### 1.2 Gillespie's method [1] [2] [3]

This method was found to simulate numerically the stochastic time evolution of  $N$  chemical species in reaction through  $M$  equations  $R_\mu$ .

We define  $\forall \mu \in \llbracket 1, M \rrbracket$ ,  $a_\mu d\tau =$  probability that the reaction  $R_\mu$  will occur in  $[t, t + d\tau]$  and

$$a = \sum_{\mu=1}^M a_\mu$$

$a_\mu$  depends on the concentration of reactive molecules and reaction coefficients.

We obtain the probability that the next reaction will occur in  $[t + \tau, t + \tau + d\tau]$  and will be the  $R_\mu$  reaction :

$$P(\tau, \mu) d\tau = a_\mu e^{-a\tau} d\tau \quad (1.1)$$

Thus, we can obtain the probability  $P_1(\tau) d\tau$  that the next reaction will occur in  $[t + \tau, t + \tau + d\tau]$  and the probability  $P_2(\mu|\tau) d\tau$  that the next reaction will be  $R_\mu$  knowing the time  $\tau$  of reaction.

$$P_1(\tau) = \sum_{\mu=1}^M P(\tau, \mu) = ae^{-a\tau} \quad (1.2)$$

$$P_2(\mu|\tau) = \frac{P(\tau, \mu)}{P_1(\tau)} = \frac{a_\mu}{a} \quad (1.3)$$

With these two probabilities, we want to compute the next reaction time  $\tau$  and the next reaction  $R_\mu$  with the Kinetic Monte-Carlo method. We choose two uniformly distributed random numbers in  $[0, 1]$ ,  $r_1$  and  $r_2$ . The probabilities  $P_1$  and  $P_2$  are normed such that

$$0 \leq F_1(\tau) = \int_0^\tau P_1(t) dt \leq 1 \quad \forall \tau \geq 0 \quad (1.4)$$

$$0 \leq F_2(\mu) = \sum_{k=1}^{\mu} P_2(k|\tau) \leq 1 \quad \forall \mu \in \llbracket 1, M \rrbracket \quad (1.5)$$

So, we can obtain  $\tau$  and  $\mu$  inverting the cumulative probabilities  $F_1$  and  $F_2$  such that

$$F_1(\tau) = r_1 \Rightarrow \tau = -\frac{1}{a} \ln(r_1) \quad (1.6)$$

$$F_2(\mu) = r_2 \Rightarrow \sum_{k=1}^{\mu-1} a_k < ar_2 \leq \sum_{k=1}^{\mu} a_k \quad (1.7)$$

So, for processes where there are more than one possible next event, the Gillespie's method generates randomly which will be the next event and when it will occur. This method permits to return a random number according to all probability densities (discrete or continuous) just thanks to an uniformly distributed random number in  $[0, 1]$  and the inversion of the cumulative probability.

### 1.3 First passage kinetic Monte-Carlo (FP-KMC) method [4]

This method is used for reaction-diffusion processes. It replaces small diffusion hops (in lattice model for example, which needs a small network for a convergence of solution and so creates a real slowing down of numerical simulations) by a diffusion of particles over long distances through a sequence of superhops. Each particle diffuses within its own protective domain, nonoverlapping with domains of other particles : this permit a factorization of  $N$ -body problem to 1-body problem totally solved. Efficient diffusion of one particle inside the protective domain is given by time-dependent Green's functions [6] which deliver the first passage statistics of random walks.

The master equation for a diffusive particle in  $d$ -dimension is given by

$$\frac{\partial P_D}{\partial t} = D \Delta P_D \quad (1.8)$$

where  $P_D(\vec{r}, t)$  denotes the probability distribution of a freely diffusing particle inside the domain  $G$  with a boundary  $\partial G$  which could be absorbing and/or reflecting.

Let call  $S(t)$  the surviving probability density, i.e. the probability density that the particle has not been absorbed at the domain boundary at time  $t$ ,  $\rho_b(t)$  the probability density that the particle is absorbed at the domain boundary for the first time at time  $t$ ,  $\rho_n(\vec{r}|t)$  the probability density to be at the position  $\vec{r} \in G$  knowing that the particle has not been absorbed at time  $t$  and  $\rho_f(\vec{r}|t)$  the probability density to be absorbed at the position  $\vec{r} \in \partial G$  knowing that particle is absorbed at time  $t$ .

$$S(t) = \int_G P_D(\vec{r}, t) d\vec{r} \quad (1.9)$$

$$\rho_b(t) = -\frac{dS}{dt}(t) \quad (1.10)$$

$$\rho_n(\vec{r}|t) = \frac{P_D(\vec{r}, t)}{S(t)} \text{ for } \vec{r} \in G \quad (1.11)$$

$$\rho_f(x|t) = -D \frac{\vec{\nabla} P_D(\vec{r}, t) \cdot \vec{n}_r}{\rho_b(t)} \text{ for } \vec{r} \in \partial G \quad (1.12)$$

The expression of  $P_D$  is well known thanks to Green's functions for the 1D case. It depends on the two boundary conditions : reflecting or absorbing. For higher dimensions, the expression of  $P_D(\vec{r}, t)$  is known only for some domains  $G$  thanks to Green's functions : rectangle, circle, cuboid and sphere for example. For  $L = \infty$ , we have the well-known solution in dimension  $d$

$$P_{D,\infty}(\vec{r}, t) = \frac{1}{(4\pi Dt)^{d/2}} e^{-\frac{\vec{r}^2}{4Dt}} \quad (1.13)$$

## FPKMC Algorithm

- (1) At  $t=0$ , construct nonoverlapping protective domains around all walkers.
- (2) Generate a first passage time (fpt) for each protective domain according to  $\rho_b$  within this domain and store them all in a list.
- (3) Take  $t_\mu$  the minimum of all fpt and find the corresponding domain  $\mu$ . Put  $t \leftarrow t_\mu$ .
- (4) Sample an exit position for the selected walker  $\mu$  according to  $\rho_f$ .
- (5) See the reaction condition (collision...) and take the appropriate action (remove...).
- (6) Sample a new position according to  $\rho_n$  for particles whose protective domains are close to the walker  $\mu$ 's new position.
- (7) Construct new protective domain for moving particles in (4) and (6) and replace the corresponding fpt in the list by a new one according to  $\rho_b$ .
- (8) Go to (3) until the complete reaction or the reach of a maximal time.

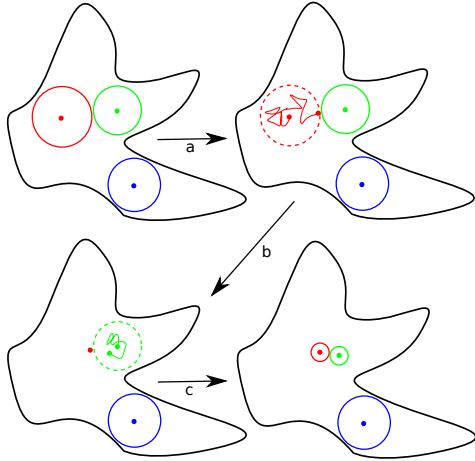


FIGURE 3 – Steps of FPKMC algorithm : a/ First passage step for the red particle. b/ No passage step for the green particle due to the new position of red particle too close to the green boundary. c/ Construction of new protective domains... and so on.

## 1.4 Intermittent ballistic transport algorithm

We need to define two probability densities according to the change of motion  $\rho_{d \rightarrow b}(t)dt$  the probability to change the motion from diffusive to ballistic between  $t$  and  $t+dt$  and  $\rho_{b \rightarrow d}(t)dt$  the probability to change the motion from ballistic to diffusive between  $t$  and  $t+dt$ .

$$\rho_{d \rightarrow b}(t) = k_d e^{-k_d t} \quad (1.14)$$

$$\rho_{b \rightarrow d}(t) = k_b e^{-k_b t} \quad (1.15)$$

- (1) At  $t=0$ , construct nonoverlapping protective domains around all walkers : they have a diffusive motion.

- (2) Generate a first passage time (fpt) for each protective domain according to  $\rho_b$  within this domain and store them all in a list.

(3) Take  $t_\mu$  the minimum of all fpt and all changing motion time and find the corresponding domain  $\mu$ . Put  $t \leftarrow t_\mu$ .

(4) If  $t_\mu$  is a fpt, repeat all steps (5),(6),(7) and (8) of FPKMC algorithm else change the motion of  $\mu$  particle.

(5) If the motion changes from diffusive to ballistic, sample the new position according to  $\rho_n$ . Take randomly the direction  $\vec{v}$  of the ballistic transport. Sample the next changing motion time of the particle as the minimum of the time according to  $\rho_{b \rightarrow d}$  and the next crossing time to the domain boundary. Else if the motion changes from ballistic to diffusive, compute the new position as  $\vec{r} \leftarrow \vec{r} + \vec{v}t$ . Construct the new protective domain of the particle, replace its corresponding fpt in the list and sample the next changing motion time of the particle according to  $\rho_{d \rightarrow b}$ .

(6) Go to (3) until the complete reaction or the reach of a maximal time.

In our simulations, a particle in ballistic motion couldn't react, hence it is allowed to be within the protection box of another particle, as long as it is ballistically moving. As soon as a particle in ballistic motion meet the domain boundary, his motion changes to diffusive.

## 1.5 Reaction-Diffusion problem with an annihilation rate [5]

This problem follows the equation

$$\frac{\partial P}{\partial t}(\vec{r}, t) = D \Delta P(\vec{r}, t) - k(\vec{r}, t)P(\vec{r}, t) \quad (1.16)$$

### Homogeneous rate $k(t)$

We define the probability density to have an annihilation at time  $t$  as :

$$\rho_a(t) = -\frac{d}{dt}[\exp(-\int_0^t k(\tau)d\tau)] \quad (1.17)$$

We only have to replace in the FPKMC algorithm the step (3) to look for the minimum of the fpt  $t_b$  according to  $\rho_b$  and the annihilation time  $t_a$  according to  $\rho_a$ . If  $t_a < t_b$ , the particle is removed at the position  $\vec{r}$  sampled with  $\rho_n(\vec{r}, t_a)$ , otherwise the a new position  $\vec{r}$  of the particle is sampled with  $\rho_n(\vec{r}, t_b)$ .

### Inhomogeneous rate $k(\vec{r}, t)$

If the annihilation rate is inhomogeneous, there is a correlation between the annihilation position and the annihilation time, hence, the idea above will fail. We define thus,

$$k_m(t) = \max_{\vec{r} \in G} |k(\vec{r}, t)| \quad (1.18)$$

$$\rho_a(t) = -\frac{d}{dt}[\exp(-\int_0^t k_m(\tau)d\tau)] \quad (1.19)$$

In this case the particle is removed if  $t_a < t_b$  and if an uniformly distributed random number in  $[0, 1]$  is smaller than  $\frac{k(\vec{r}, t)}{k_m(t)}$  which consists to the rejection sampling method. A more precise explanation, not necessary for following sections, is given in [5].

## 1.6 Sampling of random number according to an arbitrary probability density

There are two possibilities to generate random numbers according to an arbitrary probability density  $\rho$  :

1- the inversion sampling, according to the Gillespie's method seen in the section 1.2. We need to inverse the cumulative probability  $F = \int_0^t \rho(t') dt'$  to obtain a random number according to  $\rho$  from an uniformly distributed random number in  $[0, 1]$ . But this is many times too hard to do when the bisection method is the lonely way to inverse : we need to evaluate about 20 times F to have this number.

2- the rejection sampling.

This second method consists to cover the probability density  $\rho(t)$  function by an easily invertible function  $\rho_h(t)$ . For this we choose  $\rho_h$  like a piecewise constant function on N intervals such that

$$\forall t \in [t_i, t_{i+1}], i \in \llbracket 0, N - 2 \rrbracket \quad \rho_h(t) = p_i \quad (1.20)$$

$$\forall t \in [t_{N-1}, \infty], \rho_h(t) = \frac{Q}{(t + 1 - t_{N-1})^2} \quad (1.21)$$

$$\text{with } \forall i, p_i = \max_{t \in [t_i, t_{i+1}]} \rho(t) \text{ and } Q = p_i \cdot (t_{i+1} - t_i)$$

$Q$  is a constant define by  $p_0 = \rho(0)$  and  $p_1$  independant. The cumulative probability of each interval is the same :  $Q$ . The norm of  $\rho_h$  is, thus,  $k = N \cdot Q$ .

Let  $r$  a random number from a uniform distribution in the interval  $[0, 1]$ , we choose a candidate time  $t_{cand}$  according to  $\rho_h$ , we have easily the interval  $m$  corresponding to  $r$  is the integer part of  $N \cdot r$ .

$$t_{cand} = (r - \frac{m}{N}) \cdot \frac{1}{p_m} + t_m \quad \forall m \in \llbracket 0, N - 2 \rrbracket \quad (1.22)$$

$$t_{cand} = t_{N-1} - 1 + \frac{1}{N \cdot (1 - r)} \text{ for } m = N - 1 \quad (1.23)$$

To have the good probability density  $\rho$  and not  $\rho_h$ , we need to reject some of these  $t_{cand}$  : if the ratio  $\frac{\rho(t_{cand})}{\rho_h(t_{cand})}$  is smaller than an uniformly distributed random number  $r_{rej}$  in  $[0, 1]$ ,  $t_{cand}$  is rejected and a new one is chosen. For a large number of  $t_{cand}$  computed (Monte-Carlo method), the probability density of these  $t_{cand}$  corresponds very well to  $\rho$ .

The special shape of the helping density makes its inversion very fast in combination with the precalculation of the acceptance rate. The rejection rate depends only on  $k$ . It is given by  $\frac{k-1}{k}$ .

With this method, we have as average number of evaluation of  $\rho$  :  $k \sim 1.02$ .

### An example : the circle protective domain

For this protective domain the radius of circle  $R=1$  and the diffusivity  $D=1$ , the probability densities are

$$\rho_{unit,b}(t) = 4 \sum_{\alpha_0} \frac{e^{-\alpha_0^2 t}}{J_1(\alpha_0)} \quad (1.24)$$

$$\rho_{unit,n}(r|t) = \frac{\sum_{\alpha_0} r e^{-\alpha_0^2 r} \frac{J_0(\alpha_0 r)}{J_1(\alpha_0)^2}}{\sum_{\alpha_0} \frac{e^{-\alpha_0^2 t}}{\alpha_0 J_1(\alpha_0)}} \quad (1.25)$$

with  $\alpha_0$  the zeros of  $J_0$ .

$\rho_f(R, \varphi, t)$  returns only an angle  $\varphi$  uniformly distributed between 0 and  $2\pi$ .

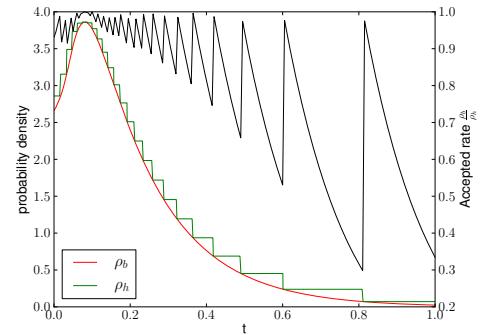


FIGURE 4 – Covering function of  $\rho_b$  for  $Q=0.05$  and the accepted rate  $\frac{\rho_b}{\rho_h}$

Normally  $Q \sim 0.001$  and  $N \sim 1000$  for a better approximation of  $\rho_b$  and so a faster code ( $k \sim 1.01$ ). If the accepted rate is bigger than 95% the time  $t_{cand}$  is accepted without evaluating  $\rho_b$ . So we evaluate only in 15% of cases  $\rho_b$ .

To cover  $\rho_{unit,n}(r|t)$ , we need to construct M time intervals such that

$$B(t_j, t_{j+1}) = \int_0^1 dr \max_{t \in [t_j, t_{j+1}]} \rho_n(r|t) < k_j \sim 1.02 \quad (1.26)$$

where the covering function of  $\max_{t \in [t_j, t_{j+1}]} \rho_n(r|t)$  is computed as precedently. Thus, we can found  $r$  with a rejection sampling for a knowing  $t$ .

We obtain for  $R \neq 1$  and  $D \neq 1$  the probability densities

$$\rho_b(t) = \frac{D}{R^2} \rho_{unit,b} \left( \frac{Dt}{R^2} \right) \quad (1.27)$$

$$\rho_{unit,n}(r|t) = \frac{1}{R} \rho_{unit,n} \left( \frac{r}{R} \left| \frac{Dt}{R^2} \right. \right) \quad (1.28)$$

Before my arrival, all these functions were written for all one-dimensional intervals, a rectangle, a circle, a sector, a cuboid, a sphere, a spherical sector, a spherical cap. I contribute writting functions for an annulus. The construction of these functions will be explain more precisely in the section 3.

## 2 Continuous domain

The first results shown in this report will be for particles moving in a continuous space-time.

The first scenario of reaction is simply the finding of a target by a searcher when the two particles collide (Sec. 2.1).

The second scenario of reaction is the finding of a target when the time during which a searcher is closer than  $\delta_r$  to the target is superior than  $\delta_t$  (Sec. 2.2).

For these two conditions, as soon as the target is found, it's removed and the searcher survives, to eventually find another target. We look for the probability density of the time of targets' finding with the FPKMC method and it's evolution with the number and the area occupied by obstacles. In the adimensional studied 2D-domain (a square or a circle), there are  $N_s$  searchers looking for  $N_t$  targets with a radius

$r_{part} = 0.005$ , diffusing both with an unitary diffusivity, and  $N_o$  obstacles with a radius  $r_{obs}$ , chosen proportional to  $r_{part}$ , which can diffuse (Sec. 2.1.2) with an unitary diffusivity.

Let call  $d_{part}$  the smallest distance to other particles,  $d_{obs}$  the smallest distance to obstacles,  $d'_{obs}$  the second smallest distance to obstacles,  $d_b$  the distance to the closest boundary,  $d'_b$  the distance to the second closest boundary,  $d_{min}$  the minimum of  $d_{part}$  and  $d_{obs}$  and  $d'_{min}$  the minimum of  $d_{part}$  and  $d'_{obs}$ . All this distances are computed without taking into account the hard core distances.

## 2.1 Collision process

For the collision problem, we look for the time of targets' finding by searchers in the next subsections, where obstacles are immobile (Sec. 2.1.1) or mobile (Sec. 2.1.2) and where searchers can have an intermittent ballistic transport (Sec. 2.1.3). The algorithm used here is the FPKMC algorithm shown in the section 1.3 where the step (5) is "Remove the target if the smallest distance from a searcher is smaller than twice the radius of particles".

### 2.1.1 Immobile obstacles

For the case where obstacles are immobiles, we have two ways to initialize uniformly the domain. For the first one, we let enough space between obstacles and boundaries which permit to a particle to pass everywhere, i.e. creating space of twice the particles' radius around obstacles. For the second one, the position of obstacles are choosen randomly just with no obstacles' overlap and so producing some non allowed area for particles.

#### 2.1.1.1 First initialization

The positions of obstacles are choosen randomly such that the distance to domain boundaries is bigger than  $r_{obs} + 2r_{part}$  and the distance to other obstacles is bigger than  $2r_{obs} + 2r_{part}$ . The positions of particles are choosen randomly in the free area such that the distance to domain boundaries is bigger than  $r_{part}$  and the distance to obstacles is bigger than  $r_{obs} + r_{part}$ .

### Protective domains

The protective domains used for this problem are :

1- If  $d_{obs} = 0$ , a semicircular sector tangent to the obstacle, centered in the particle position with a radius  $\min(0.05r_{obs}, d'_{min}, d_b)$ , not to big due to the small area rejected by this particular protective domain (Fig. 5).

2a- Else if  $d_b < d_{min}$ , a rectangle such that the smallest distance from the particle to each side is the biggest possible to have the biggest possible first passage time (Fig. 6). This is only use for a square domain.

2b- Or else if  $d_b = 0$ , a semicircular sector along the boundary centered in the particle position with a radius  $\min(d_{min}, d_b)$  for a square domain (which requires more computational time but which gives the same total probability densities).

2c- Or else if  $d_b = 0$ , a sector centered in the particle position with a radius  $d_{min}$  such that the sector is totally in the domain, for a circular domain (Fig. 7).

3- Else, a circle centered in the particle position with a radius  $\min(d_{min}, d_b)$ .

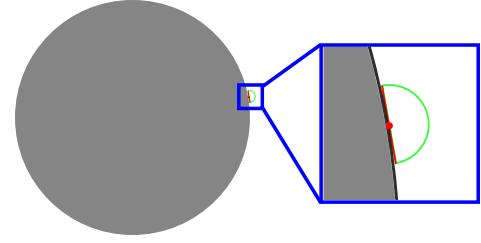


FIGURE 5 – Protective domain when the particle is close to obstacles : in red, the reflecting boundary and in green, the absorbing boundary

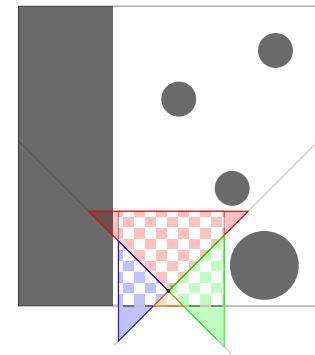


FIGURE 6 – Rectangle protective domain construction : in black the area occupied by obstacles and other protective domains which couldn't be overlaped

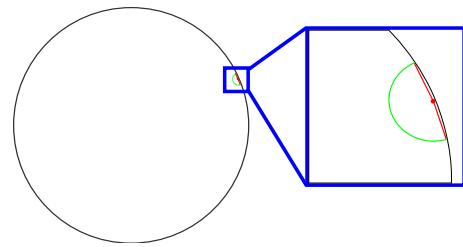


FIGURE 7 – Protective domain when the particle is close to the boundary of a circular domain : in red, the reflecting boundary and in green, the absorbing boundary

A slowing down of the numerical code is possible when a distance or a time is smaller than  $10^{-15}$  due to the double precision so the collision condition is in reality : the distance between the searcher and the target is smaller than  $2.001r_{part}$  and the condition to have the protective domain close to obstacles is  $d_{obs} < 0.001r_{part}$  which introduce an error smaller than the one due to stochastic fluctuations.

### Results for one target and one searcher

For the case where one searcher looks for one target, the computation time increase with the number of obstacles in a  $t \sim N_o^{2.5}$  law (Fig. 8).

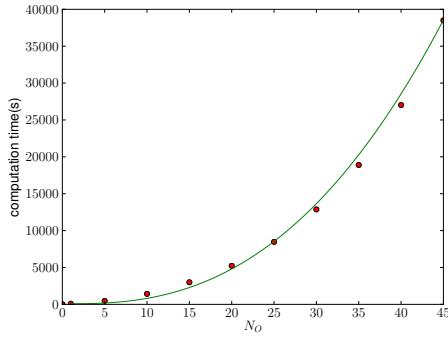


FIGURE 8 – Evolution of computational time with  $N_o$  for 100,000 samples and  $r_{obs} = 0.05$  in a square

The probability density  $\rho_c(t)$  that the target is found at the time  $t$  is reproduced in the Fig. 9. We see that

$$\rho_c(t) \sim \frac{1}{\langle t \rangle} e^{-\frac{t}{\langle t \rangle}} \quad (2.1)$$

with  $\langle t \rangle$  the average time according to  $\rho_c$ . This expression is the one of no-memory processes. The evolution is similar whatever the obstacle number after the renormalization by  $\langle t \rangle$ . So,  $\langle t \rangle$  is enough to describe the probability density and so its evolution with all different arguments ( $N_o, r_{obs}$ ).

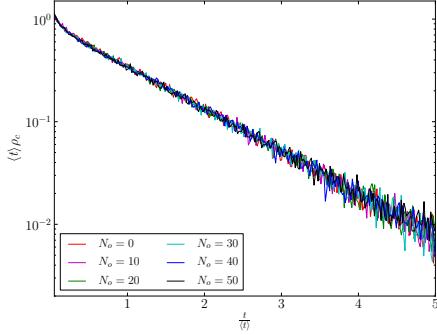


FIGURE 9 –  $\rho_c(t)$  for different values of  $N_o$  for  $r_{obs} = 0.05$  in a square

The mean collision time increases linearly with the domain area without obstacles (Fig. 10), increases with the number of obstacles at radius constant and decreases with the radius of obstacles for a constant area occupied by obstacles (Fig. 11 and 12).

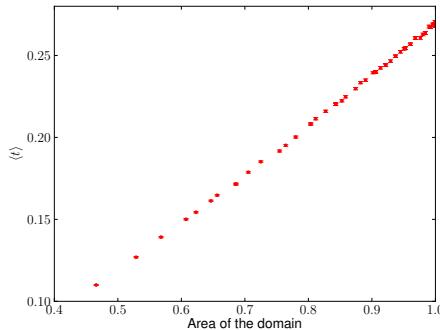


FIGURE 10 – Mean collision time vs area of a circular domain without obstacle

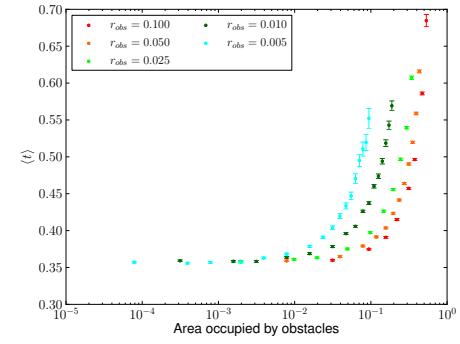


FIGURE 11 – Mean collision time vs area occupied by obstacles for many radii in a square

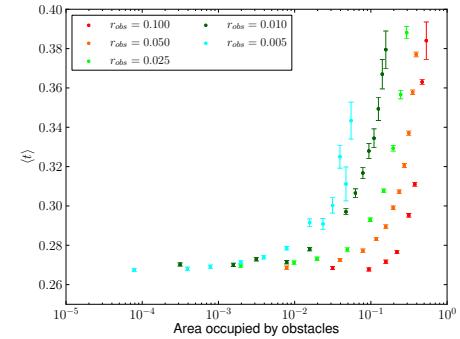


FIGURE 12 – Mean collision time vs area occupied by obstacles for many radii in a circle

The behaviour of  $\langle t \rangle$  with the area  $A$  occupied by obstacles is linear for small  $A$  with a slope decreasing with  $r_{obs}$  and diverges for  $A$  in the range of 0.6.

### Results for many targets and many searchers

For the case where many searchers look for many targets, the evolution of the proportion of targets not found at time  $t$  is shown in Fig. 13 and 14.

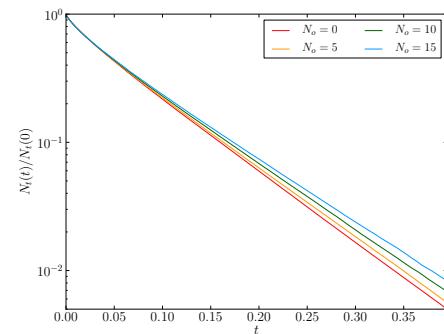


FIGURE 13 – Time evolution of  $N_t$  for  $N_s = 5$  and  $N_t = 5$  for different  $N_o$  and  $r_{obs} = 0.05$  in a square

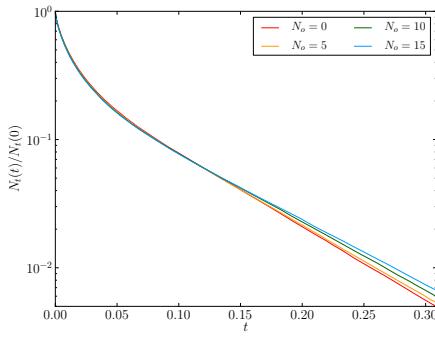


FIGURE 14 – Time evolution of  $N_t$  for  $N_s = 5$  and  $N_t = 5$  for different  $N_o$  and  $r_{obs} = 0.05$  in a circle

The evolution looks like

$$N_t(t) = N_t(0)e^{-\lambda t} \text{ with } \lambda = \frac{\ln 2}{t_{1/2}} \text{ and } N(t_{1/2}) = \frac{N_t(0)}{2} \quad (2.2)$$

So  $t_{1/2}$  represents totally  $N_t(t)$ . It's evolution with the number of obstacles presents a minimum (Fig. 15 and 16). This behaviour can be explained by the fact that if there are  $N_o$  obstacles and  $N_s$  searchers, in average, one searcher meets  $\frac{N_o}{N_s}$  obstacles before finding the target (if  $r_{obs} \geq r_{part}$ ) and so if  $N_o > \alpha N_s$ ,  $t_{1/2}$  increases with  $N_o$  and otherwise,  $t_{1/2}$  decreases with  $N_o$  only due to the fact that the area allowed for particles decreases. The value of  $\alpha$  depends on the value of  $N_o$  where the 2 phenomenon have the same weight.

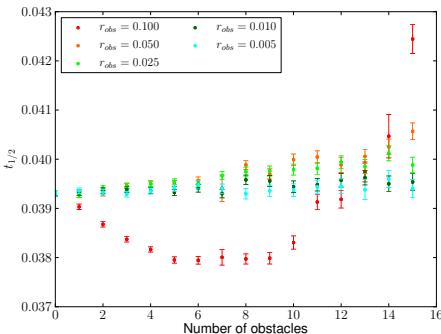


FIGURE 15 – Evolution of  $t_{1/2}$  with  $N_o$  for  $N_s = 5$  and  $N_t = 5$  for many radii in a square

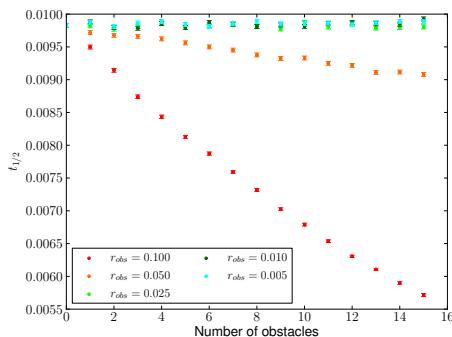


FIGURE 16 – Evolution of  $t_{1/2}$  with  $N_o$  for  $N_s = 5$  and  $N_t = 5$  for many radii in a circle

### 2.1.1.2 Second initialization

The positions of obstacles are choosen randomly such that the distance to domain boundaries is bigger than  $r_{obs}$  which assures that obstacles are in the domain and the distance to other obstacles is bigger than  $2r_{obs}$ , which assures a nonoverlaping between obstacles. The positions of particles are choosen randomly in the free area such that the distance to domain boundaries is bigger than  $r_{part}$  and the distance to obstacles is bigger than  $r_{obs} + r_{part}$ .

This configuration of obstacles can isolate particles from some space in the domain so, a configuration is accepted only if each target can be found by, at least, one searcher (Fig. 17). Otherwise, the time of meeting is infinite and  $\langle t \rangle$  has no sense.

We can easily find all separated areas as cycles in a graph containing as link, the obstacles closer to the boundary than  $r_{obs} + 2r_{part}$  or closer to another obstacles than  $2r_{obs} + 2r_{part}$ . After finding all these possible cycles (if they exist), we look the belonging cycle of each targets (it's only to look if a point is in a polygon) and if at least one searcher belongs to the same cycle. If it's the case, the first collision time is computed, otherwise another configuration is computed.

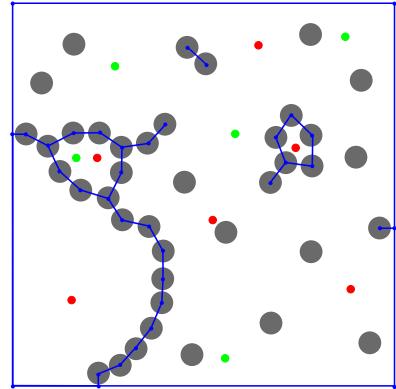


FIGURE 17 – In red, searchers and in green, targets. This configuration is accepted

### Protective domains

This different initialization requires to have two other kinds of protective domains when  $d_{obs} = 0$  and :

- $d_b = 0$ . We need to take a sector centered in the particle with an angle  $\varphi = \arccos(\frac{x_{obs}-r_{part}}{r_{obs}+r_{part}})$  and a radius  $r = \min(0.03, \sin(0.1\phi)) \cdot (r_{obs} + r_{part})$  (Fig. 18).

- $d'_{obs} = 0$ . We need to take a sector centered in the particle with an angle  $\varphi = 2 \arccos(\frac{x_{obs}-r_{part}}{2(r_{obs}+r_{part})})$  and a radius  $r = \min(0.03, \sin(0.1\phi)) \cdot (r_{obs} + r_{part})$  (Fig. 19).

The radius of these sectors is taken the smallest as possible to have the smallest error in rejected area given by the difference between orange and red angles which needs to be, at least, smaller than 10 %. This happens due to reflecting boundaries of the sector. But, this radius needs to be biggest than a certain value  $0.03 \cdot (r_{obs} + r_{part})$  to have a fast simulation.

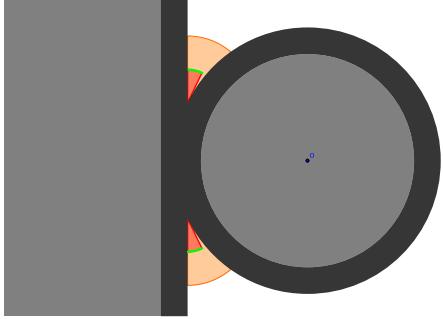


FIGURE 18 – Protective domain when the particle is close to one obstacle and one boundary with in black the particle's forbidden area, in red the reflecting boundaries and in green the absorbing boundaries

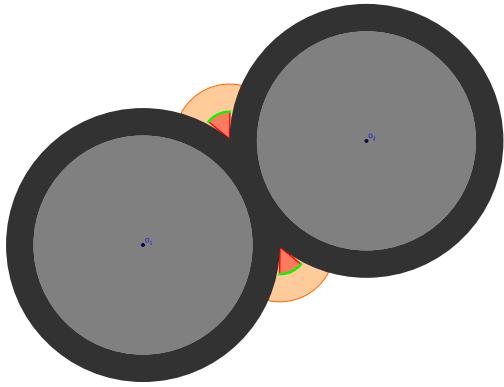


FIGURE 19 – Protective domain when the particle is close to two obstacles with in black the particle's forbidden area, in red the reflecting boundaries and in green the absorbing boundaries

### Results for one target and one searcher in a square

The probability  $\rho_c$  has the same behaviour as in the case of the first initialization and the mean collision time is quite larger (Fig. 20).

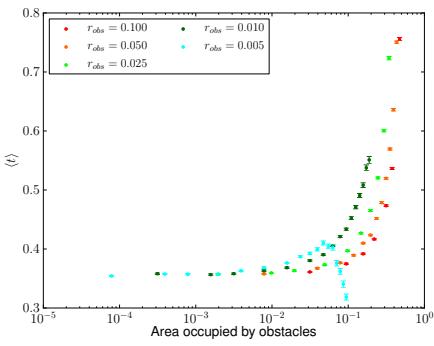


FIGURE 20 – Mean collision time vs area occupied by obstacles for many radii

### Results for many targets and many searchers in a square

The time evolution of  $N_t$  has the same behaviour as in the case of the first initialization and  $t_{1/2}$  seems to have a minimum for a certain value of  $N_o$  (Fig. 21).

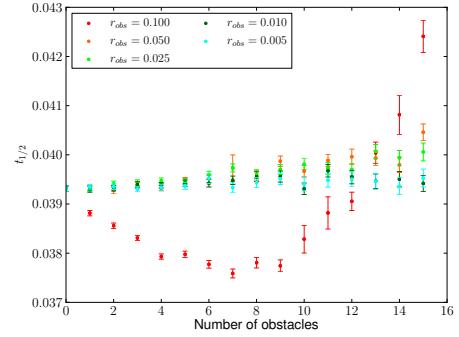


FIGURE 21 – Evolution of  $t_{1/2}$  with  $N_o$  for  $N_s = 5$  and  $N_t = 5$  for many radii

#### 2.1.2 Mobile obstacles

For the case where the obstacles are mobile, we only initialize such that all particles are totally in the domain and don't overlap each other. For this, we only use circular protective domains and semicircular sectors close to the boundary.

#### A problem of slowing down

When two or more particles are very close, with a distance closer than the double precision, a numerical slowing down appears. In this case, we have two possibilities to overcome the problem, when the distance  $d_{min}$  becomes smaller than  $d_{in}$ . We simulate

- a lattice move with small  $dx$  range where the time of steps is  $dt = \frac{dx^2}{4D} \dots$
- a Langevin move with small  $dt$  such that

$$x_{n+1} = x_n + \cos(\varphi) \sqrt{4D dt}$$

$$y_{n+1} = y_n + \sin(\varphi) \sqrt{4D dt}$$

where  $\varphi$  is an uniformly distributed random number between 0 and  $2\pi \dots$

$\dots$  until that  $d_{min}$  is larger than  $d_{out} > d_{in}$ .

For the lattice process if the distance between two particle's centers becomes smaller than  $2r_{part}$  or if the distance to the boundary is smaller than  $r_{part}$ , the last step is not accepted.

For the Langevin process, a reflection law is used when two particles collide : we find the collision time when  $d_{min}(t_c) = 0$ .

$$\sqrt{(x_1(t_c) - x_2(t_c))^2 + (y_1(t_c) - y_2(t_c))^2} = 2r_{part} \quad (2.3)$$

and we change the angle as  $\varphi \leftarrow \pi - \varphi + 2\theta$  with  $\theta$  the absolute angle between the two particles at  $t_c$ . The same work is done to find a possible collision for each particle with the boundary when  $d_b(t_b) = 0$ .

$$\min(x(t_b), 1 - x(t_b), y(t_b), 1 - y(t_b)) = r_{part} \quad (2.4)$$

and the angle is changing as  $\varphi \leftarrow \pi - \varphi$  for vertical boundaries and  $\varphi \leftarrow 2\pi - \varphi$  for horizontal boundaries.

For numerical simulations, I took  $d_{out} = 10$   $d_{in} = 100$   $dx$  and for  $dt$  the corresponding value to  $dx$ .

The two methods give the same probability density of the leaving time  $\rho_b(t)$  and the same probability density of the mass point distance  $\rho_f(R)$  traveled between reaching  $d_{in}$  and

reaching  $d_{out}$  but the first one requires twice more computation time than the second one.

For two particles, which diffuse in  $\mathbb{R}^2$ , there is an analytic expression for the temporal probability density of reaching the distance  $d_{out}$ , when having started in distance  $d_{in}$ . This probability is nothing else than the probability density  $\rho_b$  of the annulus protective domain (Sec. 3) for  $a = 2r_{part}$  and  $b = d_{out}$ .

There is also an analytic expression for the probability density of the mass point position which is  $\rho_\infty(\vec{R}, t)$  with a diffusivity  $D = D_1 + D_2$ . So, the probability density of the leaving position of the mass point  $\vec{R}$ , knowing that  $d_{out}$  is reached at  $t$ , is  $\rho_f(R|t) = \rho_\infty(\vec{R}, t) \cdot \rho_b(t)$ . The probability density of the distance traveled by the mass point is

$$\rho_f(R) = \int_{t=0}^{\infty} 2\pi R \rho_\infty(R, t) \rho_b(t) dt \quad (2.5)$$

In the Fig. 22 and 23, we compared these results to our simulation scheme.

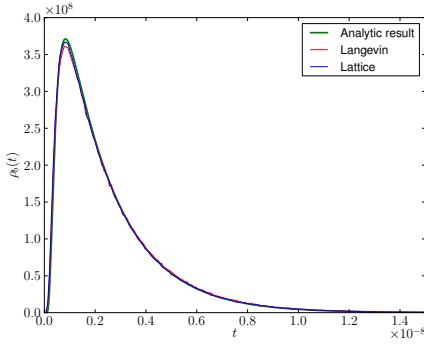


FIGURE 22 –  $\rho_b(t)$  obtained from different methods

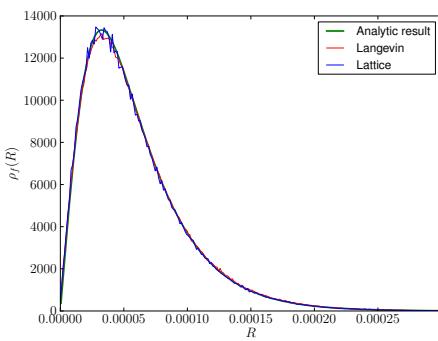


FIGURE 23 –  $\rho_f(R)$  obtained from different methods

## Results for one target and one searcher in a square

The mean collision time seems to decrease with the area occupied by mobile obstacles (Fig. 24). The computation of 100000 samples requires a lot of computational time which explains the error bars of the figure.

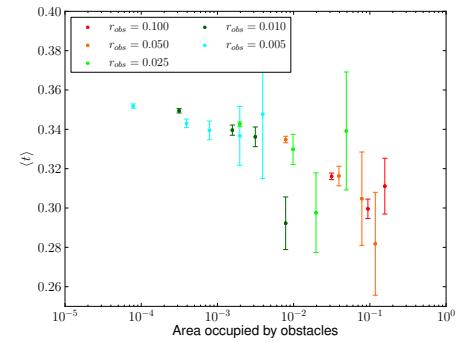


FIGURE 24 – Mean collision time vs area occupied by obstacles for many radii

### 2.1.3 Intermittent ballistic transport

For the intermittent ballistic transport problem, we use the algorithm shown in the section 1.4 for the collision condition. When a searcher is in ballistic motion, if it met an obstacle like a domain boundary, it changes directly his motion to the diffusive one.

The mean collision time is computed for many values of  $k_b$  and  $k_d$  for a diffusivity  $D = 10^{-3}$  and a velocity  $\vec{v} = \{\cos \varphi, \sin \varphi\}$  where  $\varphi$  is an uniformly distributed random number between 0 and  $2\pi$ . We see a minimum of  $\langle t \rangle$  in the space  $(k_d, k_b)$  at  $(175, 33)$  with a relative value 0.361 compared to the only-diffusive one (Fig. 25).

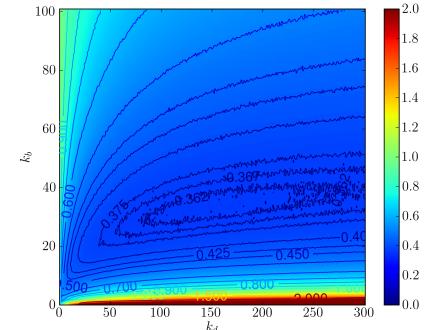


FIGURE 25 –  $\langle t \rangle$  values in the space  $(k_d, k_b)$  relative to the value obtained with an only-diffusive motion limited between  $[0, 2]$

We look for now the minimum's evolution and the evolution of its depth with the number of obstacles in the domain. The numerical simulation of the Fig. 25 takes one week, so for the moment, I don't have these evolutions.

## 2.2 Reaction process

For the reaction problem, we look for the time of targets' finding by searchers with the FPKMC algorithm shown in the section 1.3 where the step (5) is "Remove the target if the smallest distance from a searcher is smaller than  $\delta_r$  during at least a time of  $\delta_t$ ". All results present in this section will be done with the first kind of obstacles' initialization.

## Protective domains

To do this, we use circle protective domains centered in the particle's position with a radius of :

-  $0.5(d_{min} - \delta_r)$  if  $d_{min} > \delta_r$  out of the reactive domain. This permit to find the time of entering in the reactive domain.

-  $0.5(\delta_r - d_{min})$  if  $d_{min} < \delta_r$  in the reactive domain. In this case an overlap of protective domain is allowed because it gives exactly the same probability than for nonoverlapping protective domains (using a lattice motion or a Langevin motion as explain in the section 2.1.2) but the computational time is shorter.

To permit the particle to enter in the reactive domain without any slowing down, we need to overlap a little bit the protective domains and the reactive domain (Fig. 26).

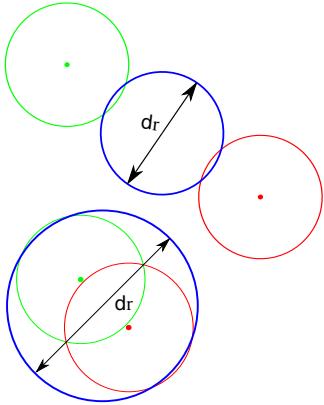


FIGURE 26 – Scheme out and in the reactive domain

For smaller overlaps, the simulation scheme becomes more and more exact. We looked for an overlap value, which doesn't influence the result anymore (Fig. 27). For next results, I use  $\delta_r = 10r_{part}$  and an overlap of  $10^{-3}\delta_r$ .

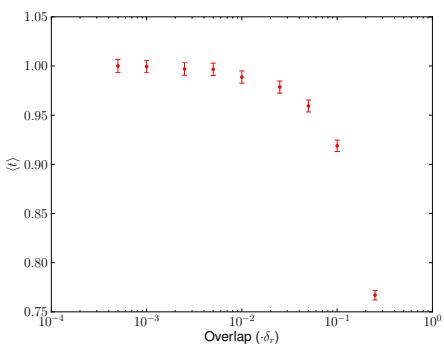


FIGURE 27 – Evolution of  $\langle t \rangle$  with the value of overlap for  $N_o = 0$

## Results for one target and one searcher in a square

For a small  $\delta_t$ , the evolution of the mean reaction time versus the area occupied by obstacles is the same as in the scenario of collision discussed in Sec. 2.1. In fact, as soon as the target enter in the reactive domain of the searcher, it is removed. And due to the fact that the expectative time to traveled in  $\delta_r$  which is  $\frac{\delta_r^2}{4D}$  is small compared to  $\langle t \rangle$ ,  $\langle t \rangle$  doesn't change.

For larger  $\delta_t$ , the mean reaction time increases more for small number of obstacles than for big number of obstacles due to the fact that the reaction time  $\delta_r$  play a significant role only when the two particles are already very close and so have no more influence from obstacles. So, we see an inversion of the evolution of  $\langle t \rangle$  versus the area occupied by obstacles which pass from an increasing one to a decreasing one (Fig. 29 and 30). This last result is in relation with the experimental results.

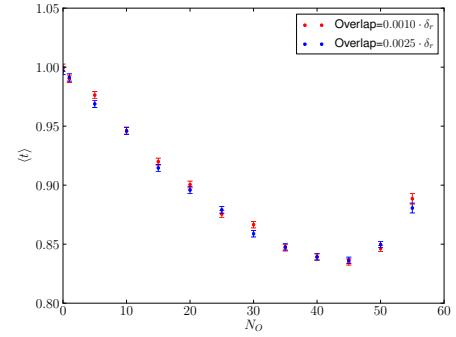


FIGURE 28 – Evolution of  $\langle t \rangle$  with  $N_o$  for different values of overlap for  $\delta_t = 0.001$

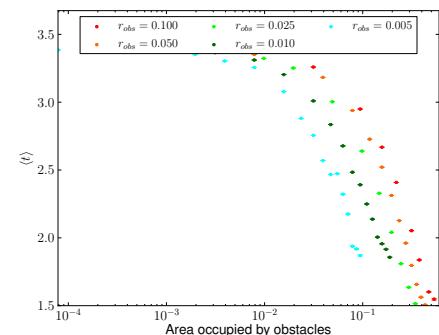
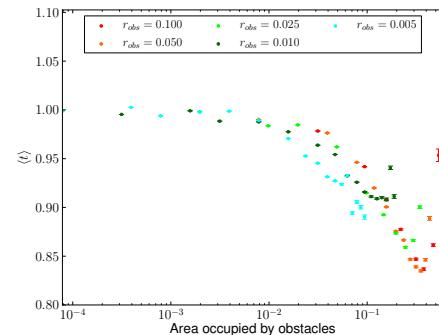
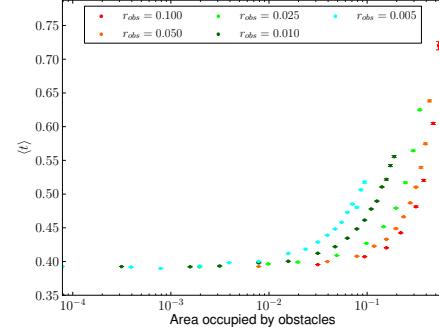


FIGURE 29 – Mean reaction time vs area occupied by obstacles for many radii and  $\delta_t = \{0.0005, 0.001, 0.005\}$

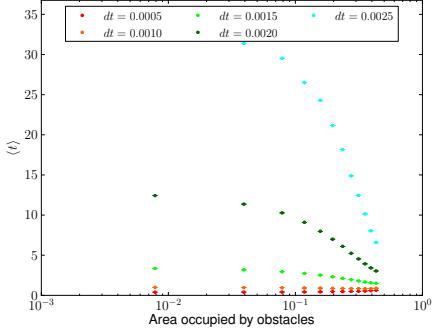


FIGURE 30 – Mean reaction time vs area occupied by obstacles for  $r_{obs} = 0.05$  and many  $\delta_t$

### Results for many targets and many searchers

The time evolution of  $N_t$  is similar as in the Sec. 2.1.1 and we define  $t_{1/2}$  as in eq. 2.2. We see for short  $\delta_t$ , the presence of a minimum as in the collision scenario and for bigger  $\delta_t$ , only a decreasing of  $t_{1/2}$  with the number of obstacles (Fig. 31).

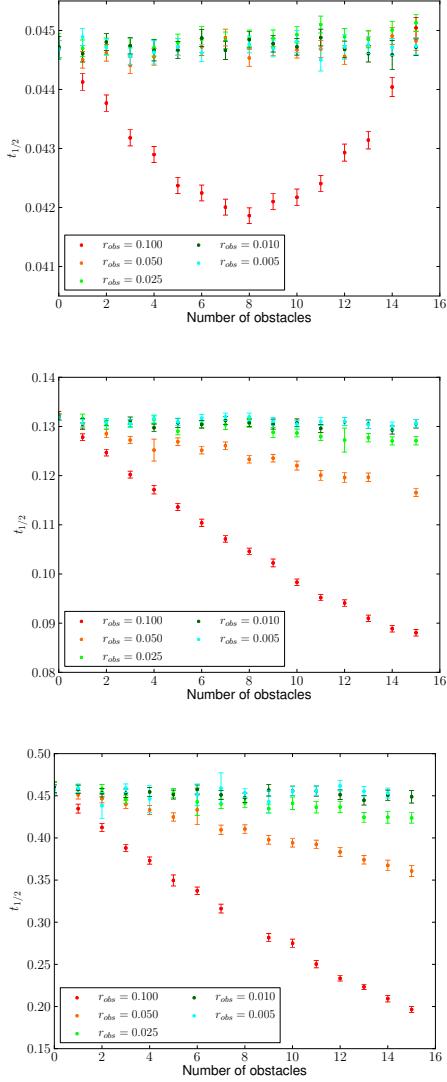


FIGURE 31 – Evolution of  $t_{1/2}$  with  $N_o$  for  $N_s = 5$  and  $N_t = 5$  for many radii and  $\delta_t = \{0.0005, 0.001, 0.005\}$  in a square

### 3 Annulus

A good possibility to replace the small hops around obstacles and the little imprecision due to the sector protective domain, is to compute the solution of the diffusion equation for the annulus protective domain (Fig. 32). The boundary close to the obstacle of radius  $a$  is reflecting (in red) and the other boundary is absorbing (green).

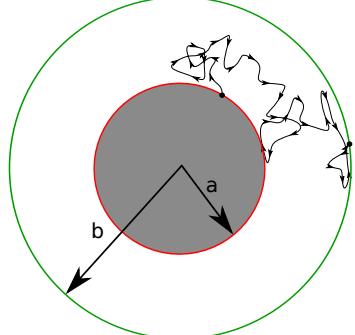


FIGURE 32 – Annulus protective domain

The particle is initially ( $t=0$ ) at  $r=a=1$  and  $D=1$  for dimensionless equation. The probability to be at  $(r, \varphi)$  at time  $t$  is

$$P_D(r, \varphi, t) = \frac{\pi}{4} \sum_{n \in \mathbb{Z}} \cos(n\varphi) \sum_{\alpha_n} \frac{J_n(\alpha_n b)^2 C_n(r) \alpha_n^2 e^{-\alpha_n^2 t}}{(1 - \frac{n^2}{\alpha_n^2}) J_n(\alpha_n b)^2 - J'_n(\alpha_n)^2} \quad (3.1)$$

with

$$C_n(r) = J'_n(\alpha_n) Y_n(\alpha_n r) - Y'_n(\alpha_n) J_n(\alpha_n r) \quad (3.2)$$

and  $\alpha_n$  such that

$$J'_n(\alpha_n) Y_n(\alpha_n b) - Y'_n(\alpha_n) J_n(\alpha_n b) = 0 \quad (3.3)$$

As shown in the section 1.3, we can find all other probabilities thanks to this last one.

The covering function will be calculated for  $b \in \{1.2, 1.4, 1.6, 1.8, 2\}$  due to the difficulty to calculate every-time the coefficient of the series depending on  $b$ .

#### 3.1 Probability density $\rho_b(t)$

The surviving probability at the time  $t$ , i.e. the probability that the particle has not reached the absorbing boundary at time  $t$  is :

$$S(t) = \int_{r=a}^b \int_{\varphi=0}^{2\pi} P_D(r, \varphi, t) r dr d\varphi \quad (3.4)$$

$$S(t) = b \sum_{\alpha_0} \frac{\pi J_0(\alpha_0 b)^2 [Y_1(\alpha_0) J_1(\alpha_0 b) - J_1(\alpha_0) Y_1(\alpha_0 b)]}{J_0(\alpha_0 b)^2 - J_1(\alpha_0)^2} e^{-\alpha_0^2 t} \quad (3.5)$$

We can deduce from  $S$ , the probability density to reach the absorbing boundary at time  $t$  :

$$\rho_b(t) = b \sum_{\alpha_0} \alpha_0^2 \frac{\pi J_0(\alpha_0 b)^2 [Y_1(\alpha_0) J_1(\alpha_0 b) - J_1(\alpha_0) Y_1(\alpha_0 b)]}{J_0(\alpha_0 b)^2 - J_1(\alpha_0)^2} e^{-\alpha_0^2 t} \quad (3.6)$$

## Calculation of $\rho_b$

The sum converge more or less fast depending on the time  $t$ . For large times, a small number of addends is required for a small relative error. We found for all times the number of addends necessary to have a relative error of the order of  $10^{-15}$  to be in the range of the double precision in C++, and not a larger one due to a larger consuming of time to evaluate  $\rho_b$  which doesn't increase the precision of the value.

For shorter times, we need a large number of addends. We found an expression of  $\rho_b$  for short times with an interpolation in

$$\frac{P(t)}{\sqrt{\pi t^3}} e^{-\frac{\beta}{4t}} \quad (3.7)$$

with  $P$  a polynom in  $t$  and  $\beta$  a positive real.

The limit time between these two ways to obtain an expression of  $\rho_b$  is taken such that the maximum number of addends in the sum is 30.

## Calculation of covering functions

For the interval  $b=1.2$ , we obtain the covering function shown in Fig. 33 with the rejection rate shown in Fig. 34 for a constant area  $Q = 0.05$  which gives  $N = 24$  and  $k = 1.2$ .

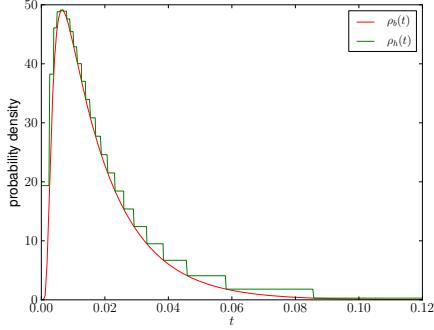


FIGURE 33 –  $\rho_b$  for  $b = 1.2$  and its corresponding covering function

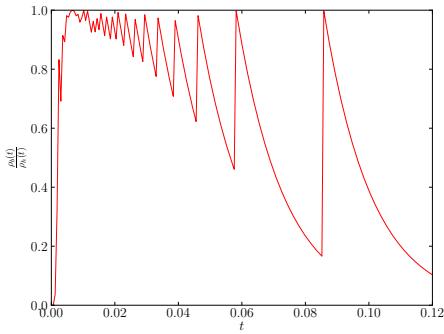


FIGURE 34 – Rejection rate for the construction of Fig. 33

For the C++ library, we cover all five functions for different  $b$  (quite similar after renormalization shown in Fig. 35) with only one step function  $\rho_h$ , which uses less computation memory, i.e. we cover the function

$$\max_{b \in \{1.2, 1.4, 1.6, 1.8, 2\}} \frac{\rho_b(\frac{t}{t_{max}})}{\rho_{b,max}}$$

where  $t_{max}$  is the time of the maximum of  $\rho_b$  and  $\rho_{max}$ , its value.  $\rho_h$  was constructed for  $Q = 0.0016$  which gives  $N = 627$  and  $k = 1.01$ . So, on average, the candidate time will be rejected only in 1% of the cases.

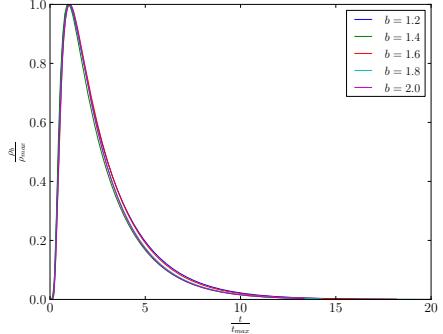


FIGURE 35 – Renormalization of  $\rho_b$

## 3.2 Probability density $\rho_n(r|t)$

First, we want to sample the no passage position  $(r, \varphi)$  at time  $t$  by sampling first the no passage radius  $r$  according to  $\rho_n(r|t)$  and then the no passage angle  $\varphi$  at time  $t$ , knowing  $r$ , according to  $\rho_n(\varphi|r, t)$  which is not actually done...

The probability to be at the radius  $r$ , under the condition of not having reached  $r = b$  yet is

$$\rho_n(r|t) = \frac{r}{S(t)} \int_{\varphi=0}^{2\pi} P_D(r, \varphi, t) d\varphi \quad (3.8)$$

$$\rho_n(r|t) = \frac{r \sum_{\alpha_0} \alpha_0 \frac{J_0(\alpha_0 b)^2 C_0(r)}{J_0(\alpha_0 b)^2 - J_1(\alpha_0)^2} e^{-\alpha_0^2 t}}{b \sum_{\alpha_0} \frac{J_0(\alpha_0 b)^2 [Y_1(\alpha_0) J_1(\alpha_0 b) - J_1(\alpha_0) Y_1(\alpha_0 b)]}{J_0(\alpha_0 b)^2 - J_1(\alpha_0)^2} e^{-\alpha_0^2 t}} \quad (3.9)$$

## Calculation of $\rho_n$

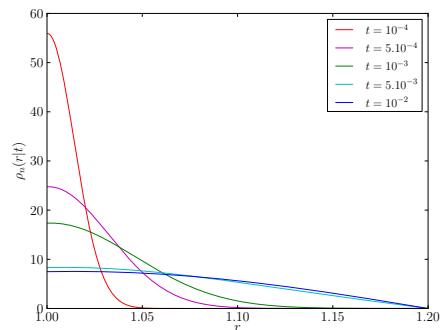


FIGURE 36 –  $\rho_n$  for  $b = 1.2$  at many times

For large times, the series converges as fast as in the case of  $\rho_b$ . Hence, we determine the necessary number of addends to have a relative error of  $10^{-15}$ .

For shorter times, we have

$$\forall t < t_0, \forall r \in [1, b] \rho_{n,b=1.2}(r|t) \sim \rho_{n,b=1.4}(r|t) \quad (3.10)$$

which permit to approximate  $\rho_n$  with the expression with smaller b which needs less addends for the same relative error.

For the case b=1.2, we need to find an approximate function for times shorter than  $10^{-3}$  like

$$\rho_n(r|t) \sim \rho_{n,1}(r|t) + \rho_{n,2}(r|t) \quad (3.11)$$

$$\text{with } \rho_{n,1}(r|t) = N(t)e^{-\frac{(r-\gamma(t))^2}{4t}} \quad (3.12)$$

a, normed in time, gaussian approximation of  $\rho_n$  and

$$\rho_{n,2}(r|t) = \beta(t)(e^{-\sigma(t)r^2} - 1)e^{-\alpha(t)r^2} \quad (3.13)$$

an approximation of  $\rho_n - \rho_{n,1}$ .  $\gamma$ ,  $\beta$ ,  $\sigma$  and  $\alpha$  are positive and are calculated with a polynomial interpolation of the real probability density. The relative error between  $\rho_n$  and magnitude of this approximation is  $10^{-3}$ .

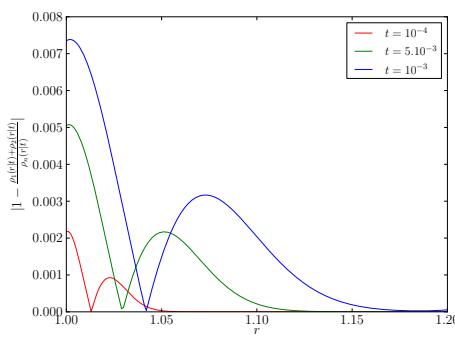


FIGURE 37 –  $\rho_n$  relative error for short times

So, we have the next time development for b=2.0 :

<b>t</b>	[0, 7.10 <sup>-4</sup> ]	[7.10 <sup>-4</sup> , 0.002]	[0.002, 0.0075]
$\rho$	$\rho_{n,1} + \rho_{n,2}$	$\rho_{n,1.2}$	$\rho_{n,1.4}$
<b>t</b>	[0.0075, 0.017]	[0.017, 0.03]	[3.10 <sup>-2</sup> , $\infty$ ]
$\rho$	$\rho_{n,1.6}$	$\rho_{n,1.8}$	$\rho_{n,2.0}$

For others b, the same time limits are used but only an approximation with probability densities with smaller b is available. This give for example  $b = 1.2$  :

<b>t</b>	[0, 7.10 <sup>-4</sup> ]	[7.10 <sup>-4</sup> , $\infty$ ]
$\rho$	$\rho_{n,1} + \rho_{n,2}$	$\rho_{n,1.2}$

### Calculation of covering functions

To do the rejection sampling with a covering function, we need to cut the time in intervals  $[t_i, t_{i+1}]$  such that

$$\int_{r=1}^b \max_{t \in [t_i, t_{i+1}]} \rho_n(r|t) dr < 1.02 \quad (3.14)$$

From these results, we have the time intervals : [0.00099, 0.00107, 0.00116, 0.00126, 0.00136, 0.00147, 0.00159, 0.00172, 0.00186, 0.00202, 0.00219, 0.00238, 0.0026, 0.00285, 0.00314, 0.00349, 0.00392, 0.00448, 0.00527, 0.00665, 30] for  $b = 1.2$ . For these intervals of time, the covering function is a step-function which covers  $\max_{t \in [t_i, t_{i+1}]} \rho_n(r|t)$  with  $N \sim 600$  which gives  $k \sim 1.02$ .

For shorter times, we cover the probability density with a function whose primitive is easy to inverted like

$$\rho_h(r, t) = \alpha(t)(r - \beta(t))e^{-\frac{(r-\beta(t))^2}{\sigma(t)^2}} \quad (3.15)$$

with  $\beta$  and  $\sigma$  calculated to have the same average and the same variance than  $\rho_n$  and  $\alpha$  a function which assures that  $\rho_h > \rho_n$  for all  $(r, t)$ .

An other way to do this, is cover  $\rho_n$  in 2D which given  $(r, \varphi)$  directly by the conservation of the volume instead of the area.

### 3.3 Probability density $\rho_f(\varphi|t)$

The probability to be at  $(b, \varphi)$  knowing that at t, the particle reached the boundary for first time is

$$\rho_f(\varphi|t) = -\frac{\partial_r P_D(r = b, \varphi, t)}{\rho_b(t)} \quad (3.16)$$

$$\rho_f(\varphi|t) = \frac{1}{2\pi} + \sum_{n=1}^{\infty} \cos(n\varphi) \frac{\sum_{\alpha_n} e^{-\alpha_n^2 t} \frac{\alpha_n^2 J_n(\alpha_n b)^2 C'_n(b)}{(1 - \frac{n^2}{\alpha_n^2}) J_n(\alpha_n b)^2 - J'_n(\alpha_n)^2}}{\pi \sum_{\alpha_0} e^{-\alpha_0^2 t} \alpha_0^2 \frac{J_0(\alpha_0 b)^2 C'_0(b)}{J_0(\alpha_0 b)^2 - J_1(\alpha_0)^2}} \quad (3.17)$$

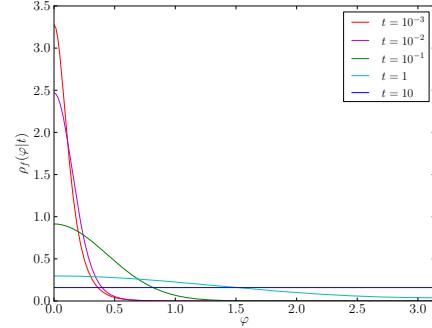


FIGURE 38 –  $\rho_f$  for  $b = 1.2$  at many times

The computation of  $\rho_f(\varphi|t)$  is similar to the case of  $\rho_n(r|t)$ , but  $\rho_f$  converges for  $t \rightarrow 0$  to a smooth function (Fig. 38), obtained with a Taylor expansion as

$$\rho_f(\varphi|t \rightarrow 0) = \sum_{m=0}^4 \sum_n a_{m,n} \cos(n\varphi) t^m \quad (3.18)$$

with  $a_{m,n}$  calculated with  $\rho_n$  derives at  $t=0$ . For this probability density, it's not necessary to do an interpolation.

Covering functions are obtained by the same way, i.e. we need to cut the time in intervals  $[t_i, t_{i+1}]$  such that

$$\int_{\varphi=0}^{2\pi} \max_{t \in [t_i, t_{i+1}]} \rho_f(\varphi|t) d\varphi < 1.02 \quad (3.19)$$

and after cover the function  $\max_{t \in [t_i, t_{i+1}]} \rho_f(\varphi|t)$ . For short times, we cover the Taylor expansion to do the rejection sampling.

## 4 Discrete domain

Another way to have results of reaction-diffusion is to use a model in a discrete domain. This way requires generally more computational time to have results with more incertitudes due to a convergence of results with a small step-size. The first section 4.1 will show results of similar problems than the section 2.1 which permit to compare the evolution and the magnitude of results. The second section 4.2 will show results with a new kind of particle : bystanders, which not react like obstacles but which accelerate the move of searchers and obstacles, which are inactive searchers, close to them.

### 4.1 Without bystanders

In this section, we want to study the evolution of mean collision time with the number of obstacles. Two different models will be investigated, first a model where particles and obstacles move on lattice sites without any idea of radius, and a second one with a radius proportional to the lattice step-size.

#### 4.1.1 Particles on lattice sites

In this case, there are  $N_s$  searchers looking for  $N_t$  targets, in a  $L \times L$  lattice with reflecting boundary conditions, with  $N_o$  obstacles. The probability to move in a neighbour lattice site is  $1/4$ . If the neighbour site is already occupied, the particle doesn't move.

#### Immobile obstacles

For one searcher and one targets, the mean collision time increases with the number of obstacles. When the proportion of sites occupied by obstacles becomes about one third of the total number of sites of the lattice, the mean collision time decreases. This happens due to the rejection of initial configurations where the target and the searcher will not find due to the obstacles initialization (Similar situation to Fig. 17). For short concentrations of obstacles, the mean collision time increases with the lattice size  $L$  at fixed concentrations. This is due to the fact that everything happens like obstacles and particles have a radius  $L^{-2}$ .

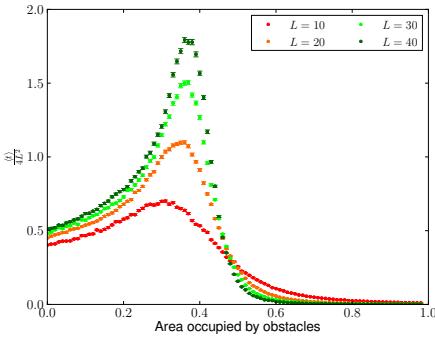


FIGURE 39 – Evolution of  $\langle t \rangle$  with the occupation of sites by obstacles for  $N_s = 1$  and  $N_t = 1$

For many searchers and many targets, the half-life time  $t_{1/2}$  define in Eq. 2.2 increases with the number of obstacles. In this case, there is no minimum.

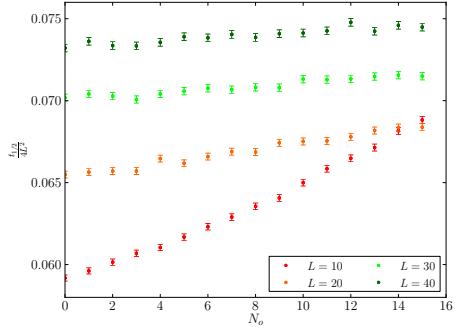


FIGURE 40 – Evolution of  $t_{1/2}$  with  $N_o$  for  $N_s = 5$  and  $N_t = 5$

#### Mobile obstacles

For one searcher and one targets, the mean collision time increases with the number of obstacles. There is no maximum as in the immobile case due to the fact that the searcher finds the target whatever the initial configuration.

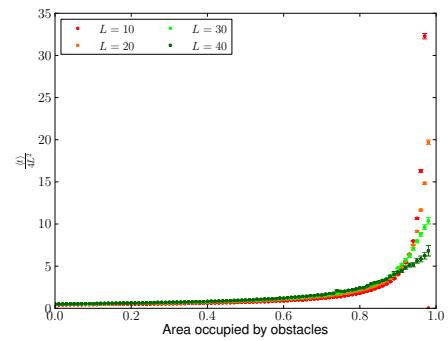


FIGURE 41 – Evolution of  $\langle t \rangle$  with the occupation of sites by obstacles for  $N_s = 1$  and  $N_t = 1$

#### 4.1.2 Particles with radius moving on discrete positions

In this model, there are  $N_s$  searchers looking for  $N_t$  targets, in a  $L \times L$  lattice with reflecting boundary conditions, with  $N_o$  obstacles. We take  $L = 1000$ , so we have  $dx = 0.001$  and we take the radius of particle as  $r_{part} = 5dx = 0.005$ . We simulate results with a radius of obstacles  $r_{obs} = \{5, 10, 25, 50, 100\}dx$  like in section 2.1. The probability to move in a neighbour lattice site is  $1/4$ . If the distance between two particles becomes smaller than  $2r_{part}$  then the last step will be rejected. The global time is nothing else than  $\frac{N}{4L^2}$ , with  $N$  the number of time-steps.

#### Immobile obstacles

For one searcher and one targets, the mean collision time involves like in Sec. 2.1.1 (Fig. 42).

For many searchers and many targets, the evolution of the half-life time  $t_{1/2}$  with  $N_o$  seems to present a minimum no really precise due to fluctuations (Fig. 43).

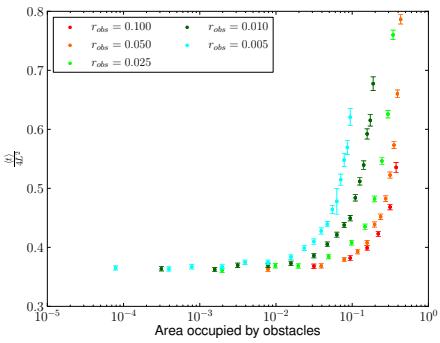


FIGURE 42 – Mean collision time vs area occupied by obstacles for many radii for  $N_s = 1$  and  $N_t = 1$

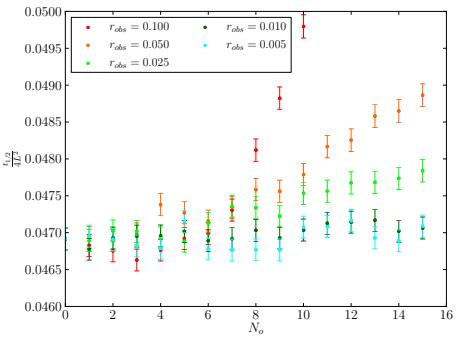


FIGURE 43 – Evolution of  $t_{1/2}$  with  $N_o$  for  $N_s = 5$  and  $N_t = 5$  for many radii

## Mobile obstacles

For one searcher and one targets, the mean collision time seems to increase with the number of obstacles, which is different to the result shown in Sec. 2.1.2. This can be produced by the fact that  $dx$  is not enough small.

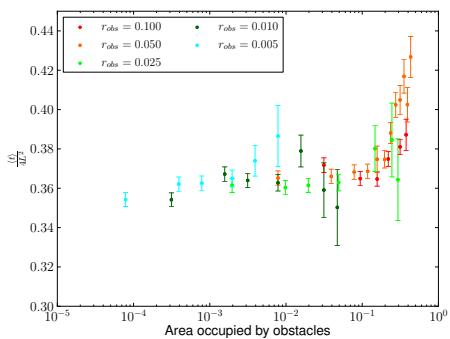


FIGURE 44 – Mean collision time vs area occupied by obstacles for many radii for  $N_s = 1$  and  $N_t = 1$

## 4.2 With bystanders

Two scenarios for the introduction of bystanders will be discussed : one consists to have bystanders on the lattice, which permit the interaction with searchers and obstacles if they are on one of neighbour sites (Sec. 4.2.1). The second one consists to have bystanders out of lattice sites which permit to

the bystander to change the moving probability of searchers and obstacles on the neighbour links (Sec. 4.2.2).

### 4.2.1 With bystanders on lattice sites

The lattice is partially occupied by four kinds of particles : searchers, targets, obstacles and bystanders. The lattice size is  $L \times L$  with, here,  $L=60$ , the smallest size which keep the good result : for  $L>60$ , results don't change. We use periodic boundary conditions.

#### The normal move

Each particle can move with a probability  $p$ . So each particle can move to a special neighbour site with a probability  $p/4$ . For a target, an obstacle or a bystander, if this special neighbour site is occupied by an other particle (whatever its kind), the target/obstacle/bystander stays in its previous site, otherwise, it moves to this special neighbour site. For a searcher, if this special neighbour site is occupied by a target, the target is removed and the searcher replaces it. If this special neighbour site is occupied by an other kind of particle (searcher, obstacle or bystander), the searcher stays on its previous site, otherwise it moves to the neighbour site.

#### The accelerated move

If a bystander occupies a neighbour site (whatever the number), the particle can move with a probability  $q \geq p$ . So, in this case, each particle can move to a neighbour site with a probability  $q/4$ . The moving of different particles is the same than before except the fact that if the special neighbour site is occupied by a bystander, the particle exchanges its position with the bystander. If nothing is precise,  $q = 1$  in the next sections.

#### Update rule

For each time step, the position of each particle is calculated with the precedent rules. We choose a uniformly distributed random number between 0 and 1. Depending on the values of  $p$  and  $q$  a classical "towersampling"-step is performed to determine whether the particle moves and if it moves, in which direction this will happen. The update is done sequentially (particle one by one) in next results but a random update (particle chosen randomly) gives same results.

#### Numbers and motion of different particles

In the following  $N_s$  denotes the number of searchers,  $N_t$  the number of targets,  $N_o$  the number of obstacles and  $N_b$  the number of bystanders. In this model, searchers are in every case moving with acceleration.

Problem	Target	Bystanders	Obstacles
<b>1</b>	Immobile	Immobile	Non Acc.
<b>2</b>	Immobile	Immobile	Acc.
<b>3</b>	Immobile	Mobile	Non Acc.
<b>4</b>	Immobile	Mobile	Acc.
<b>5</b>	Mobile	Immobile	Non Acc.
<b>6</b>	Mobile	Immobile	Acc.
<b>7</b>	Mobile	Mobile	Non Acc.
<b>8</b>	Mobile	Mobile	Acc.

$$N_b \in \{0, 1, 5, 10, 25, 50, 100, 200, 300\}$$

### Example of one sample

Case	$N_s$	$N_t$	$N_o$	Occupation
<b>20 :1</b>	200	100	1800	58.3% - 66.3%
<b>10 :1</b>	100	100	900	30.5% - 38.5%
<b>5 :1</b>	50	100	450	16.7% - 24.7%
<b>2.5 :1</b>	25	100	225	9.7% - 17.7%
<b>1 :1</b>	10	100	100	5.5% - 13.5%
<b>0.5 :1</b>	5	100	45	4.2% - 12.2%

### Obtention of results

For each sample, we choose a random position to each particle at the time  $t=0$  (according to the fact that a particle cannot have the same position as another one). At each time step,  $t \leftarrow t + 1$ , the position of each particle is updated. So, we can have the removing time of all targets :  $t_1, t_2, \dots, t_{N_b}$ .

With a big number of samples  $N_{sample}$  (about 100000 to have a relative precision of  $\pm 0.33\%$ ), we can plot the number of targets vs the time :  $N_t(t) = \text{average of not removed target number at } t \text{ on all samples}$  and the half-life time  $t_{1/2}$  is given by

$$N_t(t_{1/2}) = \frac{N_t(0)}{2} \quad (4.1)$$

### Example of a simulation

To illustrate the model before the interesting simulation, I want to show the result for  $L = 10$ ,  $N_s = 10$ ,  $N_t = 10$ ,  $N_o = 10$  and  $N_b = 10$ . With 100 000 samples, we obtain  $N_t(t)$  (Fig. 45) and  $t_{1/2} = 16$ .

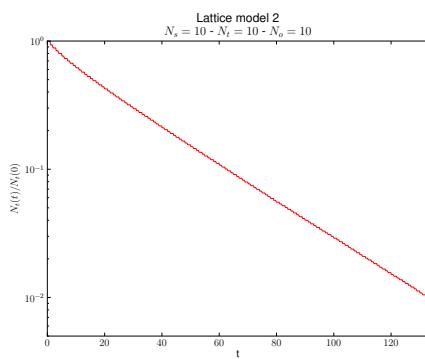


FIGURE 45 – Evolution of the number of targets with the time

So,  $N_t(t)$  involves like a decreasing exponential law :

$$N_t(t) = N_t(0)e^{-\lambda t} \text{ with } \lambda = \frac{\ln 2}{t_{1/2}} \quad (4.2)$$

like every no memory processes. So  $t_{1/2}$  is sufficient to characterize the process.

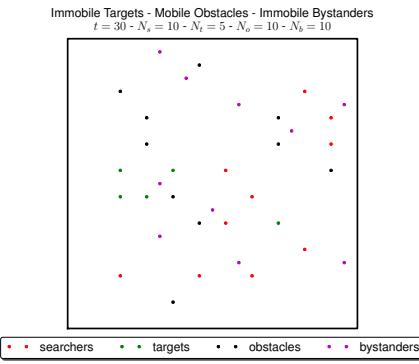


FIGURE 46 – Situation at time  $t=30$  for one sample for  $N_s = 10$ ,  $N_t = 10$ ,  $N_o = 10$  and  $N_b = 10$  at  $t=0$

In this sample,  $t_1 = 3$ ,  $t_2 = 4$ ,  $t_3 = 6$ ,  $t_4 = 8$ ,  $t_5 = 27$ ,  $t_6 = 41$ ,  $t_7 = 45$ ,  $t_8 = 86$ ,  $t_9 = 87$  and  $t_{10} = 168$ .

### Results with a probability p=0.5

With 100 000 samples and  $L=60$ , we obtain the functions  $N_t(t)$  (Fig. 47) and  $t_{1/2}(N_b)$  (Fig. 48).

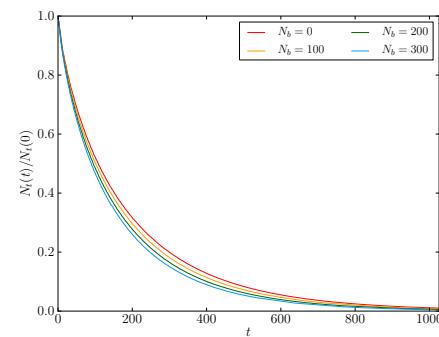


FIGURE 47 – Time evolution of  $N_t$  in the problem 2 for the case 10 :1

The half-life time  $t_{1/2}$  is always decreasing with the number of bystanders more or less fast depending on the problem. The same kind of evolution happens for other concentrations.

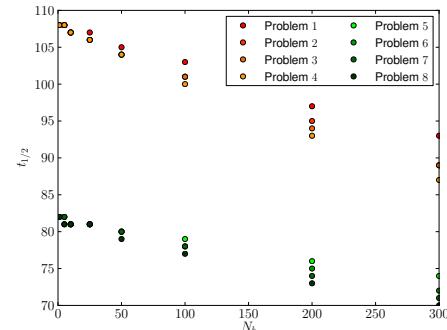


FIGURE 48 – Evolution of  $t_{1/2}$  with  $N_b$  in all problems for the case 10 :1

The value of q (the probability to move with acceleration)

takes values between  $p=0.5$  and 1. The value of  $t_{1/2}$  is optimized for  $q = 0$ .

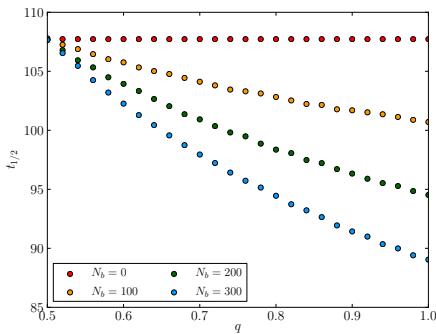


FIGURE 49 – Evolution of  $t_{1/2}$  with  $N_b$  and the probability  $q$  in the problem 2 for the case 10 : 1

#### 4.2.2 With bystanders out of lattice sites

In this model, bystanders are no longer on the lattice sites. They don't move and only influence the probability to move on a particular link which are computed at  $t=0$ . Targets are immobile and searchers and obstacles are accelerated close to bystanders. The probability of different links can be taken in different ways : asymmetric probabilities in section 4.2.2.1, symmetric probabilities independent on the number of neighbour bystanders in section 4.2.2.2 and symmetric probabilities dependent on the number of neighbour bystanders in section 4.2.2.3.

After the computation of probabilities after each initialization, we update at each time step the position of each particle (searchers and obstacles) according to the probability to stay, to go left, to go right, to go up and to go down. If the neighbour site is occupied, the particle stays at the previous position. If a searcher moves to a site occupied by a target, the target is removed and the searcher takes its place.

The time evolution of the targets number  $N_t(t)$  and the half-life time  $t_{1/2}$  are computed like in section 4.2.1.

##### 4.2.2.1 Asymmetric transition probabilities

###### Computation of transition probabilities

The presence of one bystander close to a node of the lattice reduces the probability to stay on the node ( $0.5 \rightarrow 0$ ) and increases the probability of links which connect this node to another node near the bystander ( $0.125 \rightarrow 0.375$ ). The probability of links which permit to leave bystanders (i.e. to reach a node not near to a bystander) is the same as the probability of links of a node not near to a bystander ( $0.125$ ). In this model, searchers and obstacles are moving faster along the bystanders and bystanders have no influence on links not near to them.

We can have three different situations for each node close to a bystander : 2 links close to a bystander (probability 0.375), 3 links close to a bystander (probability 0.2917) and 4 links close to a bystander (probability 0.25) to respect the normalization of transition probabilities (Fig. 50).

If the node is not close to a bystander, for each link the transition probability is 0.125 and the probability to stay is 0.5.

Similar to the precedent model, we define :  $p$ ="probability to move on a node not close to a bystander" and  $q$ ="probability to move on a node close to a bystander". We have always  $p \leq q \leq 1$ .

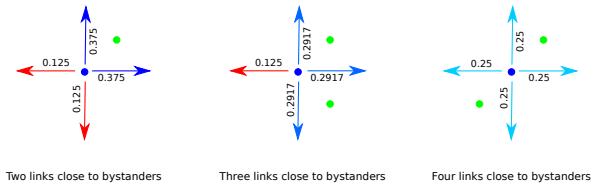


FIGURE 50 – Transitions of a node close to a bystander

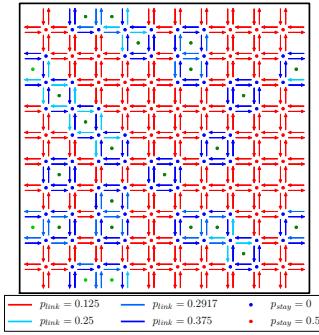


FIGURE 51 – Example of transition probabilities on a lattice

###### Results with a probability to move without acceleration $p=0.5$

With 100 000 samples and  $L=60$ , we obtain the functions  $N_t(t)$  (Fig. 52) and  $t_{1/2}(N_b)$  (Fig. 60). The half-life time decrease less faster than in the first model and the smaller  $t_{1/2}$  is obtained

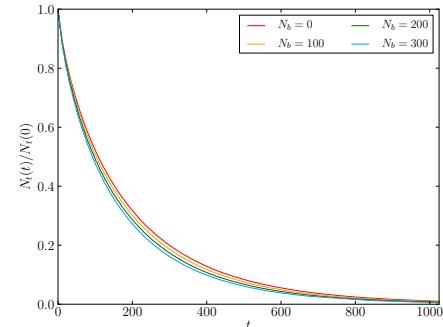


FIGURE 52 – Time evolution of  $N_t$  for the case 10 : 1

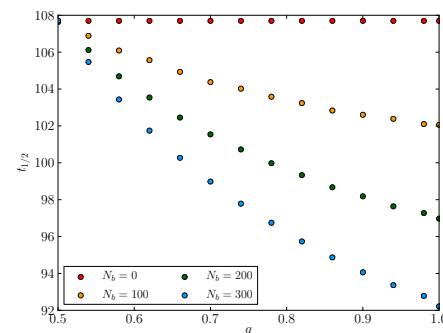


FIGURE 53 – Evolution of  $t_{1/2}$  with  $N_b$  and the probability  $q$  for the case 10 : 1

#### 4.2.2.2 Symmetric transition probabilities : independent on the number of bystanders

##### Computation of transition probabilities

In this case, the probability to move is 0.125 for each direction if there is no bystander close to the link and 0.25 if there are one or two bystanders close to the link. The probability to stay is computed with the normalization of probabilities.

Similar to the precedent models, we define :  $p$ ="probability to move on a node not close to a bystander" and  $q$ ="probability to move on a node close to a bystander" such that the probability to move on a link close to a bystander is  $0.25 \cdot q$ . We have always  $p \leq q \leq 1$ .

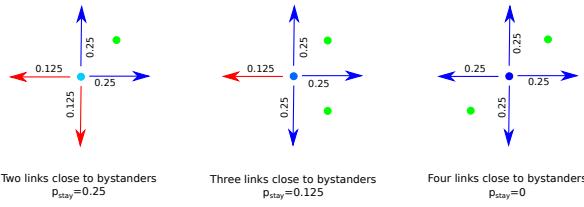


FIGURE 54 – Transitions of a node close to a bystander

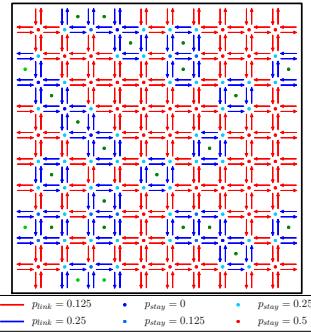


FIGURE 55 – Example of transition probabilities on a lattice

#### Results with a probability to move without acceleration $p=0.5$

With 100 000 samples and  $L=60$ , we obtain the functions  $N_t(t)$  (Fig. 56) and  $t_{1/2}(N_b)$  (Fig. 60).

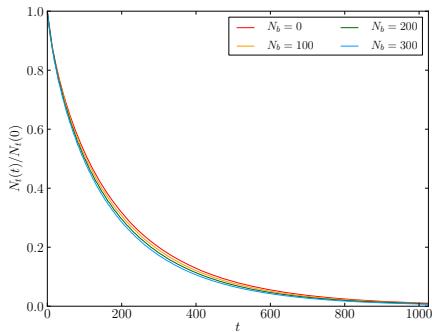


FIGURE 56 – Time evolution of  $N_t$  for the case 10 : 1

#### 4.2.2.3 Symmetric transition probabilities : dependent on the number of bystanders

##### Computation of transition probabilities

In this case, the probability to move is 0.125 for each direction if there is no bystander close to the link,  $3/16$  if there is one bystander close to the link and 0.25 if there are two bystanders close to the link. The probability to stay is computed with the normalization of probabilities.

Similar to the precedent models, we define :  $p$ ="probability to move on a node not close to a bystander" and  $q$ ="probability to move on a node close to a bystander" such that the probability to move on a link close to a bystander is  $q/16$  by the presence of each bystanders. We have always  $0 \leq q \leq 1$ , when  $q=0$ , we have a non accelerated probabilities for each node.

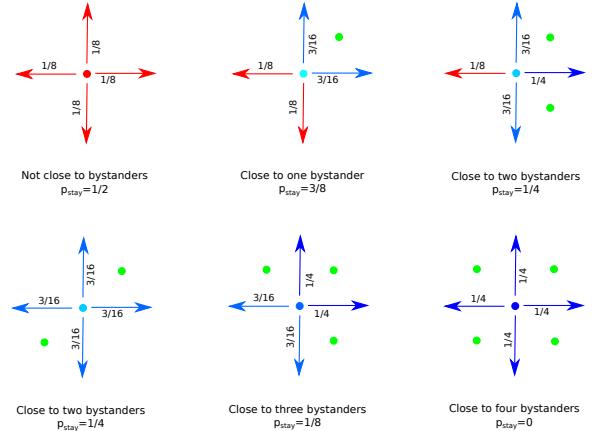


FIGURE 57 – Transitions of a node close to a bystander

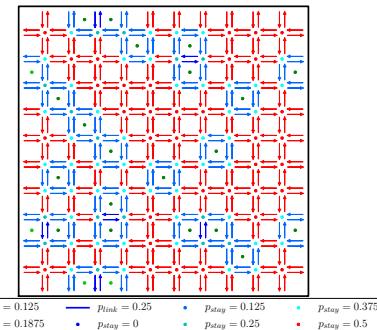


FIGURE 58 – Example of transition probabilities on a lattice

#### Results with a probability to move without acceleration $p=0.5$

With 100 000 samples and  $L=60$ , we obtain the functions  $N_t(t)$  (Fig. 59) and  $t_{1/2}(N_b)$  (Fig. 60).

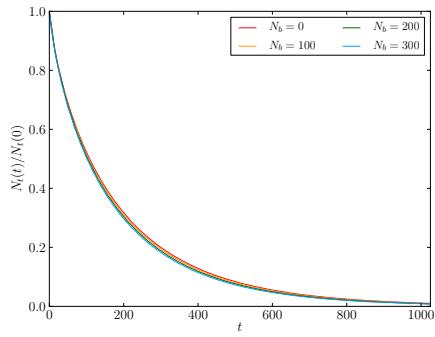


FIGURE 59 – Time evolution of  $N_t$  for the case 10 : 1

#### 4.2.3 Comparison of $t_{1/2}$ for different concentrations

The model 1 corresponds to Sec. 4.2.1, the model 2.1 to Sec. 4.2.2.1, the model 2.2 to Sec. 4.2.2.2 and the model 2.3 to Sec. 4.2.2.3. The half-life time always decreases with the number of bystanders, more pronounced for the first model due to the presence of bystanders in lattice sites which decreases the number of sites available for searchers and targets. In the second model, the decreasing is more pronounced for a smallest possible value of  $p_{stay}$  close to bystanders and for a value of  $p_{link}$  independent on the number of neighbour bystanders.

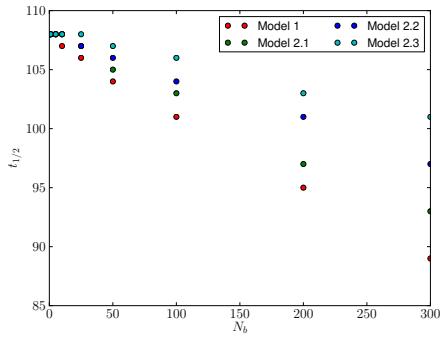


FIGURE 60 – Evolution of half-life time with the number of bystanders

The evolution is similar for all others concentration with a coefficient  $c^{\alpha:1}$  such that  $t_{1/2}^{\alpha:1} = c^{\alpha:1} t_{1/2}^{10:1}$ .

$\alpha$	20	5	2.5	1	0.5
$c^{\alpha:1}$	1.85	0.48	0.225	0.085	0.04

## Conclusion

The most efficient strategy to find targets in a two-dimensional domain in presence of immobile obstacles is to have a number of searchers similar to the number of obstacles. A large radius of obstacle is better than a small one for

a same area occupied by obstacles. If the searchers needs to be a certain time  $\delta_t$  close to targets favorises a large number of obstacles for an efficient search in the case of large  $\delta_t$ . For mobile obstacles, a large number of obstacles decreases the reaction time. The introduction in the domain of particles which accelerated the motion of searchers and obstacles is favorable.

This internship permits me to learn the different algorithm shown in Sec. 1 and the way to use them in C++ codes with an efficient computation time, through the first passage and the discrete methods. In this internship, I learned also to use the C++ library of protective domains computed by K. Schwarz and Y. Schröder and to create some of these functions.

## Acknowledgments

First of all, I want to thank M. Rieger and all persons of the laboratory to receive me for these six months and for their presence around me during this time. Then, I thanks a lot Karsten Schwarz for his help and his disponibility every days of my internship face to problems encountered and results obtained.

## References

- [1] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics* 22, pages 403–434, April 21, 1976.
- [2] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The journal of Physical Chemistry*, Vol 8, No 25, pages 2340–2361, May 12, 1977.
- [3] Daniel T. Gillespie. Monte carlo simulation of random walks with residence time dependant transition probability rates. *Journal of Computational Physics* 28, pages 395–407, November 4, 1977.
- [4] Tomas Oppelstrup, Vasily V. Bulatov, Aleksandar Donev, Malvin H. Kalos, George H. Gilmer, and Babak Sadigh. First-passage kinetic monte carlo method. *Physical Review E* 80,066701, pages 1–14, December 1, 2009.
- [5] Karsten Schwarz and Heiko Rieger. Efficient kinetic monte carlo method for reaction-diffusion problem with spatially varying annihilation rate. *Journal of Computational Physics* 237, pages 396–410, November 2012.
- [6] Jeroen S. van Zon and Pieter Rein ten Wolde. Green’s-function reaction dynamics : A particle-based approach for simulating biochemical networks in time and space. *The journal of chemical physics* 123, 234910, pages 1–16, December 22, 2005.