Analytical and Numerical Study of the Mosquito Capture Phenomenon in the Diffusion Equation

Aya Chaieb[†], Baptiste Petiot[†], Antonin Decouvelaere[†], Lucas Bourret[†], Mathieu Prioux[†],
Facundo Muñoz[‡]

February 19, 2026

Abstract

We consider a deterministic model of spatial diffusion describing the evolution of the surface density of mosquitoes. This model is formulated as a parabolic partial differential equation augmented with terms accounting for the natural mortality of mosquitoes and their capture by traps. This modeling framework is part of the study of population control strategies, in particular the SIT method (*Sterile Insect Technique*).

We first analyze the possibility of obtaining explicit solutions in simplified configurations, as well as analytical approximations based on a truncation of the Duhamel series. The numerical resolution of the model is then addressed using different numerical methods, including finite differences, finite elements, and finite volumes. These methods are compared in terms of accuracy, robustness, and computational cost, and the selected scheme appears to be the most efficient among those studied.

This numerical framework is finally used to perform parameter inference of the model from data, leading to estimated values that are physically consistent. The objective of this work is thus to provide a coherent mathematical and numerical framework for the analysis, simulation, and parameter estimation of spatial mosquito population control strategies. This study is part of a broader project conducted by the CIRAD.

[†] Student-researchers in the engineering program at CentraleSupélec, Université Paris-Saclay.
[‡] CIRAD, UMR ASTRE, F-34398 Montpellier, France.
ASTRE, Univ Montpellier, CIRAD, INRAE, Montpellier, France.

Contents

1 Introduction **3**
- 1.1 Reminder of the overall context of the study . . . 3
- 1.2 Objective and methodology . . . 3

2 Analytical Study of the Diffusion Equation with Capture **4**
- 2.1 Diffusion and Uniform Mortality: Exact Solutions . . . 4
  - 2.1.1 Heat Kernel and Pure Diffusion . . . 4
  - 2.1.2 Addition of Natural Mortality . . . 6
- 2.2 Inhomogeneous Model: Mathematical Exactness and Physical Invalidity . . . 6
  - 2.2.1 Integral Solution via Duhamel's Formula . . . 6
  - 2.2.2 Critical Analysis . . . 6
- 2.3 On the Non-Existence of an Exact Analytical Solution . . . 6
  - 2.3.1 Formal Spectral Reduction . . . 7

3 Definition of the Parameters and Discretization for the Numerical Resolution **7**
- 3.1 The Master Equation . . . 7
- 3.2 Definition of the Physical Variables . . . 7
  - 3.2.1 Identifiability of the parameters of interest . . . 8
- 3.3 Initial Condition and Conservation . . . 8

4 Finite Difference Method **8**
- 4.1 Discretization for the Finite Difference Method (FDM) . . . 9
- 4.2 Analysis of algorithmic complexity . . . 9
  - 4.2.1 Cost per time step . . . 9
  - 4.2.2 Impact of the CFL condition on time . . . 9
  - 4.2.3 Overall complexity . . . 9
- 4.3 Implementation of the method and results . . . 9
- 4.4 Analysis of results . . . 11

5 Finite Element Method **11**
- 5.1 Definition of the triangular mesh . . . 11
- 5.2 Study of the ODE system . . . 12
- 5.3 Complexity and accuracy . . . 13
- 5.4 Implementation of the method and results . . . 13
- 5.5 Analysis of results . . . 14

6 Finite Volume Method **14**
- 6.1 Summary - Recall of Finite Volume Modeling . . . 14
  - 6.1.1 Mesh and Geometry . . . 14
  - 6.1.2 Spatial and Temporal Discretization . . . 14
- 6.2 Algorithm Complexity and Accuracy . . . 14
- 6.3 Results . . . 15
- 6.4 Spatial Distribution and Diffusion . . . 15
- 6.5 Trap Efficiency Analysis . . . 15
- 6.6 Trap-Adapted Mesh . . . 15
  - 6.6.1 Mesh Generation Method . . . 15
  - 6.6.2 Results with an Adapted Mesh . . . 16

7 Inference of the physical parameters of our model from real capture data **19**
- 7.1 Motivation . . . 19
- 7.2 Identifiability of the physical parameters . . . 19
  - 7.2.1 Justification of the identifiability issue . . . 19
  - 7.2.2 Applied solution: functional constraint . . . 19

7.2.3 Identifiability of the other parameters 20
7.3 Optimization strategy and minimization algorithm 21
7.3.1 Formulation of the cost function 21
7.3.2 Justification of the L-BFGS-B algorithm 21
7.3.3 Principle of the L-BFGS-B algorithm 21
7.4 Optimization results and analysis 21
7.5 Conclusion on parameter inference using the optimization method 22

8 Conclusion **23**

Appendices **24**

A Code **24**

B Data **24**

1 Introduction

In this article, we continue the study initiated in Chaieb et al., 2025.

1.1 Reminder of the overall context of the study

Our study aims to model the dispersion dynamics of sterile mosquitoes within the framework of the SIT method (*Sterile Insect Technique*). This technique seeks to regulate certain mosquito populations that constitute a major epidemiological concern (Dengue, Zika, Chikungunya). It is based on the sterilization of male mosquitoes using X-rays or γ-rays. The sterile mosquitoes are then released and contribute to reducing the fertility of female mosquitoes. In the implementation of this technique, a detailed understanding of the spatial dynamics of mosquitoes represents a central challenge.

1.2 Objective and methodology

This work extends previous research on the calibration of the probabilistic mosquito diffusion model (Capel and Muñoz, 2024), conducted by CIRAD.

We have developed a macroscopic model, which is the deterministic limit of the probabilistic model governed by the Itô equation: the reaction-diffusion equation. After studying the well-posedness of this equation and establishing the existence of solutions in the first article Chaieb et al., 2025, we now aim to further investigate the expression of these solutions using several analytical and numerical methods and to assess their relevance. Particular attention is given to the computational cost of each approach, with the objective of identifying the most suitable method for intensive use in inverse simulations or Bayesian inference.

Our goal is to calibrate these models as accurately as possible in order to extend their application to the estimation of parameters of interest in real-world situations. To this end, we will use data generated from discrete simulations (stochastic model).

2 Analytical Study of the Diffusion Equation with Capture

We consider the following diffusion equation, defined on the plane:

$$\begin{cases} \partial_t \rho(\mathbf{r},t) = D\Delta\rho(\mathbf{r},t) - \nu\rho(\mathbf{r},t) - K(\mathbf{r})\rho(\mathbf{r},t), & \mathbf{r} \in \mathbb{R}^2, \\ \rho(\mathbf{r},0) = \rho_0(\mathbf{r}). \end{cases} \tag{1}$$

The function $\rho(\mathbf{r},t) \in L^2(\mathbb{R}^2,(0,T))$ represents the surface density of mosquitoes and $\rho_0(\mathbf{r}) \in L^2(\mathbb{R}^2)$ the associated initial mosquito distribution, both expressed in mosquitoes$\cdot\,\mathrm{m}^{-2}$. The coefficient $D > 0$ is the (constant) diffusivity, with unit $\mathrm{m}^2 \cdot \mathrm{day}^{-1}$; $\nu > 0$ is the uniform natural mortality rate; and $K(\mathbf{r}) \geqslant 0$ models mosquito capture by traps. The last two quantities are expressed in day^{-1}. The function K is intentionally left in implicit form, unlike in the following sections, in order to remain as general as possible at this stage.

The objective of this section is to determine in which cases one can obtain:

- exact analytical solutions,
- usable integral representations,
- controlled approximations.

Particular attention is paid to the physical relevance of the models under consideration.

2.1 Diffusion and Uniform Mortality: Exact Solutions

We begin with the case without traps, $K := 0$, that is, homogeneous diffusion of mosquitoes throughout space, analogous to heat diffusion. This first step forms the analytical foundation of the entire study in this section.

2.1.1 Heat Kernel and Pure Diffusion

The equation

$$\partial_t \rho = D\Delta\rho$$

admits, in any dimension, an explicit solution expressed using the heat kernel. In dimension 2, it is given by

$$G(\mathbf{r},t) = \frac{1}{4\pi D t} \exp\left(-\frac{\mathbf{r}^2}{4Dt}\right).$$

For any initial datum ρ_0, the solution is given by the convolution

$$\rho(\mathbf{r},t) = (G * \rho_0)(\mathbf{r},t).$$

This solution preserves the total mass and remains positive.

We now provide the proof of this result in dimension 1; the two-dimensional solution is obtained in a perfectly analogous manner. We define the Cauchy problem as follows:

$$\begin{cases} \partial_t \rho(x,t) = D\partial_{xx}\rho(x,t), \\ \rho(x,0) = N_0\delta(x). \end{cases} \tag{2}$$

We use the spatial Fourier transform:

$$\mathcal{F}\{\rho(x,t)\} = \int_{-\infty}^{+\infty} \rho(x,t) e^{-ikx} dx.$$

We assume that the solution belongs to the Schwartz space, defined by:

$$\mathcal{S}(\mathbb{R}) = \{\, f \in C^\infty(\mathbb{R}) \mid \forall m,n \in \mathbb{N},\ \sup_{x\in\mathbb{R}} |x^m \partial_x^n f(x)| < \infty \,\}.$$

Then

$$\mathcal{F}\{\partial_{xx}\rho\}(k,t) = -k^2\hat{\rho}(k,t),$$

and

$$\mathcal{F}\{\delta\}(k) = 1.$$

Thus, Problem (2) becomes:

$$\begin{cases} \partial_t\hat{\rho}(k,t) = -Dk^2\hat{\rho}(k,t), \\ \hat{\rho}(k,0) = N_0. \end{cases} \tag{3}$$

This is a simple ordinary differential equation in time, whose solution is

$$\hat{\rho}(k,t) = N_0 e^{-Dk^2t}.$$

Taking the inverse Fourier transform:

$$\begin{aligned} \rho(x,t) &= \frac{1}{2\pi}\int_{-\infty}^{+\infty}\hat{\rho}(k,t)e^{ikx}dk \\ &= \frac{N_0}{2\pi}\int_{-\infty}^{+\infty}e^{ikx}e^{-Dtk^2}dk. \end{aligned}$$

We write

$$ikx - Dtk^2 = -Dt\left(k - i\frac{x}{2Dt}\right)^2 - \frac{x^2}{4Dt}.$$

Thus

$$\rho(x,t) = \frac{N_0}{2\pi}e^{-\frac{x^2}{4Dt}}\int_{-\infty}^{\infty}\exp\left[-Dt\left(k - i\frac{x}{2Dt}\right)^2\right]dk.$$

Setting $u = k - i\frac{x}{2Dt}$ gives $du = dk$, so

$$\rho(x,t) = \frac{N_0}{2\pi}e^{-\frac{x^2}{4Dt}}\int_{-\infty}^{\infty}\exp\left[-Dtu^2\right]du.$$

Using the identity $\int_{-\infty}^{+\infty}e^{-\alpha x^2}dx = \sqrt{\frac{\pi}{\alpha}}$, we obtain

$$\rho(x,t) = \frac{N_0}{\sqrt{4\pi Dt}}e^{-\frac{x^2}{4Dt}}.$$

More generally, defining the heat kernel

$$G(x,t) = \frac{1}{\sqrt{4\pi Dt}}e^{-\frac{x^2}{4Dt}},$$

we have for any initial distribution $\rho_0(x)$:

$$\rho(x,t) = (G * \rho_0)(x,t),$$

where the convolution is defined by

$$(G * \rho_0)(x,t) = \int_{-\infty}^{+\infty}G(x-\xi,t)\rho_0(\xi)\,d\xi.$$

By differentiation under the integral sign,

$$\partial_t\rho = \int(\partial_t G)(x-\xi,t)\,\rho_0(\xi)\,d\xi,$$

$$\partial_{xx}\rho = \int(\partial_{xx} G)(x-\xi,t)\,\rho_0(\xi)\,d\xi.$$

Since G satisfies $\partial_t G = D\partial_{xx}G$, it follows that

$$\partial_t\rho(x,t) = D\partial_{xx}\rho(x,t).$$

Moreover, as $t \to 0^+$, $G(\cdot,t) \to \delta$ in the sense of distributions, hence

$$\lim_{t\to 0^+}\rho(x,t) = \int\delta(x-\xi)\,\rho_0(\xi)\,d\xi = \rho_0(x).$$

2.1.2 Addition of Natural Mortality

We now consider

$$\begin{cases} \partial_t \rho(\mathbf{r}, t) = D\Delta\rho(\mathbf{r}, t) - \nu\rho(\mathbf{r}, t), \\ \rho(\mathbf{r}, 0) = \rho_0(\mathbf{r}). \end{cases} \tag{4}$$

Setting $\rho(\mathbf{r}, t) = e^{-\nu t}u(\mathbf{r}, t)$, we find that u satisfies the pure diffusion equation. Hence

$$\rho(\mathbf{r}, t) = e^{-\nu t}(G * \rho_0)(\mathbf{r}, t).$$

Using semigroup notation, one may write compactly

$$\rho(\mathbf{r}, t) = e^{tL}\rho_0(\mathbf{r}),$$

where $L = D\Delta - \nu$ (see Kato, 1995 for details on semigroup theory).

Physical interpretation: Natural mortality induces a uniform exponential decay of the density, independent of position. This term corresponds to an individual death process with constant rate, while diffusion only modifies the spatial distribution. This deterministic modeling of natural mortality could nevertheless be refined in future work.

2.2 Inhomogeneous Model: Mathematical Exactness and Physical Invalidity

Before treating the density-proportional capture term, it is instructive to examine the model

$$\partial_t \rho = D\Delta\rho - \nu\rho - K(\mathbf{r}),$$

where capture is no longer proportional to the density.

2.2.1 Integral Solution via Duhamel's Formula

Letting $L = D\Delta - \nu$, the solution can be written

$$\rho(t) = e^{tL}\rho_0 - \int_0^t e^{(t-s)L}K\, ds.$$

This expression is exact and, for certain forms of K, yields explicit solutions.

2.2.2 Critical Analysis

Despite its mathematical exactness, this model is physically inadequate. The additive term imposes extraction independent of the local density. Thus, even in a nearly empty region, a finite number of mosquitoes would be removed, potentially leading to negative densities.

Physically, the number of captured mosquitoes must be proportional to the number present. The additive term therefore fails to describe the real capture mechanism and must be rejected.

2.3 On the Non-Existence of an Exact Analytical Solution

We now consider the physically relevant equation

$$\partial_t \rho(\mathbf{r}, t) = D\Delta\rho(\mathbf{r}, t) - \nu\rho(\mathbf{r}, t) - K(\mathbf{r})\rho(\mathbf{r}, t), \qquad \mathbf{r} \in \mathbb{R}^2, \tag{5}$$

where capture is proportional to the local density.

2.3.1 Formal Spectral Reduction

Using formal separation of variables $\rho(\mathbf{r},t) = e^{-\lambda t}\psi(\mathbf{r})$, Equation (5) leads to the stationary Schrödinger-type problem

$$-D\Delta\psi(\mathbf{r}) + \big(\nu + K(\mathbf{r})\big)\psi(\mathbf{r}) = \lambda\psi(\mathbf{r}). \tag{6}$$

An exact analytical solution would therefore require explicit knowledge of the spectrum and eigenfunctions of

$$H = -D\Delta + \nu + K(\mathbf{r}).$$

Exactly solvable Schrödinger equations correspond to very specific potentials with rigid algebraic structures (e.g., harmonic, Coulomb, Morse,

3 Definition of the Parameters and Discretization for the Numerical Resolution

Following the numerical study of the resolution methods (see Part 1, Chaieb et al., 2025), we now focus on the precise physical modeling of the terms in the reaction-diffusion equation. We adopt the mechanistic-statistical framework developed by Nguyen et al. Nguyen et al., 2025, which makes it possible to relate the macroscopic parameters of the PDE to the individual behavior of mosquitoes.

3.1 The Master Equation

The dynamics of the mosquito population density, denoted $\rho(x,y,t)$, is governed by the following Fokker-Planck equation, which generalizes Equation (1):

$$\frac{\partial\rho}{\partial t} = \underbrace{D\Delta\rho}_{\text{Diffusion}} - \underbrace{\nu\rho}_{\text{Mortality}} \underbrace{- \rho \sum_{k=1}^{N_{traps}} f_k(x,y)}_{\text{Trap capture}} \tag{7}$$

Unlike the generic notations $K(r)$ used previously, we explicitly detail each biological term here in order to enable their statistical identification.

3.2 Definition of the Physical Variables

Diffusion coefficient (D or σ) : The spatial diffusion coefficient D is derived from Fick's law (see Chaieb et al., 2025). At the microscopic level, the individual volatility of a single mosquito is denoted by σ. The relationship between the macroscopic coefficient D (used in our FDM code) and the microscopic volatility σ is given by $D = \frac{\sigma^2}{2}$.

- Unit: $m^2 \cdot day^{-1}$.
- Interpretation: A high value indicates rapid dispersion of mosquitoes away from the release point x_0.

Natural mortality rate (ν) : The lifetime of an individual T is modeled as a continuous random variable following an exponential distribution. The parameter ν corresponds to the constant hazard rate of the event "natural death." The survival probability at time t follows an exponential distribution over time:

$$S(t) = \exp\left(-\int_0^t \nu\, du\right) = e^{-\nu t} \tag{8}$$

In the simulation, this rate is derived from the **daily survival probability** (pds) such that $\nu = -\ln(pds)$.

- Unit: day^{-1}.
- Physical interpretation: The mean life expectancy is $1/\nu$. This term acts as a uniform linear sink ($-\nu u$) in the FDM equation.

Trap capture function (f_k) : Capture is modeled as a competing risk that depends on position. For a trap k located at q_k, the function $f_k(x, y)$ does not represent a simple "zone," but rather the *conditional instantaneous hazard rate* of capture. In accordance with the theoretical derivations (see Capel and Muñoz, 2024), this hazard follows a Gaussian decay with respect to the squared distance:

$$f_k(x, y) = \gamma \cdot \exp\left(-\frac{\|(x, y) - q_k\|^2}{R_{trap}^2}\right) \tag{9}$$

Trap efficiency (γ) : This parameter represents the maximum capture rate at the center of the trap. It aggregates both the attractiveness of the lure and the mechanical performance of the device.

- Unit: day^{-1}.

Effective radius (R_{trap}) : It corresponds to the standard deviation of the Gaussian. At a distance R_{trap} from the center, the trap efficiency decreases to 37% of γ.

- Unit: *meter*.

3.2.1 Identifiability of the parameters γ and R_{trap}

The article Capel and Muñoz, 2024 highlights an important statistical issue: the identifiability of the parameters γ and R_{trap}. Indeed, it is difficult to distinguish between intensity and spatial range effects using capture data alone. A high number of captures may result either from a highly efficient but small trap or from a weakly efficient but very wide trap.

To better handle this uncertainty, we reparameterize the model so that the parameters correspond to known target probabilities: we use the probability of capture at the trap center, $pc0 = 0.8$, and the probability of capture at 5 m, $pc5 = 0.1$.

We then have:

$$\begin{cases} P(0) = \gamma = pc_0 \\ P(5) = \gamma \cdot e^{-\frac{5^2}{R_{trap}^2}} = pc_5 \end{cases} \tag{10}$$

Which yields:

$$\gamma = pc_0 \qquad \text{and} \qquad R_{trap}^2 = \frac{5^2}{\ln(pc0) - \ln(pc5)} \tag{11}$$

3.3 Initial Condition and Conservation

The mosquito release is modeled as a discrete Dirac mass. If N_{total} mosquitoes are released at (x_0, y_0), the grid is initialized such that:

$$\sum_{i,j} \rho_{i,j}^0 \Delta x^2 = N_{total} \tag{12}$$

In our modeling, this mass is distributed over a small Gaussian neighborhood with standard deviation R_0 in order to reduce initial numerical instabilities.

4 Finite Difference Method

This easy-to-implement method allows us to approximate a solution to equation (7).

4.1 Discretization for the Finite Difference Method (FDM)

The principle of the method is detailed in Part 1 (Chaieb et al., 2025). For the numerical resolution, the spatial domain $\Omega = [-L, L] \times [-L, L]$ is discretized into an $N \times N$ grid.

- **Discrete variables:** Let $\rho_{i,j}^n$ denote the approximation of the density at point (x_i, y_j) and at time t^n. The spatial and temporal steps are denoted $\Delta x = L/N$ and Δt.
- **Trap matrix (V_{trap}):** The set of capture functions is precomputed in a static matrix V_{trap} of size $N \times N$, avoiding the need to recompute exponentials at each time iteration:

$$(V_{trap})_{i,j} = \sum_{k=1}^{N_{traps}} \gamma \exp\left(-\frac{(x_i - x_{q_k})^2 + (y_j - y_{q_k})^2}{R_{trap}^2}\right) \tag{13}$$

- **Global numerical scheme:** The discrete evolution equation implemented in our Python code then becomes:

$$\rho_{i,j}^{n+1} = \rho_{i,j}^n + \Delta t \left[D(\Delta_h \rho)_{i,j}^n - (\nu + (V_{trap})_{i,j})\rho_{i,j}^n\right] \tag{14}$$

  where $(\Delta_h \rho)$ is the 5-point discrete Laplacian operator.

4.2 Analysis of algorithmic complexity

Let N_x be the number of grid points along one spatial dimension (the grid therefore contains $N_x \times N_x$ cells).

4.2.1 Cost per time step

At each time iteration, the explicit finite difference scheme performs matrix operations (computation of the Laplacian by convolution, pointwise multiplication for the reaction term). As established by Chaieb et al., the computation of the Laplacian and the source term at each point has a complexity proportional to the total number of cells (Chaieb et al., 2025).

$$\mathcal{C}_{spatial} = \mathcal{O}(N_x^2) \tag{15}$$

4.2.2 Impact of the CFL condition on time

The number of required time steps, denoted N_t, is not independent of N_x. To guarantee numerical stability, the time step Δt must satisfy the CFL (Courant-Friedrichs-Lewy) condition:

$$\Delta t \leq \frac{h^2}{4D + h^2 K_{max}} \approx \frac{h^2}{4D} \tag{16}$$

Since the spatial step is $h = L/N_x$, we have $h^2 \propto 1/N_x^2$. Consequently, to simulate the same physical duration T_{max}, the time step Δt must decrease quadratically as the resolution increases. The number of time iterations therefore follows:

$$N_t = \frac{T_{max}}{\Delta t} \propto N_x^2 \tag{17}$$

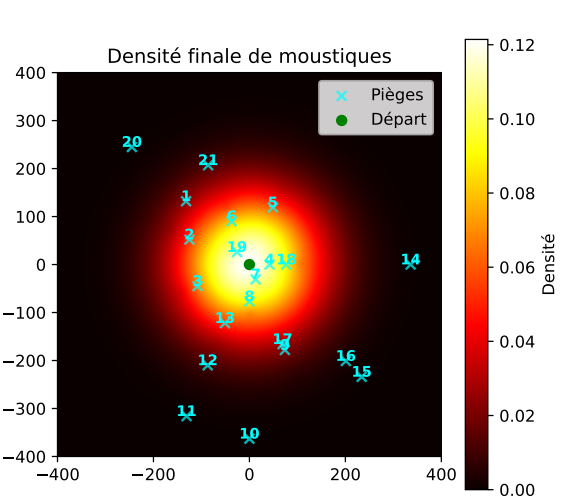
4.2.3 Overall complexity

The total algorithmic complexity of the simulation, denoted $\mathcal{C}_{total}$, is the product of the spatial cost and the number of time iterations:

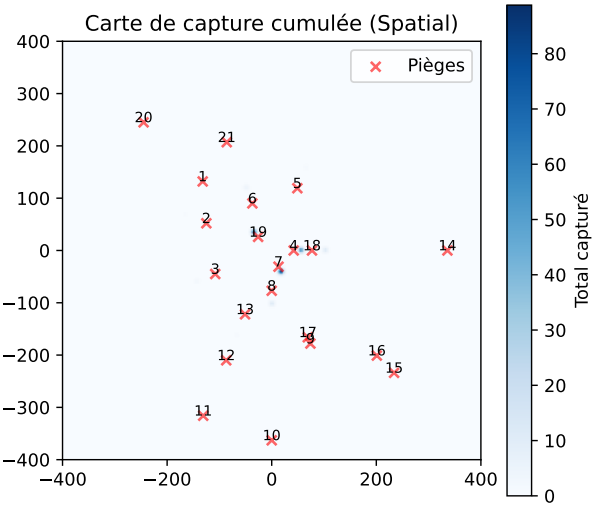
$$\mathcal{C}_{total} \approx \mathcal{C}_{spatial} \times N_t \approx \mathcal{O}(N_x^2) \times \mathcal{O}(N_x^2) = \mathcal{O}(N_x^4) \tag{18}$$

4.3 Implementation of the method and results

We implement the method using parameters derived from the probabilistic model, namely $N_{initial} = 50\,000$, $D = 150\text{m}^2 \cdot \text{day}$, $R_{trap}^2 = \frac{5^2}{\ln(pc0) - \ln(pc5)} \approx 12.02\text{m}^2$, $\gamma = 0.8\text{day}^{-1}$ and $\nu = 0.2\text{day}^{-1}$



(a) Mosquito density map after $T = 20$ days



(b) Density map of mosquitoes captured by traps

Figure 1: Simulation results with the hyperparameters of the target model

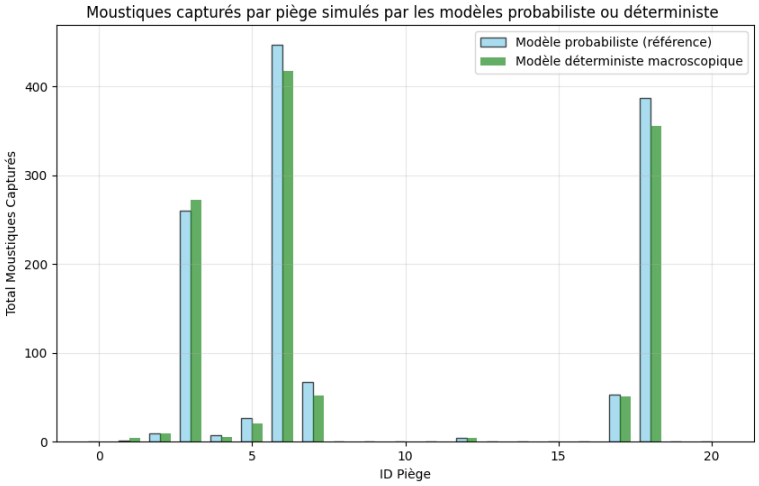


Figure 2: Comparison between the probabilistic model and the deterministic model (finite difference simulation)

4.4 Analysis of results

The simulation results 1 are consistent with the Fokker-Planck diffusion equation (7); the contributions of the traps are not visually apparent due to the relatively low capture rate compared to the number of mosquitoes dying from natural factors. Indeed, for 50 000 mosquitoes initially released, 48 292 die due to natural causes (contribution of the term $\nu\rho$) and 1 192 are captured by the traps. Figure 2 shows that our finite difference simulation reproduces the results of the probabilistic model consistently and with significantly lower computational time, highlighting the duality between the microscopic and macroscopic models. We can therefore conclude that our model is well calibrated.

5 Finite Element Method

The objective of this section is to numerically approximate a solution of the diffusion equation using the finite element method. This method offers several advantages compared to the finite difference method. First, it provides greater robustness with respect to the model parameters. Moreover, the finite element method allows the construction of an adapted mesh, locally refined in regions with high trap concentration, while guaranteeing convergence. The definition of the spaces, test functions, and the variational formulation are described in the article (Chaieb et al., 2025). In this study, we use a regular triangular mesh, independent of the presence of traps, in order to facilitate numerical implementation.

5.1 Definition of the triangular mesh

Let $h > 0$ be the discretization step and $\Omega = [-N, N] \times [-M, M]$ the diffusion domain. The grid of nodes is defined by:

$$N_h = \{(ih, jh) \mid i = -N, \dots, N,\ j = -M, \dots, M\} \tag{19}$$

where $(x_i, y_j) = (ih, jh)$ represents a mesh point.

For $i = -N, \dots, N-1$ and $j = -M, \dots, M-1$, we associate to each node the square

$$S_{i,j} = [ih, (i+1)h] \times [jh, (j+1)h] \tag{20}$$

which constitutes an elementary subdivision of Ω.

Each square $S_{i,j}$ is divided into two triangles (see Figure 3), defined by

$$T_{i,j}^{(1)} = \text{conv}\{(ih, jh),\ ((i+1)h, jh),\ ((i+1)h, (j+1)h)\} \tag{21}$$

$$T_{i,j}^{(2)} = \text{conv}\{(ih, jh),\ ((i+1)h, (j+1)h),\ (ih, (j+1)h)\} \tag{22}$$

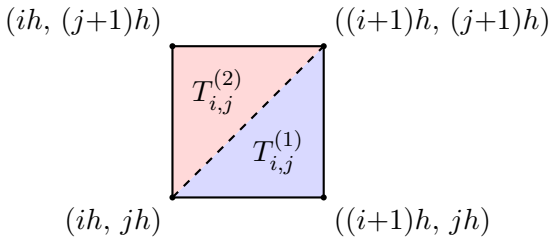


Figure 3: Subdivision of a square $S_{i,j}$ into two triangles $T_{i,j}^{(1)}$ and $T_{i,j}^{(2)}$

where

$$\begin{aligned} \text{conv} : \mathcal{P}(\mathbb{R}^d) &\longrightarrow \mathcal{P}(\mathbb{R}^d) \\ A &\longmapsto \text{conv}(A) \end{aligned}$$

denotes the mapping that associates to any finite set $A = \{x_1, \dots, x_m\} \subset \mathbb{R}^d$ its convex hull:

$$\mathrm{conv}(A) = \left\{ \lambda_1 x_1 + \cdots + \lambda_m x_m \;\middle|\; \lambda_i \geq 0,\ \sum_{i=1}^{m} \lambda_i = 1 \right\} \tag{23}$$

In other words, $\mathrm{conv}(A)$ is the smallest convex set containing A.

The complete triangular mesh[1] is therefore

$$\mathcal{T}_h = \bigcup_{i=0}^{N-1} \bigcup_{j=0}^{M-1} \{T_{i,j}^{(1)},\, T_{i,j}^{(2)}\} \tag{24}$$

5.2 Study of the ODE system

After projection onto the basis $(\varphi_{i,j})_{i,j}$, assembly leads to the system of ordinary differential equations

$$M\,\dot{U}(t) + (A + R)\,U(t) = 0 \tag{25}$$

where the matrices are defined by:

$$M_{(i,j),(k,l)} = \int_\Omega \varphi_{i,j}(x,y)\,\varphi_{k,l}(x,y)\,dxdy \tag{26}$$

$$A_{(i,j),(k,l)} = \int_\Omega D\,\nabla\varphi_{i,j}(x,y) \cdot \nabla\varphi_{k,l}(x,y)\,dxdy \tag{27}$$

$$R_{(i,j),(k,l)} = \int_\Omega K(x,y)\,\varphi_{i,j}(x,y)\,\varphi_{k,l}(x,y)\,dxdy \tag{28}$$

To study the resulting matrix ODE, we use the Crank–Nicolson scheme, which consists of approximating the time derivative by a centered difference and taking the average of the terms at t^n and t^{n+1} for the right-hand side.

The scheme is written as:

$$M\,\frac{U^{n+1} - U^n}{\Delta t} + \frac{1}{2}(A + R)\,(U^{n+1} + U^n) = 0 \tag{29}$$

Rearranging yields the following recurrence relation:

$$\left(M + \frac{\Delta t}{2}(A + R)\right) U^{n+1} = \left(M - \frac{\Delta t}{2}(A + R)\right) U^n \tag{30}$$

After numerical integration, we obtain the following coefficients for the mass, stiffness, and reaction matrices:

$$A = D \begin{pmatrix} 4 & -1 & \cdots & 0 & 0 \\ -1 & 4 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 4 & -1 \\ 0 & 0 & \cdots & -1 & 4 \end{pmatrix} \tag{31}$$

with 4 on the diagonal (integral involving twice the same test function) and -1 when computing the product integral between a test function at one node and that at a neighboring node sharing an edge.

For M, we obtain

$$M = \frac{h^2}{12} \begin{pmatrix} 6 & 1 & \cdots & 0 & 1 \\ 1 & 6 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 6 & 1 \\ 1 & 0 & \cdots & 1 & 6 \end{pmatrix} \tag{32}$$

with 6 on the diagonal (integral involving twice the same test function) and 1 when computing the product integral between a test function at one node and that at a neighboring node sharing at least one triangle (less restrictive condition).

[1] The choice of such a mesh is justified by its ease of implementation and its suitability for the problem.

5.3 Complexity and accuracy

Within the finite element framework, the complexity mainly depends on assembling the linear system (stiffness matrix and right-hand side vector), as well as solving it at each time step. For a regular mesh composed of N_x^2 triangular elements, the assembly phase has complexity $\mathcal{O}(N_x^2)$, while solving a sparse linear system of size $N_x^2 \times N_x^2$ is, in our case, of order $\mathcal{O}(N_x^3)$. The overall time complexity is therefore $\mathcal{O}(N_t \times N_x^3)$.

In terms of accuracy, linear finite elements ($\mathbb{P}_1$) on a regular mesh provide a second-order spatial approximation: the error is of order $\mathcal{O}(h^2)$, where h denotes the characteristic mesh size. In time, the use of the Crank–Nicolson method yields an error of order $\mathcal{O}(\Delta t^2)$.

5.4 Implementation of the method and results

We implement the method using parameters derived from the probabilistic model, namely $N_{initial} = 50\,000$, $D = 150$, $R_{trap}^2 = \frac{5^2}{\ln(pc0) - \ln(pc5)}$, $\gamma = 0.8$ and $\nu = 0.2$.

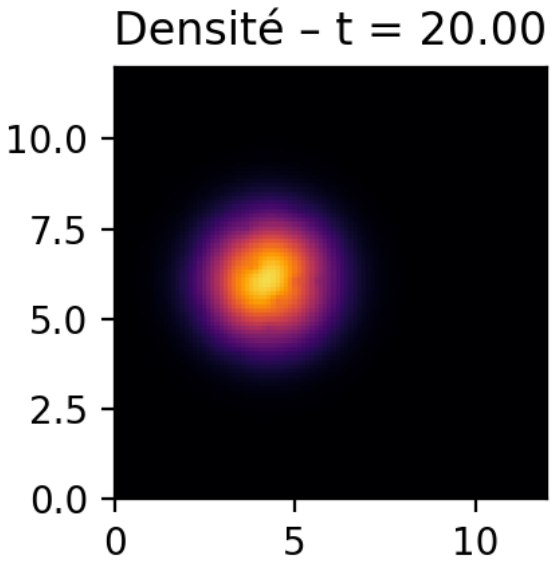


Figure 4: Final mosquito density map

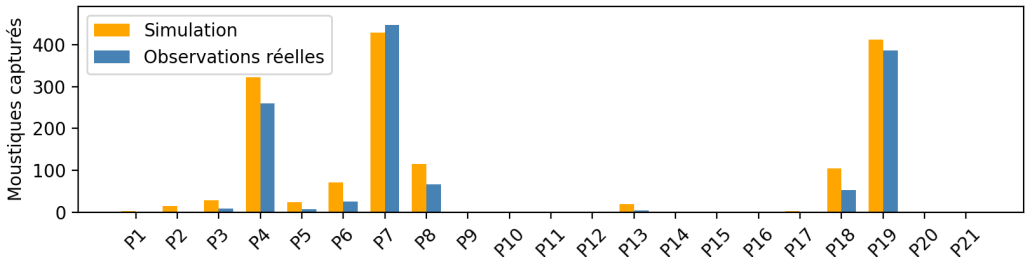


Figure 5: Comparison between the probabilistic model and the deterministic model obtained by finite element simulation

5.5 Analysis of results

We observe that our simulation using the finite element method reproduces the results of the probabilistic model consistently. However, certain parameter configurations yield better results than others. Moreover, the high computational time of the program (two to three minutes for a 100×100 grid) prevents a rigorous determination of optimal values through a gradient descent algorithm.

6 Finite Volume Method

6.1 Summary - Recall of Finite Volume Modeling

We revisit some theoretical aspects and definitions of the finite volume method (see Part 6 Chaieb et al., 2025).

The Finite Volume Method (FVM), conservative by construction, is based on the discretization of integral conservation laws on a mesh of the domain Ω.

6.1.1 Mesh and Geometry

The domain is partitioned into a set of convex cells (or control volumes), denoted C, forming an admissible mesh $\mathcal{T}$. Each cell is defined by its volume $|C|$ and a center $\mathbf{r}_C$. To ensure the consistency of numerical fluxes, the mesh must satisfy an orthogonality condition: the segment connecting the centers of two neighboring cells C and L must be orthogonal to their shared edge $e = (C|L)$.

6.1.2 Spatial and Temporal Discretization

By integrating the equation over each cell C and applying the Green-Ostrogradsky theorem, the mass balance is written as:

$$|C|\frac{d\rho_C}{dt} + \sum_{e\in\mathcal{E}_C} \Phi_e = -|C|(\nu + K_C)\rho_C \tag{33}$$

The diffusive flux Φ_e is approximated by: $\Phi_e \approx -D\frac{|e|}{d_{CL}}(\rho_L - \rho_C)$.

To ensure numerical stability with respect to reaction terms, we use an **implicit** Euler scheme. The density at time t^{k+1} is the solution of the linear system:

$$(\mathbf{V} - \Delta t D\mathbf{A} + \Delta t\mathbf{V}\mathbf{R})\,\boldsymbol{\rho}^{k+1} = \mathbf{V}\boldsymbol{\rho}^k \tag{34}$$

Where:

- $\mathbf{V}$ is the diagonal matrix of volumes $|C|$.
- $\mathbf{A}$ is the Laplacian matrix containing the transmissibilities $\frac{|e|}{d_{CL}}$.
- $\mathbf{R}$ is the diagonal matrix of cumulative loss rates: $R_{ii} = \nu + K(r_C)$.

6.2 Algorithm Complexity and Accuracy

In the context of the finite volume method, the complexity depends primarily on mesh generation and matrix assembly, as well as the computational loop for solving the linear system. The Delaunay triangulation has complexity $\mathcal{O}(N_x^2 \log N_x)$, the matrix generation and factorization has complexity $\mathcal{O}(N_x^3)$, and the simulation loop has complexity $\mathcal{O}(N_t * N_x^2 \log N_x)$. Thus, the overall time complexity amounts to $\mathcal{O}(N_t * N_x^2 \log N_x) + \mathcal{O}(N_x^2)$.

6.3 Results

The simulation was carried out on a square domain $L \in [-400, 400]$ discretized by an unstructured mesh generated via Delaunay triangulation ($N \approx 8100$ points). The mosquito population dynamics are governed by a diffusion-reaction equation accounting for:

- Diffusion ($D = 150$) simulating random movement.
- Natural mortality ($\lambda = 0.2$).
- Sink terms (traps) modeled by local Gaussian functions ($A = 0.8$, $\sigma = 3$).

The initial state corresponds to a dense concentration centered at $(0, 0)$ with radius 5.

At $t = 20$, the population transitions from the initial state (approx. 46,000 individuals) to a residual population, demonstrating the effectiveness of the trapping system over the considered time horizon.

6.4 Spatial Distribution and Diffusion

The density map (Figure 6a) illustrates the spreading of the mosquito cloud. The unstructured mesh allows the concentration gradients to be resolved correctly without favoring any axial direction. The local effect of the traps is clearly visible: low-density zones (in dark blue) appear around the traps. The center of the domain remains the densest area, as diffusion has not yet transported the entire population toward the traps.

6.5 Trap Efficiency Analysis

The capture histogram (Figure 6b) reveals strong heterogeneity in trap efficiency. This disparity is explained by the topology of the system:

Proximity to the source: Traps located close to the origin $(0, 0)$ (the center of the initial swarm) record the highest scores. They intercept the population before it disperses.

6.6 Trap-Adapted Mesh

The new code no longer places points uniformly at random. It uses a technique called Rejection Sampling to create an adaptive mesh: many small triangles are placed near the traps (for accuracy) and larger triangles are placed far from the traps (to save computation time in less critical areas).

6.6.1 Mesh Generation Method

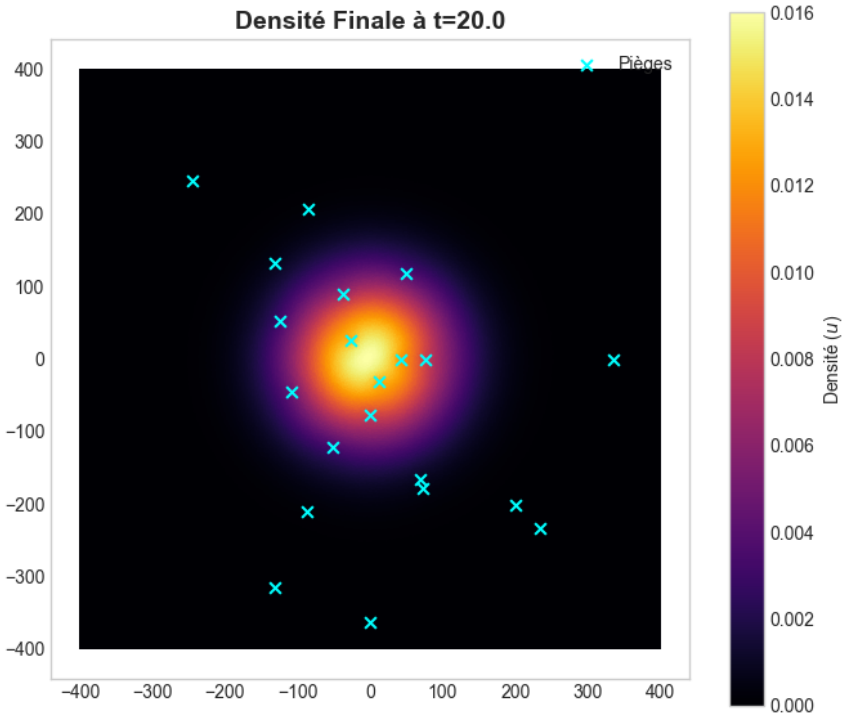
To determine the mesh, points are still generated randomly; however, the probability of generating points near the traps is increased.

To this end, we introduce a point density function $f : \Omega \to [0, 1]$ that is closer to 1 in regions where a higher density of points is desired.

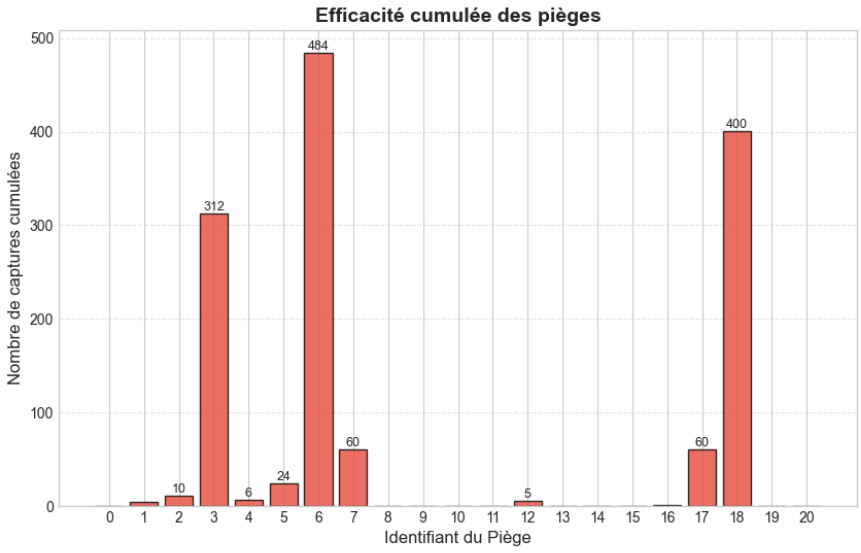
To generate the points, the following procedure is applied: a point (x, y) is drawn uniformly at random from Ω along with a random value λ; the point is retained if and only if $f(x, y) > \lambda$. This procedure is repeated until the desired number of points is reached.

In our case, we used the function: $f(r) = \max_{i \in 1,\dots,N_{\text{trap}}} (g(\|r - r_i\|))$ where r_i is the position of the i-th trap and $g(x) = \begin{cases} 1 & \text{if } x \leq R_{\text{trap}} \\ 1 - 0.85 * \frac{x-R}{R} & \text{if } R_{\text{trap}} \leq x < 2R_{\text{trap}} \\ 0.15 & \text{otherwise} \end{cases}$ where R_{trap} is the characteristic size of the traps. This yields the profile shown in Figure 7.

To construct the mesh, Delaunay triangulation is then applied. Figure 8 shows how the mesh triangles are distributed.



(a) Mosquito density



(b) Trap efficiency

Figure 6: FVM simulation results at t=20

6.6.2 Results with an Adapted Mesh

The results obtained remain very close to those in Figure 6a, and are in good agreement with the probabilistic model results (see Figure 9). The computation time is identical since the same number of points is used; the only difference is an increase in accuracy.

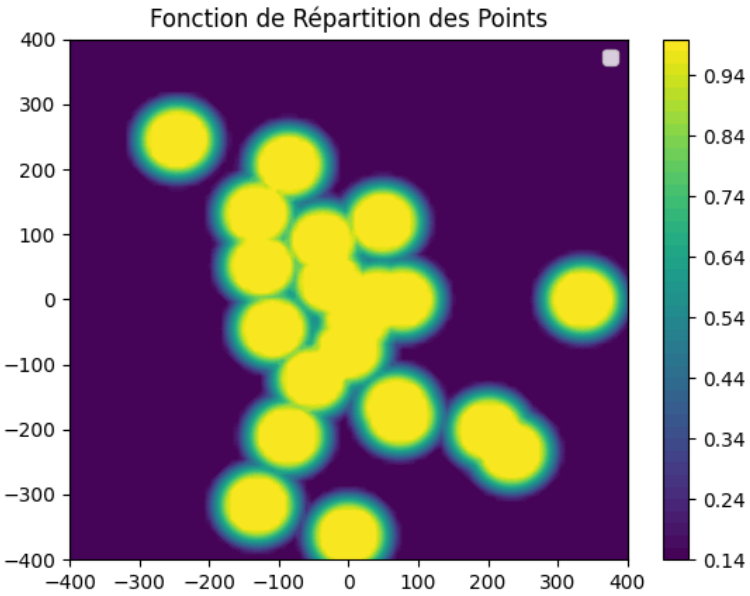


Figure 7: Chosen point density function

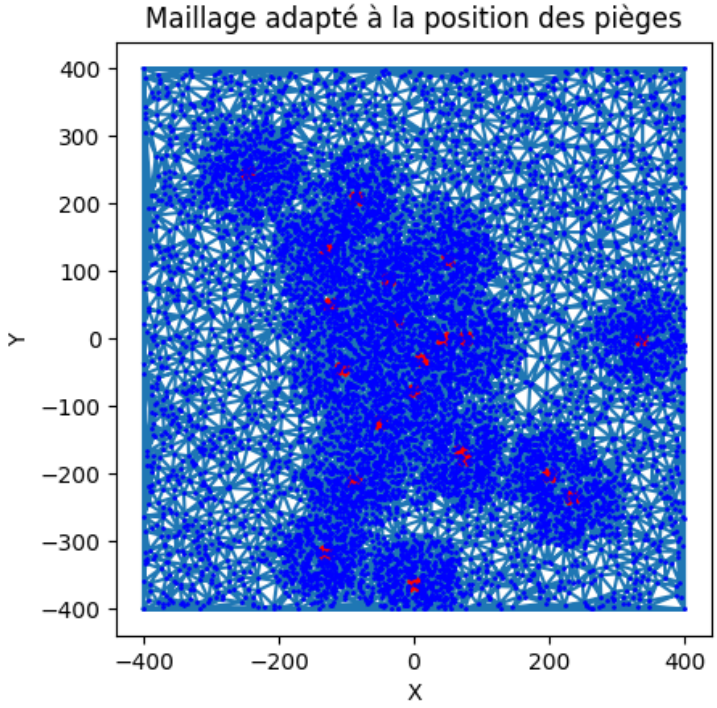


Figure 8: FVM mesh

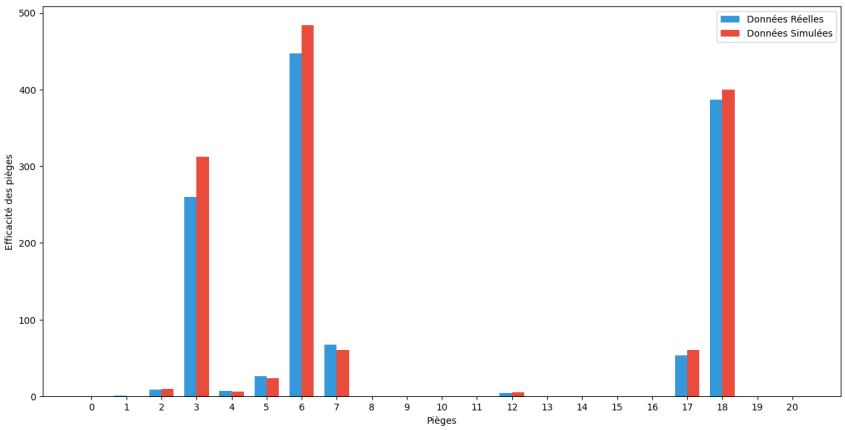


Figure 9: Comparison of FVM results with the probabilistic model

7 Inference of the physical parameters of our model from real capture data

7.1 Motivation

Now that our deterministic simulation is properly calibrated, our objective is to adapt this simulation to real-world situations by inferring, from trap capture data, the macroscopic parameters of the reaction-diffusion equation. This will enable us to provide an accurate modeling of the displacement of the entire mosquito population.

The finite difference method converges sufficiently fast to be used for optimizing these parameters.

7.2 Identifiability of the physical parameters

7.2.1 Justification of the identifiability issue

Based on the results regarding parameter identifiability presented in Capel and Muñoz, 2024, the parameters γ and R_{trap}, corresponding respectively to the maximum efficiency of the traps and their characteristic capture radius, suffer from a critical lack of identifiability.

To highlight the identifiability conflict between these parameters, the topology of the cost function with respect to γ and R_{trap} is shown in Figure 10.

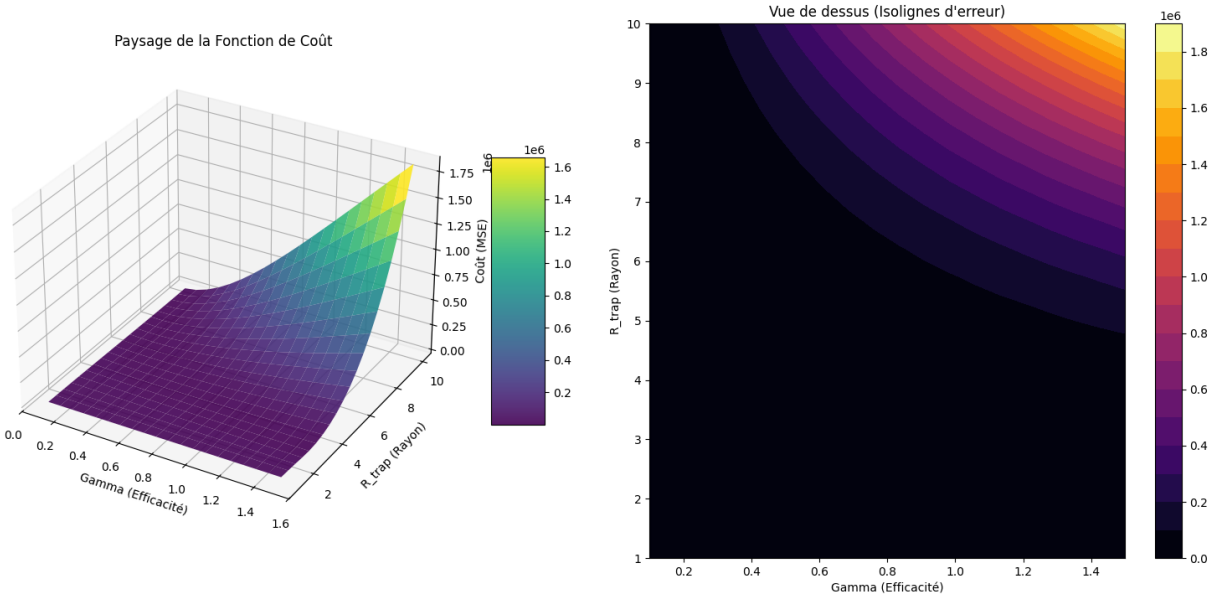


Figure 10: Cost function landscape as a function of γ and R_{trap}

Indeed, the landscape does not exhibit a well-defined basin but rather a compromise region where an increase in efficiency γ can be compensated by a decrease in the radius R_{trap}.

7.2.2 Applied solution: functional constraint

To stabilize the L-BFGS-B optimization and ensure the uniqueness of the solution, we introduced a functional relationship between the two parameters. By imposing a target capture risk at a given distance (e.g., $R_{trap} = \sqrt{\frac{25}{\ln(\gamma/0.1)}}$), we reduce the search space to a curve within this valley. This method ensures convergence toward biologically interpretable parameters while remaining consistent with the physics of the reaction-diffusion model.

7.2.3 Identifiability of the other parameters

To verify the robustness of our optimization method, we perform a GridSearch on the diffusivity parameter D and the natural mortality rate ν. In this analysis, the capture parameters γ and R_{trap} are fixed.

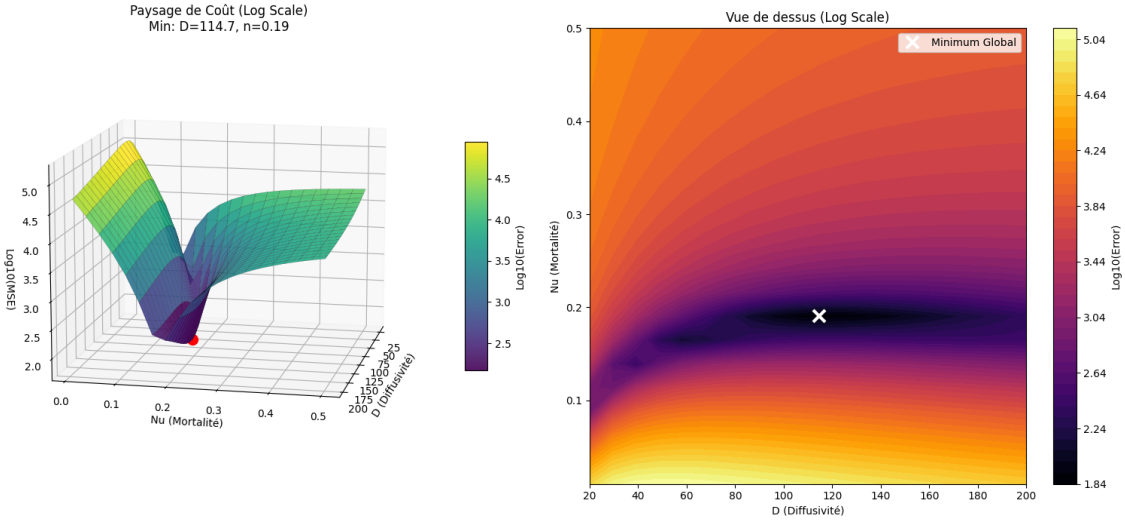


Figure 11: Cost function landscape as a function of D and ν

The results obtained, illustrated by the cost function (MSE) landscape in Figure 11, lead to the following conclusions:

- **Presence of a unique global minimum:** Unlike the capture parameters, the pair (D, ν) exhibits a clearly defined potential well. This indicates that these two parameters are structurally identifiable.
- **Decoupling of loss mechanisms:** Although diffusion (spatial dispersion) and mortality (temporal decay) both contribute to reducing the number of mosquitoes locally detected by a trap, they have distinct mathematical signatures.

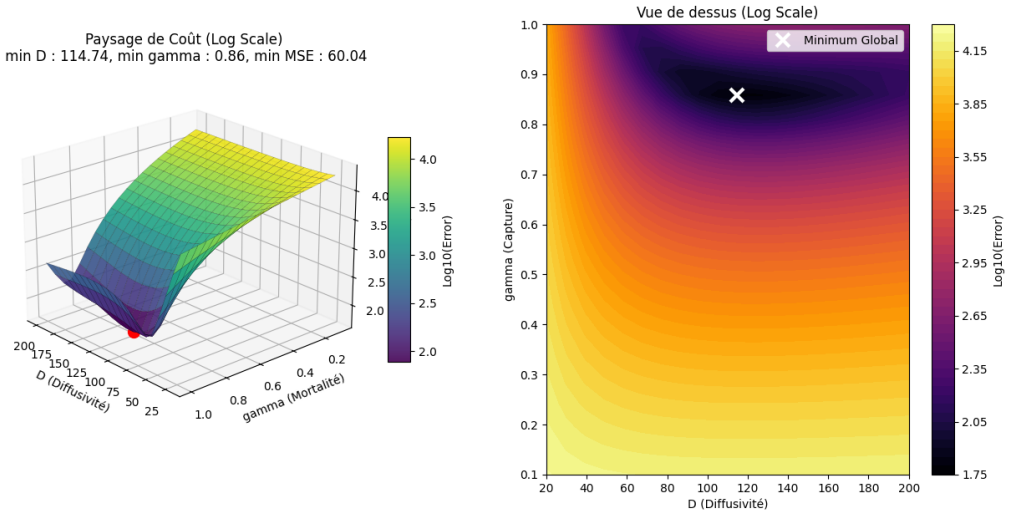


Figure 12: Cost function landscape as a function of D and γ

Moreover, the cross-sensitivity analysis between diffusivity (D) and trap efficiency (γ), presented in Appendix 12, reveals the presence of a unique global minimum, confirming that the technical trap parameters are decoupled from the dispersion dynamics.

Thus, the existence of these minima guarantees the existence of a solution to our optimization problem.

7.3 Optimization strategy and minimization algorithm

The optimization of the physical parameters of our model (D, ν, γ) constitutes a nonlinear optimization problem.

7.3.1 Formulation of the cost function

Our metric is the MSE (*Mean Square Error*) function, which computes the squared error between the number of mosquitoes captured in the simulation and the real data for each trap.

The objective is to minimize this cost function $J(\theta)$ by adjusting the model parameters using the `minimize` function from the `scipy.optimize` module.

7.3.2 Justification of the L-BFGS-B algorithm

The choice of the **L-BFGS-B** minimization algorithm (*Limited-memory Broyden Fletcher Goldfarb Shanno with Bounds*) is motivated by the following two points:

1. **Computational cost of the objective function:** Our cost function $J(\boldsymbol{\theta})$ relies on a complete numerical simulation (FDM) at each evaluation. Computing the exact Hessian matrix (second derivatives) is computationally too expensive to be performed explicitly. L-BFGS-B belongs to the class of *quasi-Newton* methods, which approximate the Hessian without computing it explicitly, providing faster convergence than simple gradient descent.
2. **Strict physical constraints (Bounds):** The model parameters represent physical quantities that cannot be negative (e.g., a diffusion coefficient $\sigma < 0$ or a radius $R_{trap} < 0$ has no physical meaning). The "-B" variant of the algorithm allows the definition of strict bounds ($\boldsymbol{l} \leq \boldsymbol{\theta} \leq \boldsymbol{u}$), ensuring the physical validity of the solution.

7.3.3 Principle of the L-BFGS-B algorithm

The algorithm seeks to minimize a regularized objective function defined as:

$$J(\boldsymbol{\theta}) = \text{MSE}(\boldsymbol{\theta}) + \lambda \cdot \Omega(\boldsymbol{\theta}) \tag{35}$$

where MSE is the mean squared error between simulated and real captures, and Ω is the regularization term (L2 penalty) that keeps the parameters close to their physical priors.
The L-BFGS algorithm approximates the inverse Hessian $H_k \approx [\nabla^2 J]^{-1}$ using curvature information accumulated during previous iterations. The update follows the formula:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_k H_k \nabla J(\boldsymbol{\theta}_k) \tag{36}$$

where α_k is the step size determined by a line search.

7.4 Optimization results and analysis

When testing our optimization procedure to recover the parameters of the target model, the optimization converges (after 23 iterations) to the following parameters:

- **Diffusion:** $D_{approx} = 134$ — target value $D = 150$ (10% error)
- **Natural mortality rate:** $\nu_{approx} = 0.18$ — target value $\nu = 0.2$ (8% error)
- **Trap parameters:** $\gamma_{approx} = 0.58$, $R_{trap_{approx}} = 3.77$ — target values: $\gamma = 0.8$ and $R_{trap} = 3.46\,\text{m}$

  However, this discrepancy is not due to an approximation error of the method, but rather to the persistent identifiability issue associated with these parameters.

The comparison of trap captures between our simulation with optimized parameters and the probabilistic simulation data is shown in Figure 13:

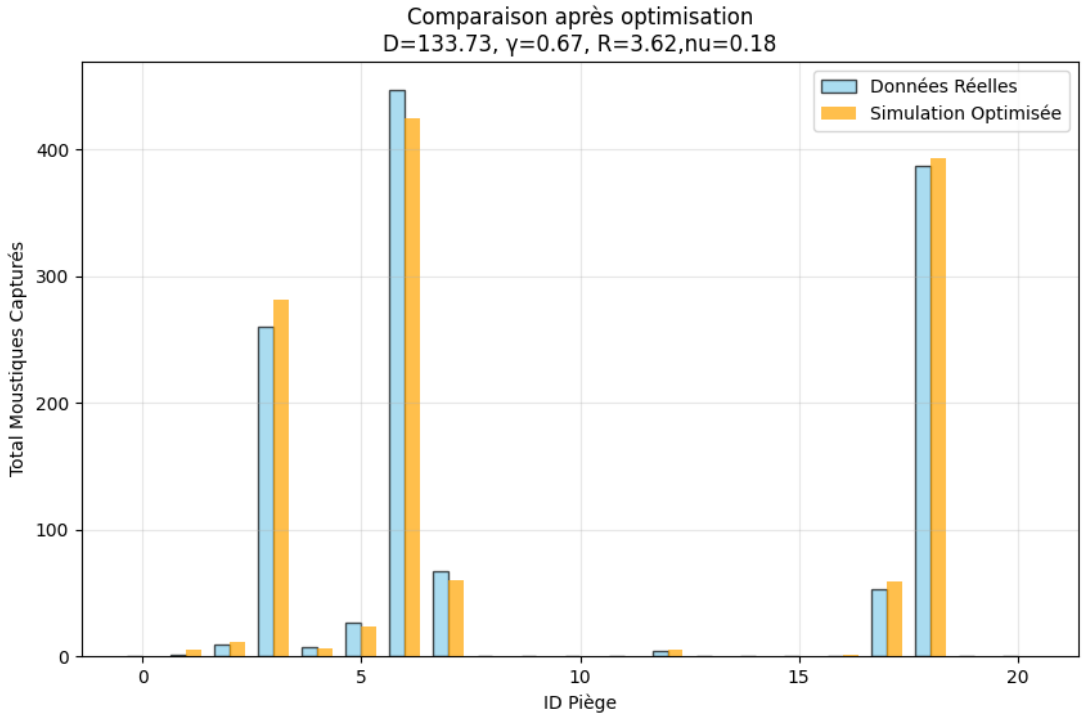


Figure 13: Optimization result allowing recovery of the target model parameters

Thus, this optimization method provides accurate and functionally reliable approximation results, with a relatively low convergence time (23 iterations).

7.5 Conclusion on parameter inference using the optimization method

The method is scientifically validated for the inference of survival and dispersion parameters, but requires fixing one capture parameter to ensure the uniqueness of trap-related solutions.

8 Conclusion

This work has established a robust mathematical and numerical framework for modeling the dynamics of mosquito populations subjected to trapping devices, with the aim of optimizing SIT-type vector control strategies.

From a theoretical perspective, our analytical study has highlighted the absence of exact closed-form solutions for physically realistic capture functions. While the perturbative approach based on the Duhamel series provides a rigorous framework for understanding short-time capture mechanisms, its combinatorial complexity severely limits its practical use, thereby justifying the reliance on numerical methods.

From a numerical standpoint, the comparative implementation of the Finite Difference Method (FDM), the Finite Element Method (FEM), and the Finite Volume Method (FVM) confirmed the consistency of the continuous model based on the diffusion equation with atomistic stochastic simulations. Although FEM and FVM offer better handling of complex geometries and local gradients around traps, the Finite Difference Method proved to be particularly efficient in terms of cost/implementation trade-off for iterative optimization tasks.

For future work, it would be interesting to calibrate the parameters of the diffusion model so that it reproduces real observed behavior. To this end, we have already implemented a method that allows us to fit the parameters of the equation to real data. However, we did not have sufficient time nor sufficient data to determine the optimal parameters corresponding to the observations, although the method is functional and can easily be applied once the data are available.

Appendices

A Code

The GitLab repository used for the project is available at this address.

B Data

The capture data used for all simulations are presented in Table 1, while the coordinates of the traps are reported in Table 2.

Table 1: Cumulative captures over the days for each trap

Day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Trap 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 2	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
Trap 3	0	0	0	0	0	0	0	1	1	4	5	6	6	7	7	8	8	9	9	9	9
Trap 4	0	8	52	103	147	173	202	218	231	237	241	246	251	254	257	259	260	260	260	260	260
Trap 5	0	0	0	0	0	0	0	0	0	1	1	1	2	2	3	4	5	5	5	7	7
Trap 6	0	0	0	0	1	3	4	6	9	12	15	18	19	21	24	25	25	25	25	25	26
Trap 7	0	37	143	244	312	353	389	403	415	422	433	437	440	443	444	447	447	447	447	447	447
Trap 8	0	0	0	3	11	18	22	28	42	46	50	55	57	61	62	64	66	66	66	67	67
Trap 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 13	0	0	0	0	0	0	1	1	1	1	1	2	2	4	4	4	4	4	4	4	4
Trap 14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 18	0	0	0	4	12	21	27	33	37	42	43	44	47	47	50	50	50	50	51	53	53
Trap 19	0	37	117	190	246	289	310	332	351	361	371	372	373	379	381	383	385	385	387	387	387
Trap 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trap 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2: Coordinates (x, y) of the traps

id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
x	-131.7	-124.7	-108.3	42.3	49.2	-37.3	12.8	0.0	73.6	0.0	-130.8	-87.0	-50.6	335.9	233.9	200.8	68.9	76.6	-26.4	-245.4	-85.7
y	131.7	51.6	-44.9	0.0	118.7	90.2	-30.8	-76.9	-177.6	-362.8	-315.8	-210.0	-122.2	0.0	-233.9	-200.8	-166.3	0.0	26.4	245.4	207.0

A small Python script also allows these data to be visualized; a frame extracted from the GIF animation created in the GitLab project is available in Figure 14.

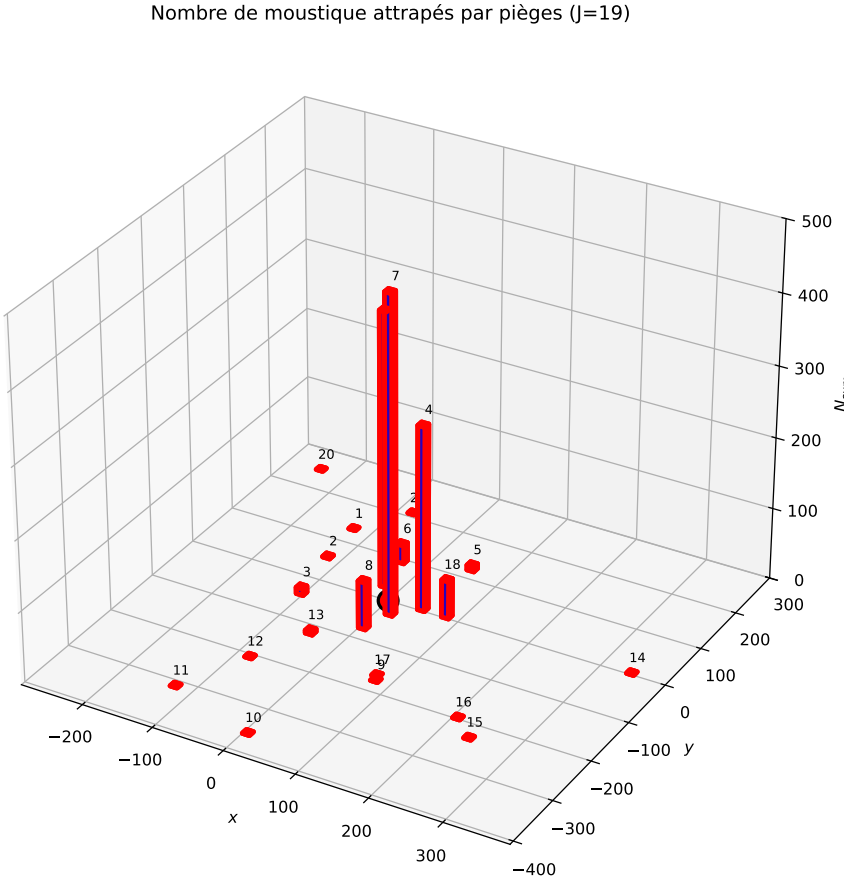


Figure 14: 3D plot of the capture data used

References

Capel, A., & Muñoz, F. (2024, September). *Survival and dispersion model using capture time for sit* (Research Report). CIRAD. https://umr-astre.pages-forge.inrae.fr/sit-methods/

Chaieb, A., Petiot, B., Mudarra, L., Bourret, L., & Muñoz, F. (2025). Points de divergence négative dans l'équation de diffusion pour la modélisation de capture des moustiques [Projet d'études - Programme ingénieur CentraleSupélec]. https://hal.science/hal-05084285v1

Ciarlet, P. G. (1978). *The finite element method for elliptic problems*. North-Holland.

CIRAD. (n.d.). Cirad - centre de coopération internationale en recherche agronomique pour le développement.

Crank, J. (1975). *The mathematics of diffusion* (2nd ed.). Oxford University Press.

Kato, T. (1995). *Perturbation theory for linear operators* (2nd ed.). Springer Berlin, Heidelberg. https://doi.org/https://doi.org/10.1007/978-3-642-66282-9

Nguyen, N., Bonnefon, O., Gato, R., Almeida, L., & Roques, L. (2025). Mechanistic-statistical inference of mosquito dynamics from mark-release-recapture data [Preprint submitted to HAL on 9 Oct 2025]. *hal-05306369*. https://hal.science/hal-05306369v1

Pironneau, O. (1988). *Finite element methods for fluids* [Notes de cours données à l'Université Pierre et Marie Curie & INRIA; à paraître en 1988, Wiley]. Wiley. Retrieved May 9, 2025, from https://www.ljll.fr/pironneau/publi/publications/OPfemInFluids.pdf

Turbiner, A., & Ushveridze, A. (1987). Spectral singularities and quasi-exactly solvable quantal problem. *Physics Letters A*, *126*(3), 181–183. https://doi.org/https://doi.org/10.1016/0375-9601(87)90456-7