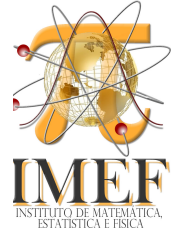




Universidade Federal do Rio Grande

Instituto de Matemática, Estatística e Física

PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA



PEDRO HENRIQUE FERNANDES LOBO

**REPRESENTAÇÃO DA NUVEM ELETRÔNICA DO
ÁTOMO DE HIDROGÊNIO E ESTUDO DA FORMAÇÃO DE
CLUSTERS EM UM MODELO DE PERSEGUIÇÃO E FUGA
UTILIZANDO O MÉTODO MONTE CARLO**

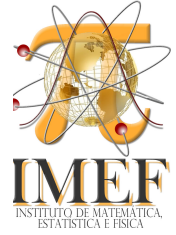
Rio Grande-RS, 2019



Universidade Federal do Rio Grande

Instituto de Matemática, Estatística e Física

PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA



PEDRO HENRIQUE FERNANDES LOBO

**REPRESENTAÇÃO DA NUVEM ELETRÔNICA DO
ÁTOMO DE HIDROGÊNIO E ESTUDO DA FORMAÇÃO DE
CLUSTERS EM UM MODELO DE PERSEGUIÇÃO E FUGA
UTILIZANDO O MÉTODO MONTE CARLO**

Dissertação apresentada à Universidade Federal do Rio Grande, como parte das exigências do Programa de Pós-Graduação em Física, para obtenção do título de Mestre.
Orientador: Professor Dr. Everaldo Arashiro

Rio Grande-RS, 2019

Ficha catalográfica

L799r Lobo, Pedro Henrique Fernandes.
Representação da nuvem eletrônica do átomo de hidrogênio e estudo da formação de *clusters* em um modelo de perseguição e fuga utilizando o método Monte Carlo / Pedro Henrique Fernandes Lobo. – 2019.
57 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Física, Rio Grande/RS, 2019.
Orientador: Dr. Everaldo Arashiro.

1. Monte Carlo 2. Átomo de Hidrogênio 3. Modelo Baseado em Agentes 4. Dinâmica Estocástica 5. *Clusters* de Agentes I. Arashiro, Everaldo II. Título.

CDU 544.112:546.11

Representação da Nuvem Eletrônica do Átomo de Hidrogênio e Estudo da Formação de Clusters em um Modelo de Perseguição e Fuga Utilizando o Método Monte Carlo

Pedro Henrique Fernandes Lobo

Orientador:

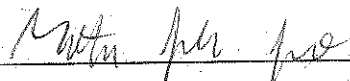
Prof. Dr. Everaldo Arashiro

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Física no Curso de Mestrado em Física, como parte dos requisitos necessários à obtenção do título de Mestre em Física.

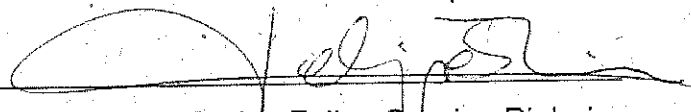
Aprovada por:



Prof. Dr. Everaldo Arashiro



Prof. Dr. Matheus Jatkoske Lazo



Prof. Dr. Carlos Felipe Saraiva Pinheiro

Rio Grande
Outubro de 2019

Resumo

Esta dissertação trata-se da aplicação do Método de Monte Carlos em dois trabalhos com objetivos distintos. O primeiro visa mostrar como representar a nuvem eletrônica do átomo de hidrogênio. Utilizamos o método de Neumann para representar distribuições de probabilidade em problemas introdutórios de partículas em caixas, aumentando gradativamente a dificuldade para desenhar os orbitais atômicos. Já no segundo trabalho, foi desenvolvido um modelo estocástico baseado em agentes, que estuda a formação de *clusters* de seus indivíduos, chamados de predadores e presas, em uma rede construída através da geometria do táxi. Para entendermos as condições nas quais os *clusters* se formam, foram variadas as concentrações dos dois tipos de agentes e um parâmetro σ , que atua como um mediador, amplificando ou atenuando as “forças” de atração e repulsão entre eles. As simulações revelaram que tanto o σ quanto a concentração dos agentes são decisivos para formação de *clusters*.

Palavras-chave: Monte Carlo, átomo de hidrogênio, modelo baseado em agentes, dinâmica estocástica, clusters de agentes.

Abstract

This dissertation deals with the application of the Monte Carlo Method in two works with different goals. The first aims to represent the electron cloud of the hydrogen atom. We use the Neumann method to represent distributions of probability in introductory problems of particles in the boxes, increasing the difficulty gradually to plot atomic orbitals. In the second work, we developed a stochastic agent-based model, which studies the formation of clusters of their individuals, called predators and prey, in a lattice created through the taxicab geometry. In order to understand the conditions under which clusters form, the concentrations of both agents types, and a parameter σ , which acts as a mediating, amplifying or reducing the “forces” of attraction and repulsion between them. The simulations revealed that both the σ and the concentration of the agents are decisive to formation of the clusters.

Keywords: Monte Carlo, hydrogen atom, agent-based model, stochastic dynamics, agents clusters.

Introdução

O Método Monte Carlo (MMC) é uma importante ferramenta numérica que está em destaque na maior parte desta dissertação e por isto faremos uma breve introdução sobre o tema e apresentaremos a sua aplicação em dois trabalhos na sequência.

No primeiro, aplicamos conceitos iniciais de mecânica quântica, em problemas típicos de partículas na caixa e finalizamos ao representarmos os orbitais atômicos do átomo de hidrogênio. Esta mesma metodologia já foi explorada anteriormente (ver [1]), mas focamos em cobrir todos passos mais detalhadamente.

O segundo trabalho trata-se de um modelo baseado em agentes, inspirado na dinâmica de perseguição envolvendo predadores e presa, na qual em certas condições emergem os comportamentos coletivos destes dois tipos, podendo resultar na formação de manadas. Neste caso, procuramos estabelecer parâmetros genéricos que explicam este fenômeno sem adentrarmos em níveis detalhamentos como espécies, bem como suas estruturas hierárquicas ou qualquer outra particularidade. Nós consideramos que todos indivíduos (ou agentes) são autônomos e estão sujeitos as mesmas regras, com exceção de umas delas que se aplicam a cada um destes tipos, ou seja, os predadores e presas. Além disto, também demos continuidade a este trabalho, ao fazermos uma análise mais detalhada que será apresentado adiante.

Método Monte Carlo

A simulação consiste em tratar um problema real através da sua reprodução em ambiente controlado. O ambiente é frequentemente assistido por equipamentos eletrônicos, capaz auxiliar na reprodução de um sistema em estudo, inclusive em menor escala. Um exemplo é aplicação do Princípio de Pascal na construção de um elevador hidráulico, no qual um pistão é acionado por algum motor elétrico [2]. Isto constitui um estudo de sistema que tem todos os seus componentes são conhecidos ou, pelo menos dedutíveis, e cujo comportamento tem um caráter determinístico intrínseco, pelo menos no nível macroscópico.

Em outros casos, alguns ou mesmo todos os componentes podem estar sujeitos a flutuações aleatórias. Eles não podem ser descritos rigorosamente através de equações, mas podem ser exploradas através de cálculos probabilísticos. Sistemas como estes são temas de natureza estatística e qualquer que seja o caminho escolhido para simulá-los haverá sempre um componente estocástico, ou seja, será baseado em distribuições de probabilidade, sendo tema para ser incorporado à simulação estocástica.

Uma grande quantidade de problemas estão inseridas nesta categoria. Na percolação por exemplo, uma rede vazia é preenchida gradualmente com partículas inserindo-as uma a uma aleatoriamente a cada passo de tempo [3, 4]. Neste contexto, podem surgir discussões a respeito da formação de *clusters* a partir de grupos de células ocupadas. Inclusive há muitas vezes interesse em determinar um limite de percolação, que marca uma transição de fase, isto é o valor de concentração mínima de células ocupadas na qual há conectividade entre bordas opostas na rede [3, 4]. Este modelo tem aplicações em problemas físicos tais como a permeabilidade rochas porosas e condutividade de mistura de materiais dielétricos e condutores [4].

Do ponto de vista da estatística, estes sistemas apresentam funções possivelmente complicadas de variáveis aleatórias. Nosso objetivo é basicamente reproduzir estas variáveis em ambientes controlados, independentemente da complexidade de estruturas vinculadas a estas variáveis. Embora anteriormente este tipo simulação tenha sido realizada manualmente (ver [5]), atualmente utilizamos computador como ferramenta preferencial para tratar estes casos.

Para produzir uma quantidade viável amostras aleatórias e então propor modelos estocásticos, utilizamos o MMC. Ou seja, é uma técnica voltada para modelar um sistema governado por fatores incertos e que podem ser utilizados para extrair resultados qualitativos, embora a maior parte dos casos estamos interessados em uma resposta numérica.

Trabalhos com a temática MMC se baseiam em dois princípios da teoria da probabilidade: A lei dos grandes números e o teorema do limite central [6]. Em resumo, quando utilizado em situações com interesse de se obter respostas quantitativas, o MMC expressa um valor esperado que interpretamos como uma média amostral. A lei dos

grandes números demonstra que quando o número de repetições se aproxima do infinito, a sua média amostral se aproxima de um valor assintótico que interpretamos este como valor esperado. Porém não podemos afirmar, que esta média amostral converge para a populacional (talvez esta nem exista). Por outro lado, através do teorema do limite central, podemos correlacionar o desvio da média amostral com a da média populacional, e com isto obter um intervalo de confiança que possivelmente contenha a média de interesse e que desconhecemos.

Para exemplificarmos este processo, iremos utilizar o MMC como uma quadratura para verificarmos numericamente a seguinte expressão:

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx = \pi \quad (1)$$

Para tal, considere uma função $z(x)$ onde x é uma variável aleatória contínua e $f(x)$ é uma função densidade de probabilidade (FDP). O valor esperado de $z(x)$ é definido por (ver [6]):

$$\langle z \rangle \equiv \int_a^b z(x)f(x)dx \quad (2)$$

Sendo que na teoria da probabilidade, há uma lista de distribuições contínuas usuais frequentemente aplicadas em problemas reais [6]. Porém, por uma questão de simplificação e conveniência, adotaremos a FDP uniforme dada por:

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{se } a \leq x \leq b \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

Por outro lado, a média amostral é dada por:

$$\bar{z} = \frac{1}{N} \sum_{i=1}^N z(x_i) \quad (4)$$

Onde N é o número total de amostras x_i . De acordo com a lei dos grandes números, para $N \rightarrow \infty$ o valor esperado se iguala a média aritmética de amostras (ver [6]). Ou seja:

$$\langle z \rangle = \bar{z} \quad (5)$$

Combinando as expressões (2), (3), (4) e (5) e assumindo que todo x_i é uma variável aleatória discreta uniforme, contida no intervalo $[a, b]$, sendo $z(x)$ uma função “bem comportada¹” neste mesmo intervalo e que $N \rightarrow \infty$, obtemos a seguinte relação:

$$\int_a^b z(x) dx = \frac{b-a}{N} \sum_{i=1}^N z(x_i) \quad (6)$$

Ou seja, de acordo com este resultado, podemos utilizar o somatório de amostras discretas para calcular uma integral definida. Entretanto, por se tratar de um processo numérico, devemos considerar que N é “grande o suficiente”, uma vez que os computadores tradicionais operam apenas com números discretos e finitos.

Sendo assim, para avaliarmos numericamente a integral proposta, prosseguiremos comparando o lado esquerdo da equação (1) com expressão (6). Conseqüentemente, obtemos o seguinte resultado:

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx \approx \frac{(1) - (-1)}{N} \sum_{i=1}^N \frac{1}{\sqrt{1-x_i^2}} = R_N \quad (7)$$

Note que a função $z(x)$ não está definida para $x = -1$ e $x = 1$, assim devemos sortear valores para $x_i \in (-1, 1)$. Além disto, como de antemão sabemos o valor exato desta integral, podemos verificar o erro percentual η comparando o resultado numérico R_N com o lado direito da equação 1. Um exemplo de implementação deste algoritmo está presente em Anexo 1 e os resultados para vários tamanhos de amostras N estão dispostos na tabela 1:

Como previsto, a tabela 1 indica que quanto maior o tamanho N da amostra, mais o resultado R_N se aproxima de um valor assintótico π , e menor é o erro percentual η obtido. Perceba também que foi possível verificar o erro porque que dispunhamos da solução da integral. Mas em geral o MMC é indicado apenas quando não conhecemos a solução

¹ funções contínuas e diferenciáveis em qualquer ordem, pelo menos dentro de um intervalo de interesse.

N	$R_N = (b - a)\bar{z}$	$\eta = 1 - R_N/S_E \times 100\%$
10^1	4.63648	47.58
10^2	2.74977	12.47
10^3	3.22624	2.69
10^4	3.10771	1.08
10^5	3.15431	0.40
10^6	3.14549	0.12
10^7	3.14029	0.04
10^8	3.14197	0.01
10^9	3.14174	< 0.01

Tabela 1: Resultado numérico R_N da integral presente na equação 1, resolvido através do MMC para alguns tamanhos de amostra N . O termo η é erro percentual, \bar{z} é a média amostral, S_E é a solução exata ($S_E = \pi$) e b e a são os limites de integração ($b = 1$ e $a = -1$).

analítica, e por isso, nestes casos devemos proceder de outra maneira para verificarmos se o resultado é satisfatório. Para tal, utilizaremos o teorema do limite central.

Este teorema afirma que dado uma amostra que obedeça qualquer distribuição, a média amostral irá se aproximar de uma distribuição normal a medida que o tamanho da amostra aumenta (ver [6]). Matematicamente esta relação é expressa através da equação seguinte, onde \bar{z} obtido de uma distribuição com média $\langle z \rangle$ e desvio padrão amostral $\sigma(z)$ (ver [6, 7]).

$$\lim_{N \rightarrow \infty} \text{Prob} \left\{ \alpha \leq \frac{\bar{z} - \langle z \rangle}{\sigma(z)/\sqrt{N}} \leq \beta \right\} = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{u^2/2} du \quad (8)$$

O lado direito da equação anterior representa uma gaussiana normalizada com a média $\langle z \rangle$ centrada na origem. Quando esta integral é resolvida, procuramos estabelecer intervalos de integração simétricos, por isto iremos assumir que $\beta = -\alpha = \lambda$. Se considerarmos que número N de amostras é “grande o suficiente”, podemos aproximar¹ esta equação e substituir o desvio padrão populacional $\sigma(z)$ pelo desvio padrão amostral $s(z)$. Ao realizarmos todas estas considerações obtemos a seguinte expressão:

$$\text{Prob} \left\{ \bar{z} - \lambda \frac{s(z)}{\sqrt{N}} \leq \langle z \rangle \leq \bar{z} + \lambda \frac{s(z)}{\sqrt{N}} \right\} \approx \frac{1}{\sqrt{2\pi}} \int_{-\lambda}^{\lambda} e^{u^2/2} du \quad (9)$$

Onde $s(z)$ é definido por:

¹ O desvio padrão amostral se aproxima do desvio padrão populacional em grandes amostras [7].

$$s(z) \equiv \sqrt{\frac{1}{N-1} \sum_{i=1}^N [z(x_i) - \bar{z}]^2} \tag{10}$$

A parte à direita da aproximação (9) depende de λ e representa o de nível de confiança. Para solucionar esta integral, geralmente obtemos os resultados através de dados truncados e tabelados em livros de estatística [8] ou por meio de algum método de integração numérica.

Logo, para um dado nível de confiança, o MMC estima $\langle z \rangle$ usualmente representado como $\bar{z} \pm \lambda s(z)/\sqrt{N}$. Por exemplo, para $\lambda = 1.0$, com um nível de confiança de 0.6827 a verdadeira média $\langle z \rangle$ está contida no intervalo de confiança $[\bar{z} - \lambda s(z)/\sqrt{N}, \bar{z} + \lambda s(z)/\sqrt{N}]$.

Sendo assim, prosseguiremos aplicando o teorema do limite central no problema proposto. Neste caso, como temos o computador à disposição, o lado direito da expressão (9) será obtido numericamente assim como o intervalo de confiança. Este algoritmo já implementado está disponível no Anexo 2 e os resultados expressos na tabela 2 para alguns valores de N e λ .

N	λ	nív. Conf.	$R_N = (b - a)\bar{z}$	$\pm(b - a)\lambda s(z)/\sqrt{N}$	$\langle z \rangle \in \bar{z} \pm \lambda s(z)/\sqrt{N} ?$
10^3	0.5	0.3829	3.35	± 0.08	X
10^3	1.0	0.6827	3.02	± 0.09	X
10^3	1.5	0.8664	3.1	± 0.1	✓
10^3	2.0	0.9545	3.2	± 0.2	✓
10^3	2.5	0.9876	3.1	± 0.2	✓
10^9	0.5	0.3829	3.1418	± 0.0001	X
10^9	1.0	0.6827	3.1414	± 0.0002	✓
10^9	1.5	0.8664	3.1415	± 0.0003	✓
10^9	2.0	0.9545	3.1417	± 0.0004	✓
10^9	2.5	0.9876	3.1419	± 0.0005	✓

Tabela 2: Resultado numérico R_N da integral presente na equação 1, resolvido através do MMC para alguns tamanhos de amostra N e diferentes λ . O termo \bar{z} é a média amostral, $s(z)$ é desvio padrão amostras e b e a são os limites de integração ($b = 1$ e $a = -1$). Na última coluna e verificado a solução exata da integral está contida no intervalo de confiança.

Como já antecipado, geralmente não há como determinar qual o erro da solução numérica porque não é possível compará-la com o valor real na maioria dos casos, por isto apenas estimamos. Porém, como convenientemente escolhemos uma integral definida

que possui solução analítica, verificamos na última coluna da tabela 2 se o intervalo de confiança contém a o verdadeiro valor médio. De acordo com os resultados apenas três destes testes apresentaram uma estimativa ruim, até porque o nível de confiança percentual 38, 29% e 68, 27% são relativamente baixos. Além disto, constatamos que se aumentando o λ o intervalo de confiança tende a diminuir e quanto maior o N , mais precisa é a estimativa.

Em resumo, na condição em o tamanho de amostra $N \rightarrow \infty$ a lei dos grandes números garante a média amostral converge para um valor assintótico e que, de acordo com teorema do limite central, o intervalo de confiança tende a zero, podendo este conter ou não a verdadeira média de interesse.

A partir de agora iremos fazer a aplicação destes conceitos em dois trabalhos, iniciando com o manuscrito intitulado “A smooth path to plot hydrogen atom via Monte Carlo method” e que se trata de uma aplicações qualitativa do MMC.

3 A smooth path to plot hydrogen atom via Monte Carlo method

A smooth path to plot hydrogen atom via Monte Carlo method

Pedro Henrique Fernandes Lobo^{*1}, Everaldo Arashiro¹, Alcides Castro e Silva², Carlos Felipe Saraiva Pinheiro²

¹Universidade Federal do Rio Grande, Instituto de Matemática, Estatística e Física, Av. Itália km 8 Carreiros, Rio Grande, RS, Brasil

²Universidade Federal de Ouro Preto, Instituto de Ciências, Exatas e Biológicas, Departamento de Física, Ouro Preto, MG, Brasil.

Received on March 30, 2019; Revised on May 16, 2019; Accepted on May 25, 2019.

In this paper, we show how to build a basic computer program using the Monte Carlo method to display the hydrogen atomic orbitals. For this, in a heuristic way, we applied a von Neumann acceptance-rejection method in simple problems of potential wells, and we end with the hydrogen orbitals representation. In this technique, we spread points uniformly on the 1D and 2D charts of probability density distributions, then we filtered points under these “curves or surfaces”, and we extended this logic to 3D cases. Throughout the work, we also made some comments to help beginner students better understand the term “wave function” present in the Schrödinger equation. Also, we made all source code available at a third-party platform, for any purpose under the MIT license.

Keywords: hydrogen atom, quantum mechanics, Monte Carlo method, Neumann acceptance-rejection method.

1. Introduction

The introduction of quantum mechanics, even in physics or chemistry undergraduate courses is usually done by a historical review of several atomic models. Bohr developed one of these models, and it was essential to explain concepts of energy quantization and emission of spectral lines predicted by Rydberg [1]. Although the Bohr model is obsolete, it can still be used for beginner students to get acquainted with the quantization theme, and this is convenient because this model requires relatively simple conceptual and mathematical treatment as compared to its successor.

On the other hand, this practice may induce students to develop a misperception of the most modern and precise structures of quantum mechanics [2]. Therefore, researchers in the teaching area have suggested using computational tools to explain the Schrödinger models, making the current concepts of quantum mechanics more understandable [3–7].

In this context, the present study shows how to use a computer for learning in this area. In more detail, this project aims to teach undergraduate students to build their programs using the Monte Carlo technique in order to picture the hydrogen orbitals in a heuristic¹ way.

This method is convenient because it induces the student to understand the real meaning of probability density introduced by Born, which derives from the wave function present in the time-independent version of the Schrödinger equation. So, we hope that students could make a suitable link between this function and the concept of particle-wave duality. In that way, we will give a brief presentation of the Monte Carlo method and quantum mechanics, applying these in some simple systems and ending up with the hydrogen atom.

2. The Monte Carlo method - MCM

In many science fields like physics [6, 7, 9–12], biology [9, 10, 13–17], engineering [9], and finance [18] many problems are solved today by MCMs. That is, by a broad class of methods that use random sampling, to obtain numerical results or provide the probability distributions.

This simple formulation suggests we need a more proper definition, and indeed, this could come in handy at this point. However, as these MCMs have quickly become popular and new techniques, ideas, and concepts continue to appear, then it becomes difficult to establish an overview.

For this reason, we will introduce the MCM by applying it to the following problem: take, as an example, the boundary of an irregular picture on a rectangular sheet of paper. One way of estimating the inner region of this figure is by throwing many darts randomly inside this sheet. The area of this figure is approximately the area

*Correspondence email address: pedrohflobo@gmail.com

¹Heuristic is “a method which, on the basis of experience or judgment, seems likely to yield a reasonable solution to a problem, but which cannot be guaranteed to produce the mathematically optimal solution.” – Edward A. Silver (see [8]).

of paper multiplied by the ratio of darts that hit the drawing and any other point.

Let us illustrate the process using the method to calculate the area of the circle inscribed in a square, and also sketch these regions with points. In this case, the equation of the circumference centered on the origin is given by:

$$r^2 = x^2 + y^2 \quad (1)$$

where r is the radius of the circle, x and y are cartesian coordinates. Then we will make y dependent on x , keeping the r constant, so we obtain the following expression with domain D_f and range R_f :

$$\begin{aligned} y(x) &= \sqrt{r^2 - x^2} \\ D_f &: \{x \in \mathbb{R} \mid -r \leq x \leq r\} \\ R_f &: \{y \in \mathbb{R} \mid 0 \leq y \leq r\} \end{aligned} \quad (2)$$

This function is represented graphically by an arc above the x -axis. The bottom part is $-y(x)$. In this case, the range of $-y(x)$ will be $R_f : \{y \in \mathbb{R} \mid -r \leq y \leq 0\}$. So, we can use a sequence of steps below to draw the circle and get its area.

1. Set a positive r .
2. Set an amount p -points to plot.
3. Set an integer h -counter and reset it to zero.
4. Set an integer i -counter and reset it to zero.
5. Generate a random $x \in [-r, r]$.
6. Generate a random $w \in [-r, r]$.
7. If $w \in [-y(x), y(x)]$, then x and w are stored in the first list and add +1 to h . Otherwise, they are stored in the second list.
8. Add +1 to i .
9. Repeat all steps from the 5th until i equals p .
10. Evaluate a circle area by $S = (2r)^2 h/p$.
11. Make a single chart with the two lists with different colors.

Note that to chart the whole process can be summed up in randomly picking a pair of coordinates $\{x, w\}$, and then representing the points in the external and internal limits in different colors. See the result in Figure 1.

Concerning the calculation of the circle area, the reader may be wondering: why not calculate it by integration? Indeed, the exact solution is usually the best choice. Nevertheless, there are cases where integrals are challenging to manipulate to get a known analytical solution², or even are multidimensional.

In these cases, we must use numerical tools such as Trapezoidal and Simpson rules or even use the MCM to solve integrals [19–21]. Although for low dimensions cases, the fixed numerical quadratures³ provide better

²It is not always easy to substitute non-elementary integrals for simple power series anti-derivative functions.

³Numerical quadrature is nothing more than a technique for making approximations of integrals by sums.

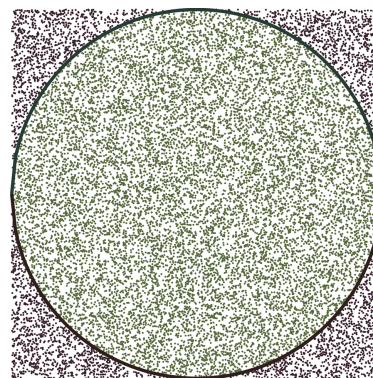


Figure 1: A circle (radius $r = 3$) inscribed in a square (side $l = 2r$), with two thousand points homogeneously distributed, covering the whole area. The top and bottom are given by the functions $y(x)$ and $-y(x)$ respectively.

convergence rates, this MCM takes advantage because its accuracy is almost independent of dimension [19, 20]. Also, this MCM avoids technical issues that appear in the other alternatives [19]. For these reasons and also for being easily implementable, the MCM is seen as a robust numerical tool.

However, be careful because there are at least two ways to calculate areas or volumes by MCMs. The technique we applied in our algorithmic is called von Neumann acceptance-rejection method (also known as hit-or-miss method), while that MCM which works as quadrature numeric is called sample-mean, also called crude Monte Carlo (see [21–23] for disambiguation). By the way, comparing these two options, only the von Neumann method can sketch areas and volumes, while getting these results avoiding the whole calculus formulation [21–23, 25].

Furthermore, for the area of simple figures like circles, it is easy to show that these two types of MCMs, the more we increase the sample size, the closer we get to the analytical solution [21–23]. Be that as it may, as we shall see further on, rather than get numerical results, we will use the MCM to fill multidimensional spaces with points.

3. Random numbers and sampling

For these MCMs to work is essential that generators provide an unpredictable sequence of numbers, and identically given by some probability distribution [10, 24, 25]. In other words, good random number generators must produce uncorrelated numbers. When random numbers are produced homogeneously in some range, we describe the generator as a uniform random number generator [24, 25].

Also, when it comes to a programming environment, pseudo-random numbers are widely used. That is: they seem to be random, but they appear in deterministic sequences initiated by seeds [24–26]. However, for different sets of values to be provided, i.e., seeming unpredictable, the actual time given by the CPU clock is used at least

once to choose a seed [26]. That way, every time the programs run, they seem to give unique results.

Since random numbers are present at the core of MCMs, then they have some basic requirements that generators must satisfy in order to be considered good [24,25]. Some tests suggest that those present in the software are unreliable, but fortunately, there are adequate packages ready to be used in more popular programming languages [27].

4. The wave equation

After the introduction of the Bohr model, de Broglie, motivated by the photoelectric effect, suggested that there could be a correlation between the electromagnetic equations with particles in movement (see the English version of [28] in [29]). Subsequently, Schrödinger, inspired by de Broglie hypothesis, or the concept we today know as particle-wave duality, formulated his homonymous equation below, which was later interpreted by Born, and which currently represents the primary tool of all non-relativistic quantum mechanics.

$$\hat{H}\psi = E\psi \quad (3)$$

In the previous equation, the Hamiltonian operator \hat{H} is associated with classical mechanics by an analogy to the combined operators of kinetic and potential energy, while the term ψ is the eigenfunction or wavefunction, and E is the eigenvalue or energy, we get both simultaneously when solving this equation [33,34].

For the terms \hat{H} and E , if beginners in quantum mechanics understand the concept of Newtonian mechanics and energy quantization introduced by Bohr, then they should have no difficulty in assimilating them. However, if the idea of electrons moving around the nucleus in precise orbits persists, there is no way to make a correct connection with ψ .

The explanation comes from the concept of particle-wave duality. The particles, by their nature, are located in points, on the other hand, the wavefunction (as the name suggests) spreads through space [33,34], and consequently, there is no reason to represent electrons in orbits. Born made an interpretation that tells us that $\psi(\vec{r}, t)$ is a probability amplitude, although it can often be complex⁴, $|\psi(\vec{r}, t)|^2$ is probability density and $|\psi(\vec{r}, t)|^2 d^3r$ is the probability $dP(\vec{r}, t)$ to find the particle in a volume d^3r , located between \vec{r} and $\vec{r} + d\vec{r}$, at a time t [33,34].

$$|\psi(\vec{r}, t)|^2 d^3r = dP(\vec{r}, t) \quad (4)$$

If we integrate the previous equation into the whole space S , then the particles will inevitably be somewhere. Therefore, the probability of finding them under these conditions is 100%.

⁴Complex in this context because it contains the imaginary number i .

$$\int_S |\psi(\vec{r}, t)|^2 d^3r = 1 \quad (5)$$

5. Monte Carlo method to provide density plots

A common strategy involving MCMs is a von Neumann acceptance-rejection technique [35]. In his method, pseudo-random numbers are generated uniformly to produce other non-uniform distributions. At some stage of this algorithm, the screening of these numbers is made using relational operators, which result in the establishment of acceptance and rejection zones, similarly⁵ to the example of the circle already introduced. Also, as we propose in this paper, this same von Neumann technique was used before to draw hydrogen orbitals [6].

From now on, we will apply this MCM, to produce images of the qualitative probability density, and the number of points should be in a subjective range, so that image is not empty (without contrast) or overloaded. We will start from typical potential well problems, and end up with the hydrogen atom.

5.1. The one-dimensional infinite potential well

The Hamiltonian previously introduced (see section 4) is explicitly given in the following form $\hat{H} = -\hbar^2 \nabla^2 / 2m + V(\vec{r}, t)$, where $V(\vec{r}, t)$ is the potential term, \hbar is the reduced Planck constant, m is the mass of the particle, and ∇^2 is the Laplacian operator. Also, the time-independent version for equation (3) in one dimension, applied to the problem of the infinite potential well of length L is:

$$-\frac{\hbar}{2m} \frac{d^2\psi(x)}{dx^2} + V(x)\psi(x) = E\psi(x) \quad (6)$$

where $V(x)$ is given by:

$$V(x) = \begin{cases} 0, & \text{if } 0 < x < L \\ +\infty, & \text{otherwise.} \end{cases} \quad (7)$$

In this case, the particle can only occupy the limited region in the box from 0 to L , and the solution of the previous differential equation, including the application of the boundary conditions in this same interval is (see [31,36,37]):

$$\psi_n(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi}{L}x\right) \quad (8)$$

where n is a positive integer ($n = 1, 2, 3, \dots$), also known as a quantum number, and it comes from the application of the boundary conditions of this problem.

⁵In his original work [35], von Neumann created a linear function to convert one interval of random numbers into another. We thought this was necessary because, the generators at the time, perhaps only would provide values from 0 to 1. Today there are tools in programming languages that provide random numbers in any desired range, and we can take advantage of this in some cases.

When we calculate the density of probability, we obtain the following expression.

$$\begin{aligned}
 |\psi_n(x)|^2 &= \frac{2}{L} \sin^2\left(\frac{n\pi}{L}x\right) \\
 D_f &: \{x \in \mathbb{R} \mid 0 \leq x \leq L_x\} \\
 R_f &: \left\{|\psi_n(x)|^2 \in \mathbb{R} \mid 0 \leq |\psi_n(x)|^2 \leq 2/L\right\}
 \end{aligned}
 \tag{9}$$

In a similar way to the example of the circle, we will use the function $|\psi_n(x)|^2$ as the filter to draw only the points within the domain, and ranges of the previous expression, by following the steps below:

1. Set a positive quantum number n .
2. Set an amount p -points to plot.
3. Set an integer i -counter and reset it to zero.
4. Generate a random $x \in [0, L]$.
5. Generate a random $w \in [0, 2/L]$.
6. If $w \leq |\psi_n(x)|^2$, then x and w are stored in the list and add +1 to i .
7. Repeat all steps from the 4th until i equals p .
8. Draw a graph with $\{x, w\}$ from the list.
9. Draw the curve $|\psi_n(x)|^2$ (optional).

The typical result is available in Figure 2.

5.2. The two-dimensional infinite potential well

In the two-dimensional stationary version of the same problem, the equation (3) is given by:

$$\begin{aligned}
 -\frac{\hbar}{2m} \left(\frac{\partial^2 \psi(x, y)}{\partial x^2} + \frac{\partial^2 \psi(x, y)}{\partial y^2} \right) V(x, y) \psi(x, y) \\
 = E\psi(x, y)
 \end{aligned}
 \tag{10}$$

The potential $V(x, y)$ in this case is:

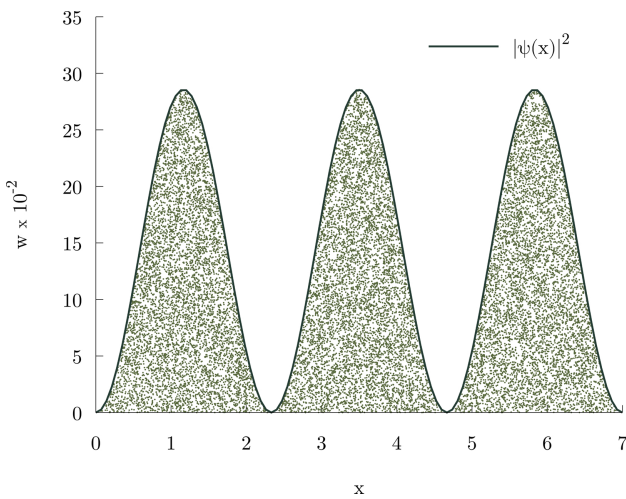


Figure 2: Two thousand points below the probability density function $|\psi(x)|^2$, for a well of infinite potential of dimension $L = 7$, and quantum number $n = 3$.

$$V(x, y) = \begin{cases} 0, & \text{if } 0 \leq x \leq L_x, 0 \leq y \leq L_y \\ +\infty, & \text{otherwise.} \end{cases}
 \tag{11}$$

where L_x and L_y are the dimensions of the well, and the solution of equation (10) is (see [36, 38]):

$$\psi_{n_x n_y}(x, y) = \frac{2}{\sqrt{L_x L_y}} \sin\left(\frac{n_x \pi}{L_x} x\right) \sin\left(\frac{n_y \pi}{L_y} y\right)
 \tag{12}$$

In the last expression, the numbers n_x and n_y are the positive integers that appear when the boundary conditions are applied. When calculating the probability density, we obtain the following relation:

$$\begin{aligned}
 |\psi_{n_x n_y}(x, y)|^2 &= \frac{4}{L_x L_y} \sin^2\left(\frac{n_x \pi}{L_x} x\right) \sin^2\left(\frac{n_y \pi}{L_y} y\right) \\
 D_f &: \{(x, y) \in \mathbb{R} \mid 0 \leq x \leq L_x, 0 \leq y \leq L_y\} \\
 R_f &: \left\{|\psi_{n_x n_y}(x, y)|^2 \in \mathbb{R} \mid 0 \leq |\psi_{n_x n_y}(x, y)|^2 \leq \frac{4}{L_x L_y}\right\}
 \end{aligned}
 \tag{13}$$

From this point on, the one-dimensional case algorithm is reproduced again for the function $|\psi_{n_x n_y}(x, y)|^2$, making some adaptations. In the 1st item we set the values of n_x and n_y , in the 4th and 5th items, we generate the x, y , and w in the intervals defined in domain D_f and range R_f , and then we make changes in the 6th and 8th items to plot the points that satisfy $w \leq |\psi_{n_x n_y}(x, y)|^2$. The typical result is shown in Figure 3.

5.3. The three-dimensional infinite potential well

In the three-dimensional stationary version of the potential well, the equation (3) is given by:

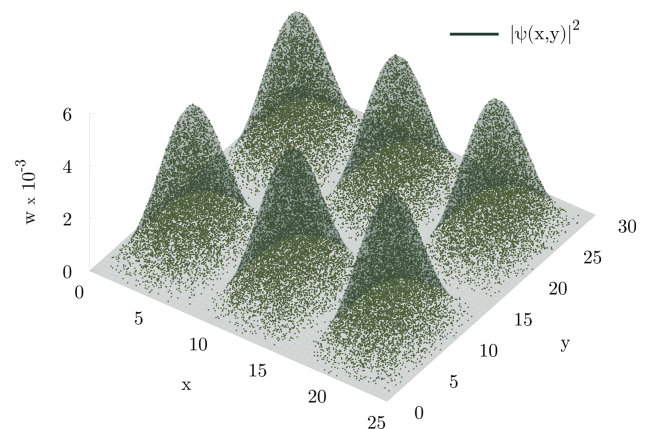


Figure 3: Five thousand points distributed below the probability density function $|\psi(x, y)|^2$, for a infinite potential well of dimensions $L_x = 25, L_y = 30$, and quantum numbers $n_x = 3$, and $n_y = 2$.

$$\left(-\frac{\hbar}{2m}\nabla^2 + V(x, y, z)\right)\psi(x, y, z) = E\psi(x, y, z) \quad (14)$$

where the potential $V(x, y, z)$ is:

$$V(x, y, z) = \begin{cases} 0, & \text{if } 0 \leq x \leq L_x, \\ & 0 \leq y \leq L_y, \ 0 \leq z \leq L_z \\ +\infty, & \text{otherwise.} \end{cases} \quad (15)$$

and the solution of the equation (14) is (see [34, 39]):

$$\psi_{n_x n_y n_z}(x, y, z) = \sqrt{\frac{8}{L_x L_y L_z}} \sin\left(\frac{n_x \pi}{L_x} x\right) \sin\left(\frac{n_y \pi}{L_y} y\right) \sin\left(\frac{n_z \pi}{L_z} z\right) \quad (16)$$

The constant L_z is the additional dimension of the box, and n_z is the third quantum number. When we calculate the density of probability, we obtain the following expression:

$$\begin{aligned} |\psi_{n_x n_y n_z}(x, y, z)|^2 &= \frac{8}{L_x L_y L_z} \\ &\sin^2\left(\frac{n_x \pi}{L_x} x\right) \sin^2\left(\frac{n_y \pi}{L_y} y\right) \sin^2\left(\frac{n_z \pi}{L_z} z\right) \\ D_f : \{(x, y, z) \in \mathbb{R} \mid 0 \leq x \leq L_x, 0 \leq y \leq L_y, \\ &0 \leq z \leq L_z\} \\ R_f : \{|\psi_{n_x n_y n_z}(x, y, z)|^2 \in \mathbb{R} \mid 0 \\ &\leq |\psi_{n_x n_y n_z}(x, y, z)|^2 \leq \frac{8}{L_x L_y L_z}\} \end{aligned} \quad (17)$$

Once again, we will change the algorithm in the one-dimensional case to include new coordinates, box dimensions, and quantum numbers. In this case, the program will provide x, y, z , and the term w . Although, there is no way to graph these four variables simultaneously, and we will represent the points that satisfy $w \leq |\psi_{n_x n_y n_z}(x, y, z)|^2$. We show a typical result in Figure 4:

5.4. The hydrogen atom

In previous problems involving particle in boxes, the potential $V(\vec{r})$ is responsible for the confinement of the particle, being infinite outside of limits and null in its interior. However, for the hydrogen atom alone⁶, i.e., a system with a single proton and its electron, the Coulomb force between them combined with the kinetic energy is responsible for the entire motion of the electron around

⁶The same applies to hydrogen-like ions such as He^+ , Li^{2+} , Be^{3+} , B^{4+} , and so on. The only difference is that the atomic number Z must be changed in each case.

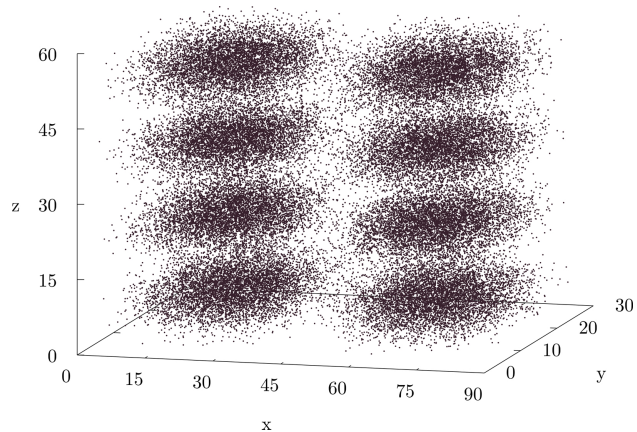


Figure 4: Fifty thousand points distributed in three-dimensional space, whose values of w are below the probability density function $|\psi(x, y, z)|^2$, for a infinite potential well, with dimensions $L_x = 90, L_y = 30, L_z = 60$, and quantum numbers $n_x = 2, n_y = 1, n_z = 4$.

the nucleus⁷. We show this relation mathematically in the following equation (see [32]):

$$-\left(\frac{\hbar}{2m}\nabla^2 + \frac{e^2}{4\pi\epsilon_0 r}\right)\psi(r, \theta, \phi) = E\psi(r, \theta, \phi) \quad (18)$$

The terms e and m are charge and mass of the electron, ϵ_0 is the constant of electrical permittivity in the vacuum. In this problem, we use spherical coordinates, i.e., r is the radius, ϕ e θ are the azimuthal and polar angles respectively (see Figure 5). The previous equation has the following analytical solution (see [40]):

$$\psi(r, \theta, \phi) = R(r)Y(\theta, \phi) \quad (19)$$

where the radial part $R(r)$ is:

$$R_{nl}(r) = -\sqrt{\left(\frac{2Z}{na_0}\right)^3 \frac{(n-l-1)!}{2n[(n+l)!]^3}} e^{-\rho/2} \rho^l L_{n+l}^{2l+1}(\rho) \quad (20)$$

and,

$$\rho(r) := \frac{2Zr}{na_0}. \quad (21)$$

The term n is the **radial quantum number**, which allows values $n = \{1, 2, 3, \dots\}$. The index l corresponds to the **azimuthal quantum number**, which can assume the values $l = \{0, \dots, n-1\}$. The constant a_0 is known as the Bohr radius, Z is the atomic number ($Z := 1$ for the hydrogen atom), and the term $L_{n+l}^{2l+1}(\rho)$ is the associated Laguerre polynomial, expressed as follows:

⁷If we were strict, we should consider these two bodies could vibrate, rotate around the center mass of the system, and along to their axis, or even interact with external fields. Although, all these effects are minimal and are usually neglected to ensure that the problem has an analytical solution [32].

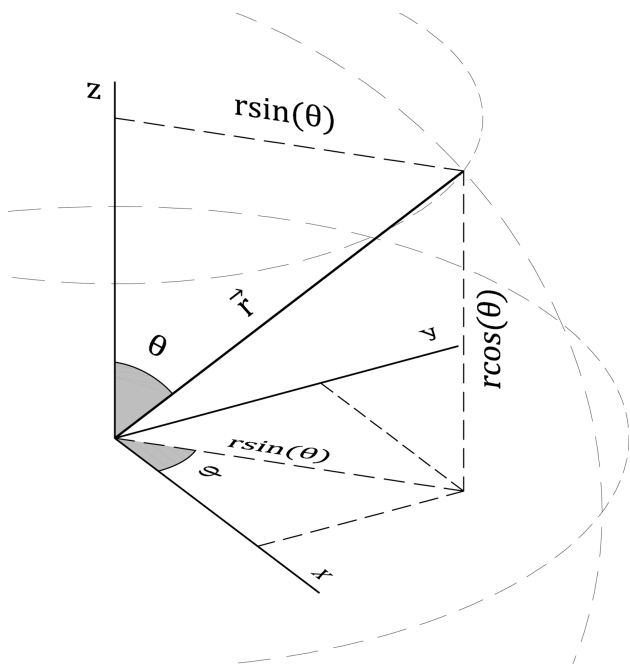


Figure 5: Spherical coordinates, where the radial distance is r , the polar angle is θ , and the azimuth angle is ϕ .

$$L_{n+l}^{2l+1}(\rho) = \sum_{k=0}^{n-l-1} (-1)^{k+1} \frac{[(n+l)!]^2}{(n-l-1-k)!(2l+1+k)!k!} \rho^k \quad (22)$$

In the angular part, we use the following relation:

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}} P_l^{|m|}(\cos(\theta)) e^{im\phi} \quad (23)$$

where the constant m is an integer, it represents the **magnetic quantum number** and can assume the values $m = \{-l, \dots, +l\}$. The term $P_l^{|m|}(\cos(\theta))$ comes from the next two relations (24) and (25), respectively known as polynomials of Legendre associated and Legendre polynomials.

$$P_l^{|m|}(x) = (-1)^{|m|} (1-x^2)^{|m|/2} \frac{d^{|m|}}{dx^{|m|}} P_l(x) \quad (24)$$

$$P_l(x) = \frac{1}{2^l l!} \frac{d^l}{dx^l} (x^2-1)^l \quad (25)$$

At this point, we have all the terms of the equation (19) available, and we can proceed by calculating the probability density numerically. Moreover, as an alternative, at least for the first quantum numbers, it is also possible to perform the calculations by the wave functions present in Table 1. Therefore, whichever path we choose, we must obtain the following probability density:

Table 1: First normalized wave functions of the hydrogen atom. This is an adaptation of the table 6.1 from [31].

n	l	m	$\psi_{n,l,m}(r, \theta, \phi)$
1	0	0	$\frac{1}{\sqrt{\pi a_0^3}} e^{-r/a_0}$
2	0	0	$\frac{1}{4\sqrt{2\pi a_0^3}} \left(2 - \frac{r}{a_0}\right) e^{-r/2a_0}$
2	1	0	$\frac{1}{4\sqrt{2\pi a_0^3}} \frac{r}{a_0} e^{-r/2a_0} \cos \theta$
2	1	± 1	$\frac{1}{8\sqrt{3\pi a_0^3}} \frac{r}{a_0} e^{-r/2a_0} \sin \theta e^{\pm i\phi}$
3	0	0	$\frac{1}{81\sqrt{3\pi a_0^3}} \left(27 - 18\frac{r}{a_0} + 2\frac{r^2}{a_0^2}\right) e^{-r/3a_0}$
3	1	0	$\frac{1}{81\sqrt{3\pi a_0^3}} \left(6 - \frac{r}{a_0}\right) \frac{r}{a_0} e^{-r/3a_0} \cos \theta$
3	1	± 1	$\frac{1}{81\sqrt{3\pi a_0^3}} \left(6 - \frac{r}{a_0}\right) \frac{r}{a_0} e^{-r/3a_0} \sin \theta e^{\pm i\phi}$
3	2	0	$\frac{1}{81\sqrt{6\pi a_0^3}} \frac{r^2}{a_0^2} e^{-r/3a_0} (3 \cos^2 \theta - 1)$
3	2	± 1	$\frac{1}{81\sqrt{\pi a_0^3}} \frac{r^2}{a_0^2} e^{-r/3a_0} \sin \theta \cos \theta e^{\pm i\phi}$
3	2	± 2	$\frac{1}{162\sqrt{\pi a_0^3}} \frac{r^2}{a_0^2} e^{-r/3a_0} \sin^2 \theta e^{\pm 2i\phi}$

$$|\psi_{nlm}(r, \theta, \phi)|^2 = |R_{nl}(r)|^2 |Y_m^l(\theta, \phi)|^2 \quad (26)$$

Note that in the equation (23) there is an exponential term that carries the imaginary number i , making $\psi_{nlm}(r, \theta, \phi)$ non-measurable, but $|Y_m^l(\theta, \phi)|^2$ does not contain the imaginary number i and independent of ϕ . So, $|\psi_{nlm}(r, \theta, \phi)|^2$ is also real and is a function only of the coordinates r and θ .

Now we got $|\psi_{nlm}(r, \theta, \phi)|^2$ already explained, and we can determine its domain and the range to apply the MCM, as we have done before. So, the first thing we need to do is to be able to switch from cartesian to a spherical coordinate system. To achieve this, we can extract this information from Figure 5 to get the relation (27) below.

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \arccos \left(\frac{z}{\sqrt{x^2 + y^2 + z^2}} \right) \quad (27)$$

For the domain of equation (26), we know that $\theta \in [0, \pi]$, and $r \in [0, +\infty)$. Since the radius r has no upper limit, it is impossible to search the electron in the entire space. Alternatively, we will check all volume by an arbitrary box which contains the location most likely to find the single electron.

Therefore, we place a box of dimensions Δx , Δy , Δz , centralized at the origin of the coordinate system. Then we use the relation (27) to evaluate $r_L = (1/2)\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$. This r_L is the magnitude of a vector that goes from the origin to any corner and will be used as the upper limit of the domain.

In respect of range, $|\psi_{nlm}(r, \theta, \phi)|^2$ is lower bounded by 0, and it reaches different top values depending on the quantum numbers. Unfortunately, finding a generic function that calculates the global maximum for each set

⁸The product between a number and its conjugate always gives a non-negative real number: $z \cdot z^* = |z|^2$

involving all of them is no easy task. A first alternative would be to find a solution analytically for each case, but this process becomes increasingly tedious as quantum numbers grow. Another option would be to use some searching process to seek global maximum numerically [41], after all, we have the computer available. Instead, let us proceed more smoothly.

The simplest way is to split the space⁹ into a 2D regular grid, over the r and θ “axis”, and for each node (grid point) we calculate $|\psi_{nlm}(r, \theta, \phi)|^2$. Then we update a variable that stores the highest value M found, and then increase it to ensure it is above the global maximum of this function. The Figures 6, 7, and 8 exemplify this method.

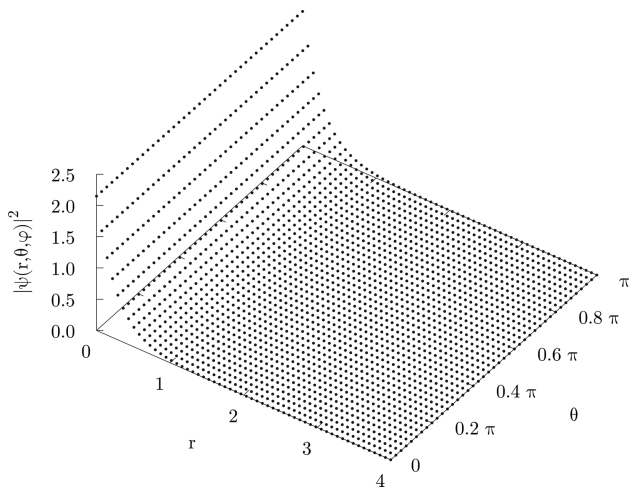


Figure 6: The probability density as a function of r and θ for the hydrogen atom, with quantum numbers $n = 1, l = 0, m = 0$. The points are projections of nodes in the function $|\psi(r, \theta, \phi)|^2$.

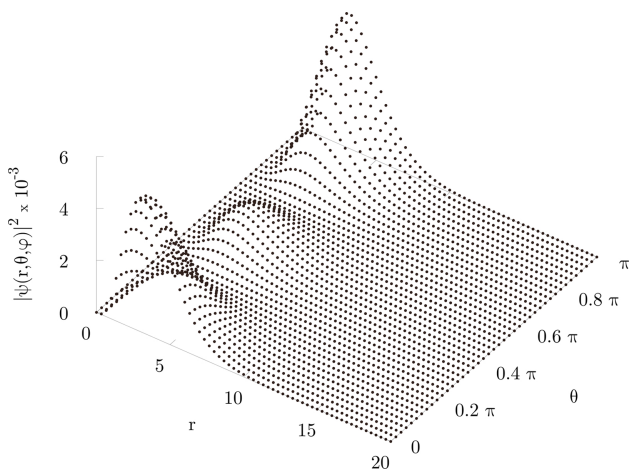


Figure 7: The probability density as a function of r and θ for the hydrogen atom, with quantum numbers $n = 3, l = 2, m = 0$. The points are projections of nodes in the function $|\psi(r, \theta, \phi)|^2$.

⁹For example, r and θ starts at 0 and reaches r_L and π respectively, so if the length r_L is divided by 5 and angle π by 4, then we have 30 nodes ($30 = (5 + 1) \times (4 + 1)$) to check.

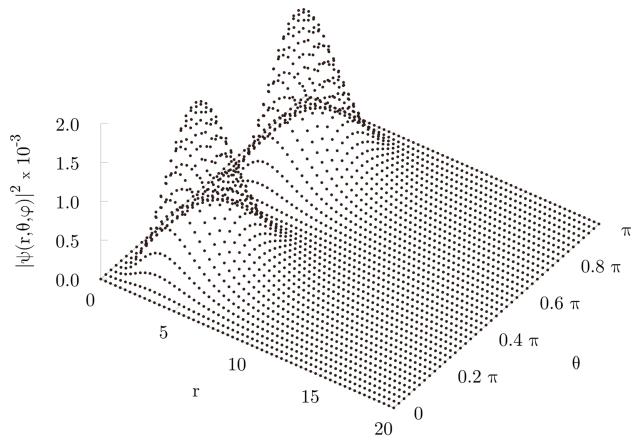


Figure 8: The probability density as a function of r and θ for the hydrogen atom, with quantum numbers $n = 3, l = 2, m = 1$. The points are projections of nodes in the function $|\psi(r, \theta, \phi)|^2$.

This procedure may not be elegant, but it is practical. If the reader is uncomfortable about this, do the same with the problem in 1D, then he will realize that the method will discard points above the function $|\psi_n(x)|^2$ in any way.

The final comment should be made about the Legendre associated polynomials presented in the equation (24). Some environments operate with algebraic or symbolic computations to interpret equations like that, but to reproduce this method in several programming languages, we must soften this step. In this case, it is better to use suitable libraries for this purpose, because the implementation of simpler algorithms may cause problems of numerical instability [42]. A good choice for the C++ programming language is the *BOOST* library [43] and the *SHTOOLS* [44] for FORTRAN and Python.

Finally, with all this information gathered, we have the conditions to represent the orbitals of the hydrogen atom. For that, we will adapt the algorithm seen in the examples of the confined particle in boxes, presented in the following.

1. Set the quantum numbers $n \in [1, +\infty)$, $l \in [0, n - 1]$, $m \in [-l, l]$.
2. Set the box dimensions $\Delta x, \Delta y$ e Δz .
3. Estimate an upper limit for M above of $|\psi_{nlm}(r, \theta, \phi)|^2$.
4. Set an amount p -points to plot.
5. Start an integer i -counter and reset it to zero.
6. Generate a random $x \in [-\Delta x/2, \Delta x/2]$, $y \in [-\Delta y/2, \Delta y/2]$ e $z \in [-\Delta z/2, \Delta z/2]$.
7. Evaluate r and θ as the functions of x, y e z .
8. Generate a random $w \in [0, M]$.
9. If $w \leq |\psi_{nlm}(r, \theta, \phi)|^2$, then x, y and z are stored in the list, and add +1 to i .
10. Repeat all steps from the 6th until i equals p .
11. Make a graph with the coordinates from the list.

To estimate the upper limit M , we calculate r_L to set up the grid, then we scan it to get an approximate global

maximum of $|\psi_{nlm}(r, \theta, \phi)|^2$, and finally, we add 5% over this value found. To make this grid, we suggest that use nearly 50 nodes per Angström along of the r radius, and 160 nodes over the θ angle.

For plotting the orbitals in $2D$, turn the box into a rectangle by choosing the null value for some side. For example, if the interest is in the $x-y$ plane, then $\Delta z = 0$, and so on. We show some orbitals in Figures 9, 10 and 11.

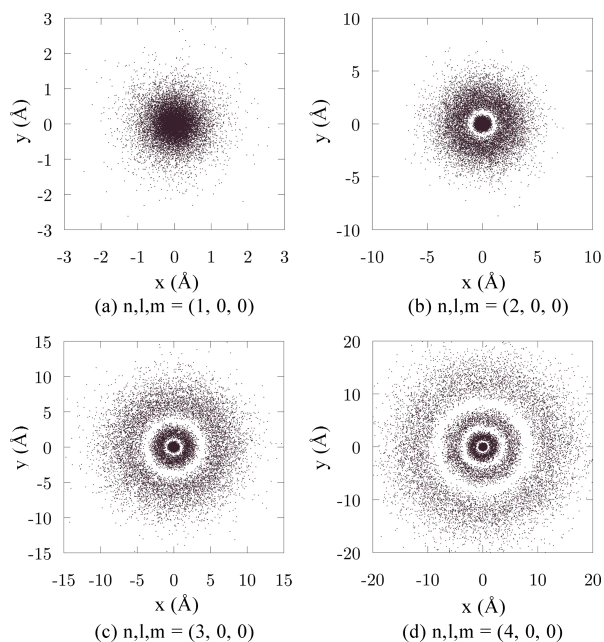


Figure 9: The simulated hydrogen atom with fifteen thousand points.

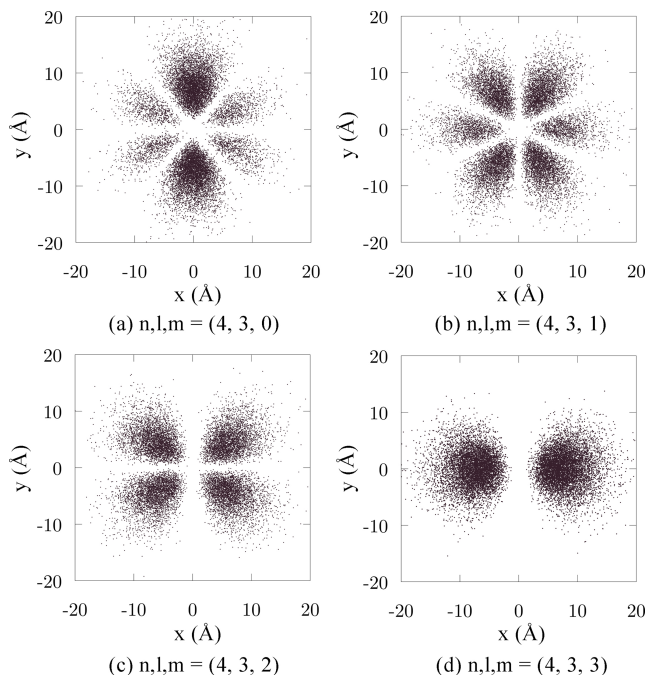


Figure 10: The simulated hydrogen atom with fifteen thousand points.

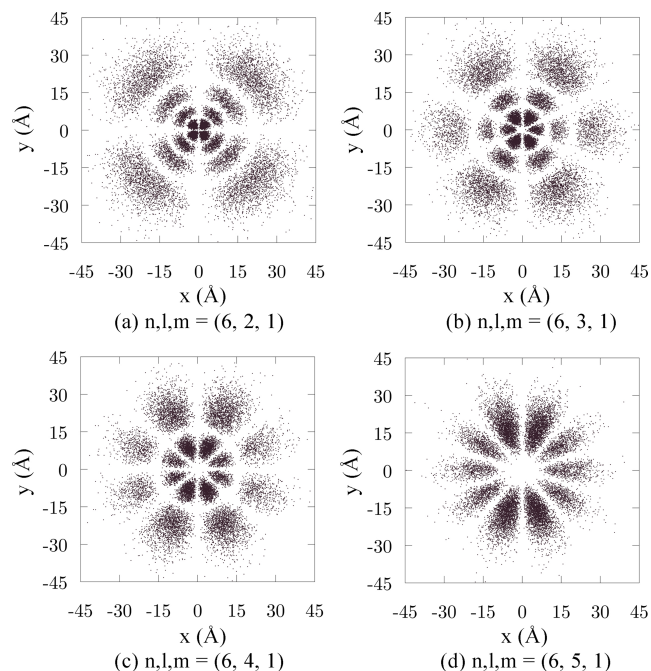


Figure 11: The simulated hydrogen atom with fifteen thousand points.

6. Conclusion

This paper shows in a heuristic way that it is easy to represent the orbitals of the hydrogen atom using basic quantum mechanics and coding skills.

We also made all source code available at a third-party platform [45], for any uses under the MIT license.

Acknowledgment

This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001*.

References

- [1] A. Lakhtakia, *Models and Modelers of Hydrogen* (World Scientific Publishing Company, Singapura 1996).
- [2] H. Fischler and M. Lichtfeldt, *International Journal of Science Education* **14**, 181 (1992).
- [3] D.T. Cromer, *J. Chem. Educ.* **45**, 626 (1968).
- [4] B. Coto, A. Arencibia and I. Suárez, *Computer Applications in Engineering Education* **24**, 765 (2016).
- [5] B.G. Moore, *J. Chem. Educ.* **77**, 785 (2000).
- [6] M. Goto and V.M. Aquino, *Semina: Ciências Exatas e Tecnológicas* **13**, 255 (1992).
- [7] V.M. Aquino, V.C. Aguilera-Navarro, M. Goto and H. Iwamoto, *American Journal of Physics* **69**, 788 (2001).
- [8] E.A. Silver, *Journal of the Operational Research Society* **55**, 936 (2004).
- [9] M. Shlomo and S. Mordechai, *Applications of Monte Carlo Method in Science and Engineering* (IntechOpen, London, 2011).

- [10] D.P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics, Second Edition* (Cambridge University Press, New York, 2005).
- [11] V.L. Líbero, Rev. Bras. Ens. Fis. **22**, 346 (2000).
- [12] R. da Silva and J.R.D. de Felício, Rev. Bras. Ens. Fis. **24**, 103 (2002).
- [13] E. Arashiro, J.R.D. de Felício and U.H.E. Hansmann, Physical Review E **73**, 040902 (2006).
- [14] F.M. Ruziska, E. Arashiro and T. Tomé, Physica A: Statistical Mechanics and its Applications **489**, 56 (2018).
- [15] N.R.S. Ortega, C.F.S. Pinheiro, T. Tomé and J.R.D. de Felício, Physica A: Statistical Mechanics and its Applications **255**, 189 (1998).
- [16] F. Keesen, A. Castro e Silva, E. Arashiro, C.F.S. Pinheiro, Ecological Modelling **344**, 38 (2017).
- [17] A. Castro e Silva and A.T. Bernardes, Physica A: Statistical Mechanics and its Applications **352**, 535 (2005).
- [18] C.A. Waldspurger, T. Hogg, B.A. Huberman, J.O. Kephart and W.S. Stornetta, IEEE Transactions on Software Engineering **18**, 103 (1992).
- [19] F. James, Reports on Progress in Physics **43**, 1145 (1980).
- [20] R.E. Caffisch, Acta Numerica **7**, 1 (1998).
- [21] H. Gould, J. Tobochnik and W. Christian *An introduction to computer simulation methods: applications to physical systems-3rd Edition* (Pearson Addison Wesley, 2007).
- [22] R.Y. Rubinstein, *Simulation and the Monte Carlo Method* (Wiley, Hoboken, 1981).
- [23] Y. Badis, *Computational Physics* (WORLD SCIENTIFIC, Singapore, 2017).
- [24] D.P. Kroese, T. Taimre and Z.I. Botev, *Handbook of Monte Carlo Methods* (John Wiley & Sons, Hoboken, 2011).
- [25] J.K.S. William and L. Dunn, *Exploring Monte Carlo Methods* (Elsevier, Cambridge, 2011).
- [26] M. Lee, *C++ Programming for the Absolute Beginner, 2nd Edition* (Course Technology Press, Boston, 2009).
- [27] <https://ieeexplore.ieee.org/document/977250>.
- [28] https://www.academie-sciences.fr/pdf/dossiers/Broglie/Broglie_pdf/CR1923_p507.pdf.
- [29] <https://www.nature.com/articles/112540a0>.
- [30] V.V. Raman and P. Forman, Historical Studies in the Physical Sciences **1**, 291 (1969).
- [31] A. Beiser, *Concepts of Modern Physics* (McGraw-Hill, Boston, 2002).
- [32] I. Levine, *The Hydrogen Atom* (Pearson, Boston, 2014).
- [33] D.J. Griffiths *Introduction to Quantum Mechanics* (Cambridge University Press, Cambridge, 2017).
- [34] N. Zettili, *Quantum Mechanics: Concepts and Applications* (Wiley, Hoboken, 2009).
- [35] J. Von Neumann, National Bureau of Standards Applied Mathematics **3**, 36 (1951).
- [36] G.D.W.W. Bauer, in *University Physics with Modern Physics* (McGraw-Hill, New York, 2011).
- [37] [https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_\(Physical_and_Theoretical_Chemistry\)/Quantum_Mechanics/05.5%3A_Particle_in_Boxes/Particle_in_a_1-Dimensional_box](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Quantum_Mechanics/05.5%3A_Particle_in_Boxes/Particle_in_a_1-Dimensional_box).
- [38] [https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_\(Physical_and_Theoretical_Chemistry\)/Quantum_Mechanics/05.5%3A_Particle_in_Boxes/Particle_in_a_2-Dimensional_box](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Quantum_Mechanics/05.5%3A_Particle_in_Boxes/Particle_in_a_2-Dimensional_box).
- [39] [https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_\(Physical_and_Theoretical_Chemistry\)/Quantum_Mechanics/05.5%3A_Particle_in_Boxes/Particle_in_a_3-Dimensional_box](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Quantum_Mechanics/05.5%3A_Particle_in_Boxes/Particle_in_a_3-Dimensional_box).
- [40] L. Pauling, *Introduction to quantum mechanics: with applications to chemistry* (Dover Publications, New York, 1985).
- [41] J. Nocedal and S. Wright *Numerical Optimization* (Springer Science & Business Media, Berlin, 2006).
- [42] W. Press, *Numerical recipes in C : the art of scientific computing* (Cambridge University Press, Cambridge, 1992).
- [43] https://www.boost.org/doc/libs/1_56_0/libs/math/doc/html/index.html.
- [44] <https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2018GC007529>.
- [45] https://github.com/pedrohflobo/hydrogen_orbitals_by_Monte_Carlo_Method.

O Modelo Baseado em Agentes

Modelos são uma forma matemática, física ou conceitual de representar um fenômeno que é difícil de se observar diretamente. Eles são desenhados para explicar ou prever comportamentos objetos reais ou sistemas e são aplicados em um amplo intervalo de áreas científicas como física, ecologia e economia. Os modelos são aproximações de objetos e sistemas, e por isto devemos enfatizar que eles possuem as suas limitações. E isto, muitas vezes implica em uma constante demanda por aprimoramento.

Na história dos modelos atômicos por exemplo, o modelo de Bohr deu um passo adiante em relação ao de Rutherford, porque ao introduzir a teoria quântica, explicou com sucesso, como cargas negativas podem permanecer orbitando ao redor de cargas positivas. Porém, ele não é tão preciso quanto ao de Schrödinger, que além de incorporar os todos avanços de modelos atômicos anteriores, este modelo considera o conceito de dualidade partícula-onda. Por isto, ao analisarmos um modelo, devemos nos ater a sua proposta e suas limitações.

A proposta de modelos variam. O modelo planetário Kepleriano foi construído a partir de dados experimentais coletados por Brahe, e pelo menos para Kepler o objetivo era validar a hipótese do heliocentrismo. Já os modelos preditivos como os climáticos por exemplo, são baseados em dados de fenômenos passados, e são mesclados com análises matemáticas para que sirvam para prever ocorrências futuras. No entanto, por causa de sua natureza estocástica ou por simplificações convenientes, os modelos deste tipo não se comprometem a descrever o futuro ou mesmo os fenômenos com exatidão. Outro tipo de modelo e que será mostrado adiante é classificado como modelo baseado em agentes (MBAs), e que é usado para simular interações entre agentes autônomos, para estudar seus efeitos no sistema quando visto na macroescala. Os MBAs são construídos através de simples regras e a partir delas sugere a explicação de fenômenos emergentes.

Uma das aplicações típicas de MBAs é no estudo de formação de nuvens de pássaro e cardumes, que possivelmente se formam como uma estrutura de defesa contra predadores. Apesar de percebermos a sincronia do grupo como um todo, sugerindo que um indivíduo se comporta como líder, provavelmente são as interações locais entre pássaros ou peixes

que produz o efeito coletivo. Motivado por esta hipótese, Couzin [9] desenvolveu um MBAs propondo três regras simples, com parâmetros ajustáveis e que se aplicam somente aos agentes do mesmo grupo. Neste modelo, Couzin mostrou como reproduzir fielmente o comportamento de bandos e cardumes na ausência de predadores. Porém, posteriormente ao adicionar uma quarta regra, Couzin [10] simulou a dispersão dos agentes, tipicamente quando sob ataque de predadores. Neste caso, esta perturbação adicional se propaga em toda a rede, criando locais vazios ao redor do agressor, inclusive promovendo a separação momentânea do grupo.

No caso do trabalho de nossa autoria, a ser apresentado na sequência, também nos inspiramos na dinâmica de perseguição entre predadores e presas, para entender se seria possível emergir o comportamento de manada, a partir de indivíduos que quase não interagem entre si. Neste caso, negligenciamos efeitos de estruturas sociais de grupos de animais, assim como a personalidade ou qualquer outra particularidade dos indivíduos. Na prática, nos estabelecemos uma rede que representa o ambiente e sobre esta criamos campos potenciais que servem de base para locomoção de indivíduos. Ou seja, dependendo de algumas condições, através destes campos, os predadores tenderão a se aproximar das presas, enquanto estas tenderão a se afastar dos predadores.

Além deste modelo propor as simplificações já mencionadas, possivelmente a maior diferença desde MBAs com a realidade é o espaço métrico que utilizamos. No mundo real, quando não é necessário cálculos de agrimensura, as distâncias geralmente são obtidas de acordo com a métrica euclideana, enquanto utilizamos a Manhattan neste modelo. Isto é devido ao fato de ser muito mais fácil implementar desta forma, porque naturalmente o computador trabalha com variáveis discretizadas.

O MMC também está presente neste modelo, porque toda vez que um agente se movimenta, a sua posição de destino é determinada aleatoriamente. Além disso, como será visto adiante, realizamos cálculos de curvas que representam médias que usamos como formas de avaliar a formação de *clusters*. A medida em que o número de amostras aumentam, mais delineadas ficam estas curvas médias, por consequência da atuação do MMC.

Na sequência apresentaremos o trabalho intitulado “Formação de *clusters* em um

modelo de perseguição e fuga” e que foi publicado no evento 8º MCSul. Em seguida apresentaremos outros resultados deste trabalho.

4.1 Formação de clusters em um modelo de perseguição e fuga



Formação de clusters em um modelo de perseguição e fuga

Lobo, P.H.F.¹

Arashiro, E.²

Lazo, M.J.³

Mendonça, S.M.⁴

Resumo: Predadores procurando por presas, assim como presas fugindo de seus predadores são capazes de exibir comportamentos coletivos. Inspirados por este tipo de efeito, foi desenvolvido um modelo estocástico baseado em agentes, com o objetivo de estudar a movimentação e a formação de *clusters* de seus indivíduos. Neste projeto, foi montada uma rede virtual discretizada bidimensional, com condições periódicas de contorno, onde se movimentam dois tipos de agentes, sendo que um deles se locomove como presa e é programado para se afastar do segundo tipo, que se comporta como predador. Nesse modelo, como o interesse principal foi estudar os padrões formados pela dinâmica de perseguição e fuga, as proporções destes agentes são mantidas fixas em cada simulação. As regras de movimentação são baseadas em uma caminhada aleatória assimétrica, que faz com que os dois tipos de agentes, com seus respectivos comportamentos, executem um movimento browniano quando muito afastados. Porém a dinâmica de perseguição começa a ficar mais intensa quando estes dois se aproximam. Para analisar as condições nas quais os *clusters* emergem, foram variadas as concentrações dos dois tipos de agentes e um parâmetro σ , que atua como um mediador, amplificando ou atenuando as "forças" de atração/repulsão entre os indivíduos. Foram realizadas simulações que revelaram padrões de movimentação de agentes inicialmente posicionados de forma aleatória na rede, assim como a contagem do número médio de *clusters* ao longo do tempo.

Palavras-chave: modelo baseado em agentes, dinâmica estocástica, *clusters* emergentes, perseguição.

¹Bacharel em Física, Programa de Pós-Graduação em Física, Universidade Federal do Rio Grande, pedrohflobo@gmail.com

²Doutor em Física Aplicada à Medicina e Biologia, Programa de Pós-Graduação em Física, Universidade Federal do Rio Grande, earashiro@furg.br

³Doutor em Física, Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal do Rio Grande, matheuslazo@furg.br

⁴Graduanda em Física Bacharelado com ênfase em Física Médica, Universidade Federal do Rio Grande, suziellim@gmail.com

1 Introdução

O modelo de Lotka-Volterra clássico (Lotka, 1926; Volterra, 1990 apud Arashiro et al., 2008), também conhecido como modelo presa-predador, foi o primeiro a descrever matematicamente como a população de presas e predadores variam com o passar do tempo. Ele consiste em duas equações diferenciais ordinárias acopladas para a densidade de presas e predadores, e trata-se de uma analogia às leis de ação das massas (Schnakenberg, 1977). No entanto, este modelo não leva em consideração, pelo menos de forma explícita, a estrutura espacial do ambiente onde as espécies coexistem (Arashiro, Rodrigues, de Oliveira, & Tomé, 2008). Portanto, quando é de interesse saber detalhes da movimentação e a disposição espacial das espécies, devemos abordar o tema de outra forma, e uma alternativa corriqueira, é através de modelos baseados em agentes (MBAs).

Os MBAs são uma classe de modelos computacionais utilizados para simular ações, comportamentos e interações entre indivíduos ou grupos, com o objetivo de explorar o impacto dos agentes no sistema quando visto como um todo (Clarke, 2014). Além disto, eles têm aplicações em áreas interdisciplinares que envolvem estudos de comportamento organizacional (Secchi, 2015), comportamento do consumidor (Garifullin, Borshchev, & Popkov, 2007), tráfego de veículos (Huynh, Cao, & Wickramasuriya, 2014), evacuação (Kirchner & Schadschneider, 2002) e epidemias (Perez & Dragicevic, 2009). Na biologia eles são empregados também para descrever distribuições espaciais de indivíduos presentes em um ecossistema, com a adição inclusive de uma dinâmica populacional (Satulovsky & Tomé, 1994; Keesen, e Silva, Arashiro, & Pinheiro, 2017; Ruziska, Arashiro, & Tomé, 2018; Argolo et al., 2016; Boccara, Roblin, & Roger, 1994).

Motivados também por aplicações na área da ecologia e inspirados por comportamentos de bando que observamos em grandes grupos de animais como os gnus, por exemplo, propomos um modelo baseado em agentes (MBA) com o objetivo principal de estudar efeitos que advém da dinâmica de perseguição e fuga que ocorrem entre predadores e presas.

No entanto, salientamos que neste trabalho embora tenhamos nos inspirado no modelo de Lotka-Volterra, não se trata de um modelo presa-predador, uma vez que o objetivo aqui é estudar efeitos de bandos, e este não considera morte e

nascimento de nenhum indivíduo de ambas espécies. Além disto, por uma questão de simplificação, ao contrário do modelo clássico, que é determinístico, abordamos o problema fundamentando em princípios estocásticos. Isto é oportuno para compensar eventuais variedades internas do sistema, como diferenças entre os indivíduos ou mesmo no ambiente, típicas de sistemas complexos. Isto garante uma evolução única de uma mesma rede para cada número pseudo-aleatório fornecido como entrada.

2 Fundamentação teórica

2.1 Modelos Baseados em Agentes

Seria conveniente neste momento definir "agentes" antes de aplicarmos ao nosso propósito, mas infelizmente, apesar do seu uso comum, não existe uma versão precisa e universalmente aceita. (Chen, 2012) faz algumas comparações entre diferentes definições encontradas na literatura, em seguida, argumenta que existem duas características de agentes que são acordados por pessoas de área afins: autonomia e habilidade social. Isto os tornam capazes de se comportar de forma independente, mas com flexibilidade para reagir ao ambiente e interagir com outros indivíduos.

Isto significa que na prática, podemos interpretar agentes como indivíduos autônomos inseridos em um ambiente, capazes de interagir com os demais e que executam um conjunto de regras para os quais foram projetados. Em ambiente de simulação computacional por exemplo, os agentes em seus respectivos ambientes podem representar veículos nas estradas, moradores em um prédio ou mesmo vírus em seus hospedeiros.

2.2 A rede

Para que possamos utilizar o computador como ferramenta de trabalho, devemos realizar algumas adaptações necessárias da realidade a ser simulada. A primeira delas é propormos uma discretização no espaço e tempo. Isto é realizado ao inserirmos os agentes em uma rede bidimensional, composta por uma quantidade finita de células iguais, por onde se locomovem "saltando" de uma célula a outra em intervalos iguais de tempo.

Existem várias maneiras de elaborar a rede, e para nosso propósito, utilizaremos

uma das mais simples e comuns. Ela possui formato retangular, construída com células quadradas e admite condições periódicas de contorno nas bordas. Isto significa que a superior se comunica com a inferior, assim como as laterais esquerda e direita se conectam. Ou seja, podemos representar esta rede como a superfície de um toroide.

Além disso, o deslocamento é baseado na geometria do táxi, isto é, o agente só pode migrar de uma célula central para uma das células vizinhas ao norte, sul, leste ou oeste. Em outras palavras, este tipo configuração admite movimentação somente sobre a vizinhança de *Neumann* de intervalo $r = 1$.

Como nosso modelo opera com uma grande quantidade de agentes, é necessário inserirmos uma origem do sistema de coordenadas global para mapearmos a rede posteriormente. Sendo assim, iremos etiquetar as células por suas posições genericamente representadas pelo par (i, j) . A escolha da célula origem, assim como a sequência numérica que representam os eixos de coordenadas são arbitrários, desde que i e j sejam valores discretizados em intervalos regulares. Por conveniência trabalharemos com a representação matricial. Isto é: i aumenta de cima para baixo, enquanto j cresce da esquerda para a direita, assumindo valores inteiros $i = \{1, 2, \dots, m\}$ e $j = \{1, 2, \dots, n\}$.

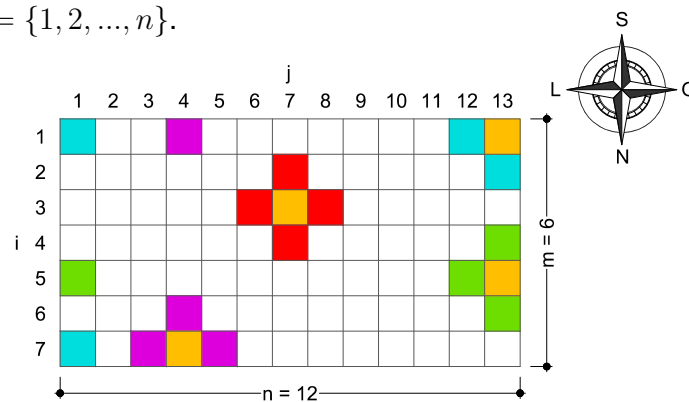


Figura 1: Algumas células destacadas em laranja com suas respectivas vizinhanças de *Neumann* em diversas cores. Esta rede considera condições periódicas de contorno nas bordas.

2.3 Os agentes

Embora todo organismo vivo possa ser único quando o examinamos em níveis de detalhamento mais elevados (Elsasser, 1981), simplificaremos ao considerarmos que todos os indivíduos de uma mesma classe são indistinguíveis. Este procedimento é frequentemente observado em MBAs que lidam com "agentes biológicos simulados". Deste modo, é denominado então como "x-agente" quando a intenção é fazer referência

aos membros das duas únicas classes presentes neste projeto. Entretanto, quando for conveniente fazer menção a presas e predadores de maneira específica, aqui denotados por H e P respectivamente, substituiremos estes em alguns índices representados genericamente pelo referido 'x'.

Esta distinção é necessária porque os membros de cada uma dessas classes se comportam de maneiras diferentes ao se movimentarem. Isto porque, embora sejam semelhantes, eles compartilharão somente dois dos três objetivos explicitadas a seguir:

Classe predador (P):

- i. Procurem se **aproximar das presas**.
- ii. Não migrem para células ocupadas.
- iii. Procurem se manter em movimento.

Classe presa (H):

- i. Procurem se **afastar dos predadores**.
- ii. Não migrem para células ocupadas.
- iii. Procurem se manter em movimento.

2.4 O campo e a movimentação

O conjunto formado por cada classe de agentes produzem separadamente dois campos escalares, genericamente chamados de C^x e é determinado através da equação (1):

$$C_{i,j}^x = \sum_{k=1}^K \exp \left[-\frac{(r_i^{x,k} + r_j^{x,k})^2}{2\sigma^2} \right] \quad (1)$$

onde:

$$\begin{aligned} r_i^{x,k} &= \min(|i - i^{x,k}|, m - i + i^{x,k}) \\ r_j^{x,k} &= \min(|j - j^{x,k}|, n - j + j^{x,k}) \end{aligned} \quad (2)$$

A equação (1) nada mais é do que um somatório sobre gaussianas bidimensionais com desvio padrão σ , elaboradas na geometria do táxi. O numerador no argumento da exponencial corresponde ao quadrado da distância de *Manhattan*, calculado entre uma célula de coordenada (i, j) e um k -ésimo x -agente situado em $(i^{x,k}, j^{x,k})$. Note que $r_i^{x,k}$ e $r_j^{x,k}$ explicitados em (2), tratam-se de distâncias ao longo dos eixos i e j , adaptadas para rede $m \times n$ com periodicidade nas bordas. Além do mais, embora estes estados $C_{i,j}^x$ sejam capazes de mapear a rede inteira, é necessário calcular apenas os valores correspondentes a vizinhança imediata, que são as quatro posições para as quais os agentes efetivamente possuem uma probabilidade de se deslocarem. Por isso é conveniente isolar tais estados de células em uma tupla

$$T^x = \{C_{i-1,j}^x, C_{i,j+1}^x, C_{i+1,j}^x, C_{i,j-1}^x\}.$$

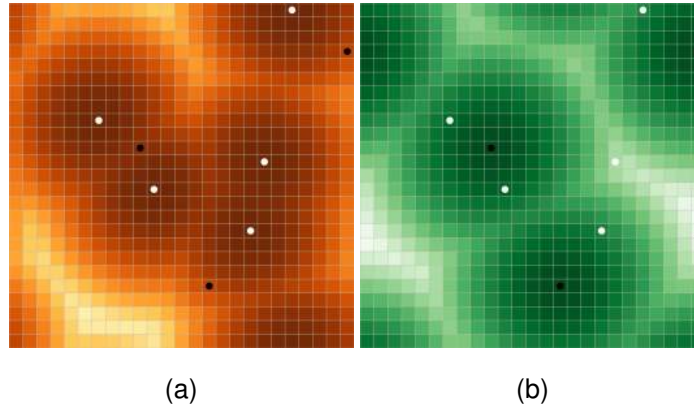


Figura 2: Campo C^H gerado por 5 presas (pontos brancos) em (a) e C^P produzidos por 3 predadores (pontos pretos) em (b). Em ambos, os casos eles foram obtidos através da equação (1), sendo que os locais mais escuros e mais claros representam os maiores e menores valores dos campos respectivamente.

Este campo C^x produzido é proposto desta forma para mapear gradientes das "zonas de ameaça e de predação" associadas aos agentes, que são na prática, difíceis de determinar e são responsáveis por orientá-los durante a dinâmica de perseguição. Por isto, o termo σ é flexível, para fazermos ajustes em cada caso. Já o somatório é inserido porque os animais agrupados podem ser mais facilmente percebidos pelos seus "inimigos", tornando o campo mais intenso em locais mais concentrados.

Para promover o deslocamento dos agentes na rede, procuramos criar regras baseadas no primeiro objetivo das duas classes. Isto é, do ponto de vista dos predadores, que procuram perseguir as presas, calculamos a probabilidade P_n^P de um agente migrar para cada uma das n -células da sua vizinhança ortogonal, tomando uma relação de proporção direta a T_n^H . No caso das presas, como elas tendem a se afastar dos predadores, estabelecemos a probabilidade P_n^H por uma relação de proporção inversa a T_n^P .

É necessário considerar também, que dois ou mais indivíduos, devido ao segundo objetivo, não podem coexistir no mesmo espaço ao mesmo tempo. Além disso, como todos eles são atualizados simultaneamente, se uma célula fica vaga, nenhum outro poderá ocupá-la a menos que demais agentes, quando possível, já tenham se movimentado. Em outras palavras, no mesmo quadro, aqui considerado como atualização completa da rede, não é permitida a permanência em regiões já visitadas. Por isto, vamos considerar um estado u , no qual é $u_{i,j} = 1$, quando se a célula em

questão está vazia e $u_{i,j} = 0$ se está ocupada. Da mesma maneira, consideremos um estado w , no qual $w_{i,j} = 0$ se algum agente esteve nela no atual quadro, e $w = 1$ caso contrário. Com isto determinamos os conjuntos $S = \{u_{i-1,j}, u_{i,j+1}, u_{i+1,j}, u_{i,j-1}\}$ e $R = \{w_{i-1,j}, w_{i,j+1}, w_{i+1,j}, w_{i,j-1}\}$.

Por fim, através da combinação da equação 1 com os termos T^x , S e R , determinamos a probabilidade de um agente migrar para cada uma das quatro células de sua vizinhança ortogonal, obtida através das relações 3 e 4 apresentadas em sequência. Explicitamente, estas probabilidades são dadas pelo conjunto $P^x = \{p_{i-1,j}^x, p_{i,j+1}^x, p_{i+1,j}^x, p_{i,j-1}^x\}$.

$$P_n^P = \begin{cases} S_n R_n T_n^H \left(\sum_{n=1}^4 S_n R_n T_n^H \right)^{-1} & , \text{ se } \sum_{n=1}^4 S_n R_n \neq 0 \\ 0 & , \text{ caso contrário} \end{cases} \quad (3)$$

$$P_n^H = \begin{cases} \frac{S_n R_n}{T_n^P} \left(\sum_{n=1}^4 \frac{S_n R_n}{T_n^P} \right)^{-1} & , \text{ se } \sum_{n=1}^4 S_n R_n \neq 0 \\ 0 & , \text{ caso contrário} \end{cases} \quad (4)$$

Note que a probabilidade calculada através das equações anteriores, determina se os agentes poderão se mover. Desta forma, executamos a regra baseada no terceiro e último objetivo comum às duas classes. Sendo assim, é conveniente agora explicar como todas estas equações trabalham juntas na etapa a seguir:

2.5 O algoritmo

Inicialmente, é utilizado como entrada o valor fixado de σ , as dimensões m e n da rede, assim como a quantidade de presas H e predadores P . Em seguida os agentes são inseridos em posições aleatórias, sem que mais de um ocupe a mesma célula, com suas devidas posições temporárias coincidindo com as localizações de seus agentes.

Na sequência, inicia-se um *loop*, que consiste em selecionar uma das duas classes, depois um dos seus membros e atualizar sua posição na rede. Ou seja, quando um agente é escolhido, é calculada a probabilidade P^x dele se deslocar para sua vizinhança. E caso ele se mova, é memorizado de onde ele veio, como uma posição temporária, para que nenhum outro agente a ocupe no mesmo quadro.

Quando todos os indivíduos forem verificados, estas posições temporárias serão

descartadas, e será contabilizado o número de *clusters*, repetindo o *loop anterior* até que seja alcançada uma quantidade desejável de quadros. No final, temos para cada quadro o número de *clusters*, armazenado usualmente em forma de arquivos com extensão ".dat" ou em um banco de dados (*MongoDB, MariaDB, MySQL, etc*).

Desta maneira, podemos utilizar os procedimentos anteriores para produzirmos resultados suficientes para realizar um tratamento estatístico, e completar o ciclo através do último *loop*. Isto consiste em determinamos o número médio de *clusters* em cada quadro, acompanhados das medidas de dispersão, os quais nos limitaremos aos desvio padrão neste trabalho. Ou seja, a saída desta etapa é um arquivo com o número médio de *clusters* para cada quadro. A figura 3 na sequência, ilustra todo o algoritmo com mais detalhes.

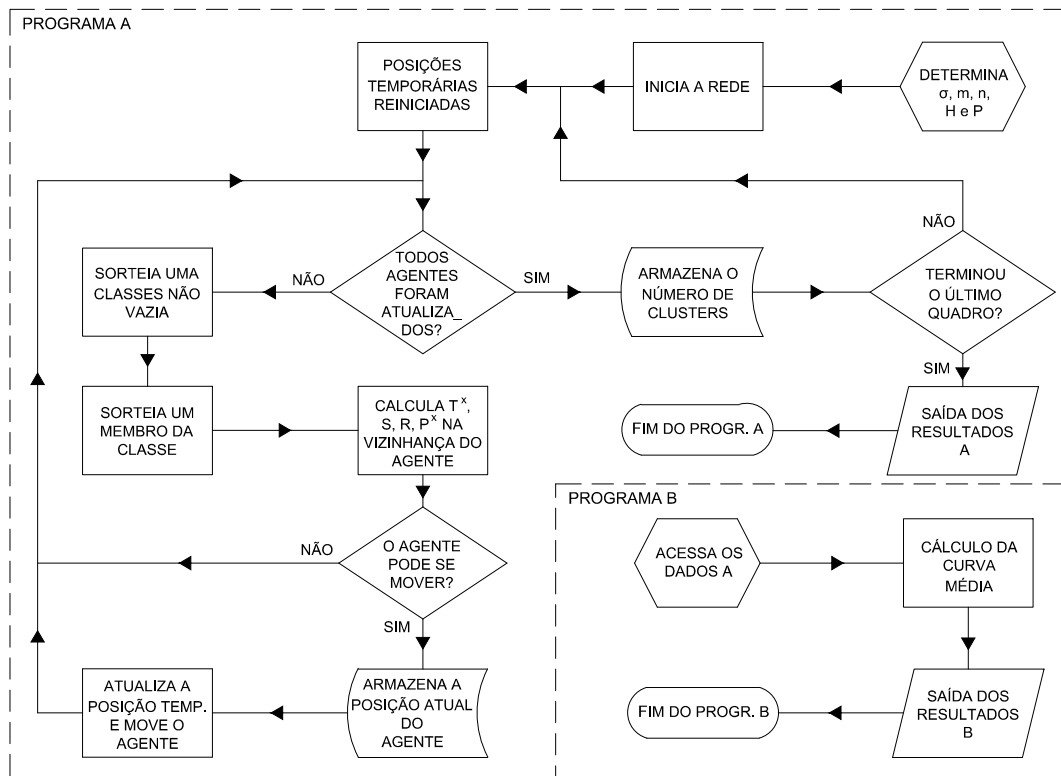


Figura 3: Algoritmo em forma de diagrama.

3 Simulações

Para realizar as simulações, em um primeiro ensaio, as concentrações dos dois tipos de agentes são mantidos constantes, enquanto o valor de σ é variado. O objetivo nesta etapa é investigar se este termo é capaz de influir na "atração/repulsão" produzida entre

predadores e presas.

Posteriormente, em um segundo teste, é mantido fixo o valor de σ , assim como o número de presas e são realizadas algumas concentrações diferentes de predadores. Com isso, espera-se verificar sob quais condições é possível formar grupos ou rebanhos de predadores ou presas.

Por fim, verificamos a evolução do número médio de *clusters* z formados por presas, influenciado pelos termos σ e H . Nas simulações, são preservadas em cada ensaio as condições iniciais como dimensões da rede, estabelecidas inicialmente por H , P e σ . Para tal verificação, realizamos um grande número de ensaios, nos quais o gerador de números aleatórios garante uma evolução distinta do MBA para cada simulação.

Para realizar esta contagem, consideramos um *cluster* como o número de agentes de mesma classe, conectadas através de suas vizinhanças, mesmo quando isolados, aqui considerados como *clusters* unitários.

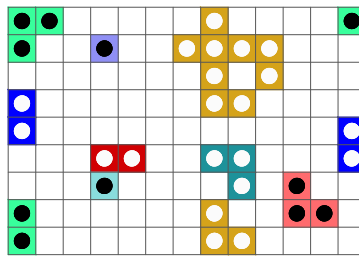


Figura 4: *Clusters* de diversos tamanhos formados por predadores e presas, representados por círculos pretos e brancos respectivamente. Nesta imagem, as cores diferenciadas nas células representam seus *clusters*.

4 Resultados

Inicialmente, fizemos alguns testes rápidos, que nos revelou na prática, como os membros das duas classes se comportam. Como previsto, os predadores perseguiram as presas e houve formação de *clusters* em algumas casos (ver figura 5).

Em seguida, para iniciar a investigação das condições necessárias para formar grupos, adotamos um valor fixo para σ e arbitramos $[H]$ relativamente alto, quando comparado com $[P]$. Neste caso, foi observado que as presas passaram a formar rebanhos, enquanto os predadores tenderam a ficar isolados (figura 6a). Em seguida, quando invertemos estas quantidades, verificamos que as poucas presas foram cercadas por vários predadores (figura 6b).

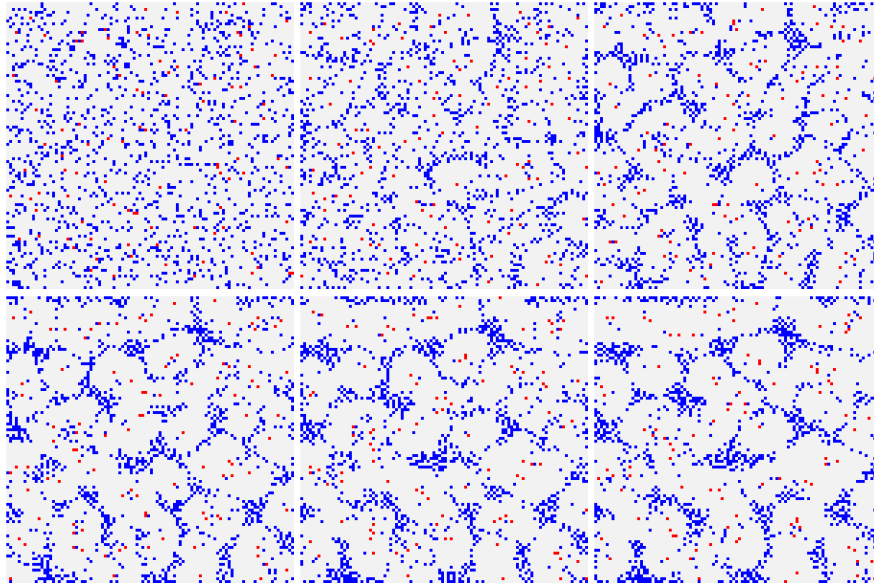
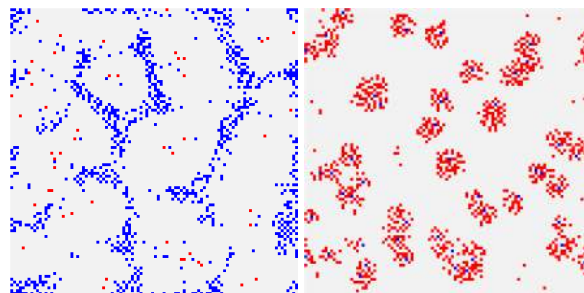


Figura 5: Evolução temporal a partir do início da simulação, para uma rede 100×100 , a cada 4 quadros com $\sigma = 1.5$. O espaço foi populado com $[H] = 12.0\%$ de presas em azul e $[P] = 1.0\%$ de predadores em vermelho.



(a)

(b)

Figura 6: Redes de dimensões 100×100 , utilizando $\sigma = 1.5$ e evoluída para 30 quadros. Em (a) ela é composta por $[P] = 0,5\%$ de predadores em vermelho e $[H] = 10,0\%$ de presas em azul. Já em (b) a rede é populada por $[P] = 10,0\%$ de predadores e $[H] = 0,5\%$ de presas.

Ao variarmos a concentração $[P]$, mantendo σ e $[H]$ constante, verificamos que a diferença entre a quantidade de membros das duas classes favoreceu a formação de *clusters* (figura 7). Na presença de muitos predadores (figura 7a), é possível que o excesso deles tenha "pressionado" as presas por todos os lados, diminuindo os espaços para movimentação e conseqüentemente, dificultando a formação de grandes *clusters*. Por outro lado, se há poucos predadores, (figura 7c), a ação do campo produzido por eles pouco influencia nas presas mais distantes, fazendo com que elas passem a executar uma caminhada aleatória tradicional nestes locais. Ou seja, a tendência é criar rebanhos

somente nas proximidades dos predadores.

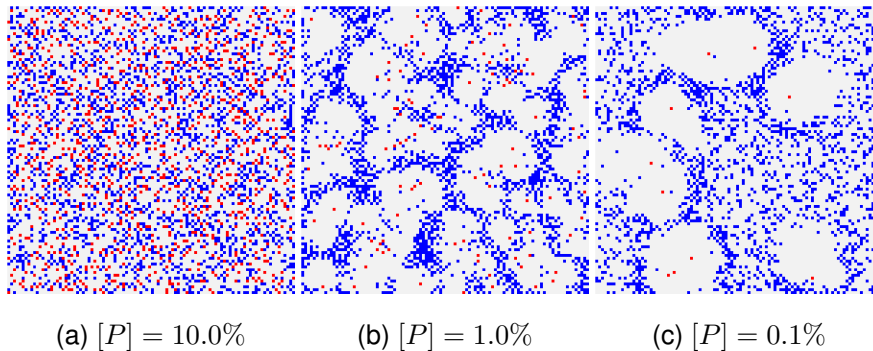


Figura 7: Diferentes simulações de concentrações de predadores $[P]$ em vermelho, para uma rede 100×100 , evoluída a 30 quadros. A concentração de presas inseridas é $[H] = 20.0\%$ em azul, para $\sigma = 1.5$.

Quando as concentrações $[H]$ e $[P]$ são mantidas fixas, porém é variado o valor de σ , observamos que nos extremos de um intervalo de $\sigma = (0, 10]$, as presas passam a não se agrupar (ver figuras 8a e 8f). Entretanto, para valores intermediários, notamos a formação de grupos que parecem mais compactos quando $\sigma = 1.5$ (ver figura 8d). Já para $\sigma \approx 0$, os agentes produzem campos em forma de picos. Ou seja, eles são intensos nas suas origens e praticamente nulos em suas vizinhanças. Porém, quando σ é suficientemente grande, aqui representado por $\sigma = 10.0$, também não se unem porque os agentes produzem campos homogêneos. Isto implica na probabilidade equilibrada deles se movimentarem para qualquer sentido, executando movimentos brownianos.

Na sequência, para quantificarmos os números de *clusters* formados com o passar do tempo, variando $[H]$ e σ , realizamos algumas simulações, cujo os resultados serão apresentados nas próximas imagens.

De acordo com as figuras 9 e 10, entendemos que após uma pequena instabilidade, os números médios de *clusters* ($\langle z \rangle$) diminuiram. Como o número de agentes é sempre constante, então isto significa que houve no início uma grande quantidade de pequenos *clusters* que se fundiram. O mesmo efeito é observado na maioria dos casos, quando a concentração de H é a mesma para vários ensaios com diferentes valores de σ (ver figura 11).

Nesta circunstância, escolha de $\sigma = 1.5$ entre todos os simulados, foi a que aparentemente mais favoreceu a formação de rebanhos. Porém, para $\sigma = 10.0$, houve um aumento no número de *clusters* unitários, ou seja, neste caso, a rede não apresentou

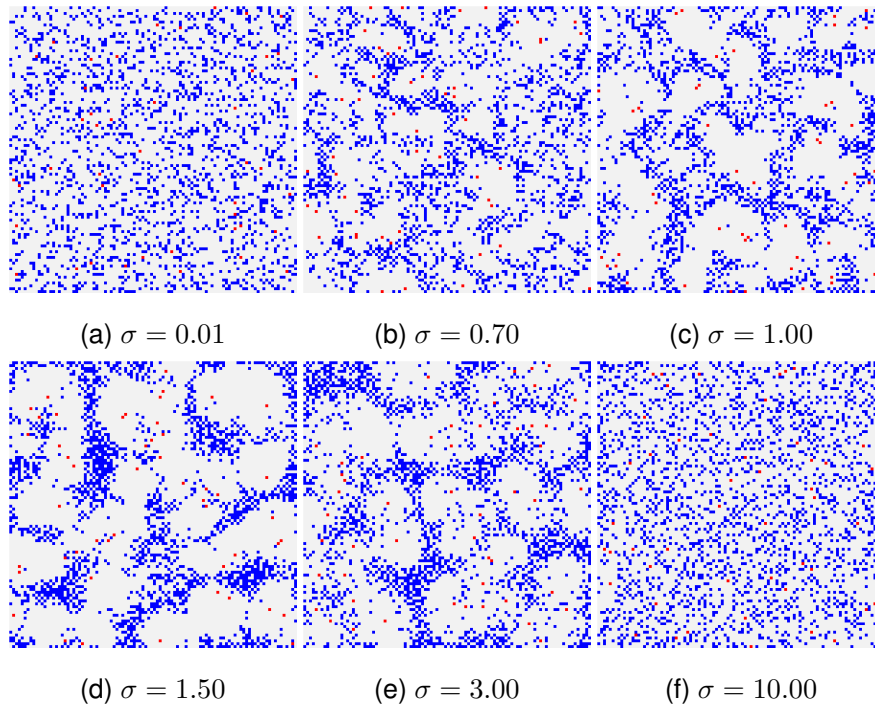


Figura 8: Simulações evoluídas para 30 quadros sobre uma rede de 100×100 , com valores diversos de σ . O espaço foi populado com $[H] = 20.0\%$ de presas em azul, e $[P] = 1.0\%$ de predadores em vermelho.

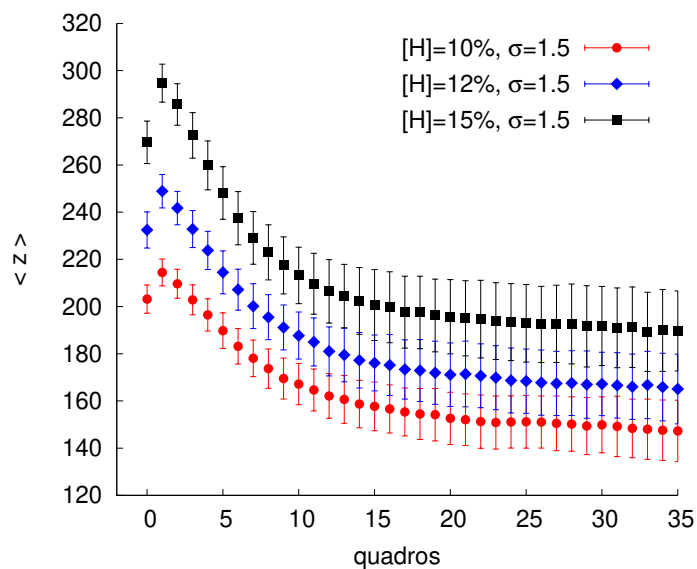


Figura 9: Número médio de *clusters* formados ao longo do tempo para uma rede de dimensões 50×50 com a utilização de um valor fixo de $\sigma = 1.5$. Para cada quadro as médias foram obtidas realizando-se 500 ensaios e a concentração de predadores foi mantida a $[P] = 1\%$ em relação ao tamanho da rede.

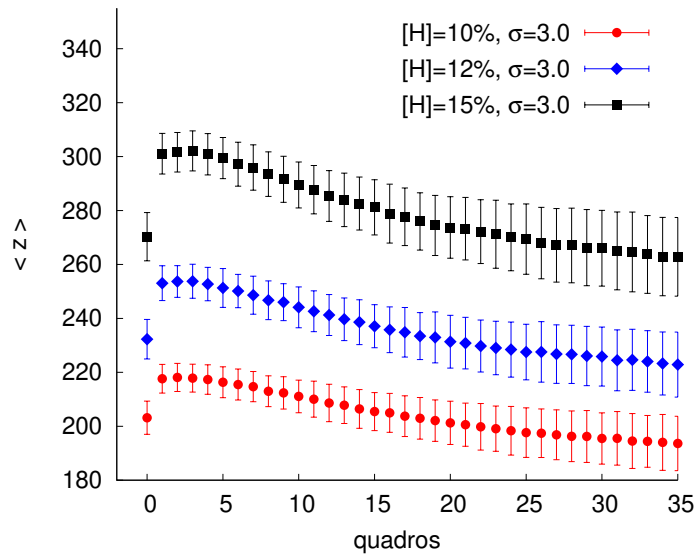


Figura 10: Número médio de *clusters* formados ao longo do tempo para uma rede de dimensões 50×50 com a utilização de um valor fixo de $\sigma = 3.0$. Para cada quadro, as médias foram obtidas realizando-se 500 ensaios e a concentração de predadores foi mantida a $[P] = 1\%$ em relação ao tamanho da rede.

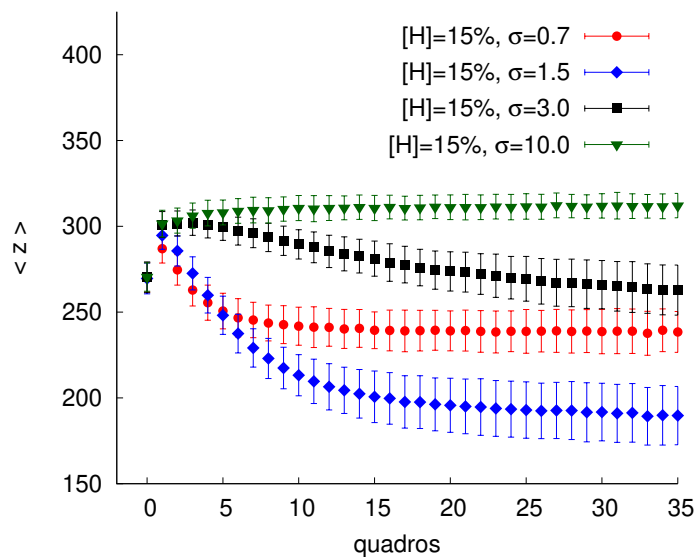


Figura 11: Comparação entre números médios de *clusters* formados ao longo do tempo para diferentes valores de σ adotados. Nestas simulações, para cada quadro as médias foram obtidas realizando-se 500 ensaios. A concentração de predadores e presas foram mantidas constantes com $[P] = 1\%$ e $[H] = 15\%$ em relação a rede de tamanho 50×50 .

tendência a formar grandes grupos de presas. Este resultado coincide com o observado na figura 8f.

5 Conclusão e considerações finais

As simulações revelaram que este MBA é capaz de reproduzir a formação de bandos de animais, frequentemente observados na natureza, utilizando regras simples focadas na interação entre presas e predadores. Neste contexto, foi possível observar as condições que determinam o fenômeno de emergência de *clusters* de tamanhos variados, em ambiente de simulação computacional. Ou seja, isto é realizado ao controlarmos as concentrações de presas e predadores, assim como o desvio padrão σ , que atua como mediador de intensidade com que cada um dos membros destas classes interagem com seus "inimigos".

Um efeito interessante verificado, é que os agentes executam o tempo inteiro uma caminhada aleatória assimétrica, especialmente quando membros de classes opostas estão próximos, mas a medida em que se afastam, eles passam a exibir um típico movimento browniano. Isto é conveniente porque, é razoável propor que a dinâmica de perseguição e fuga se intensifica a medida que estes "inimigos" se aproximam.

Agradecimentos

Pedro Henrique F. Lobo agradece a CAPES pelo suporte financeiro para realização deste projeto e a Mariana Gauterio Tavares pela revisão.

Referências

- Arashiro, E., Rodrigues, a. L., de Oliveira, M. J., & Tomé, T. (2008). Time correlation function in systems with two coexisting biological species. *Physical Review E*, 77(6), 061909. doi: 10.1103/PhysRevE.77.061909
- Argolo, C., Barros, P., Tomé, T., Arashiro, E., Gleria, I., & Lyra, M. L. (2016). Threshold of coexistence and critical behavior of a predator-prey stochastic model in a fractal landscape. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(8), 083204.
- Boccaro, N., Roblin, O., & Roger, M. (1994, Dec). Automata network predator-prey model with pursuit and evasion. *Phys. Rev. E*, 50, 4531–4541. doi: 10.1103/PhysRevE.50.4531
- Chen, L. (2012, jun). Agent-based modeling in urban and architectural research: A brief literature review. *Frontiers of Architectural Research*, 1(2), 166–177. doi: 10.1016/j.foar.2012.03.003
- Clarke, K. C. (2014). Handbook of Regional Science. In M. M. Fischer & P. Nijkamp (Eds.), *Journal of regional science* (Vol. 54, pp. 1217–1231). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-23430-9

- Elsasser, W. M. (1981). Principles of a new biological theory: A summary. *Journal of Theoretical Biology*, 89(1), 131 - 150. doi: [https://doi.org/10.1016/0022-5193\(81\)90182-X](https://doi.org/10.1016/0022-5193(81)90182-X)
- Garifullin, M., Borshchev, A., & Popkov, T. (2007). Using Anylogic and Agent-Based Approach To Model Consumer Market. *EUROSIM 2007, Ljubljana, Slovenia.*, 1–5.
- Hanson, J. (2012). A visit to taxicab geometry. *International Journal of Mathematical Education in Science and Technology*, 43(8), 1109-1123. doi: 10.1080/0020739X.2012.662291
- Huynh, N., Cao, V., & Wickramasuriya, R. (2014). An Agent Based Model for the Simulation of Road Traffic and Transport Demand in A Sydney Metropolitan Area. *8th International Workshop on Agents in Traffic and Transportation (ATT-2014)*, 1–7. doi: 10.13140/2.1.4023.2961
- Jennings, N. R., Sycara, K., & Wooldridge, M. (1998, Mar 01). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1), 7–38. doi: 10.1023/A:1010090405266
- Keesen, F., e Silva, A. C., Arashiro, E., & Pinheiro, C. (2017). Simulations of populations of *sapajus robustus* in a fragmented landscape. *Ecological Modelling*, 344, 38 - 47. doi: <https://doi.org/10.1016/j.ecolmodel.2016.11.003>
- Kirchner, A., & Schadschneider, A. (2002). Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, 312(1-2), 260–276. doi: 10.1016/S0378-4371(02)00857-9
- Lotka, A. J. (1926). Elements of physical biology. *Science Progress in the Twentieth Century (1919-1933)*, 21(82), 341–343.
- Perez, L., & Dragicevic, S. (2009). An agent-based approach for modeling dynamics of contagious disease spread. *International Journal of Health Geographics*, 8(1), 1–17. doi: 10.1186/1476-072X-8-50
- Ruziska, F. M., Arashiro, E., & Tomé, T. (2018). Stochastic dynamics for two biological species and ecological niches. *Physica A: Statistical Mechanics and its Applications*, 489, 56 - 64. doi: <https://doi.org/10.1016/j.physa.2017.07.016>
- Satulovsky, J. E., & Tomé, T. (1994, Jun). Stochastic lattice gas model for a predator-prey system. *Phys. Rev. E*, 49, 5073–5079. doi: 10.1103/PhysRevE.49.5073
- Schnakenberg, J. (1977). G. Nicolis und I. Prigogine: *Self-Organization in Nonequilibrium Systems*. From Dissipative Structures to Order through Fluctuations. J. Wiley & Sons, New York, London, Sydney, Toronto 1977. 491 Seiten, Preis: £ 20.-, \$ 34.-. *Berichte der Bunsengesellschaft für physikalische Chemie*, 82(6), 672–672. doi: 10.1002/bbpc.197800155
- Secchi, D. (2015, mar). A case for agent-based models in organizational behavior and team research. *Team Performance Management: An International Journal*, 21(1/2), 37–50. doi: 10.1108/tpm-12-2014-0063
- Volterra, V. (1990). *Leçons sur la théorie mathématique de la lutte pour la vie*. Paris: J. Gabay.

Resultados complementares

A partir de agora faremos uma reavaliação, mas de outra perspectiva, verificando a influência do termo σ e concentrações de presas e predadores. Além disso, realizamos simulações com a intenção de observar o efeito em diferentes tamanhos de rede.

Sendo assim, tudo que foi apresentado anteriormente se aplica nesta parte a seguir. Porém, devemos adicionar o termo “perímetro” no contexto de MBAs, porque este será útil futuramente. Um *cluster* formado por uma das classes de agentes, possui uma borda delimitada por agentes mais externos. Nós consideramos como “perímetro” o número de células externas a esta borda, da sua vizinhança de Neumann imediata. Logo, um *cluster* unitário possui perímetro de 4 unidades, assim um *cluster* de dois agentes possui perímetro de 6 unidades.

Inicialmente simulamos algumas redes quadradas de tamanhos variados, com a intenção de determinarmos o tempo (ou quadro) no qual cada dimensão atinge o estado estacionário. A melhor forma que encontramos de compararmos redes de diferentes dimensões foi calcular a razão entre o perímetro médio de *clusters* em relação ao número de células da rede, para situações que de antemão sabíamos que *clusters* se formariam.

Para exemplificarmos, uma rede 30×30 possui 900 células. Desta forma, especificamente para este tamanho de rede, além de fixamos o valor de σ e mantermos as concentrações de presas e predadores contantes, em cada quadro dividimos a média de perímetros de *clusters* de presas por 900.

De acordo com o resultado observado na figura 1, nós consideramos que para redes de dimensões 50×50 e menores, as simulações atingiram o estado estacionário antes do quadro = 1000. Este resultado é importante, porque nos permite determinar um tamanho de rede ideal para se trabalhar a partir de agora. Redes pequenas demais não são convenientes de se utilizar, porque a inserção de poucos agentes implica em um drástico aumento de concentrações, enquanto redes grandes exigem muito esforço computacional e portanto, muito no tempo de simulação. Por isto, para as próximas simulações, adotaremos o redes de dimensões 50×50 por considerarmos como um tamanho

intermediário e adequado.

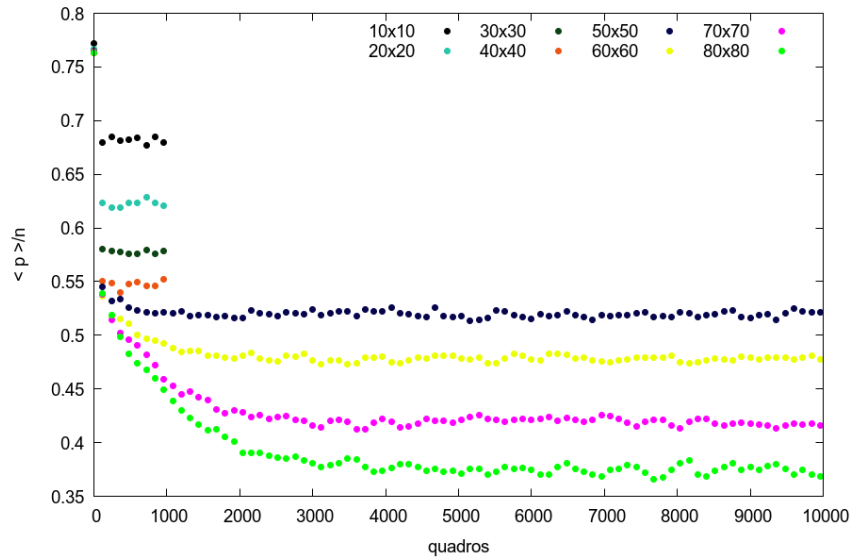


Figura 1: Evolução temporal para diferentes tamanhos de rede, da razão entre o perímetro médio de presas e seus respectivos número células. Nesta simulação foram utilizadas 200 amostras e para $\sigma = 3.00$, $[H] = 30\%$ e $[P] = 1\%$.

A partir de agora, iremos verificar a influência de várias combinações de quantidade de agentes na rede, por isto nós varremos o espaço procurando alterar as concentrações dos dois tipos de agentes e introduzimos uma nova concentração $[V]$ que chamamos de “concentração de vazios”, fazendo referência aos espaços da rede desocupados. Para calcularmos o número médio de *clusters* de V , em cada quadro, após todos os agentes da classe P e H se movimentem na redes, nos contabilizamos todos os espaços desocupados para obter os *clusters*, assim como fizemos com as outras classes.

Para mostrarmos simultaneamente as variáveis $[H]$, $[P]$ e $[V]$, produzimos um triângulo de concentrações, o qual expressa valores de maneira idêntica ao triângulo de Maxwell. O procedimento é análogo ao que gera o espectro cromático visível, e nós apenas substituímos as intensidades que representam as cores primárias aditivas (*red*, *green*, *blue*), por estas concentrações de agentes.

As figuras 2 e 3 a seguir expressam o número médio de *clusters*, para 36 concentrações distintas de ambos os tipos de agentes, no início da simulação e no seu estado estacionário. Estas imagens são adequadas para representar as situações quando a menor concentração de agentes é de pelo menos 10%, ao contrário dos casos anteriores simulados próximos a 1%. Lembramos também que um aumento no número de *clusters* representa uma diminuição

dos seus tamanhos médios, e portanto, nesta escala, os locais próximos a 400 são os mais populados com pequenos *clusters*, enquanto se forma um único grande rebanho em regiões representado por 1.

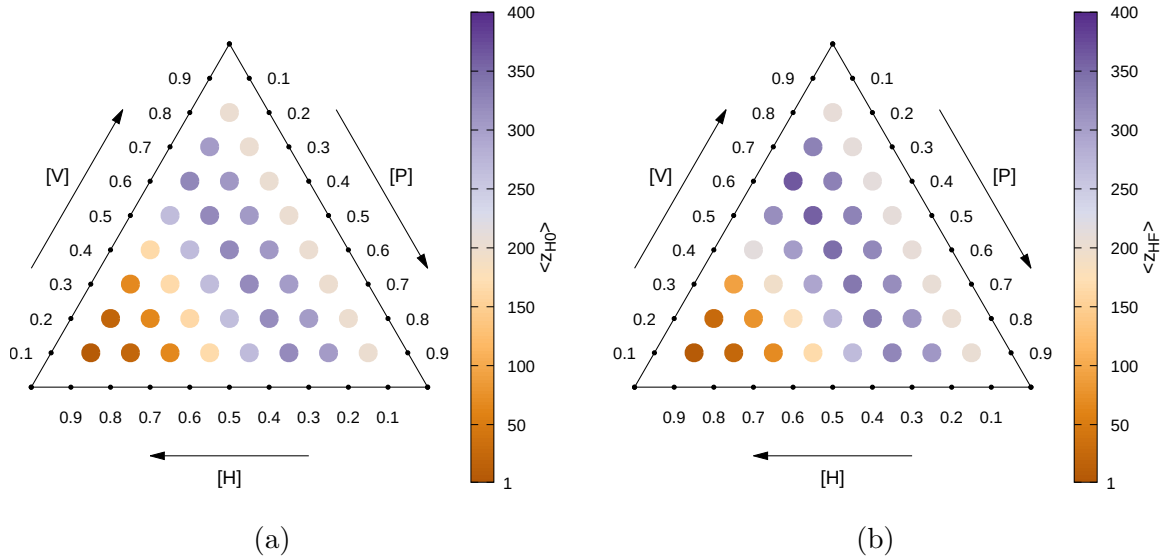


Figura 2: Triângulos de concentrações para número médio de *clusters* de presas inicial (quadro = 0) em (a) e final (quadro = 1000) em (b). A rede utilizada possui dimensões 50×50 e adotamos $\sigma = 3.0$. Os termos $[H]$, $[P]$ e $[V]$ representam concentrações de presas, predadores e “vazios” respectivamente. Nesta simulação foram utilizadas 200 amostras.

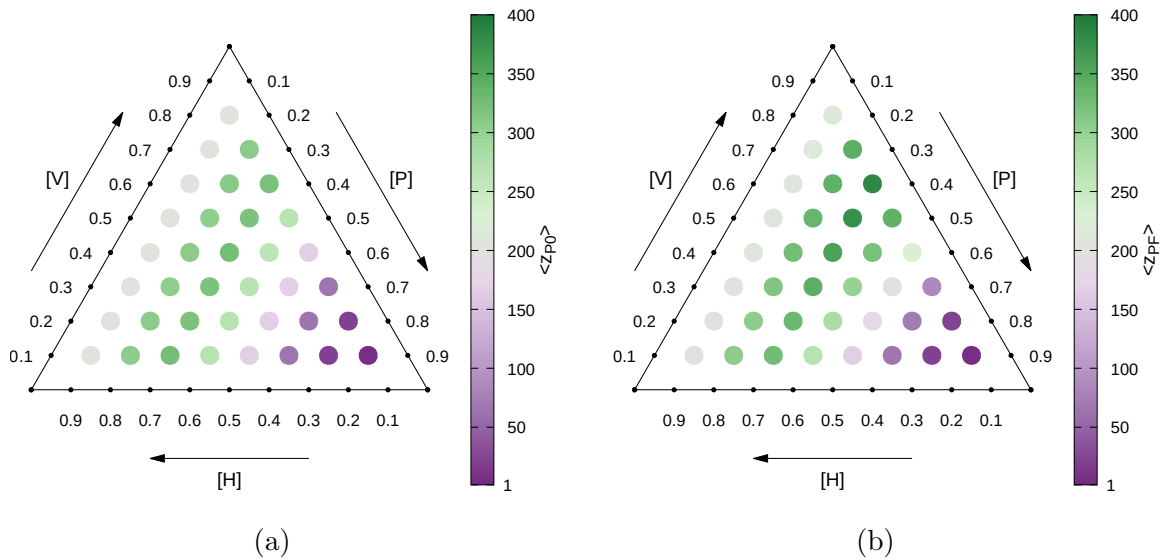


Figura 3: Triângulos de concentrações para número médio de *clusters* de predadores inicial (quadro = 0) em (a) e final (quadro = 1000) em (b). A rede utilizada possui dimensões 50×50 e adotamos $\sigma = 3.00$. Os termos $[H]$, $[P]$ e $[V]$ representam concentrações de presas, predadores e “vazios” respectivamente. Nesta simulação foram utilizadas 200 amostras.

Comparando o lados esquerdos e direitos das figuras 2 e 3, é difícil averiguar se houve formação de *clusters* com o passar do tempo, uma vez que estes expressam valores absolutos e as diferenças são muito sutis. Sendo assim, prosseguiremos calculando em cada

caso, a razão entre o número médio de *clusters* no estado estacionário, pelo número inicial médio de *clusters* ($\langle z_F \rangle / \langle z_0 \rangle$). Esta abordagem nos permite comparar as situações as quais formam rebanhos de presas e predadores, uma vez que a deposição de agentes na rede no tempo inicial é sempre uniforme. Os resultados estão presentes na figura 4 a seguir.

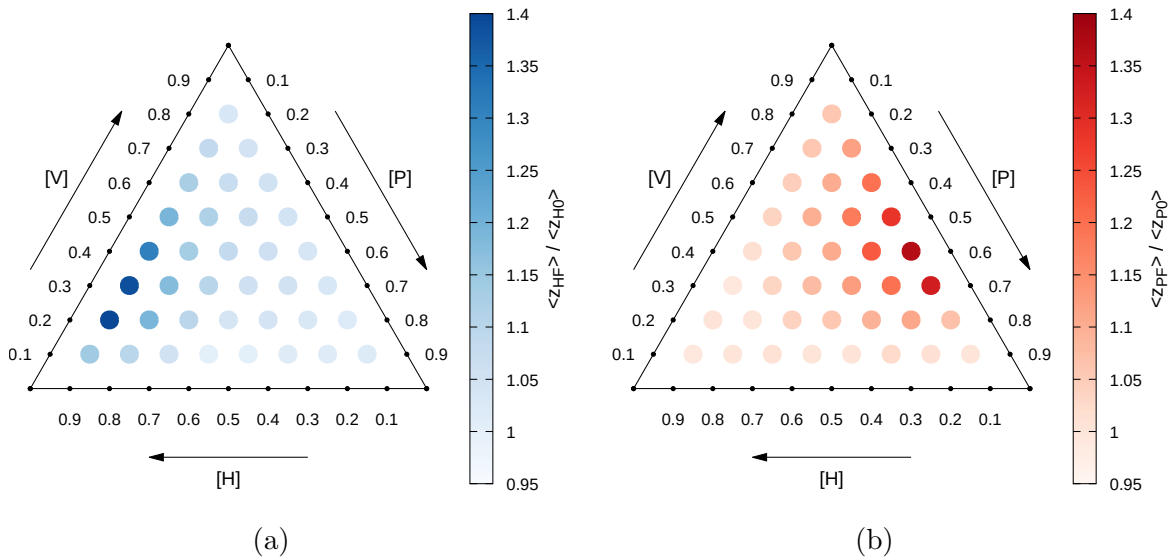


Figura 4: Triângulos de concentrações para a razão entre o número médio de *clusters* no estado estacionário (quadro = 1000), pelo número inicial médio (quadro = 0) de *clusters* ($\langle z_F \rangle / \langle z_0 \rangle$). Em (a) é referente a presas e em (b) predadores. A rede utilizada possui dimensões 50×50 e adotamos $\sigma = 3.0$. Os termos $[H]$, $[P]$ e $[V]$ representam concentrações de presas, predadores e “vazios” respectivamente. Nesta simulação foram utilizadas 200 amostras.

De acordo com as figuras 4a e 4b, os pontos mais claros revelaram que o número médio de *clusters* permaneceu aproximadamente constante, enquanto os pontos mais escuros indicaram um aumento mais expressivo. Em outras palavras, como $(\langle z_F \rangle / \langle z_0 \rangle) \geq 1$, em média, para presas e predadores, nenhuma destas 36 situações culminou no surgimento de grandes *clusters* por efeito da dinâmica de perseguição e fuga, inclusive favoreceu a dispersão de agentes em alguns casos (pontos escuros). Portanto, há outro motivo que justifica a formação de grandes rebanhos previstos nas figuras 2b e 3b.

Ao observarmos a figura 2, na situação em que $([H], [P], [V]) = (0.8, 0.1, 0.1)$, houve uma grande quantidade de presas inseridas na rede, e por isto formaram-se várias conexões entre elas, produzindo grandes *clusters*. Analogamente, o mesmo efeito ocorre com os predadores observados na figura 3 para $([H], [P], [V]) = (0.1, 0.8, 0.1)$. As figuras 5 e 6 a seguir ilustram estes cenários.

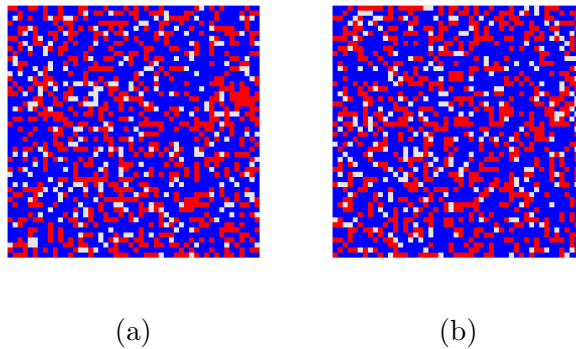


Figura 5: Rede 50×50 no quadro = 0 em (a) e no quadro = 1000 (estado estacionário) em (b), para concentrações $[H] = 60\%$ e $[P] = 30\%$, para $\sigma = 3.00$, utilizando uma única amostra. Os pontos em azuis e vermelhos representam presas e predadores respectivamente.

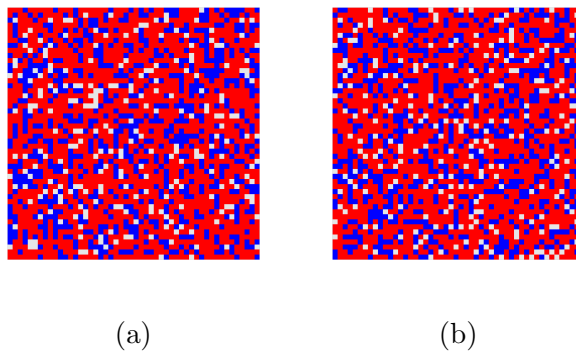


Figura 6: Rede 50×50 no quadro = 0 em (a) e no quadro = 1000 (estado estacionário) em (b), para concentrações $[H] = 30\%$ e $[P] = 60\%$, Para $\sigma = 3.00$, utilizando uma única amostra. Os pontos em azuis e vermelhos representam presas e predadores respectivamente.

Ao reduzirmos as concentrações $[H]$ e $[P]$ para valores mais moderados e assim facilitar a locomoção de agentes, observamos que tanto para presas e predadores, a rede permaneceu ocupada por vários *clusters* pequenos como visto nas figuras 7 e 8 a seguir.

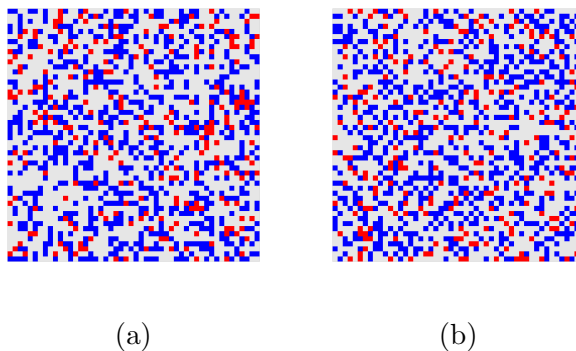


Figura 7: Rede 50×50 no quadro = 0 em (a) e no quadro = 1000 (estado estacionário) em (b), para concentrações $[H] = 30\%$ e $[P] = 10\%$, para $\sigma = 3.00$, utilizando uma única amostra. Os pontos em azuis e vermelhos representam presas e predadores respectivamente.

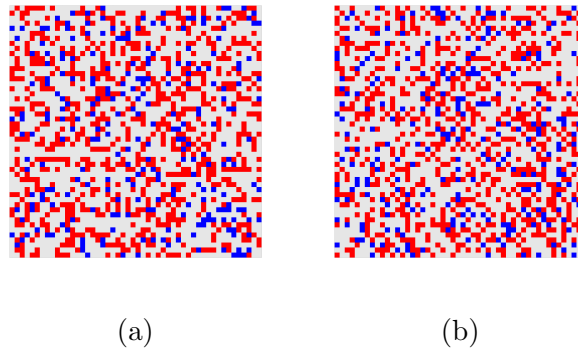


Figura 8: Rede 50×50 no quadro = 0 em (a) e no quadro = 1000 (estado estacionário) em (b), para concentrações $[H] = 10\%$ e $[P] = 30\%$, para $\sigma = 3.00$, utilizando uma única amostra. Os pontos em azuis e vermelhos representam presas e predadores respectivamente.

Ao diminuirmos a menor das concentrações de 10% para 1%, notamos que há tendência a formar *clusters* como visto nas figuras 9 e 10 na sequência. Isto sugere que entre estes dois há um valor que satura a rede dificultando a criação de grandes rebanhos.

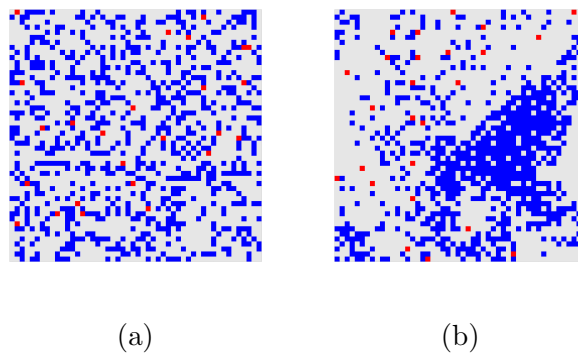


Figura 9: Rede 50×50 no quadro = 0 em (a) e no quadro = 1000 (estado estacionário) em (b), para concentrações $[H] = 30\%$ e $[P] = 1\%$, para $\sigma = 3.00$, utilizando uma única amostra. Os pontos em azuis e vermelhos representam presas e predadores respectivamente.

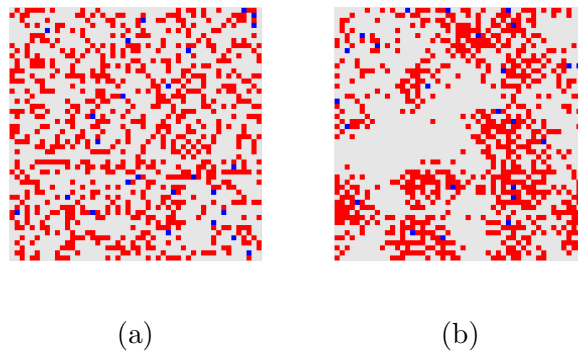


Figura 10: Rede 50×50 no quadro = 0 em (a) e no quadro = 1000 (estado estacionário) em (b), para concentrações $[H] = 1\%$ e $[P] = 30\%$, para $\sigma = 3.00$, utilizando uma única amostra. Os pontos em azuis e vermelhos representam presas e predadores respectivamente.

Apesar das figuras 4a e 4b nos fornecer uma noção geral do efeito de várias

configurações de concentrações de agentes, elas podem dificultar a distinção de valores em algumas regiões. Por isto, faremos uma nova inspeção para alguns casos particulares. De acordo com a figura 11 a seguir, ao fixarmos $[H] = 30\%$ observamos que ao longo do tempo, a medida que aumentamos algumas concentrações de predadores, em média nota-se uma diminuição no número *clusters* de presas formados. O mesmo comportamento é esperado quando analisamos o número médio relativo *clusters* de predadores ao longo do tempo (ver figura 12).

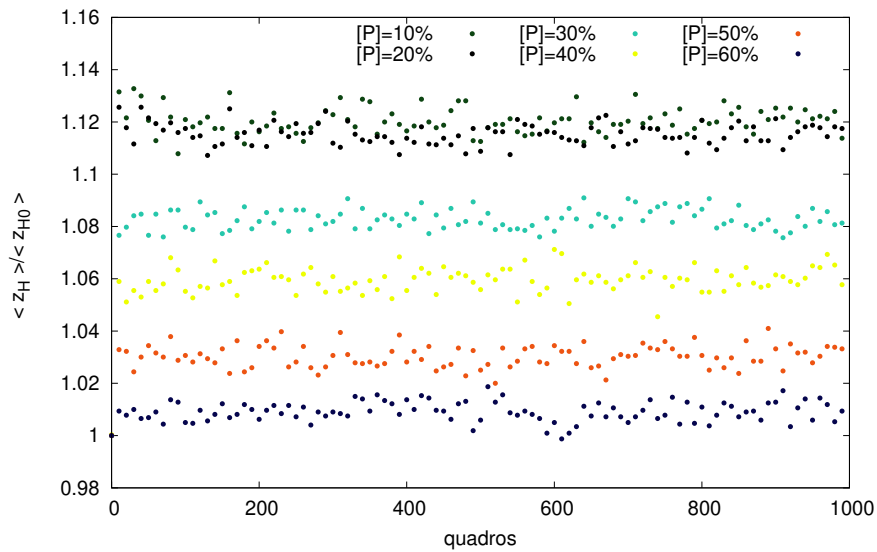


Figura 11: Número médio de *clusters* relativo de presas ao longo do tempo, para diferentes concentrações de predadores. Nesta simulação foram utilizadas 200 amostras, para $\sigma = 3.00$ e mantivemos a concentração de presas $[H] = 30\%$, para uma rede de tamanho 50×50 .

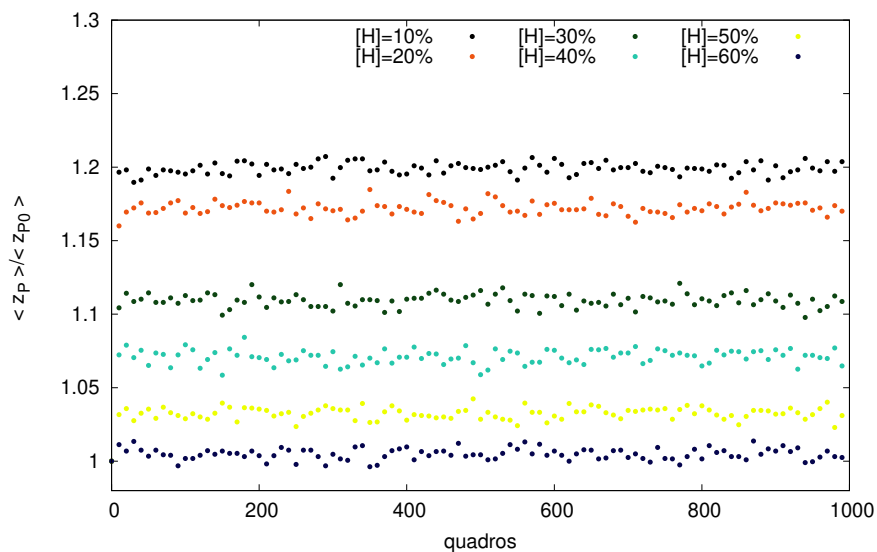


Figura 12: Número médio de *clusters* relativos de predadores ao longo do tempo, para diferentes concentrações de presas. Nesta simulação foram utilizadas 200 amostras, para $\sigma = 3.00$ e mantivemos a concentração de predadores $[P] = 30\%$, para uma rede de tamanho 50×50 .

Para este mesmo conjunto de concentrações, ao considerarmos as médias de *clusters* relativo de presas no estado estacionário, observamos na figura 13 que em geral, o aumento na concentração de predadores implica na formação de menos *clusters* de presas. O mesmo resultado é observado se fizermos a mesma análise do ponto de vista dos predadores (ver figura 14).

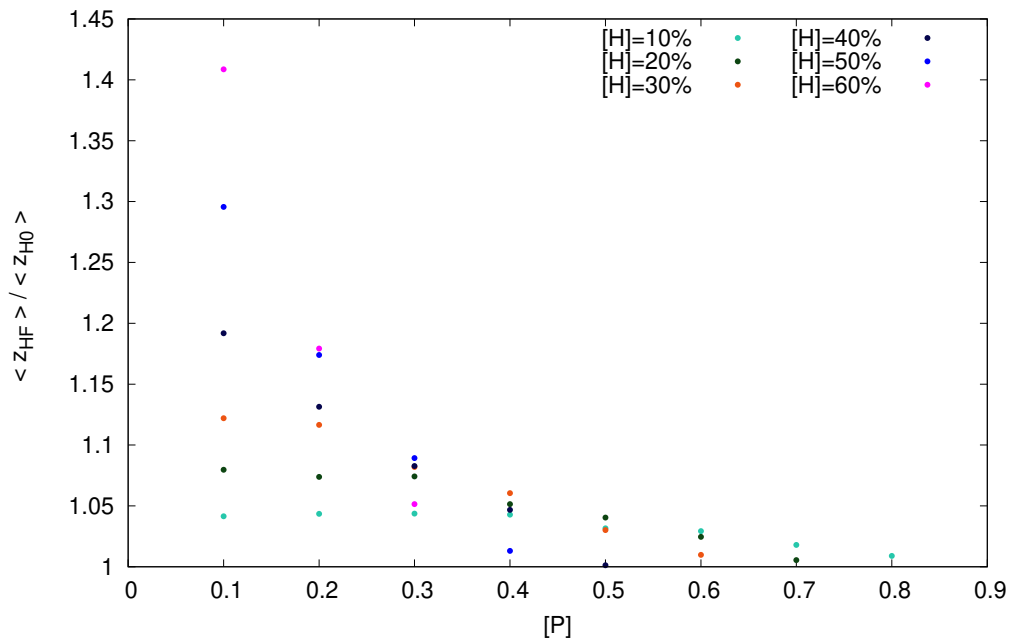


Figura 13: Número médio de *clusters* de presas em função de $[H]$ para uma rede 50×50 no estado estacionário (quadro = 1000). Nesta simulação foram utilizadas 200 amostras.

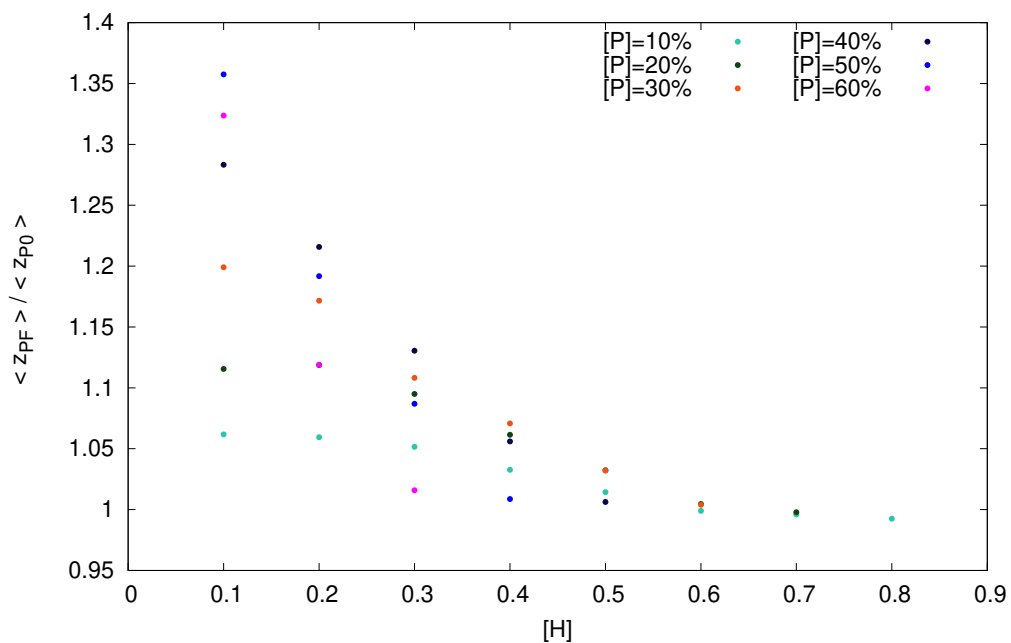


Figura 14: Número médio de *clusters* de predadores em função de $[P]$ para uma rede 50×50 no estado estacionário (quadro = 1000). Nesta simulação foram utilizadas 200 amostras.

A última análise que iremos apresentar é referente ao desvio padrão da gaussiana. Inicialmente calculamos médias de *clusters* de presas ao longo do tempo para vários valores de σ . Os resultado está presente na figura 15 a seguir e este mesmo estudo já foi apresentado no artigo, porém para uma evolução temporal curta. Ou seja, de antemão já sabemos o porquê de $\sigma \approx 0$ e $\sigma = 10$ não formar *clusters* grandes de presas.

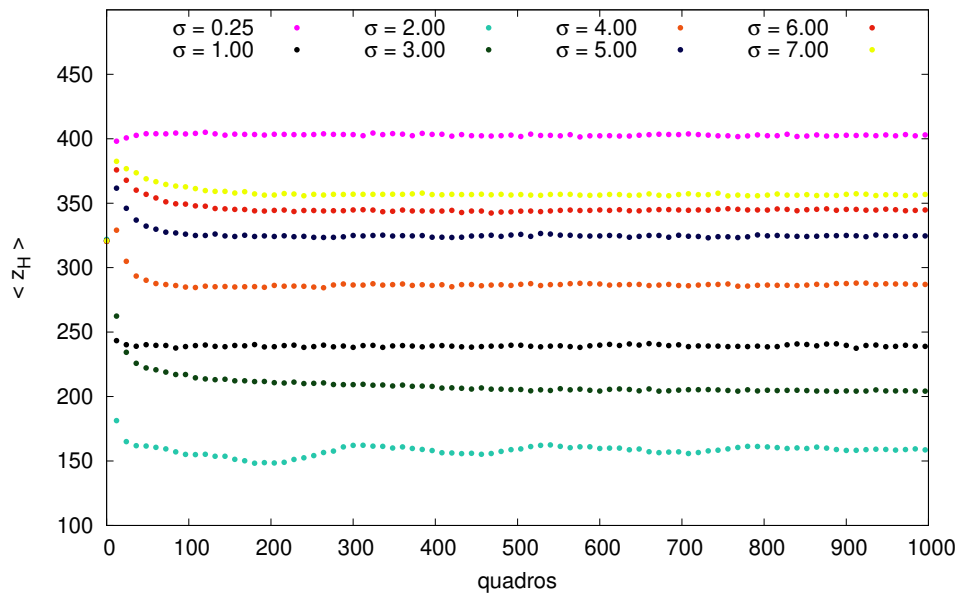


Figura 15: Número médio de *clusters* de presas ao longo do tempo para diferentes valores de σ , em uma rede 50×50 , para $[P] = 1\%$ e $[H] = 30\%$. Nesta simulação foram utilizadas 200 amostras.

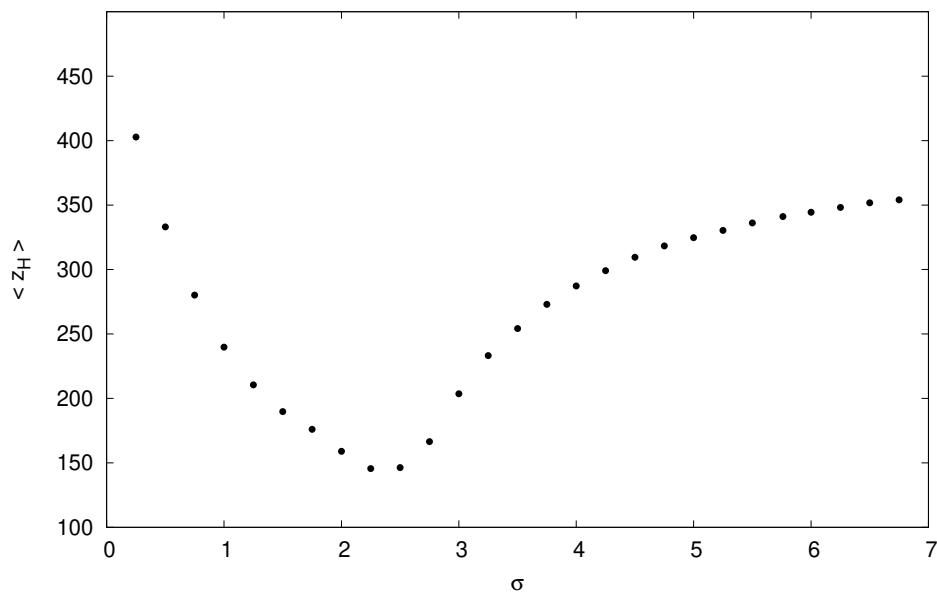


Figura 16: Número médio de *clusters* de presas em função do σ para uma rede 50×50 no estado estacionário (quadro = 1000), para $[P] = 1\%$ e $[H] = 30\%$. Nesta simulação foram utilizadas 200 amostras.

Em algum ponto entre estes dois extremos, sabemos que há uma mudança de comportamento que leva a formação rebanhos de presas. Sendo assim, acrescentamos um novo gráfico (ver figura 16), que avalia no estado estacionário, a média de *clusters* de presa em função de σ e concluímos que no intervalo $\sigma \in [2, 3]$, o número de *clusters* atinge o valor mínimo. Ou seja, é a situação que produz em média os maiores rebanhos.

Conclusão

Quanto ao artigo envolvendo o modelo de Schrödinger, reforçamos que os pontos que esboçam o átomo de hidrogênio, representam uma área “abaixo” função densidade de probabilidade (versão independente do tempo) e que os locais mais concentrados são regiões com maior probabilidade de se encontrar o elétron. Sendo assim, o método de Neumann apenas converte uma distribuições uniformes nesta mesma função.

Em relação ao MBAs desenvolvido, mostramos que simples regras de interações são capazes de formar grupos de presas e predadores. Este trabalho revelou que tanto o desvio padrão da gaussiana σ , quanto as diferentes concentrações das duas classes de agentes, foram fatores decisivos para formação de grupos de diversos tamanhos. Inclusive enfatizamos que, apesar dos triângulos de concentrações médias com valores absolutos, indicarem a presença de grandes *clusters* em alguns casos, a “formação” destes ocorrem especialmente porque a rede estava saturada. Os grandes *clusters* que advém da dinâmica de perseguição entre presas e predadores emergem quando há uma concentração mínima de um tipo de agente, e a rede é “dopada” minimamente com (aproximadamente 1%) de agentes do segundo tipo.

Verificamos também que é possível construir MBAs como este, em que estados de células atuam como campo escalar, guiando agentes que transitam na rede. Outra constatação realizada, é que neste modelo, a não ser pelo fato de indivíduos não poderem ocupar simultaneamente a mesma célula, não há interação entre agentes iguais. Isto é, a presença do agente de uma classe praticamente influencia apenas o comportamento dos demais da outra classe, e a intensidade deste efeito transmitido, varia com a distância entre eles.

Referências

- [1] GOTO, M.; AQUINO, V. M. de. Uma aplicação ingênua do método de Monte Carlo: visualização de orbitais atômicos. *Semina: Ciências Exatas e Tecnológicas*, Universidade Estadual de Londrina, v. 13, n. 4, p. 255, dec 1992. ISSN 1676-5451.
- [2] MOBLEY, R. K. Fluid power dynamics. In: _____. [S.l.]: Butterworth-Heinemann, 1999. (Plant Engineering Maintenance Series), cap. 2. ISBN 9780750671743,0750671742.
- [3] ADLER, J. Bootstrap percolation. v. 171, p. 453–470, 1991.
- [4] GUEGUEN, J. D. Y. Transport properties of rocks from statistics and percolation. *Mathematical Geology*, v. 21, n. 1, p. 1–13, 1989. ISSN 1573-8868.
- [5] DUNN, J. K. S. W. L. Introduction. In: _____. *Exploring Monte Carlo Methods*. [S.l.: s.n.], 2012. p. 7–20. ISBN 9780444515759.
- [6] DUNN, J. K. S. W. L. The basis of monte carlo. In: _____. *Exploring Monte Carlo Methods*. [S.l.: s.n.], 2012. p. 20–46. ISBN 9780444515759.
- [7] DEVORE, J. L. Statistical intervals based on a single sample. In: _____. *Probability and Statistics for Engineering and the Sciences*. 9. ed. [S.l.]: Cengage Learning, 2015. cap. 7. ISBN 1305251806, 9781305251809.
- [8] DEVORE, J. L. Appendix tables. In: _____. *Probability and Statistics for Engineering and the Sciences*. 9. ed. [S.l.]: Cengage Learning, 2015. cap. A-1. ISBN 1305251806, 9781305251809.
- [9] COUZIN, I. D. et al. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, v. 218, n. 1, p. 1 – 11, 2002. ISSN 0022-5193. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0022519302930651>>.
- [10] COUZIN, I. D.; KRAUSE, J. Self-organization and collective behavior in vertebrates. *Advances in the Study of Behavior*, v. 32, p. 1–75, 2003. ISSN 0065-3454.

Anexo 1: Programa referente a lei dos grandes números (tabela 1).

```

1 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //FileName: lawoflargenumbers.cpp //
3 //FileType: C++ Source file //
4 //Author : Pedro Henrique Fernandes Lobo //
5 //Created On : 7/04/2019 09:56:39 AM //
6 //Last Modified On : 8/04/2019 11:53:23 AM //
7 //Copyrights (c) : Pedro Henrique Fernandes Lobo, //
8 // Distributed under the MIT software license, see //
9 // http://www.opensource.org/licenses/mit-license.php //
10 //Warning: You might want to compile in this way "g++ -std=c++11 //
11 // lawoflargenumbers.cpp -o lawoflargenumbers.out" //
12 //Description : This program evaluates the integral  $\int_{-1}^1$  //
13 //  $\frac{1}{\sqrt{1-x^2}}dx$  with the Monte Carlo //
14 // Integration technique and compares the results with its //
15 // analytical solution  $\pi$ . //
16 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
17
18 #include <iostream>
19 #include <math.h>
20 #include <iomanip>
21 #include <random>
22 #include <limits>
23
24 int requestTheSampleSize();
25 double getTheOpenedLowerBoundOfDistribution(const double lowerBound);
26 double evaluateZbar(const int sampleSize,
27 std::uniform_real_distribution<double> &uniformDist,
28 std::mt19937_64 &generator);
29
30 int main(){
31
32     const double a = -1.0; // lower limit of integral
33     const double b = 1.0; // upper limit of integral
34
35     std::random_device rd;
36     std::mt19937_64 generator(rd());
37     double openedLowerBound = getTheOpenedLowerBoundOfDistribution(a);
38     std::uniform_real_distribution<double> uniformDist(openedLowerBound,b);
39     int sampleSize = requestTheSampleSize();
40     double zBar = evaluateZbar(sampleSize, uniformDist, generator);
41     double rn = (b - a) * zBar;
42     double relativeError = std::abs(1.0 - rn / MPI);
43
44     std::cout << "Sample size [N]: " << sampleSize << std::setprecision(5)
45             << std::fixed << ", R_N: " << rn << ", Relative error: "
46             << relativeError << std::endl;
47
48     return 0;
49 }
50
51 int requestTheSampleSize(){
52
53     float inputVar = 0.0;
54     bool inputFail = true;
55
56     while(inputFail){

```

```

57     std::cout << "Enter the sample size [N]: ";
58     std::cin >> inputVar;
59     inputFail = std::cin.fail();
60     if(inputVar < 2 || fmod(inputVar, 1) != 0.0){
61         std::cout << "Error: [N] must be an integer"
62             << " greater than 1.\n";
63         inputFail = true;
64     }
65     std::cin.clear();
66     std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
67         '\n');
68 }
69
70 int sampleSize = (int)inputVar;
71 return sampleSize;
72 }
73
74 double getTheOpenedLowerBoundOfDistribution(const double lowerBound){
75
76     /* The lower bound is shifted by adding the lowest double,
77     because the std::uniform_real_distribution bound is closed
78     at this point. See https://en.cppreference.com/w/cpp/numeric/
79     random/uniform\_real\_distribution for more details. */
80
81     const double lowestDouble = std::numeric_limits<double>::min();
82     return lowerBound + lowestDouble;
83 }
84
85 double evaluateZbar(const int sampleSize,
86     std::uniform_real_distribution<double> &uniformDist,
87     std::mt19937_64 &generator){
88
89     // zBar is:  $\bar{z} = \frac{1}{N} \sum_{i=1}^N \frac{1}{\sqrt{1-x_i^2}}$ 
90     //
91
92     double xi;
93     double sumZ_xi = 0.0;
94
95     for(int i = 0; i < sampleSize; i++){
96         xi = uniformDist(generator);
97         sumZ_xi += 1.0 / sqrt(1.0 - pow(xi, 2.0));
98     }
99
100     return sumZ_xi / sampleSize;
101 }

```

Anexo 2: Programa referente ao teorema do limite central (tabela 2).

```

1 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2 //FileName: limitcentraltheorem.cpp //
3 //FileType: C++ Source file //
4 //Author : Pedro Henrique Fernandes Lobo //
5 //Created On : 12/04/2019 03:06:21 PM //
6 //Last Modified On : 17/04/2019 11:01:29 PM //
7 //Copyrights (c) : Pedro Henrique Fernandes Lobo, //
8 // // Distributed under the MIT software license, see //
9 // // http://www.opensource.org/licenses/mit-license.php //
10 //Warning: You might want to compile in this way "g++ -std=c++11 //
11 // // limitcentraltheorem.cpp -o limitcentraltheorem.out sample.h //
12 // // sample.cpp". Also this program requires the Boost C++ //
13 // // library. See https://www.boost.org/ for more details. //
14 //Description : This program estimates the integral  $\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx$  //
15 // // with the Monte Carlo //
16 // // Integration technique and provides confidence //
17 // // coefficient and confidence interval. //
18 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////
19
20 #include <iostream>
21 #include <math.h>
22 #include <iomanip>
23 #include <random>
24 #include <limits>
25 #include "sample.h"
26
27 int getAmountOfDecimalDigits(double value);
28 int requestTheSampleSize();
29 double requestTheLambda();
30 double getTheOpenedLowerBoundOfDistribution(const double lowerBound);
31 double evaluateZ_xi(std::uniform_real_distribution<double> &uniformDist,
32 // std::mt19937_64 &generator);
33
34 int main(){
35
36 // const double a = -1.0; // lower limit of integral
37 // const double b = 1.0; // upper limit of integral
38
39 int sampleSize = requestTheSampleSize();
40 float lambda = requestTheLambda();
41
42 std::random_device rd;
43 std::mt19937_64 generator(rd());
44 double openedLowerBound = getTheOpenedLowerBoundOfDistribution(a);
45 std::uniform_real_distribution<double> uniformDist(openedLowerBound, b);
46
47 Sample s;
48 s.setLambda(lambda);
49
50 for(int i = 0; i < sampleSize; i++){
51 // s.addVariable(evaluateZ_xi(uniformDist, generator));
52 }
53
54 double marginOfError = (b - a) * s.getMarginOfError();
55 double rn = (b - a) * s.getSampleMean();
56

```



```
57     std::cout << std::fixed
58             << std::setprecision(getAmountOfDecimalDigits(marginOfError))
59             << "confidence interval: " << rn << " +/- " << marginOfError
60             << "; confidence coefficient: " << std::setprecision(3)
61             << s.getConfidenceCoefficient()
62             << "; sample size: " << sampleSize
63             << "; lambda: " << lambda << std::endl;
64
65     return 0;
66 }
67
68 int getAmountOfDecimalDigits(double value){
69
70     int decimalDigitsAmount = 0;
71     while(value < 1.0){
72         value *= 10;
73         decimalDigitsAmount++;
74     }
75
76     return decimalDigitsAmount;
77 }
78
79 int requestTheSampleSize(){
80
81     float inputVar = 0.0;
82     bool inputFail = true;
83
84     while(inputFail){
85         std::cout << "Enter the sample size [N]: ";
86         std::cin >> inputVar;
87         inputFail = std::cin.fail();
88         if(inputVar < 2 || fmod(inputVar, 1) != 0.0){
89             std::cout << "Error: [N] must be an integer"
90                     << " greater than 1.\n";
91             inputFail = true;
92         }
93         std::cin.clear();
94         std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
95                        '\n');
96     }
97
98     int sampleSize = (int)inputVar;
99     return sampleSize;
100 }
101
102 double requestTheLambda(){
103
104     double lambda = -1.0;
105     bool inputFail = true;
106
107     while(inputFail){
108         std::cout << "Enter the lambda value: ";
109         std::cin >> lambda;
110         inputFail = std::cin.fail();
111         if(lambda < 0.0 || inputFail){
112             std::cout << "Error: lambda must be a real "
113                     << "greater or equal than 0.0.\n";
114             inputFail = true;
115         }
116     }
117
118     return lambda;
119 }
```

```

115     }
116     std::cin.clear();
117     std::cin.ignore(std::numeric_limits<std::streamsize>::max(),
118                   '\n');
119 }
120
121 return lambda;
122 }
123
124 double getTheOpenedLowerBoundOfDistribution(const double lowerBound){
125
126     /* The lower bound is shifted by adding the lowest double,
127        because the std::uniform_real_distribution bound is closed
128        at this point. See https://en.cppreference.com/w/cpp/numeric/
129        random/uniform_real_distribution for more details. */
130
131     const double lowestDouble = std::numeric_limits<double>::min();
132
133     return lowerBound + lowestDouble;
134 }
135
136 double evaluateZ_xi(std::uniform_real_distribution<double> &uniformDist,
137                   std::mt19937_64 &generator){
138
139     // z_xi is: $z(x_i) = \frac{1}{\sqrt{1-{x_i}^2}}$
140     double xi;
141     xi = uniformDist(generator);
142     return 1.0 / sqrt(1.0 - pow(xi, 2.0));
143 }

```

```

1  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  //FileName: sample.h                                                                    //
3  //FileType: C++ Source file                                                            //
4  //Author : Pedro Henrique Fernandes Lobo                                             //
5  //Created On : 13/04/2019 04:06:32 AM                                                //
6  //Last Modified On : 13/04/2019 04:59:00 AM                                         //
7  //Copyrights (c) : Pedro Henrique Fernandes Lobo,                                  //
8  //                Distributed under the MIT software license, see                   //
9  //                http://www.opensource.org/licenses/mit-license.php                 //
10 //Warning: This class requires the Boost C++ library.                               //
11 //        See https://www.boost.org/ for more details.                               //
12 //Description : This program evaluates the mean and variance of normal              //
13 //               distributions. The boost library can provide the mean                //
14 //               and standard deviation, but we calculate separately for             //
15 //               educational purposes.                                               //
16 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
17
18 #ifndef SAMPLE_H
19 #define SAMPLE_H
20
21 #include <boost/math/distributions/normal.hpp> // for normal_distribution
22
23 class Sample{
24 public:
25     Sample();
26     ~Sample();
27     double getSampleMean();
28     double getMarginOfError();

```

```

29     double getConfidenceCoefficient ();
30     void addVariable (double x);
31     void setLambda (double zScore);
32     double getSampleVariance ();
33
34 private:
35     int sampleSize;
36     double sampleMean;
37     double sumTermInVariance; //  $\sum_{i=1}^n (x - \bar{x})^2$ 
38     double lambda;
39     double sampleVariance;
40     void updateSampleMean (double x);
41     void updateSampleVariance (double x);
42
43 };
44 #endif

```

```

1  //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
2  //FileName: sample.cpp                                                    //
3  //FileType: C++ Source file                                              //
4  //Author : Pedro Henrique Fernandes Lobo                                 //
5  //Created On : 8/04/2019 01:08:21 PM                                     //
6  //Last Modified On : 09/04/2019 00:01:49 PM                              //
7  //Copyrights (c) : Pedro Henrique Fernandes Lobo,                       //
8  //      Distributed under the MIT software license, see                  //
9  //      http://www.opensource.org/licenses/mit-license.php //
10 //Description : Implementation of sample.h methods                         //
11 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
12
13 #include "sample.h"
14
15 Sample::Sample () {
16     sampleSize = 0;
17     sampleMean = 0.0;
18     sumTermInVariance = 0.0;
19     lambda = 0.0;
20     sampleVariance = 0.0;
21 }
22
23
24 Sample::~Sample () {
25 }
26
27
28 double Sample::getSampleMean () {
29     return sampleMean;
30 }
31
32
33 double Sample::getMarginOfError () {
34     return lambda * sqrt (sampleVariance / sampleSize);
35 }
36
37
38 double Sample::getConfidenceCoefficient () {
39     boost::math::normal distribution (0.0, 1.0);
40

```

```
41     return cdf(distribution, lambda) - cdf(distribution, - lambda);
42 }
43
44 void Sample::addVariable(double x){
45     sampleSize++;
46     updateSampleMean(x);
47     updateSampleVariance(x);
48 }
49
50 void Sample::updateSampleVariance(double x){
51     // Welford's method for computing sample variance.
52
53     double previousSampleMean;
54
55     if(sampleSize > 1){
56         double dummyVar = sampleSize * sampleMean - x;
57         previousSampleMean = dummyVar / (sampleSize - 1);
58     }
59     else{
60         previousSampleMean = sampleMean;
61     }
62
63     double delta = x - previousSampleMean;
64     double delta2 = x - sampleMean;
65     sumTermInVariance += delta * delta2;
66     sampleVariance = sumTermInVariance / (sampleSize - 1);
67 }
68
69 void Sample::updateSampleMean(double x){
70
71     sampleMean += (x - sampleMean) / (1.0 * sampleSize);
72 }
73
74 void Sample::setLambda(double lambda){
75
76     this -> lambda = lambda;
77 }
78
79 double Sample::getSampleVariance(){
80
81     return sampleVariance;
82 }
83
84 }
```