

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-graduação em Física

Ana Luiza Novaes Dias

**TEORIA DE CAMPO MÉDIO PARA UM
MODELO DE VICSEK EM REDE**

Belo Horizonte
2022

Ana Luiza Novaes Dias

TEORIA DE CAMPO MÉDIO PARA UM MODELO DE VICSEK EM REDE

Dissertação apresentada ao Programa de Pós-Graduação em Física do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para obtenção do título de Mestre em Ciências.

Orientador: Ronald Dickman

Belo Horizonte
2022

Dados Internacionais de Catalogação na Publicação (CIP)

D541t Dias, Ana Luiza Novaes.
Teoria de campo médio para um modelo de Vicsek em rede / Ana Luiza Novaes Dias. – 2022.
61 f. : il.

Orientador: Ronald Dickman.
Dissertação (mestrado) – Universidade Federal de Minas Gerais,
Departamento de Física.
Bibliografia: f. 41-42.

1. Modelos matemáticos. 2. Teoria de campos. 3. Transição de fase. I.
Título. II. Dickman, Ronald. III. Universidade Federal de Minas Gerais,
Departamento de Física.

CDU – 539.2 (043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

ATA DE DEFESA DE DISSERTAÇÃO

ATA DA SESSÃO DE ARGUIÇÃO DA 679ª DISSERTAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA, DEFENDIDA POR ANA LUIZA NOVAES DIAS, orientada pelo professor Ronald Dickman, para obtenção do grau de **MESTRE EM FÍSICA**. Às 14 horas de doze de julho de 2022, por videoconferência, reuniu-se a Comissão Examinadora, composta pelos professores **Ronald Dickman** (Orientador - Departamento de Física/UFMG), **Lucas Lages Wardil** (Departamento de Física/UFMG) e **Leonardo Gregory Brunnet** (Instituto de Física/UFRGS) para dar cumprimento ao Artigo 37 do Regimento Geral da UFMG, submetendo a bacharel **ANA LUIZA NOVAES DIAS** à arguição de seu trabalho de dissertação, que recebeu o título de “Teoria de Campo Médio para um Modelo de Vicsek em Rede”. A candidata fez uma exposição oral de seu trabalho durante aproximadamente 50 minutos. Após esta, os membros da comissão prosseguiram com a sua arguição e apresentaram seus pareceres individuais sobre o trabalho, concluindo pela aprovação da candidata.

Belo Horizonte, 12 de julho de 2022.

Prof. Ronald Dickman
Orientador da estudante
Departamento de Física/UFMG

Prof. Lucas Lages Wardil
Departamento de Física/UFMG

Prof. Leonardo Gregory Brunnet
Instituto de Física/UFRGS

Candidata: Ana Luiza Novaes Dias



Documento assinado eletronicamente por **Lucas Lages Wardil, Professor do Magistério Superior**, em 13/07/2022, às 10:36, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Leonardo Gregory Brunnet, Usuário Externo**, em 13/07/2022, às 11:38, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Ana Luiza Novaes Dias, Usuária Externa**, em 13/07/2022, às 14:51, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Ronald Dickman, Professor do Magistério Superior**, em 14/07/2022, às 08:43, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1598152** e o código CRC **8B282BCB**.

Para Olivia,

Agradecimentos

Gostaria de agradecer em primeiro lugar ao meu orientador Ronald Dickman pela dedicação em me guiar ao longo do desenvolvimentos desse trabalho. Agradeço também à minha família e amigos pelo apoio incondicional, em especial aos meus avós Natividade, Araci (*in memoriam*) e João (*in memoriam*) que desempenharam papel fundamental na minha formação. Por fim, agradeço ao CNPq pelo apoio financeiro que permitiu que eu me dedicasse exclusivamente à pós-graduação e pelo suporte essencial à ciência no Brasil.

Resumo

Recebe o nome de matéria ativa um grupo de indivíduos que consomem energia continuamente para se movimentar no meio em que se encontram. Eles são encontrados na natureza em diferentes escalas, por exemplo, bandos de pássaros e colônias de bactérias, e fora do campo biológico uma coleção de motores vibrantes também se classificam como tal. Apesar das claras diferenças entre esses sistemas, eles têm em comum o aparecimento de movimentos coletivos e entender como eles surgem é o objetivo principal do estudo de matéria ativa. O modelo de Vicsek foi o primeiro a obter resultados condizentes com o que se observa em grupos de organismos autopropulsados, mostrando que o surgimento de comportamentos coletivos pode ser tratado como uma transição de fase dependente da densidade e do ruído e cujo parâmetro de ordem está associado à velocidade dos indivíduos. A partir dele novos modelos com diversas características foram criados e mostraram que existe uma rica variedade de comportamentos que podem surgir na dinâmica de indivíduos autopropelidos. Nesse trabalho foi estudado o comportamento de um modelo de matéria ativa em uma rede triangular sujeita a uma interação de alinhamento e de volume excluído que é delimitada por uma aproximação de campo médio. Ao analisar a evolução das probabilidades de ocupação dos sítios que constituem a rede vê-se que o modelo passa por uma transição de fase descontínua na qual uma direção de velocidade domina a rede.

Palavras-chave: Matéria Ativa, Modelo de Vicsek, Teoria de Campo Médio, Transição de Fase.

Abstract

Active matter is a group of individuals that continuously consume energy to move in the environment. They are found in nature at different scales, for example flocks of birds and colonies of bacteria, and outside the biological field, systems such as a collection of vibrating motors. Despite the clear differences between these systems, they have in common the emergence of collective motion. Understanding how collective motion arises is the main goal of the study of active matter. The Vicsek model was the first to obtain results consistent with observations of groups of self-propelled organisms, showing that the emergence of collective behavior can be treated as a phase transition dependent on density and noise, and whose order parameter is associated with the velocity of particles. From it, new models with different features were created showing a rich variety of behaviors that can arise in the dynamics of interacting self-propelled individuals. In this work I study the behavior of an active matter model on a triangular lattice subject to an alignment interaction, and to excluded volume, using a mean-field approximation. Analyzing the evolution of the probabilities for the occupation of sites in the lattice, we find that the model goes through a discontinuous phase transition in which a particular direction of motion dominates.

Keywords: Active Matter, Vicsek Model, Mean Field Theory, Phase Transition.

Sumário

1	INTRODUÇÃO	11
2	MODELO DE VICSEK	13
2.1	Introdução	13
2.2	O Modelo de Vicsek	13
2.3	Matéria Ativa em Espaços Discretos	16
3	TEORIA DE CAMPO MÉDIO PARA O MODELO DE VICSEK EM REDE	18
3.1	Introdução	18
3.2	Teoria de Campo Médio para o Modelo de Vicsek em Rede	18
4	ANÁLISE COMPUTACIONAL	21
4.1	Introdução	21
4.2	A Equação de Movimento	21
4.2.1	Integração Numérica	25
4.3	Processo de Transferência de Densidades	26
4.4	A Rede	27
5	RESULTADOS	29
5.1	Introdução	29
5.2	Alinhamento	29
5.2.1	Diferentes Condições Iniciais	32
5.2.2	Os Zeros da Equação de Movimento	33
5.3	Movimento das Partículas	36
6	CONCLUSÕES E TRABALHOS FUTUROS	40
	REFERÊNCIAS	41
	APÊNDICE A – MÉTODO DE RUNGE-KUTTA PARA RESOLUÇÃO DE EQUAÇÕES DIFERENCIAIS	43
	APÊNDICE B – MÉTODO DE RUNGE-KUTTA PARA RESOLUÇÃO DAS EQUAÇÕES DE MOVIMENTO DO MODELO	45

APÊNDICE C – CÓDIGO DO MODELO EM C	46
--	----

1 Introdução

Matéria ativa diz respeito a um grupo de organismos ou outras entidades que absorvem e dissipam energia para se propelirem. Os sistemas formados por partículas autopropelidas são sistemas fora do equilíbrio e tem como característica principal o surgimento de comportamentos coletivos devido às interações entre os indivíduos e entre os indivíduos e o meio em que se encontram.

O primeiro modelo capaz de reproduzir o comportamento de sistemas de matéria ativa foi descrito por Vicsek *et al.* em 1995 [1]. Nesse modelo vemos o surgimento espontâneo de movimentos coletivos como uma transição de fase governada por alguns parâmetros básicos do sistema. Como resultado, variações do modelo de Vicsek foram utilizados para estudar o comportamento de rebanhos de ovelhas [2], de colônias de bactérias [3], de filamentos em motores moleculares no nível celular [4] e, fora do campo biológico, o movimento de partículas Janus [5].

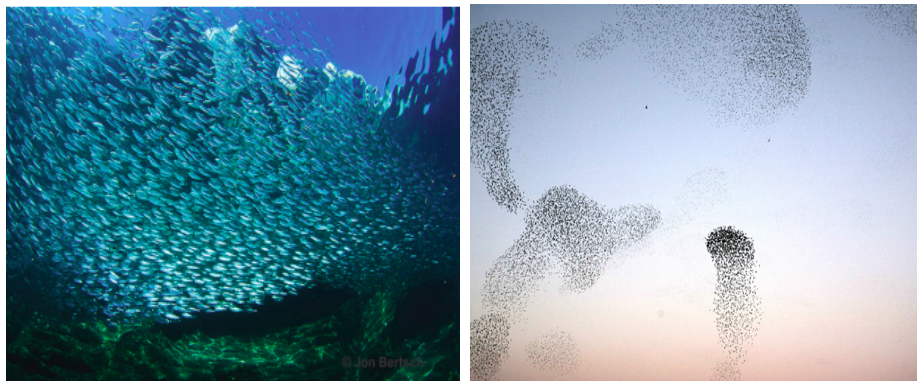


Figura 1 – Exemplos de sistemas de matéria ativa encontrados na natureza. À esquerda um cardume de sardinhas e à direita bandos de pássaros. Figuras retiradas de [6] e [7], respectivamente.

No geral, modelos para matéria ativa consideram a existência de uma interação que faz com que os indivíduos tenham a tendência de seguir a mesma direção que seus vizinhos. Dessa maneira determinar uma vizinhança é parte importante no estudo do comportamento de sistemas de partículas autopropelidas. Normalmente isso é feito delimitando uma região ao redor da partícula de interesse e considerando que qualquer indivíduo que esteja nesse espaço interage com a partícula. Outra maneira de limitar a vizinhança é através de uma interação topológica [8] na qual se determina um número fixo de vizinhos com os quais os organismos vão interagir, estudos sugerem que para o comportamento coletivo de animais essa forma de interação é a mais realista [7]. Em modelos em rede, a princípio, poderia considerar que todos os sítios da rede interagem entre si de modo a garantir que nenhuma

flutuação passe despercebida durante a dinâmica; essa definição porém é quase impossível de estudar para sistemas grandes. Assim, a construção mais sensata é uma aproximação entre primeiros vizinhos que deve no mínimo fornecer informações qualitativas sobre o comportamento.

Com o objetivo de estudar os diferentes comportamentos que podem surgir devido as diferentes configurações de sistemas de matéria ativa, nesse trabalho é proposto o estudo de um modelo de Vicsek em rede na qual se utiliza uma análise de campo médio para delimitar o alcance da interação entre partículas .

No Capítulo 2 é feita uma descrição do modelo de Vicsek com seus principais resultados a respeito do surgimento espontâneo de movimentos coletivos em um modelo simples que segue apenas uma regra de alinhamento das orientações das velocidades das partículas. É apresentado também alguns resultados para modelos de matéria ativa em espaços discretos, focando principalmente no trabalho desenvolvido por Peruani e colaboradores [9].

A descrição do modelo a ser estudado é feita no Capítulo 3, onde é apresentado as grandezas de interesse principal para seguir o comportamento do sistema na rede. E no Capítulo 4 é desenvolvido as equações necessárias para avaliar a evolução temporal das partículas e os algoritmos utilizados para realizar a análise computacional.

Os resultados obtidos são apresentados no Capítulo 5. Para o movimento de partículas na rede não foram alcançados resultados que consideramos caracterizar de fato a dinâmica do modelo e dessa maneira deixamos em aberto a questão sobre como as partículas se organizam quando há dependência espacial. Esperamos resolver o problema em trabalhos futuros que visam melhorar a análise computacional para o modelo.

Por fim, no Capítulo 6 é feita uma breve conclusão sobre os resultados do trabalho e os novos caminhos que se pode seguir a partir deles.

2 Modelo de Vicsek

2.1 Introdução

O modelo de Vicsek (MV) [1] é um modelo simples, mas capaz de descrever corretamente o movimento coletivo de grupos de organismos. Ele segue uma única regra que diz que a direção da velocidade das partículas deve se orientar com a direção média de movimento dentro de uma vizinhança de raio R_0 . Porém a presença de um ruído aditivo no sistema pode atrapalhar esse alinhamento.

Como há diferentes maneiras que matéria ativa pode aparecer na natureza, é preciso modelar sistemas que implementem diferentes características como o tipo de ruído e tipo de interação, o alcance da interação entre partículas e até mesmo o espaço no qual o sistema está contido, podendo ser contínuo ou discreto. Pela sua simplicidade, o MV se tornou um ponto de partida para a elaboração desses novos modelos.

2.2 O Modelo de Vicsek

O modelo de Vicsek trata-se de um sistema com N partículas autopropelidas que se movem com velocidade constante v_0 em um espaço contínuo de tamanho $L \times L$ e com condições periódicas de contorno. Seja $\mathbf{r}_i(t)$ a posição da partícula i e $\mathbf{v}_i = v_0(\cos \theta_i, \sin \theta_i)$ a velocidade com direção de movimento definida pelo ângulo θ_i . A cada intervalo de tempo Δt a posição da partícula no sistema é atualizada conforme

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t). \quad (2.1)$$

Um círculo de raio R_0 centrado em \mathbf{r}_i determina a vizinhança de interação da partícula i , que deve reorientar sua velocidade de modo a se alinhar com a velocidade média dentro do círculo, conforme ilustra a Figura 2. Assim $\theta_i(t)$ segue

$$\theta_i(t + \Delta t) = \text{Arg} \left[\sum_j e^{i\theta_j(t)} \right] + \eta \xi_i(t), \quad (2.2)$$

em que o somatório é sobre todas as partículas dentro da vizinhança, inclusive a partícula i , e a função Arg fornece o ângulo médio das velocidades na vizinhança. $\xi_i(t)$ é uma variável aleatória uniformemente distribuída no intervalo $[-\pi, \pi]$ que cumpre o papel de um ruído de intensidade η e tem como propriedades $\langle \xi_i(t) \rangle = 0$ e $\langle \xi_i(t) \xi_j(t') \rangle = \Gamma \delta_{i,j} \delta_{t,t'}$, em que $\Gamma = \text{var}[\xi_i(t)] = \pi^2/3$. O uso de $\delta_{t,t'}$ ao invés da função delta de Dirac $\delta(t - t')$ se dá pelo fato de o tempo t ser considerado uma variável discreta no modelo. O que o ruído faz na

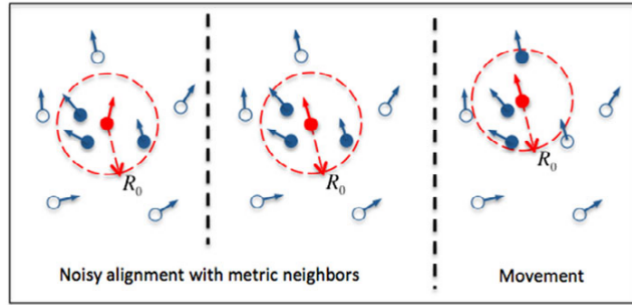


Figura 2 – Dinâmica do modelo de Vicsek: a partícula em vermelho tende a se alinhar com a direção média da velocidade dentro do círculo de raio R_0 , em seguida se move seguindo sua nova orientação e uma nova vizinhança é formada. Figura retirada de [8].

equação para θ_i é adicionar uma perturbação à direção média de movimento e por isso se classifica como um ruído aditivo.

Sem perda de generalidade, escolhe-se $\Delta t = 1$ e $R_0 = 1$. Restando apenas três parâmetros de controle para o comportamento do sistema, a densidade $\rho = N/L^2$, a amplitude do ruído η e a velocidade v_0 . O parâmetro de ordem é definido como sendo a orientação média da velocidade das N partículas que compõem o sistema:

$$\Phi(t) = \frac{1}{Nv_0} \left| \sum_{i=1}^N \mathbf{v}_i(t) \right|. \quad (2.3)$$

Caso a direção de movimento das partículas esteja distribuída aleatoriamente $\langle \Phi \rangle_t = 0$ e, se houver ordenamento, $\langle \Phi \rangle_t > 0$. Dessa forma a média temporal do parâmetro de ordem indica a ocorrência de uma quebra espontânea de simetria característica dos fenômenos de transição de fase. Tal comportamento é similar ao que ocorre na transição ferromagneto/paramagneto de modelos de spins com campo magnético externo nulo [10]. Nesse caso o parâmetro de ordem é a magnetização \mathbf{m} , que é nula na fase paramagnética, e para $T < T_c$ o sistema passa por uma magnetização espontânea na qual os spins se alinham e portanto $\mathbf{m} \neq 0$.

O estudo do comportamento das partículas no modelo de Vicsek segue da seguinte maneira: v_0 e ρ são mantidos fixos enquanto η varia, o sistema é iniciado com uma distribuição aleatória das direções das velocidades das partículas e a cada passo de tempo as posições e orientações de todas as partículas são atualizados simultaneamente conforme Eq.(2.1) e Eq.(2.2), respectivamente. Dessa maneira descobriu-se que o modelo sofre uma transição para movimentos coletivos, onde um grupo passa a seguir uma mesma direção que é escolhida aleatoriamente.

Para ruído alto as partículas não são capazes de se alinhar e o sistema se mantém desordenado. Ao diminuir η , grupos seguindo em uma mesma direção começam a surgir

caracterizando a formação de bandas; nessa situação o sistema passa a apresentar ordenamento local e $\langle \Phi \rangle_t > 0$. Por fim, para valores pequenos do ruído se alcança uma fase de ordenamento homogêneo em que uma única direção de movimento é dominante sobre todo o sistema, $\langle \Phi \rangle_t \simeq 1$.

As fases do modelo estão relacionadas com a densidade e o ruído, no sentido que quanto maior for a densidade mais fácil é manter o ordenamento. Isso ocorre porque quando a densidade é alta quase todas as vizinhanças definidas pelo círculo de raio R_0 acabam se sobrepondo e a informação sobre a direção das velocidades se dissemina mais facilmente entre as partículas. Por sua vez, se considerarmos $\rho \ll 1$ as vizinhanças podem ficar totalmente isoladas uma das outras e o ruído se sobressai à interação de alinhamento. Nesse sentido a formação de bandas ordenadas implica a existência de locais com densidades mais altas no sistema. A relação entre ρ e η fornece o diagrama de fase para MV, Fig.3-(a).

A transição de fase que ocorre no modelo foi inicialmente descrita como sendo uma transição contínua, porém estudos posteriores para sistemas maiores mostraram que a transição de fase do modelo de Vicsek é na verdade sempre descontínua [11].

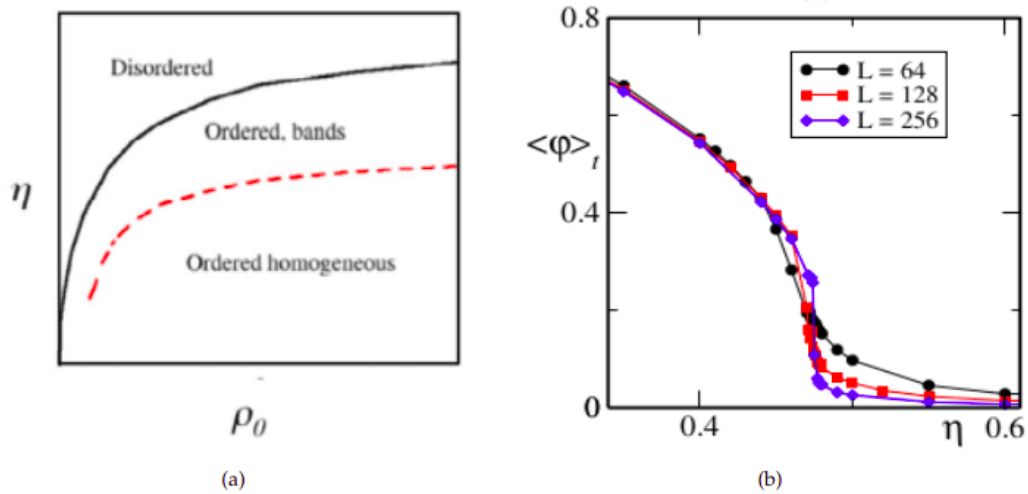


Figura 3 – (a) Diagrama de fase para o modelo de Vicsek indicando as três fases do sistema. (b) Parâmetro de ordem em função do ruído para diferentes tamanhos do sistema. Figuras retiradas de [8].

2.3 Matéria Ativa em Espaços Discretos

Seja uma rede com geometria qualquer formada por N sítios nos quais os organismos autopropulsados podem estar. Os modelos para matéria ativa dentro de uma rede apresentam duas diferenças importantes em relação aos sistemas em espaços contínuos, como é o caso do modelo de Vicsek. A primeira diferença está relacionada com as velocidades das partículas que compõem o sistema, pois a geometria do espaço impõe que as orientações sejam discretizadas para coincidir com as direções dos elos. A segunda diferença é que as partículas estão ao menos a uma distância a uma das outras, sendo a a distância entre dois sítios vizinhos.

Modelos de matéria ativa em rede são normalmente utilizados para estudar sistemas com condições de volume excluído e dão origem a variados padrões no movimento coletivo dos organismos. O trabalho de Peruani *et al.* [9] demonstra bem os diferentes arranjos que podem surgir em um sistema com essas características. No modelo estudado as partículas se movem em uma rede quadrada com condições periódicas de contorno e há apenas quatro velocidades possíveis que correspondem a movimentos para a esquerda, para a direita, para cima e para baixo. A dinâmica do modelo é governada por:

$$T_R((\mathbf{x}, \mathbf{v}) \rightarrow (\mathbf{x}, \mathbf{w})) = \exp\left(g \sum_{\mathbf{y} \in A(\mathbf{x})} \langle \mathbf{w} | \mathbf{V}(\mathbf{y}) \rangle\right), \quad (2.4)$$

$$T_M((\mathbf{x}, \mathbf{v}) \rightarrow (\mathbf{y} = \mathbf{x} + \mathbf{v}, \mathbf{v})) = \begin{cases} v_0, & \text{se } \mathbf{y} \text{ está vazio} \\ 0, & \text{se } \mathbf{y} \text{ está ocupado.} \end{cases}$$

Em que T_R é a taxa de reorientação para a direção \mathbf{w} para uma partícula na posição \mathbf{x} e com direção \mathbf{v} . O somatório é sobre os primeiros vizinhos do sítio em \mathbf{x} dentro de uma região definida por $A(\mathbf{x})$ e $\langle | \rangle$ representa o produto interno entre a velocidade na direção \mathbf{w} e a direção da velocidade das partículas nos sítios vizinhos $\mathbf{V}(\mathbf{y})$. Isso define a interação de alinhamento do modelo que, ao invés de considerar a presença de um ruído, inclui o parâmetro g na taxa de reorientação para controlar a intensidade da interação entre uma partícula e seus vizinhos. Por sua vez, T_M representa a taxa de migração de \mathbf{x} para um sítio em \mathbf{y} que está na direção de movimento da partícula. A condição de volume excluído é implementada pela exigência de que o sítio vizinho esteja vazio para que a transferência ocorra a uma taxa v_0 .

Variações do parâmetro de controle levam a quatro fases distintas. Para g baixo (Fig.4-(a)), não há ordenamento no sistema. Conforme g aumenta as partículas começam a se alinhar com seus vizinhos e dão origem as estruturas mostradas na Fig.4-(b); as partículas no meio do aglomerado estão impedidas de se mover devido a condição de volume excluído, esse padrão recebe o nome de *traffic jam*. Para sair dessa configuração é preciso que a intensidade da interação cresça de modo que o aglomerado passa a ter duas velocidades dominantes (Fig.4-(c)), para cima (em azul) e para a esquerda (em preto).

Por fim, aumentando ainda mais o parâmetro de controle uma única direção é escolhida e temos uma região de alta densidade com todas as partículas com a mesma orientação. A formação de uma banda na qual as partículas têm velocidade orientada para cima é mostrada na Fig.4-(d). Note que as partículas que compõem a banda não se movem, exceto na interface com a região desordenada. A transição de fase em que o sistema sai de uma configuração homogênea para a formação de agregados ordenados é uma transição descontínua.

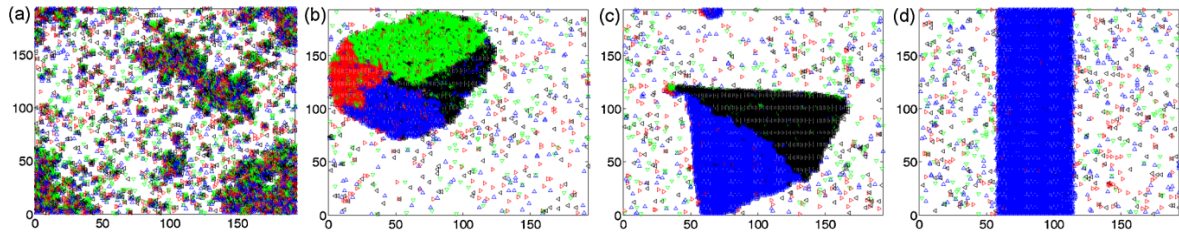


Figura 4 – Os diferentes padrões de movimento coletivo que surgem do modelo proposto por Peruani. Figura retirada de [9].

A formação de aglomerados ordenados também ocorre em um modelo em rede com uma interação de alinhamento que é intrinsecamente estocástica e há oito possíveis orientações para a velocidade [12]. É vista também em um modelo em que não se considera interações de alinhamento e a transição de fase é induzida pela motilidade das partículas na rede [6, 13].

3 Teoria de Campo Médio para o Modelo de Vicsek em Rede

3.1 Introdução

Propomos o estudo de um modelo de matéria ativa em um rede triangular sujeita a uma interação de alinhamento cujo alcance dentro da rede é fornecido por uma aproximação de campo médio. Nesse capítulo damos uma visão geral do modelo, detalhando os principais parâmetros que vão definir o comportamento do sistema.

3.2 Teoria de Campo Médio para o Modelo de Vicsek em Rede

Seja uma rede bidimensional triangular de tamanho finito. Considerando um espaçamento $a = 1$ entre os sítios e h e w as dimensões da rede, há (hxw) sítios que podem ser ocupados por partículas autopropelidas cujas velocidades são iguais em módulo, mas podem assumir apenas três orientações que se diferem por um ângulo de 120° , como mostra a Fig.5.

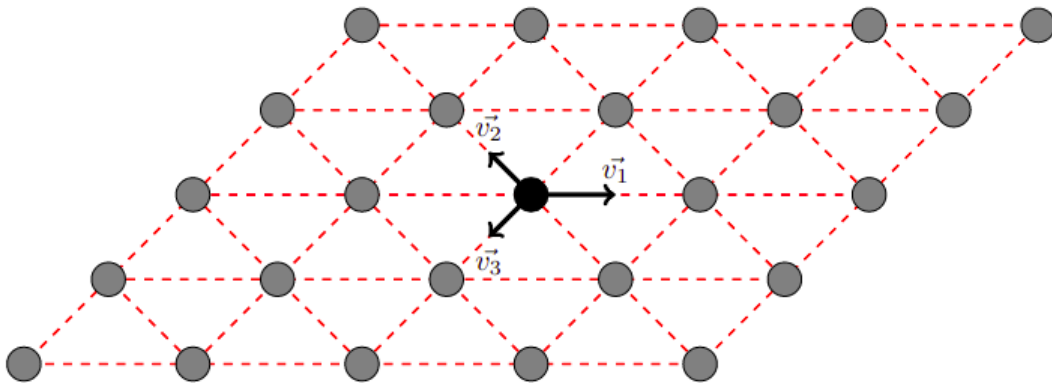


Figura 5 – Rede triangular com (hxw) sítios, sendo $h = w = 5$, que podem estar ocupados por partículas autopropelidas com as velocidades \mathbf{v}_1 , \mathbf{v}_2 ou \mathbf{v}_3 .

Consideramos a presença de N partículas nessa rede. O estado de cada uma é definido pela posição na rede e pela orientação da velocidade. Dessa maneira podemos representar a fração de partículas em cada estado como

$$f_i = \frac{N_i}{N}, \quad i = 1, 2, 3. \quad (3.1)$$

Sendo N_i o número de partículas com velocidade \mathbf{v}_i e $N = \sum_{i=1}^3 N_i$, o que leva à restrição

$$f_1 + f_2 + f_3 = 1. \quad (3.2)$$

Uma das principais diferenças desse modelo é o fato de que vamos considerar as probabilidades de ocupação dos sítios da rede. Dizemos que um sítio s tem uma probabilidade ρ_s de estar ocupado e, estando ocupado, uma probabilidade f_{is} de que a partícula tenha a orientação i . Isso define $(\rho f_i)_s$ como a probabilidade de ocupação do sítio s para o estado i . A densidade da rede, ou a probabilidade média de ocupação, é dada por

$$\rho_g = \frac{1}{A} \sum_s \rho_s, \quad (3.3)$$

sendo A a área da rede.

Como no modelo de Vicsek original, estabelecemos uma interação de alinhamento entre as partículas que podem reorientar a direção de sua velocidade para se alinhar com a direção majoritária entre seus vizinhos. Tal interação tem como efeito variar as frações de partículas em cada estado, pois uma delas tende a se sobressair as outras e, conseqüentemente, se torna cada vez mais provável encontrar outras partículas no estado da maioria. Há também a presença de um ruído η que é implementado como a probabilidade de uma partícula se alinhar com uma direção diferente por perceber erroneamente qual é a velocidade da maioria.

Uma vez definida o tipo de interação precisamos delimitar a vizinhança que vai interferir no estado de uma partícula. Dizemos que cada partícula está sujeita à influência média dos estados de sete vizinhos, seus seis primeiros vizinhos e ela mesma. A grandeza que define a reorientação de uma partícula é a probabilidade de sua vizinhança ter uma das \mathbf{v}_i direções como maioria. Essa probabilidade é calculada a partir da fatoração das probabilidades de ocupação de cada sítio que compõe essa vizinhança, considerando que os estados prováveis para os sítios são independentes. Dessa maneira se define uma aproximação de campo médio para o alcance das interações.

A Fig.6 mostra a vizinhança que estamos considerando. A partícula central (em preto) é a partícula cujo comportamento queremos analisar; ela tem velocidade \mathbf{v}_i e seus vizinhos (em cinza) podem ou não estar ocupados. Uma possível reorientação na direção de sua velocidade vai depender dos estados de cada sítio que compõe sua vizinhança. No contexto de probabilidade de ocupação, quando uma partícula reorienta sua velocidade por causa da interação média com seus vizinhos isso significa dizer que houve um aumento da probabilidade de ocupação para um dos i estados. Dentro de uma vizinhança pode acontecer de não haver uma maioria e, devido à presença do ruído, se essa configuração ocorre as partículas têm a mesma probabilidade de manter sua velocidade ou mudar para um dos outros dois estados disponíveis.

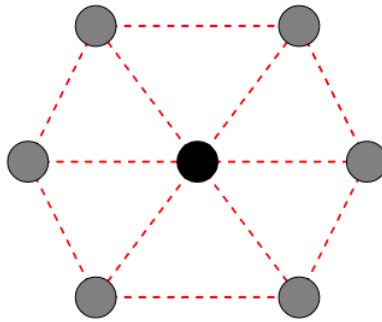


Figura 6 – Vizinhança de interação de uma partícula.

Após a reorientação das velocidades, as partículas podem se mover pela rede nas direções de \mathbf{v}_1 , \mathbf{v}_2 e \mathbf{v}_3 . O movimento é visto como um fluxo de densidades entre dois sítios que ocorre a uma taxa proporcional às probabilidades para cada estado. Nesse passo incluímos uma dependência espacial no sistema, pois agora as densidades dos sítios estão também sujeitas a variações no tempo. O número de partículas no sistema deve se manter constante, portanto a ocupação total da rede é um parâmetro fixo e as variações em ρ_s devem ocorrer de maneira a manter essa restrição.

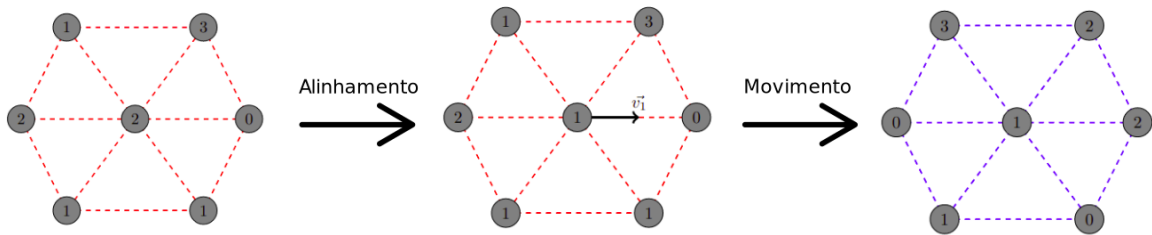


Figura 7 – Dinâmica do modelo: Uma partícula inicialmente com velocidade \mathbf{v}_2 muda para \mathbf{v}_1 para se alinhar com a velocidade majoritária da vizinhança; a partícula então se move para o sítio na direção de \mathbf{v}_1 , que estava vazio, e forma uma nova vizinhança.

4 Análise Computacional

4.1 Introdução

Neste capítulo apresentamos as equações de movimento para as frações de partículas em cada estado e o método utilizado para resolvê-las. A evolução das frações segundo as equações é o que vai modelar o alinhamento do sistema. Posteriormente, explicitamos o processo de transferência de partículas dentro da rede, onde a diferença entre as densidades de dois sítios governa o movimento das partículas. Por fim, apresentamos a construção do modelo em que esses dois passos ocorrem para ser estudado computacionalmente.

4.2 A Equação de Movimento

As frações de partículas em cada estado variam no tempo devido aos efeitos da interação e do ruído. Para escrever uma equação de movimento para as frações precisamos considerar a probabilidade de ocupação de cada sítio para determinar qual estado é dominante em uma vizinhança.

Dentro de uma vizinhança temos quatro eventos possíveis:

- M_1 - a maioria das partículas têm velocidade \mathbf{v}_1 ;
- M_2 - a maioria das partículas têm velocidade \mathbf{v}_2 ;
- M_3 - a maioria das partículas têm velocidade \mathbf{v}_3 ;
- M_0 - não há uma maioria.

Para uma partícula com velocidade \mathbf{v}_i , queremos determinar a probabilidade de ocorrer o evento M_j dentro da vizinhança em que ela se encontra. Há diferentes configurações que levam a um mesmo evento (ver Fig.8). Seja,

$$\mathcal{P}_c = (\rho f_i)_s \prod_{k=1}^6 (\rho f_l)_k, \quad (4.1)$$

a probabilidade de uma configuração que é calculada ao percorrer as probabilidades de ocupação para os estados em cada um dos sítios que formam a vizinhança. Caso um sítio k esteja desocupado, um termo $(1 - \rho_k)$ deve entrar no produto \prod_k .

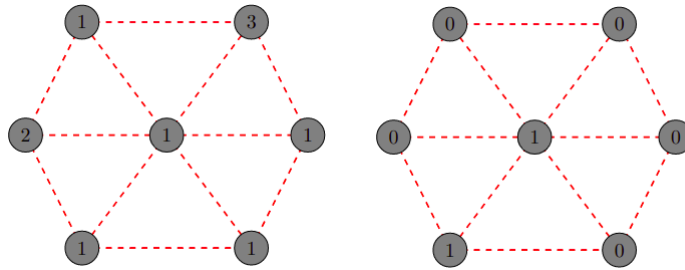


Figura 8 – Exemplo de duas configurações em uma vizinhança que resultam no evento M_1 : Todos os sítios ocupados com cinco deuses com velocidade \mathbf{v}_1 (esquerda) e apenas dois sítios ocupados e com velocidade \mathbf{v}_1 (direita).

A probabilidade total de um evento M_j ocorrer para uma partícula no estado i , denominada como $\mathcal{P}\{M_j|i\}$, é dada pela soma das probabilidades das várias configurações que levam ao evento em questão,

$$\mathcal{P}\{M_j|i\} = \sum_c \mathcal{P}_c\{M_j|i\}. \quad (4.2)$$

Para calcular $\mathcal{P}\{M_j|i\}$ foi construído um algoritmo que permite considerar todas as configurações que uma vizinhança pode assumir e a probabilidade que elas aconteçam, seguindo a Eq.(4.1). O algoritmo é detalhado abaixo.

Algorithm 1 Cálculo das Probabilidades dos Eventos

```

for  $i = 1$  até 3 do
   $p[i] \leftarrow (\rho_s f_{is})$   $\triangleright p[i]$  recebe a probabilidade de ocupação do sítio  $s$  para o estado  $i$ 
  for  $j = 1$  até 4 do  $\triangleright 1^\circ$  vizinho de  $s$ 
    if  $j = 4$  then
       $p[i] \leftarrow p[i] * (1 - \rho_{s_1})$   $\triangleright$  pode estar desocupado
    else
       $p[i] \leftarrow p[i] * (\rho f_j)_{s_1}$ 
    end if
  for  $k = 1$  até 4 do  $\triangleright 2^\circ$  vizinho de  $s$ 
    if  $k = 4$  then
       $p[i] \leftarrow p[i] * (1 - \rho_{s_2})$   $\triangleright$  pode estar desocupado
    else
       $p[i] \leftarrow p[i] * (\rho f_k)_{s_2}$ 
    end if
  for  $l = 1$  até 4 do  $\triangleright 3^\circ$  vizinho de  $s$ 
    if  $l = 4$  then
       $p[i] \leftarrow p[i] * (1 - \rho_{s_3})$   $\triangleright$  pode estar desocupado
    else
       $p[i] \leftarrow p[i] * (\rho f_l)_{s_3}$ 
    end if
  for  $m = 1$  até 4 do  $\triangleright 4^\circ$  vizinho de  $s$ 
    if  $m = 4$  then
       $p[i] \leftarrow p[i] * (1 - \rho_{s_4})$   $\triangleright$  pode estar desocupado
    else
       $p[i] \leftarrow p[i] * (\rho f_m)_{s_4}$ 
    end if
  for  $n = 1$  até 4 do  $\triangleright 5^\circ$  vizinho de  $s$ 
    if  $n = 4$  then
       $p[i] \leftarrow p[i] * (1 - \rho_{s_5})$   $\triangleright$  pode estar desocupado
    else
       $p[i] \leftarrow p[i] * (\rho f_n)_{s_5}$ 
    end if
  for  $o = 1$  até 4 do  $\triangleright 6^\circ$  vizinho de  $s$ 
    if  $o = 4$  then
       $p[i] \leftarrow p[i] * (1 - \rho_{s_6})$   $\triangleright$  pode estar desocupado
    else
       $p[i] \leftarrow p[i] * (\rho f_o)_{s_6}$ 
    end if
     $\mathcal{P}\{M_j|i\} \leftarrow \sum_i p[i]$ 
  end for
end for
end for
end for
end for
end for

```

Definidas as probabilidades, precisamos incluir a ação do ruído para descrever o comportamento esperado de uma partícula ao interagir com sua vizinhança. Seja K o estado da partícula cujo comportamento queremos estudar. Para cada evento possível, a fração de partículas com velocidade \mathbf{v}_1 dentro de um sítio s deve variar da seguinte maneira:

$$M_1 \longrightarrow \begin{cases} K=1: f_{1s} \text{ decresce com probabilidade } \eta; \\ K=2: f_{1s} \text{ aumenta com probabilidade } (1 - \eta); \\ K=3: f_{1s} \text{ aumenta com probabilidade } (1 - \eta). \end{cases}$$

$$M_2 \longrightarrow \begin{cases} K=1: f_{1s} \text{ decresce com probabilidade } (1 - \eta/2); \\ K=2: f_{1s} \text{ aumenta com probabilidade } \eta/2; \\ K=3: f_{1s} \text{ aumenta com probabilidade } \eta/2. \end{cases}$$

$$M_3 \longrightarrow \begin{cases} K=1: f_{1s} \text{ decresce com probabilidade } (1 - \eta/2); \\ K=2: f_{1s} \text{ aumenta com probabilidade } \eta/2; \\ K=3: f_{1s} \text{ aumenta com probabilidade } \eta/2. \end{cases}$$

$$M_0 \longrightarrow \begin{cases} K=1: f_{1s} \text{ decresce com probabilidade } 2\eta/3; \\ K=2: f_{1s} \text{ aumenta com probabilidade } \eta/3; \\ K=3: f_{1s} \text{ aumenta com probabilidade } \eta/3. \end{cases}$$

A análise do comportamento de f_{2s} e f_{3s} é feita de maneira similar e permite escrever as equações de movimento que governam o modelo.

$$\begin{aligned} \frac{df_{is}}{dt} = & -f_{is} \left[\eta P\{M_i|i\} + \left(1 - \frac{\eta}{2}\right) (P\{M_j|i\} + P\{M_k|i\}) + \frac{2\eta}{3} P\{M_0|i\} \right] \\ & + f_{js} \left[(1 - \eta) P\{M_i|j\} + \frac{\eta}{2} (P\{M_j|j\} + P\{M_k|j\}) + \frac{\eta}{3} P\{M_0|j\} \right] \\ & + f_{ks} \left[(1 - \eta) P\{M_i|k\} + \frac{\eta}{2} (P\{M_j|k\} + P\{M_k|k\}) + \frac{\eta}{3} P\{M_0|k\} \right], \end{aligned} \quad (4.3)$$

em que i, j e k assumem os valores 1, 2 e 3 conforme os estados.

4.2.1 Integração Numérica

Para avaliar a evolução das frações em cada estado e, conseqüentemente, o grau de ordenamento do sistema, precisamos resolver a Eq.(4.3) para cada fração para o estado estacionário. Por serem equações altamente não-lineares é necessário integrá-las numericamente e isso é feito utilizando o método de Runge-Kutta de quarta ordem¹ [14]. Tal método segue o algoritmo descrito abaixo para a resolução de EDO's.

Algorithm 2 Método de Runge-Kutta de Quarta Ordem

```

 $f(t, y) \leftarrow dy/dt$  ▷  $f(t, y)$  a equação diferencial a ser resolvida
 $y_n \leftarrow y(t_0)$  ▷ Condição inicial
 $n \leftarrow (t - t_0)/h$  ▷  $h$  o passo de tempo para sair de  $t_0$  e chegar a  $t$ 
for  $i = 1$  até  $n$  do
   $K1 \leftarrow h * f(t_0, y_n)$ 

   $K2 \leftarrow h * f(t_0 + h/2, y_n + K1/2)$ 

   $K3 \leftarrow h * f(t_0 + h/2, y_n + K2/2)$ 

   $K4 \leftarrow h * f(t_0 + h, y_n + K3)$ 

   $K \leftarrow (1/6) * (K1 + 2 * K2 + 2 * K3 + K4)$ 

   $y_n \leftarrow y_n + K$ 
   $t_0 \leftarrow t_0 + h$ 
end for

```

Para resolver as equações de movimento do modelo é preciso considerar que cada uma das equações depende de todas as três frações f_{is} ao mesmo tempo. Apesar de as equações de movimento não apresentarem dependência explícita com o tempo, utilizamos as variáveis t_0 , t e h como tempo inicial, tempo final e passo de tempo, respectivamente. Elas definem o número de iterações a serem feitas e t deve ser grande o suficiente para garantir que, ao completar as iterações, os valores de f_{is} obtidos sejam aqueles que resultam em $df_{is}/dt = 0$. O algoritmo modificado para a resolução das equações de interesse pode ser encontrado no Apêndice B.

¹ Uma breve discussão sobre resoluções de equações diferenciais utilizando o método de Runge-Kutta é encontrada no Apêndice A

4.3 Processo de Transferência de Densidades

Os valores de f_{is} fornecidos pela integração numérica modelam o alinhamento na rede causada pela interação entre primeiros vizinhos. É necessário ainda implementar o movimento das partículas nesse sistema. Para isso precisamos considerar um processo de transferência de densidades entre dois sítios no qual adotamos a condição de volume excluído, ou seja, duas partículas não podem ocupar o mesmo sítio e ρ_s está restrita ao intervalo $[0, 1]$.

Seja o sítio n uma fonte com densidade $s(t)$ e o sítio $n + 1$ um receptor com densidade $r(t)$. Existe uma corrente de densidades, $J_\rho(t)$ entre eles que obedece a relação,

$$\frac{dr(t)}{dt} = -\frac{ds(t)}{dt} = J_\rho(t), \quad (4.4)$$

tal que

$$J_\rho(t) = v_0 s(t) [1 - r(t)]. \quad (4.5)$$

Sendo v_0 o módulo da velocidade das partículas. Fazendo uma mudança de variável, $u(t) = 1 - r(t)$, reescrevemos as equações 4.4 e 4.5 como

$$\frac{du(t)}{dt} = \frac{ds(t)}{dt} = -J_\rho(t) \quad (4.6)$$

e

$$J_\rho(t) = v_0 s(t) u(t). \quad (4.7)$$

Seja s_0 e u_0 o valor das densidades para um tempo t_0 qualquer. E seja Δs a quantidade transferida da fonte para o receptor. O tempo necessário para a transferência de densidades, ou seja, para que a densidade da fonte varie de s_0 para $s_0 - \Delta s$ é dado por

$$\frac{dt}{ds} = -\frac{1}{v_0 s_0 u_0}, \quad (4.8)$$

conforme as equações 4.6 e 4.7. Se s_0 vai diminuir uma quantidade Δs , u_0 deve decrescer o mesmo tanto pela sua definição. Temos a relação

$$\int_0^t dt = -\frac{1}{v_0} \int_0^{\Delta s} \frac{dy}{(s_0 - y)(u_0 - y)}, \quad (4.9)$$

em que a variável y controla o fluxo de densidade. Para resolver a equação acima e obter uma expressão para a transferência de densidade entre dois sítios precisamos considerar dois casos: $s_0 \neq u_0$ e $s_0 = u_0$.

- se $s_0 \neq u_0$

$$t = -\frac{1}{v_0} \int_0^{\Delta s} \frac{dy}{(s_0 - y)(u_0 - y)} = \frac{1}{v_0(s_0 - u_0)} [\ln(y - u_0) - \ln(y - s_0)] \Big|_{y=0}^{y=\Delta s}$$

$$t = \frac{1}{v_0(s_0 - u_0)} \ln \left(\frac{\Delta s/u_0 - 1}{\Delta s/s_0 - 1} \right).$$

$$\Rightarrow \Delta s(t) = -\frac{s_0 [e^{v_0 t(s_0 - u_0)} - 1]}{e^{v_0 t(s_0 - u_0)} - s_0/u_0}. \quad (4.10)$$

- se $s_0 = u_0$

$$t = -\frac{1}{v_0} \int_0^{\Delta s} \frac{dy}{(s_0 - y)^2} = \frac{1}{v_0} \frac{1}{(y - s_0)} \Big|_{y=0}^{y=\Delta s}$$

$$t = \frac{1}{v_0} \frac{\Delta s}{s_0(\Delta s - s_0)}$$

$$\Rightarrow \Delta s(t) = \frac{v_0 t s_0^2}{v_0 t s_0 - 1}. \quad (4.11)$$

A transferência de uma partícula de um sítio para outro deve ser modelada segundo as equações para $\Delta s(t)$. Por exemplo, para mover uma partícula no sítio n para o sítio vizinho que está direção da velocidade \mathbf{v}_1 consideramos a taxa de transferência como sendo $\Delta s(t) f_{1n}$.

4.4 A Rede

Nas seções anteriores foram discutidos os dois sub-passos do modelo, alinhamento e transferência de partículas. Apresentamos agora a construção da rede na qual esses fenômenos irão ocorrer.

No Capítulo 3 foi dito que as partículas ativas se encontram em uma rede triangular bidimensional. Essa geometria é implementada computacionalmente usando uma rede quadrada que é topologicamente equivalente à rede triangular da Fig. 5. Nesse novo formato as orientações das velocidades \mathbf{v}_1 e \mathbf{v}_2 permanecem as mesmas, enquanto \mathbf{v}_3 sofre uma rotação de 45° que compensa a rotação da rede (ver Fig.9). Com essa mudança a rede pode ser vista como uma matriz $L \times L$ (sendo L o tamanho do sistema) cujas células armazenam as informações sobre os sítios.

Seja ρ_g a densidade e η o ruído do sistema que é uniforme inicialmente, assim $\rho_s = \rho_g$ e f_{is} são iguais para todos os sítios. Considere uma célula na posição (m, n) da matriz com $m, n \in [1, L]$; ela guarda a probabilidade de ocupação do sítio para as três frações de partículas em cada estado, $(\rho f_i)_{mn}$.

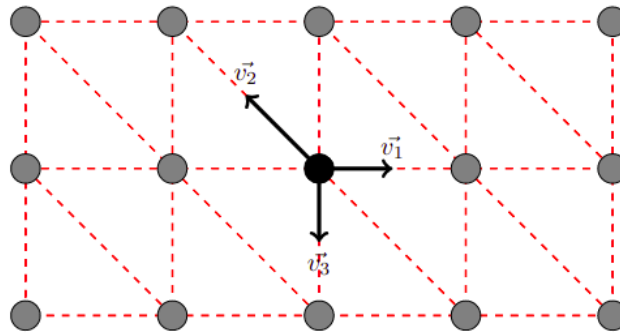


Figura 9 – Rede quadrada com as orientações permitidas para as velocidades.

A rede deve obedecer condições de contorno periódicas,

$$\text{se } m, n < 1 \Rightarrow m, n = L,$$

$$\text{se } m, n > L \Rightarrow m, n = 1,$$

o que permite construir a vizinhança das partículas de modo que haja sempre seis primeiros vizinhos. Definida a vizinhança, as probabilidades de ocupação de cada sítio que a compõe fornece a probabilidade de ocorrer cada um dos quatro eventos M_j . Os valores de $\mathcal{P}\{M_j|i\}$ são incorporados às equações de movimento que são resolvidas através de integração numérica seguindo o Algoritmo 3.

Após a interação de alinhamento uma pequena perturbação na densidade de alguns sítios quebra a homogeneidade da rede e permite o movimento das partículas na direção de sua nova velocidade, desde que o sítio vizinho esteja desocupado. Novas vizinhanças são formadas, mas agora dentro de uma configuração em que as probabilidades de ocupação variam de sítio para sítio. As partículas interagem com seus novos vizinhos e se movem pela rede, esse processo se repete por tempo suficiente para poder revelar um padrão dentro do modelo.

Devido a condição de volume excluído, quando todos os sítios estão ocupados ($\rho_g = 1$) não há movimentação das partículas no sistema. A única ação possível é a reorientação das velocidades causada pela interação com os vizinhos e nesse caso temos um modelo similar ao modelo de Potts de três estados [15] no qual o ruído do sistema cumpre o papel da temperatura.

5 Resultados

5.1 Introdução

Nesse capítulo apresentamos os resultados obtidos para o modelo proposto. A evolução do sistema foi dividida em dois passos: o alinhamento, onde vemos os efeitos da análise de campo médio no modelo, e a transferência, onde o movimento das partículas na rede leva à comportamentos coletivos.

5.2 Alinhamento

Iniciamos o sistema como uma rede homogênea com densidade ρ_g qualquer. Dessa maneira $\rho_s = \rho_g$ e $f_{is} = f_i$ para todo sítio s , sendo f_i a média das frações no estado i . Devido a natureza da interação é preciso definir uma maioria inicial. Assim, consideramos que a maioria das partículas se encontram inicialmente com velocidade \mathbf{v}_1 , enquanto a fração de estados para as outras duas velocidades são iguais, $f_2 = f_3 = f$, tal que

$$\begin{cases} f_1 = 1/3 + \Delta \\ f = 1/3 - \Delta/2. \end{cases} \quad (5.1)$$

A equação obedece à restrição imposta pela Eq.(3.2) desde que $0 < \Delta \leq 2/3$. O caso $\Delta = 0$ não é considerado apesar de satisfazer a restrição, pois implica frações iguais para cada estado. Os resultados foram obtidos para $\Delta = 10^{-3}$ a não ser que seja dito o contrário.

O grau de ordenamento no estado estacionário é influenciado pelo ruído e existe um valor crítico η_c que separa a fase ordenada da fase desordenada. Para $\eta < \eta_c$ as partículas se alinham com a orientação da maioria e o estado estacionário tem \mathbf{v}_1 como velocidade dominante. Conforme o ruído aumenta essa dominância perde força, pois a diferença entre f_1 e f diminui. Quando $\eta \geq \eta_c$ a interação de alinhamento é superada e a diferença entre as frações vai a zero, caracterizando o estado desordenado. A Figura 10 mostra o comportamento descrito.

Com base nesse comportamento definimos o parâmetro de ordem do sistema como

$$\phi(f_1) = f_1 - \frac{1}{3}. \quad (5.2)$$

De modo que para valores altos do ruído $\langle \phi(f_1) \rangle_t = 0$, o que justifica a inclusão do fator $1/3$ na definição.

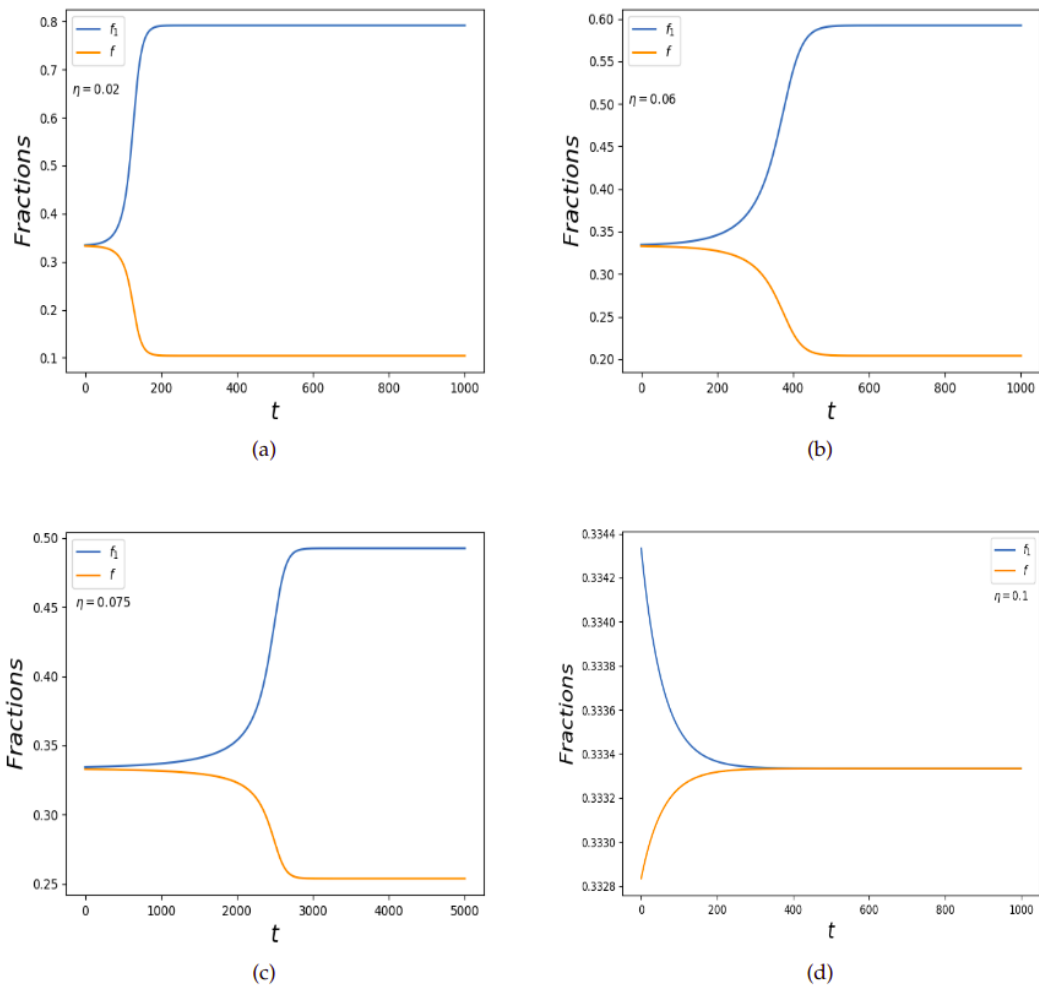


Figura 10 – Evolução temporal de um sistema com densidade $\rho_g = 0.7$ cujo ruído crítico é $\eta_c = 0.0767$. (a) $\eta = 0.02$. (b) $\eta = 0.06$. (c) $\eta = 0.075$. (d) $\eta = 0.1$.

Alterando o valor da densidade para um mesmo ruído, vemos que existe uma relação entre ρ_g e ϕ . A Fig.11-(a) mostra que quanto mais provável for os sítios estarem desocupados menor vai ser a média temporal do parâmetro de ordem. Com isso vemos que a transição de fase do tipo ordem-desordem pela qual o sistema passa tem como parâmetros principais a densidade e o ruído e a relação entre essas duas variáveis fornece o diagrama de fase do modelo, Figura11-(b).

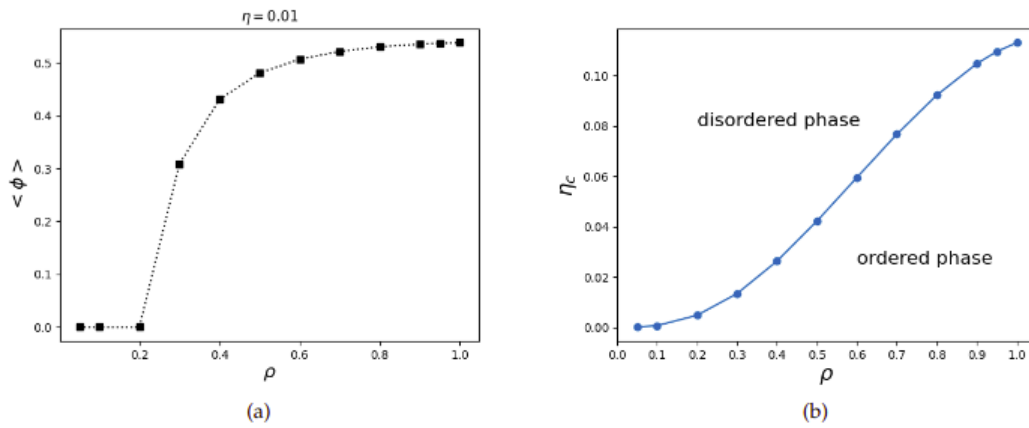


Figura 11 – (a) Parâmetro de ordem em função da densidade para ruído fixo em $\eta = 0.01$. (b) Diagrama de fase do modelo: Acima da curva o sistema se encontra na fase desordenada e abaixo na fase ordenada.

Na Figura 12 vemos uma queda abrupta do parâmetro de ordem quando o sistema atinge o ruído crítico, isso ocorre para todas as densidades consideradas de modo que a aproximação de campo médio resulta em transições de fase descontínuas para o modelo.

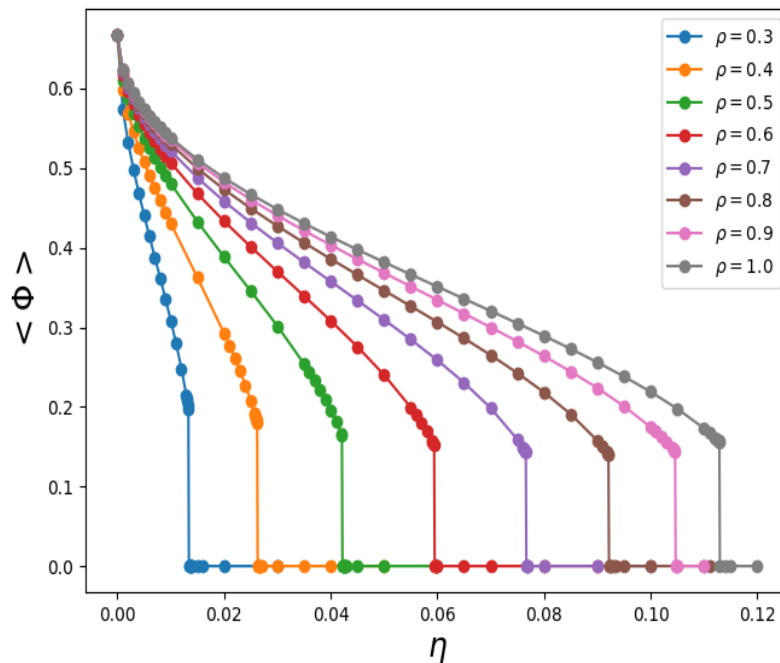


Figura 12 – Parâmetro de ordem em função do ruído para diversos valores de densidade.

5.2.1 Diferentes Condições Iniciais

A Eq.5.1 define as condições iniciais da rede através do parâmetro Δ que controla a diferença entre as frações em cada estado. Fazendo $\Delta \approx 0$ o sistema começa muito próximo do estado desordenado e fazendo $\Delta = 2/3$ iniciamos com todas as partículas com velocidade \mathbf{v}_1 .

Soluções das equações de movimento para diferentes valores de Δ mostram que o ruído crítico não é função apenas da densidade, as condições iniciais do sistema também influenciam. Na Fig.13 vemos que η_c cresce com Δ , ou seja, sistemas mais ordenados são mais difíceis de passar para a fase desordenada. Essa relação porém é limitada, há um $\eta_{c,min}$ e $\eta_{c,max}$ para o qual a transição de fase ocorre e esses limites definem as curvas espinodais do sistema. Para $\rho_g = 1.0$, $\eta_{c,min}$ ocorre para $\Delta \leq 10^{-4}$ e $\eta_{c,max}$ ocorre para $\Delta \geq 7 \times 10^{-2}$.

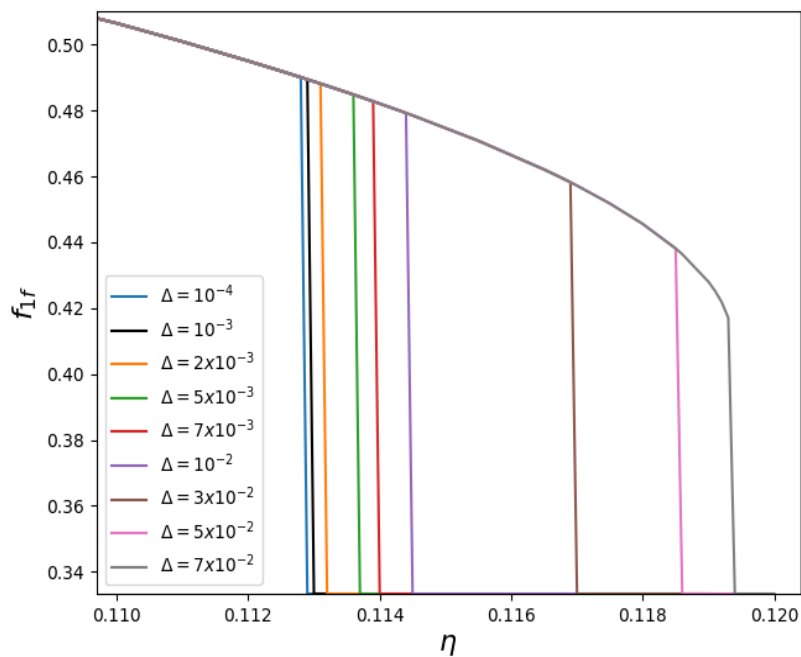


Figura 13 – Região crítica para o sistema com diferentes condições iniciais. As linhas azul e cinza definem os limites do ruído para o qual a transição ocorre. Resultados para $\rho_g = 1.0$

5.2.2 Os Zeros da Equação de Movimento

Ao definir $f_2 = f_3$ como condição inicial, a Eq.(3.2) pode ser reescrita como $f_1 + 2f = 1$ e, portanto,

$$f_1 = 1 - 2f, \quad (5.3)$$

$$f = \frac{1 - f_1}{2}. \quad (5.4)$$

Além disso, o fato de estarmos considerando uma rede uniforme implica $\mathcal{P}\{M_j|2\} = \mathcal{P}\{M_j|3\}$. As equações de movimento podem ser simplificadas e escritas em função de apenas uma variável, para df_1/dt temos

$$\begin{aligned} \frac{df_1}{dt} = & -f_1 \left[\eta \mathcal{P}\{M_1|1\} + \left(1 - \frac{\eta}{2}\right) (\mathcal{P}\{M_2|1\} + \mathcal{P}\{M_3|1\}) + \frac{2\eta}{3} \mathcal{P}\{M_0|1\} \right] \\ & + (1 - f_1) \left[(1 - \eta) \mathcal{P}\{M_1|2\} + \eta \mathcal{P}\{M_2|2\} + \frac{\eta}{3} \mathcal{P}\{M_0|2\} \right]. \end{aligned} \quad (5.5)$$

Sabemos que há valores de f_1 que levam o sistema ao estado estacionário e por causa da restrição sabemos também que $0 \leq f_1 \leq 1$. Uma maneira de avaliar a Eq.5.5 é percorrer o intervalo em que f_1 está contido para encontrar os valores que resultam em $df_1/dt = 0$, para diferentes valores do ruído. A evolução das outras frações é totalmente definida a partir da Eq.5.4, uma vez que se obtém os resultados para f_1 .

Para η em um determinado intervalo a equação de movimento para f_1 é nula em três pontos distintos. A Figura 14(a) mostra que $f_1^* = 1/3$ é sempre um zero da equação; condizente com o fato de que esse valor indica que o sistema está desordenado e, portanto, não há uma maioria a ser seguida e as frações se mantêm constantes. Os outros dois zeros da equação sofrem influência do ruído do sistema, porém o interesse principal é no comportamento do ponto nulo que é maior que f_1^* por corresponder à condição inicial com $f_1 > f$ que adotamos. Denominamos o ponto de interesse como $f_1^>$ e vemos que ele decresce com o aumento do ruído. Ao atingir um certo valor de η , $f_1^>$ deixa de existir e apenas o ponto f_1^* permanece como um zero da equação, como indica a Fig.14(b).

Avaliando os zeros da equação para diferentes valores de η , vemos que o ponto f_1^* é uma solução instável, enquanto os outros dois são soluções estáveis. f_1^* se torna estável quando as duas outras soluções não mais existem, conforme indicado na Fig.15(a), e o ruído para o qual essa mudança ocorre está em acordo com os valores para o ruído crítico obtidos utilizando o método de Runge-Kutta para resolver as equações de movimento. Além disso, o fato de $f_1^>$ deixar de existir abruptamente nesse ponto confirma a existência de uma transição de fase descontínua na teoria de campo médio. Os resultados para o comportamento de $f_1^>$ em função do ruído estão em concordância com os valores de f_1 no estado estacionário fornecido pela integração numérica (ver Fig.15(b)).

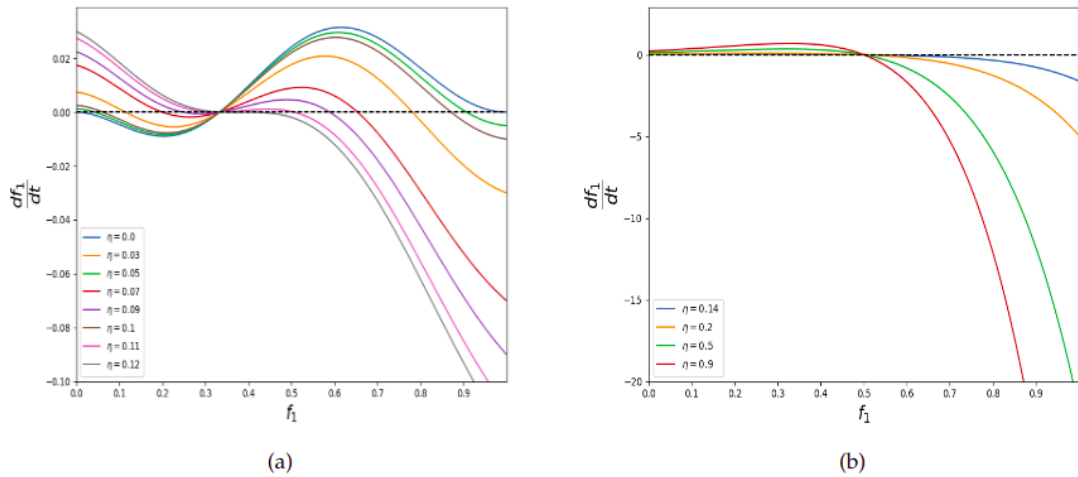


Figura 14 – Equação de movimento em função de f_1 para $\rho = 1.0$. (a) Quando $\eta < \eta_c$ há três pontos em que a equação é nula. (b) Para $\eta > \eta_c$ há apenas um zero correspondente a f_1^* .

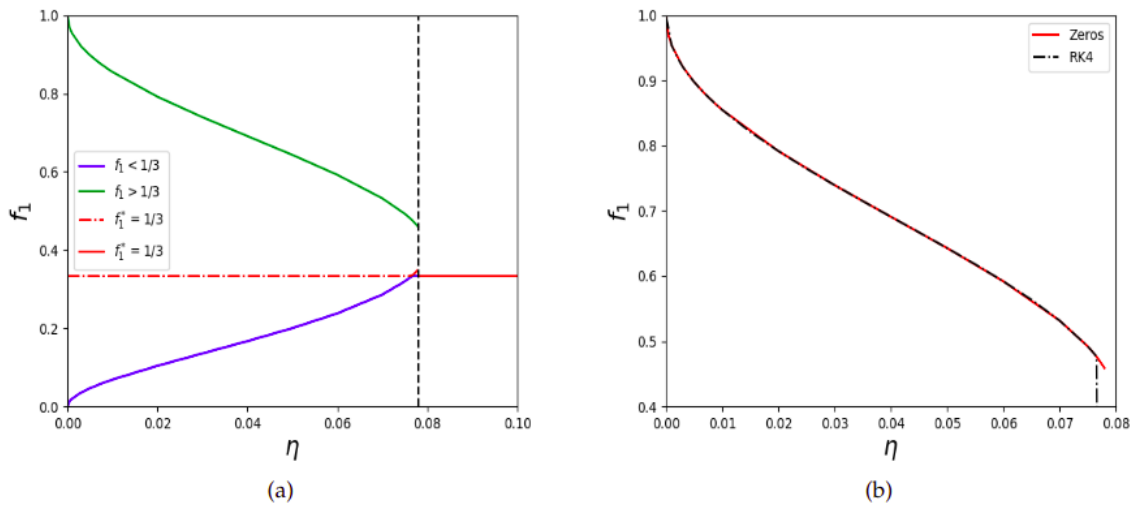


Figura 15 – (a) Zeros da equação de movimento em função do ruído, linhas sólidas indicam soluções estáveis e a linha pontilhada indica solução instável. (b) Evolução de $f_1^>$ na análise dos zeros da equação concorda com os resultados obtidos para f_1 usando o método de Runge-Kutta. Resultados para $\rho_g = 0.7$.

A existência do ponto $f_1^< < f_1^*$ como um dos zeros da equação de movimento parece indicar que há uma fase ordenada e estável em que $f > f_1$, isto é, f_2 e f_3 são maioria. Para confirmar esse resultado foi feita a integração numérica das equações de movimento

utilizando o método de Runge-Kutta de quarta ordem com a seguinte condição inicial,

$$\begin{cases} f_1 = 0.2 \\ f_2 = 0.4 + \epsilon \\ f_3 = 0.4 - \epsilon, \end{cases} \quad (5.6)$$

sendo $\epsilon \ll 1$ uma pequena perturbação. Os valores de f_1 para o estado estacionário coincidem com os zeros da equação para diferentes valores do ruído (ver Fig.16(a)) e as duas outras velocidades dividem a dominância na rede. Porém a Fig.16(b) mostra que essa fase não é de fato estável, pois ao acrescentar a perturbação a velocidade \mathbf{v}_2 eventualmente domina o sistema, enquanto f_3 decresce até se igualar com f_1 .

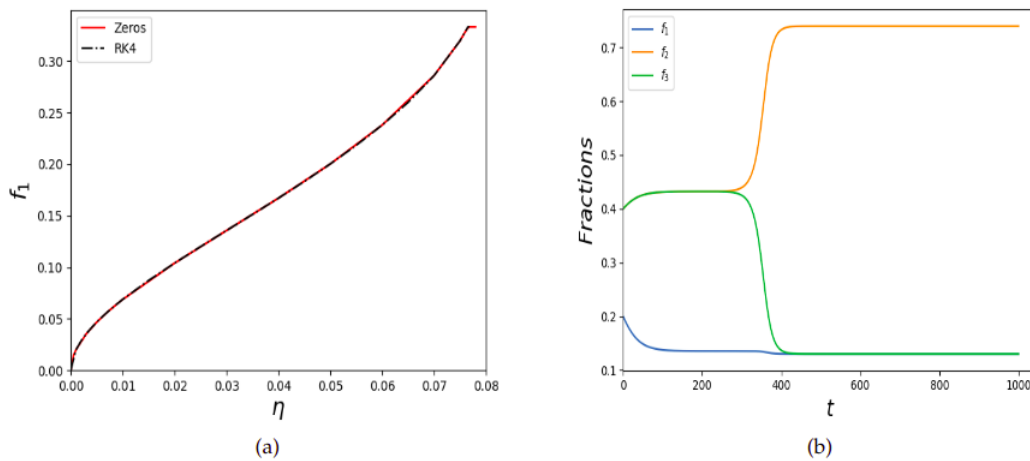


Figura 16 – (a) Evolução de $f_1^<$ na análise dos zeros da equação concorda com os resultado obtidos para f_1 usando o método de Runge-Kutta ($\epsilon = 0$). (b) Evolução temporal das frações em cada estado com condições iniciais dadas pela Eq.(5.6) com $\epsilon = 10^{-10}$ e $\eta = 0.03$. Resultados para $\rho_g = 0.7$

5.3 Movimento das Partículas

Após o alinhamento, as partículas se movem na direção de suas novas velocidades. Como discutido na Seção 4.3, esse passo é modelado como um processo de transferência de densidades entre uma fonte e um receptor¹. Definimos a velocidade das partículas como $v_0 = 0.5$ e os movimentos ocorrem em um intervalo de tempo h que é igual ao passo de tempo definido na integração numérica seguindo o método de Runge-Kutta. Foram estudadas redes de tamanhos $L = 5$ e $L = 10$.

Primeiramente observamos o comportamento do sistema ao adicionar uma pequena perturbação na densidade da rede logo após o alinhamento. Essas perturbações foram realizadas na direção de \mathbf{v}_1 e de \mathbf{v}_3 separadamente. Considerando que a rede é homogênea inicialmente ($\rho_s = \rho_g$) e que ela é tratada como uma matriz $L \times L$, uma perturbação ϵ na direção de \mathbf{v}_1 implica fazer com que uma coluna tenha densidade $\rho = \rho_g + \epsilon$ enquanto a coluna adjacente passa a ter densidade $\rho = \rho_g - \epsilon$, e as demais não se alteram. Na direção de \mathbf{v}_3 essas mudanças são realizadas nas linhas da matriz. As partículas então se movem em uma rede não uniforme, novas vizinhanças são formadas e o alinhamento ocorre novamente.

Para condições iniciais dadas pela Eq.(5.1), a perturbação na direção da velocidade 1 faz com que as densidades e frações em cada estado oscilem em torno do estado estacionário alcançado no passo de alinhamento para a densidade e ruído considerados, como mostra a Fig.17 A amplitude da oscilação diminui a cada passo de tempo até que o sistema volta ao estado estacionário e a rede é homogênea novamente. Ocorrem pequenas mudanças na fração média de partículas em cada estado durante a dinâmica, mas não tem efeito significativo no grau de ordenamento da rede.

Para perturbação na direção de \mathbf{v}_3 na rede com maioria com velocidade \mathbf{v}_1 , o sistema também tende a voltar para o estado estacionário mas o comportamento oscilatório não ocorre. Há uma variação grande nos valores das densidades e frações nas linhas que sofreram a perturbação e então o sistema rapidamente volta à configuração inicial (ver Fig.18).

Não houve mudanças no ordenamento da rede causados pela perturbação. Para o ruído ligeiramente menor que o ruído crítico as variações nos parâmetros não são grandes o suficiente para levar a desordem e o comportamento segue o que foi dito anteriormente para ambas direções de perturbação. Já para $\eta \geq \eta_c$ não foram observados comportamentos oscilatórios, a rede sempre volta rapidamente para a configuração de desordem uniforme.

¹ Os resultados apresentados foram obtidos a partir de uma solução aproximada da Eq.(4.4 ao invés da solução exata apresentada no Cap.4

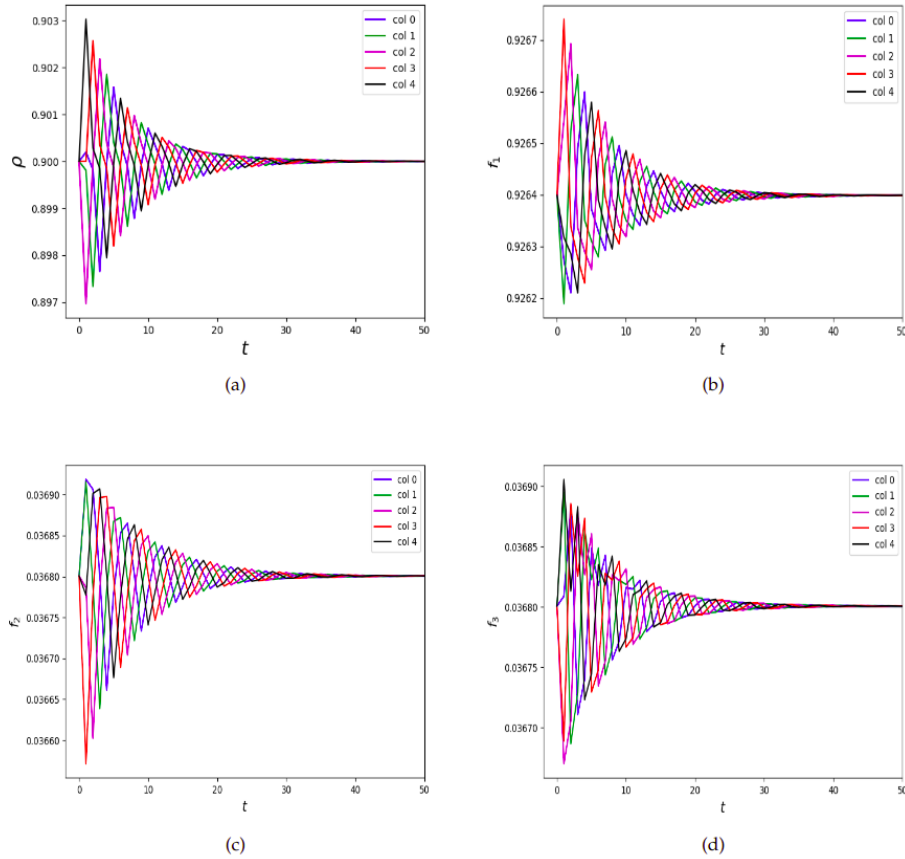


Figura 17 – Variação dos parâmetros do modelo quando uma perturbação na densidade na direção de \mathbf{v}_1 é adicionada. (a) Variação da densidade nas colunas. (b) a (d) Variações das frações nas colunas. $\rho = 0.9$, $\eta = 0.03$, $L = 5$

Na busca pelo surgimento de configurações diferentes as condições iniciais foram alteradas. Consideramos uma região em que os sítios estão todos ocupados, $\rho_s = 1$, e com velocidade na direção 1. O resto da rede tem densidade baixa e se encontra no estado desordenado, $f_1 = f_2 = f_3$. Dessa maneira a rede é iniciada com uma banda orientada na direção de \mathbf{v}_3 e com a fração média de partículas com \mathbf{v}_1 sendo maior que a fração média para os outros estados. Dentro da banda não há movimento de partículas na direção \mathbf{v}_1 devido a condição de volume excluído (ver Fig.19).

Ao deixar o sistema evoluir seguindo os passos de alinhamento seguido da transferência de partículas foi observado que a densidade tende a se uniformizar e eventualmente chega-se em $\rho_s = \rho_g$ para todos os sítios. O comportamento das frações de partículas em cada estado por sua vez foi inesperado, pois ocorre uma mudança repentina no ordenamento da rede: f_1 decresce para valores próximos de zero e passa a evoluir de maneira semelhante a f_3 , ao mesmo tempo a velocidade 2 cresce rapidamente e domina a rede. Eventualmente um estado estacionário uniforme é alcançado com \mathbf{v}_2 sendo a direção preferencial na rede. Esse comportamento foi observado para os casos em que 2/5 e 1/5 da rede estavam totalmente ocupados inicialmente e pode ser visto na Fig.20.

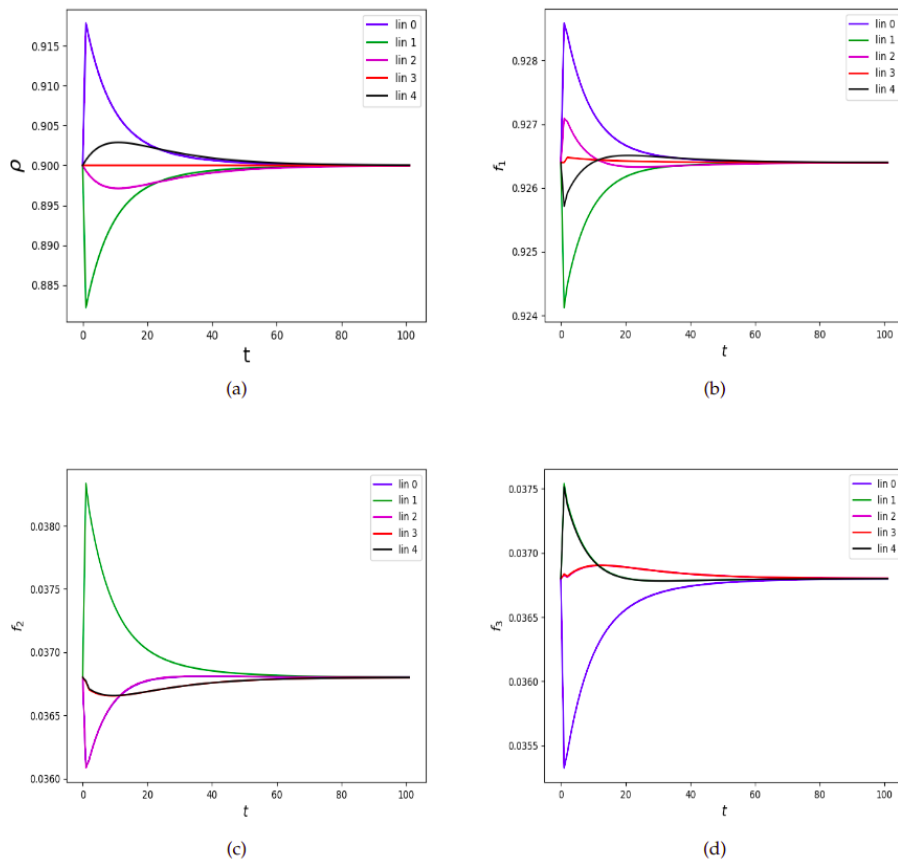


Figura 18 – Variação dos parâmetros do modelo quando uma perturbação na densidade na direção de \mathbf{v}_3 é adicionada. (a) Variação da densidade nas linhas. (b) a (d) Variações das frações nas linhas. $\rho = 0.9$, $\eta = 0.03$, $L = 5$

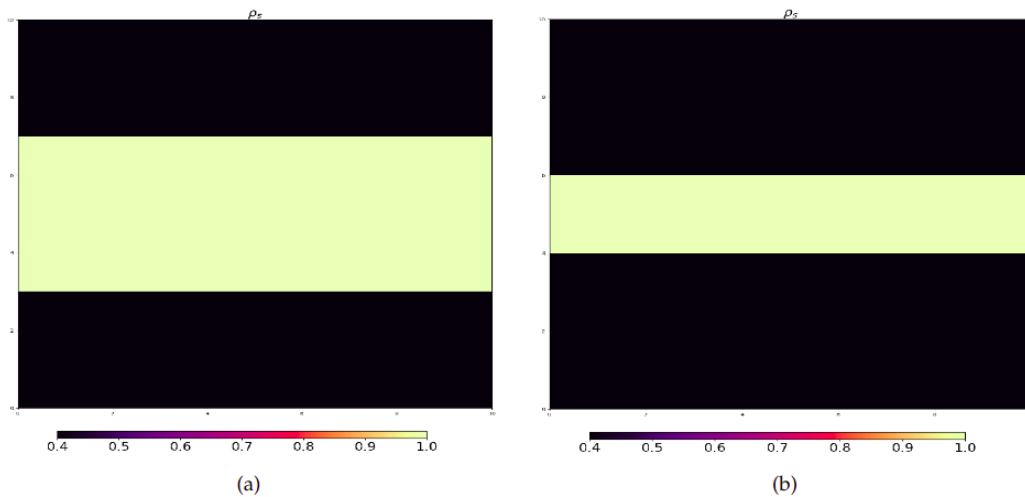


Figura 19 – Rede de tamanho $L = 10$ com densidade não uniforme inicialmente. (a) 2/5 da rede tem densidade $\rho_s = 1.0$ com $\rho_g = 0.64$. (b) 1/5 da rede tem densidade $\rho_s = 1.0$ com $\rho_g = 0.52$.

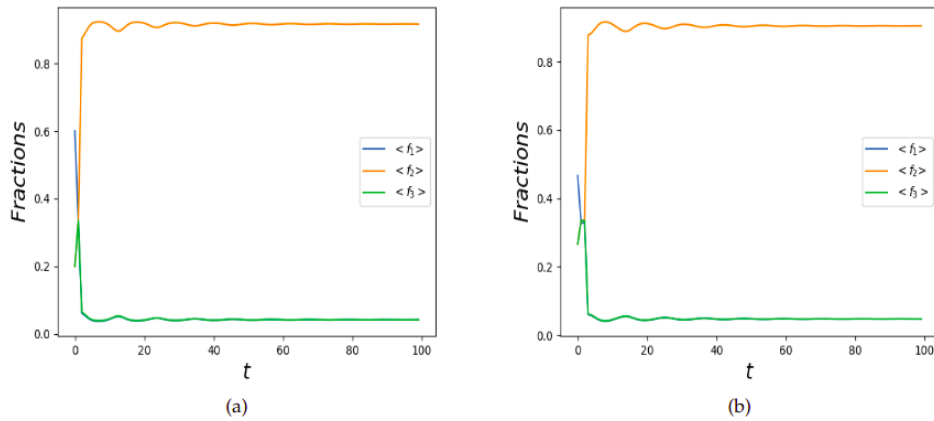


Figura 20 – Variação das frações médias de partículas em cada estado para (a) $2/5$ da rede com densidade alta inicialmente e (b) $1/5$ da rede com densidade alta inicialmente. $\eta = 0.003$, $L = 10$.

Apesar de a velocidade 1 dominar a banda no início, o surgimento de partículas que se alinham com \mathbf{v}_2 e \mathbf{v}_3 nessa região é esperado devido a estocasticidade do modelo. Porém, \mathbf{v}_1 é a velocidade majoritária da rede no início e a queda brusca de f_1 para ceder a dominância na rede para uma direção que era minoria precisa ser melhor analisada, pois parece ir contra os resultados já obtidos para o ordenamento a ruídos baixos.

6 Conclusões e Trabalhos Futuros

A análise de campo médio para um modelo de matéria ativa em rede no qual os indivíduos podem assumir apenas três velocidades e estão sujeitos a uma interação de alinhamento resulta em uma transição de fase do tipo ordem-desordem que é descontínua. Esse resultado é obtido pela integração numérica das equações de movimento do modelo com condições iniciais bem definidas e corroborado pelo método de identificar os zeros da equação, que vale para rede uniforme. A ocorrência da transição de fase é controlada pela relação entre densidade e ruído, que fornece o diagrama de fase do modelo. E também pelas condições iniciais para as frações em cada estado, o que permite definir as curvas espinodais. A quebra espontânea de simetria na transição de fase do modelo é indicada pelo comportamento do parâmetro de ordem quando sai de $\langle \phi \rangle_t = 0$, no estado em que todas as partículas tem probabilidades iguais de estar em um dos três estados, para $\langle \phi \rangle_t > 0$, quando uma direção de velocidade começa a dominar.

Os resultados obtidos para dependência espacial porém não são satisfatórios e indicam que ainda é preciso melhorar o desempenho do programa desenvolvido para a análise computacional do modelo. Feito isso, planejamos estudar o comportamento de sistemas maiores e comparar os resultados da aproximação de campo médio com simulações que estão sendo realizadas para um modelo similar cuja vizinhança de interação é maior e no qual a influência dos vizinhos no estado de uma partícula é definida de maneira mais precisa.

Queremos utilizar o modelo desenvolvido como base para o estudo de modelos ainda mais complexos. Usando a teoria de campo médio vamos investigar o comportamento de redes quando: a condição de volume excluído não é obedecida; o número de partículas na rede não é mais uma quantidade conservada devido a inclusão de uma dinâmica populacional no sistema.

Referências

- [1] Vicsek, T., A. Czirók, E. Ben-Jacob, I. Cohen e O. Shochet: *Novel Type of Phase Transition in a System of Self-Driven Particles*. Phys. Rev. Lett., 75:1226–1229, 1995. Citado 2 vezes nas páginas 11 e 13.
- [2] Ginelli, F., F. Peruani, M. H. Pillot, H. Chaté, G. Theraulaz e R. Bon: *Intermittent collective dynamics emerge from conflicting imperatives in sheep herds*. Proceedings of the National Academy of Sciences, 112(41):12729–12734, 2015. Citado na página 11.
- [3] Zhang, H. P., A. Be'er, E. L. Florin e H. L. Swinney: *Collective motion and density fluctuations in bacterial colonies*. Proceedings of the National Academy of Sciences, 107(31):13626–13630, 2010. Citado na página 11.
- [4] Volker, S., C. Weber, C. Semmrich, E. Frey e A. R. Bausch: *Polar patterns of driven filaments*. Nature, 467(7311):73–77, 2010. Citado na página 11.
- [5] Nosenko, V., F. Luoni, A. Kaouk, M. Rubin-Zuzic e H. Thomas: *Active Janus particles in a complex plasma*. Phys. Rev. Research, 2:033226, 2020. Citado na página 11.
- [6] Marchetti, M. C., J.F. Jean, S. Ramaswamy, T. B. Liverpool, J. Prost, M. Rao e R. A. Simha: *Hydrodynamics of soft active matter*. Reviews of modern physics, 85(3):1143, 2013. Citado 2 vezes nas páginas 11 e 17.
- [7] Ballerini, M., N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, M. Procaccini, M. Viale e V. Zdravkovic: *Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study*. Proceedings of the National Academy of Sciences, 105(4):1232–1237, 2008. Citado na página 11.
- [8] Ginelli, F.: *The Physics of the Vicsek model*. The European Physical Journal Special Topics, 225(11):2099–2117, 2016. Citado 3 vezes nas páginas 11, 14 e 15.
- [9] Peruani, F., T. Klauss, A. Deutsch e A. Voss-Boehme: *Traffic Jams, Gliders, and Bands in the Quest for Collective Motion of Self-Propelled Particles*. Phys. Rev. Lett., 106:128101, 2011. Citado 3 vezes nas páginas 12, 16 e 17.
- [10] Tomé, T. e M. J. Oliveira: *Dinâmica Estocástica e Irreversibilidade*. Editora da Universidade de São Paulo (Edusp), 2014. Citado na página 14.
- [11] Grégoire, G. e H. Chaté: *Onset of Collective and Cohesive Motion*. Phys. Rev. Lett., 92:025702, 2004. Citado na página 15.

-
- [12] Menon, S. N., T. Bagarti e A. Chakraborty: *Jamming in a lattice model of stochastically interacting agents with a field of view*. EPL (Europhysics Letters), 117(5):50007, 2017. Citado na página 17.
- [13] Whitelam, S., K. Klymko e D. Mandal: *Phase separation and large deviations of lattice active matter*. The Journal of Chemical Physics, 148(15):154902, 2018. Citado na página 17.
- [14] Cheever, Eric: *Approximation of Differential Equations by Numerical Integration*, 2014. <https://lpsa.swarthmore.edu/NumInt/NumIntIntro.html>. Citado na página 25.
- [15] Wu, F. Y.: *The Potts model*. Rev. Mod. Phys., 54:235–268, 1982. Citado na página 28.

APÊNDICE A – Método de Runge-Kutta para Resolução de Equações Diferenciais

O método de Runge-Kutta é um dos métodos mais populares para resolver EDO's. Ele se baseia na expansão em séries de Taylor da função que se quer resolver. Seja uma equação diferencial dada por

$$\frac{dy(t)}{dt} = f(t, y(t)), \quad y(t_0) = y_0. \quad (\text{A.1})$$

O objetivo é determinar o valor de $y(t)$, para um t arbitrário. Fazendo a expansão em séries de Taylor em torno de um valor t_0 , temos

$$y(t) = y(t_0) + \frac{dy(t_0)}{dt}(t - t_0) + \frac{1}{2} \frac{d^2y(t_0)}{dt^2}(t - t_0)^2 + \mathcal{O}((t - t_0)^3). \quad (\text{A.2})$$

Truncando a expansão na primeira derivada e definindo um passo de tempo h para ir de t_0 a t , aproximamos a expressão acima para

$$y(h) \approx y(t_0) + \frac{dy(t_0)}{dt}h. \quad (\text{A.3})$$

O termo $dy(t_0)/dt$ é calculado a partir das condições iniciais da equação diferencial e definido como k_1 . Dessa maneira se tem uma aproximação para a derivada no ponto de interesse e, a partir dela, se obtém o valor aproximado da função no ponto $(t + h)$,

$$\begin{aligned} k_1 &= f(t_0, y(t_0)) \\ y(t_0 + h) &= y(t_0) + k_1h. \end{aligned} \quad (\text{A.4})$$

Repetindo o processo calcula-se $y(t + 2h)$ e segue até chegar ao valor da função no ponto de interesse, $y(t)$.

O método descrito é chamado Runge-Kutta de primeira ordem, ou método de Euler, e tem um erro global da ordem de $\mathcal{O}(h)$.

Seguindo a ideia do método de Euler, é possível obter resultados mais precisos para o valor da função $y(t)$ ao considerar o ponto médio do intervalo na aproximação. Esse é o método de Runge-Kutta de segunda ordem cujo erro global passa a ser $\mathcal{O}(h^2)$.

Consideramos novamente a Eq.A.1 e a aproximação expressa na Eq.A.3. k_1 é usado para calcular o valor da função no ponto médio do intervalo, ou seja, em $(t_0 + h/2)$. Por sua vez, $y(t_0 + h/2)$ permite estimar a inclinação da curva nesse ponto e obter uma aproximação melhor para $y(t + h)$ a partir de $f(t_0 + h/2, y(t_0 + h/2))$. Portanto, a integração numérica

segue

$$\begin{aligned}
 k_1 &= f(t_0, y(t_0)) \\
 y_1(t_0 + h/2) &= y(t_0) + \frac{1}{2}k_1h \\
 k_2 &= f(t_0 + h/2, y_1(t_0 + h/2)) \\
 y(t_0 + h) &= y(t_0) + k_2h.
 \end{aligned} \tag{A.5}$$

Que novamente deve ser iterada até obter $y(t)$.

Para resultados ainda mais precisos é utilizado o método de Runge-Kutta de quarta ordem. Agora o intervalo é dividido em quatro partes iguais com k_1 e k_2 agindo da mesma forma que no caso de segunda ordem. Porém, incluímos um novo ponto no meio do intervalo que vai ser calculado através de k_2 e cuja inclinação é dada por k_3 . Com k_3 é feita a estimativa da função no final do intervalo que tem inclinação dada pela derivada nesse ponto, k_4 . A função no intervalo é dada pela média das quatro inclinações k_1, k_2, k_3 e k_4 , com as inclinações no ponto médio k_2 e k_3 recebendo um peso maior.

$$\begin{aligned}
 k_1 &= f(t_0, y(t_0)) \\
 y_1(t_0 + h/2) &= y(t_0) + \frac{1}{2}k_1h \\
 k_2 &= f(t_0 + h/2, y_1(t_0 + h/2)) \\
 y_2(t_0 + h/2) &= y(t_0) + \frac{1}{2}k_2h \\
 k_3 &= f(t_0 + h/2, y_2(t_0 + h/2)) \\
 y_3(t_0 + h) &= y(t_0) + k_3h \\
 k_4 &= f(t_0, y_3(t_0)) \\
 y(t_0 + h) &= y(t_0) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h.
 \end{aligned} \tag{A.6}$$

O erro global nesse caso é da ordem de $\mathcal{O}(h^5)$. Os pesos para as inclinações estão associados ao método utilizado para a resolução da EDO. Acima foi descrito o método de Runge-Kutta clássico de quarta ordem, no qual as inclinações no ponto médio tem peso igual a dois. Os chamados método de Runge-Kutta explícitos apresentam pesos diferentes para as inclinações.

APÊNDICE B – Método de Runge-Kutta para Resolução das Equações de Movimento do Modelo

Para resolver as equações de movimento 4.3 levamos em conta o fato de que a evolução de uma fração está interligada à evolução das outras duas e que não há uma dependência temporal explícita. O algoritmo do método de Runge-Kutta precisou então ser alterado para que essas diferenças fossem consideradas durante a integração numérica.

Algorithm 3 Método de Runge-Kutta de Quarta Ordem para as Equações de Movimento

```

 $df_j(f_1, f_2, f_3) \leftarrow df_j/dt$  ▷ Equação de movimento para a fração  $j$ 
 $f_j \leftarrow f_j(t_0)$  ▷ Condição inicial para a fração  $j$ 
 $n \leftarrow (t - t_0)/h$  ▷  $h$  o passo de tempo para sair de  $t_0$  e chegar a  $t$ 
for  $i = 1$  até  $n$  do
   $K1 \leftarrow h * df_1(f_1, f_2, f_3)$ 
   $C1 \leftarrow h * df_2(f_1, f_2, f_3)$ 
   $Q1 \leftarrow h * df_3(f_1, f_2, f_3)$ 

   $K2 \leftarrow h * df_1(f_1 + K1/2, f_2 + C1/2, f_3 + Q1/2)$ 
   $C2 \leftarrow h * df_2(f_1 + K1/2, f_2 + C1/2, f_3 + Q1/2)$ 
   $Q2 \leftarrow h * df_3(f_1 + K1/2, f_2 + C1/2, f_3 + Q1/2)$ 

   $K3 \leftarrow h * df_1(f_1 + K2/2, f_2 + C2/2, f_3 + Q2/2)$ 
   $C3 \leftarrow h * df_2(f_1 + K2/2, f_2 + C2/2, f_3 + Q2/2)$ 
   $Q3 \leftarrow h * df_3(f_1 + K2/2, f_2 + C2/2, f_3 + Q2/2)$ 

   $K4 \leftarrow h * df_1(f_1 + K3, f_2 + C3, f_3 + Q3)$ 
   $C4 \leftarrow h * df_2(f_1 + K3, f_2 + C3, f_3 + Q3)$ 
   $Q4 \leftarrow h * df_3(f_1 + K3, f_2 + C3, f_3 + Q3)$ 

   $K \leftarrow (1/6) * (K1 + 2 * K2 + 2 * K3 + K4)$ 
   $C \leftarrow (1/6) * (C1 + 2 * C2 + 2 * C3 + C4)$ 
   $Q \leftarrow (1/6) * (Q1 + 2 * Q2 + 2 * Q3 + Q4)$ 

   $f_1 \leftarrow f_1 + K$ 
   $f_2 \leftarrow f_2 + C$ 
   $f_3 \leftarrow f_3 + Q$ 

   $t_0 \leftarrow t_0 + h$ 
end for

```

APÊNDICE C – Código do Modelo em C

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<math.h>
4 #include<time.h>
5
6 #define Lx  10
7 #define Ly  10
8 int x, y;
9 double p, delta, f1, f2, f3;
10 double prob_11, prob_21, prob_31, prob_01, prob_12, prob_22,
    prob_32, prob_02, prob_13, prob_23, prob_33, prob_03;
11 double n, n0, n1, n2, n3, n4;
12 double h, vpar;
13
14 void def_rede (double (*info)[4][Lx][Ly]);
15 void array3d (double (*arr)[4][Lx][Ly]);
16 double df1dt (double f1c, double f2c, double f3c);
17 double df2dt (double f1c, double f2c, double f3c);
18 double df3dt (double f1c, double f2c, double f3c);
19 void alinhamento (double (*info)[4][Lx][Ly]);
20 void transferencia (double (*info)[4][Lx][Ly]);
21
22 int main (int argc, char *argv[]){
23     char Data[100], Data1[100], Data2[100], Data3[100], Data4[100];
24     FILE *data, *data1, *data2, *data3, *data4;
25     p = atof(argv[1]);
26     n = atof(argv[2]);
27     sprintf(Data, "avg_Lx%d_dens%.2lf_eta%.3lf_delta%lf.txt", Lx,
        p, n, delta);
28     sprintf(Data1, "f1_Lx%d_dens%.2lf_eta%.3lf_delta%lf.txt", Lx,
        p, n, delta);
29     sprintf(Data2, "f2_Lx%d_dens%.2lf_eta%.3lf_delta%lf.txt", Lx,
        p, n, delta);
30     sprintf(Data3, "f3_Lx%d_dens%.2lf_eta%.3lf_delta%lf.txt", Lx,
        p, n, delta);
31     sprintf(Data4, "rho_Lx%d_dens%.2lf_eta%.3lf_delta%lf.txt", Lx,
        p, n, delta);
32     data = fopen (Data, "w");

```

```
33 data1 = fopen(Data1, "w");
34 data2 = fopen(Data2, "w");
35 data3 = fopen(Data3, "w");
36 data4 = fopen(Data4, "w");
37
38 delta = 0.0;
39 f1 = 1.0/3.0 + delta;
40 f2 = f3 = 1.0/3.0 - delta/2.0;
41
42 n0 = 1.0 - (n/2.0);
43 n1 = (2.0/3.0)*n;
44 n2 = 1.0 - n;
45 n3 = n/2.0;
46 n4 = n/3.0;
47 h = 0.01;
48 vpar = 0.5; //velocidade da particula
49 int ar = Lx*Ly;
50 double vr = 0.01; //parametro de variacao da densidade
51 double info[2][4][Lx][Ly];
52 for (int a = 0; a < 2; a++){
53     for (int b = 0; b < 4; b++){
54         for (int c = 0; c < Lx; c++){
55             for (int d = 0; d < Ly; d++){
56                 info[a][b][c][d] = 0.0;
57             }
58         }
59     }
60 }
61 def_rede(info);
62
63 for (x = 1; x <= 2; x++){
64     for (y = 0; y < Ly; y++){
65         info[0][0][x][y] = 0.99;
66         info[0][1][x][y] = 1.0;
67         info[0][2][x][y] = 0.0;
68         info[0][3][x][y] = 0.0;
69     }
70 }
71
72 for (x = 0; x < Lx; x++){
73     for (y = 0; y < Ly; y++){
74         fprintf(data4, "%.10lf ", info[0][0][x][y]);
```

```
75     fprintf(data1, "%.10lf ", info[0][1][x][y]);
76     fprintf(data2, "%.10lf ", info[0][2][x][y]);
77     fprintf(data3, "%.10lf ", info[0][3][x][y]);
78 }
79 }
80 fprintf(data1, "\n");
81 fprintf(data2, "\n");
82 fprintf(data3, "\n");
83 fprintf(data4, "\n");
84 array3d(info);
85
86 for (int m = 0; m < 700000; m++){
87     double avrho = 0.0;
88     double avf1 = 0.0;
89     double avf2 = 0.0;
90     double avf3 = 0.0;
91     double avrho2 = 0.0;
92     double avf12 = 0.0;
93     double avf22 = 0.0;
94     double avf32 = 0.0;
95     double varrho, varf1, varf2, varf3;
96     for (x = 0; x < Lx; x++){
97         for (y = 0; y < Ly; y++){
98             double rh = info[0][0][x][y];
99             double f1a = info[0][1][x][y];
100            double f2a = info[0][2][x][y];
101            double f3a = info[0][3][x][y];
102            avrho += rh;
103            avf1 += f1a;
104            avf2 += f2a;
105            avf3 += f3a;
106        }
107    }
108    avrho = avrho/ar;
109    avf1 = avf1/ar;
110    avf2 = avf2/ar;
111    avf3 = avf3/ar;
112    for (x = 0; x < Lx; x++){
113        for (y = 0; y < Ly; y++){
114            double rh_2 = info[0][0][x][y] - avrho;
115            double f1a_2 = info[0][1][x][y] - avf1;
116            double f2a_2 = info[0][2][x][y] - avf2;
```

```
117     double f3a_2 = info[0][3][x][y] - avf3;
118     avrho2 += rh_2*rh_2;
119     avf12 += f1a_2*f1a_2;
120     avf22 += f2a_2*f2a_2;
121     avf32 += f3a_2*f3a_2;
122 }
123 }
124 varrho = avrho2/ar;
125 varf1 = avf12/ar;
126 varf2 = avf22/ar;
127 varf3 = avf32/ar;
128
129 fprintf(data, "%.201f %.201f %.201f %.201f %.201f %.201f
%.201f %.201f\n", avrho, varrho, avf1, varf1, avf2, varf2,
avf3, varf3);
130
131 alinhamento(info);
132 transferencia(info);
133
134 for (x = 0; x < Lx; x++){
135     for (y = 0; y < Ly; y++){
136         fprintf(data4, "%.101f ", info[0][0][x][y]);
137         fprintf(data1, "%.101f ", info[0][1][x][y]);
138         fprintf(data2, "%.101f ", info[0][2][x][y]);
139         fprintf(data3, "%.101f ", info[0][3][x][y]);
140     }
141 }
142 fprintf(data1, "\n");
143 fprintf(data2, "\n");
144 fprintf(data3, "\n");
145 fprintf(data4, "\n");
146 }
147 fclose(data);
148 fclose(data1);
149 fclose(data2);
150 fclose(data3);
151 fclose(data4);
152 }
153
154 /*Equacoes de movimento*/
155 double df1dt (double f1c, double f2c, double f3c) {
```

```
156     return -f1c*(n*prob_11 + n0*(prob_21 + prob_31) + n1*prob_01)
        + f2c*(n2*prob_12 + n3*(prob_22 + prob_32) + n4*prob_02) +
        f3c*(n2*prob_13 + n3*(prob_23 + prob_33) + n4*prob_03);
157 }
158
159 double df2dt (double f1c, double f2c, double f3c) {
160     return -f2c*(n*prob_22 + n0*(prob_12 + prob_32) + n1*prob_02)
        + f1c*(n2*prob_21 + n3*(prob_11 + prob_31) + n4*prob_01) +
        f3c*(n2*prob_23 + n3*(prob_13 + prob_33) + n4*prob_03);
161 }
162
163 double df3dt(double f1c, double f2c, double f3c) {
164     return -f3c*(n*prob_33 + n0*(prob_13 + prob_23) + n1*prob_03)
        + f1c*(n2*prob_31 + n3*(prob_11 + prob_21) + n4*prob_01) +
        f2c*(n2*prob_32 + n3*(prob_12 + prob_22) + n4*prob_02);
165 }
166
167 void alinhamento (double (*info)[4][Lx][Ly]){
168     double al[4][Lx][Ly]; //array auxiliar
169     for (int a = 0; a < 4; a++){
170         for (x = 0; x < Lx; x++){
171             for (y = 0; y < Ly; y++){
172                 info[1][a][x][y] = info[0][a][x][y];
173             }//fecha a
174         }//fecha b
175     }//fecha c
176
177     /*Parametros de RK4*/
178     double t0 = 0.0;
179     double t = 0.01;
180     int r = (int) ((t-t0)/h);
181     double f1_0, f2_0, f3_0;
182
183     for (int g = 1; g <= r; g++){ // inicia a iteracao RK4
184         double prob_c, prob[7][3][4], prob_t[3][4];
185         for (int a = 0; a < 4; a++){
186             for (x = 0; x < Lx; x++){
187                 for (y = 0; y < Ly; y++){
188                     al[a][x][y] = info[1][a][x][y];
189                 }//fecha a
190             }//fecha y
191         }//fecha x
```

```
192  /*Loop da rede*/
193  for (x = 0; x < Lx; x++){
194      for (y = 0; y < Ly; y++){
195          /*Condições de contorno*/
196          int x1 = x+1;
197          int x2 = x-1;
198          int y1 = y+1;
199          int y2 = y-1;
200          if (x1 == Lx){
201              x1 = 0;
202          }
203          if (y1 == Ly){
204              y1 = 0;
205          }
206          if (x2 < 0){
207              x2 = Lx-1;
208          }
209          if (y2 < 0){
210              y2 = Ly-1;
211          }
212          /*Densidades e dens. de prob. em (x, y)*/
213          double rho = al[0][x][y];
214          double f1c = al[1][x][y];
215          double f2c = al[2][x][y];
216          double f3c = al[3][x][y];
217          double dp1c = rho*f1c;
218          double dp2c = rho*f2c;
219          double dp3c = rho*f3c;
220          double central[3] = {dp1c, dp2c, dp3c};
221          /*Densidades e dens. de prob. em (x, y+1)*/
222          double rho_p1 = al[0][x][y1];
223          double dp1_p1 = rho_p1*al[1][x][y1];
224          double dp2_p1 = rho_p1*al[2][x][y1];
225          double dp3_p1 = rho_p1*al[3][x][y1];
226          double dp0_p1 = 1 - rho_p1;
227          double viz_1[4] = {dp1_p1, dp2_p1, dp3_p1, dp0_p1};
228          /*Densidades e dens. de prob. em (x, y-1)*/
229          double rho_n1 = al[0][x][y2];
230          double dp1_n1 = rho_n1*al[1][x][y2];
231          double dp2_n1 = rho_n1*al[2][x][y2];
232          double dp3_n1 = rho_n1*al[3][x][y2];
233          double dp0_n1 = 1 - rho_n1;
```

```
234     double viz_2[4] = {dp1_n1, dp2_n1, dp3_n1, dp0_n1};
235     /*Desnsidades e dens. de prob. em (x-1, y-1)*/
236     double rho_p2 = al[0][x2][y2];
237     double dp1_p2 = rho_p2*al[1][x2][y2];
238     double dp2_p2 = rho_p2*al[2][x2][y2];
239     double dp3_p2 = rho_p2*al[3][x2][y2];
240     double dp0_p2 = 1- rho_p2;
241     double viz_3[4] = {dp1_p2, dp2_p2, dp3_p2, dp0_p2};
242     /*Densidades e dens. de prob. em (x+1, y+1)*/
243     double rho_n2 = al[0][x1][y1];
244     double dp1_n2 = rho_n2*al[1][x1][y1];
245     double dp2_n2 = rho_n2*al[2][x1][y1];
246     double dp3_n2 = rho_n2*al[3][x1][y1];
247     double dp0_n2 = 1 - rho_n2;
248     double viz_4[4] = {dp1_n2, dp2_n2, dp3_n2, dp0_n2};
249     /*Densidades e dens. de prob. em (x+1, y)*/
250     double rho_p3 = al[0][x1][y];
251     double dp1_p3 = rho_p3*al[1][x1][y];
252     double dp2_p3 = rho_p3*al[2][x1][y];
253     double dp3_p3 = rho_p3*al[3][x1][y];
254     double dp0_p3 = 1 - rho_p3;
255     double viz_5[4] = {dp1_p3, dp2_p3, dp3_p3, dp0_p3};
256     /*Densidades e dens. de prob. em (x-1, y)*/
257     double rho_n3 = al[0][x2][y];
258     double dp1_n3 = rho_n3*al[1][x2][y];
259     double dp2_n3 = rho_n3*al[2][x2][y];
260     double dp3_n3 = rho_n3*al[3][x2][y];
261     double dp0_n3 = 1 - rho_n3;
262     double viz_6[4] = {dp1_n3, dp2_n3, dp3_n3, dp0_n3};
263     for (int a = 0; a < 3; a++){
264         for (int b = 0; b < 4; b++){
265             prob_t[a][b] = 0;
266         }//fecha a
267     }//fecha b
268     /*Formacao da vizinhanca*/
269     for (int i = 0; i < 3; i++){//particula central
270         for (int a = 0; a < 4; a++){
271             prob[0][i][a] = 1;
272             prob[1][i][a] = prob[0][i][a];
273         }
274         for (int j = 0; j < 4; j++){//primeiro vizinho
275             for (int a = 0; a < 4; a++){
```



```
318         prob[5][2][3] = prob[0][2][3];
319         prob[5][3][3] = prob[0][3][3];
320     }
321     prob[6][i][a] = prob[5][i][a];
322 }
323 for (int o = 0; o < 4; o++){//sexto vizinho
324     for (int a = 0; a < 4; a++){
325         if (o == a){
326             prob[6][i][a] *= viz_6[o];
327         } else{
328             prob[6][i][a] = prob[5][i][a];
329         }
330     }
331     prob_c = central[i] * prob[6][i][0] *
prob[6][i][1] * prob[6][i][2] * prob[6][i][3];
332     int v;
333     /*Contador*/
334     int b = 0;
335     int c = 0;
336     int d = 0;
337     if (i == 0){
338         b++;
339     }
340     if (i == 1){
341         c++;
342     }
343     if (i == 2){
344         d++;
345     }
346     if (j == 0){
347         b++;
348     }
349     if (j == 1){
350         c++;
351     }
352     if (j == 2){
353         d++;
354     }
355     if (k == 0){
356         b++;
357     }
358     if (k == 1){
```

```
359         c++;
360     }
361     if (k == 2){
362         d++;
363     }
364     if (l == 0){
365         b++;
366     }
367     if (l == 1){
368         c++;
369     }
370     if (l == 2){
371         d++;
372     }
373     if (m == 0){
374         b++;
375     }
376     if (m == 1){
377         c++;
378     }
379     if (m == 2){
380         d++;
381     }
382     if (n == 0){
383         b++;
384     }
385     if (n == 1){
386         c++;
387     }
388     if (n == 2){
389         d++;
390     }
391     if (o == 0){
392         b++;
393     }
394     if (o == 1){
395         c++;
396     }
397     if (o == 2){
398         d++;
399     }
400     int cont[3] = {b, c, d};
```

```
401         if ((cont[0] == cont[1] && cont[0] >
cont[2]) || (cont[0] == cont[2] && cont[0] > cont[1]) ||
(cont[1] == cont[2] && cont[1] > cont[0]) || (cont[0] ==
cont[1] && cont[0] == cont[2])){
402             v = 0; //sem maioria
403         }
404         if (cont[0] > cont[1] && cont[0] >
cont[2]){
405             v = 1; //vizinhanca com maioria 1
406         }
407         if (cont[1] > cont[0] && cont[1] >
cont[2]){
408             v = 2; //vizinhanca com maioria 2
409         }
410         if (cont[2] > cont[0] && cont[2] >
cont[1]){
411             v = 3; //vizinhanca com maioria 3
412         }
413         prob_t[i][v] += prob_c;
414
415         prob_11 = prob_t[0][1];
416         prob_21 = prob_t[0][2];
417         prob_31 = prob_t[0][3];
418         prob_01 = prob_t[0][0];
419         prob_12 = prob_t[1][1];
420         prob_22 = prob_t[1][2];
421         prob_32 = prob_t[1][3];
422         prob_02 = prob_t[1][0];
423         prob_13 = prob_t[2][1];
424         prob_23 = prob_t[2][2];
425         prob_33 = prob_t[2][3];
426         prob_03 = prob_t[2][0];
427     }//fecha o
428     }//fecha n
429     }//fecha m
430     }//fecha l
431     }//fecha k
432     }//fecha j
433 }//fecha i
434
435 /*Passos de RK4*/
436 double k0, k1, k2, k3, k4;
```

```
437     double c0, c1, c2, c3, c4;
438     double q0, q1, q2, q3, q4;
439     double H = 0.5;
440
441     k1 = h*df1dt(f1c, f2c, f3c);
442     c1 = h*df2dt(f1c, f2c, f3c);
443     q1 = h*df3dt(f1c, f2c, f3c);
444
445     k2 = h*df1dt(f1c+H*k1, f2c+H*c1, f3c+H*q1);
446     c2 = h*df2dt(f1c+H*k1, f2c+H*c1, f3c+H*q1);
447     q2 = h*df3dt(f1c+H*k1, f2c+H*c1, f3c+H*q1);
448
449     k3 = h*df1dt(f1c+H*k2, f2c+H*c2, f3c+H*q2);
450     c3 = h*df2dt(f1c+H*k2, f2c+H*c2, f3c+H*q2);
451     q3 = h*df3dt(f1c+H*k2, f2c+H*c2, f3c+H*q2);
452
453     k4 = h*df1dt(f1c+k3, f2c+c3, f3c+q3);
454     c4 = h*df2dt(f1c+k3, f2c+c3, f3c+q3);
455     q4 = h*df3dt(f1c+k3, f2c+c3, f3c+q3);
456
457     k0 = (1.0/6.0)*(k1 + 2.0*k2 + 2.0*k3 + k4);
458     c0 = (1.0/6.0)*(c1 + 2.0*c2 + 2.0*c3 + c4);
459     q0 = (1.0/6.0)*(q1 + 2.0*q2 + 2.0*q3 + q4);
460
461     f1_0 = f1c + k0;
462     f2_0 = f2c + c0;
463     f3_0 = f3c + q0;
464
465     info[1][1][x][y] = f1_0;
466     info[1][2][x][y] = f2_0;
467     info[1][3][x][y] = f3_0;
468     }//fecha y
469     }//fecha x
470 }//fecha RK4
471 printf("df1=%.10lf\tdf2=%.10lf\tdf3=%.10lf\n", df1dt(f1_0,
472     f2_0, f3_0), df2dt(f1_0, f2_0, f3_0), df3dt(f1_0, f2_0,
473     f3_0));
474 }//fecha alinhamento
475
476 void transferencia(double (*info)[4][Lx][Ly]){
477     double tr[4][Lx][Ly]; //array auxiliar
```

```
476 double aux_0[3][Lx][Ly], aux_1[Lx][Ly], aux_2[Lx][Ly],
    aux_3[Lx][Ly];
477 double aux[3][Lx][Ly];
478 for (int a = 0; a < 4; a++){
479     for (x = 0; x < Lx; x++){
480         for (y = 0; y < Ly; y++){
481             tr[a][x][y] = info[1][a][x][y];
482         }
483     }
484 }
485 for (int b = 0; b < 3; b++){
486     for (x = 0; x < Lx; x++){
487         for (y = 0; y < Ly; y++){
488             aux_0[b][x][y] = 0.0;
489             aux[b][x][y] = 0.0;
490         }
491     }
492 }
493 for (x = 0; x < Lx; x++){
494     for (y = 0; y < Ly; y++){
495         aux_1[x][y] = 0.0;
496         aux_2[x][y] = 0.0;
497         aux_3[x][y] = 0.0;
498     }
499 }
500 for (x = 0; x < Lx; x++){
501     for (y = 0; y < Ly; y++){
502         /*Condicoes de contorno*/
503         int x1 = x+1;
504         int x2 = x-1;
505         int y1 = y+1;
506         int y2 = y-1;
507         if (x1 == Lx){
508             x1 = 0;
509         }
510         if (y1 == Ly){
511             y1 = 0;
512         }
513         if (x2 < 0){
514             x2 = Lx-1;
515         }
516         if (y2 < 0){
```

```
517     y2 = Ly-1;
518 }
519 double rho = tr[0][x][y];
520 double drho = 1.0 - rho;
521 double dp1 = rho*tr[1][x][y];
522 double dp2 = rho*tr[2][x][y];
523 double dp3 = rho*tr[3][x][y];
524 aux[0][x][y] = dp1;
525 aux[1][x][y] = dp2;
526 aux[2][x][y] = dp3;
527
528 /*Tranferencia na direcao de v1*/
529 double rho_v1 = tr[0][x][y1];
530 double dp_v1 = rho_v1*tr[1][x][y1];
531 double u1 = 1.0 - dp_v1 - rho_v1*(tr[2][x][y1] +
tr[3][x][y1]);
532 double q1 = dp1 + u1;
533 double z1 = -vpar*h*q1;
534 double delta_v1 = ((dp1*u1)/q1)*(1.0 - exp(z1));
535
536 aux_1[x][y] = delta_v1;
537
538 /*Tranferencia na direcao de v2*/
539 double rho_v2 = tr[0][x2][y2];
540 double dp_v2 = rho_v2*tr[2][x2][y2];
541 double u2 = 1.0 - dp_v2 - rho_v2*(tr[1][x2][y2] +
tr[3][x2][y2]);
542 double q2 = dp2 + u2;
543 double z2 = -vpar*h*q2;
544 double delta_v2 = ((dp2*u2)/q2)*(1.0 - exp(z2));
545
546 aux_2[x][y] = delta_v2;
547
548 /*Tranferencia na direcao de v3*/
549 double rho_v3 = tr[0][x1][y];
550 double dp_v3 = rho_v3*tr[3][x1][y];
551 double u3 = 1.0 - dp_v3 - rho_v3*(tr[1][x1][y1] +
tr[2][x1][y]);
552 double q3 = dp3 + u3;
553 double z3 = -vpar*h*q3;
554 double delta_v3 = ((dp3*u3)/q3)*(1.0 - exp(z3));
555
```

```
556     aux_3[x][y] = delta_v3;
557     }//fecha y
558 }//fecha x
559
560 /*Atualizacao das densidades de probabilidades*/
561 /*Condicoes de contorno*/
562 for (x = 0; x < Lx; x++){
563     for (y = 0; y < Ly; y++){
564         int x1 = x+1;
565         int x2 = x-1;
566         int y1 = y+1;
567         int y2 = y-1;
568         if (x1 == Lx){
569             x1 = 0;
570         }
571         if (y1 == Ly){
572             y1 = 0;
573         }
574         if (x2 < 0){
575             x2 = Lx-1;
576         }
577         if (y2 < 0){
578             y2 = Ly-1;
579         }
580         /*Atualiza os valores de dp1*/
581         aux[0][x][y] -= aux_1[x][y];
582         aux[0][x][y] += aux_1[x][y2];
583
584         /*Atualiza os valores de dp2*/
585         aux[1][x][y] -= aux_2[x][y];
586         aux[1][x][y] += aux_2[x1][y1];
587
588         /*Atualiza os valores de dp3*/
589         aux[2][x][y] -= aux_3[x][y];
590         aux[2][x][y] += aux_3[x2][y];
591     }
592 }
593 /*Atualizacao da rede*/
594 for (x = 0; x < Lx; x++){
595     for (y = 0; y < Ly; y++){
596         double rho_new = aux[0][x][y] + aux[1][x][y] +
aux[2][x][y];
```

```
597     double f1_new = aux[0][x][y]/rho_new;
598     double f2_new = aux[1][x][y]/rho_new;
599     double f3_new = aux[2][x][y]/rho_new;
600
601     info[0][0][x][y] = rho_new;
602     info[0][1][x][y] = f1_new;
603     info[0][2][x][y] = f2_new;
604     info[0][3][x][y] = f3_new;
605     if (rho_new > 1.0 || rho_new < 0.0){
606         printf("Condicao de normalizacao violada\n");
607     }
608     }//fecha y
609 } //fecha x
610 }//fecha transferencia
611
612 void array3d (double (*arr)[4][Lx][Ly]){
613     for (int m = 0; m < 2; m++){
614         for (int i = 0; i < 4; i++){
615             for (int j = 0; j < Lx; j++){
616                 //printf("{");
617                 for (int k = 0; k < Ly; k++){
618                     printf(" %.10lf ", arr[m][i][j][k]);
619                 }
620                 //printf("}");
621                 printf("\n");
622             }
623             printf("\n");
624         }
625         printf("\n");
626     }
627 }
628
629 void def_rede (double (*info)[4][Lx][Ly]){
630     for (x = 0; x < Lx; x++){
631         for (y = 0; y < Ly; y++){
632             info[0][0][x][y] = p;
633             info[0][1][x][y] = f1;
634             info[0][2][x][y] = f2;
635             info[0][3][x][y] = f3;
636         }
637     }
638 }
```