# Computação Científica e Mecânica Estatística: sistema de discos em duas dimensões

AG França, HA Fernandes, PF Gomes (UFG-Jataí), RS Grisotto (Unicamp), BM Rocha e FAAMN Soares (UFG-Goiânia)

# Seção 1

1. Introdução
2. Modelo Teórico
3. Implementação computacional
4. Resultados
   - 4.1 Monte Carlo Cadeia de Eventos
   - 4.2 HOOMD-blue HPMC
5. Conclusões
6. Extras

# Fases: sólido, líquido e gasoso



Água nas 3 fases. Fonte: Wikipedia.

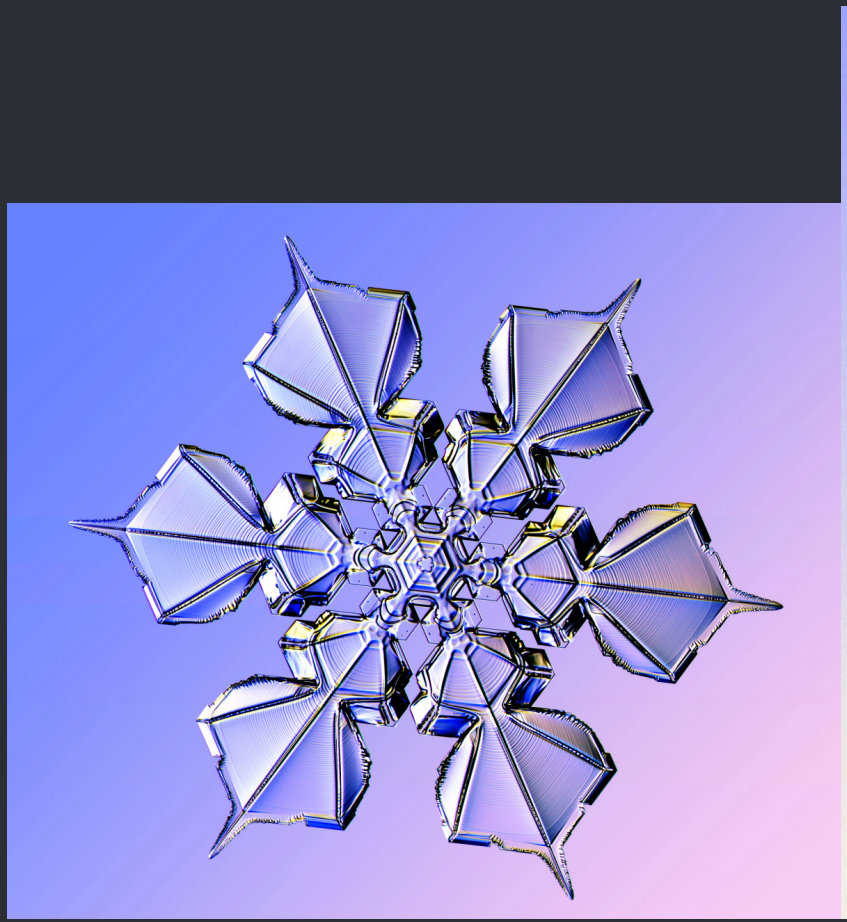
# Parâmetro de ordem

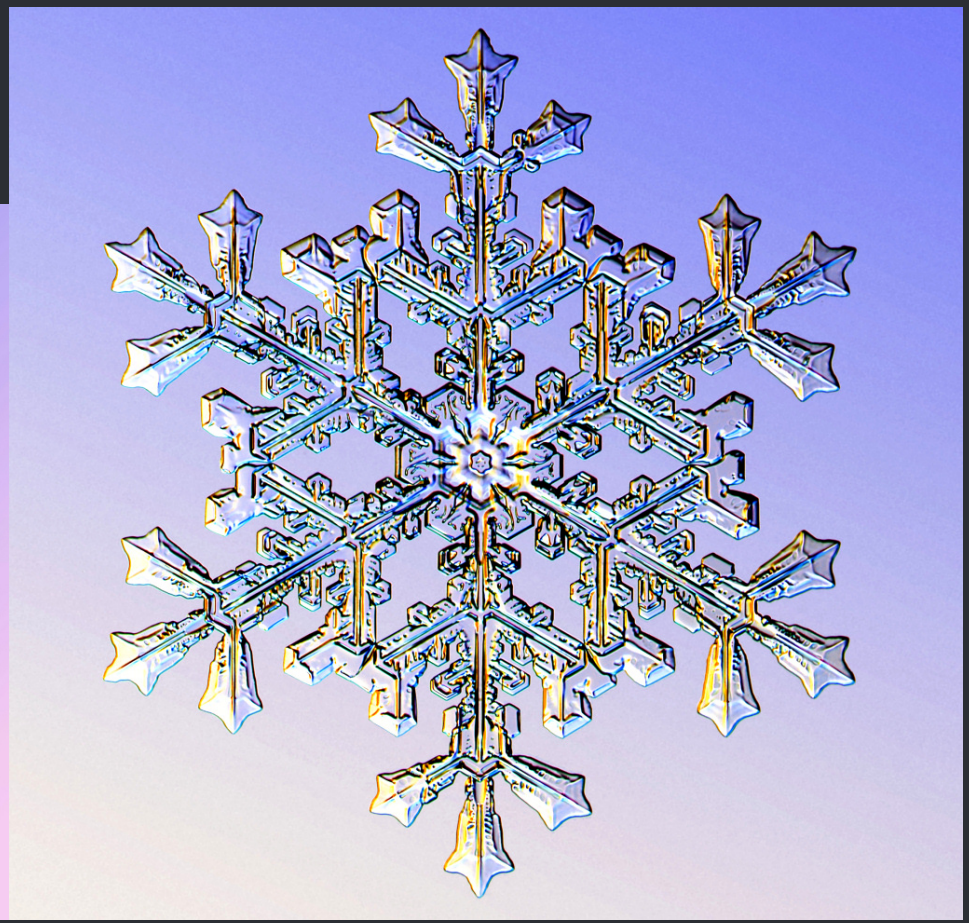
- Cada fase é caracterizada por um parâmetro de ordem.
- Água: organização espacial das moléculas de $H_2O$.

# Parâmetro de ordem

- Cada fase é caracterizada por um parâmetro de ordem.
- Água: organização espacial das moléculas de $H_2O$.
- Fase sólida: moléculas formam uma rede cristalina (hexagonal).
- Líquida: não há rede cristalina mas aglomerados.
- Gás: nem rede cristalina nem atração (uma molécula não sente interação da outra).

# Fase sólida





http://www.snowcrystals.com/.

# Transição de Fase

- Parâmetro de controle: determina a fase. Exemplo: temperatura para a água.

# Transição de Fase

- Parâmetro de controle: determina a fase. Exemplo: temperatura para a água.
- Outros fenômenos apresentam mais de uma fase: magnetismo, interações coletivas, eletricidade.
- Algumas propiedades são observadas em diferentes sistemas.
- A busca desses padrões é de grande relevância em Física.

# Fase hexática

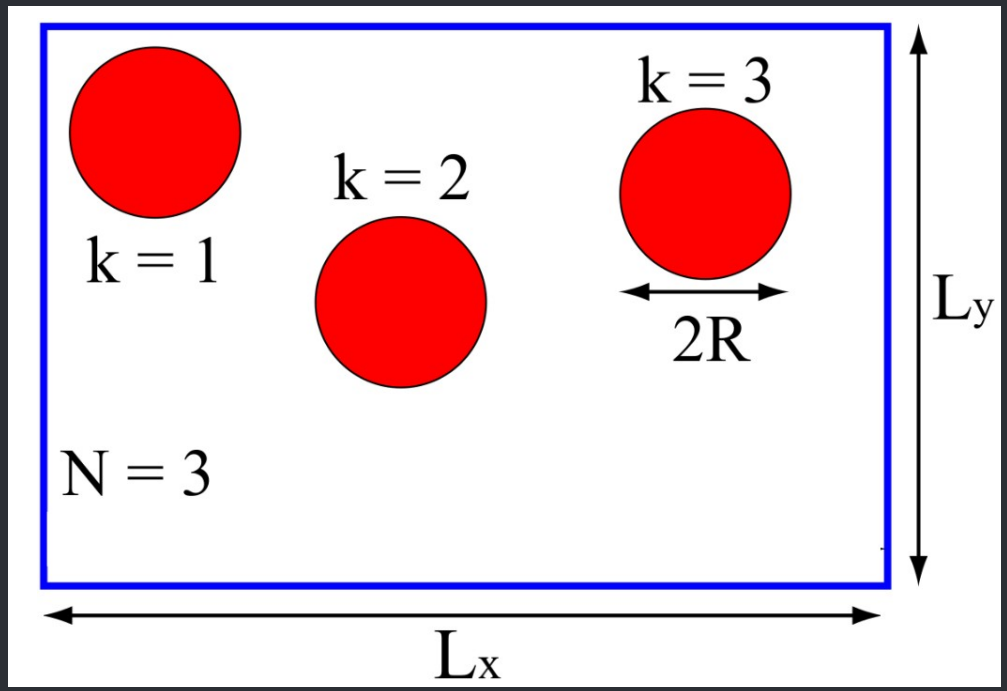
- Transições de fase em 2D: diferenças em relação ao caso 3D.
- Exisência da fase hexática (e suas transições): é um tópico muito estudado em Mecânica Estatística.
- É observado experimentalmente [1]!
- Este trabalho: caracterização de um sistema 2D que apresenta a fase hexática.

# Seção 2

1. Introdução
2. Modelo Teórico
3. Implementação computacional
4. Resultados
   - 4.1 Monte Carlo Cadeia de Eventos
   - 4.2 HOOMD-blue HPMC
5. Conclusões
6. Extras

# Sistema

- $N$ discos rígidos (não deformáveis) de raio $R$ em um espaço retangular de duas dimensões $L_x$ e $L_y$.
- Superposição entre discos é proibida: distância mínima $2R$.

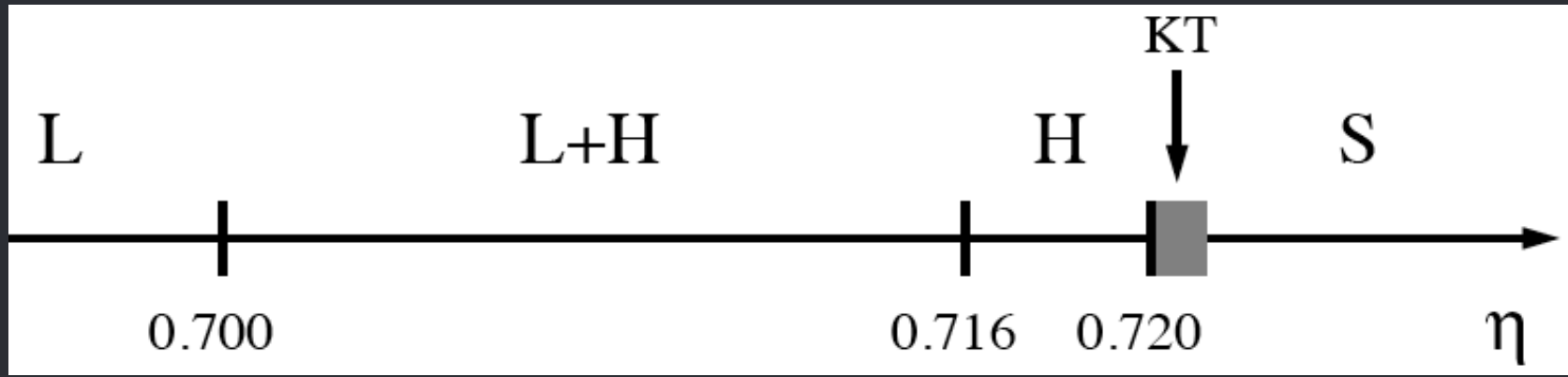


# Discos rígidos em 2D

- Parâmetro de controle: densidade

$$\eta = \frac{N\pi R^2}{L_x L_y}.$$

- Faz o mesmo papel que a temperatura no caso da água.



# Orientação global

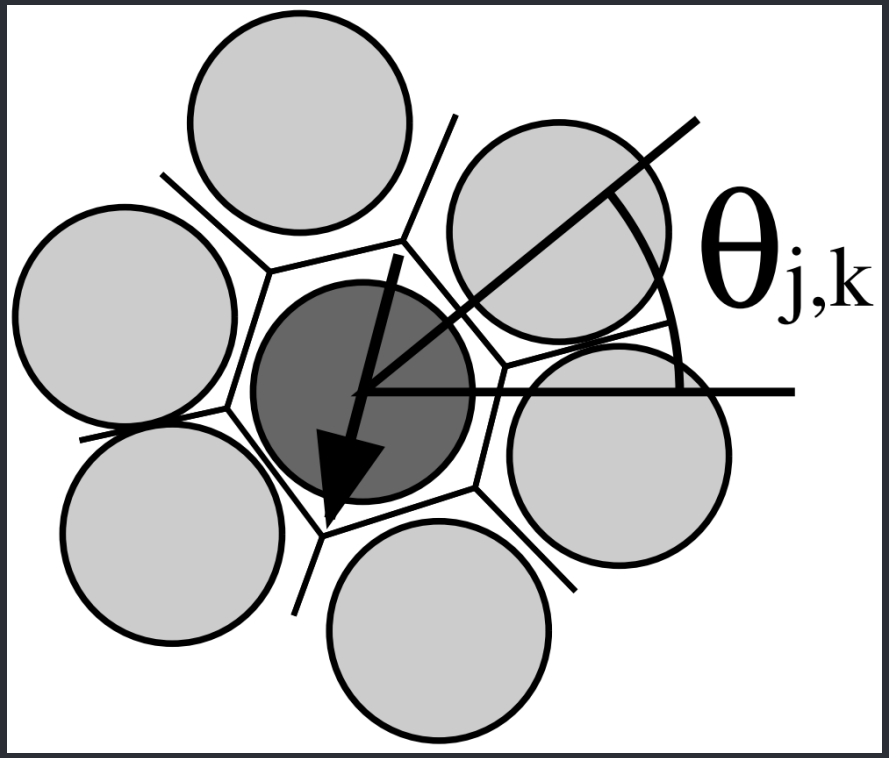
- A orientação global $\Psi$ é a média da orientação local $\psi_k$:

$$\Psi = \left| \frac{1}{N} \sum_{k=1}^{N} \psi_k \right|,$$

$$\psi_k(x_k, y_k) = \frac{1}{6} \sum_{j=1}^{6} e^{6i\theta_{j,k}},$$

onde $(x_k, y_k)$ é a posição do disco $k$, $i = \sqrt{-1}$ e $\theta_{kl}$ é o ângulo da reta que une os discos $k$ e $l$ com o eixo $x$.

# Orientação local



# Constante de tempo

- Estudamos a dependência temporal: $\Psi = \Psi_0 e^{-t/\tau}$ onde $\tau$ é a constante de tempo [7].
- Para determinar $\tau$: linearização de $\Psi$ de forma que $y = a + bx$ onde

$$y = \ln \Psi, \qquad a = \ln \Psi_0, \qquad b = -\frac{1}{\tau}, \qquad x = \ln t.$$

- Ajuste linear de $y$ vs. $x$: $\tau = -1/b$, onde $b$ é o coeficiente angular.

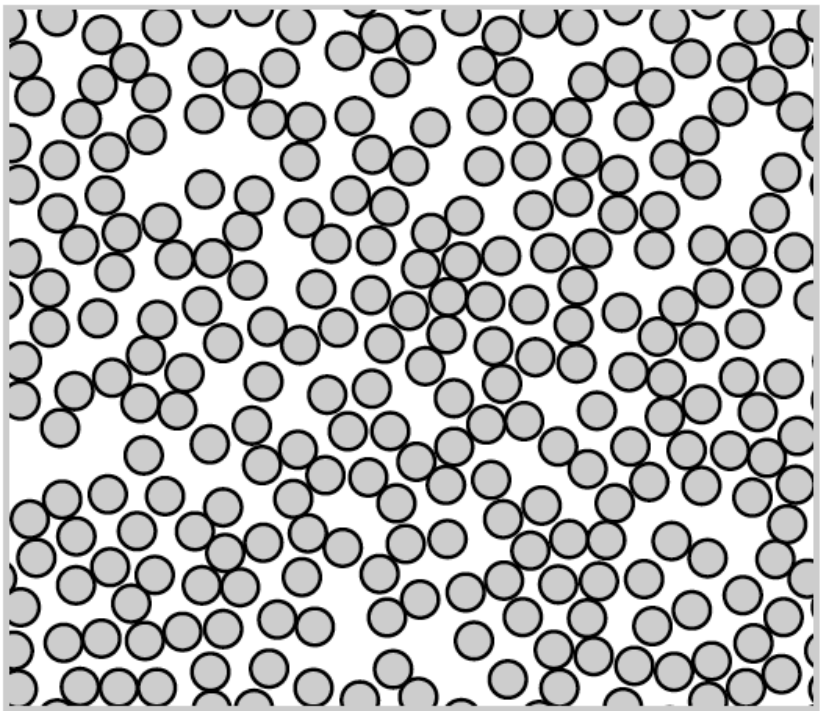
# Objetivos

- Nosso objetivo: avaliar o comportamento da constante de tempo $\tau$ em função do número de discos.

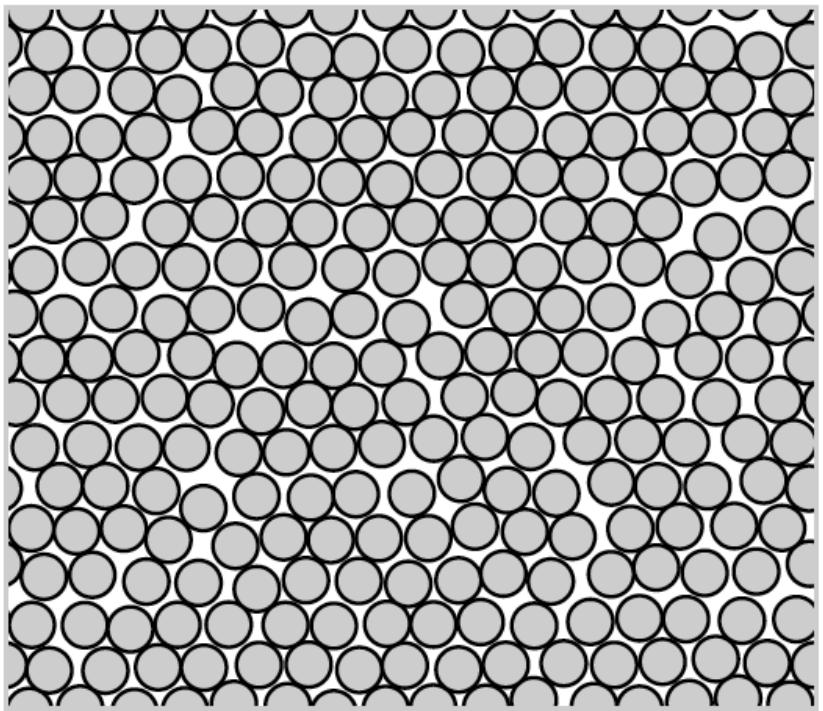
# Objetivos

- Nosso objetivo: avaliar o comportamento da constante de tempo $\tau$ em função do número de discos.
- Inédito para esse sistema.
- Há várias sugestões teóricas para esse comportamento mas nenhuma numérica.

# Discos rígidos em 2D



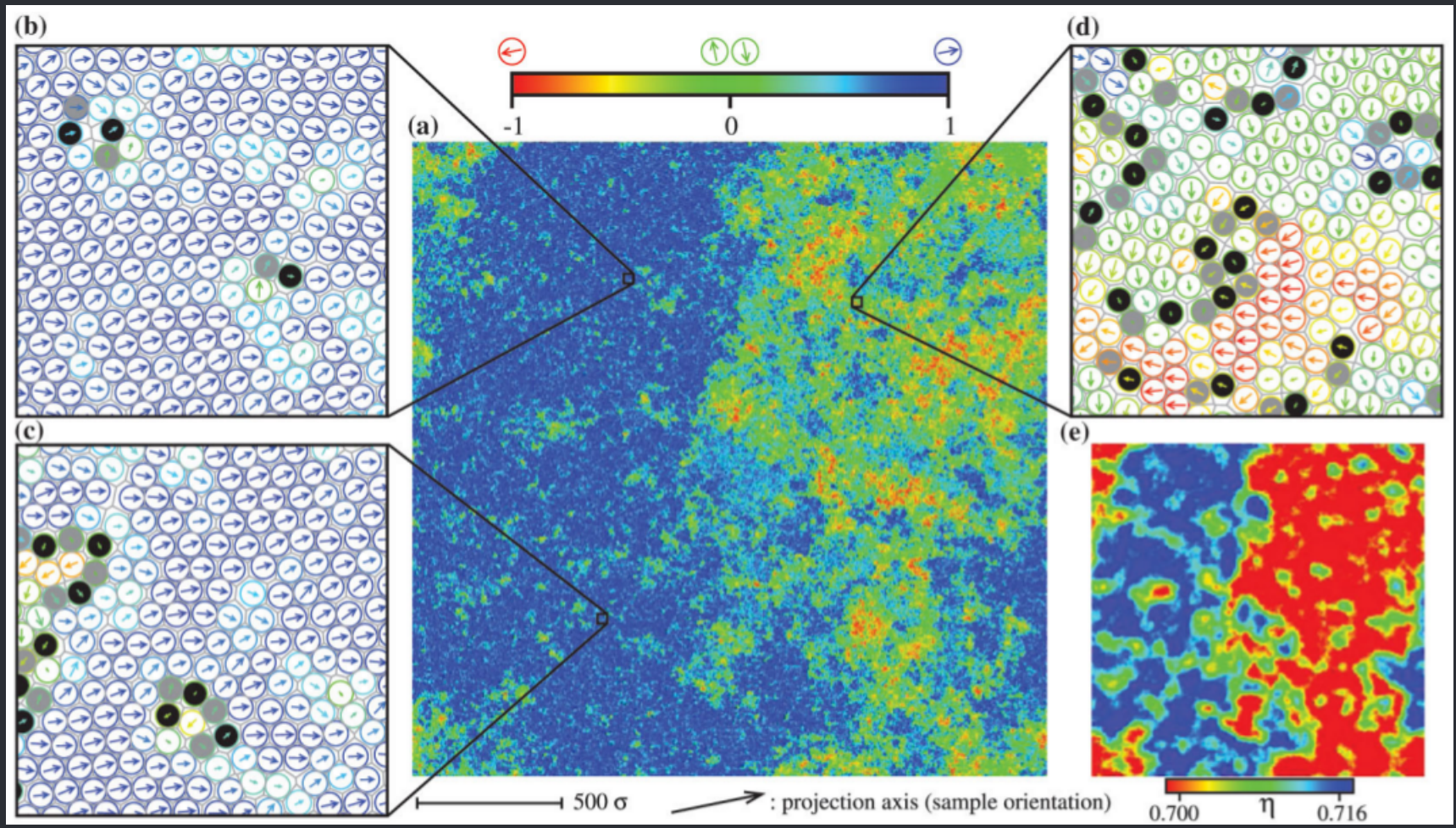
$\eta = 0.48$



$\eta = 0.72$

Fonte: [7].

# Fase hexática




Fonte: [7]

# Seção 3

1. Introdução
2. Modelo Teórico
3. Implementação computacional
4. Resultados
   - 4.1 Monte Carlo Cadeia de Eventos
   - 4.2 HOOMD-blue HPMC
5. Conclusões
6. Extras

# Evolução temporal

- Dinâmica temporal: evolução temporal do sistema.
- O tempo é medido em iterações. Em cada iteração calculamos $\Psi$.
- Estado final no equilíbrio ($t > t_e$): fase macroscópica observada.
- Nosso trabalho: estudamos a dependência temporal de $\Psi$ antes do equilíbrio ($t < t_e$).

# Algoritmo 1

- Monte Carlo com Cadeia de Eventos.
- Em cada iteração vários discos se movem em $+x$ ou $+y$, um por vez, até a distância total percorrida seja $\ell$.

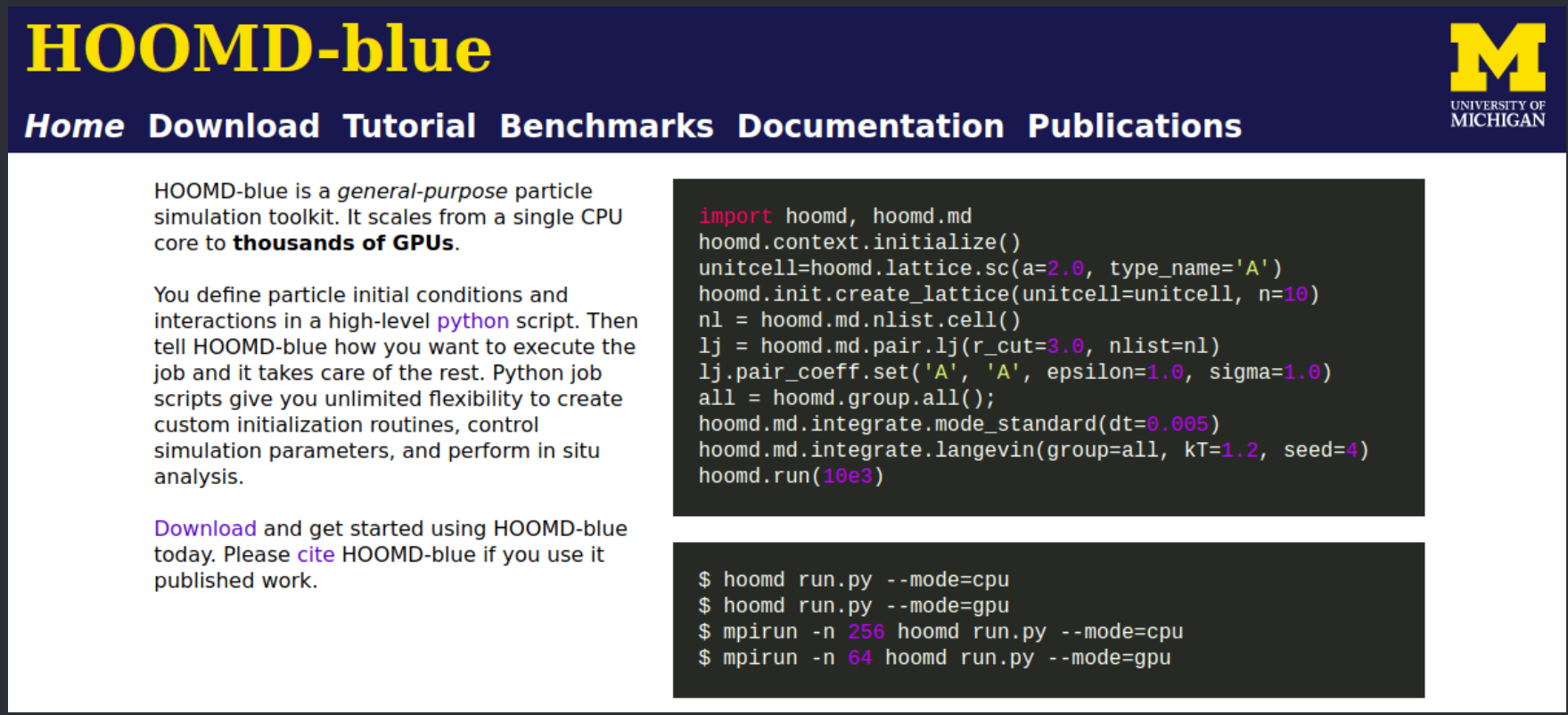
# Algoritmo 1

- Monte Carlo com Cadeia de Eventos.
- Em cada iteração vários discos se movem em $+x$ ou $+y$, um por vez, até a distância total percorrida seja $\ell$.
- Construimos uma implementação do zero em C$++$.
- Cálculo de distâncias entre os discos: é a parte mais demorada do algoritmo.
- Lista de células: mais rápido que força bruta.

# Algoritmo 2

- Monte Carlo com Cadeia de Markov: módulo HPMC [2] do pacote HOOMD-Blue [3, 5, 4].
- Semelhante ao MCCE porém com apenas um disco se movendo por iteração.
- Versão paralelizada com possibilidade de execução na GPU também.
- Facilidade: funções implementadas em Python (Miniconda).

# HOOMD-Blue




http://glotzerlab.engin.umich.edu/hoomd-blue/index.html

# Seção 4

1. Introdução
2. Modelo Teórico
3. Implementação computacional
4. Resultados
   - 4.1 Monte Carlo Cadeia de Eventos
   - 4.2 HOOMD-blue HPMC
5. Conclusões
6. Extras

# Procedimento

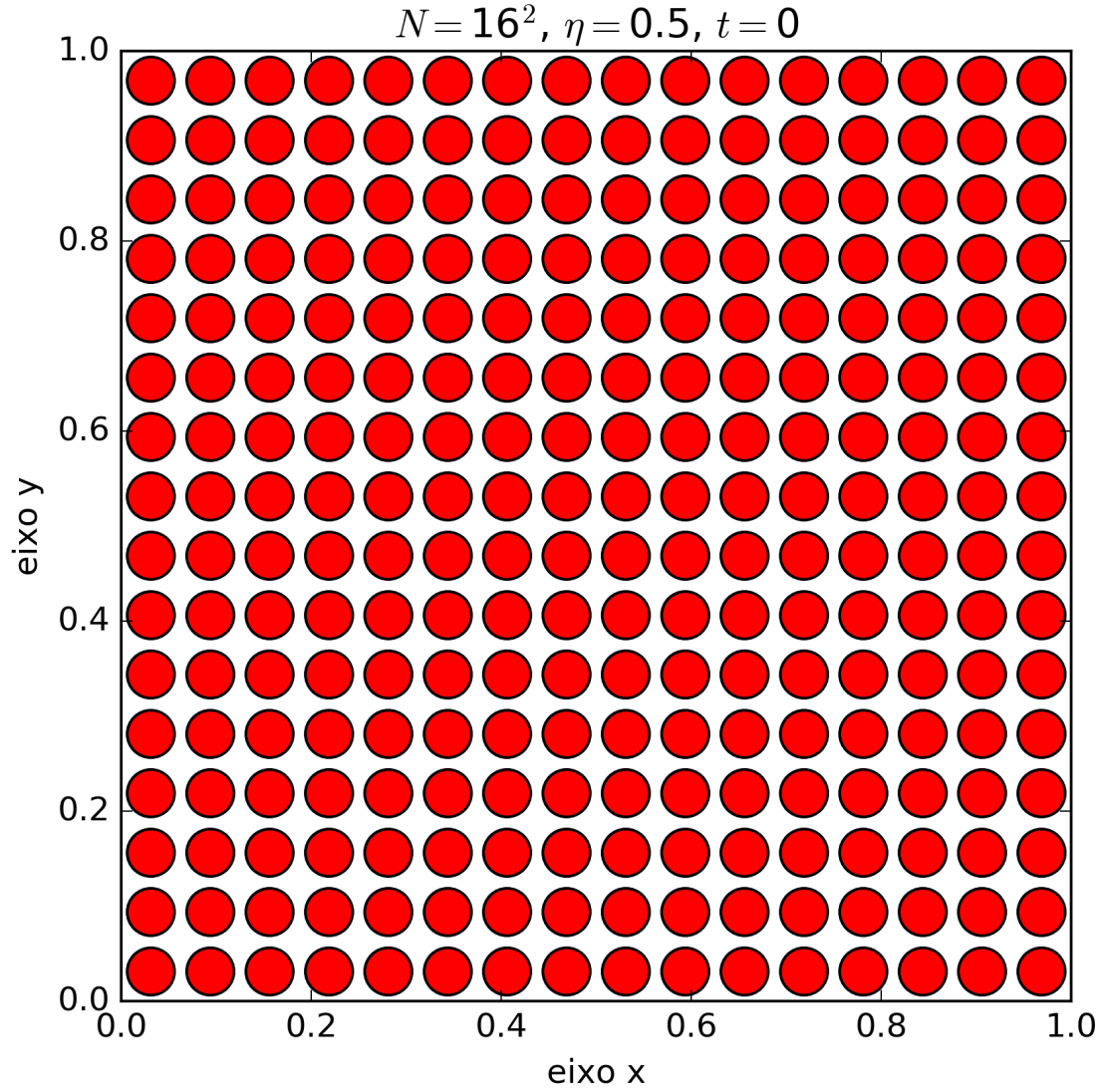
1. Cálculo de $\Psi(t)$ em função do tempo para $t = 0, 1, 2, \ldots, Q$.
2. Gráfico loglog de $\Psi(t)$.
3. Ajuste linear de $\ln \Psi(t)$ vs $\ln t$ para obtenção de $\tau$.
4. Números de discos calculados
   $N = 64^2, 128^2, 256^2, 400^2, 512^2, 660^2, 810^2, 960^2$ e $1024^2$.
5. Densidades calculadas $\eta = 0.695, 0.705$ e $0.718$.

# Algoritmo MCCE

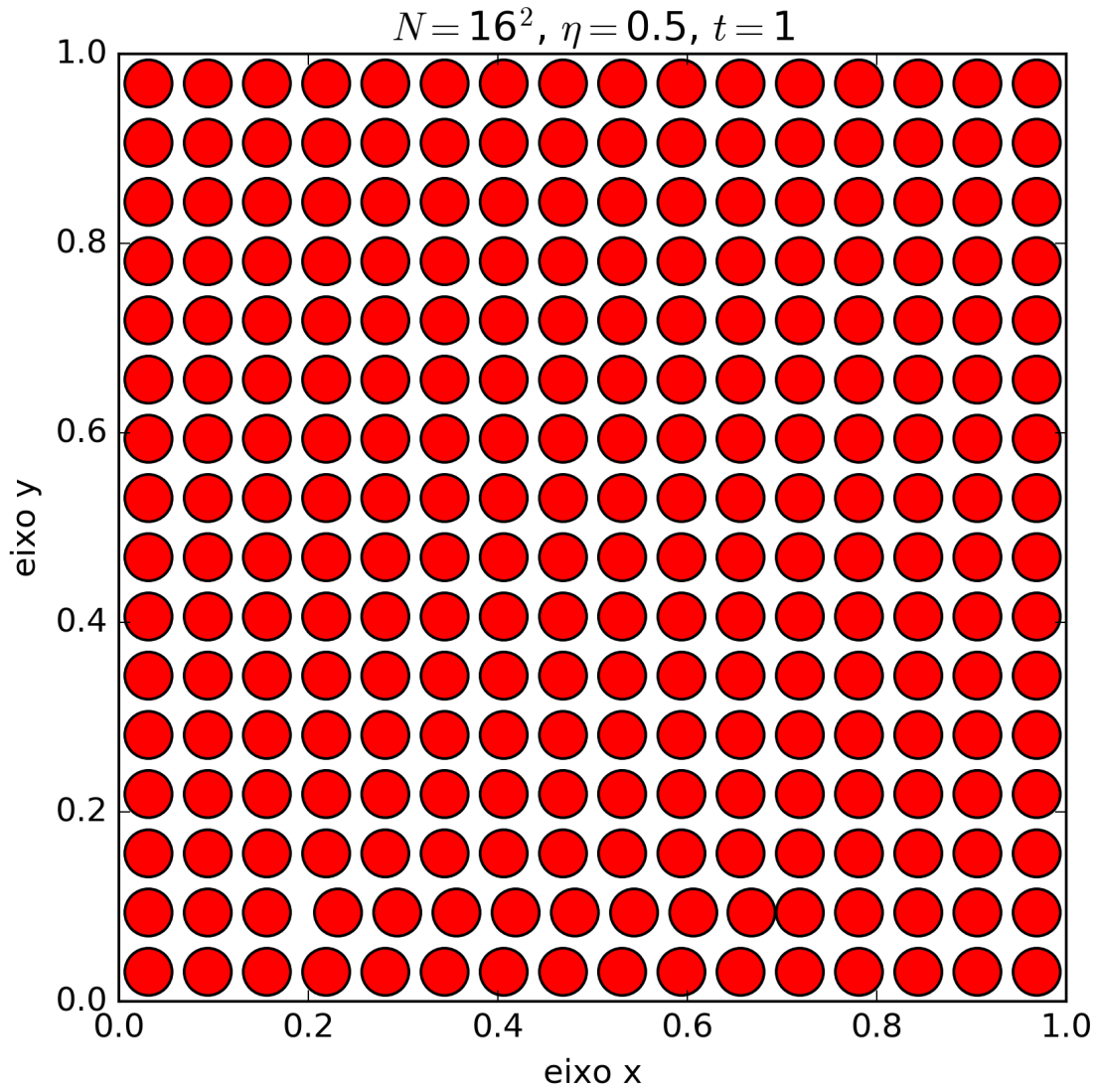
**Algorithm 1** Evolução temporal com MCCE

```
1:  procedure MCCE
2:      Input: N, ℓ, η, S, Q, L_x
3:      listcell ← create_list(N)
4:      for k ∈ S do
5:          LxLy ← initial(N, η, L_x)
6:          add_disks(N, listcell)
7:          seed ← k
8:          for t ∈ Q do
9:              LxLy ← newL(LxLy, N, η, ℓ, seed, listcell)
10:             Psi(t) ← abs_Psi(η, LxLy, listcell)
11:     file ← time_Psi.csv
12:     write(t,Psi,file)
```

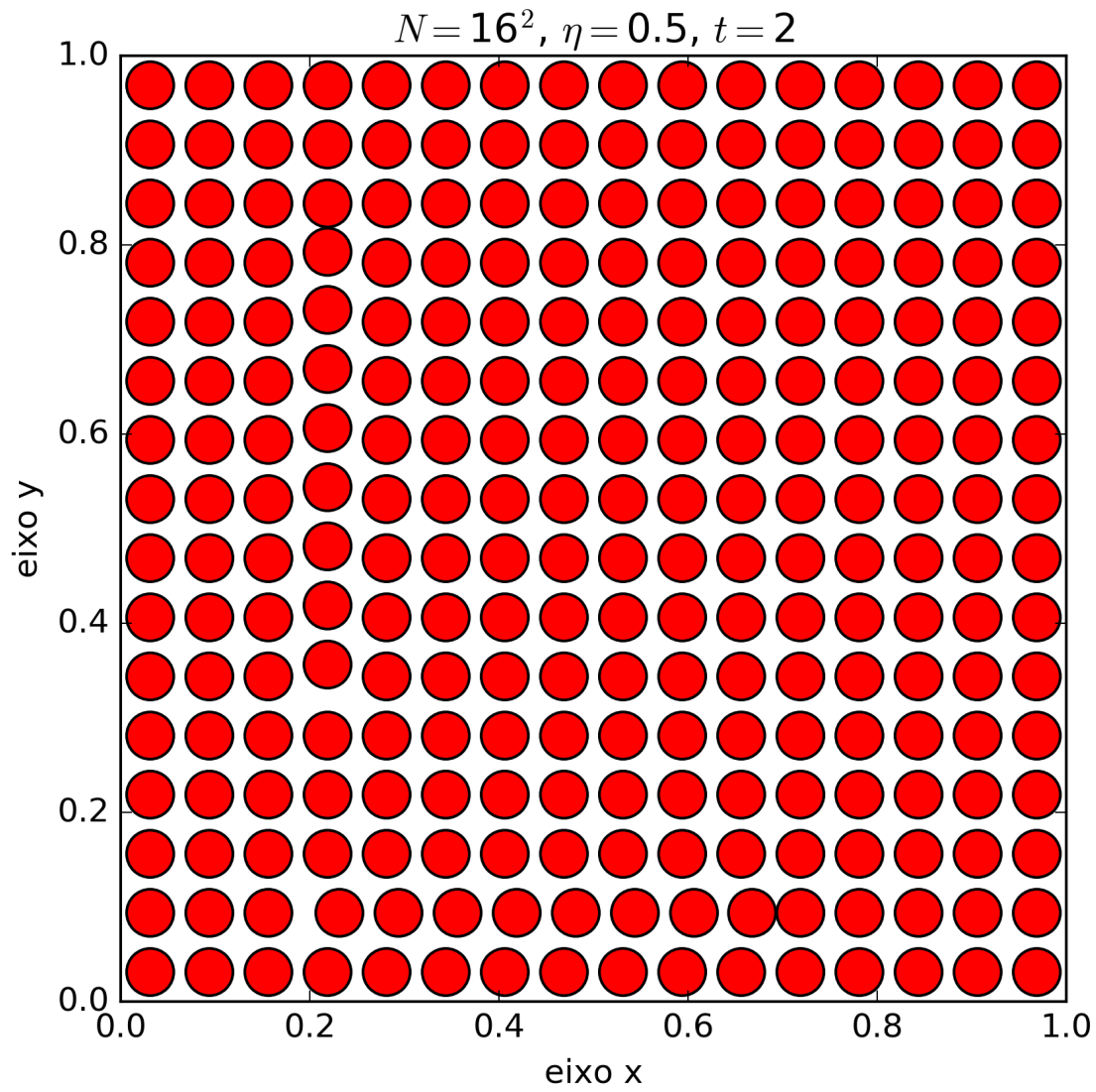
# Evolução Temporal



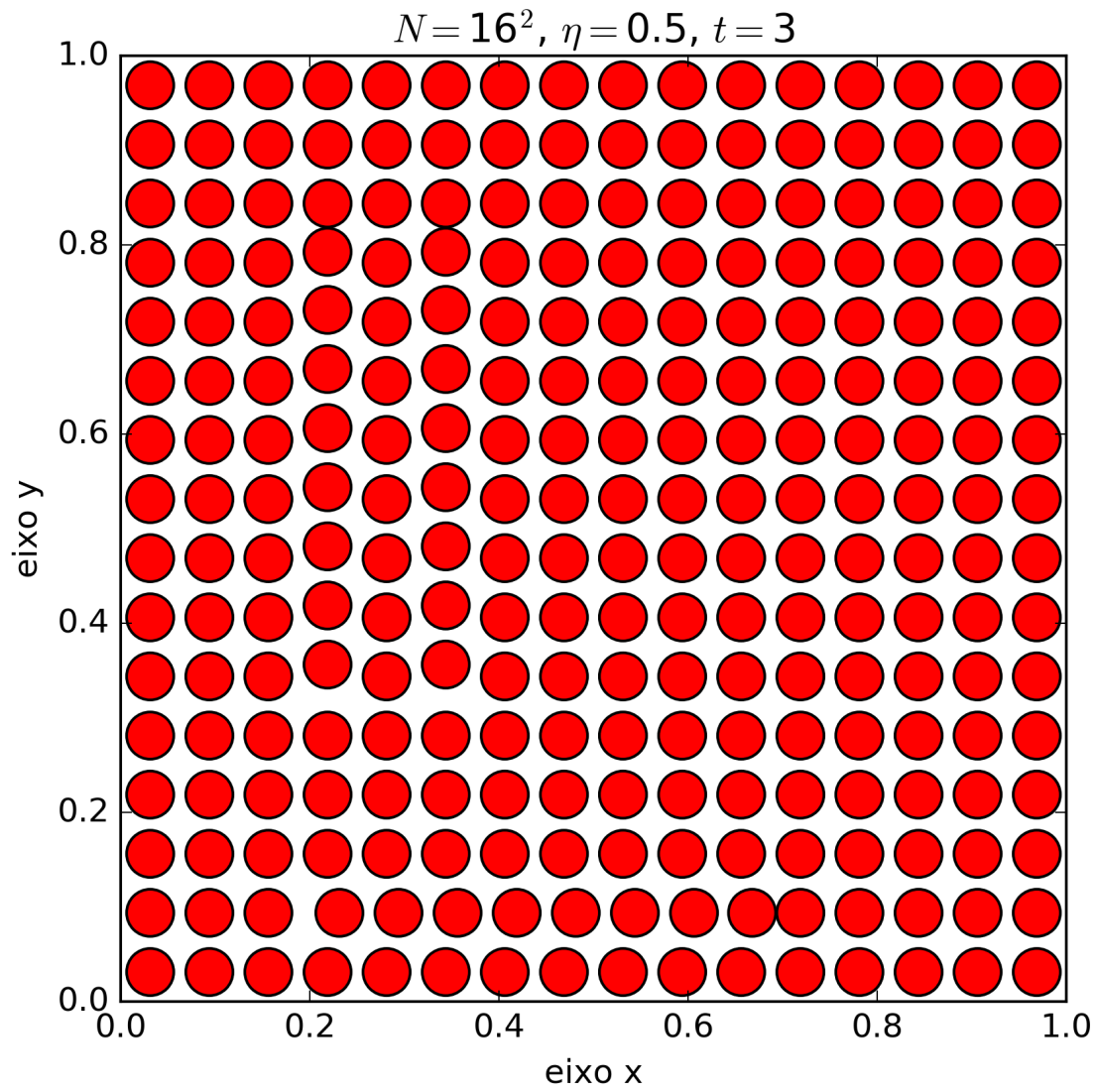
# Evolução Temporal



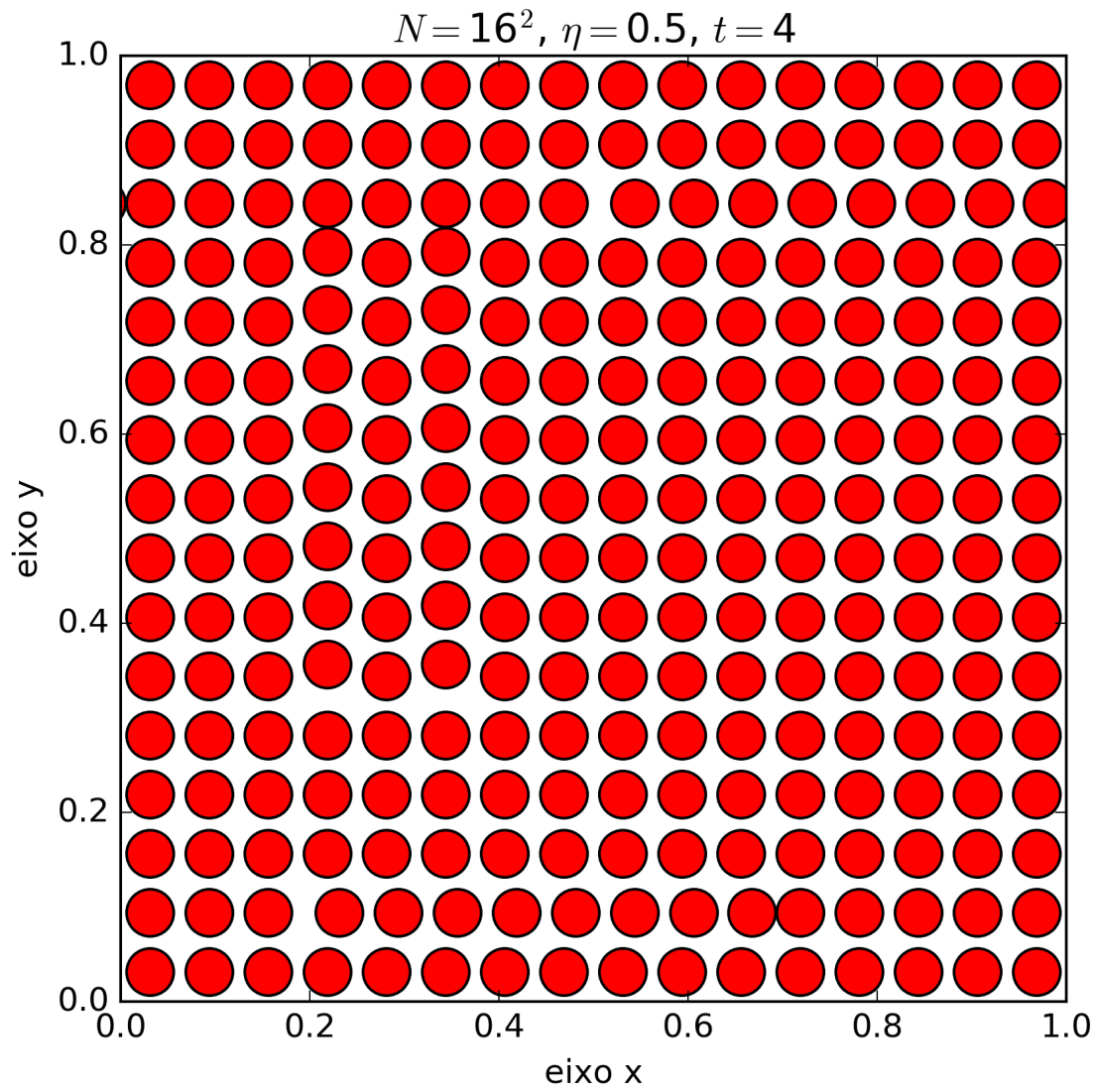
# Evolução Temporal



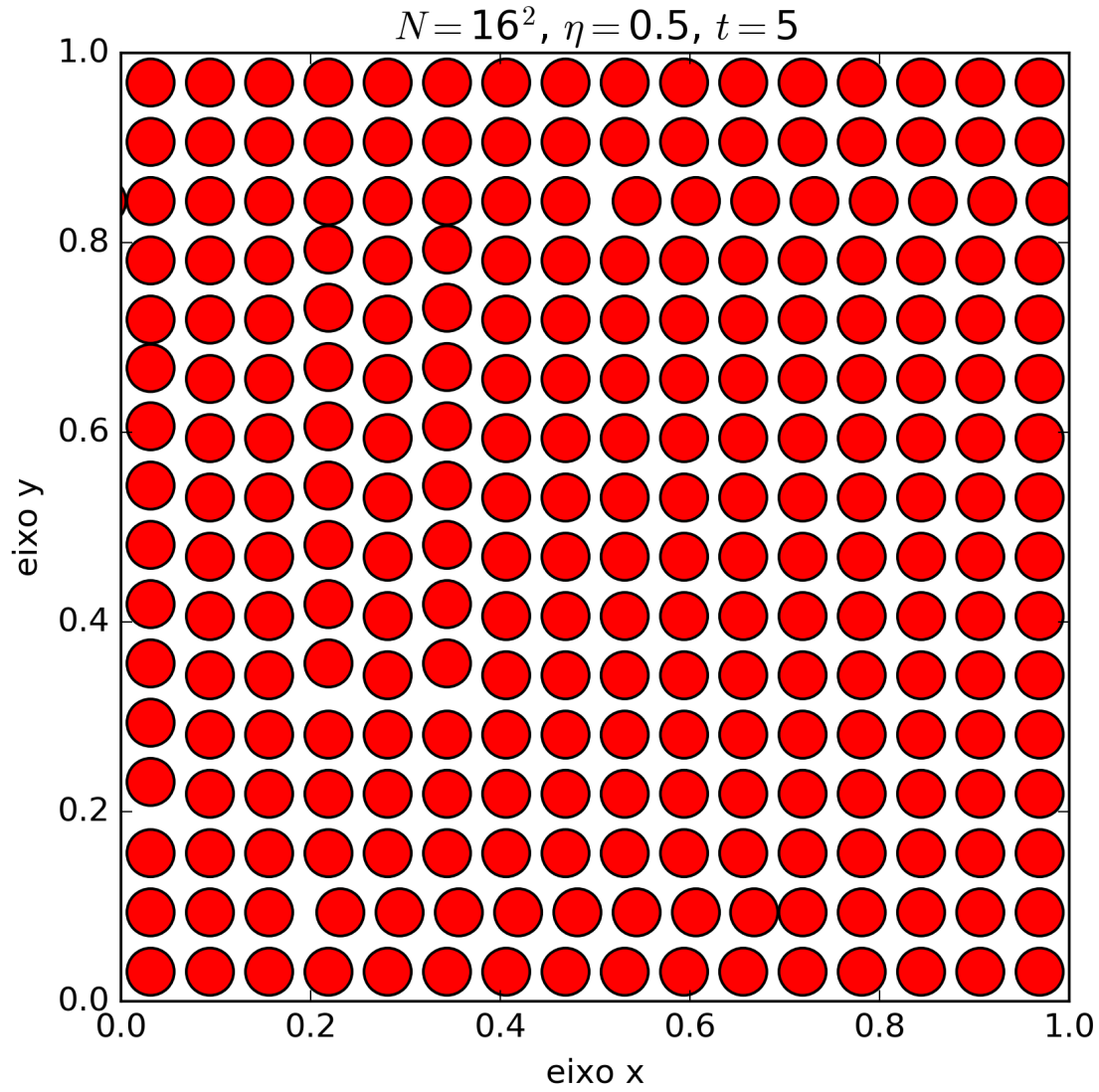
# Evolução Temporal



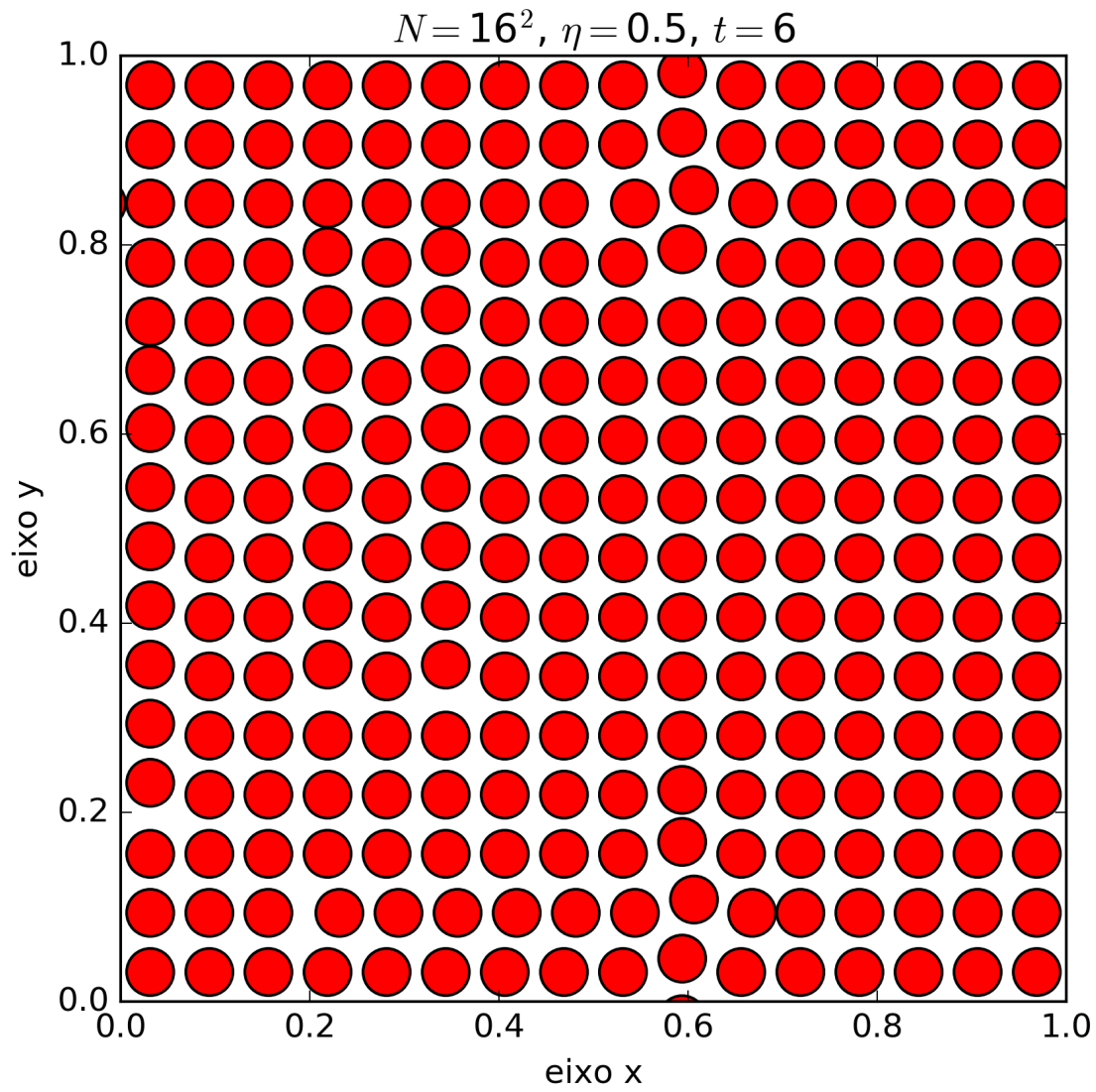
# Evolução Temporal



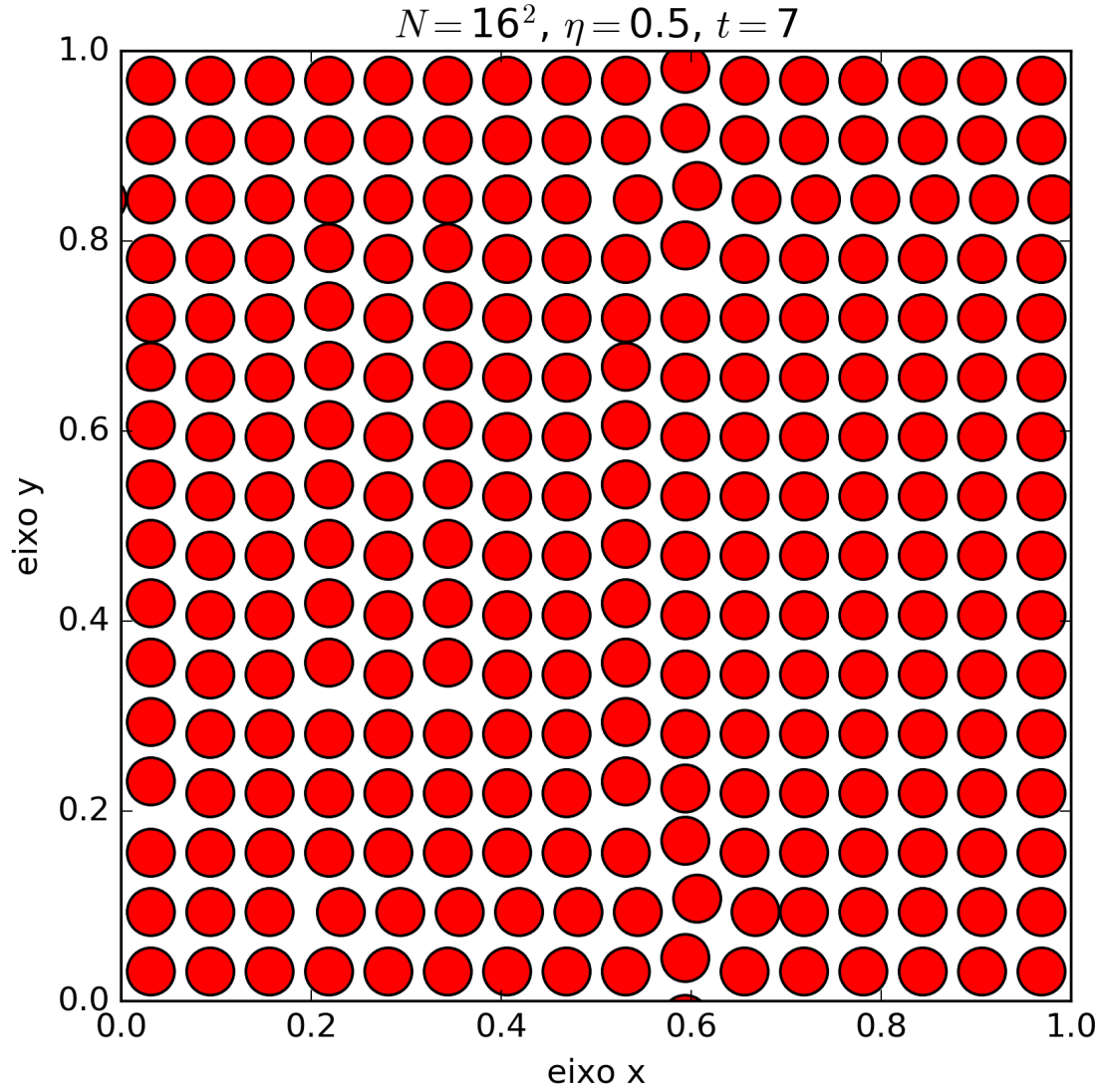
# Evolução Temporal



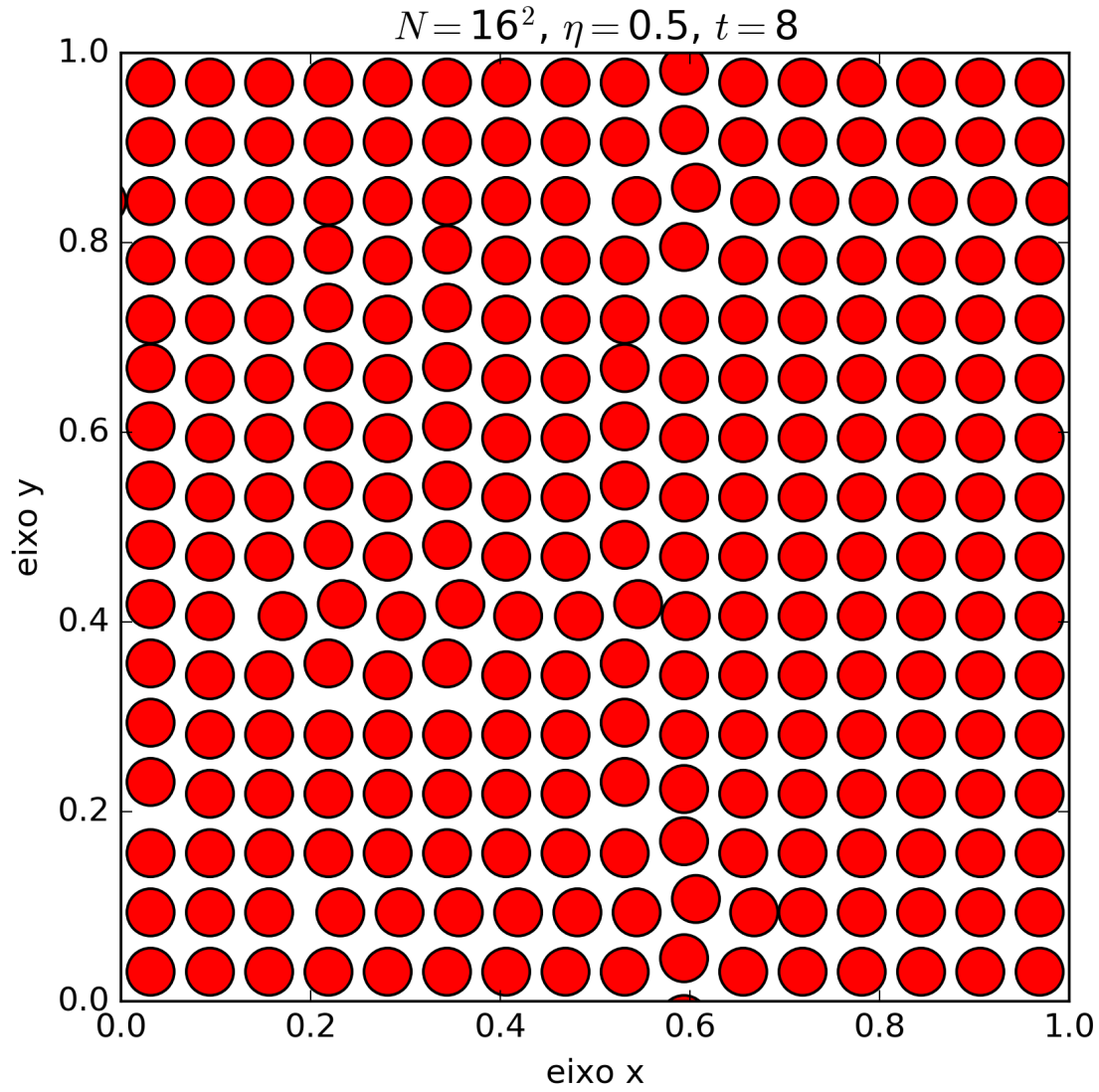
# Evolução Temporal



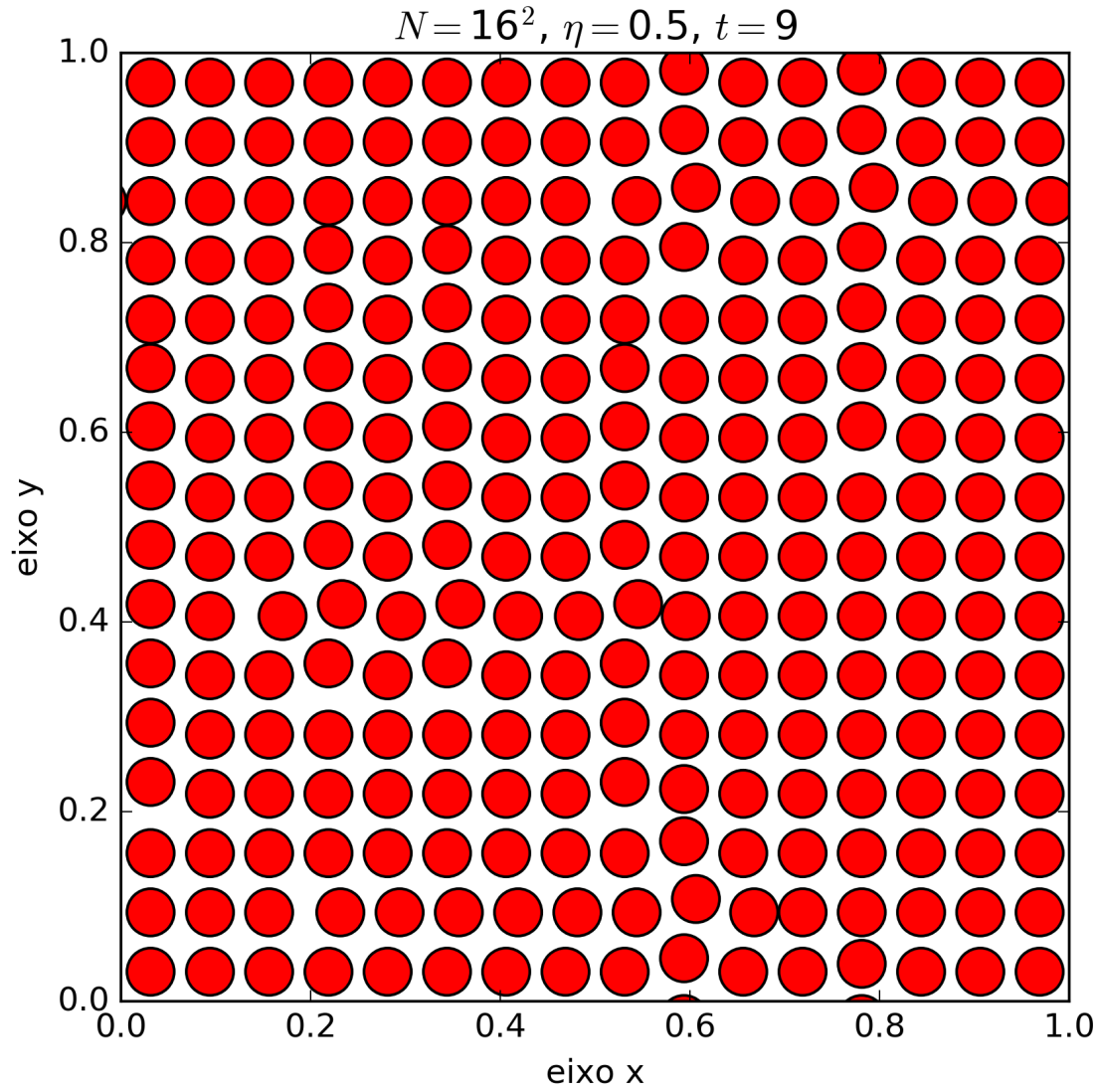
# Evolução Temporal



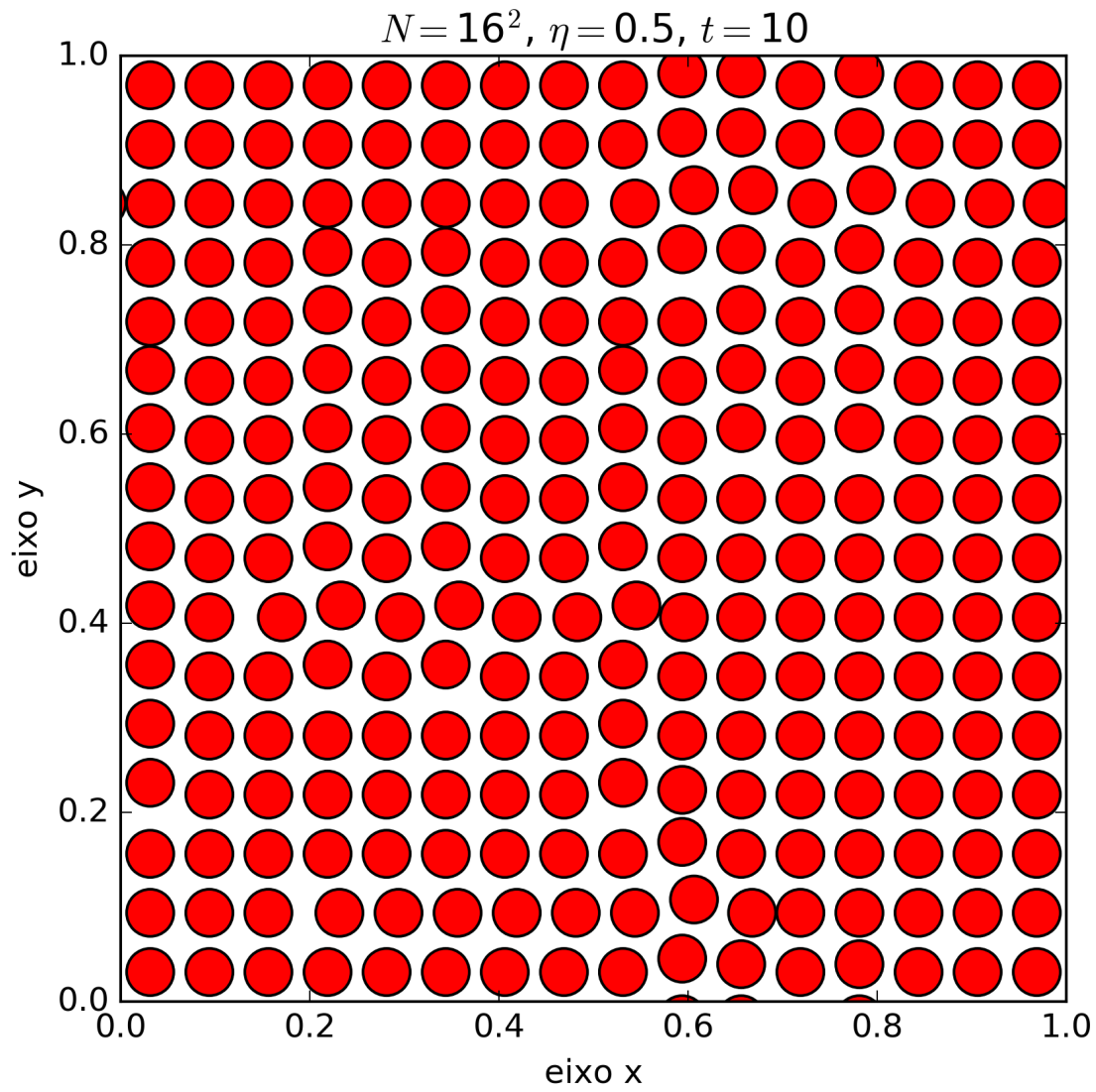
# Evolução Temporal



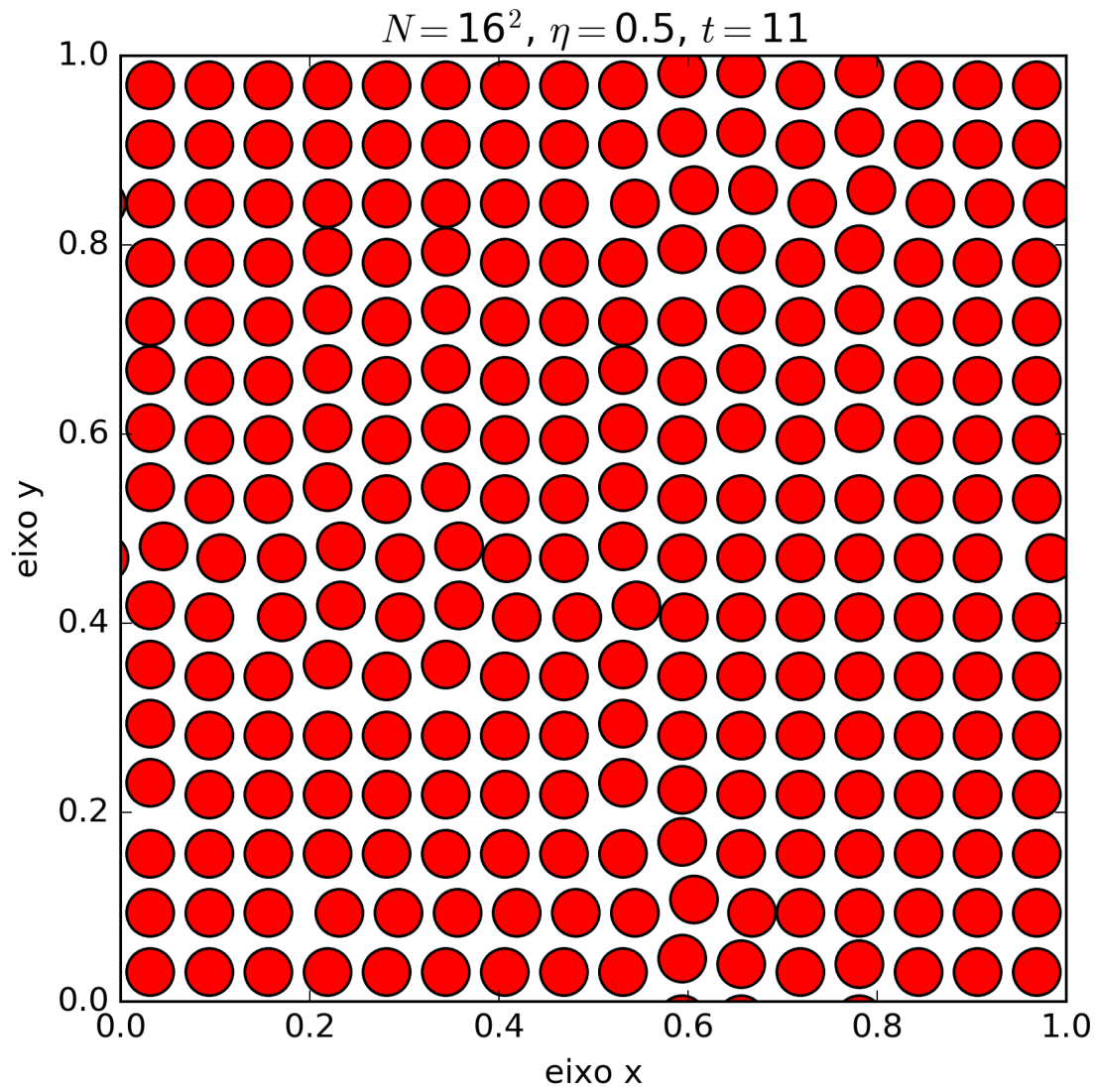
# Evolução Temporal



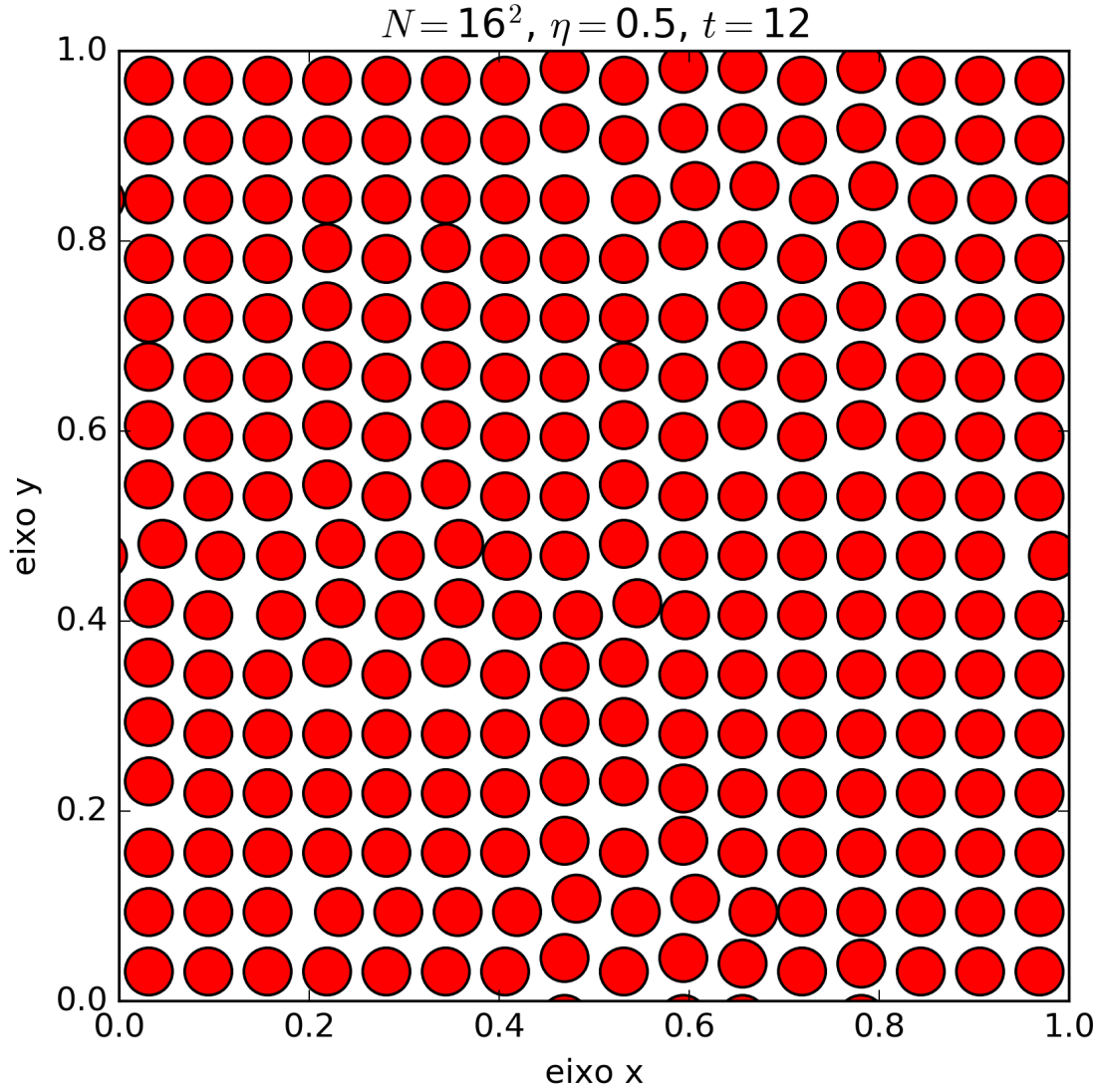
# Evolução Temporal

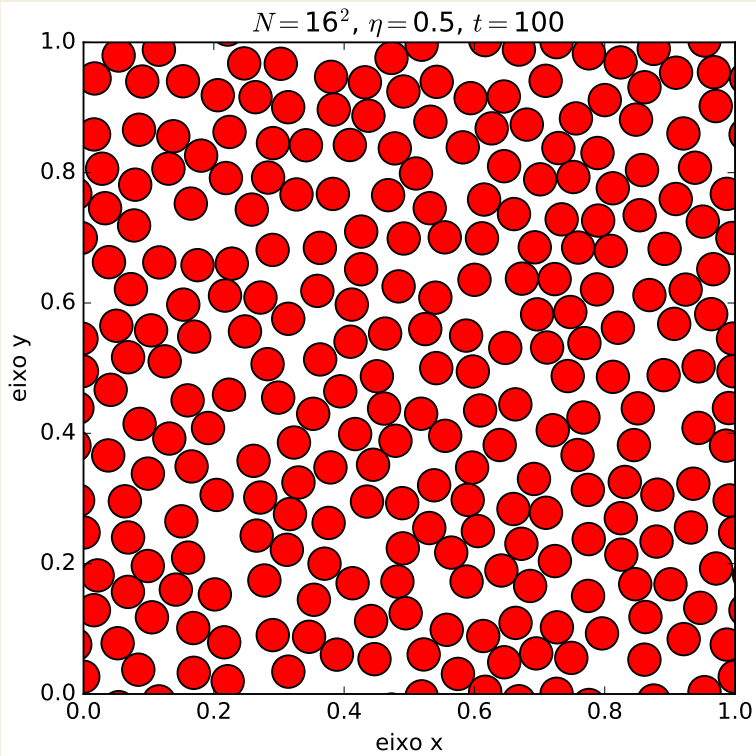


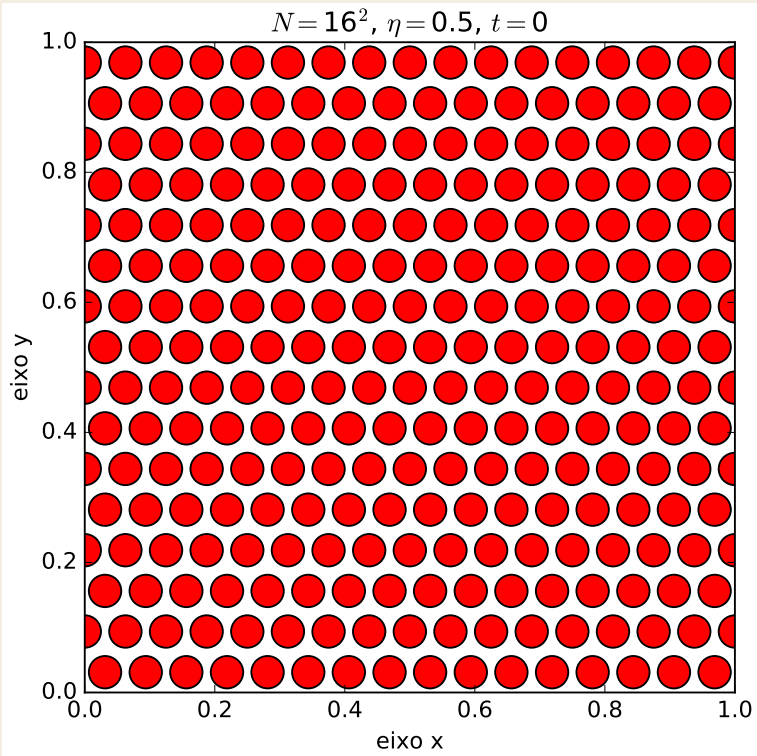
# Evolução Temporal



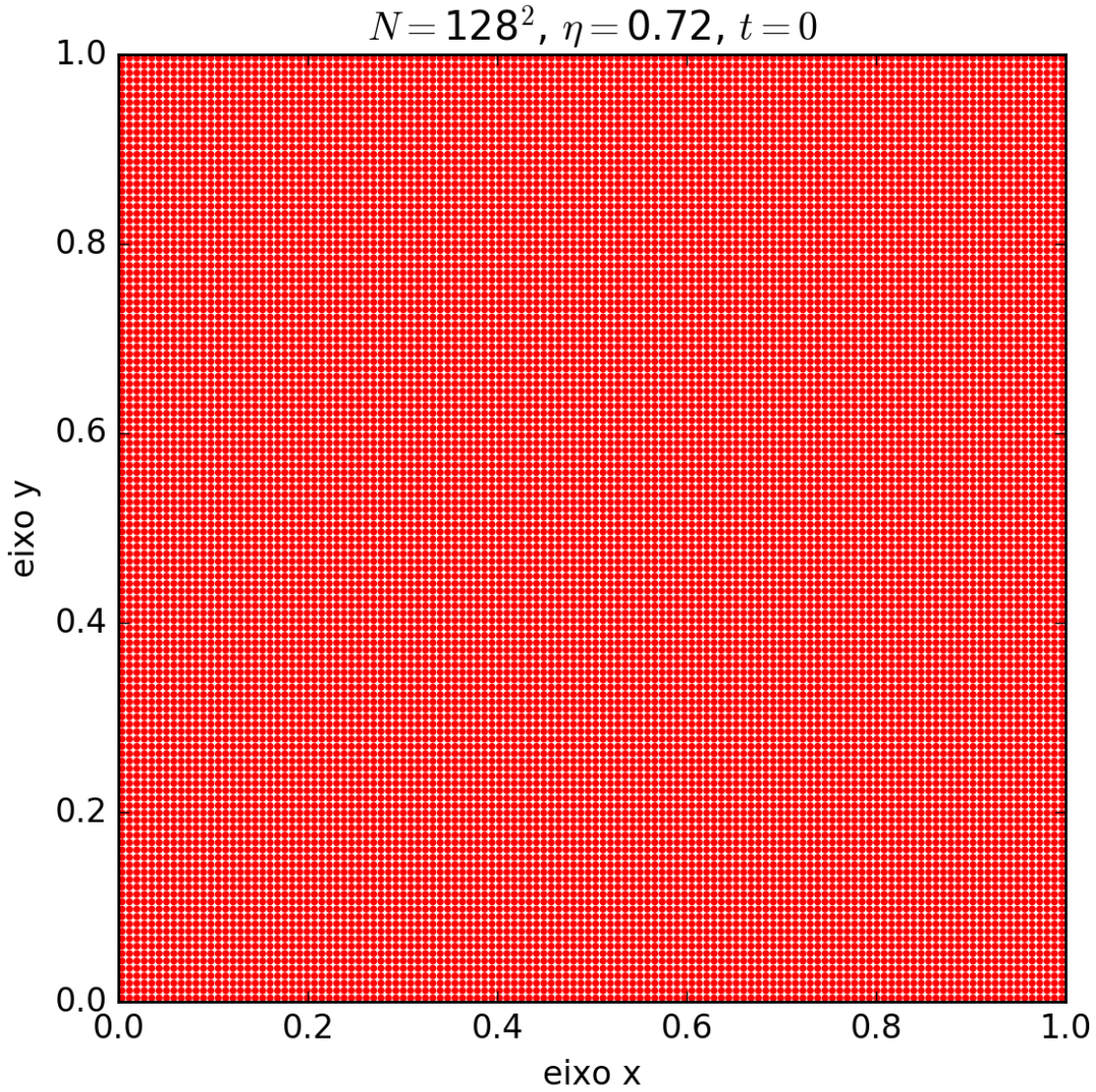
# Evolução Temporal



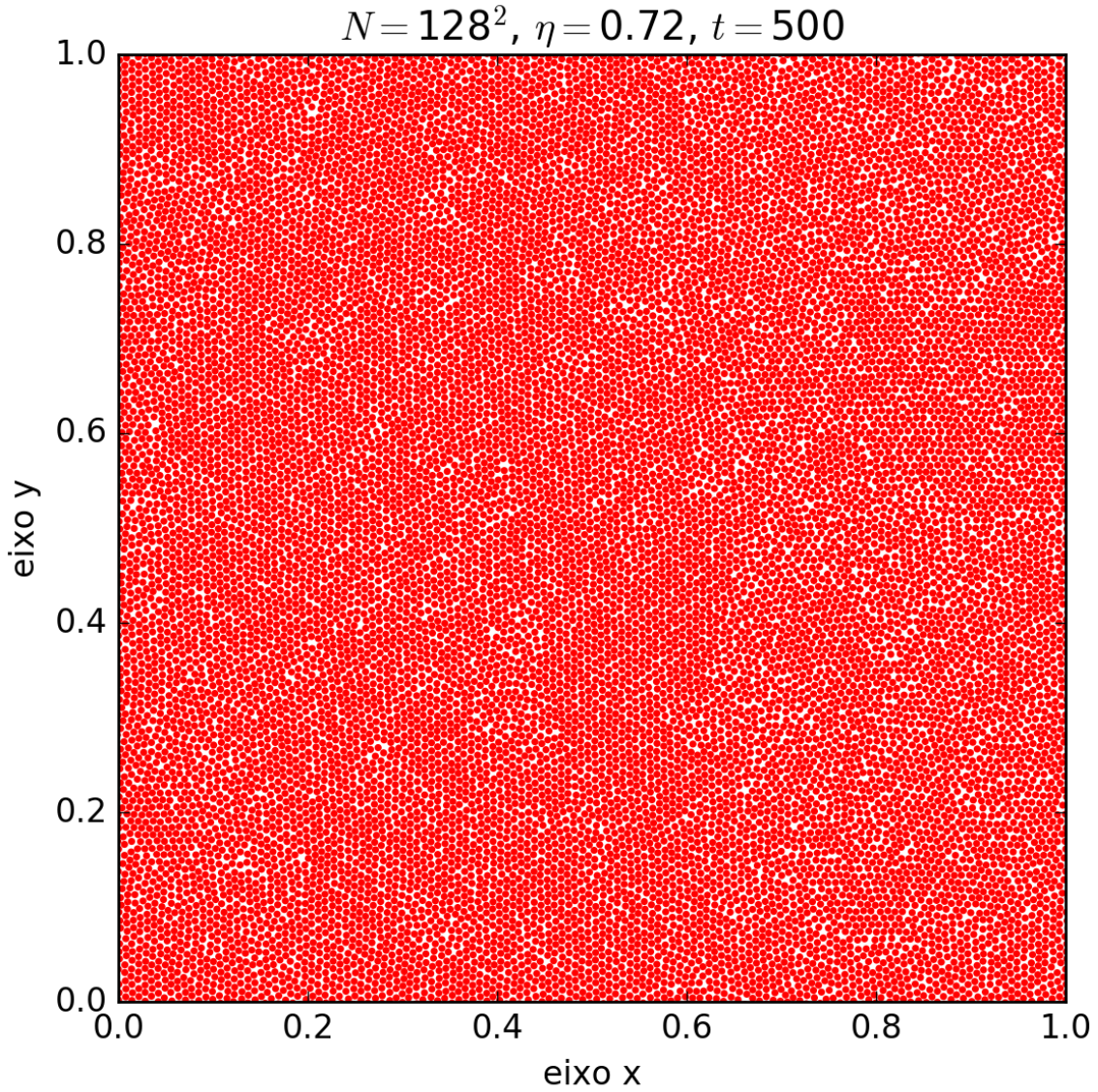

$N = 16^2$, $\eta = 0.5$, $t = 100$
eixo y
eixo x
0.0
0.2
0.4
0.6
0.8
1.0


$N = 16^2$, $\eta = 0.5$, $t = 0$
eixo y
eixo x
0.0
0.2
0.4
0.6
0.8
1.0

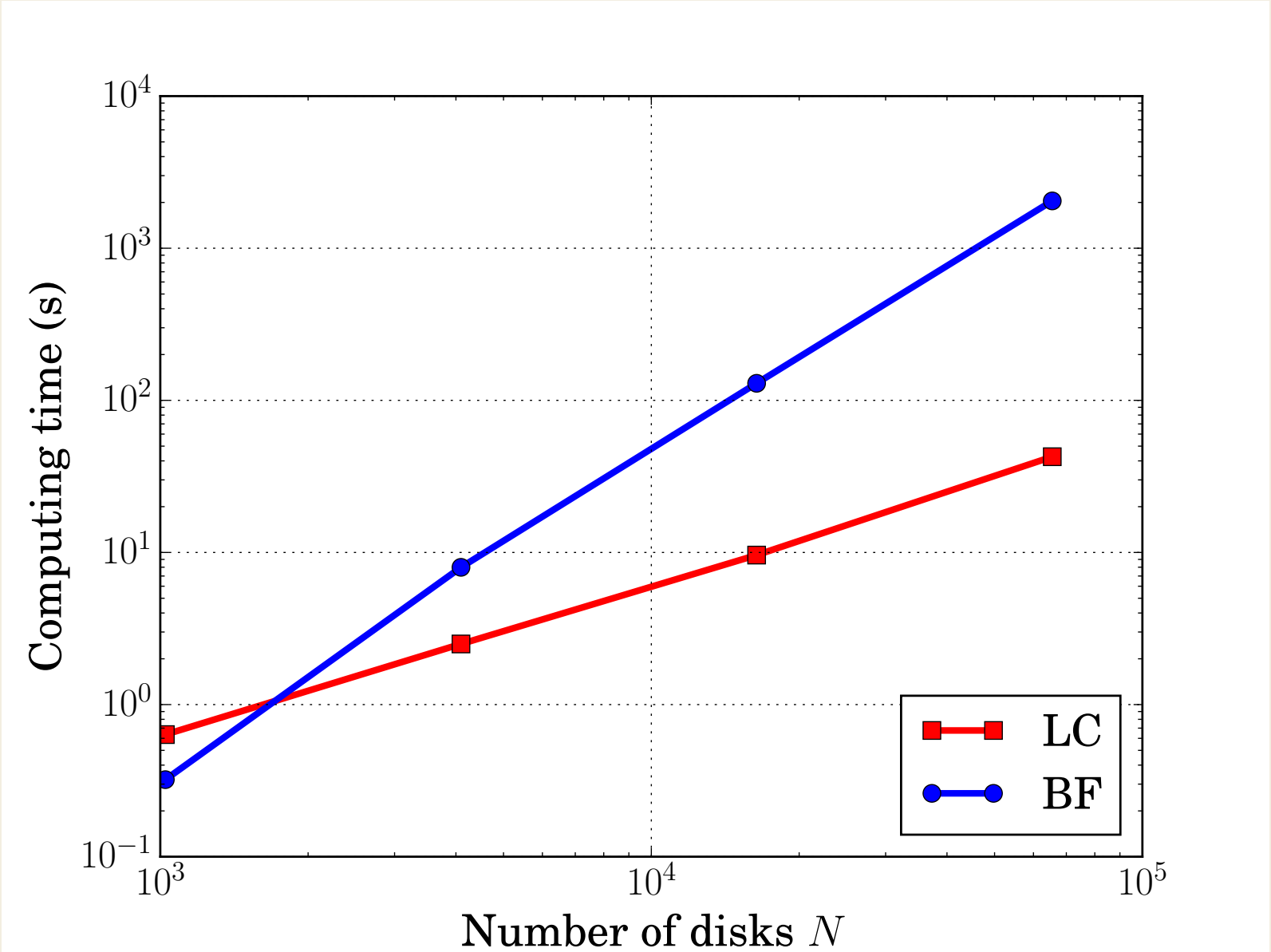
# $N = 128^2 = 16384$



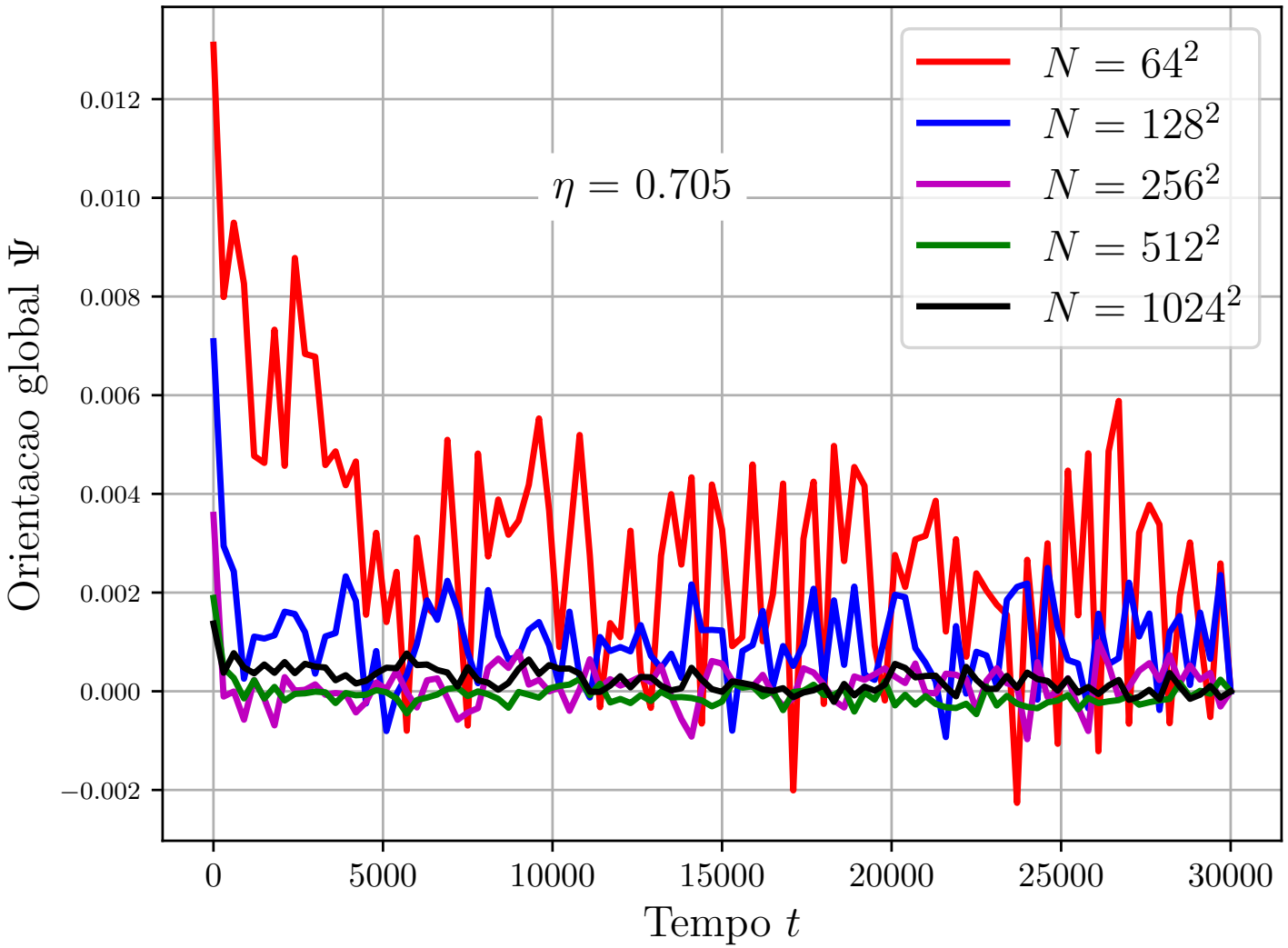
# $N = 128^2 = 16384$



# Força bruta vs. Lista de células



# Monte Carlo Cadeia de Eventos



# Conclusões

1. Dinâmica temporal usando Monte Carlo com Cadeia de Eventos: bem sucedida.
2. Cálculo de $\Psi$: alguns ajustes precisam ser feitos.

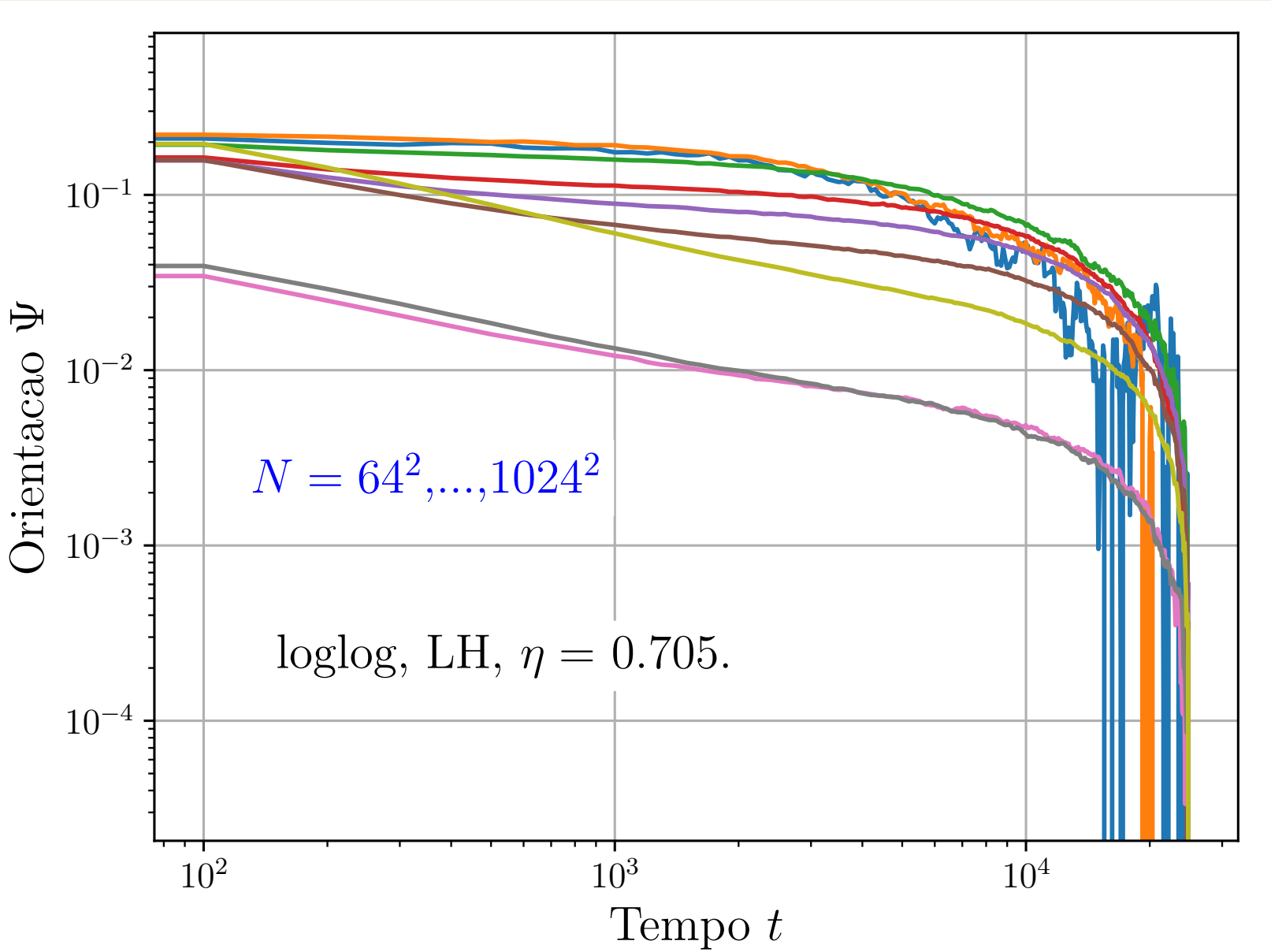
# Conclusões

1. Dinâmica temporal usando Monte Carlo com Cadeia de Eventos: bem sucedida.
2. Cálculo de $\Psi$: alguns ajustes precisam ser feitos.
3. Porém, o desenvolvimento deste algoritmo nos fez compreender os detalhes da dinâmica temporal e sua implementação.
4. Isso nos permitiu entender e usar adequadamente o módulo HPMC do pacote HOOMD-blue.

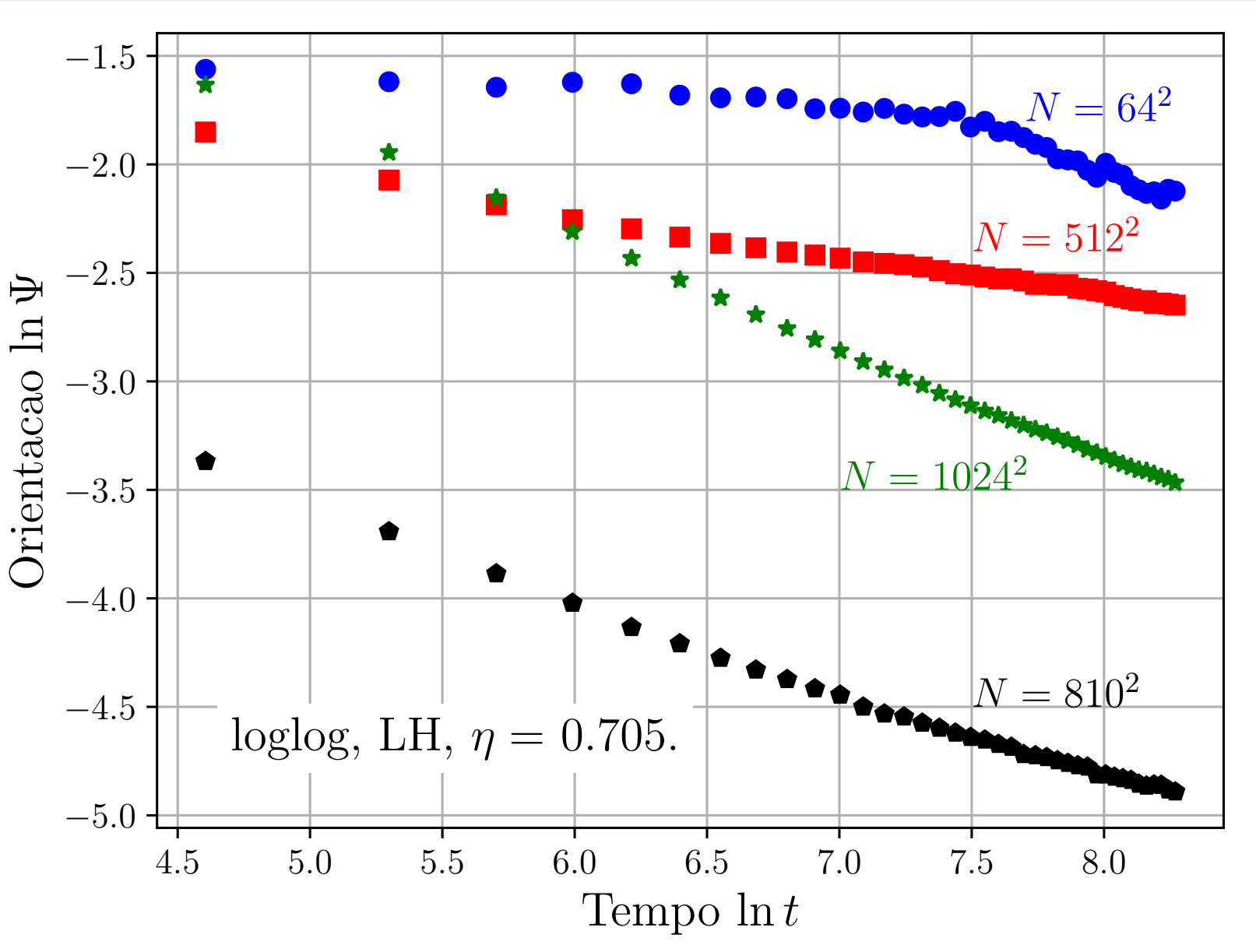
# HPMC HOOMD-Blue

```python
import hoomd, freud
import hoomd.hpmc
from freud.order import HexOrderParameter

class PsiAnalyzer:

    def __init__(self, system):
        self.system = system
        self.psi_soma_array = np.zeros(Q, dtype=np.complex64)
        self.psi2_soma_array = np.zeros(Q, dtype=np.complex64)
        self.order_parameter = HexOrderParameter(2.8, 6.0, 6)
        #self.total_particles = system.take_snapshot().particles.N
        self.initial_configuration = system.take_snapshot();

    def __call__(self, step):
        snap = self.system.take_snapshot()
        pos = snap.particles.position
        box = freud.box.Box(snap.box.Lx, snap.box.Ly, is2D=True)
        self.order_parameter.compute(box, pos)
        self.psi_soma_array[step % Q] += np.sum(self.order_parameter.getPsi())
        self.psi2_soma_array[step % Q] += np.sum(self.order_parameter.getPsi()**2)

    def get_soma_psi_abs(self):  #Metodos
        return  np.abs(self.psi_soma_array)

    def get_soma_psi2_abs(self):
        return  np.abs(self.psi2_soma_array)
```

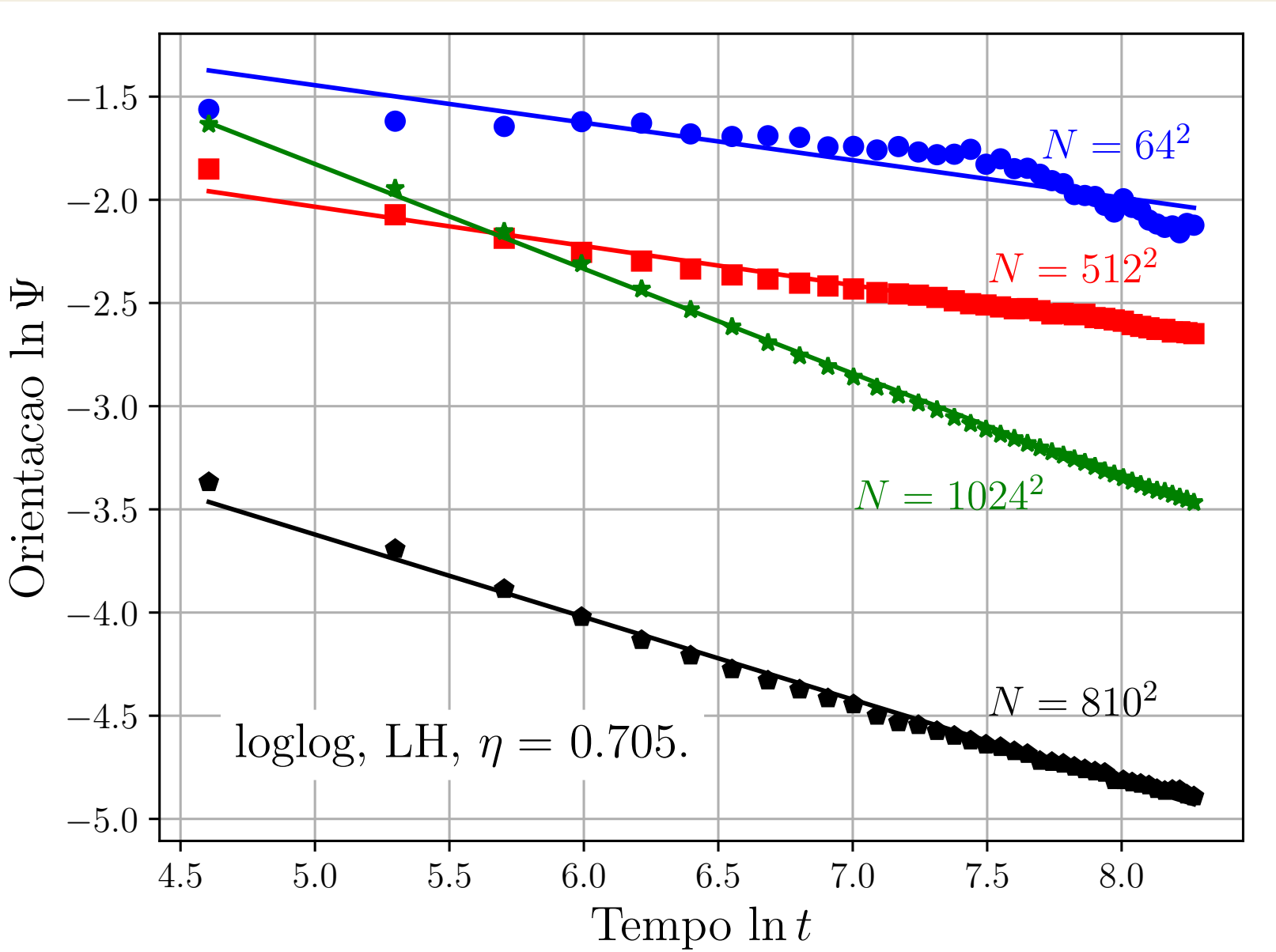
# HPMC HOOMD-Blue



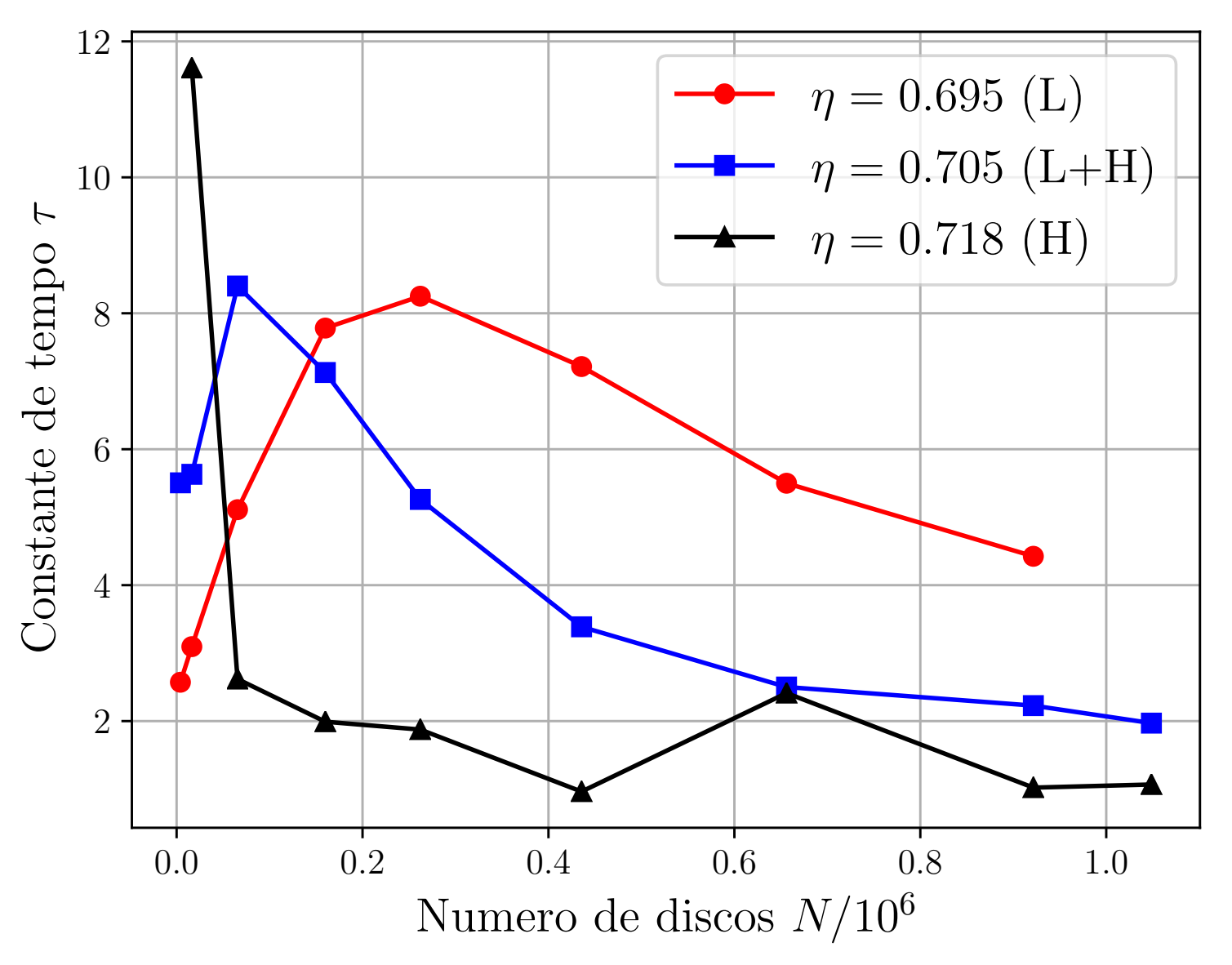
# Ajuste exponencial



# Ajuste exponencial



# Costante de tempo



# Seção 5

1. Introdução
2. Modelo Teórico
3. Implementação computacional
4. Resultados
   - 4.1 Monte Carlo Cadeia de Eventos
   - 4.2 HOOMD-blue HPMC
5. Conclusões
6. Extras

# Conclusões

1. Usando HPMC do pacote HOOMD-Blue, calculamos a constante de decaimento $\tau$ para vários números de discos $N$ em diferentes fases.
2. Novos cálculos (longos e demorados) estão sendo feitos para outras fases.

# Conclusões

1. Usando HPMC do pacote HOOMD-Blue, calculamos a constante de decaimento $\tau$ para vários números de discos $N$ em diferentes fases.
2. Novos cálculos (longos e demorados) estão sendo feitos para outras fases.
3. A física precisa cada vez mais da colaboração com a Computação para conseguir resolver problemas trabalhosos!

# Bibliography


[1] S. Deutschländer *et al*, PRL **113**, 127801 (2014).

[2] J. A. Anderson *et al*, Comp. Phys. Comm., **204**, 21-30 (2016).

[3] J. A. Anderson *et al*, Jou. Comput. Phys. 227, 5342 (2008).

[4] `http://glotzerlab.engin.umich.edu/hoomd-blue`

[5] J. Glaser *et al*, Comp. Phys. Comm., **192**, 97 (2015).

[6] Etienne Bernard. Algorithms and applications of the Monte Carlo method : Two-dimensional melting and perfect sampling. Université Pierre et Marie Curie 2011.

[7] E. P. Bernard *et al*, PRE **80**, 056704 (2009).

Obrigado pela atenção!

# Seção 6

1. Introdução

2. Modelo Teórico

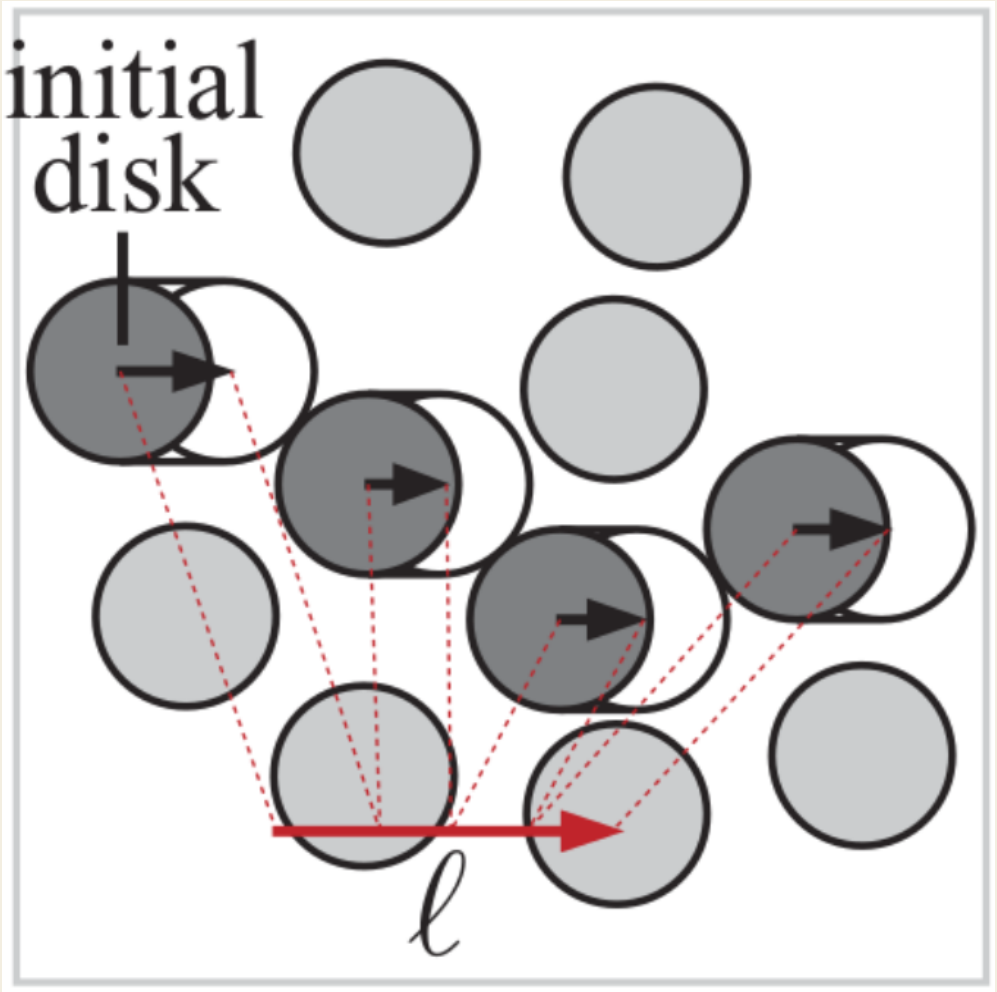
3. Implementação computacional

4. Resultados
   - 4.1 Monte Carlo Cadeia de Eventos
   - 4.2 HOOMD-blue HPMC

5. Conclusões

6. Extras

# Orientação local



# Transição de Fase

- O comportamento espacial de $\psi$ e $\phi$ pode ser classificado em termos do alcance: curto, longo e quase longo.
- A fase hexática é observada apenas em sistemas 2D.

| Ordem | Sólido | Hexático | Líquido |
|---|---|---|---|
| $\phi$ | Quase longo | Curto | Curto |
| $\psi$ | Longo | Quase longo | Curto |

Tabela: Fonte: Refs. [?, ?].

# Alcance

| Alcance | Comportamento |
|---|---|
| Curto | $\exp\left(-r/\xi_6\right)$ |
| Quase longo | $r^{-\eta_6}$ |
| Longo | constante $\neq 0$ |

Tabela: Definições dos tipos de alcance em função da distância [1].