

# **III ENCONTRO DE PÓS-GRADUAÇÃO EM FÍSICA E ASTRONOMIA DA UFSC**

## **Métodos Computacionais em Simulações de Monte Carlo**

**Prof. Lucas Nicolao**



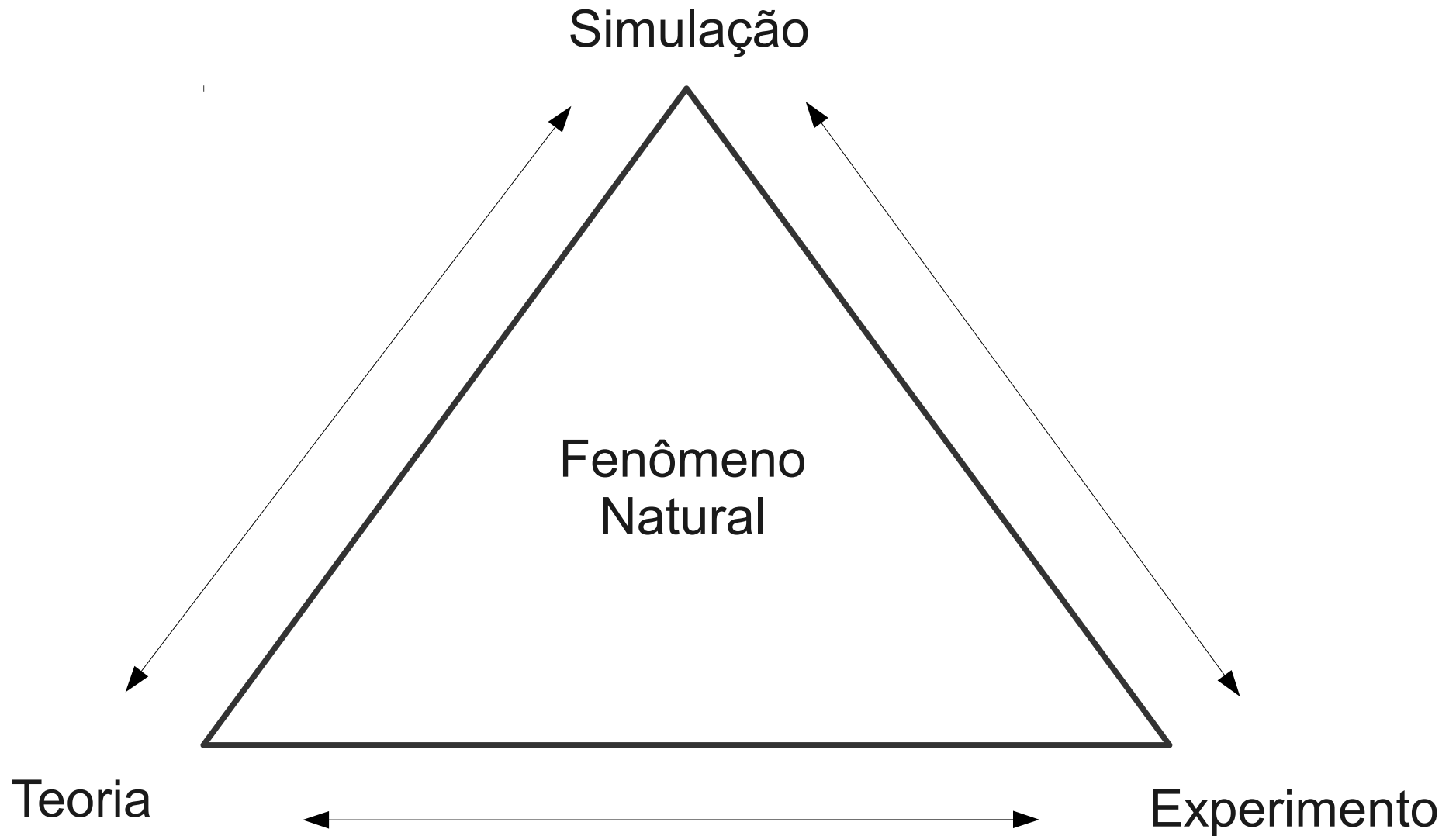
# Proposta desse mini-curso

- Método de Monte Carlo
  - Física estatística
  - Técnicas computacionais
- 
- Aula 1: teoria
  - Aula 2: desempenho
  - Aula 3: prática

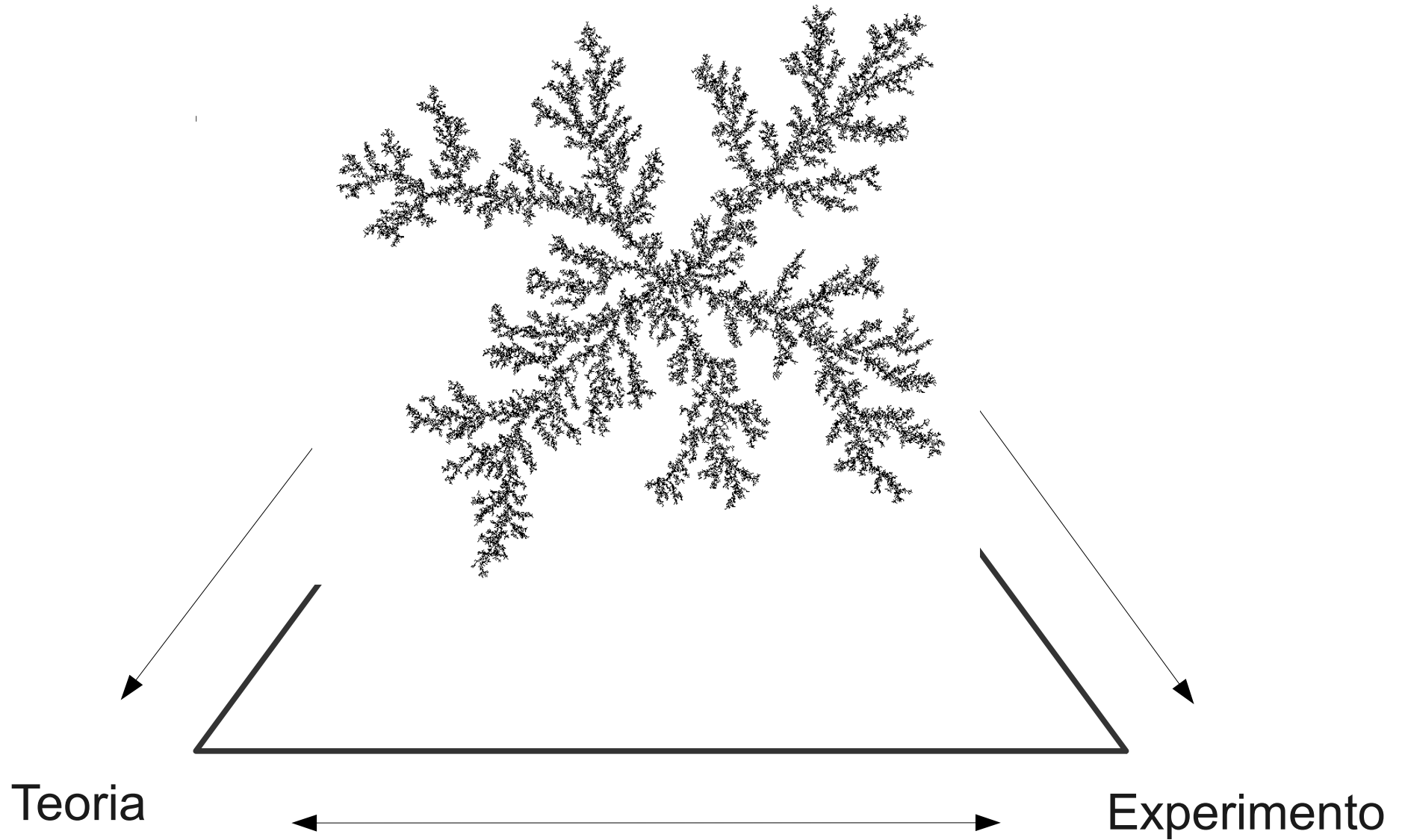
# Aula 1

- Introdução ao método
- Algoritmo de Metropolis
- Modelo de Ising

# Física computacional



# Física computacional



# Uso de números aleatórios para resolver problemas:

## Cálculo de $\pi$

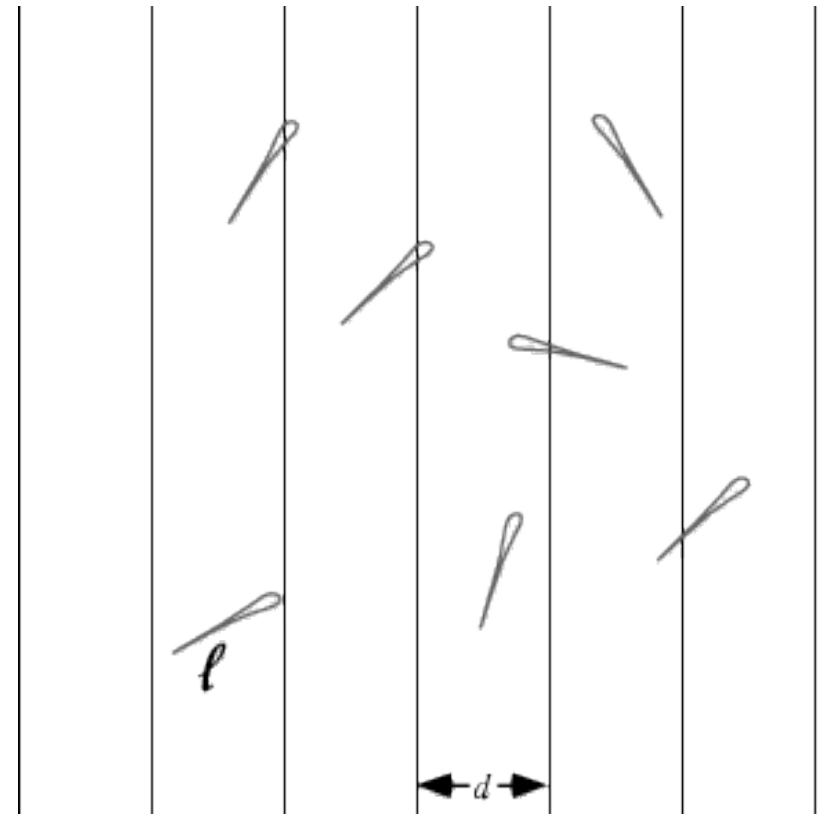
- Agulha de Buffon (1773):

probabilidade de agulha cruzar uma linha:

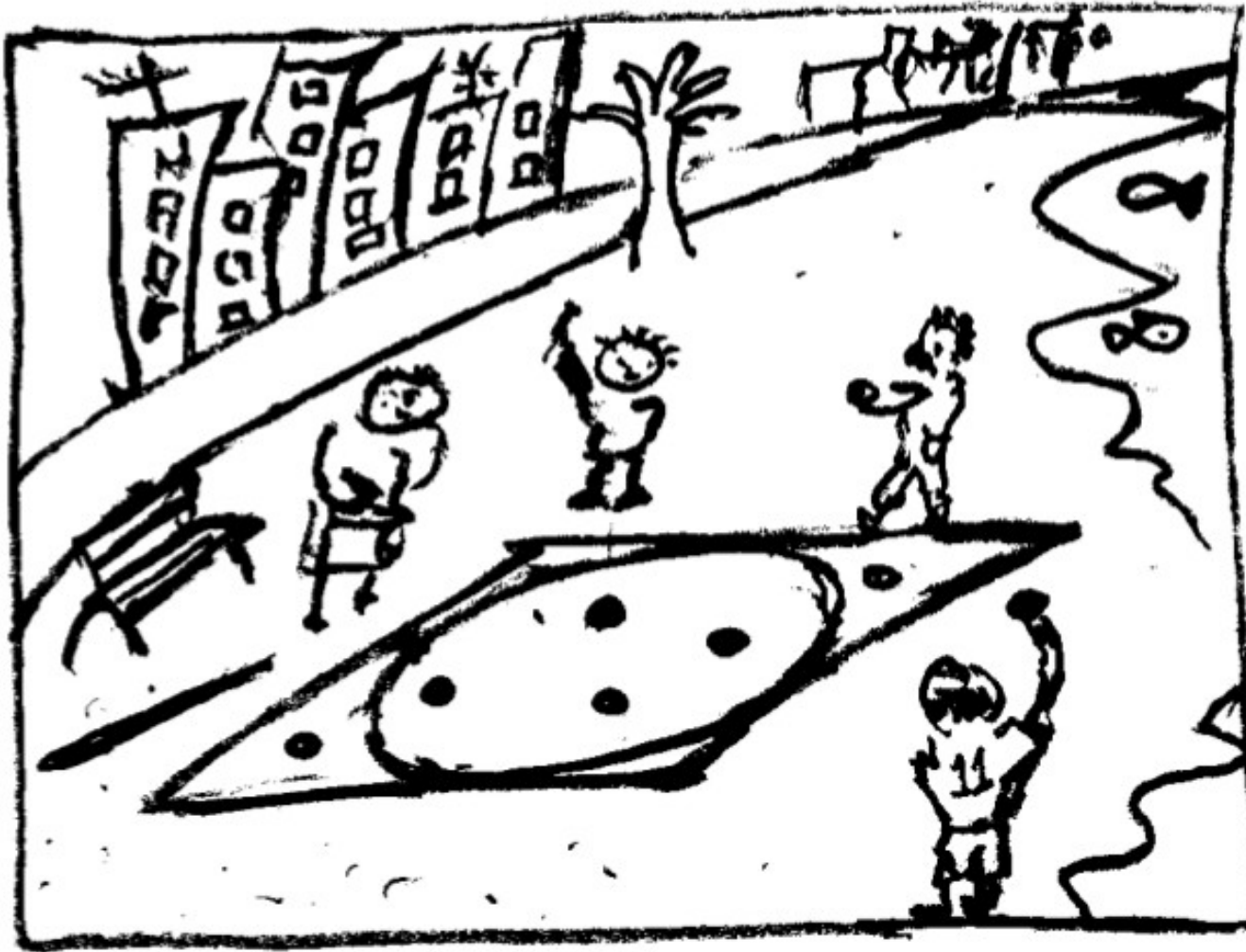
$$P = 2/\pi$$

$$P \rightarrow \frac{\text{no. de acertos}}{\text{no. total de jogadas}}$$

500 jogadas  $\sim 3.11(7)$



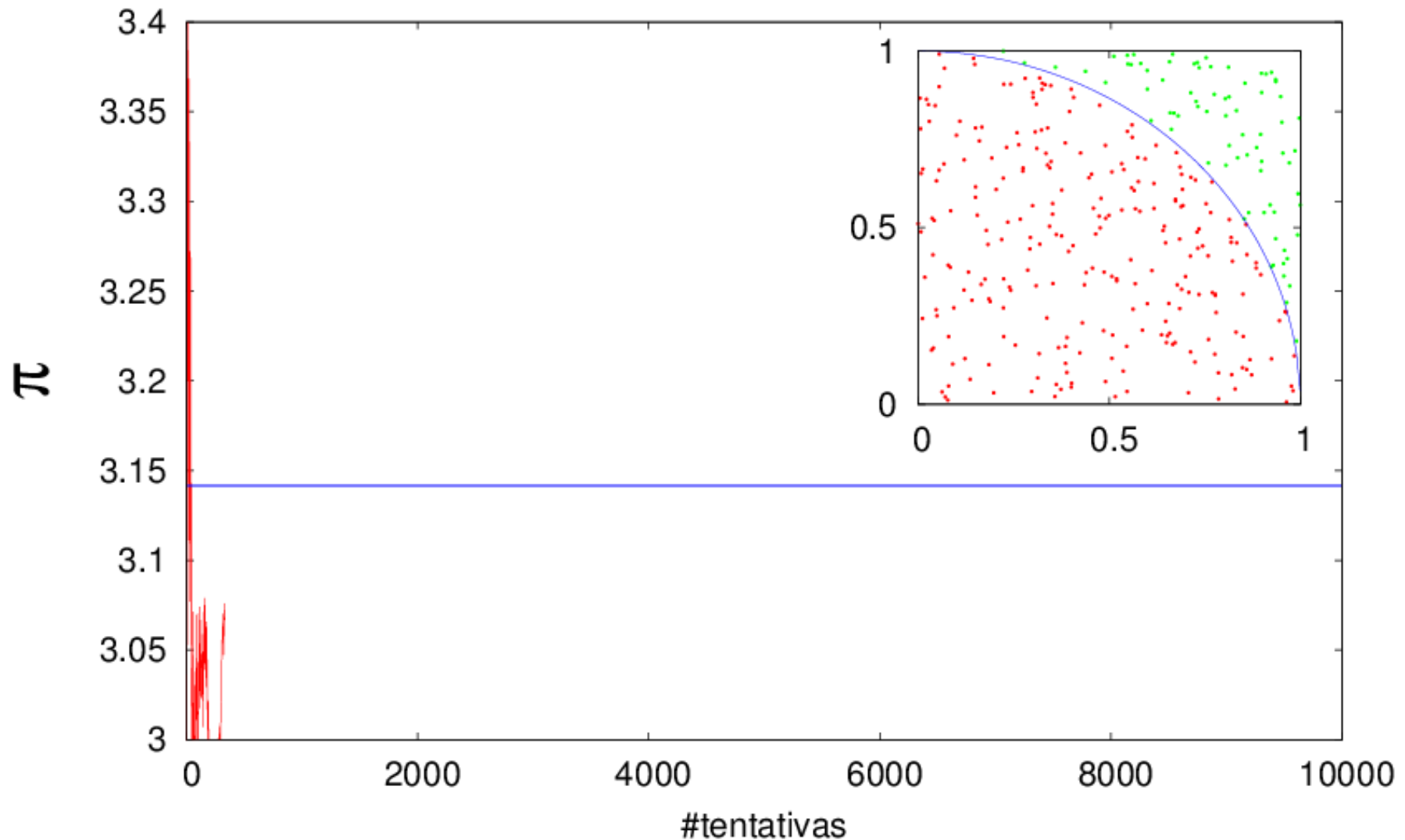
# Uso de números aleatórios para resolver problemas: Cálculo de $\pi$



$$\frac{\text{Area}_{\bigcirc}}{\text{Area}_{\square}} = \frac{\pi R^2}{4R^2} \approx \frac{\text{no.acertos}(\bigcirc)}{\text{no.tentativas}(\square)}$$

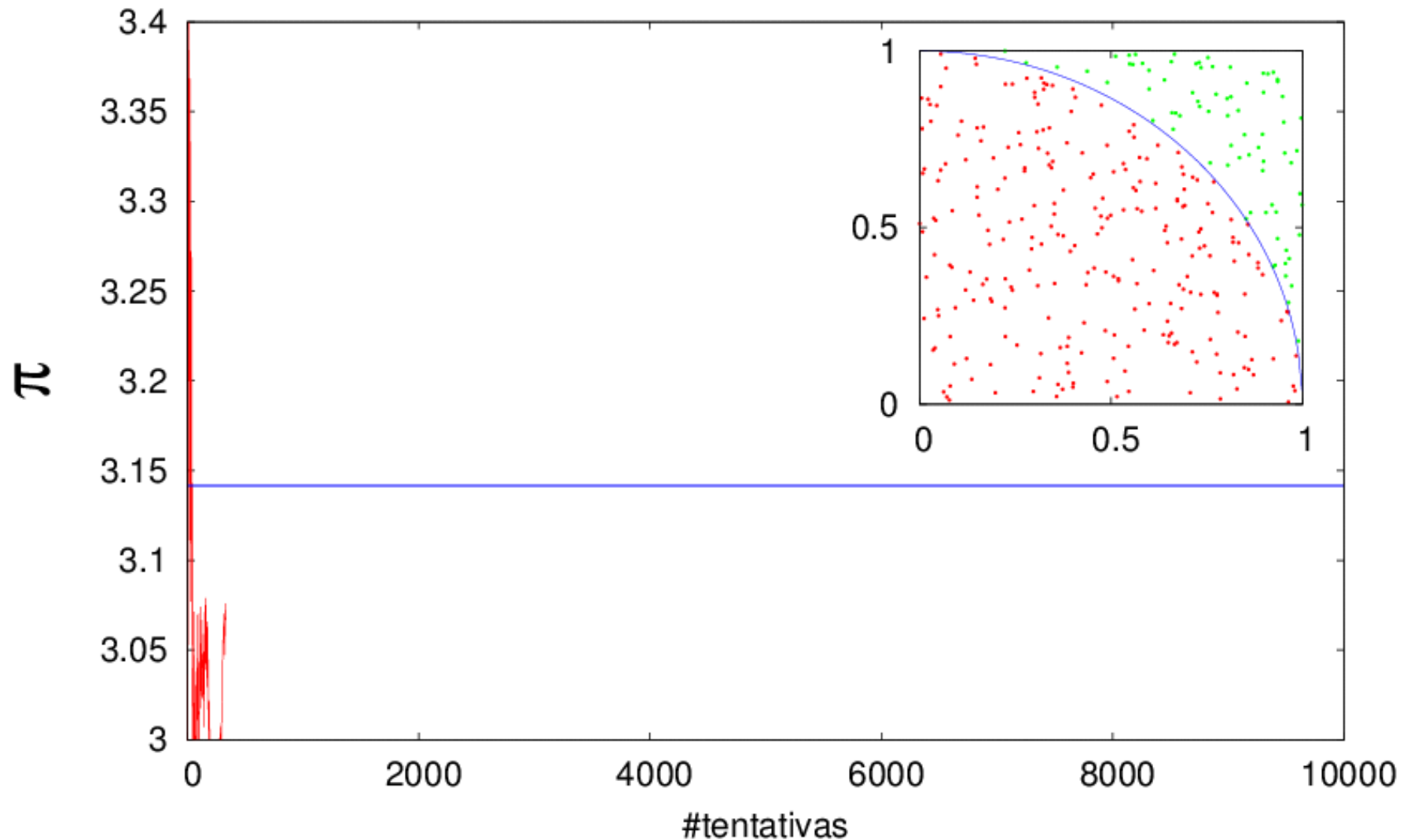
# Uso de números aleatórios para resolver problemas:

## Cálculo de $\pi$



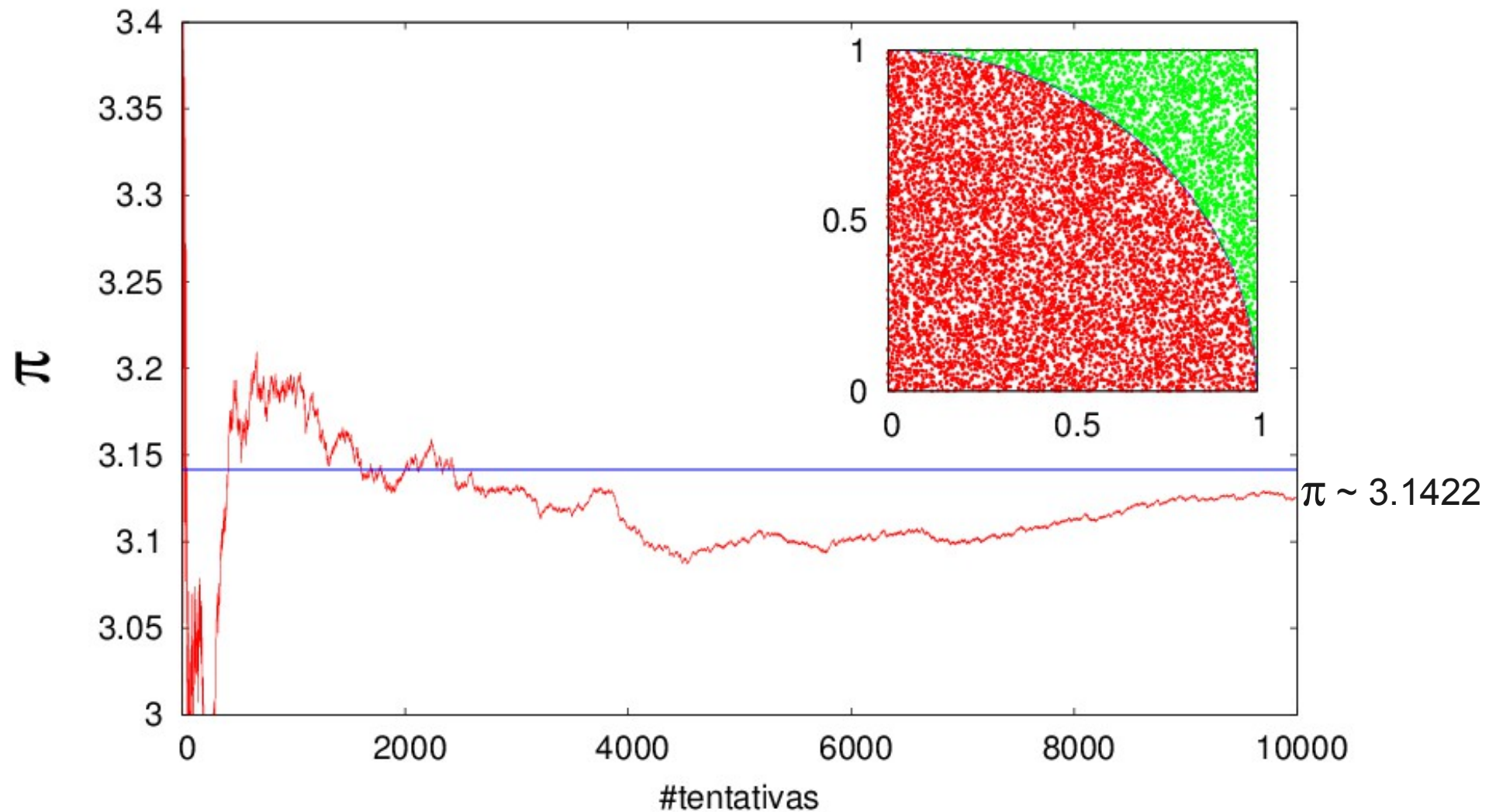


# Uso de números aleatórios para resolver problemas: Cálculo de $\pi$

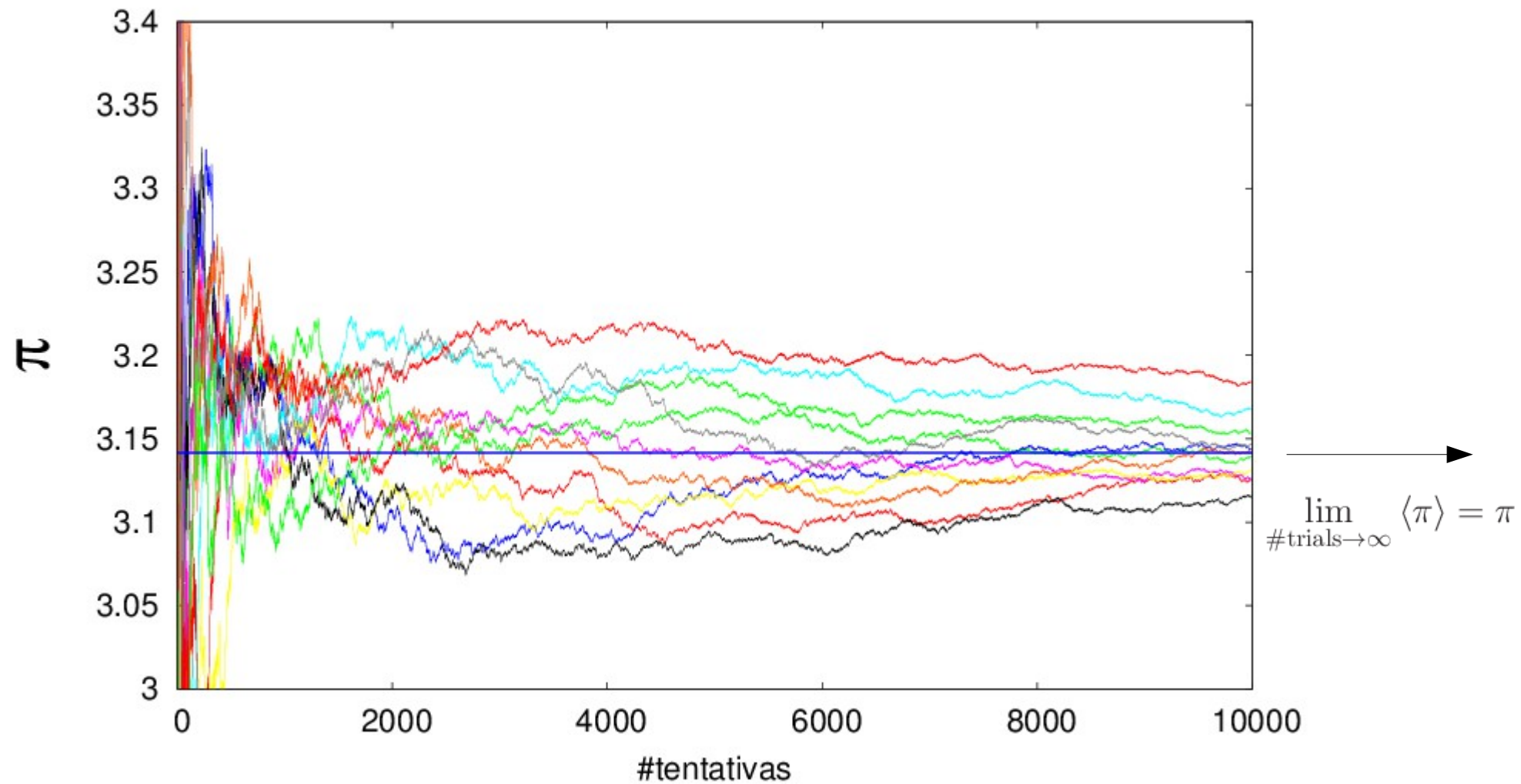


# Uso de números aleatórios para resolver problemas:

## Cálculo de $\pi$

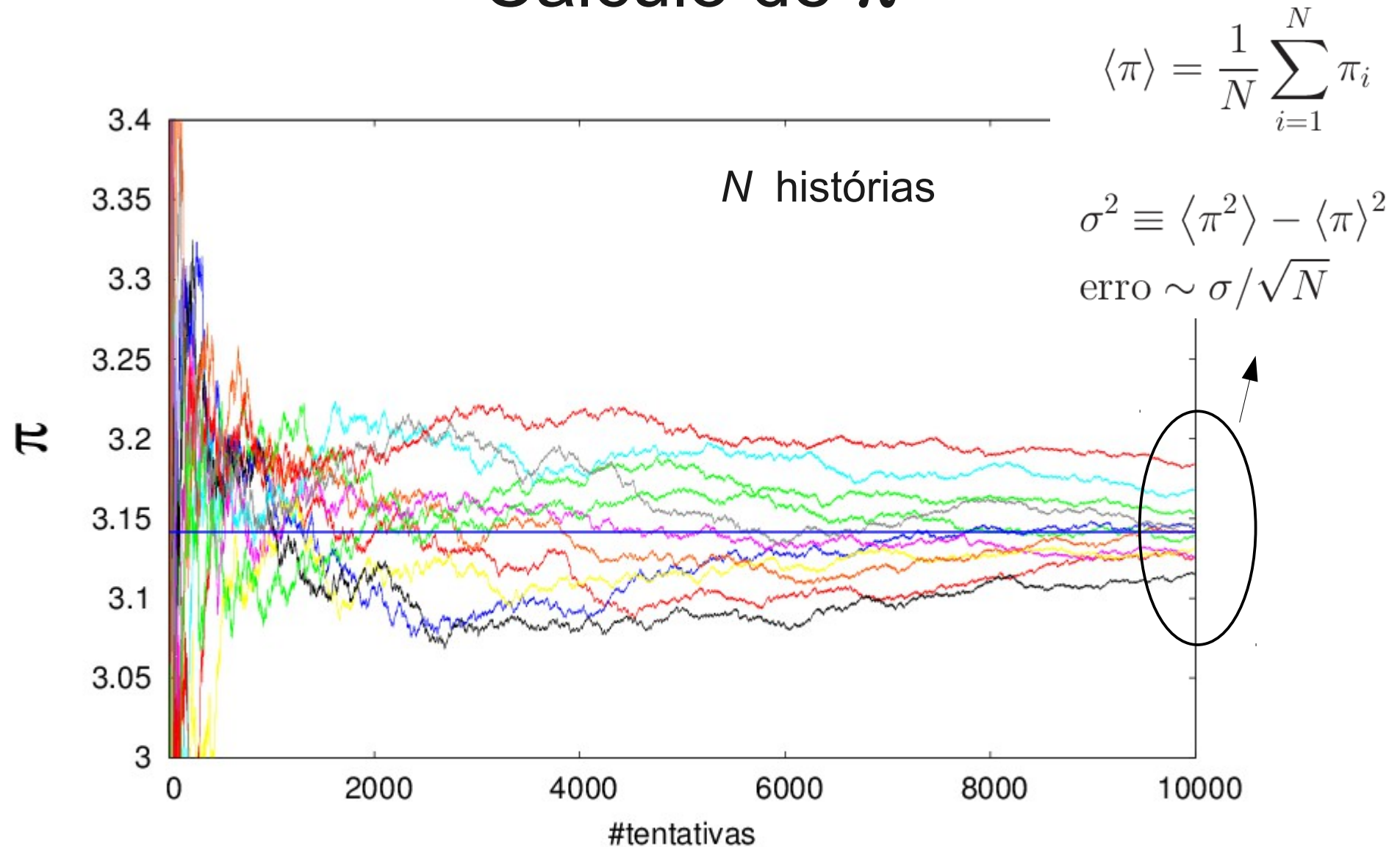


# Uso de números aleatórios para resolver problemas: Cálculo de $\pi$



# Uso de números aleatórios para resolver problemas:

## Cálculo de $\pi$



# Uso de números aleatórios para resolver problemas:

## Cálculo de $\pi$

$N$	$\langle \pi \rangle$	erro
100	3.143	0.002
900	3.1408	0.0005
2700	3.1414	0.0003
24300	3.1416	0.0001
218700	3.14159	0.00003

→ erro cai com  $\sqrt{N}$

# Uso de números aleatórios para resolver problemas:

## Cálculo de $\pi$

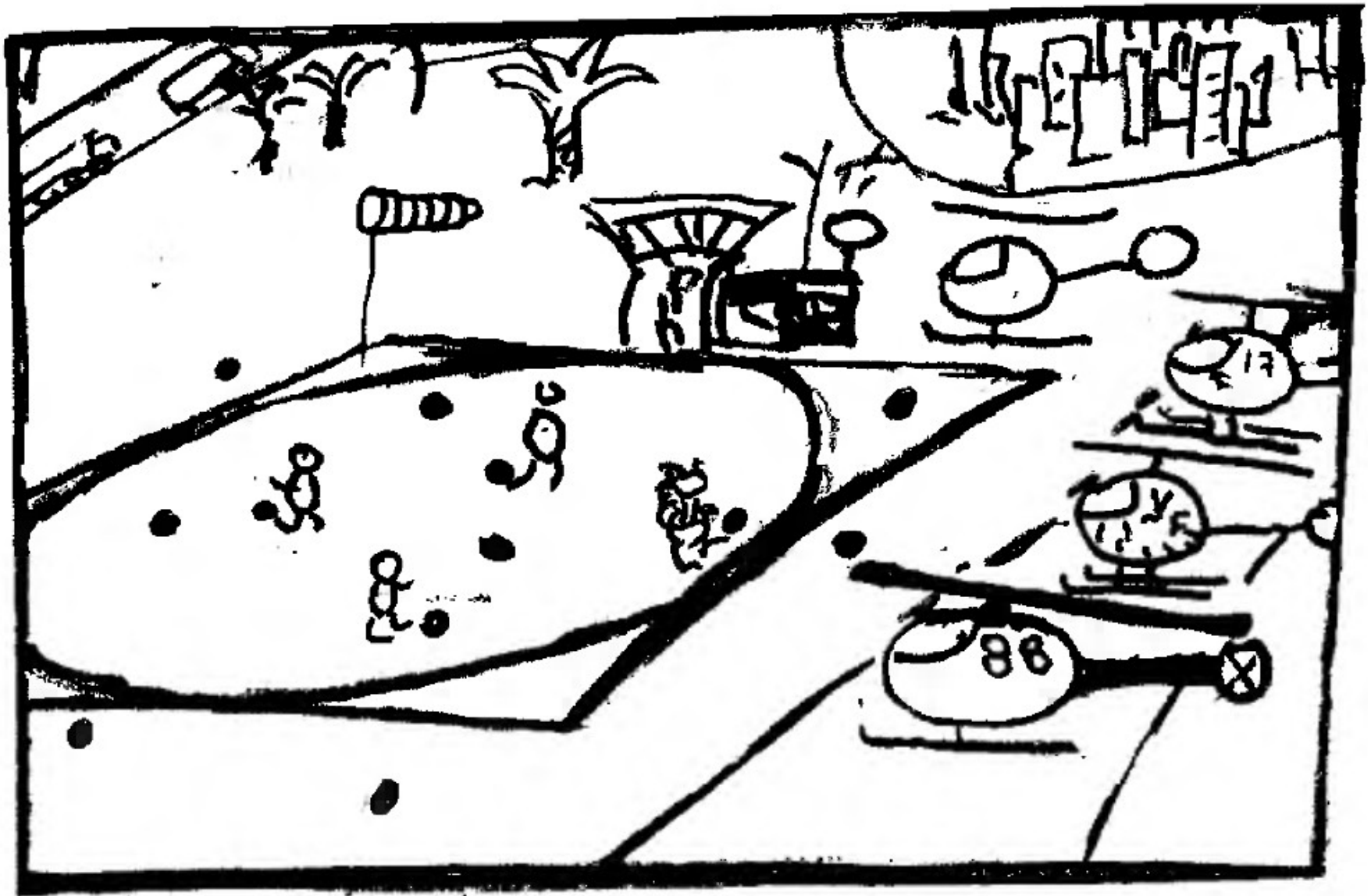
$N$	$\langle \pi \rangle$	erro
100	3.143	0.002
900	3.1408	0.0005
2700	3.1414	0.0003
24300	3.1416	0.0001
218700	3.14159	0.00003

```
count = 0;
for( n=0 ; n<10000 ; n=n+1 ){
    x = RANDOM;
    y = RANDOM;

    if( x*x + y*y < 1.0 )
        count = count + 1;
}
pi = 4.0*count/10000. ;
```

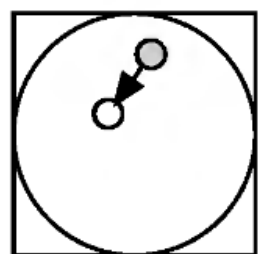
→ Monte Carlo por amostragem direta (simples)

# Amostragem por importância

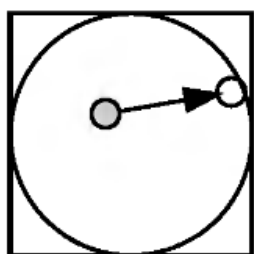




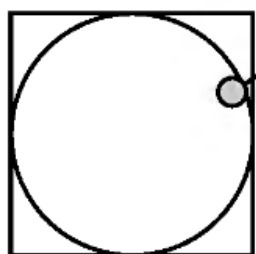
# Amostragem por importância



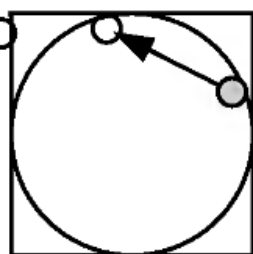
$i = 1$



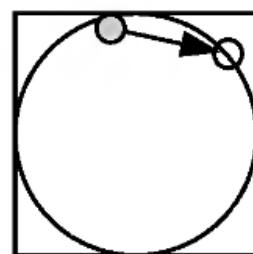
$i = 2$



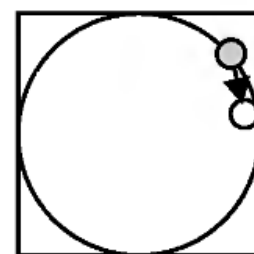
$i = 3$  (rej.)



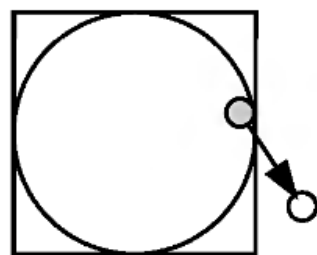
$i = 4$



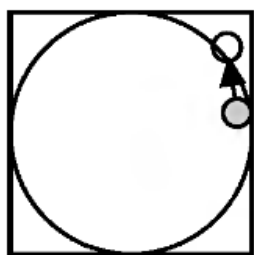
$i = 5$



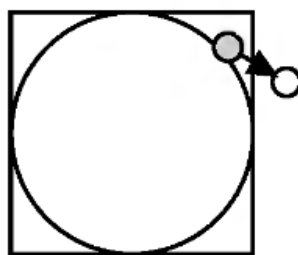
$i = 6$



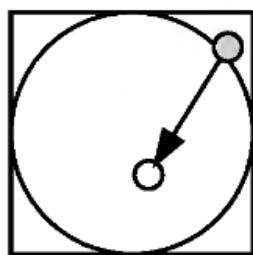
$i = 7$  (rej.)



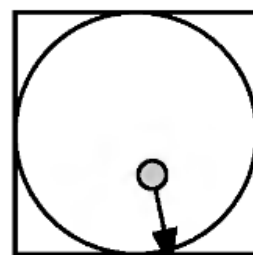
$i = 8$



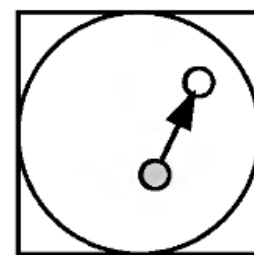
$i = 9$  (rej.)



$i = 10$



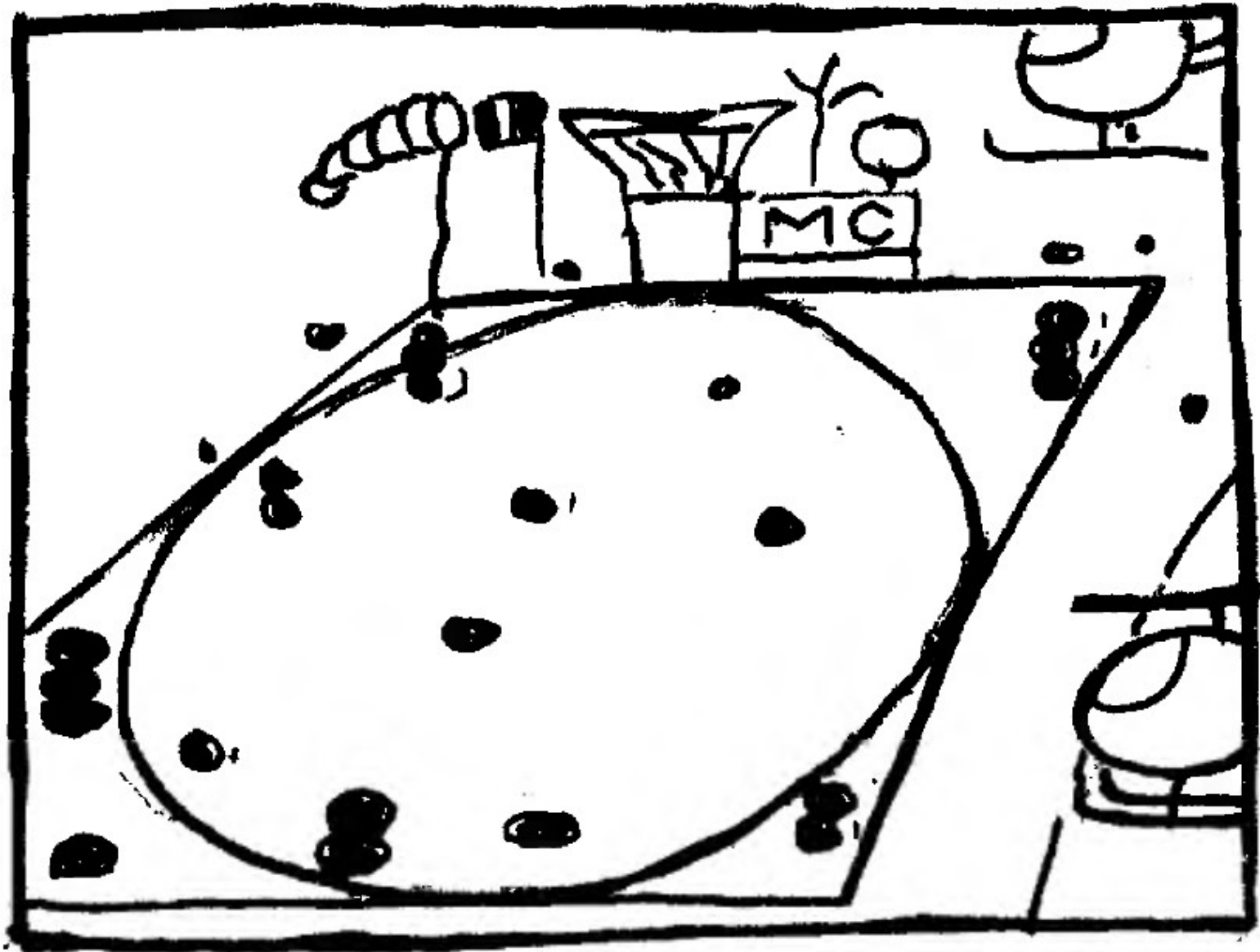
$i = 11$  (rej.)



$i = 12$



# Amostragem por importância



# Amostragem por importância

```
delta = 1.0;

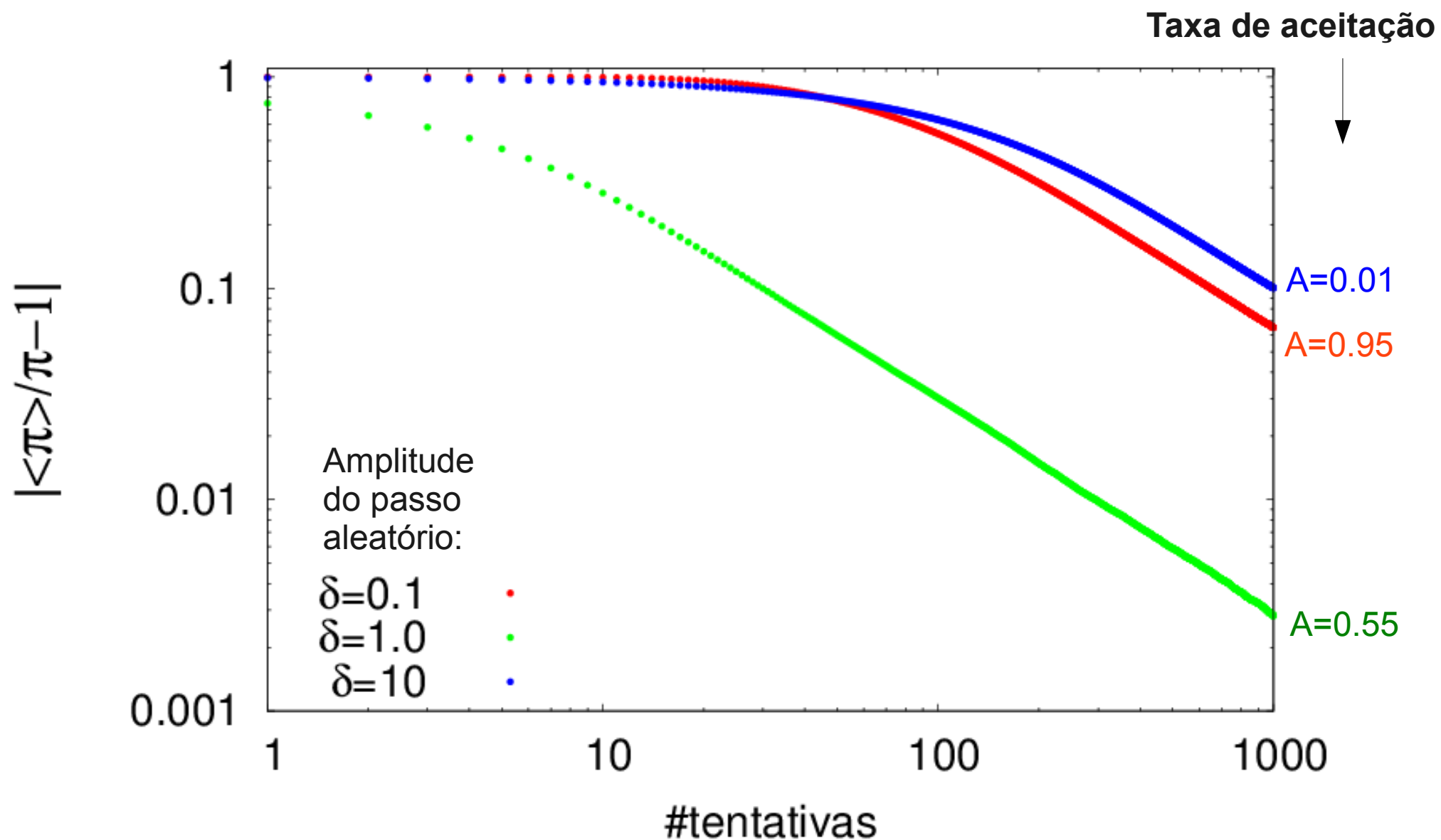
x=1;
y=1;
count=0;

for( n=0 ; n<1000 ; n++ ){
    rx = delta*(1.0f-2.*RANDOM);
    ry = delta*(1.0f-2.*RANDOM);

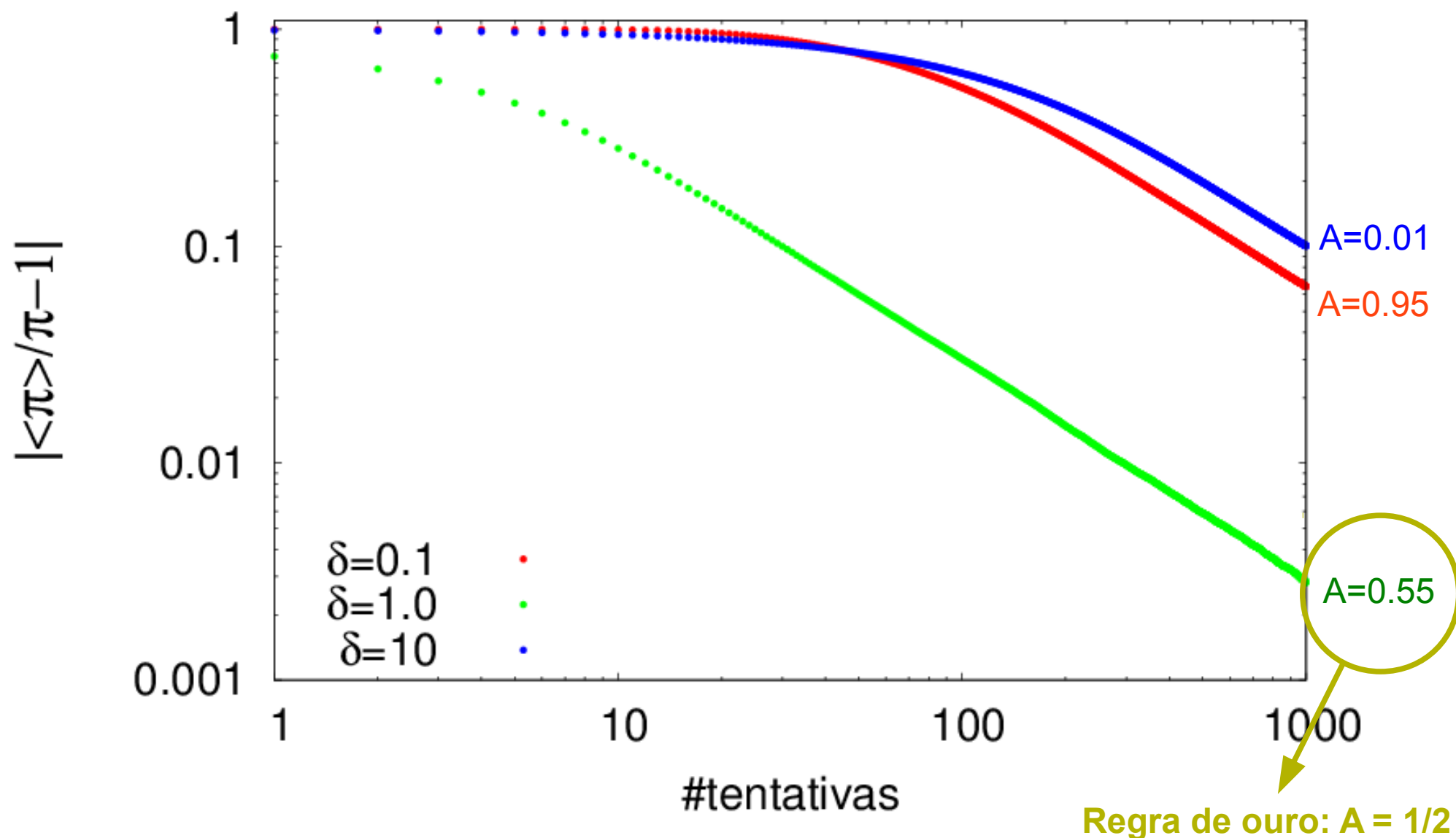
    if(fabs(x+rx) < 1.0  &&  fabs(y+ry) < 1.0){
        x+=rx;
        y+=ry;
    }
    if(x*x + y*y < 1.0)
        count++;

}
pi = 4.0 * count / 1000.0;
```

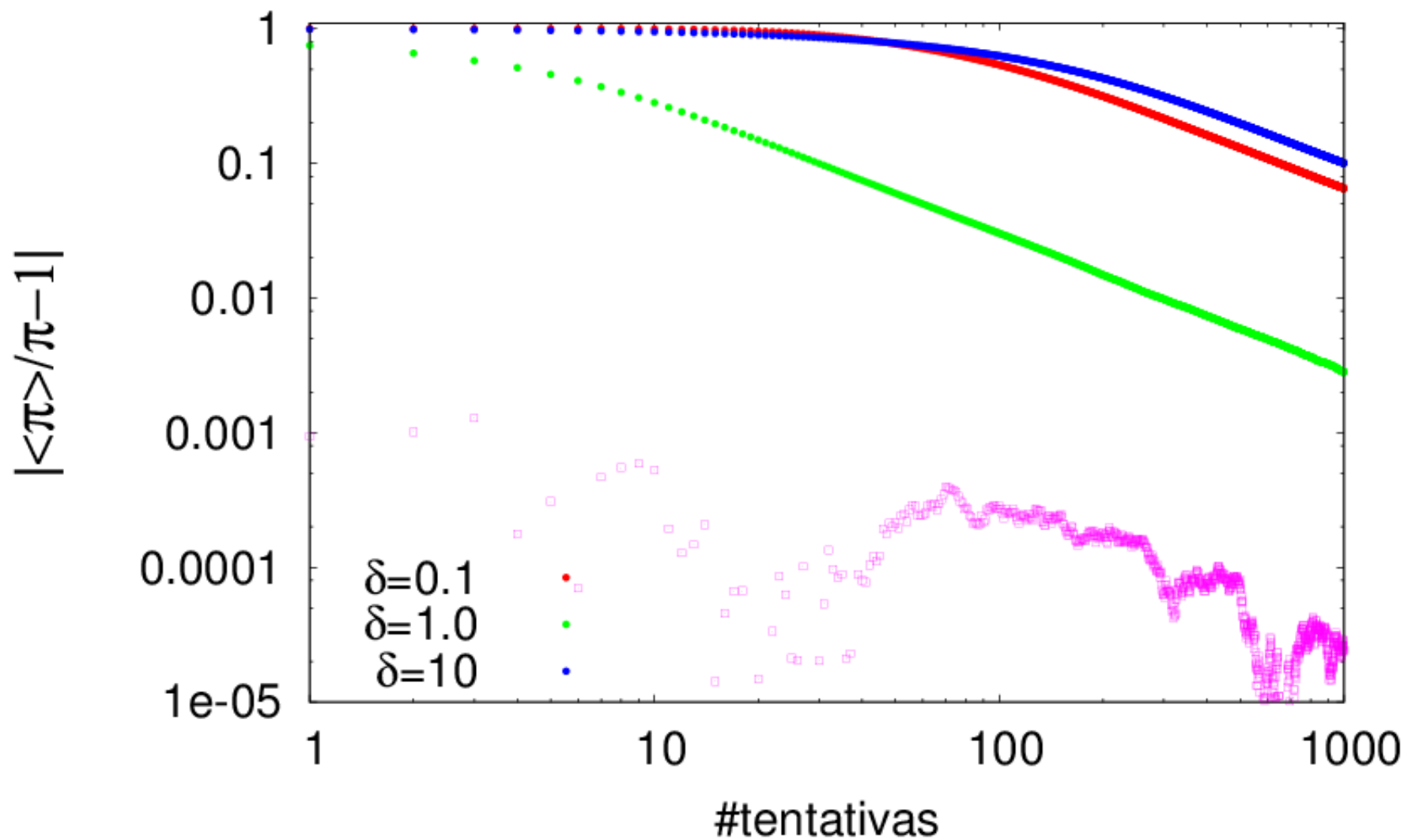
# Amostragem por importância



# Amostragem por importância



# Amostragem por importância



# Método de Monte Carlo

- Fermi
- Ulam
- von Neumann
- **Metropolis**

THE JOURNAL OF CHEMICAL PHYSICS

VOLUME 21, NUMBER 6

JUNE, 1953

## Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,  
*Los Alamos Scientific Laboratory, Los Alamos, New Mexico*

AND

EDWARD TELLER,\* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N V(\vec{d}_{ij}).$$

$$\bar{F} = \left[ \int F \exp(-E/kT) d^{2N}p d^{2N}q \right] /$$

Cálculo da integral (função de partição):

$2N$  – dimensional, escolha aleatória:

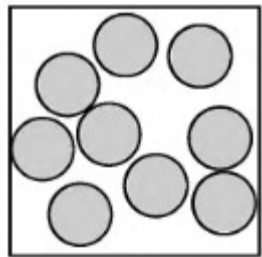
soma de fatores:  $\exp(-E/kT)$

(com alta probabilidade de serem  
muitíssimo pequenos)...

Método de Monte Carlo modificado:

$$\left[ \int \exp(-E/kT) d^{2N}p d^{2N}q \right]$$

employ is actually a modified Monte Carlo scheme, where, instead of choosing configurations randomly, then weighting them with  $\exp(-E/kT)$ , we choose configurations with a probability  $\exp(-E/kT)$  and weight them evenly.



$\Delta E$  { If  $\Delta E < 0$  we allow the move  
If  $\Delta E > 0$  we allow the move with probability  $\exp(-\Delta E/kT)$

$$X \rightarrow X + \alpha \xi_1$$

$$Y \rightarrow Y + \alpha \xi_2,$$

$$\xi_3 < \exp(-\Delta E/kT)$$

$$\bar{F} = (1/M) \sum_{j=1}^M F_j$$

# Cadeias de Markov

Mecanismo para gerar uma sequência de estados (configurações) aleatórios do sistema (estados discretos): a partir de um estado  $i$ , gera um novo estado  $j$  – probabilidade de ocupação e prob. transição

$$p_i(t+1) = \sum_j p_j(t) w_{ji}$$

$$\sum_i p_i(t) = 1$$

$$\sum_j w_{ij} = 1 \quad \forall i$$

$$p_i(t+1) - p_i(t) = \sum_j p_j(t) w_{ji} - \sum_j p_i(t) w_{ij}$$

$$\pi_i = \lim_{t \rightarrow \infty} p_i(t)$$

$$\pi_i = \sum_j \pi_j w_{ji}$$

$$0 = \sum_j \pi_j w_{ji} - \sum_j \pi_i w_{ij}$$

$$\pi_i w_{ij} = \pi_j w_{ji} \quad \forall i, j$$



# Balanço detalhado

$$\pi_i w_{ij} = \pi_j w_{ji} \quad \forall i, j$$

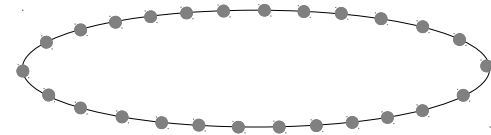
Mesmo assim, muitas soluções

$$\frac{w_{ji}}{w_{ij}} = \frac{\pi_i}{\pi_j} = e^{-\beta(E_i - E_j)}$$

→ o que desejamos  
(mec. estatística)

Uma possível  
solução:

$$\begin{aligned} w_{ji} &= \pi_i \\ w_{ij} &= \pi_j \end{aligned}$$



M estados  
 $\pi_i = 1/M$

Outra solução:

$$\begin{aligned} w_{ji} &= \pi_i / \pi_j \\ w_{ij} &= 1 \end{aligned}$$

→ Metropolis:

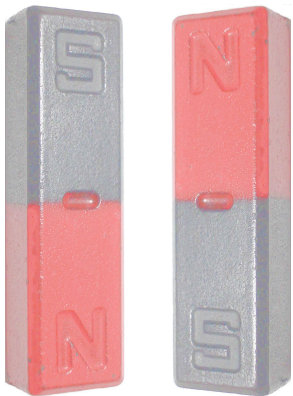
$$w_{ji} = \min(1, \pi_i / \pi_j)$$

Transição garantida se novo estado  $i$  tem prob + alta.  
E é improvável ir para estados de prob. muito baixa.

# Modelo de Ising

Modelo para ferromagnetismo (Lenz  $\rightarrow$  Ising) 1924

$$E = -J \sum_{\langle i,j \rangle} s_i s_j \quad s_i = \pm 1$$



momentos magnéticos

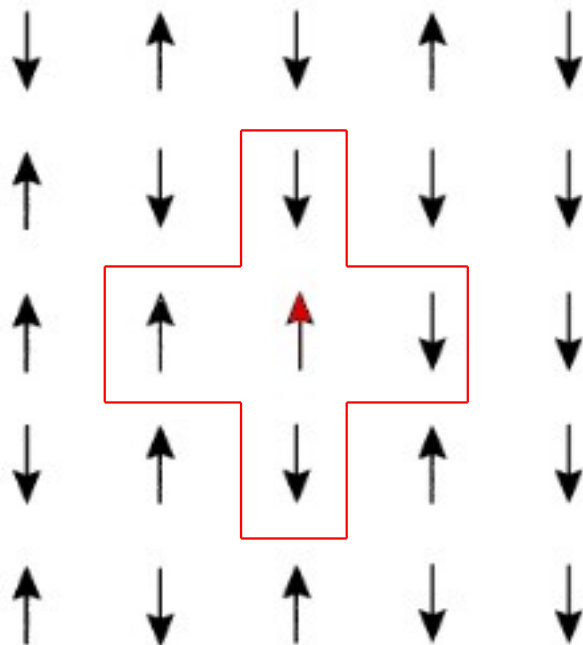
Ex: Ferro

Interação de Heisenberg

- Modelo simples
- Exatamente solúvel (Onsager 1944)
- Transição de fase

# Modelo de Ising

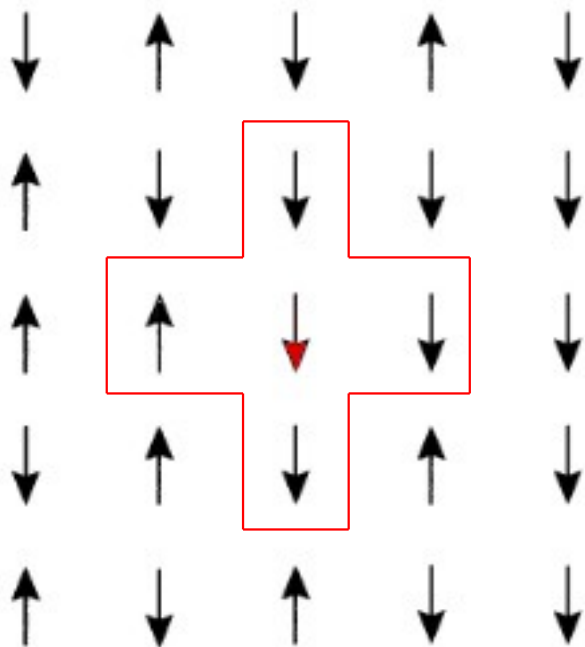
$$E = -J \sum_{\langle i,j \rangle} s_i s_j \quad s_i = \pm 1$$



$$E_i = -J(+1)(+1 - 1 - 1 - 1) = 2J$$

# Modelo de Ising

$$E = -J \sum_{\langle i,j \rangle} s_i s_j \quad s_i = \pm 1$$



$$E_i = -J(-1)(+1 - 1 - 1 - 1) = -2J$$

# Modelo de Ising

$$E \rightarrow E/J$$

$$T \rightarrow T/k_B$$

$$Z = \sum_{s_1=\pm 1} \sum_{s_2=\pm 1} \dots \sum_{s_N=\pm 1} e^{\beta \sum_{\langle i,j \rangle} s_i s_j}$$

$$m(\vec{s}) = \frac{M(\vec{s})}{N} = \frac{1}{N} \sum_{i=1}^N s_i$$

$$e(\vec{s}) = \frac{E(\vec{s})}{N} = -\frac{1}{N} \sum_{\langle i,j \rangle} s_i s_j$$

$$Z = \sum_{\vec{s}} e^{-E(\vec{s})/T}$$

$$\langle A \rangle = \sum_{\vec{s}} A(\vec{s}) P_{GB}(\vec{s})$$

$$P_{GB}(\vec{s}) = \frac{e^{-E(\vec{s})/T}}{Z} \quad \begin{cases} T \rightarrow \infty \\ T \rightarrow 0 \end{cases}$$

# Energia Livre de Landau

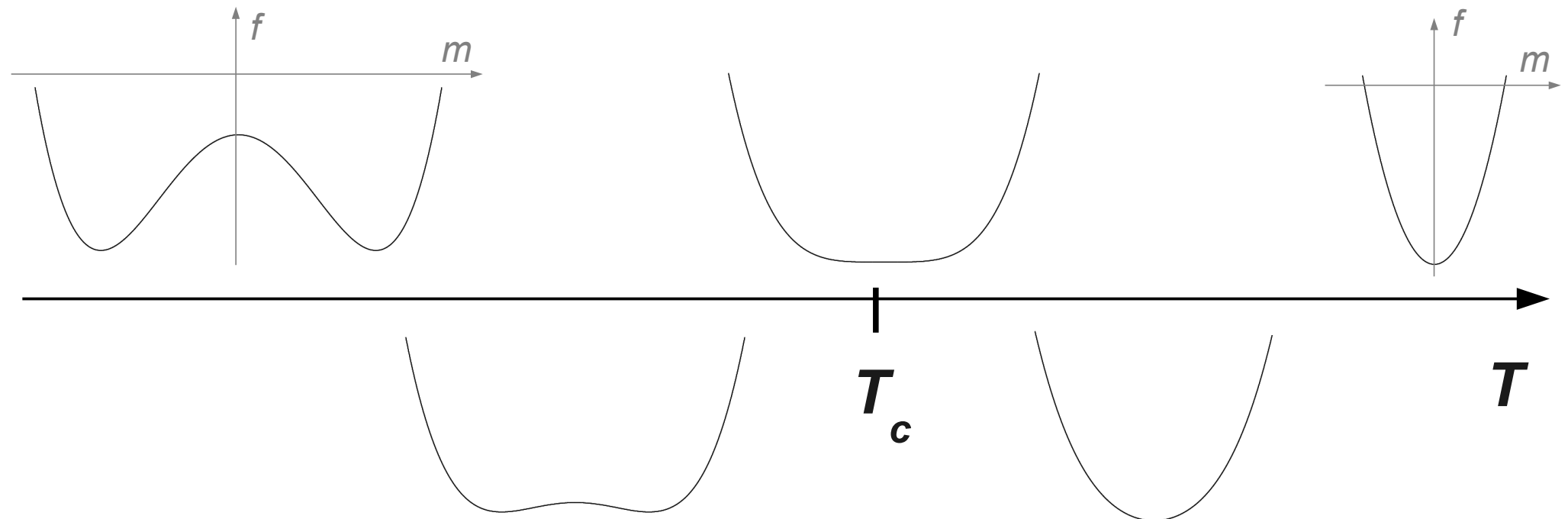
$$\mathcal{P}(m) \equiv \sum_{\substack{\vec{s}: \\ \sum_i s_i = Nm}} P_{GB}(\vec{s})$$

$$f(m) \simeq (T_c - T)m^2 + m^4$$

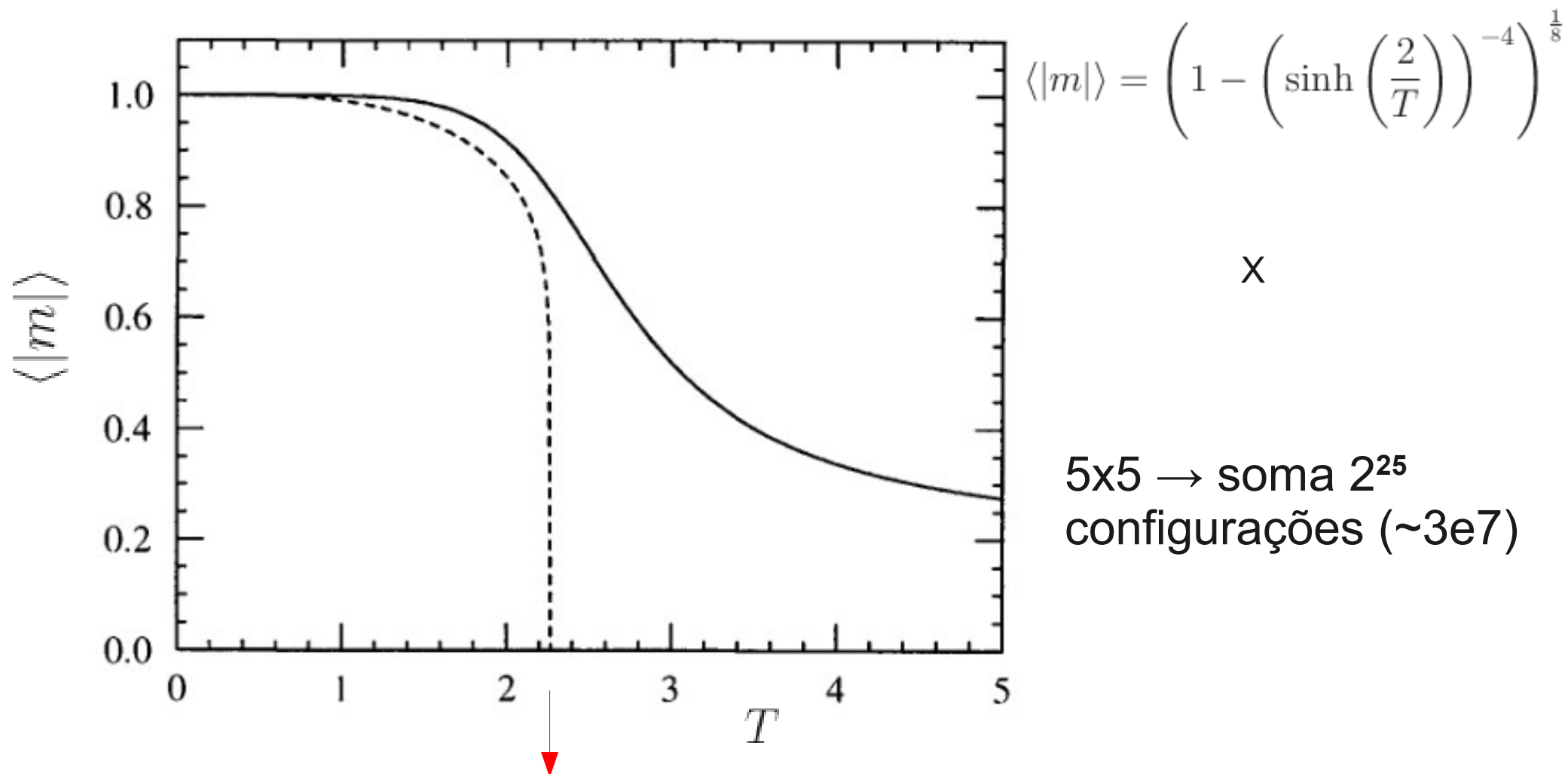
$$F = E - TS$$

Equilíbrio = min(f)  
barreira de energia

$$f(m) \equiv -\frac{T \log(\mathcal{P}(m))}{N}$$



# Soluções Exatas



5x5  $\rightarrow$  soma  $2^{25}$   
configurações ( $\sim 3e7$ )

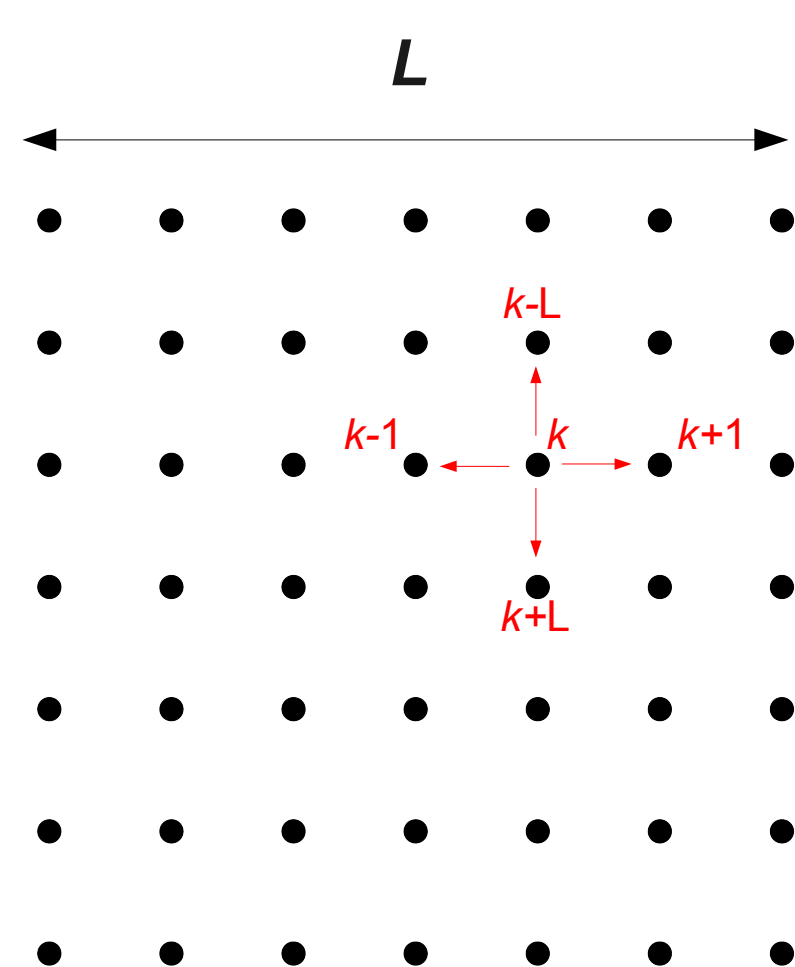
$$T_c = 2 / \log(1 + \sqrt{2}) \simeq 2.269 \quad (k_B/J)$$

# Um passo do algoritmo de Metropolis

- 1) Escolhemos aleatoriamente um spin  $\mathbf{s}_k$  para inverter  $\rightarrow$  nova configuração proposta  $\vec{\mathbf{s}}'$  difere da antiga apenas por esse spin (*single spin flip dynamics*).
- 2) Calculamos a diferença de energia  $\Delta E = E(\vec{\mathbf{s}}') - E(\vec{\mathbf{s}})$ .
- 3) Se  $\Delta E \leq 0$ , a nova configuração é aceita e o passo temporal é finalizado.
- 4) Senão, extraímos um no. aleatório  $r$  uniforme em  $[0, 1]$ :
  - Se  $r \leq \exp(-\Delta E / T)$  aceitamos a nova configuração  $\vec{\mathbf{s}}'$ .
  - Se  $r > \exp(-\Delta E / T)$  rejeitamos a nova configuração.



# Para o caso do modelo de Ising 2D



$N = L^2$  spins

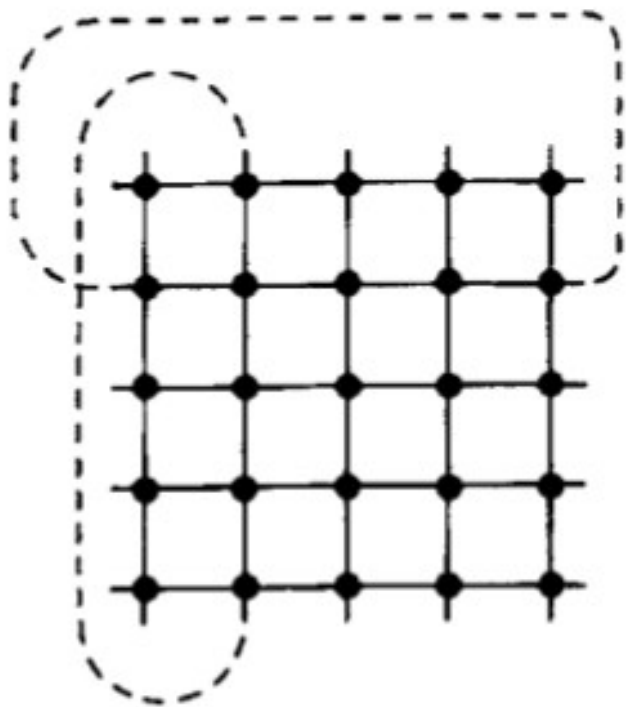
$$E(\vec{s}) = - \sum_{\langle i,j \rangle \neq k} s_i s_j - s_k (s_{k+1} + s_{k-1} + s_{k+L} + s_{k-L})$$

$$E(\vec{s}') = - \sum_{\langle i,j \rangle \neq k} s_i s_j + s_k (s_{k+1} + s_{k-1} + s_{k+L} + s_{k-L})$$

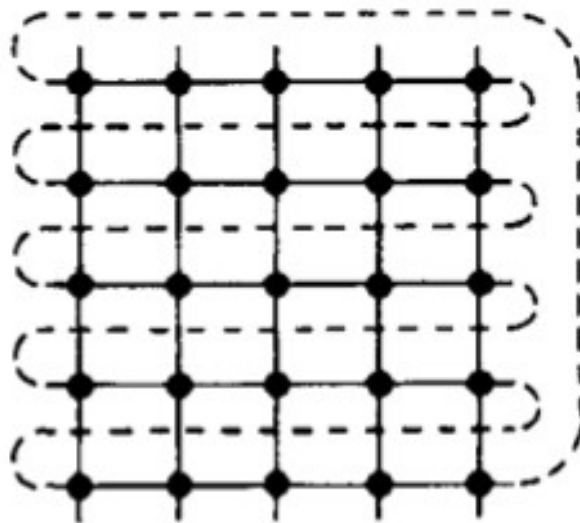
$$\Delta E = E(\vec{s}') - E(\vec{s}) = 2s_k (s_{k+1} + s_{k-1} + s_{k+L} + s_{k-L})$$

$$\Delta E = (-8, -4, 0, +4, +8) \\ \in [-2, +2] \times 4$$

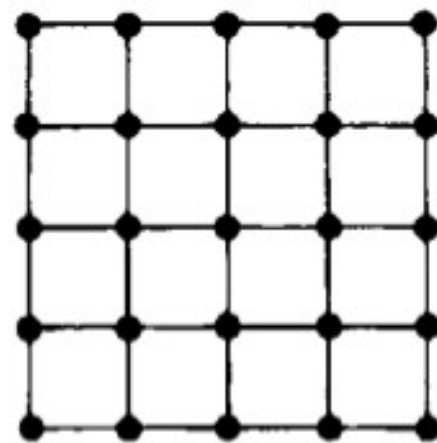
# Condições de contorno



CC Periódicas



CC Helicoidais



CC livres

# Implementação básica

- Inicialização dos spins
- Atualização de Monte Carlo
- Medições de quantidades

# Inicialização

Condição inicial desordenada:

Fase paramagnética ( $T \rightarrow \infty$ )

```
int s[N];  
  
for( i=0 ; i<N ; i++ )  
    if( FRANDOM > 0.5 )  
        s[i] = 1;  
    else  
        s[i] = -1;
```

Condição inicial ordenada:

Fase ferromagnética ( $T = 0$ )

```
for(i=0;i<N;i++)  
    s[i] = 1 ;
```

//que em C também pode ser escrito assim:

```
for(i=0;i<N;i++)  
    s[i] = ((FRANDOM > 0.5) ? 1 : -1);
```

# Atualização

...

```
for (iy = 1; iy < L-1; iy++) {  
    for (ix = 1; ix < L-1; ix++) {  
  
        site++;  
  
        soma = s[site] * (s[site-L] + s[site-1] + s[site+1] + s[site+L]);  
  
        if (soma <= 0)  
            s[site] = -s[site];  
  
        else if( FRANDOM < exp(-2.0*soma/temperatura))  
            s[site] = -s[site];  
  
    }  
}
```

...

# Medições

```
energia=0;
magnetizacao=0;
site=0;

...

for (iy = 1; iy < L-1; iy++) {
    for (ix = 1; ix < L-1; ix++) {

        energia -= s[site] * (s[site-L] + s[site-1] + s[site+1] + s[site+L]);
        magnetizacao += s[site];

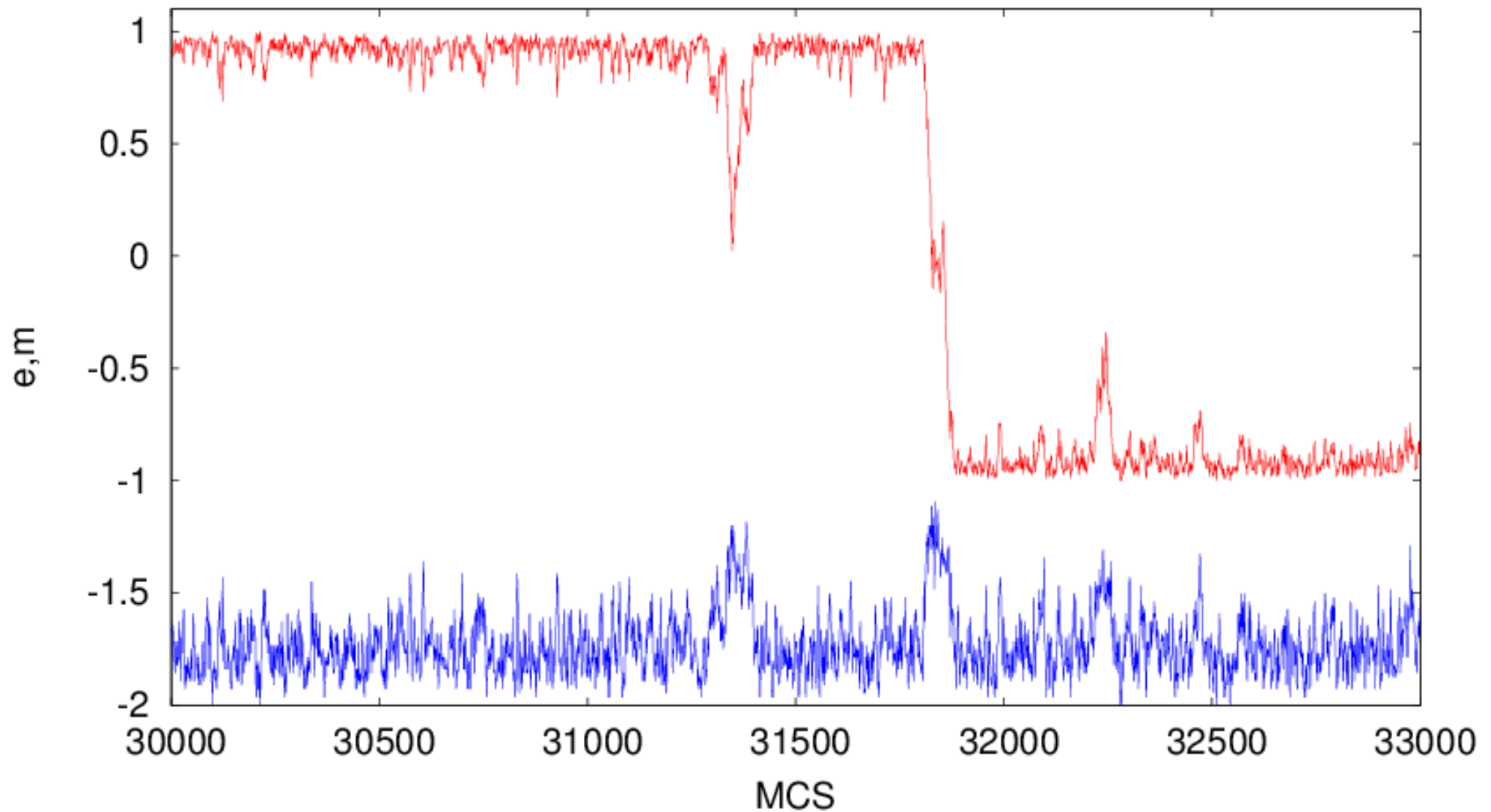
        site++;
    }
}

...

magnetizacao = magnetizacao/N;
energia = energia/N/2.0;
```

# Barreiras de energia

L=15    T=2.0



# Próxima aula

- Implementação
- Como utilizar (boas práticas)
- Otimização
- Alternativas