



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS**  
**DEPARTAMENTO DE FÍSICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA**

**JONATHAN MARCIO AMÂNCIO SALES**

**PERCOLAÇÃO INVASIVA DE SÍTIOS EM REDES QUADRADAS**

**FORTALEZA, CEARÁ**

**2018**

## SUMÁRIO

<b>1</b>	<b>PERCOLAÇÃO INVASIVA . . . . .</b>	<b>1</b>
<b>1.1</b>	<b>Avalanches . . . . .</b>	<b>10</b>
<b>2</b>	<b>ALGORITMO PRIORITY HEAP . . . . .</b>	<b>14</b>
<b>2.1</b>	<b>Árvore binária armazenada em um vetor . . . . .</b>	<b>14</b>
<b>2.2</b>	<b>Min-heap . . . . .</b>	<b>14</b>
	<b>APÊNDICE A - CÓDIGO EM C++ DO MODELO DE PERCOLAÇÃO</b>	
	<b>INVASIVA SEM APRISIONAMENTO . . . . .</b>	<b>17</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>20</b>

## 1 PERCOLAÇÃO INVASIVA

O modelo de Percolação Invasiva é um processo de percolação dinâmico desenvolvido em 1983 por D. Wilkinson e J. F. Willemsen [1] com o intuito de estudar o escoamento de dois fluidos imiscíveis em meios porosos [2]. Entretanto, este modelo, a princípio, pode ser aplicado a qualquer tipo de processo de invasão que prossiga por um caminho de menor resistência [1]. Considere o exemplo em que água é injetada em um meio poroso preenchido com óleo. Nesse caso, quando o processo de injeção é feito muito lentamente, isto é, em um regime quase estático, o que significa um baixo número Capilar ( $Ca \ll 1$ ) [2, 3], as forças capilares tornam-se dominantes em relação às forças viscosas e a dinâmica do sistema é determinada localmente pelo tamanho dos poros [4]. Além disso, assume-se que as pressões capilares nas interfaces entre os fluidos estão em equilíbrio [2]. Podemos então, definir um fluido invasor e um fluido defensor, como por exemplo, a água e o óleo, respectivamente, usados no exemplo anterior.

A teoria da percolação invasiva é baseada no mapeamento do tamanho de cada poro em um meio idealizado [2]. Assim, computacionalmente, o meio poroso é representado por uma rede (matriz)  $L \times L$  com cada sítio  $i$  associado a um número aleatório  $r_i$ , uniformemente distribuído entre  $[0, 1]$  [1, 5], representando o raio de cada poro [4]. Além disso, um ou vários sítios invasores são definidos para iniciar a dinâmica e um ou vários sítios defensores, para encerrar a dinâmica; também é definido a chamada interface ou fronteira, ou ainda, sítios de crescimento entre os fluidos, como aqueles sítios defensores que são vizinhos dos sítios invasores [4]. A dinâmica do sistema, portanto, consiste em, a cada passo de tempo, invadir o sítio de menor raio  $r_i$  pertencente à fronteira e, em seguida, atualizar a fronteira, isto é, a lista dos sítios de crescimento [4]. Vale ressaltar que o efeito gravitacional pode ser adicionado ao estudo de percolação invasiva (veja as referências [6, 7, 8, 9]). Entretanto, nas simulações realizadas neste trabalho, o efeito gravitacional não foi considerado.

Figura 1 – Esquema ilustrativo do modelo de percolação invasiva em uma rede quadrada.

0.05	0.10	0.03	0.82	0.08	0.05	0.10	0.03	0.82	0.08	0.05	0.10	0.03	0.82	0.08	0.05	0.10	0.03	0.82	0.08
0.37	0.41	0.74	0.03	0.63	0.37	0.41	0.74	0.03	0.63	0.37	0.41	0.74		0.63	0.37	0.41	0.74		0.63
0.58	0.72			0.83	0.58	0.72			0.83	0.58	0.72			0.83	0.58	0.72			0.83
0.44	0.91	0.39	0.55	0.93	0.44	0.91	0.39	0.55	0.93	0.44	0.91	0.39	0.55	0.93	0.44	0.91		0.55	0.93
0.73	0.08	0.71	0.01	0.28	0.73	0.08	0.71	0.01	0.28	0.73	0.08	0.71	0.01	0.28	0.73	0.08	0.71	0.01	0.28

Fonte: Figura retirada da referência [3].

A Figura 1 mostra um esquema ilustrativo do modelo de percolação invasiva em

uma rede quadrada com  $L = 5$ , a fim de facilitar a compreensão. Nesse exemplo, após preencher aleatoriamente a matriz com valores entre 0 e 1, e definir o sítio central como invasor (célula completamente preta), o processo inicia-se identificando os elementos da fronteira (valores na cor vermelha), isto é, os defensores (valores na cor preta) que são vizinhos dos invasores. Em seguida, o elemento da fronteira com o menor valor é invadido. Então, a fronteira é atualizada adicionando novos elementos, e o processo se repete invadindo o menor elemento da fronteira, e assim por diante. O Algoritmo 1, a seguir, descreve o modelo de percolação invasiva.

---

**Algoritmo 1:** PERCOLAÇÃO INVASIVA

---

```

1 início
2   Preencher a matriz aleatoriamente:  $r_i \in [0, 1]$ ;
3   Definir invasores;
4   Definir fronteira;
5   enquanto o sistema não percolar faça
6     Invadir o menor sítio da fronteira;
7     Atualizar fronteira;
8   fim
9 fim

```

---

Neste trabalho utilizamos uma estrutura de dados em árvore binária para identificar o sítio pertencente à fronteira que será invadido a cada passo de tempo. Tal algoritmo que utiliza a estrutura de árvore binária será explicado em detalhes no próximo capítulo (Capítulo 2). Vale destacar que esse algoritmo faz com que as simulações de percolação invasiva sejam realizadas de forma bastante rápida e eficiente [10, 6].

Diferentemente do modelo tradicional de percolação, em que deve-se procurar a probabilidade crítica de percolação,  $p_c$ , na percolação invasiva isso não é necessário, visto que esse modelo tem criticalidade auto-organizada [11], ou seja, há uma auto-regulação que leva a convergência natural para a probabilidade crítica de ocupação; a própria dinâmica do sistema escolhe uma estrutura crítica, onde naturalmente são excluídos aqueles sítios cujas probabilidades estão acima do valor crítico de percolação [2]. Podemos observar esse fato a partir da Figura 2, que mostra a distribuição normalizada de probabilidade  $P_{inv}(r_i)$  dos sítios invadidos, isto é, a probabilidade de que um sítio de valor  $r_i$  seja invadido. Essa distribuição exhibe, de maneira esperada, uma transição em  $p \approx p_c$ , onde  $p_c = 0.59275$  [12] é a probabilidade crítica de percolação para uma rede quadrada [13]. Em outras palavras, a criticalidade auto-organizada significa que o sistema evolui espontaneamente para a criticalidade [4]. Essa característica faz com que a relação entre a massa (ou número de sítios) do agregado invasor,  $M$ , e o número total

Figura 2 – Distribuição normalizada de probabilidade dos sítios invadidos.

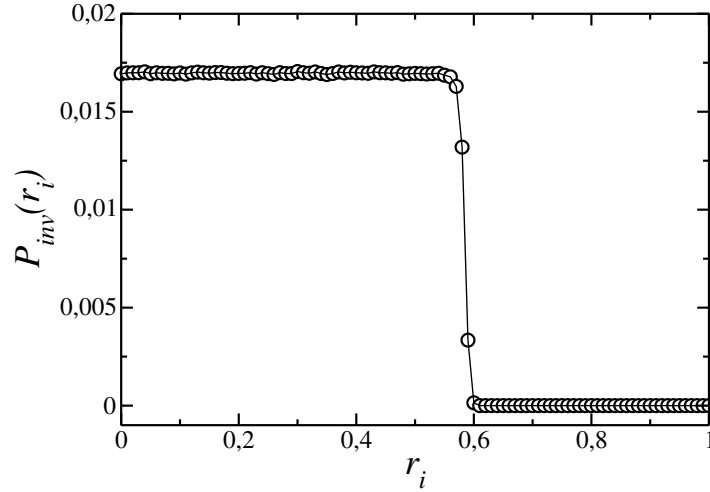


Gráfico da distribuição normalizada da probabilidade de invasão obtido a partir de 200 realizações em uma rede quadrada bidimensional de tamanho  $L = 512$ . Fonte: Elaborado pelo autor.

de elementos (ou sítios) da rede,  $N$ , obedeça a seguinte lei de potência [2, 6]

$$M(N) \propto N^{D_f/d_E}, \quad (1.1)$$

onde  $D_f$  é a dimensão fractal do agregado invasor e  $d_E$  é a dimensão euclidiana da rede. Considerando uma rede quadrada regular de tamanho  $L$ , o número total de elementos da rede será  $N = L^{d_E}$ , assim podemos reescrever a relação 1.2 em função do tamanho da rede como

$$M(L) \propto L^{D_f}. \quad (1.2)$$

Assim, os sítios invadidos se comportam da mesma maneira que o agregado percolante no ponto crítico ( $p_c$ ) [4], pois de acordo com a teoria de percolação, a massa (ou número de elementos) do maior agregado está relacionada com o tamanho do sistema através da seguinte relação:

$$M(L, p) \propto \begin{cases} \ln L, & \text{se } p < p_c, \\ L^{D_f}, & \text{se } p = p_c, \\ L^{d_E}, & \text{se } p > p_c, \end{cases} \quad (1.3)$$

onde  $p$  é a probabilidade de ocupação na teoria de percolação e  $d_E$  é a dimensão euclidiana da rede.

Há dois modelos de percolação invasiva. O primeiro modelo tenta descrever o comportamento de um fluido infinitamente compressível, sendo chamado de modelo de *percolação invasiva sem aprisionamento* (em inglês, *No Trapping Invasion Percolation*) [4], que foi o modelo utilizado nas simulações apresentadas nas Figuras 3, 4 e 5a, onde a dimensão fractal (o expoente da relação 1.2) é  $D_f = 91/48 \simeq 1.8958 \simeq 1.89$ , para duas dimensões [14, 15]. Já o segundo modelo tenta descrever o comportamento de um fluido incompressível, ou seja, quando

há a formação permanente de bolhas do fluido defensor; este modelo é chamado de *percolação invasiva com aprisionamento* (em inglês, *Trapping Invasion Percolation*) (Figura 5b) e o expoente torna-se um pouco menor,  $D_f \simeq 1.82$  [4, 14], para duas dimensões. Esse decréscimo no valor da dimensão fractal está relacionado às regiões de aprisionamento, que são como “buracos” na zona invadida [2]. Para três dimensões, a dimensão fractal é  $D_f \simeq 2.53$  [10, 6] para ambos os casos, pois quase não existe aprisionamento [4, 10].

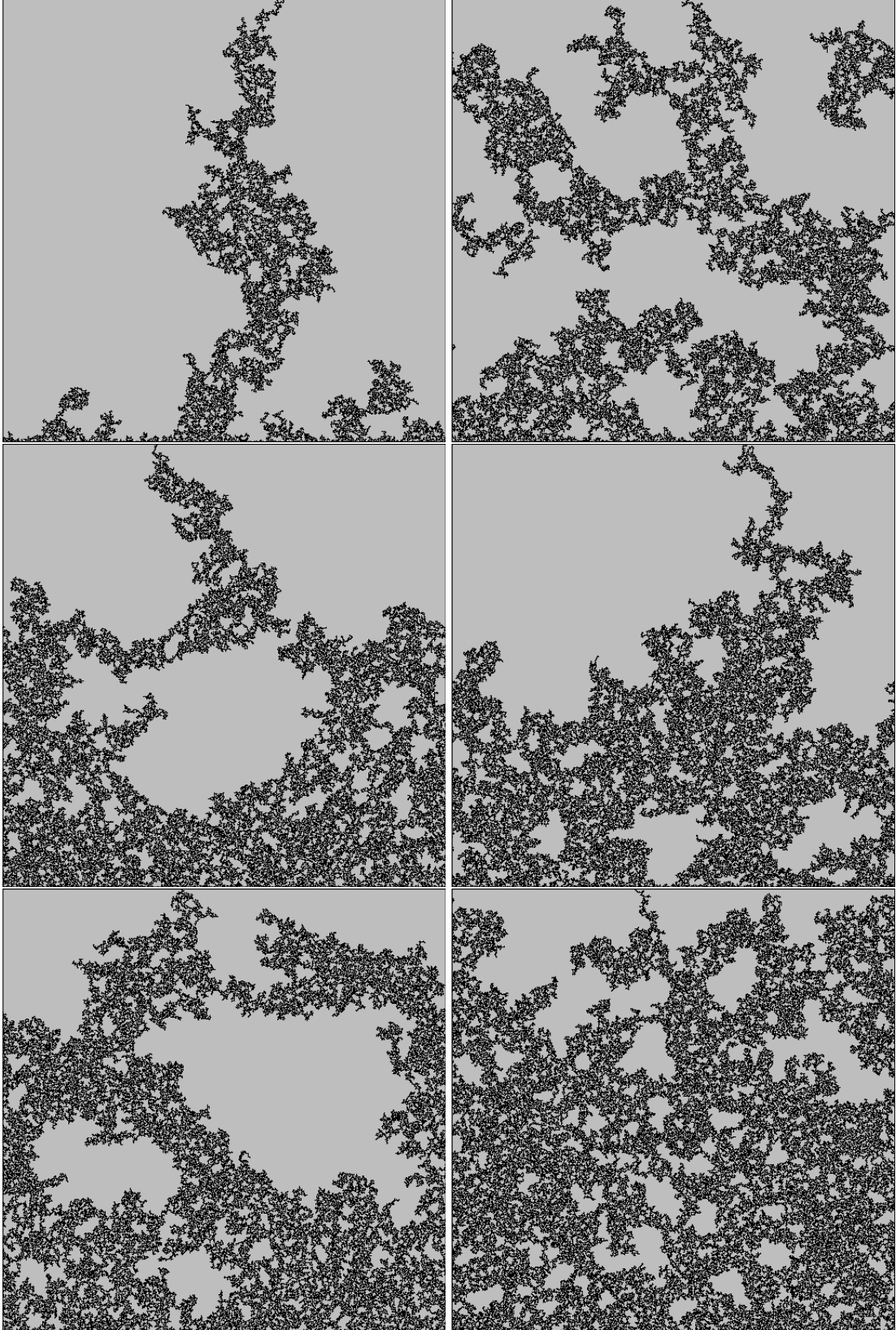
Nas simulações mostradas na Figura 3, para o modelo de percolação invasiva sem aprisionamento, considerou-se toda a primeira linha de uma rede quadrada, de tamanho  $L = 512$ , como sendo sítios invasores (células em preto), então após o sistema evoluir de acordo com as regras anteriormente citadas, a dinâmica foi encerrada quando o fluido invasor percola o sistema, ou seja, toca um sítio defensor (células em cinza) da última linha, formando um caminho que conecta o início ao fim da rede. Além disso, foram aplicadas condições de contorno periódicas nas bordas laterais da rede. Já nas simulações mostradas na Figura 4, para redes de diversos tamanhos, o processo de invasão se deu a partir de uma célula central e a dinâmica foi encerrada quando o fluido invasor percolou em qualquer direção. Nessas simulações temos o fluido invasor na cor preta, o fluido defensor na cor branca e a fronteira em vermelho.

Para encontrar, numericamente, a dimensão fractal  $D_f$  do agregado invasor, pode-se utilizar dois métodos principais, a saber, o *método do escalonamento de tamanho finito* e o *método do raio de giração*. O método do escalonamento de tamanho finito, consiste em medir o número de elementos ou a massa  $M(L)$  do agregado invasor para vários (e cada vez maiores) tamanhos  $L$  do sistema. Então, para um dado espaço amostral, o valor do expoente  $D_f$  é obtido através da inclinação do gráfico de  $M$  contra  $L$  em escala logarítmica [4]. Como mostra a Figura 6, através das simulações realizadas, foi obtido o expoente  $D_f = 1.89378$  para o modelo de percolação invasiva sem aprisionamento em duas dimensões, com um erro de  $\pm 0.00125$ . Utilizando o método do escalonamento de tamanho finito, calculamos também dimensão fractal da massa (ou número de elementos) da fronteira; e como mostra a Figura 7, foi obtido o expoente  $d_f = 1.89136 \pm 0.00154$  para a massa da fronteira do modelo de percolação invasiva sem aprisionamento em duas dimensões. Na Tabela 1, a seguir, temos os valores de cada ponto dos gráficos das Figuras 6 e 7, obtidos a partir do método de escalonamento de tamanho finito.

Na Figura 8, temos o gráfico massa do agregado invasor em função do tamanho da rede, para o modelo de percolação invasiva sem aprisionamento em três dimensões. A dimensão fractal, obtida através do método de escalonamento de tamanho finito, para três dimensões, foi de  $D_f = 2.4933 \pm 0.0020$ . Os dados da Figura 8 estão registrados na Tabela 2.

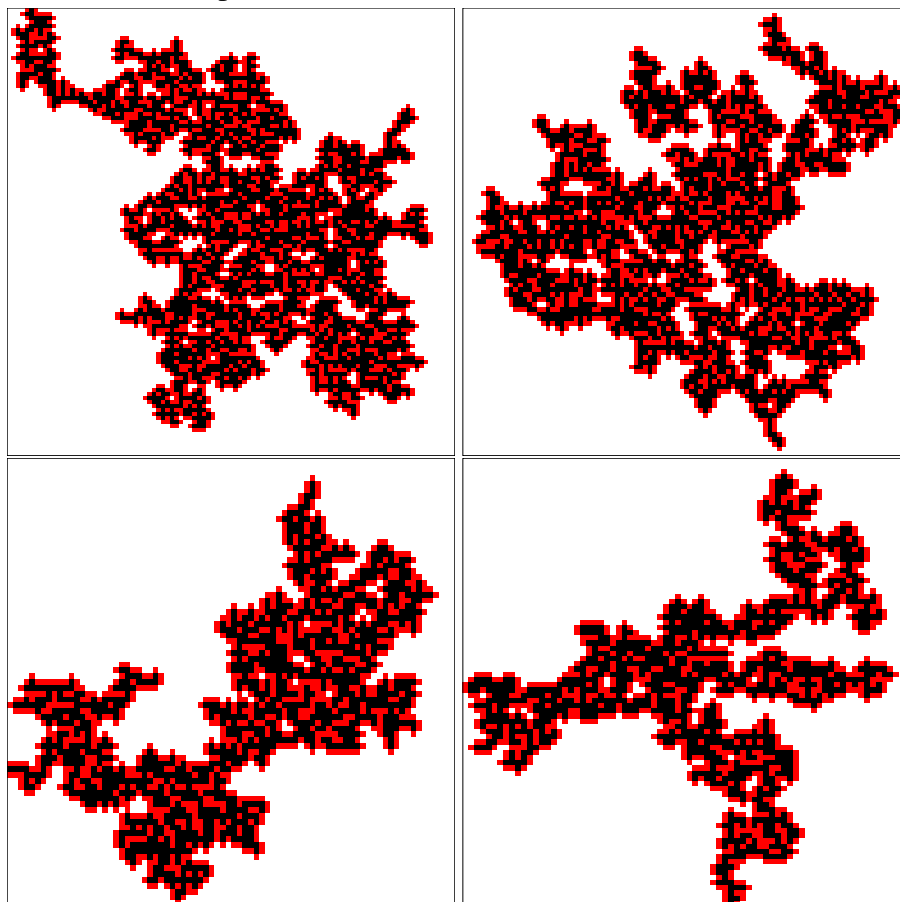
Já na Figura 9, temos o gráfico massa do agregado invasor em função do tamanho da rede, para o modelo de percolação invasiva com aprisionamento em duas dimensões. A

Figura 3 – Exemplos de percolação invasiva sem aprisionamento para uma rede de lado  $L = 512$ .



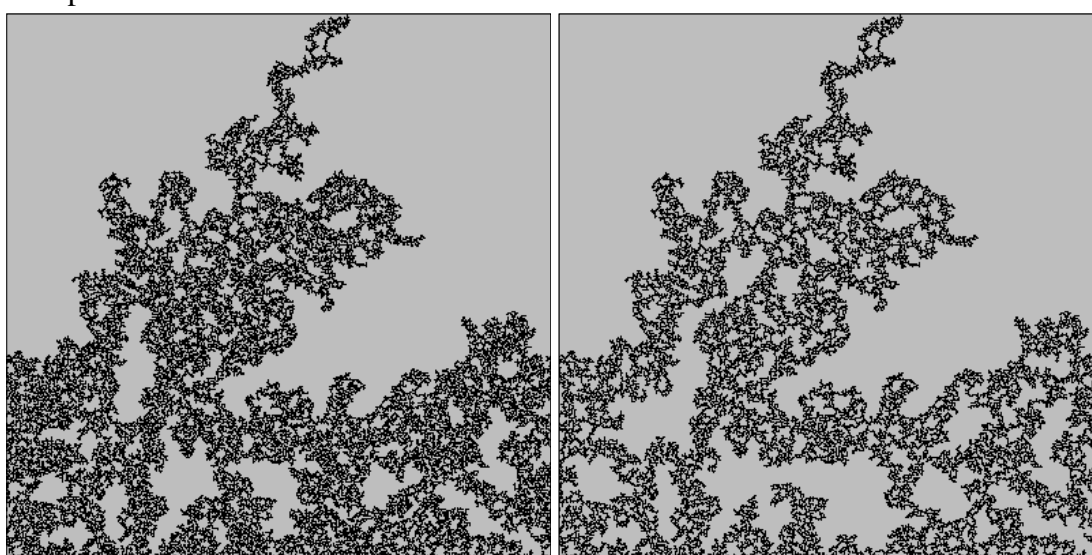
Em preto estão os sítios invasores e em cinza, os defensores. Fonte: Elaborado pelo autor.

Figura 4 – Exemplos de percolação invasiva sem aprisionamento com a invasão iniciada a partir de uma célula central, para redes de tamanhos diversos.



Em preto estão os sítios invasores, em branco, os defensores e em vermelho, a fronteira. Fonte: Elaborado pelo autor.

Figura 5 – Comparação visual entre os modelos de percolação invasiva com e sem aprisionamento para uma rede bidimensional de lado  $L = 400$ .



(a) Sem aprisionamento.

(b) Com aprisionamento.

Em preto estão os sítios invasores e em cinza, os defensores. Fonte: Elaborado pelo autor.



Figura 6 – Massa do agregado invasor em função do tamanho da rede, para duas dimensões.

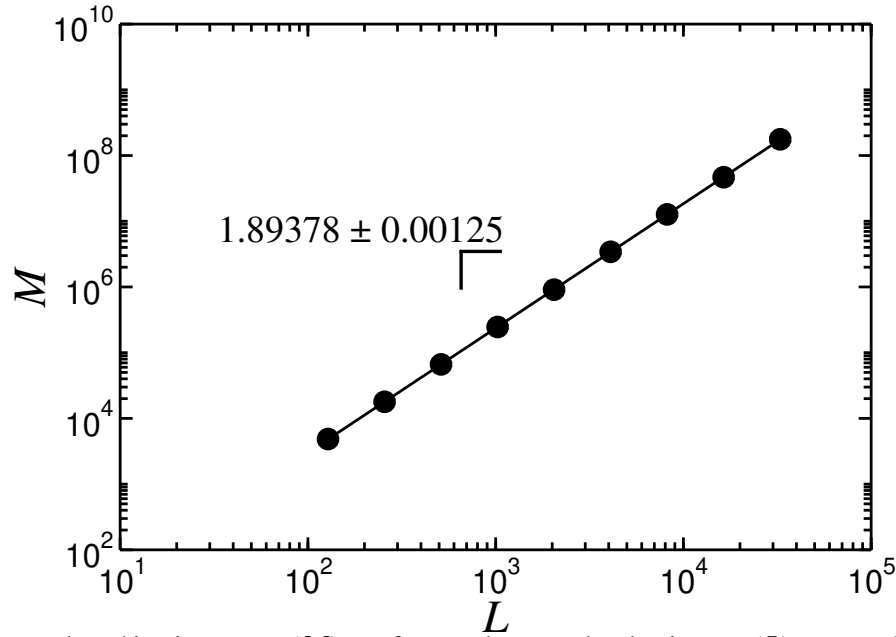


Gráfico da massa dos sítios invasores ( $M$ ) em função do tamanho do sistema ( $L$ ) em escala logarítmica, para o modelo de percolação invasiva sem aprisionamento em duas dimensões. Foram realizadas várias simulações de sistemas de diversos tamanhos conforme detalhado na Tabela 1. O expoente da lei de potência da Equação 1.2, obtido através do método de mínimos quadrados, foi  $D_f = 1.89378 \pm 0.00125$  ( $M = 0.49195L^{1.89378}$ ). Fonte: Elaborado pelo autor.

Figura 7 – Massa da fronteira em função do tamanho da rede, para duas dimensões.

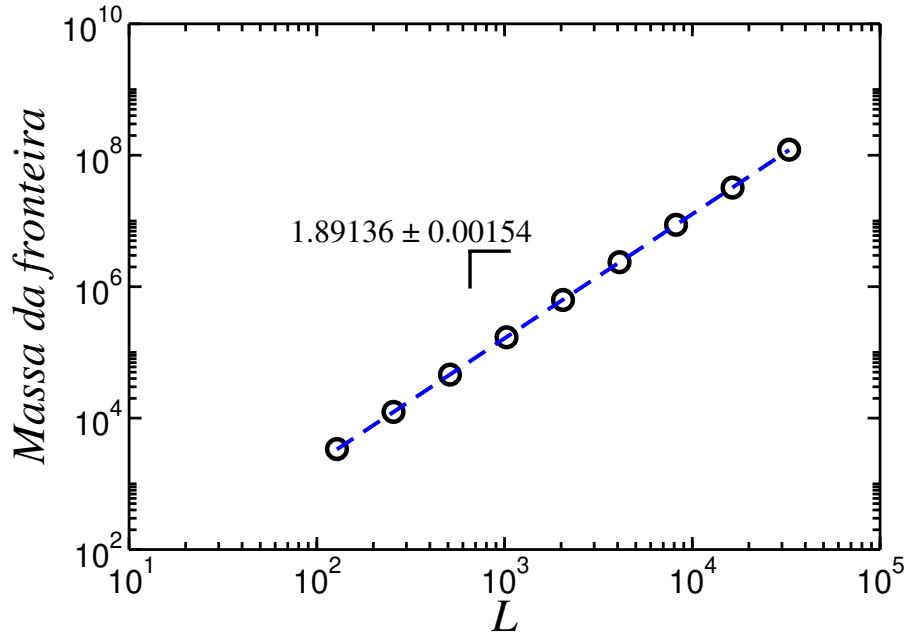


Gráfico da massa da fronteira em função do tamanho do sistema ( $L$ ) em escala logarítmica, para o modelo de percolação invasiva sem aprisionamento em duas dimensões. Foram realizadas várias simulações de sistemas de diversos tamanhos conforme detalhado na Tabela 1. O expoente da lei de potência da Equação 1.2, obtido através do método de mínimos quadrados, foi  $d_f = 1.89136 \pm 0.00154$  ( $M = 0.34603L^{1.89136}$ ). Fonte: Elaborado pelo autor.

dimensão fractal, também obtida através do método de escalonamento de tamanho finito, foi de  $D_f = 1.8204 \pm 0.0040$ . Os dados de cada ponto da Figura 9 estão registrados na Tabela 3.

Tabela 1: Massa do agregado invasor e massa da fronteira em função do tamanho da rede, para o modelo de percolação invasiva sem aprisionamento em duas dimensões (dados das Figuras 6 e 7).

$L$	$M(L)$	$M\text{-}front(L)$	$Amostras$
128	4847.17	3382.36	25601
256	17906.3	12429.4	12800
512	66296.5	45866.4	12800
1024	246128	169850	12400
2048	910086	627057	9950
4096	3426720	2358240	2201
8192	12698400	8745660	2250
16384	46736600	32136000	2400
32768	176692000	121447000	751

Figura 8 – Massa do agregado invasor em função do tamanho da rede, para três dimensões.

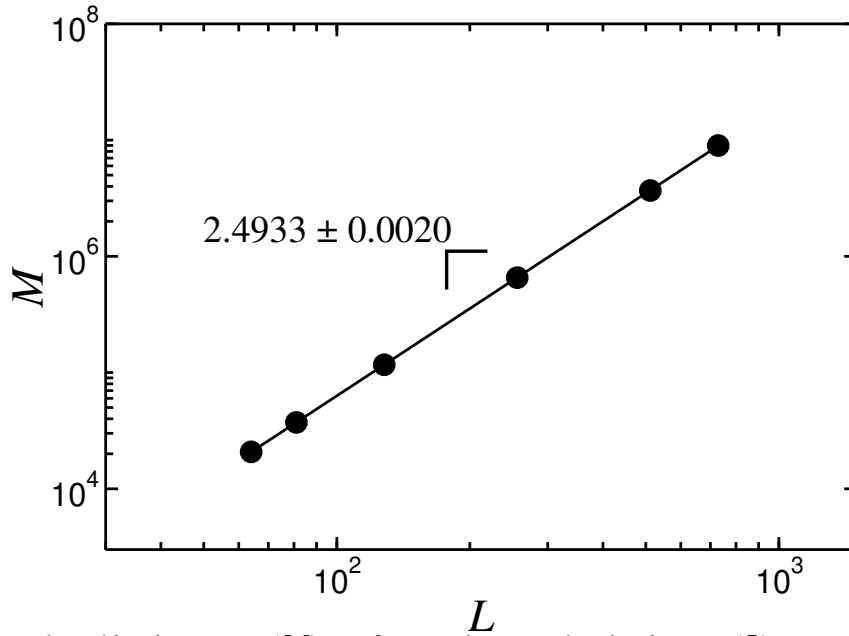


Gráfico da massa dos sítios invasores ( $M$ ) em função do tamanho do sistema ( $L$ ) em escala logarítmica, para o modelo de percolação invasiva sem aprisionamento em três dimensões. Foram realizadas várias simulações de sistemas de diversos tamanhos conforme detalhado na Tabela 2. O expoente da lei de potência da Equação 1.2, obtido através do método de mínimos quadrados, foi  $D_f = 2.4933 \pm 0.0020$  ( $M = 0.65059L^{2.4933}$ ). Fonte: Elaborado pelo autor.

O segundo método que pode ser utilizado para calcular a dimensão fractal é o método do raio de giração. O raio de giração  $R$  do agregado invasor, para cada passo de tempo, é definido como [4]:

$$R^2 = \sum_{i=1}^m \frac{|\vec{r}_i - \vec{r}_0|^2}{m}, \quad (1.4)$$

onde  $m = M(L, t)$  é a massa do agregado invasor, no passo de tempo  $t$ ,  $\vec{r}_i$  é a posição do  $i$ -ésimo elemento pertencente ao agregado e  $\vec{r}_0$  é a posição do centro de massa do agregado, no

Tabela 2: Massa do agregado invasor e massa da fronteira em função do tamanho da rede, para o modelo de percolação invasiva sem aprisionamento em três dimensões (dados da Figura 8).

$L$	$M(L)$	$M\text{-}front(L)$	$Amostras$
64	20782.4	33010.3	60001
81	37233.3	61595.1	40001
128	116713	205852	20001
256	655672	1239830	10001
512	3681170	7295860	5001
729	8984770	18150500	3001

Figura 9 – Massa do agregado invasor em função do tamanho da rede, para o modelo de percolação invasiva com aprisionamento em duas dimensões.

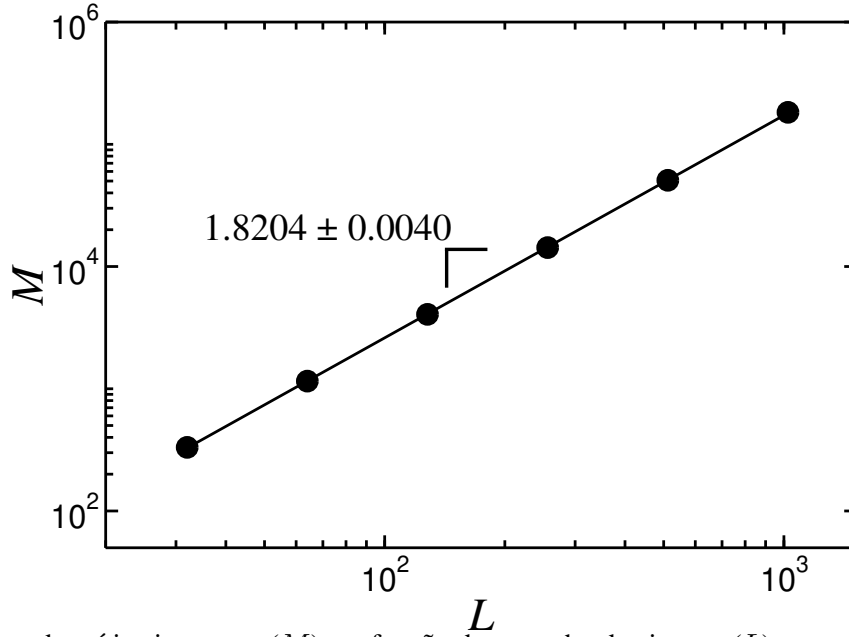


Gráfico da massa dos sítios invasores ( $M$ ) em função do tamanho do sistema ( $L$ ) em escala logarítmica, para o modelo de percolação invasiva com aprisionamento em duas dimensões. Foram realizadas várias simulações de sistemas de diversos tamanhos conforme detalhado na Tabela 3. O expoente da lei de potência da Equação 1.2, obtido através do método de mínimos quadrados, foi  $D_f = 1.8204 \pm 0.0040$  ( $M = 0.5963L^{1.8204}$ ). Fonte: Elaborado pelo autor.

passo de tempo  $t$ , dada por,

$$\vec{r}_0 = \sum_{i=1}^m \frac{\vec{r}_i}{m}. \quad (1.5)$$

Considerando a estatística sobre várias amostras, no limite termodinâmico, tem-se que

$$M = \langle m \rangle \propto \langle R^{D_f} \rangle. \quad (1.6)$$

Outra quantidade de interesse no modelo de percolação invasiva (com aprisionamento) é a envoltória de percolação  $h(L)$ , que é definida como os sítios “externos” ao agregado invasor, isto é, toda a fronteira externa ao agregado invasor. Essa quantidade também escala com o tamanho do sistema  $L$  ( $h(L) \propto L^{d_f}$ ) e é identificada como um fractal de dimensão  $d_f$  [4].

Tabela 3: Massa do agregado invasor em função do tamanho da rede, para o modelo de percolação invasiva com aprisionamento em duas dimensões (dados da Figura 9).

$L$	$M(L)$	$Amostras$
32	4847.17	40001
64	4847.17	20001
128	4847.17	10001
256	17906.3	5001
512	66296.5	1901
1024	246128	90

Dependendo dos detalhes da definição exata do perímetro externo, obtém-se  $d_f = 7/4 = 1.75$  [15, 16] ou  $d_f = 4/3 \simeq 1.33$  [15, 16, 17].

### 1.1 Avalanches

No processo de invasão estão presentes as *avalanches*, que ocorrem quando há uma invasão sequencial de sítios menores [2]. Uma avalanche pode ser definida de duas maneiras. A primeira definição consiste em chamar de avalanche um agregado conectado de sítios que foram adicionados durante um período de tempo contínuo; essa definição, embora mais natural, é difícil de se usar experimentalmente [14]. Em resumo, isso ocorre ao invadir um sítio  $i$  de raio  $r_i$ , e em seguida também invadir uma série de outros sítios interconectados a partir do sítio  $i$  ( $i \rightarrow j \rightarrow k \rightarrow l \rightarrow \dots$ ) cujos raios sejam cada vez menores do que  $r_i$  ( $r_i > r_j > r_k > r_l > \dots$ ), para cada sítio invadido. Essas invasões sequenciais podem ser bastante frequentes (embora não muito extensas) na percolação invasiva, uma vez que, após invadir um sítio  $i$  existe uma certa probabilidade de encontrar sítios com raios menores nos novos sítios de crescimento (isto é, os sítios defensores vizinhos do sítio  $i$  que agora farão parte da fronteira) do que no restante da fronteira. O número  $n$  de sítios invadidos em uma invasão sequencial é a extensão dessa invasão. A Figura 10 mostra os histogramas (em escala semi-logarítmica) ( $P(n) \propto e^{-\alpha n}$ ) obtidos utilizando a primeira definição de avalanche.

De acordo com uma segunda definição, uma avalanche ocorre ao invadir um sítio  $i$  de raio  $r_i$ , e em seguida também invadir uma série de outros sítios (a partir de qualquer parte do agregado invasor) cujos raios sejam menores do que  $r_i$ . O número de sítios invadidos nessa sequência de invasões é o tamanho da avalanche,  $s$ . Avalanches não estão presentes apenas na percolação invasiva. Na verdade, pode-se calcular as avalanches em qualquer série temporal, da forma como se segue. Considerando uma série temporal  $X(t)$ , pode-se definir o tamanho  $s$  de uma avalanche (ou *burst*, em inglês [14]) como um intervalo de tempo  $\Delta t$  (ou número de passos de tempo) de  $X(t)$  que inicia-se em um ponto  $t = t_a$ , de valor (ou altura)  $X(t_a) = H$  e termina em um ponto  $t = t_b$ , de altura igual ou superior à  $H$ , isto é,  $X(t_b) \geq H$ , tal que  $t_a < t_b$

Figura 10 – Invasões sequenciais - primeira definição de avalanche.

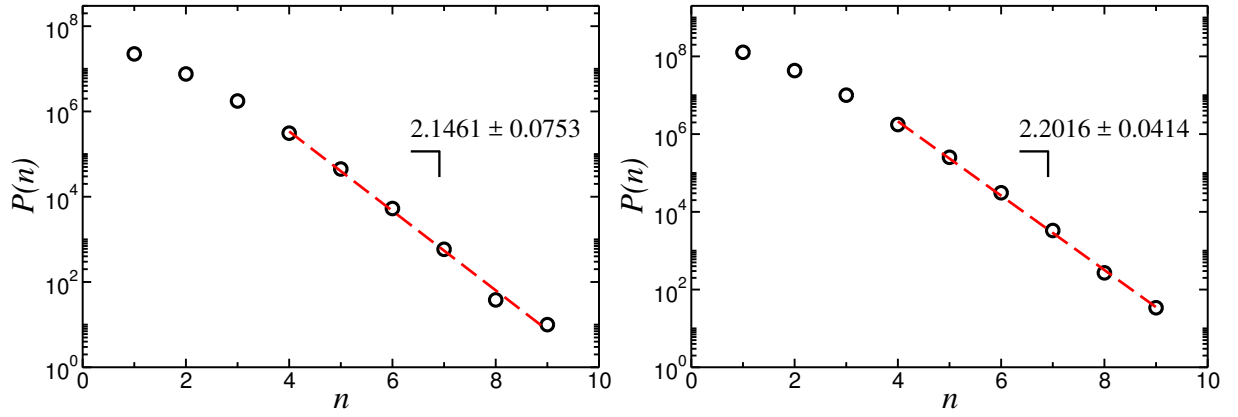
(a) 500 realizações de uma rede com  $L = 1024$ .(b) 50 realizações de uma rede com  $L = 8192$ .

Gráfico do histograma (em escala semi-logarítmica) dos tamanhos das avalanches (utilizando a primeira definição). Fonte: Elaborado pelo autor.

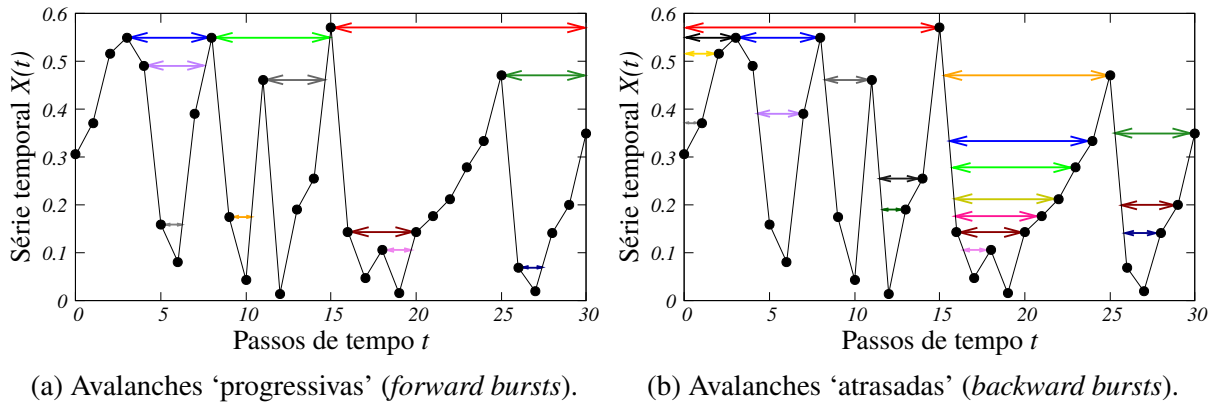
e  $X(t) < H, \forall t \in (t_a, t_b)$ . Assim, tendo em vista essas considerações, o tamanho da avalanche que inicia-se no ponto  $t_a$  é o número de passos de tempo entre  $t_a$  e  $t_b$ , isto é, no intervalo aberto  $(t_a, t_b)$ ; e pode ser escrito como

$$s(t_a) = \Delta t(t_a, t_b) = \frac{1}{\delta t}(t_b - t_a) - 1, \quad (1.7)$$

onde  $\delta t = t_{i+1} - t_i$  é o tamanho do passo de tempo e tem valor fixo, e a subtração de uma unidade ( $-1$ ) se deve ao fato de que o ponto  $t_b$  não entra na contagem das avalanches, já que  $X(t_b) \geq H$ , por definição. Vale ressaltar que as avalanches são calculadas para cada ponto (ou valor) da série temporal. A avalanche, do modo como foi definida acima, é chamada de avalanche ‘progressiva’ (em inglês, *forward bursts*), uma vez que a avalanche em um dado ponto  $t$  é calculada levando em conta os pontos posteriores. Se, por outro lado, levarmos em conta os pontos anteriores ao ponto  $t$ , no cálculo da avalanche, então estaremos utilizando avalanches ‘atrasadas’ (em inglês, *backward bursts*). Para uma melhor compreensão, a Figura 11 ilustra como é calculado o tamanho das avalanches (‘progressivas’ e ‘atrasadas’), onde as setas mostram onde ocorrem as avalanches e indicam seus respectivos tamanhos.

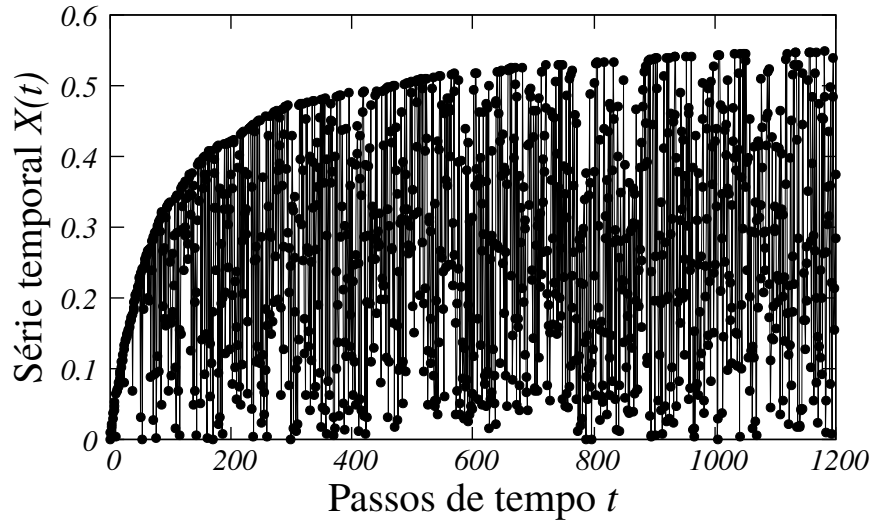
Na percolação invasiva, a série temporal  $X(t)$  carrega os valores dos raios  $r_i$  de todos os sítios invadidos em ordem cronológica (Figura 12). Dessa forma,  $X(t_j) = r_j$  é o valor do raio do sítio invadido no tempo  $t_j$ ,  $X(t_{j+1}) = r_{j+1}$  é o raio do sítio invadido no tempo seguinte  $t_{j+1}$ , e assim por diante. Na percolação invasiva, a distribuição de avalanches também apresenta um comportamento em lei de potência [2, 3],  $P(s) \propto s^{-\tau}$ , onde  $\tau = 139/91 \simeq 1.527$  para redes bidimensionais e considerando as avalanches ‘atrasadas’ [14, 13]. A Figura 13 mostra os histogramas dos tamanhos das avalanches obtidos em nossas simulações de percolação invasiva sem aprisionamento. Como se pode ver, obtivemos um expoente de  $\tau = 1.4582 \pm 0.0133$

Figura 11 – Definições de avalanche.



As setas mostram onde ocorrem as avalanches e indicam seus respectivos tamanhos. Fonte: Elaborado pelo autor.

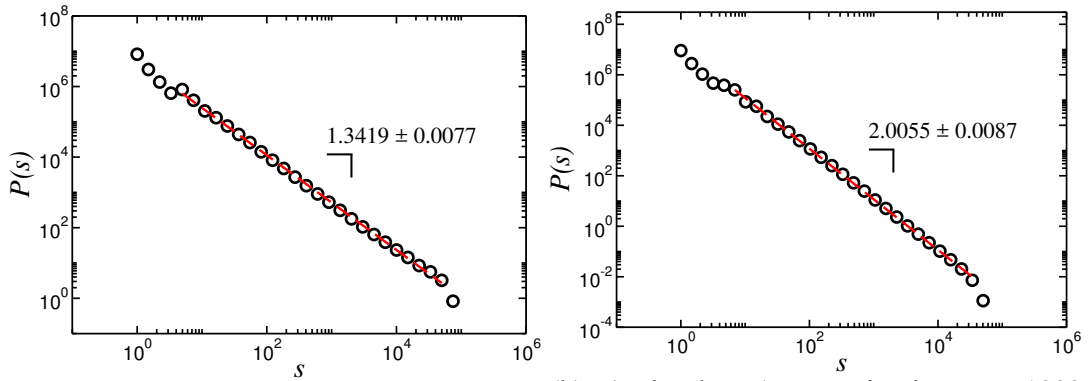
Figura 12 – Trecho inicial da série temporal na percolação invasiva.



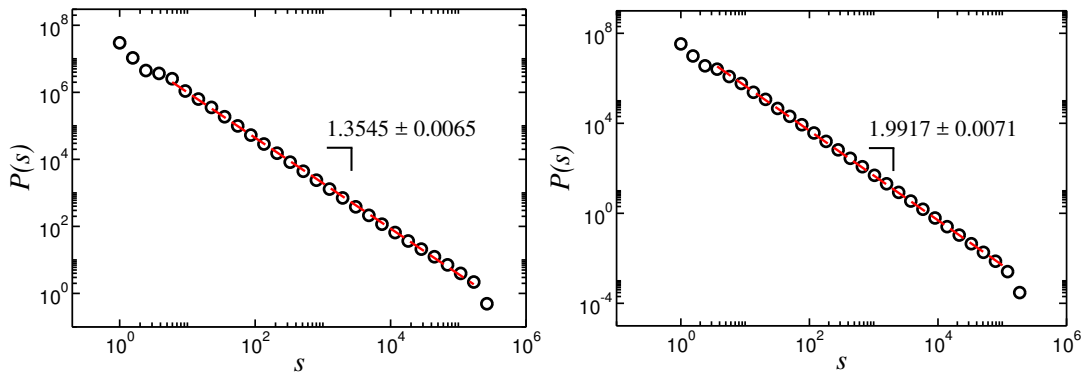
A série temporal  $X(t)$  carrega os valores dos raios  $r_i$  de todos os sítios invadidos. Fonte: Elaborado pelo autor.

para avalanches ‘atrasadas’, com 10 realizações em uma rede de tamanho  $L = 16384$ ; e  $\tau = 2.0050 \pm 0.0052$  para avalanches ‘progressivas’, também com 10 realizações em uma rede de tamanho  $L = 16384$ .

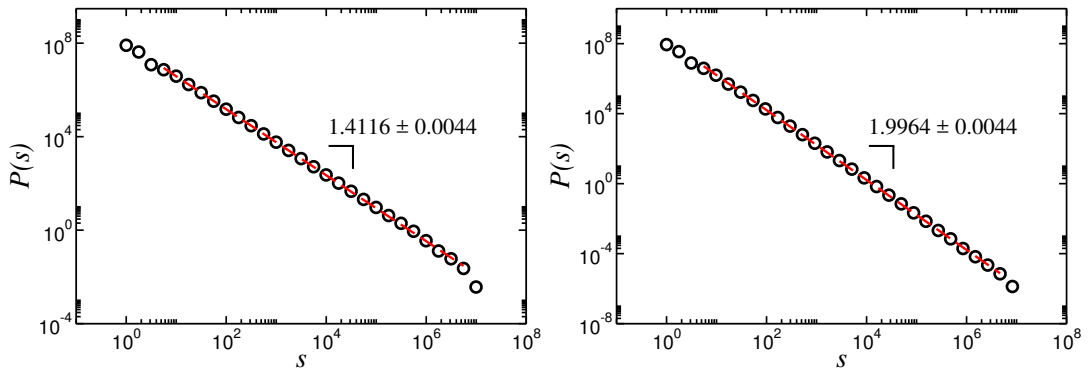
Figura 13 – Distribuição das avalanches.



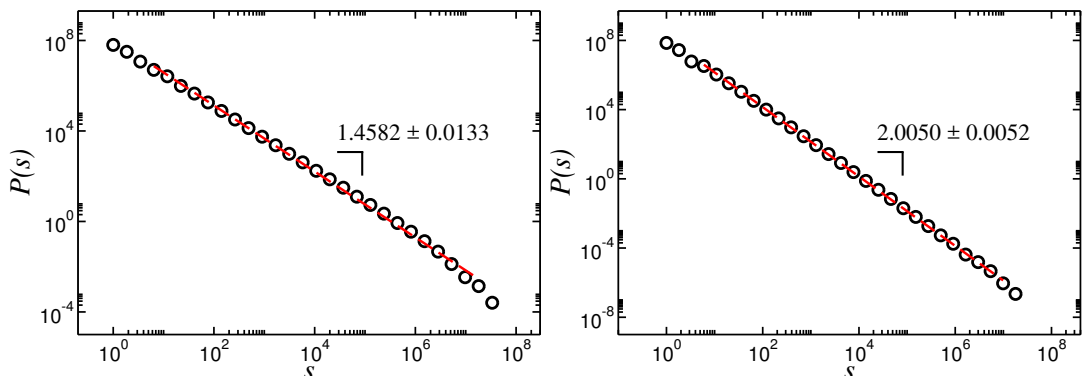
(a) Avalanches ‘atrasadas’, para 1000 realizações de uma rede com  $L = 512$ . (b) Avalanches ‘progressivas’, para 1000 realizações de uma rede com  $L = 512$ .



(c) Avalanches ‘atrasadas’, para 1000 realizações de uma rede com  $L = 1024$ . (d) Avalanches ‘progressivas’, para 1000 realizações de uma rede com  $L = 1024$ .



(e) Avalanches ‘atrasadas’, para 50 realizações de uma rede com  $L = 8192$ . (f) Avalanches ‘progressivas’, para 50 realizações de uma rede com  $L = 8192$ .



(g) Avalanches ‘atrasadas’, para 10 realizações de uma rede com  $L = 16384$ . (h) Avalanches ‘progressivas’, para 10 realizações de uma rede com  $L = 16384$ .

Gráfico dos histogramas logarítmicos dos tamanhos das avalanches (segunda definição), para a percolação invasiva sem aprisionamento em redes quadradas bidimensionais. Fonte: Elaborado pelo autor.

## 2 ALGORITMO PRIORITY HEAP

A descrição do algoritmo *Priority Heap*, a seguir, foi baseada na referência [18].

### 2.1 Árvore binária armazenada em um vetor

Suponha que queremos rearranjar um dado um vetor  $A[1, \dots, n]$  utilizando a estrutura de árvore binária. Por enquanto, deixaremos de lado os valores armazenados no vetor ( $A[1], \dots, A[i], \dots, A[n]$ ), e nos ateremos a certas relações entre os índices  $(1, \dots, i, \dots, n)$  do vetor. Para todo índice  $i$ , temos a seguinte relação (como ilustrado na Figura 14):

$$\begin{aligned} \text{int}(i/2) & \text{ é o pai do índice } i, \text{ se } i \neq 1 \\ 2i & \text{ é o filho esquerdo de } i, \text{ se } 2i \leq n \\ 2i + 1 & \text{ é o filho direito de } i, \text{ se } 2i + 1 \leq n, \end{aligned} \quad (2.1)$$

onde  $\text{int}(N)$  é a parte inteira do número  $N$ . Adotando essa nomenclatura de pais e filhos e seguindo a relação 2.1, o vetor  $A$  adquire uma estrutura de árvore binária (Figura 15) quase completa e os elementos do vetor, identificados pelos índices 1 a  $n$ , passam a ser chamados *nós*. O primeiro nó (nó 1) será chamado de *raiz* da árvore; e qualquer nó  $i$  é raiz da subárvore formada por  $i$ , seus filhos, seus netos, etc. Ou seja, a subárvore com raiz  $i$  é o vetor:  $A[i, 2i, 2i + 1, 4i, 4i + 1, 4i + 2, 4i + 3, 8i, \dots, 8i + 7, \dots]$ .

Figura 14 – Relação entre os índices de uma árvore binária.

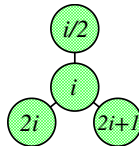


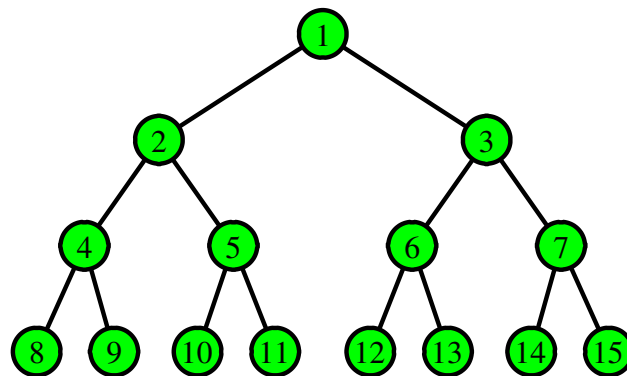
Ilustração da relação 2.1. Fonte: Elaborado pelo autor.

### 2.2 Min-heap

Em termos bem simples, um *min-heap* (ou árvore hierárquica) é uma árvore binária quase completa em que cada pai é menor ou igual que qualquer um de seus filhos. Ou seja, um vetor  $A[1, \dots, n]$  será um *min-heap* se  $A[\text{int}(i/2)] \leq A[i]$ , para  $i = 2, \dots, n$ . De outra forma,  $A[1, \dots, n]$  é um *min-heap* se  $A[j] \leq A[2j]$  e  $A[j] \leq A[2j + 1]$  sempre que os índices não ultrapassem  $n$ . Dessa forma, o menor valor armazenado no vetor estará sempre no primeiro elemento do vetor,  $A[1]$ , o que faz com que o heap seja uma estrutura de dados tão útil. Além disso, se  $A[1, \dots, n]$  é um *min-heap* e se o valor de  $A[1]$  for alterado, então este vetor pode ser ‘consertado’ muito rapidamente através do algoritmo que será descrito a seguir, o algoritmo



Figura 15 – Árvore binária.



Note que os índices do vetor (os números dentro dos círculos) obedecem a relação 2.1. Fonte: Elaborado pelo autor.

*Corrige-Descendo.* Da mesma forma, um vetor  $A[1, \dots, n]$  arbitrário pode ser transformado em um min-heap de forma bastante rápida.

A principal ferramenta de manipulação de um heap é o algoritmo que resolve o seguinte problema: *Dado um vetor  $A[1, \dots, n]$  e um índice  $i$  tal que a subárvore com raiz  $2i$  é um min-heap e a subárvore com raiz  $2i + 1$  é um min-heap, reorganizar o vetor de modo que a subárvore com raiz  $i$  seja um min-heap.* A ideia do algoritmo, que chamaremos de Corrige-Descendo, é simples: se  $A[i]$  é menor ou igual que seus filhos então não é preciso fazer nada; senão, troque  $A[i]$  com o menor dos filhos e repita o processo para o filho envolvido na troca. O algoritmo Corrige-Descendo está descrito com maiores detalhes a seguir (Algoritmo 2).

---

**Algoritmo 2:** CORRIGE-DESCENDO( $A, n, i$ )

---

```

1  início
2       $j \leftarrow i$ 
3      enquanto  $2j \leq n$  faça
4           $f \leftarrow 2j$ 
5          Se  $f < (n - 1)$  e  $A[f] > A[f + 1]$ 
6               $f \leftarrow f + 1$ 
7          Se  $A[j] \leq A[f]$ 
8               $j \leftarrow n$ 
9          Senão
10             troque  $A[j] \leftrightarrow A[f]$ 
11              $j \leftarrow f$ 
12     fim
13 fim
```

---

Outra ferramenta de manipulação de um heap é o algoritmo que resolve o seguinte

problema: *Dado um vetor  $A[1, \dots, m-1]$ , que já seja um min-heap, adicionar um novo elemento no vetor ( $A[m]$ ), e rearranjar o vetor de modo que  $A[1, \dots, m]$  que seja um min-heap.* A ideia desse algoritmo, que chamaremos de Corrige-Subindo, é a seguinte: se o  $\text{pai}(A[m])$ ,  $A[\text{int}(m/2)]$ , for menor ou igual que seu filho  $A[m]$  então não é preciso fazer nada; senão, troque  $A[m]$  com  $A[\text{int}(m/2)]$  e repita o processo para o  $\text{pai}(\text{pai}(A[m]))$  envolvido na troca, e assim por diante. O algoritmo Corrige-Subindo está descrito com maiores detalhes a seguir (Algoritmo 3).

---

**Algoritmo 3:** CORRIGE-SUBINDO( $A, m$ )

---

```

1 início
2    $i \leftarrow m$ 
3   enquanto  $i \geq 2$  e  $A[\text{int}(i/2)] > A[i]$  faça
4     troque  $A[\text{int}(i/2)] \leftrightarrow A[i]$ 
5      $i \leftarrow \text{int}(i/2)$ 
6   fim
7 fim
```

---

Para construir um min-heap a partir de um vetor  $A[1, \dots, n]$  arbitrário basta utilizar o algoritmo *Constrói-Min-Heap* 4, abaixo.

---

**Algoritmo 4:** CONSTRÓI-MIN-HEAP( $A, n$ )

---

```

1 início
2   para  $i \leftarrow \text{int}(n/2)$  decrescendo até 1 faça
3     Corrige-Descendo( $A, n, i$ )
4   fim
5 fim
```

---

## APÊNDICE A - CÓDIGO EM C++ DO MODELO DE PERCOLAÇÃO INVASIVA SEM APRISIONAMENTO

```

1 //Percolacao Invasiva Sem Aprisionamento com algoritmo Priority Heap;
2 //Percolacao de sitios em rede quadrada com primeiros vizinhos de Von
3 //Neuman, com condicoes de contorno periodico na horizontal;
4 //O vetor 'Heap' guarda o endereco (i,j) da fronteira M[i][j] (Heap[q]=i*W +j);
5
6 #include <iostream>
7 #include <vector>
8 #include <math.h>
9 #include <random> //C++11
10
11 using namespace std;
12
13 #define N 1024
14 #define def 2 //Defensor [def,def+1]
15 #define inv 4 //Invasor
16 //Fronteira [0,1]
17 bool percola;
18 int Massa_inv, Massa_front;
19 int const W=pow(10,1+int(log10(N)));
20 vector< vector<float> > M(N, vector<float>(N));
21 vector<unsigned int> Heap;
22
23 void Inicializar(int seed) {
24     std::mt19937 generator (seed);
25     std::uniform_real_distribution<double> dis(0.0, 1.0);
26     for(int i=0; i<N; i++)
27     { for(int j=0; j<N; j++)
28       { M[i][j] = def + dis(generator); } //Defensor;
29     }
30     //Cond. Inicial (M): primeira Linha;
31     for(int j=0; j<N; j++) {M[0][j] = inv;} //Invasor;
32 }
33 struct pt { int x; int y; };
34 struct pt V[4];
35 void Vizinhos(int i, int j) {
36     V[0].x = i-1; V[0].y = j; //bottom
37     V[1].x = i; V[1].y = j-1; //left
38     V[2].x = i; V[2].y = j+1; //righth
39     V[3].x = i+1; V[3].y = j; //top
40     //Sem periodicidade na vertical;
41     if(V[0].x== -1){V[0].x=0;} //bottom
42     if(V[3].x== N){V[3].x=N-1;} //top
43     //Com periodicidade na horizontal;
44     if(V[1].y== -1){V[1].y=N-1;} //left
45     if(V[2].y== N){V[2].y=0;} //righth

```

```

46 }
47 int I(int i) {return int(Heap[i]/W);}
48 int J(int i) {return (Heap[i]-W*I(i));}
49 void Corrige_Subindo(int m) {
50     int i=m, aux;
51     //Enquanto nao for o primeiro item & o pai for maior;
52     while((i>=1) & (M[I(int(0.5*i))][J(int(0.5*i))]>M[I(i)][J(i)])) {
53         //Trocar: Heap[int(i/2)]->Heap[i];
54         aux=Heap[int(0.5*i)]; Heap[int(0.5*i)]=Heap[i];
55         Heap[i]=aux; i=int(0.5*i);
56     }
57 }
58 void Corrige_Descendo(int i) {
59     int f;    //filhos;
60     int j=i;  //pai;
61     int aux, q=Heap.size();
62     while(2*j < q) { //Enquanto nao chegar no ultimo elemento...
63         f = 2*j;
64         if(f<q-1){ if(M[I(f)][J(f)]>M[I(f+1)][J(f+1)]){f++;} } //Ver qual filho e' maior;
65         if(M[I(j)][J(j)]<=M[I(f)][J(f)]) {j=f;} //Sair do loop;
66         else //Trocar: Heap[j]->Heap[f];
67         { aux=Heap[j]; Heap[j]=Heap[f]; Heap[f]=aux; j=f; }
68     }
69 }
70 void Remover() {
71     Heap[0]=Heap[Heap.size()-1];
72     Heap.pop_back(); //Remover o ultimo elemento do vetor;
73     Corrige_Descendo(0);
74 }
75 void Inserir(int I) {
76     if(Heap.size()<N*N){
77         Heap.push_back(I); //Inserir no ultimo elemento do vetor;
78         Corrige_Subindo(Heap.size()-1);
79     } else{cout<<"\n\tERRO!\n";}
80 }
81 void Invadir() {
82     int x, y, I, J;
83     I=int(Heap[0]/W); J=Heap[0] - W*I;
84     M[I][J]=inv; Remover(); Massa_inv++; Massa_front--;
85
86     if(I==N-1){percola=true;}
87     Vizinhos(I,J); //Atualiza V[4] recebendo os vizinhos de M[I][J];
88     for(int k=0; k<4; k++)
89     { x=V[k].x; y=V[k].y;
90         if( (M[x][y]>=def) & (M[x][y]<=def+1) ) //Se for Defensor...;
91         { M[x][y]-=def; Inserir(x*W + y); Massa_front++; } //Fronteira;
92     }
93 }
94 void Analisar() {
95     int x, y;
96     bool stop=0; //limita a analise [p/ cond.inicial na primeira linha];
97
98     for(int i=0; i<N; i++)
99     { stop=1;

```

```

100     for(int j=0; j<N; j++)
101     {   if(M[i][j]==inv)
102         { stop=0; Massa_inv++;
103           Vizinhos(i,j); //Atualiza V[4] recebendo os vizinhos de M[i][j];
104           for(int k=0; k<4; k++)
105           {   x=V[k].x; y=V[k].y;
106               if( (M[x][y]>=def) & (M[x][y]<=def+1) ) //Se for Defensor...
107               { M[x][y]-=def; Inserir(x*W + y); Massa_front++; } //Fronteira;
108           }
109         }
110     }   if(stop==1){i=N;}
111 }
112 }
113 int main(int argc, char* argv[]){
114     int seed = atoi(argv[1]);
115     Massa_inv=0; Massa_front=0;
116     Inicializar(seed); //Preencher Matriz com defensores e definir invasores;
117     Analisar();        //Definir a Fronteira ->[colocar na Heap];
118
119     percola=false;
120     do{ Invadir(); }while(percola==false);
121     cout <<Massa_inv<<"\t"<<Massa_front<<"\t"<<seed<<"\n";
122     return 0;
123 }

```

## REFERÊNCIAS

- [1] WILKINSON, D.; WILLEMSSEN, J. F. Invasion percolation: a new form of percolation theory. *Journal of Physics A: Mathematical and General*, IOP Publishing, v. 16, n. 14, p. 3365, 1983.
- [2] OLIVEIRA, C. L. N. d. Escoamento bifásico em meios porosos: aplicações na recuperação de óleo e infiltração de fluidos adesivos. *Tese de Doutorado, Universidade Federal do Ceará*, 2009.
- [3] ROMEU, M. C. Percolação invasiva entre múltiplos poços. *Dissertação de Mestrado, Universidade Federal do Ceará*, 2007.
- [4] OLIVEIRA, E. A. d. Linhas divisoras de águas e fraturas de caminhos ótimos em meios desordenados. *Tese de Doutorado, Universidade Federal do Ceará*, 2012.
- [5] BARABÁSI, A.-L. Invasion percolation and global optimization. *Physical Review Letters*, APS, v. 76, n. 20, p. 3750, 1996.
- [6] MASSON, Y.; PRIDE, S. R. A fast algorithm for invasion percolation. *Transport in porous media*, Springer, v. 102, n. 2, p. 301–312, 2014.
- [7] MASSON, Y. A fast two-step algorithm for invasion percolation with trapping. *Computers & Geosciences*, Elsevier, v. 90, p. 41–48, 2016.
- [8] MEAKIN, P. et al. Invasion percolation in a destabilizing gradient. *Physical Review A*, APS, v. 46, n. 6, p. 3357, 1992.
- [9] GLASS, R.; YARRINGTON, L. Simulation of gravity fingering in porous media using a modified invasion percolation model. *Geoderma*, Elsevier, v. 70, n. 2-4, p. 231–252, 1996.
- [10] SHEPPARD, A. P. et al. Invasion percolation: new algorithms and universality classes. *Journal of Physics A: Mathematical and General*, IOP Publishing, v. 32, n. 49, p. L521, 1999.
- [11] BAK, P.; TANG, C.; WIESENFELD, K. Self-organized criticality. *Physical review A*, APS, v. 38, n. 1, p. 364, 1988.
- [12] STAUFFER, D.; AHARONY, A. Introduction to percolation theory. Taylor & Francis, 1994.
- [13] ARAÚJO, A.; JR, J. A.; HERRMANN, H. Multiple invaded consolidating materials. *Physical Review E*, APS, v. 70, n. 6, p. 066150, 2004.
- [14] ROUX, S.; GUYON, E. Temporal development of invasion percolation. *Journal of Physics A: Mathematical and General*, IOP Publishing, v. 22, n. 17, p. 3693, 1989.
- [15] PACZUSKI, M.; MASLOV, S.; BAK, P. Avalanche dynamics in evolution, growth, and depinning models. *Physical Review E*, APS, v. 53, n. 1, p. 414, 1996.

- [16] SALEUR, H.; DUPLANTIER, B. Exact determination of the percolation hull exponent in two dimensions. *Physical review letters*, APS, v. 58, n. 22, p. 2325, 1987.
- [17] GROSSMAN, T.; AHARONY, A. Structure and perimeters of percolation clusters. *Journal of Physics A: Mathematical and General*, IOP Publishing, v. 19, n. 12, p. L745, 1986.
- [18] FEOFILOFF, P. *Análise de Algoritmos - A estrutura heap*. 2015. [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/heap.html](https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/heap.html). Acesso em: 10 set. 2018. Instituto de Matemática e Estatística da Universidade de São Paulo.