

Física Estatística Computacional - Projeto 4

Jorge Augusto Salgado Salhani

September 2018

1 Introdução

A modelagem matemática de fenômenos físicos mensuráveis é reducionista. O número de fatores externos e internos a determinado ente físico em um sistema real é incontável, e muitas destas variáveis devem ser a princípio negligenciadas para, após um conhecimento quase axiomático de seu comportamento, poder ser aos poucos adicionadas ao experimento a fim de descrevê-lo na forma mais geral possível. Estudos de sistemas com caráter complexo estão englobados em Teorias de Sistemas Dinâmicos [1] com sistemas de equações diferenciais (discretas ou contínuas) como modelador matemático.

Comportamentos de agentes biológicos e a evolução de características físicas que os descrevem possui elevada complexidade devido a presença de um número incontável de fatores que influenciam direta e/ou indiretamente tais agentes, o que os torna um bom exemplo de sistemas complexos. Dinâmicas de organismos possuem descrição tanto de seu movimento frente a condições ambientais [2, 3] como de sua desenvoltura quanto ser biológico, através de processos internos de variações genéticas [4], e um dos moldes computacionais utilizados se faz através de autômatos celulares.

Autômatos celulares [5] são sistemas matriciais discretos espacial e temporalmente com proposta de responder como determinadas condições iniciais evoluem nestas duas dimensões. Cada elemento de matriz (denominado 'célula') sofre constante atualização conforme é transcorrido um tempo Δt de acordo com sua vizinhança próxima, ou seja, uma célula é alterada conforme sua proximidade a dita.

Com uso de duas matrizes (principal e transicional), podem ser estudados fenômenos emergentes com algoritmos simples e de rápido processamento em ramos de áreas diversificadas, vagando de física [6] a biologia [2] e economia [7].

2 Objetivo

Este trabalho tem por objetivo estudar padrões emergentes de autômatos celulares simples bastante conhecidos. O "Experimento Crucial", o "Jogo da Vida" e o modelo de "Pilha de Areia" serão descritos através do método de implementação utilizado seguidos de análises qualitativas e quantitativas decorrentes de cada fenômeno observado para cada simulação.

3 Metodologia

Utilizando duas matrizes N -dimensional denominadas principal (sobre a qual condições iniciais são definidas e um passo temporal é contabilizado para cada atualização) e transicional (onde equações que ditam a evolução espacial são tomadas para que seus resultados alimentem a matriz principal e consolide a automatização do sistema de forma síncrona), para cada célula o estado de seus vizinhos próximos é guardado e utilizado na evolução do autômato. Estes estados são discretizados e variam de binário, ternário, quaternário, etc. dependendo do sistema em análise. O "Experimento Crucial" e "Jogo da Vida" utilizarão condições de contorno toroidais, ou seja, periódicas enquanto, para o modelo "Pilha de areia", será utilizada condições de fronteira livre.

3.1 Experimento Crucial

Com nome cativante dado por Stephen Wolfram, neste modelo o autômato apresenta estados binários e atualiza a linha i de acordo com os estados da linha anterior $i - 1$. O estado de uma célula na posição $c_{i+1,j}^t$ é alterado de acordo com o estado das células vizinhas $c_{i,j-1}^t$, $c_{i,j}^t$ e $c_{i,j+1}^t$, onde $c_{x,y}^t$ representa a célula, no instante t na posição (x, y) .

A alteração de estados de cada linha segue uma regra bem definida dada por condições associadas a cada uma das 8 combinações possíveis de estado das 3 células adjacentes (dado o estado binário de cada uma, há $2^3 = 8$ possibilidades). O conjunto de estados da vizinhança é dado por $\{(1,1,1), (1,1,0), (1,0,1), (0,1,1), (1,0,0), (0,1,0), (0,0,1), (0,0,0)\}$.

Diferentemente dos modelos seguintes, este é invariante temporalmente, uma vez que para estados definidos das células da linha $i - 1$, há apenas uma alteração possível da linha i de acordo com a regra definida. Assim, como otimização do processamento, podemos utilizar apenas uma matriz para este modelo.

3.2 Jogo da Vida

Primeiramente descrito por John Conway, o "Jogo da Vida" é um autômato celular com estados binários que evolui conforme a vizinhança de seus oito vizinhos próximos, ou seja, há 2^8 possíveis combinações responsáveis

por modificar o estado da célula $c_{i,j}^t$. Para que este número de possibilidades seja reduzido, argumentos de sobrevivência e morte celular são lançados, o que caracteriza o nome dado ao modelo. Desta forma, caso existam 2 células vizinhas em estado ligado (ou "vivas"), o estado da célula central é inalterado, ou seja, $c_{i,j}^t = c_{i,j}^{t-1}$. Se 3 células vizinhas apresentam-se "vivas", $c_{i,j}^t = 1$. Em todos os demais casos, $c_{i,j}^t = 0$, isto é, a célula central "morre".

Estes argumentos podem ser vislumbrados através do aspecto biológico considerando que há um crescimento celular apenas caso haja um número ótimo de células vivas (neste caso, o número é 3). Para valores menores que 3, o reduzido número de células não é capaz de estabelecer uma relação mútua de sobrevivência e o sistema é extinto, já para valores maiores que 3, há células suficiente para esgotar os recursos necessários para a sobrevivência da célula central.

3.3 Pilha de Areia

Para este modelo onde buscamos simular o crescimento de pilhas de areia sujeitas pontos de instabilidade que resultam em desmoronamento, os estados de cada célula pode descrito em termos binário (presença ou ausência de partícula) ou z -ário, onde z representa o valor crítico do número de grãos de areia que podem ser empilhados até atingir o estado instável. Nesta descrição o estado quaternário será utilizado.

Para cada passo temporal, uma célula é escolhida aleatoriamente e, sobre ela, uma partícula de areia será depositada, alterando o estado $c_{i,j}^t = 0$ (supondo ausência de grãos no instante t) para $c_{i,j}^{t+1} = 1$. Este incremento pode ser feito até o valor z constante. Quando $c_{i,j}^t = z$, a pilha desta posição atinge a instabilidade e desmorona, fazendo com que a cada um de seus vizinhos próximos seja acrescido um número n de grãos.

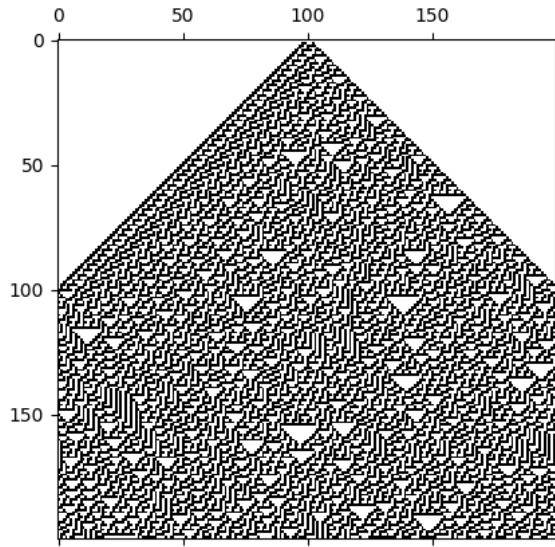
4 Resultados

Para cada um dos modelos supracitados, os resultados são mostrados a seguir, sequencialmente, com discussões sobre como condições iniciais alteram a evolução espaço-temporal de cada um e qual a sua representatividade para situações reais onde tais descrições matemáticas se aplicam.

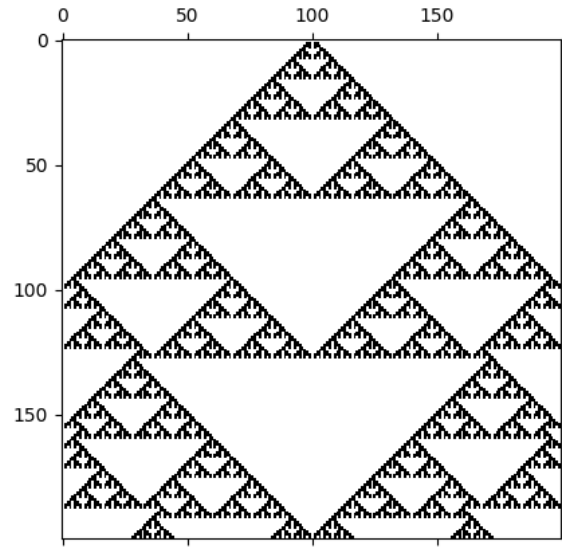
4.1 Experimento Crucial

Para uma matriz de dimensão (N, N) com $N = 200$, considerando a princípio condições iniciais fixas tais que $c_{0,49}^t = 1$, $c_{0,50}^t = 1$ e $c_{0,51}^t = 1$, com $c_{i,j}^t = 0$ caso contrário, a evolução espacial de algumas regras foram analisadas tomando condições de fronteira cilíndrica (aberta nos limites inferior e superior e cíclica nos limites laterais).

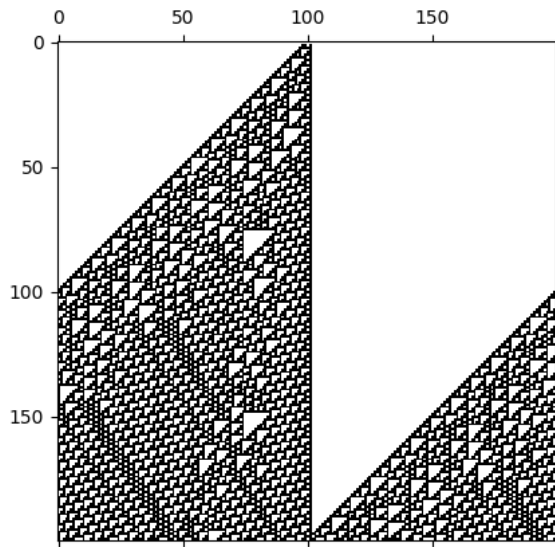
Na figura 1 e 2 as células iniciais foram sujeitas às regras 30, 90, 110, 126, 184, 218, 51 e 133 respectivamente.



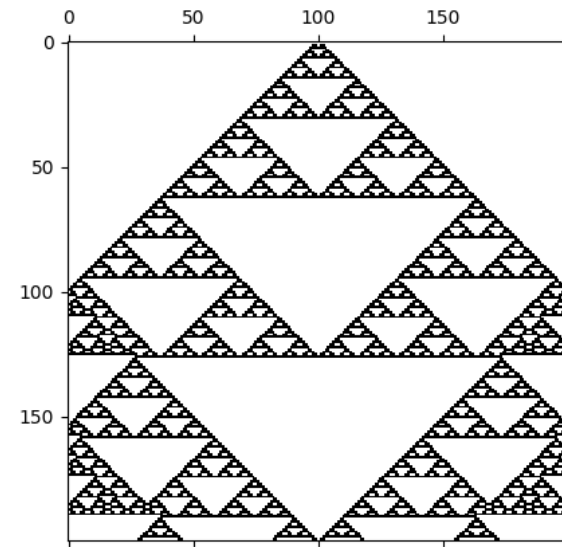
(a)



(b)



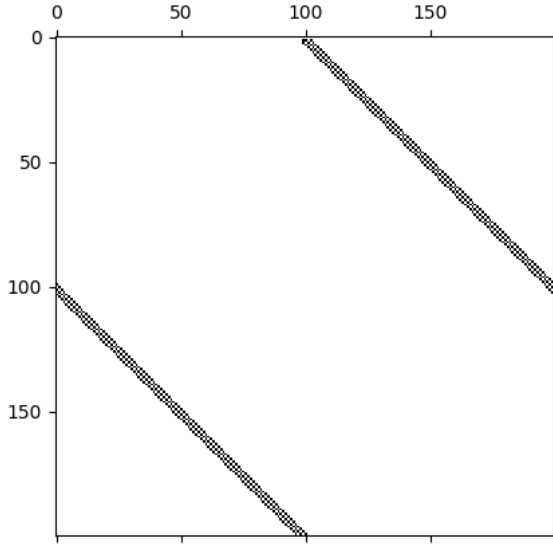
(c)



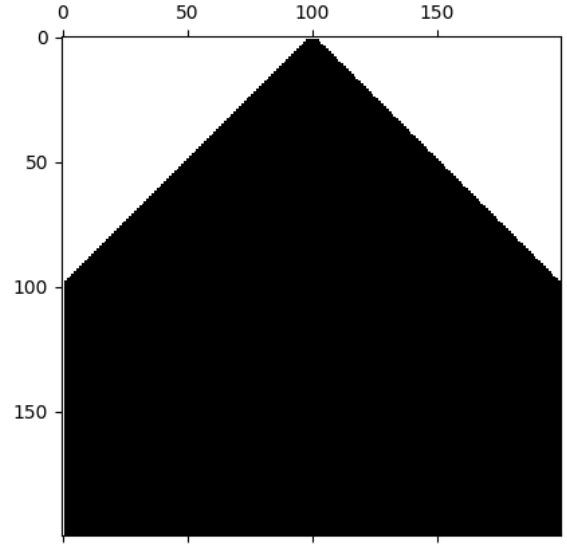
(d)

Figure 1: Padrão formado pelas regras 30 (a), 90 (b), 110 (c) e 126 (d) utilizando-se de condições iniciais não aleatórias e condições de fronteira cilíndricas.

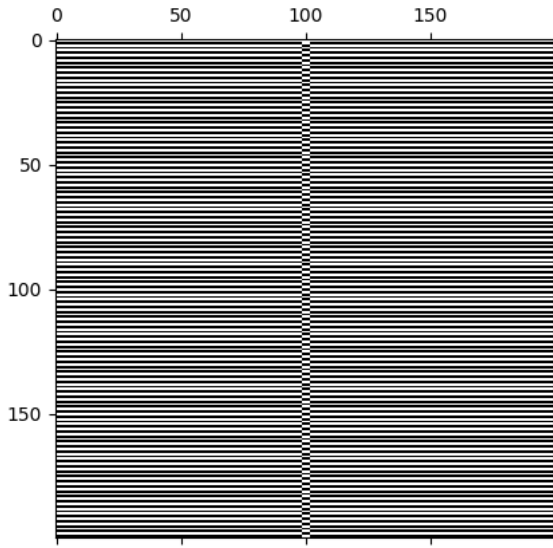
Algumas regras apresentam padrão espacial emergente bem definido, como as regras 51 correspondente à "reflexão" (figura 2c); 90 e 126 em forma de "cristais regulares" (figuras 1b e 1d); 30 e 110 em forma de "cristais irregulares" (figuras 1a e 1c); e 184, como "diagonal" (figura 2a). A beleza do padrão de "cristais irregulares",



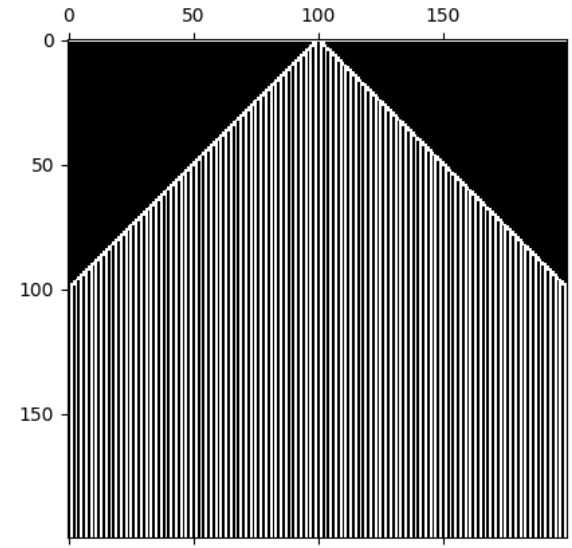
(a)



(b)



(c)



(d)

Figure 2: Padrão formado pelas regras 184 (a), 218 (b), 51 (c) e 133 (d) utilizando-se de condições iniciais não aleatórias e condições de fronteira cilíndricas.

por exemplo, pode ser vislumbrada através de formações de manchas regulares encontradas em conchas de caramujos marinhos da espécie *Conus textile*, como pode ser vista na figura 3. Tomando condições aleatórias, ou seja, alterando o estado da célula $c_{0,j}^t = 0$ em 1 através da função *random()* da biblioteca *numpy* com probabilidade igual a 30%, padrões espaciais distintos são formados e mostrados na figura 4, utilizando-se das

regras 30 e 126.

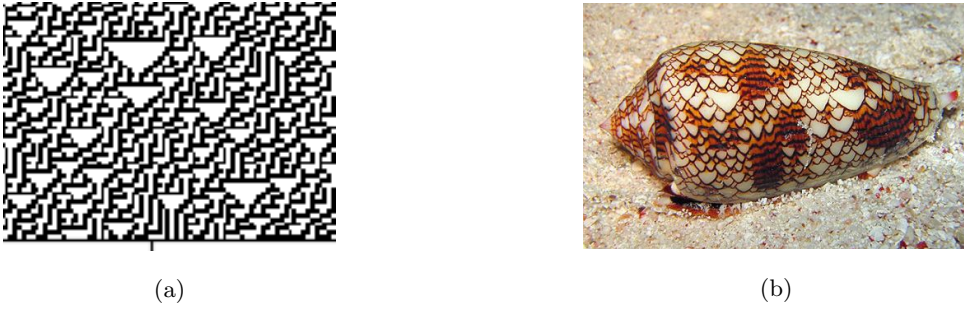


Figure 3: Comparação entre autômato celular de regra 30 e padrão de cores da concha do caramujo marinho *Conus textile*. Corte de uma porção do autômato celular (a) apresentado na figura 1a e fotografia [8] da concha do caramujo (b).

4.2 Jogo da Vida

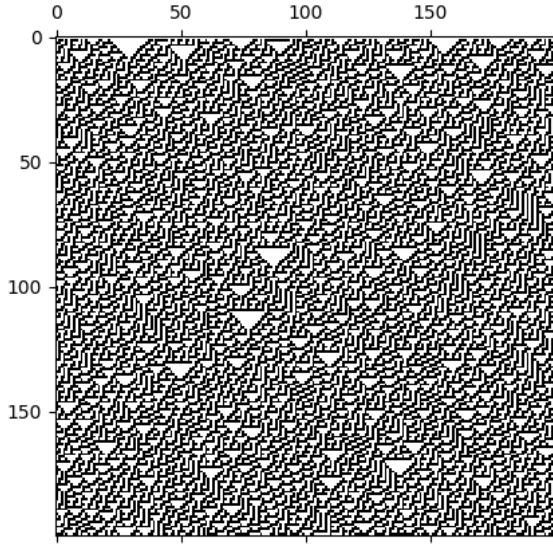
Para uma malha de dimensão (N, N) com $N = 70$, de acordo com as regras de evolução espacial e temporal atribuídas a este modelo (apresentadas na seção 3.2), existem apenas 2 configurações que determinam uma possível sobrevivência (i.e., manutenção de ao menos uma célula no estado de valor 1). Logo, para dada distribuição aleatória de células "vivas" e "mortas", apenas alguns aglomerados são mantidos após o primeiro passo temporal. Estes aglomerados seguem algumas formas bem conhecidas, nomeadas de acordo com seu comportamento no espaço discreto disponível. A figura 5 apresenta este caso, onde a condição inicial é dada ativando aleatoriamente as células presentes no sistema na proporção de 50%. A subsequente evolução espacial mostra que apenas um conjunto limitado de arranjos permanecem "vivos".

Alguns dos arranjos mais conhecidos são os chamados *gliders* (escaladores), que percorrem um caminho linear com velocidade fixa, *blinkers* (piscantes) que são mantidos fixos porém alternando entre três células vivas em uma linha e em uma coluna, e *blocks* (blocos), que permanecem inalterados.

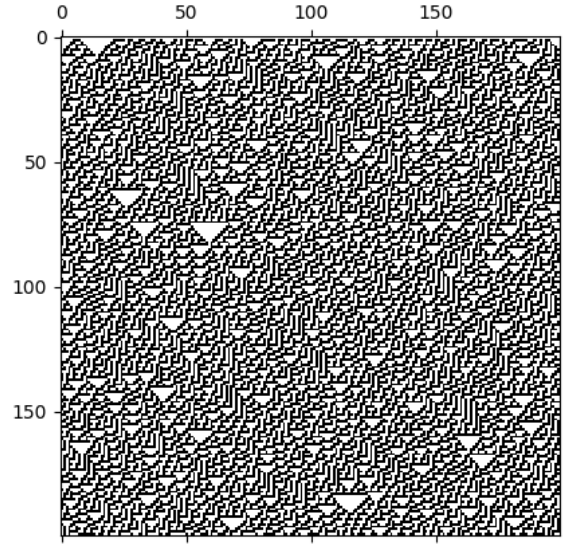
Um dos sistemas compostos mais conhecidos é o *Glider gun* (arma de escaladores), capaz de inesgotavelmente produzir *gliders*. Este sistema, assim como *blinkers* e *blocks* são apresentados na figura 6.

4.3 Pilha de Areia

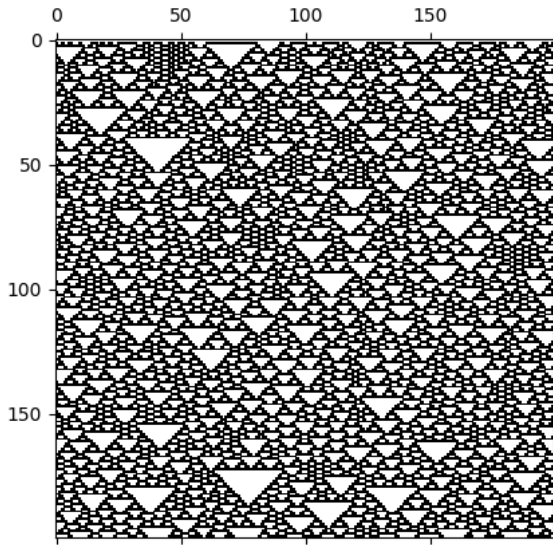
Inicialmente trataremos o caso determinístico, onde a cada espaço temporal um único grão é depositado sobre a posição central da matriz (N, N) para $N = 50$. Espera-se que a pilha espalhe seu conteúdo para posições mais distantes do centro conforme o tempo evolui. Em seguida, para o caso estocástico, a uma coordenada matricial escolhida aleatoriamente é adicionada uma partícula de areia. O número crítico de partículas é dado constante e igual a 4 (explicando o uso de estados quaternários explicitado na seção 3.3) em ambos os casos.



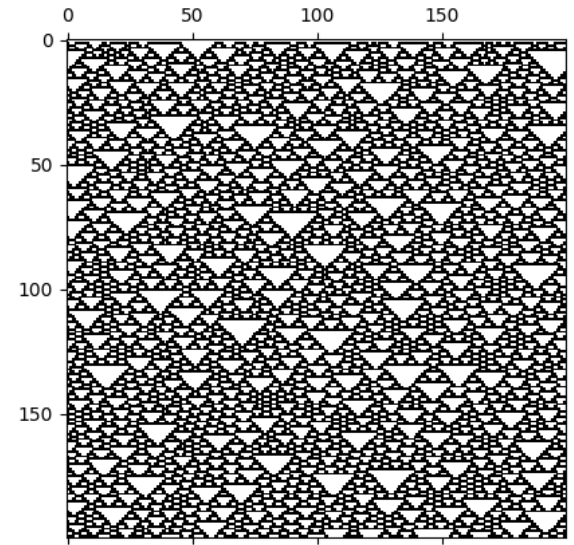
(a)



(b)



(c)



(d)

Figure 4: Padrão formado pelas regras 30 (a, b) e 126 (c, d) utilizando-se de condições iniciais aleatórias mantendo condições de fronteira cilíndricas.

Como observado na figura 7, o raio sobre o qual existem partículas cresce com simetria radial, como esperado para a evolução de sistemas determinísticos síncronos com atualização pontual única.

Para a simulação determinística, o número de avalanches ao longo do tempo pode ser avaliado através do gráfico apresentado na figura 8. Como podemos observar, após alguns passos temporais, a frequência de avalanches

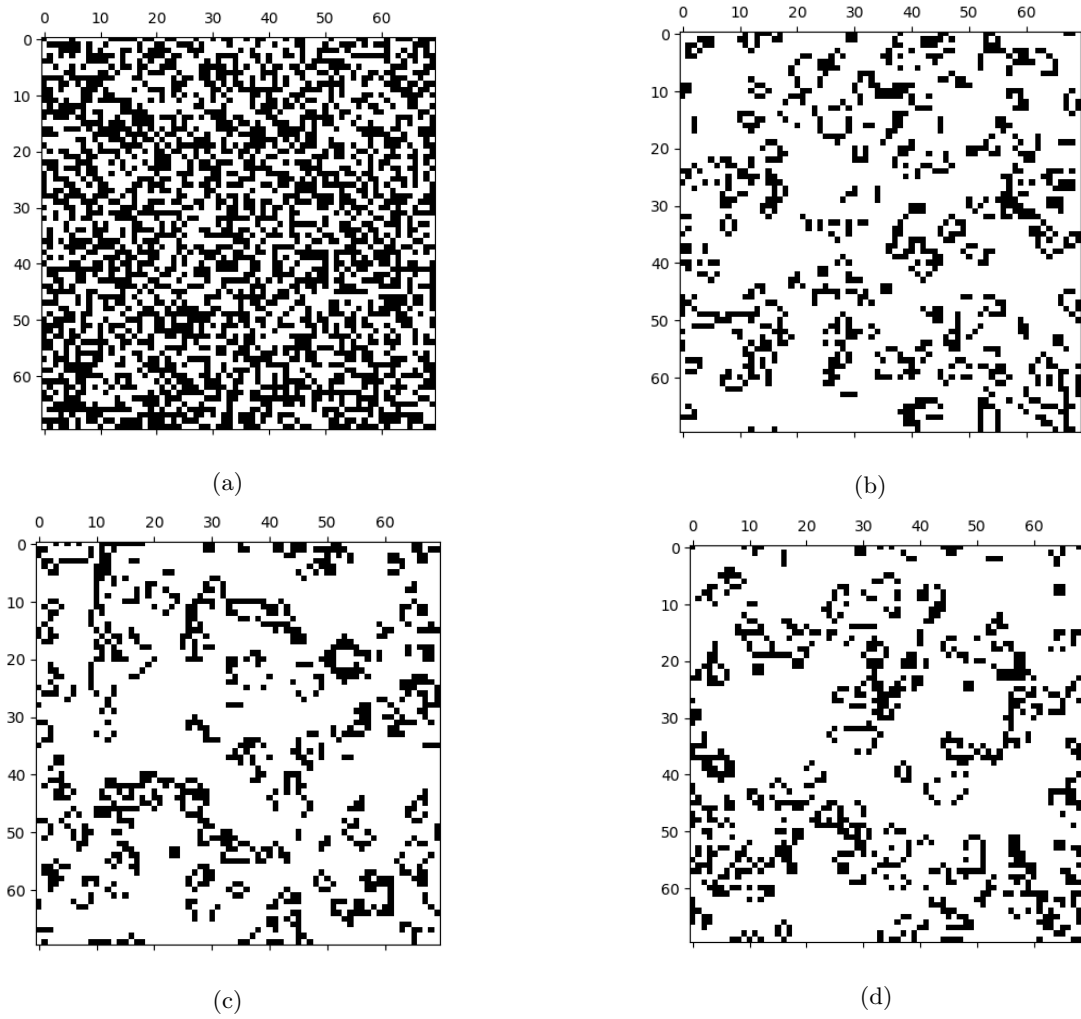


Figure 5: Evolução espacial-temporal do sistema para as regras dadas pelo Jogo da Vida após o intervalo temporal igual a 1 (a), 10 (b), 20 (c) e 30 (d) para condições iniciais de células "vivas" aleatoriamente posicionadas sobre a matriz, com aproximadamente 50% de seus sítios ocupados.

atinge uma dinâmica bem definida pela relação linear dada pela escala logarítmica das grandezas analisadas. Com reta de regressão linear dada pela equação $y = ax + b$, seus coeficientes são $a = 2.017(3)$ e $b = -4.11(2)$. Desta forma é possível relacionar o número f de avalanches com o tempo transcorrido através da equação $f = Ct^a$, com C constante, indicando a dependência com uma potência temporal próxima a 2, com erro associado de 0.1%.

Por fim, para condições aleatórias, onde uma posição da matriz é escolhida e sobre ela é depositada uma partícula de areia, a frequência de avalanches possui aspecto distinto daquele apresentado na figura 8, no entanto para vários experimentos, como mostrado na figura 9, todos apresentam resultados similares. Para tempos maiores que $t = 2000$, notamos a predominância de comportamento linear independente do fator determinístico associado aos experimentos.

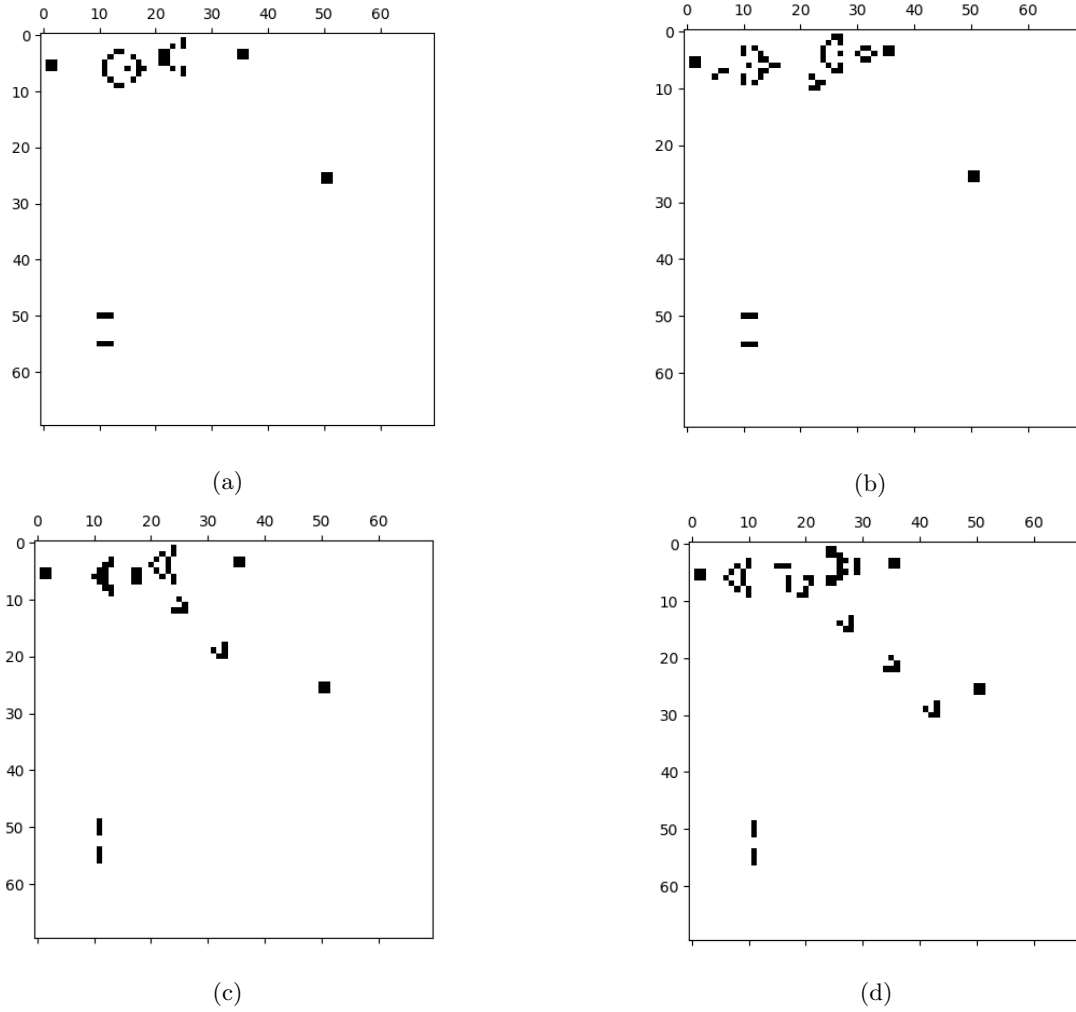
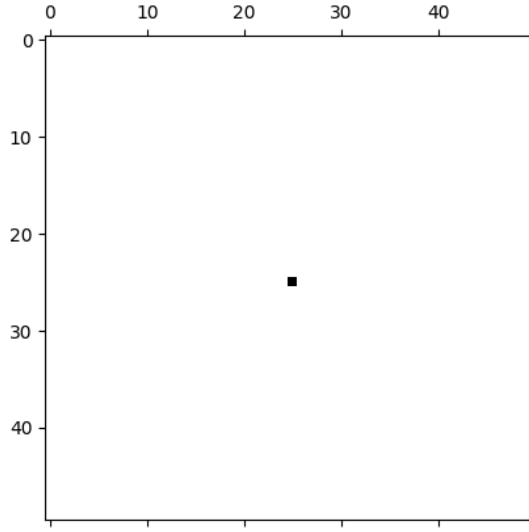


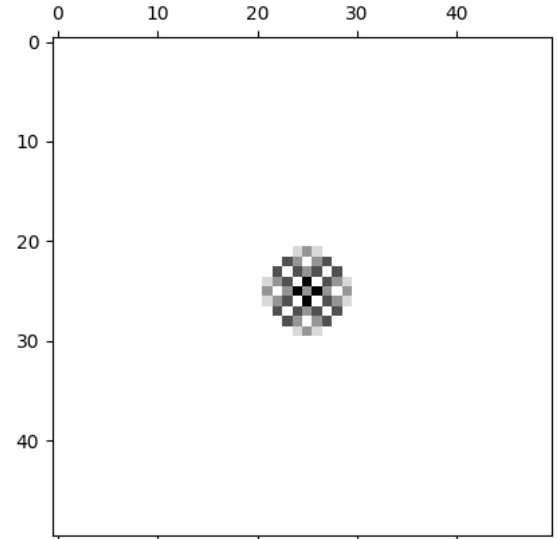
Figure 6: Alguns dos padrões duradouros do Jogo da Vida para os tempos $t = 1$ (a), $t = 20$ (b), $t = 39$ (c) e $t = 99$ (d). Observando a figura 6a, podemos notar as condições iniciais da *Glider gun* no canto superior esquerdo e os elementos *blocks*, próximo à coordenada $(25, 50)$, e *blinkers*, próximo a $(50, 10)$ e $(55, 10)$.

Através da regressão linear apresentada na figura 9a, a reta dada pela equação $y = ax + b$ apresenta coeficientes $a = 27.228(7)$ e $b = -52997(27)$, o que indica a relação entre a frequência f de avalanche dada por $f = at + K$, com K constante.

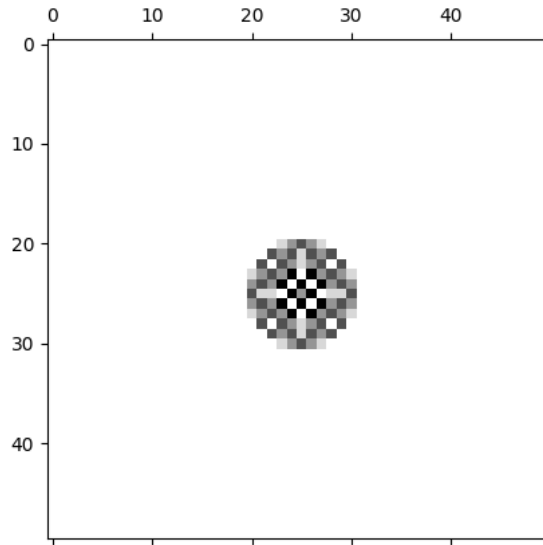
Uma das variações do resultado apresentado para a frequência de avalanches impondo condições aleatórias deve-se ao tamanho do espaço físico disponível, i.e. a dimensão N da matriz. Caso a dimensão seja alterada para $N = 40$, a relação linear se mantém porém o momento onde o sistema passa a se apresentar massivas avalanches é deslocado de aproximadamente $t = 2000$ para $t = 3500$ (figura 9b). Os parâmetros de regressão linear possuem valores iguais a $a = 49.98(3)$ e $b = -169368(151)$, havendo um aumento de ≈ 1.83 na frequência de avalanches. A alteração da frequência de avalanches deve-se a presença de um número maior de pilhas presentes no sistema.



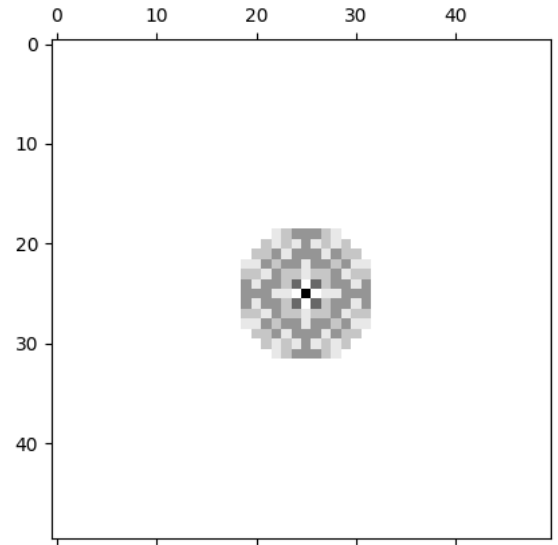
(a)



(b)



(c)



(d)

Figure 7: Pilha de areia determinística com condição inicial dada por 4 grãos depositos sobre a posição central (25,25). Para cada intervalo de tempo $t = 1$ (a), $t = 100$ (b), $t = 200$ (c) e $t = 300$ (d), um grão é adicionado à posição central. A escala de cinza varia de 0 (branco) a 4 (preto), representando o número de partículas em determinada coordenada espacial.

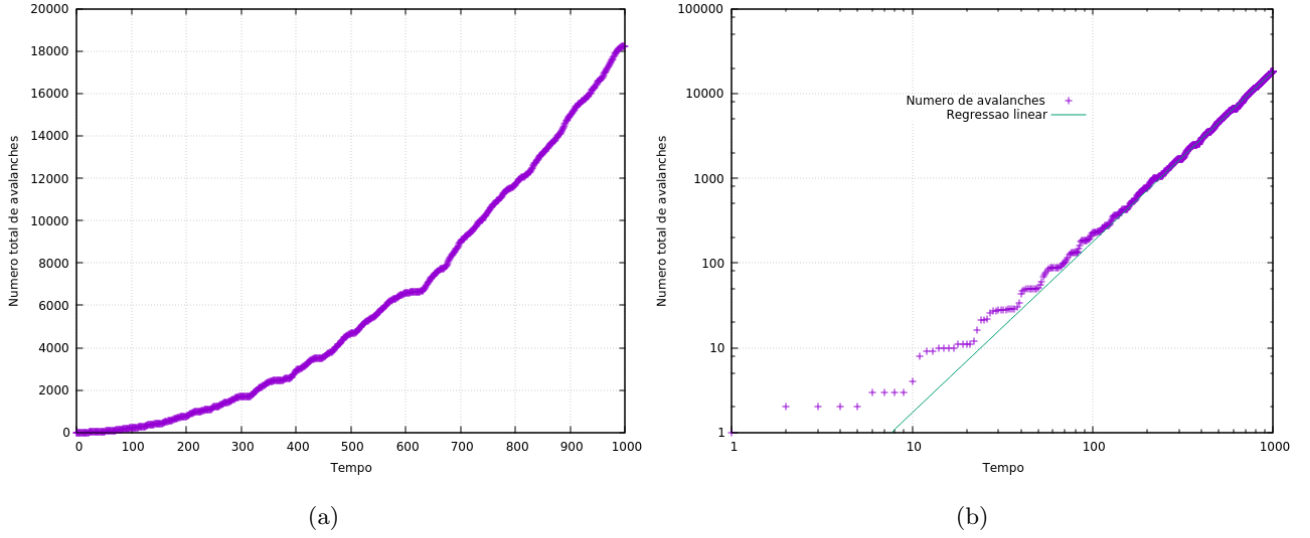


Figure 8: Evolução temporal do número total de avalanches para o sistema determinístico de deposição de grãos de areia apenas na posição central $(25, 25)$ do sistema N -dimensional, com $N = 50$, conforme indica a figura 7. Em (a) a relação possui forma aproximadamente parabólica e, tomando o logaritmo de cada eixo (b), a relação possui aspecto linear para tempos prolongados. A reta contínua verde apresenta a regressão linear para valores de $x \in [100, 1000]$.

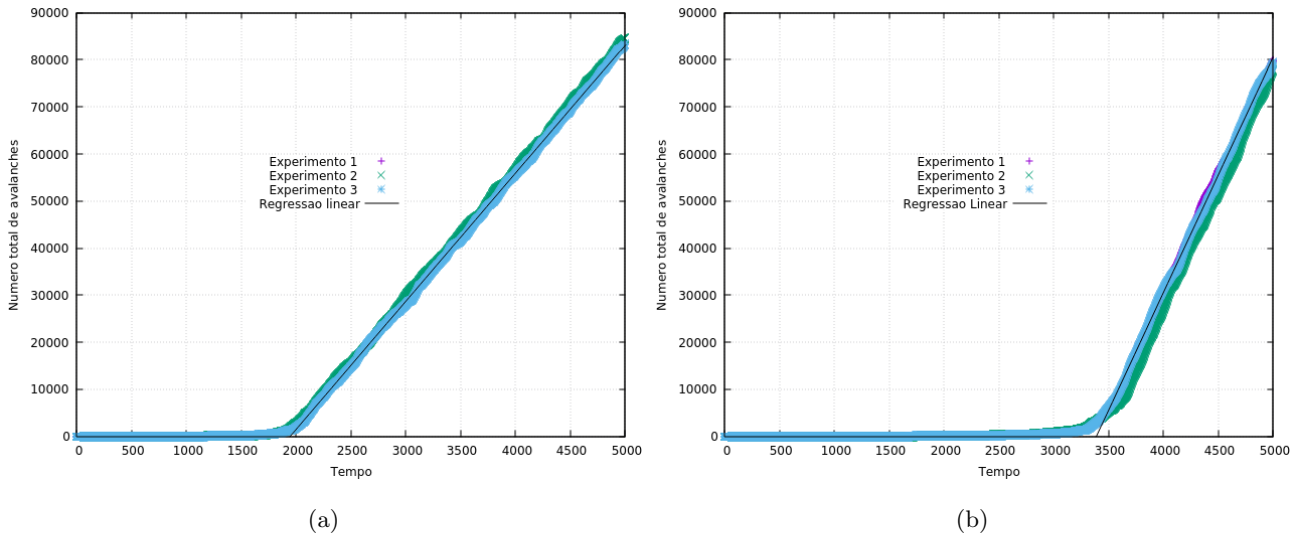


Figure 9: Evolução temporal do número total de avalanches para o sistema aleatório de deposição de grãos de areia em sistema N -dimensional, com $N = 30$ (a) e $N = 40$ (b). Distintamente da frequência de avalanches no caso determinístico apresentada na figura 8, a relação linear é vista sem necessidade de alteração logarítmica das escalas consideradas. A reta contínua (cor preta) representa a regressão linear para valores de $x \in [2000, 5000]$ (a) e $[3500, 5000]$.

5 Conclusão

O modelo de autômatos celulares apresentado neste trabalho torna possível a simulação de sistemas complexos com relativa facilidade, uma vez que o algoritmo utilizado apresenta rápido processamento, apesar de ser necessário, devido a evolução síncrona de duas matrizes, o armazenamento duplo de todo dado utilizado.

Os resultados obtidos possuem parâmetros físicos quantitativos acessíveis e conferem à simulação boa precisão, mostrando a adequação do modelo computacional proposto.

6 Bibliografia

- [1] Smith, L. B., Thelen, E. A Dynamic Systems Approach to the Development of Cognition and Action, 1996
- [2] Rodrigues, L. A. D., Varriale, M. C., Godoy, W. A. C. Spatiotemporal dynamics of an insect population in response to chemical substances. *Ecological Complexity* vol. 16, pp. 51–58, 2012.
- [3] Comins, H. N., Hassel, M. P., May, R. M. The spatial dynamics of host-parasitoid systems. *Ecological Complexity* vol. 61, pp. 735–748, 1992
- [4] Sirakoulis G. Ch. et al. A cellular automaton model for the study of DNA sequence evolution. *Computers in Biology and Medicine* vol. 33, Issue 5, pp. 439-453, 2003.
- [5] Wolfram, S. A New Kind of Science. Ch. 1 The Foundations for a New Kind of Science, pp. 19-22, 2002.
- [6] M. W. Janowicz, et al. Cellular Automaton Approach to Electromagnetic Wave Propagation in Dispersive Media. *Proceedings: Mathematical, Physical and Engineering Sciences*, vol. 462, 2074, pp. 2927-2948, 2016.
- [7] Lopez-Ruiz, R. An Economic Model of Coupled Exponential Maps. *IEEE Workshop on Nonlinear Maps and Applications*. pp. 13-14, 2007
- [8] Textile Cone Snail. Imagem disponível em < http://www.aquariumofpacific.org/onlinelearningcenter/species/-textile_cone_snail >. Acesso em 21 de outubro de 2018, às 15:00.

7 Códigos

Em sequência estão apresentados os códigos de cada um dos modelos apresentados neste trabalho. Para cada um há uma breve descrição de seu funcionamento.

7.1 Experimento Crucial

Para cada evolução espacial, a linha $i + 1$ é alterada conforme uma dada regra. Tal regra é passada ao programa em forma decimal para que o mesmo o converta em binário e possa atribuir, a cada 3 vizinhos $c_{i,j-1}^t$, $c_{i,j}^t$ e $c_{i,j+1}^t$, um estado à posição $c_{i,j}^t$.

```
N=200 #Dimensao da matriz
M1=np.zeros((N,N)) #Criacao de uma matriz (N,N) na qual todos seus elementos sao nulos

rule=[0,0,0,0]

bina=126 #Leitura da regra em decimal
i=2
while i<len(bin(bina)): #Alteracao da regra para a base binaria
    rule.append(int(bin(bina)[i]))
    i+=1
print(rule)

k=0
while k<N:
    if np.random.random()<=0.3: #Condicao inicial aleatoria, tornando $\approx 30\%$ em estado ativado
        M1[0,k]=1
        k+=1

m=1
while m<N:
    n=0
    while n<N:
        neighs=[M1[(m-1)%N,(n-1)%N], M1[(m-1)%N,n%N], M1[(m-1)%N,(n+1)%N]] #Armazenamento dos vizinhos
        proximos
        if neighs==[1,1,1]:
            M2[m%N,n%N]=rule[-8]

        elif neighs==[1,1,0]:
            M2[m%N, n%N] = rule[-7]

        elif neighs==[1,0,1]:
            M2[m%N, n%N] = rule[-6]

        elif neighs==[1,0,0]:
            M2[m%N, n%N] = rule[-5]
```

```

elif neighs==[0,1,1]:
    M2[m%N, n%N] = rule[-4]

elif neighs==[0,1,0]:
    M2[m%N, n%N] = rule[-3]

elif neighs==[0,0,1]:
    M2[m%N, n%N] = rule[-2]

else:
    M2[m%N, n%N] = rule[-1]

n+=1

M1=M2
m+=1

plt.matshow(M1, cmap=plt.cm.Greys)
plt.show()

```

7.2 Jogo da Vida

Para cada elemento da matriz representativa do espaço físico, 8 vizinhos próximos são armazenados e as regras do Jogo da Vida (mencionados na seção 4.2) são aplicadas.

```

N = 70 #Dimensao da matriz
M1=np.zeros((N,N)) #Definicao de todos os elementos da matriz como nulos

#Posicoes ativas para a criacao da $\textit{Glider Gun}$
M1[1,25]=1
M1[2,25]=1
M1[2,23]=1
M1[3,22]=1
M1[3,21]=1
M1[3,14]=1
M1[3,13]=1
M1[4,12]=1
M1[4,16]=1
M1[4,21]=1

```

$M1[4,22]=1$
 $M1[5,1]=1$
 $M1[5,2]=1$
 $M1[5,11]=1$
 $M1[5,17]=1$
 $M1[5,21]=1$
 $M1[5,22]=1$
 $M1[6,1]=1$
 $M1[6,2]=1$
 $M1[6,11]=1$
 $M1[6,15]=1$
 $M1[6,17]=1$
 $M1[6,18]=1$
 $M1[6,23]=1$
 $M1[6,25]=1$
 $M1[7,11]=1$
 $M1[7,17]=1$
 $M1[7,25]=1$
 $M1[8,12]=1$
 $M1[8,16]=1$
 $M1[9,13]=1$
 $M1[9,14]=1$
 $M1[3,35]=1$
 $M1[3,36]=1$
 $M1[4,35]=1$
 $M1[4,36]=1$

#Posicoes ativas para a criacao de $\text{\textit{Blinkers}}$

$M1[50,10]=1$
 $M1[50,11]=1$
 $M1[50,12]=1$

$M1[55,10]=1$
 $M1[55,11]=1$
 $M1[55,12]=1$

#Posicoes ativas para a criacao de $\text{\textit{blocks}}$

$M1[25,50]=1$
 $M1[25,51]=1$
 $M1[26,50]=1$
 $M1[26,51]=1$

```

def update(data):
    global M1
    M2 = M1.copy() #Copia a matriz M1 para a nova variavel M2
    i=0
    while i<N:
        j=0
        while j<N:
            neighs = [M1[i, (j-1)%N], M1[i, (j+1)%N], M1[(i-1)%N, j], M1[(i+1)%N, j], M1[(i-1)%N,
                (j-1)%N],
                M1[(i-1)%N, (j+1)%N], M1[(i+1)%N, (j-1)%N], M1[(i+1)%N, (j+1)%N]] #Definicao dos
                vizinhos proximos

            #Condicoes do Jogo da Vida
            if sum(neighs) == 2:
                M2[i,j] = M1[i,j]

            elif sum(neighs) == 3:
                M2[i,j] = 1

            else:
                M2[i,j] = 0

            j+=1
        i+=1

    mat.set_data(M2) #Aplicacao dos dados para uma nova variavel responsavel por crair uma animacao
    M1 = M2
    return [mat]

fig, ax = plt.subplots()
mat = ax.matshow(M1, cmap=plt.cm.Greys)
ani = animation.FuncAnimation(fig, update, frames=200, repeat=False, interval=500)
plt.show()

```

7.3 Pilha de Areia

Para cada elemento de matriz, 4 vizinhos próximos são armazenados e, caso haja um número crítico de partículas igual a 4 em uma dada posição, os vizinhos próximos recebem uma partícula cada.

```
N = 40 #Definicao da dimensionalidade da matriz
M1=np.zeros((N,N)) #Tornar todos os elementos como nulos
M1[int(np.random.random()*N),int(np.random.random()*N)] += 4 #Inicializador com 4 particulas de areia
    para uma dada posicao escolhida aleatoriamente
arq=open('avalFreq40Rand3.dat', 'w') #Arquivo de entrada de dados
tot = 0 #Contador do numero de avalanches

def update(data):
    global M1, tot

    M1[int(np.random.random()*N),int(np.random.random()*N)] += 1 #Para cada passo temporal, uma
        partícula e depositada sobre uma posicao aleatoriamente escolhida
    M2 = M1.copy() #Copia da matriz principal M1 atribuida a variavel M2
    x = 1
    while x < N - 1:
        y = 1
        while y < N - 1:
            if M1[x, y] >= 4: #Condições de avalanche, em uma posicao haja 4 ou mais particulas
                M2[x + 1, y] += 1
                M2[x - 1, y] += 1
                M2[x, y - 1] += 1
                M2[x, y + 1] += 1
                M2[x, y] -= 4
                tot+=1

            else:
                pass

            y += 1
        x += 1

    mat.set_data(M2)
    M1 = M2
    arq.writelines([str(data), '\t', str(tot), '\n'])

fig, ax = plt.subplots()
```

```
mat = ax.matshow(M1, cmap=plt.cm.Greys)
ani = animation.FuncAnimation(fig, update, frames=5000, repeat=False, interval=5)
plt.show()
```
