

# Física Estatística Computacional

## Quarto Projeto: Autômatos Celulares

### Instruções

- Crie um diretório “PROJ4\_nome” em `/home/public/FIESTCOMP/PROJ4`
- Não proteja seu diretório para não ser lido por “g” e “o” :-)
- Deixe no diretório um arquivo de nome “rel\_proj4.pdf”, com um relatório (elaborado utilizando `latex`) sobre o projeto, incluindo texto, listagens dos códigos, gráficos e tabelas.

### Introdução

Autômatos celulares são sistemas definidos em uma grade (i.e. espaço discreto) regular, cujas *células* podem estar em um número finito de estados. Dada a configuração da *vizinhança* de uma célula — geralmente tomada como a própria célula mais as células em sítios primeiros vizinhos (sítios que “tocam” a célula) e segundos vizinhos (a distância de  $\sqrt{2}$  passos da célula, ou seja vizinhos diagonais) — em um tempo (discreto)  $t$ , é atualizada a configuração do sistema de acordo com uma dada *regra de atualização*. Note que a atualização é feita como um todo, ou seja cada célula é atualizada no tempo  $t$  em função dos valores de suas vizinhas no tempo  $t - 1$ , sendo geralmente necessário manter duas cópias completas do sistema na memória do computador. (O mesmo não vale em geral para a evolução temporal por métodos de Monte Carlo e dinâmica molecular.)

As regras de atualização são tipicamente determinísticas (daí o nome *autômato*), definindo uma equação diferencial parcial de primeira ordem no tempo. Tais sistemas tem uma dinâmica rica e uma variedade de aplicações, como a descrição de padrões na natureza e a realização de cálculos computacionais para sistemas discretos. Faça uma boa revisão do assunto e inclua-a em seu relatório, juntamente com a solução dos exercícios abaixo.

## Exercícios

1) **Autômatos celulares unidimensionais:** considere uma cadeia de  $L$  sítios, com células que podem estar nos estados 0 ou 1. Tome condições periódicas de contorno e regra de evolução dada para o sítio  $i$  pelos valores nos sítios  $i - 1$ ,  $i$  e  $i + 1$  no tempo anterior. Como existem 8 configurações possíveis para a vizinhança, há 256 regras possíveis. Tais regras estão em correspondência unívoca com números binários de 8 dígitos, e podem ser expressas por seu “código de Wolfram”, dado pelo número binário na base 10. Por exemplo, a regra da maioria

<b>111</b>	<b>110</b>	<b>101</b>	<b>100</b>	<b>011</b>	<b>010</b>	<b>001</b>	<b>000</b>
↓	↓	↓	↓	↓	↓	↓	↓
<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>

é indicada como regra 232, pois

$$0 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 + = 232.$$

Faça um código que calcule a evolução do autômato acima para o caso geral. Leia a regra (em base 10) e imprima a saída em forma de uma matriz, sendo a linha superior a condição inicial e seguindo a evolução temporal de cima para baixo. Utilize condição inicial de apenas uma célula ocupada, no centro e também condição inicial aleatória.

Investigue (comentando seu nome e significado) as regras: 255 (da epidemia), 51 (reflexão), regras para cristais regulares (e.g. a regra “módulo-2” 90 e as regras 126, 218) e irregulares (e.g. a regra “da concha” 30 e a regra 110), a regra da diagonal 184. Inclua outras, se desejar. (Consulte o artigo do Wolfram no site da disciplina.) Teste a diferença entre o caso de condição inicial fixa e aleatória.

2) **O jogo da vida:** este autômato bidimensional foi introduzido em 1970 por Conway. Considere uma rede de células dispostas em um quadrado, ocupando estados 0 (“morta”) ou 1 (“viva”). Tome a vizinhança composta por 8 sítios, como descrito acima. A evolução será dada da seguinte forma. A cada tempo  $t$ , examine os estados da vizinhança de cada célula no tempo  $t - 1$ , e conte o número de células vivas. Se (exatamente) duas células vizinhas estão ainda vivas, a célula central permanecerá no estado em que se encontra (viva ou morta). Se (exatamente) três células vizinhas estão ainda vivas, a célula central estará viva no tempo  $t$ , independentemente de seu estado em  $t - 1$ . Em todos os outros casos, a célula central morre. Simule este autômato para várias condições iniciais, incluindo os famosos padrões do tipo: **blinkers** (grupo de três células vivas em linha horizontal, que se transforma

em vertical no próximo espaço de tempo, e vice-versa) e **gliders** (grupo de algumas células que se “propaga” pela rede sem mudar de forma, pesquise as várias formas). Utilize condições de contorno periódicas. Escreva a saída de seu programa (evolução temporal para tempos sucessivos  $t$ ) com as configurações (i.e. duplas de coordenadas para as células no estado “1” no tempo  $t$ ) separadas pelo símbolo # (em linha separada), de forma que o arquivo possa ser visualizado como um filme, utilizando o script abaixo.

```
#!/bin/csh -f
# usage: movie.csh dados L
set file = $argv[1]
set L = $argv[2]
echo "set xrange [0:$L]" > movie.gnu
echo "set yrange [0:$L]" >> movie.gnu
cat << EOF >! movie.awk
    BEGIN {i=0}{if(NR==1||/#/)i++;if(!/#/)print >>"tmp-"i}
EOF
cat $file|awk -f movie.awk
set i = 1
while (-f tmp-${i})
    echo plot \"tmp-${i}\" w l >> movie.gnu
    echo pause 3 >> movie.gnu
    set i = `expr $i + 1`
end
gnuplot < movie.gnu
rm -f tmp_*
```

2) **Modelo da pilha de areia:** este autômato descreve de forma simplificada o comportamento de avalanches, apresentando a característica de *criticalidade auto-organizada*, bastante importante para sistemas complexos em mecânica estatística. Considere uma rede quadrada bidimensional, em que grãos de areia caem com distribuição uniforme, formando pilhas. Para cada sítio  $i$  da rede, a altura da pilha de grãos é dada pela variável  $z_i$ . Quando a pilha ultrapassar um valor  $z_c$  ela se tornará instável, resultando em um processo de avalanche. Vamos supor  $z_c = 3$ , portanto os valores estáveis para  $z_i$  serão 0, 1, 2, 3. Quando o valor de  $z_i$  ultrapassar  $z_c$  a pilha vai “desmoronar”, o que é representado por  $z_i \rightarrow z_i - 4$  e os 4 vizinhos mais próximos de  $i$  terão acréscimo de 1 grão em suas pilhas. Uma iteração do algoritmo consiste em adicionar um grão aleatoriamente à rede e testar todos os sítios até que nenhum esteja mais instável. (Isso equivale a seguir a avalanche até seu término.) Simule o

autômato para uma rede de lado 100. Considere que quando o grão cair para fora da rede ele será perdido (condições de contorno abertas). Represente graficamente a dinâmica do sistema utilizando cores diferentes para os diferentes estados das pilhas. Calcule o tamanho e a duração das avalanches. Faça gráficos da frequência e duração de avalanches em função de seu tamanho.