BIO2045 – Séance 2
Les tâches et les rayures

Contenu

Concepts principaux

Indexation basique dans les matrices

Dans la séance précédente, nous avons vu comment créer des matrices, et comment lire le contenu à une ligne et colonne particulière. On peut, en pratique, faire beaucoup plus avec des matrices.

Prenons l'exemple de la matrice suivante, avec trois lignes et cinq colonnes, qui contient des nombres aléatoires entiers entre 1 et 5:

```
V = rand(1:5, 3, 5)
```

```
3×5 Matrix{Int64}:
 3  4  5  4  3
 4  1  3  5  5
 1  1  3  1  2
```

On peut accéder à la première ligne de cette matrice avec

```
V[1,:]
```

```
5-element Vector{Int64}:
 3
 4
 5
 4
 3
```

et à sa deuxième colonne avec

```
V[:,2]
```

```
3-element Vector{Int64}:
 4
 1
 1
```

On peut aussi prendre les deux premières lignes, et les trois dernières colonnes, avec

```
V[begin:(begin+1), (end-2):end]
```

```
2×3 Matrix{Int64}:
 5  4  3
 3  5  5
```

Ce qui est la même chose que

```
V[1:2, 3:5]
```

```
2×3 Matrix{Int64}:
 5  4  3
 3  5  5
```

mais sans avoir besoin d'avoir les coordonées exactes de la dernière colonne.

Indexation avancée dans les matrices

Les matrices ont toutes un système de coordonées, qui sont soit les coordonées Cartésiennes:

```
collect(CartesianIndices(V))
```

```
3×5 Matrix{CartesianIndex{2}}:
 CartesianIndex(1, 1)  CartesianIndex(1, 2)  CartesianIndex(1,
3)  CartesianIndex(1, 4)  CartesianIndex(1, 5)
 CartesianIndex(2, 1)  CartesianIndex(2, 2)  CartesianIndex(2,
3)  CartesianIndex(2, 4)  CartesianIndex(2, 5)
 CartesianIndex(3, 1)  CartesianIndex(3, 2)  CartesianIndex(3,
3)  CartesianIndex(3, 4)  CartesianIndex(3, 5)
```

soit les coordonées linéaires:

```
collect(LinearIndices(V))
```

```
3×5 Matrix{Int64}:
 1  4  7  10  13
 2  5  8  11  14
 3  6  9  12  15
```

Notez que les coordonées linéaires suivent les colonnes: Julia est un langage *column-major*, qui va stocker les colonnes ensemble dans la mémoire. Si on veut améliorer la performance de nos simulations, opérer sur les colonnes sera souvent beaucoup plus rapide que d'opérer sur les lignes.

Une caractéristique importante des indices est qu'ils sont *relatifs*. Par exemple, si on veut exprimer la position qui est "la cellule à gauche de la position 3, 4", on peut l'écrire

```
CartesianIndex(3, 4) + CartesianIndex(-1, 0)
```

```
CartesianIndex(2, 4)
```

La position `CartesianIndex(-1, 0)` signifie: une colonne avant, sur la même ligne. Nous allons *beaucoup* utiliser cette propriété pour nous déplacer rapidement dans des matrices.

Iteration

Dans la séance précédente, nous avions utilisé une boucle `for`, qui permettait de répéter un processus plusieurs fois. Dans cette séance, nous allons formaliser ce concept, qui est fondamental pour le reste du cours.

Une boucle `for` est une structure qui s'écrit en général de la manière suivante:

```
for ELEMENT in COLLECTION
  instructions
end
```

La variable `ELEMENT` n'existe pas en dehors de la boucle. C'est une nuance importante: elle est créée par la boucle, et détruite quand la boucle est terminée. Une boucle `for` va simplement prendre chaque valeur de `COLLECTION`, les stocker dans `ELEMENT`, et on pourra donc appliquer des opérations de manière itérative.

Par exemple, si on veut multiplier par deux tous les éléments du vecteur `[1, 2, 3, 4]`, et afficher le résultat sur une nouvelle ligne avec `println`, on peut utiliser une boucle `for`:

```
for x in [1, 2, 3, 4]
    println(2x)
end
```

```
2
4
6
8
```

Iteration avancée dans les matrices

On peut traverser des matrices de façon beaucoup plus efficace en combinant les boucles `for` et les techniques d'indexation. Pour rappel, dans cette section, nous utilisons la matrice suivante:

```
V = rand(1:9, 3, 4)
```

```
3×4 Matrix{Int64}:
 6  3  7  2
 8  3  1  8
 7  8  3  3
```

Par exemple, on peut prendre chaque élément d'une matrice sans devoir spécifier les lignes et les colonnes:

```
for v in V
    println(v)
end
```

```
6
8
7
3
3
8
7
1
3
2
8
3
```

Remarquez que l'ordre des éléments suit ici le `LinearIndex`. On peut aussi aller chercher directement les indices des matrices:

```
for i in eachindex(V)
    println(i)
end
```

```
1
2
3
4
5
6
7
8
9
10
11
12
```

Mais les indices sont eux-même retournés sous forme de matrice. On peut donc itérer sur les indices Cartésiens:

```
for ci in CartesianIndices(V)
    println(ci)
end
```

```
CartesianIndex(1, 1)
CartesianIndex(2, 1)
CartesianIndex(3, 1)
CartesianIndex(1, 2)
```

```
CartesianIndex(2, 2)
CartesianIndex(3, 2)
CartesianIndex(1, 3)
CartesianIndex(2, 3)
CartesianIndex(3, 3)
CartesianIndex(1, 4)
CartesianIndex(2, 4)
CartesianIndex(3, 4)
```

Cette structure est particulièrement utile, parce que nous aurons souvent besoin de faire des tâches comme: pour chaque cellule, prendre la cellue du dessus, et si cette cellule est dans la matrice, effectuer une opération sur sa valeur.

```
for position in CartesianIndices(V)
    dessous = position + CartesianIndex(0, -1)
    if dessous in CartesianIndices(V)
        println(dessous)
    end
end
```

```
CartesianIndex(1, 1)
CartesianIndex(2, 1)
CartesianIndex(3, 1)
CartesianIndex(1, 2)
CartesianIndex(2, 2)
CartesianIndex(3, 2)
CartesianIndex(1, 3)
CartesianIndex(2, 3)
CartesianIndex(3, 3)
```

On utilise ici la structure `if un truc in plusieurs trucs`, qui renvoie `true` si l'élément `un truc` fait partie de la collection `plusieurs trucs`.

On peut enfin itérer d'une manière qui nous renvoie à la fois la position et la valeur:

```
for (position, valeur) in enumerate(V)
    println("La position $position contient la valeur $valeur")
end
```

```
La position 1 contient la valeur 6
La position 2 contient la valeur 8
La position 3 contient la valeur 7
La position 4 contient la valeur 3
La position 5 contient la valeur 3
La position 6 contient la valeur 8
```

```
La position 7 contient la valeur 7
La position 8 contient la valeur 1
La position 9 contient la valeur 3
La position 10 contient la valeur 2
La position 11 contient la valeur 8
La position 12 contient la valeur 3
```

Nombres (pseudo)-aléatoires

```
import Random
Random.seed!(2045)
```

```
Random.TaskLocalRNG()
```

Un automate cellulaire pour la pigmentation

État initial

```
"""
    etat_initial(rows, cols, p_activation)

Initialise une grille avec des cellules activées aléatoirement.

Arguments:
- `rows::Int`: Nombre de lignes dans la grille
- `cols::Int`: Nombre de colonnes dans la grille
- `p_activation::Float64`: Probabilité d'activation initiale

Retourne:
- `Array{Bool, 2}`: Grille initialisée
"""
function etat_initial(rows, cols, p_activation)
    lattice = zeros(Bool, rows, cols)
    for row in 1:rows
        for col in 1:cols
            lattice[row, col] = rand() < p_activation
        end
    end
    return lattice
end
```

```
Main.var"##369".etat_initial
```

Règles biologiques

Mise à jour de l'activation des cellules

Résultat final

```
using CairoMakie
```

Seed

```
"""
    voisins_valides(lattice, row, col, rayon)

Retourne les voisins valides d'une cellule dans un certain
rayon.

Arguments:
- `lattice::Array{Bool, 2}`: Grille de cellules
- `row::Int`: Ligne de la cellule
- `col::Int`: Colonne de la cellule
- `rayon::Int`: Rayon de recherche des voisins

Retourne:
- `Array{Bool, 2}`: Sous-grille des voisins valides
"""
function voisins_valides(lattice, row, col, rayon)
    d_lignes = max(row - rayon, 1)
    f_lignes = min(row + rayon, size(lattice, 1))
    d_colonnes = max(col - rayon, 1)
    f_colonnes = min(col + rayon, size(lattice, 2))
    return lattice[d_lignes:f_lignes, d_colonnes:f_colonnes]
end

"""
    nombre_voisins(lattice, row, col, rayon)

Calcule le nombre de voisins d'une cellule dans un certain
rayon.

Arguments:
- `lattice::Array{Bool, 2}`: Grille de cellules
- `row::Int`: Ligne de la cellule
- `col::Int`: Colonne de la cellule
- `rayon::Int`: Rayon de recherche des voisins

Retourne:
- `Int`: Nombre de voisins
"""
function nombre_voisins(lattice, row, col, rayon)
    voisinnage = voisins_valides(lattice, row, col, rayon)
    n_voisins = count(voisinnage)
    return n_voisins
end
```

```
"""
    nouvel_etat(Na, Ni, wa, wi)

Détermine le nouvel état d'une cellule en fonction du nombre de
voisins activés et inhibés.

Arguments:
- `Na::Int`: Nombre de voisins activés
- `Ni::Int`: Nombre de voisins inhibés
- `wa::Float64`: Poids de l'activation
- `wi::Float64`: Poids de l'inhibition

Retourne:
- `Bool`: Nouvel état de la cellule (true pour activé, false
pour désactivé)
"""
function nouvel_etat(Na, Ni, wa, wi)
    etat = wa * Na > wi * Ni
    return etat
end

"""
    afficher_matrice(matrice)

Affiche une matrice de cellules, où les cellules activées sont
représentées par '█' et les cellules désactivées par un espace.

Arguments:
- `matrice::Array{Bool, 2}`: Matrice de cellules à afficher

Retourne:
- Rien
"""
function afficher_matrice(matrice)
    for i in 1:size(matrice, 1)
        for j in 1:size(matrice, 2)
            if matrice[i, j] == 1
                print("█")
            else
                print(" ")
            end
        end
        println()
    end
end
```

```
Main.var"##369".afficher_matrice
```

Variables

```
wa = 1.0   # Poids de l'activation
wi = 0.12  # Poids de l'inhibition
Ra = 2  # Rayon d'activation
Ri = 9  # Rayon d'inhibition
```

```
lignes = 95  # Nombre de lignes dans la grille
colonnes = 65  # Nombre de colonnes dans la grille

p_activation = 0.05  # Probabilité d'activation initiale
temps = 100  # Nombre de générations à simuler
```

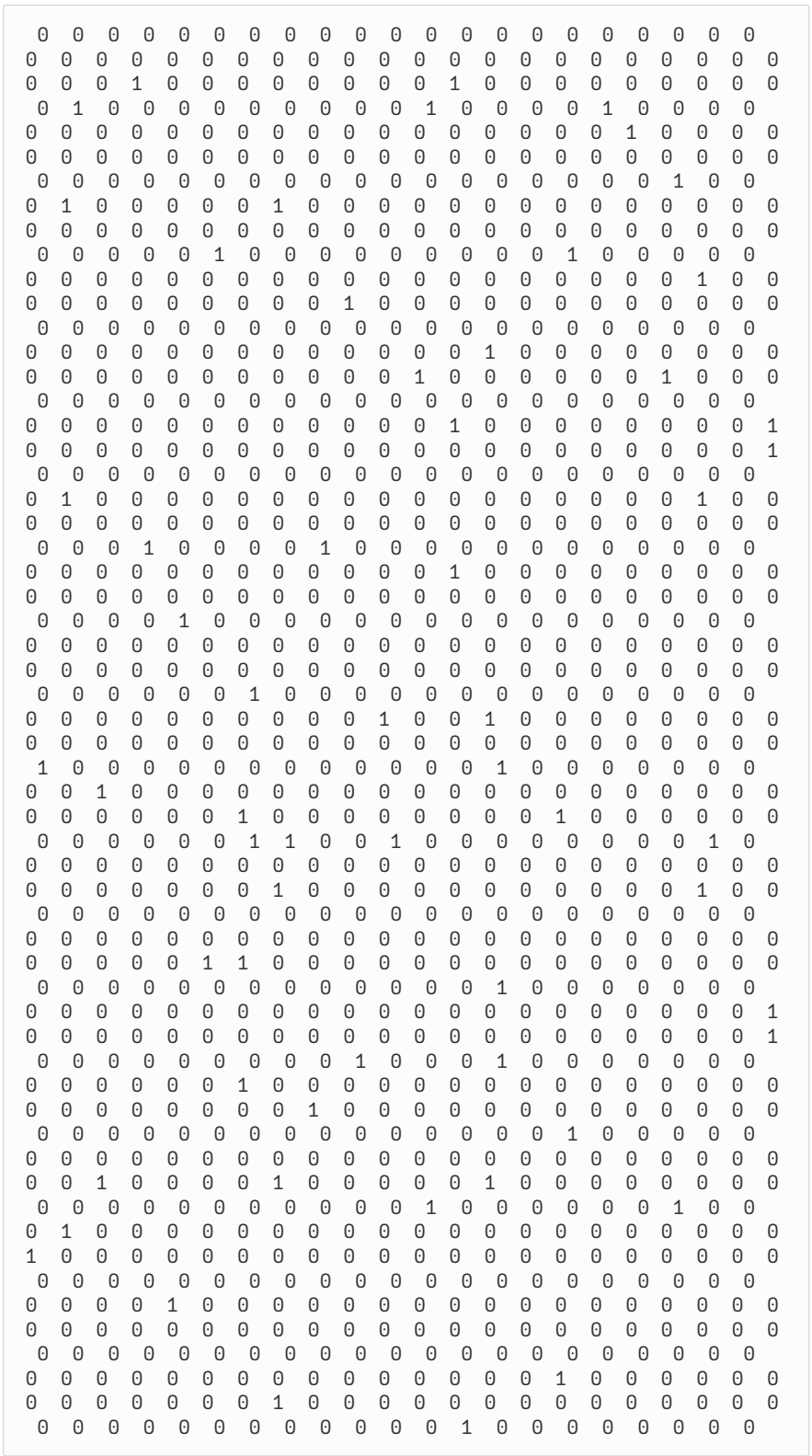
```
100
```

Initialisation de la grille

```
lattice = etat_initial(lignes, colonnes, p_activation)
```

```
95×65 Matrix{Bool}:
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0
 0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
 0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0
0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0
0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  1  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0
```

```
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  1  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0
 0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  1  0  1  1  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
 0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  1
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0
0  1  0  0  0  0  1  0  1  0  0  0  0  1  0  0  0  0  1  0  0  0
```


0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

```
0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  1
 0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0
0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0
0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0
1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
0  0  1  0  0  0  1  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  1  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0
```

```
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  1  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  1  0  0  1  0  0  0  0  1  0  0  0  0
1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  1  0  0  0  0  0  0  0  0  1  1  1  0  1  0  0  0  0
0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0
```

```
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Pour chaque génération

```
for gen in 1:temps
    # Grille au temps suivant
    temps_suivant = zeros(Bool, lignes, colonnes)
    # Pour chaque cellule
    for row in 1:lignes
        for col in 1:colonnes
            # Calcul du nombre de voisins activés et inhibés
            activation = nombre_voisins(lattice, row, col, Ra)
            inhibition = nombre_voisins(lattice, row, col, Ri)
            # Détermination du nouvel état de la cellule
            temps_suivant[row, col] = nouvel_etat(activation,
inhibition, wa, wi)
        end
    end
    for i in 1:lignes
        for j in 1:colonnes
            lattice[i, j] = temps_suivant[i, j]
        end
    end
end


# Visualisation de type heatmap
heatmap(
    # On passe d'abord l'objet a visualiser
    lattice,
    # Puis on fixe les deux couleurs à blanc et noir
    # pour resp. `false` et `true`
    colormap=[:white, :black],
    # On spécifie que les cellules du heatmap
    # sont des carrés
    axis=(; aspect=DataAspect()),
    # Et on fixe enfin un plus grand nombre de pixles pour avoir
    # une meilleure résolution
    figure=(; figure_padding=0)
)

# On termine enfin cette figure en retirant les axes et les
```

```
graduations,
# puis en affichant la figure finale
hidespines!(current_axis())
hidedecorations!(current_axis())
current_figure()
```