TP 9 bis : Percolation

La percolation [a] désigne le passage d'un fluide à travers un solide poreux. Ce terme fait bien entendu référence au café produit par le passage de l'eau à travers une poudre de café comprimée, mais dans un sens plus large peut aussi bien s'appliquer à l'infiltration des eaux de pluie jusqu'aux nappes phréatiques ou encore à la propagation des feux de forêt par contact entre les feuillages des arbres voisins.
L'étude scientifique des modèles de percolation s'est développée à partir du milieu du XX^e^ siècle et touche aujourd'hui de nombreuses disciplines, allant des mathématiques à l'économie en passant par la physique et la géologie.

Choix d'un modèle

Nous allons aborder certains phénomènes propres à la percolation par l'intermédiaire d'un modèle très simple : une grille carrée $n \times n$, chaque case pouvant être ouverte (avec une probabilité p) ou fermée (avec une probabilité $1-p$). La question à laquelle nous allons essayer de répondre est la suivante : est-il possible de joindre le haut et le bas de la grille par une succession de cases ouvertes adjacentes ?

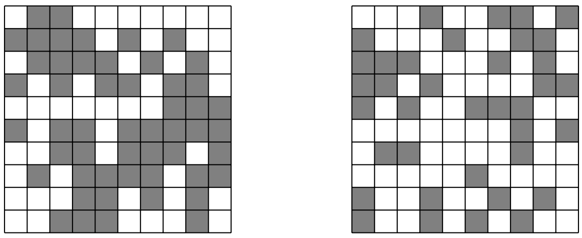

Figure 1 – *deux exemples de grilles* 10×10 *; la percolation n'est possible que dans le second cas (les cases ouvertes sont les cases blanches).*

On conçoit aisément que la réussite ou non de la percolation dépend beaucoup de p : plus celle-ci est grande, plus les chances de réussite sont importantes. Nous auront l'occasion d'observer l'existence pour de grandes valeurs de n d'un seuil critique p_0 en delà duquel la percolation a toutes les chances de réussir et en deçà duquel la percolation échoue presque à chaque fois.

Création et visualisation de la grille

Nous allons représenter la grille par le type `array` du module numpy, bien adapté à la manipulation de tableaux multidimensionnels. La fonction `zeros((n, p))` renvoie un tableau de n lignes et p colonnes contenant le nombre flottant 0.0 dans chacune de ses cases. Une fois un tableau `tab` créé, la case d'indice (i, j) est référencée par `tab[i][j]` et peut être lue et modifiée (comme d'habitude, les indices débutent à 0).

On notera que si `tab` est un tableau, la méthode `n, p = tab.shape` permet d'obtenir le nombre de lignes et de colonnes de ce dernier.

Dans la suite de ce document, on représentera une grille de percolation par un tableau $n \times n$, les cases fermées contenant le nombre flottant 0.0 et les cases ouvertes le nombre flottant 1.0.

1. Définir une fonction nommée `creation_grille` à deux paramètres : un nombre réel p (qu'on supposera dans l'intervalle $[0,1[$) et un entier naturel n, qui renvoie un tableau $n \times n$ dans lequel chaque case sera ouverte avec la probabilité p et fermée sinon.

Pour visualiser simplement la grille, nous allons utiliser la fonction `matshow` du module matplotlib.pyplot : appliquée à un tableau, celle-ci présente ce dernier sous forme de cases colorées en fonction de leur valeur [b] (suivant une échelle chromatique allant par défaut du bleu au rouge). La case d'indice $(0, 0)$ est située en haut à gauche et les lignes sont présentées horizontalement.
Puisque nos grilles ne contiennent pour l'instant que les valeurs 0 ou 1, les cases fermées apparaîtrons en bleu, et les cases ouvertes, en rouge.

a. du latin *percolare* : couler à travers.
b. Ne pas oublier la fonction show pour afficher la grille si le mode interactif est desactivé.

Percolation

Une fois la grille crée, les cases ouvertes de la première ligne sont remplies par un fluide, ce qui sera représenté par la valeur 0.5 dans les cases correspondantes (couleur verte par défaut pour `matshow`). Le fluide pourra ensuite être diffusé à chacune des cases ouvertes voisines d'une case contenant déjà le fluide jusqu'à remplir toutes les cases ouvertes possibles.

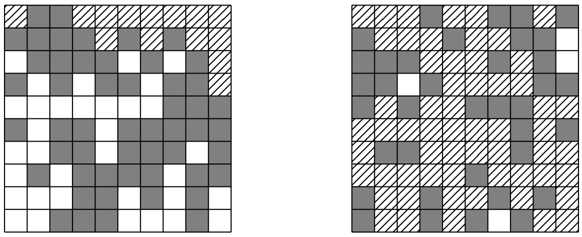

FIGURE 2 – *les deux grilles de la figure 1, une fois le processus de percolation terminé (le fluide est représenté par des hachures).*

2. Écrire une fonction `percolation` qui prend en argument une grille et qui remplit de fluide celle-ci.

On dit que la percolation est réussie lorsqu'à la fin du processus au moins une des cases de la dernière ligne est remplie du fluide.

3. Écrire une fonction `teste_percolation` qui prend en argument un réel $p \in [0,1[$ et un entier $n \in \mathbb{N}^*$, crée une grille, effectue la percolation et retourne :
   — `True` lorsque la percolation est réussie ;
   — `False` dans le cas contraire.

Seuil critique

Nous allons désormais travailler avec des grilles de taille 100×100 [c]. Faire quelques essais de percolation avec différentes valeurs de p. Vous observerez assez vite qu'il semble exister un seuil p_0 en deçà duquel la percolation échoue presque à chaque fois, et au delà duquel celle-ci réussit presque à chaque fois.

4. Nous allons noter $\mathrm{P}(p)$ la probabilité pour le fluide de traverser la grille. Proposer une démarche simple pour déterminer une valeur approchée de celle-ci en fonction de p, et rédiger la fonction `proba` correspondante.

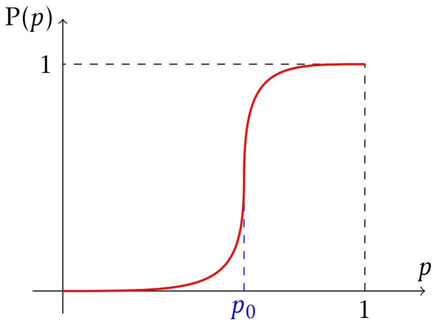

FIGURE 3 – *L'allure théorique du graphe de la fonction* P.

5. À l'aide de cette fonction, chercher à évaluer le plus précisément possible la valeur numérique du seuil p_0.

Propriétés macroscopiques

A l'instar de la thermodynamique, on peut décrire le comportement d'un système lors d'une transition de phase en introduisant des propriétés macroscopiques. Dans le cas de la percolation, on peut par exemple définir la densité moyenne $d(p)$ de cases ouvertes atteintes par le fluide.

6. Définir une fonction `densite` qui prend en argument une grille dans laquelle la percolation a eu lieu et qui retourne la valeur de sa densité.
7. À l'aide d'un nombre raisonnable d'échantillons, définir alors la fonction `d` qui à une probabilité $p \in [0,1[$ associe la densité moyenne de la percolation dans une grille 100×100.
8. Tracer le graphe de $d(p)$ pour $p \in [0,1[$.

c. Baisser cette valeur si le temps de calcul sur votre ordinateur sont trop longs.