



CY CERGY PARIS UNIVERSITÉ

L3 PHYSIQUE
PROJET NUMÉRIQUE : MONTE CARLO

Projet intermédiaire : Percolation orientée

Élève :

Mizaan-abbas KATCHERA

Enseignant :

Andréas HONECKER

Mardi 07 Février 2023

Table des matières

Logiciel : Python - Replit	1
Représentation graphique	2
Résumé (bref)	3
Résultats et niveau de confiance	3

Logiciel : Python - Replit

```
# Importation des bibliothèques.

import numpy as np
import random as r
import matplotlib.pyplot as plt

# Définition de la fonction percolation orientée par liens.

def percolation_orientee(probabilite, nb_tests, dim):
    # Calcul des différentes probabilités de survie.
    probabilites = np.arange(probabilite + 1) / probabilite
    probabilites_survie = []
    for probabilite_actuelle in probabilites:
        percolation_reussie = 0
        # Exécute le test nb_tests fois.
        for num_test in range(nb_tests):
            # Crée un tableau rempli de zéros (dim x dim)
            reseau = np.zeros((dim, dim))
            for i in range(dim):
                for j in range(dim):
                    if r.random() <= probabilite_actuelle:
                        reseau[i][j] = 1
            # Vérifie s'il existe un chemin de la première à la dernière colonne.
            percolation_reussie += np.sum(reseau[:,0] * reseau[:, dim-1]) > 0
        # Stocke les résultats de chaque test
        probabilites_survie.append(percolation_reussie / nb_tests)
    # Calcul de la variance et de l'écart type
    variance = np.var(probabilites_survie)
    ecart_type = np.sqrt(variance)
    # Calcul de l'erreur (écart type divisé par la racine du nombre de tests)
    erreur = ecart_type / np.sqrt(nb_tests)
    # Graphe et résultats finaux.
    print("Graphique pour la dimension", dim, "affiche.")
    print("Le seuil de percolation est", probabilites[np.argmax(probabilites_survie)])
    plt.plot(probabilites, probabilites_survie, color='red')
    plt.errorbar(probabilites, probabilites_survie, yerr=erreur, fmt='.', color='blue')
    plt.show()

# Appele la fonction percolation_orientee pour différentes dimensions.
for dim in [10, 100, 1000, 10000]:
    percolation_orientee(100, 100, dim)
```

FIGURE 1 – Python 3.8.2 : Replit

Représentation graphique

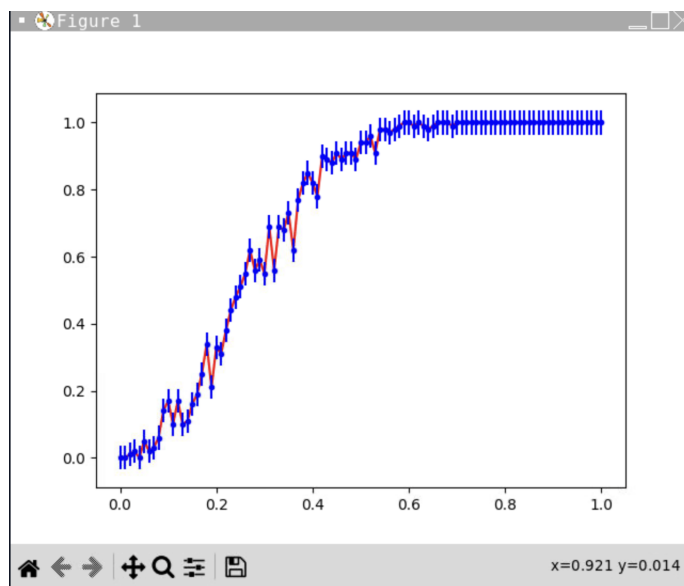


FIGURE 2 – Graphe pour la dimension 10

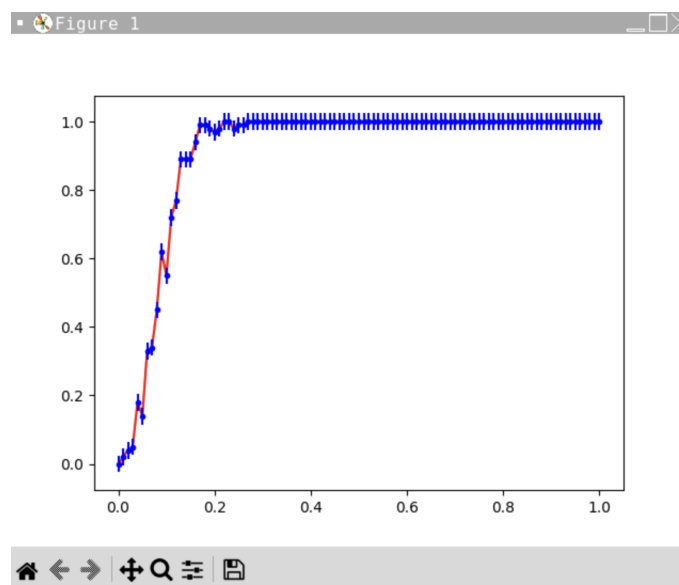


FIGURE 3 – Graphe pour la dimension 100

Les graphes en dimension 1000 et 10 000 étant trop long à générer, ils n'ont pas été ajoutés dans la section.

Résumé (bref)

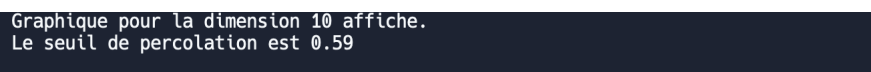
L'étude a pour but la simulation d'une analyse de percolation orientée (écoulement d'un liquide dans une direction préférentielle) pour un fluide perméable dans un réseau de différentes dimensions (10x10, 100x100, 1000x1000, 10000x10000) à travers l'obtention de la valeur du seuil de percolation, le tracé du graphique de la probabilité de survie en fonction de p pour différents t ainsi que les barres d'erreurs. La percolation (ou perméabilité du fluide dans notre exercice) est mesurée en fonction de la probabilité de passage de la fluide à travers les différents éléments du réseau. Notre programme utilise la bibliothèque numpy pour les matrices (sous forme de tableaux), la bibliothèque random pour obtenir des nombres aléatoires et la bibliothèque matplotlib pour afficher le graphique avec les barres d'erreur.

Dès lors, on évalue le seuil de percolation, autrement dit on détermine à quelle probabilité la fluide s'écoule à travers le réseau de manière continue. Le programme effectue plusieurs tests (correspondant au nbtests) avec des différentes probabilités (correspondant à probabillite) à travers des boucles for, et il enregistre le nombre de fois où la percolation (perméabilité) est réussie pour chaque probabilité. Ainsi, la probabilité de percolation réussie (probabilité de perméabilité) pour chaque probabilité de test est alors enregistrée dans la liste probabilitessurvie.

En outre, notre programme affiche les résultats sous forme de graphique et le tracé la probabilité de survie en fonction de la probabilité de perméabilité du fluide. Le seuil de percolation est déterminé en trouvant la probabilité maximale de percolation réussie (perméabilité). Enfin, notre programme est exécuté pour différentes dimensions du réseau et la simulation de peut donc varier en fonction de la taille du réseau.

Résultats et niveau de confiance

L'affichage console en dimension 1000 et 10 000 étant trop long à générer, ils n'ont pas été ajoutés dans la section.



```
Graphique pour la dimension 10 affiche.
Le seuil de percolation est 0.59
```

FIGURE 4 – Console

L'erreur qui est associée au résultat dépend de notre niveau de confiance. En outre, plus le niveau de confiance est élevé, plus la précision de valeur approché est fiable. Dans notre programme, l'erreur est défini par $\text{ecarttype}/\text{np.sqrt}(\text{nbtests})$. Si le nombre de tests est assez élevés alors la valeur du seuil de percolation sera comprise dans un intervalle de confiance correspondant à l'erreur avec une probabilité de 59 % environ. (en dimension 10 dans notre cas). De plus, la dimension influe aussi grandement sur le seuil de percolation le rapprochant de 1, il faut donc augmenter le nombre de tests pour une meilleure précision. On peut donc augmenter notre niveau de confiance en augmentant le nombre de tests et ainsi obtenir l'erreur associée à la valeur estimée mais aussi augmenté la complexité de notre programme (long à la réponse).