

Programme de colle 21

Semaine du 23 mars 2026

Chapitre 18 : Continuité

➤ Cours à connaître

- ✓ Définition continuité, continuité à droite et continuité à gauche en un point.
- ✓ Prolongement par continuité.
- ✓ Continuité et limites : composition avec une fonction continue et point fixe.
- ✓ Continuité sur un intervalle.
- ✓ Opérations et continuité.
- ✓ TVI et théorème de la bijection.

➤ Exercices type

- ✓ Étudier la continuité d'une fonction en un point (calcul de limite)
- ✓ Étudier la continuité d'une fonction sur un intervalle (utiliser les fonctions usuelles)
- ✓ Étudier la possibilité d'un prolongement continu en un point.
- ✓ Utiliser le TVI pour déterminer l'existence de solutions à des équations en étudiant les fonctions associées. (On pensera à se ramener à des équations du type $f(x) = 0$.)
- ✓ Utiliser le TVI ou le théorème de la bijection pour déterminer l'existence (et l'unicité) de solutions à des équations en étudiant les fonctions associées. (On pensera à se ramener à des équations du type $f(x) = 0$.)
- ✓ Exercice plus long et guidé : étude d'une suite (u_n) définie par $f_n(u_n) = 0$ en appliquant le théorème de la bijection à la fonction f_n

Chapitre 19 : Systèmes linéaires

➤ Cours à connaître

- ✓ Définition système linéaire, coefficients du système, second membre, système homogène, systèmes équivalents.
- ✓ Système de Cramer
- ✓ Connaître les opérations élémentaires.
- ✓ Théorème sur le nombre de solutions d'un système.

➤ Exercices type

- ✓ Résoudre un système à l'aide de la méthode du pivot de Gauss sous forme de système.
- ✓ Résoudre un système de Cramer homogène : il a pour unique solution le n -uplet $(0, \dots, 0)$.

La résolution d'un système à l'aide du pivot de Gauss a été travaillée ce vendredi, les étudiants auront peut-être besoin d'un peu d'aide.

L'inversion d'une matrice sera travaillée la semaine prochaine.

Python : simulation d'expériences aléatoires

Il faut connaître les commandes de base pour générer des nombres aléatoires :

- Importer la librairie Random : `import numpy.random as rd`
- Créer un nombre aléatoire de $[0; 1[$: `rd.random()`
- Créer un vecteur de n nombres aléatoires de $[0; 1[$: `rd.random(n)`
- Créer un entier aléatoire entre p et n : `rd.randint(p, n+1)` (attention : comme dans `range`, la dernière valeur n'est pas comptée). On peut aussi ajouter un paramètre pour avoir un vecteur avec plusieurs entiers, par exemple `rd.randint(1,4,10)` donne 10 entiers entre 1 et 3.

Il faut savoir les utiliser pour simuler une expérience aléatoire : lancer de dé, Pile ou Face, ou plusieurs expériences simultanées.

➤ Écrire une fonction `Piece` qui simule un lancer de pièce à Pile ou Face

```
import numpy.random as rd
def Piece() :
    alea = rd.random()
    if alea < 1/2 :
        return "Pile"
    else :
        return "Face"
```

Autre version :

```
import numpy.random as rd
def Piece() :
    alea = rd.randint(1,3) # entier aléatoire égale à 1 ou 2
    if alea == 1 :
        return "Pile"
    else :
        return "Face"
```

➤ Écrire une fonction `De` qui prend un entier n en paramètre et simule un lancer de n dés

```
import numpy.random as rd
def De(n) :
    alea = rd.randint(1,7,n)
    return alea
```