



I - Les graphiques

---

- I.1 - Nuage de points
- I.2 - Fonction
- I.3 - Courbe paramétrée
- I.4 - Régression linéaire
- I.5 - Barres d'erreur

II - Équations différentielles

---

- II.1 - Méthode d'Euler explicite
- II.2 - Ordre 1 avec « odeint »
- II.3 - Ordre 2 avec « odeint »

III - Analyse statistique

---

- III.1 - Incertitude de type B
- III.2 - Composition des incertitudes
- III.3 - Tirage non uniforme

IV - Algèbre linéaire

---

- IV.1 - Intégration
- IV.2 - Dérivation

V - Résolution d'une équation

---

- V.1 - Méthode dichotomique
- V.2 - Fonction « bisect »

---

## Capacités exigibles du chapitre

---

- Utiliser les fonctions de base de la bibliothèque « matplotlib » pour représenter :
  - un nuage de points ; **I.1**
  - une fonction ; **I.2**
  - une courbe paramétrée. **I.3**
- Utiliser la fonction « polyfit » de la bibliothèque « numpy » pour effectuer une régression linéaire. **I.4**
- Mettre en œuvre la méthode d'Euler explicite afin de résoudre une ED d'ordre 1. **II.1**
- Utiliser la fonction « odeint » de la bibliothèque « scipy.integrate » afin de résoudre une ED d'ordre 1 ou 2. **II.2 et II.3**
- Utiliser la fonction « uniform » de la bibliothèque « numpy.random » pour simuler un tirage aléatoire uniforme. **III.1**
- Utiliser la fonction « normal » de la bibliothèque « numpy.random » pour simuler un tirage aléatoire non uniforme. **III.3**
- Mettre en œuvre la méthode des rectangles pour calculer une valeur approchée d'une intégrale sur un segment. **IV.1**
- Mettre en œuvre un algorithme pour calculer une valeur approchée de la dérivée d'une fonction. **IV.2**
- Mettre en œuvre une méthode dichotomique afin de résoudre une équation. **V.1**
- Utiliser la fonction « bisect » de la bibliothèque « scipy.optimize » afin de résoudre une équation. **V.2**