

XII - Python

Structures de programmation :

- définir une variable et l'utiliser dans un calcul.
- définir une fonction et l'appliquer en certaines valeurs (**def** / **return**).
- comprendre et écrire une boucle itérative (**for**).
- comprendre et écrire une boucle conditionnelle (**while**).
- comprendre et écrire une conditionnelle (**if** / **elif** / **else**).
- définir une liste d'entiers (**range**).
- définir une liste par compréhension (**[f(x) for x in range(a, b)]**).

Modules :

- utiliser des fonctions du module `numpy` (`exp`, `log`, `sqrt`).
- utiliser les fonctions du module `matplotlib.pyplot` pour effectuer des graphiques (`plt.plot(X, Y)`).

Analyse / Probabilités à densité :

- définir une fonction définie par morceaux.
- programmer la méthode de dichotomie.
- définir une suite à l'aide de son terme général.
- définir une suite récurrente à l'aide de variables auxiliaires.
- définir une suite récurrente en stockant les valeurs dans une liste.
- définir deux suites récurrentes couplées.
- définir une suite par récurrence lorsque la formule de récurrence dépend du rang.
- déterminer le premier indice à partir duquel le terme général d'une suite satisfait une inégalité.

Matrices :

- définir une matrice (`np.array`).
- effectuer le produit de matrices (`np.dot`) et l'utiliser pour calculer les termes généraux de suites définies par récurrence.
- calculer une somme cumulée et l'utiliser pour définir le terme général d'une série (`np.cumsum`).

Probabilités :

- utiliser le module `numpy.random` pour simuler des variables aléatoires.
- utiliser la `rd.rand` pour simuler une loi discrète quelconque.
- utiliser des simulations de lois géométriques (`rd.geom`) ou exponentielles (`rd.exp`)

Notes :

A full page of blank graph paper with a uniform grid of small squares. The grid consists of 20 columns and 20 rows, creating a total of 400 small square units. The lines are thin and black, set against a white background. There are no margins or additional markings on the page.