

Programme Python en MPSI

La prise en compte de capacités de codage en langage Python dans la formation des étudiants inclue l'utilisation de fonctions extraites de diverses bibliothèques. Elle vise à une meilleure appréhension des principes mis en oeuvre par les différents logiciels de traitement des données dont l'utilisation est par ailleurs toujours recommandée. Elle a aussi pour objectif de mobiliser ces capacités dans un contexte concret, celui de la physique et de la chimie. Cette formation par le codage permet également de développer des capacités utiles à la physique-chimie comme le raisonnement, la logique ou la décomposition d'un problème complexe en étapes plus simples.

Le tableau ci-dessous explicite ces outils ainsi que les capacités exigibles en fin de première année. Il sera complété dans le programme de physique-chimie de seconde année.

Domaines numériques	Capacités exigibles
1. Outils graphiques	
Représentation graphique d'un nuage de points.	Utiliser les fonctions de base de la bibliothèque matplotlib pour représenter un nuage de points.
Représentation graphique d'une fonction.	Utiliser les fonctions de base de la bibliothèque matplotlib pour tracer la courbe représentative d'une fonction.
Courbes planes paramétrées.	Utiliser les fonctions de base de la bibliothèque matplotlib pour tracer une courbe plane paramétrée.

Exemple imposé : En élec (filtre) : Simuler, à l'aide d'un langage de programmation, l'action d'un filtre sur un signal périodique dont le spectre est fourni. Mettre en évidence l'influence des caractéristiques du filtre sur l'opération de filtrage.

Exemple imposé : Méca (forces centrales) : A l'aide d'un langage de programmation, obtenir des trajectoires d'un point matériel soumis à un champ de force centrale conservatif.

2. Équations algébriques	
Résolution d'une équation algébrique ou d'une équation transcendante : méthode dichotomique.	Déterminer, en s'appuyant sur une représentation graphique, un intervalle adapté à la recherche numérique d'une racine par une méthode dichotomique. Mettre en oeuvre une méthode dichotomique afin de résoudre une équation avec une précision donnée. Utiliser la fonction bisect de la bibliothèque scipy.optimize (sa spécification étant fournie).

Exemple imposé : Chimie (Equilibres chimiques) : Déterminer, à l'aide d'un langage de programmation, l'état final d'un système, siège d'une transformation, modélisée par une réaction à partir des conditions initiales et valeur de la constante d'équilibre.

3. Intégration – Dérivation	
Calcul approché d'une intégrale sur un segment par la méthode des rectangles.	Mettre en oeuvre la méthode des rectangles pour calculer une valeur approchée d'une intégrale sur un segment.
Calcul approché du nombre dérivé d'une fonction en un point.	Utiliser un schéma numérique pour déterminer une valeur approchée du nombre dérivé d'une fonction en un point.

Exemple imposé : Méca (pendule pesant) : A l'aide d'un langage de programmation, mettre en évidence le non-isochronisme des oscillations.

4. Équations différentielles	
Équations différentielles d'ordre 1.	Mettre en œuvre la méthode d'Euler explicite afin de résoudre une équation différentielle d'ordre 1.
Équations différentielles d'ordre supérieur ou égal à 2	Transformer une équation différentielle d'ordre n en un système différentiel de n équations d'ordre 1. Utiliser la fonction odeint de la bibliothèque scipy.integrate (sa spécification étant fournie).

Exemple imposé : En électrocinétique (régime transitoire) : Mettre en oeuvre la méthode d'Euler à l'aide d'un langage de programmation pour simuler la réponse d'un système linéaire du premier ordre à une excitation de forme quelconque.

Exemple imposé : En Mécanique (position d'équilibre avec E_p): A l'aide d'un langage de programmation, résoudre numériquement une équation différentielle du deuxième ordre non linéaire et faire apparaître l'effet des termes non linéaires.

5. Probabilités – statistiques	
Variable aléatoire.	Utiliser les fonctions de base des bibliothèques random et/ou numpy (leurs spécifications étant fournies) pour réaliser des tirages d'une variable aléatoire. Utiliser la fonction hist de la bibliothèque matplotlib.pyplot (sa spécification étant fournie) pour représenter les résultats d'un ensemble de tirages d'une variable aléatoire. Déterminer la moyenne et l'écart-type d'un ensemble de tirages d'une variable aléatoire.
Régression linéaire.	Utiliser la fonction polyfit de la bibliothèque numpy (sa spécification étant fournie) pour exploiter des données. Utiliser la fonction random.normal de la bibliothèque numpy (sa spécification étant fournie) pour simuler un processus aléatoire.

En TP.