

Programme des capacités numériques en MP2I

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.
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 œuvre 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).
3. Intégration – Dérivation	
Calcul approché d'une intégrale sur un segment par la méthode des rectangles.	Mettre en œuvre 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.
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).
5. Probabilité – 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.

Exemples concrets fixés par le programme :

En Optique Géométrique : Tester, à l'aide d'un langage de programmation, le stigmatisme approché d'une lentille demi-boule pour les rayons proches de l'axe optique.

En Electrocinétique : - 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.

- A l'aide d'un langage de programmation, simuler la réponse d'un système linéaire du deuxième ordre à une excitation de forme quelconque.

- Simuler, à l'aide d'un langage de programmation, l'action d'un filtre d'ordre 1 ou 2 sur un signal périodique dont le spectre est fourni. Mettre en évidence l'influence des caractéristiques du filtre sur l'opération de filtrage.

En Mécanique : - Résoudre numériquement une équation différentielle du deuxième ordre non-linéaire et faire apparaître l'effet des termes non-linéaires.

- Exploiter, à l'aide d'un langage de programmation, des données astronomiques ou satellitaires pour tester les deuxième et troisième lois de Kepler.