

Informatique et Calcul Scientifique

Objectifs

04.06.2025

Introduction à Python

- ▶ Manipulation de variables
- ▶ Compréhension des différents types
- ▶ Syntaxe du code
- ▶ *Slicing*
- ▶ Fonctions natives : `print()` , `len()` , `input()`

Structures de contrôle

- ▶ Condition booléenne, opérateurs de comparaison
- ▶ Choix : `if cond1: ... elif cond2: ... else:`
- ▶ Parcours : `while condition` et `for var in iterable` et quand les utiliser.
- ▶ Itérer sur des indices VS des éléments
- ▶ Utilisation de compteurs dans nos boucles, remplissage d'objets (listes, dict, ..)
- ▶ Méthode `range(i,j,k)` pour créer un objet itérable

Fonctions et modularité

- ▶ Syntaxe d'une fonction :
 - ▶ Définition en début de code
 - ▶ Appel dans le code principal
 - ▶ Utilisation des valeurs renvoyées
- ▶ Définition d'un paramètre obligatoire VS par défaut
- ▶ Argument positionnel ou par mot-clé
- ▶ Nombre indéfini d'arguments (* ou **)
- ▶ Portée d'utilisation des variables
- ▶ Import des modules externes et des fonctions qui y sont définies

Listes

- ▶ Manipulation des listes :
 - ▶ Création de liste
 - ▶ Modification d'un élément d'une liste
 - ▶ Accès à un élément / une tranche d'une liste
 - ▶ Opération sur les listes
 - ▶ Parcours de listes
- ▶ Méthodes élémentaires vues en théorie. Si je ne veux pas que vous utilisez une fonction, je le préciserai.
 - ▶ `append()`
 - ▶ `insert()`
 - ▶ `pop()`
 - ▶ `remove()`
- ▶ Compréhension de listes
- ▶ Notion de mutabilité, et conséquence sur les copies superficielles de listes

Tuples et dictionnaires

- ▶ Manipulation :
 - ▶ Création
 - ▶ Modification d'un élément
 - ▶ Accès à un élément / une tranche
 - ▶ Opérations
 - ▶ Parcours
- ▶ Compréhension de tuple/dictionnaire

Exceptions

- ▶ Gestion des exceptions :
try: ... except: ... else: ... finally: ...
- ▶ Création d'une exception personnalisée (raise Exception)

NumPy et Matplotlib

- ▶ Compréhension générale d'un objet `ndarray` :
 - ▶ Création d'un ndarray (selon une référence ou non)
 - ▶ Opérations sur les ndarray
 - ▶ Quelques fonctions associées (pas celles du module `linalg`)
 - ▶ Masque booléen
 - ▶ Pas d'import de données
- ▶ Utilisation de quelques fonctions associées (qui seront précisées dans l'énoncé)
- ▶ `Matplotlib` : création et compréhension d'un graphique "simple"
 - ▶ `figure()` , `plot()` , `xlabel()` , `ylabel()` , `legend()` ,
`title()` , `show()`
 - ▶ Représentation d'une courbe continue, en traitillés, avec symbole, ...
- ▶ graphes moins avancés que dans les séries d'exos..

Algorithmique I

- ▶ Implémenter en Python un algorithme effectuant une tache spécifique sur la base d'instructions
- ▶ Estimation du temps de parcours $T(n)$ d'un algorithme dont le code est donné
- ▶ Notation de Landau ($\mathcal{O}(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$)
- ▶ Pour deux fonctions f et g données, savoir estimer si f est $\Theta(g)$ par exemple

- ▶ **Comprendre** l'algorithme de recherche binaire et ses variations
- ▶ **Comprendre** les algorithmes de tri et leurs variations :
 - ▶ tri par sélection
 - ▶ tri par insertion
 - ▶ tri à bulles (vu en exos)
- ▶ Le code de ces différents algorithmes vous sera donné à l'examen si nécessaire

Résolution d'équations non-linéaires

- ▶ Diverses définitions et résultats de théorèmes à connaître ainsi que les hypothèses devant être vérifiées
- ▶ Aucune démonstration mathématique ne sera demandée
- ▶ Connaissance des différents algorithmes :
 - ▶ Connaitre les algorithmes "en français", ainsi que les conditions sous lesquelles ceux-ci sont utilisables
 - ▶ Savoir compléter un code à trous décrivant un des algorithmes vus en cours ou en exos, ou une de leurs variations (différents q_k par exemple)
 - ▶ Savoir reproduire graphiquement ces algorithmes
 - ▶ Etre capable de "simuler" les étapes de ces algorithmes.
- ▶ Méthode de la bisection, des parties proportionnelles, méthode de la sécante, méthode de Newton, méthode de Picard (recherche de point fixe)

Intégration numérique

- ▶ Diverses définitions et résultats de théorèmes à connaître ainsi que les hypothèses devant être vérifiées
- ▶ Aucune démonstration mathématique ne sera demandée
- ▶ Construire les polynômes de Lagrange associés à un ensemble de points $\{t_i\}$, ou approchant une fonction en certains points
- ▶ Formules d'intégration vues en cours :
 - ▶ Formules à un point (point de gauche, point milieu, point de droite, de Riemann), deux points (trapèze), trois points (Simpson)
 - ▶ Sous leur forme composite et non-composite
 - ▶ Définition des poids de quadrature
 - ▶ Exactitude et évolution de l'erreur
- ▶ Savoir évaluer des intégrales numériquement "sur papier".