

## Outils Numériques pour l'Ingénieur·e en Physique

2023-2024

5N-028-Phy / ONIP

### Bloc 1 - Python et calcul scientifique ( 0%)

#### Concepts étudiés

- [NUM] Bases de Python pour l'ingénieur·e en Physique
- [PHYS] Mise en équation de systèmes régis par des équations différentielles
- [MATH] Systèmes d'équations linéaires
- [NUM] Résolutions numériques : calcul formel, équations différentielles, systèmes

#### Mots clefs

Python; Matrices (Numpy); Calcul formel (SymPy); Méthode d'Euler; Systèmes (control); Classes et objets

#### Sessions

- 0 Cours(s) - 1h30
- 0 TD(s) - 1h30
- 4 TD(s) Machine - 2h00
- 0 TP(s) - 4h30

#### Travail

Seul ou par équipe de 2

## Démystifier Python et résoudre des problèmes à l'aide d'outils numériques

Les physicien·es sont souvent confronté·es à des problèmes faisant appel à des résolutions numériques.

...

Ce bloc est constituée de trois grandes parties :

1. Bases du langage Python et premiers calculs
2. Résolution d'une équation différentielle (3 méthodes) :
  - (a) Calcul formel / symbolique
  - (b) Intégration numérique (Euler)
  - (c) Approche système
3. Un monde d'objets

Chacun des thèmes est présenté plus spécifiquement dans les pages suivantes.

### Acquis d'Apprentissage Visés

En résolvant ce problème, les étudiant·e·s seront capables de :

#### CÔTÉ NUMÉRIQUE

1. **Générer des signaux numériques** à partir de fonctions mathématiques
2. **Définir et documenter des fonctions** pour générer des signaux numériques
3. **Produire des figures** claires et légendées à partir de signaux numériques - incluant un titre, des axes, des légendes
4. [BONUS] **Construire des bibliothèques de fonctions**

#### CÔTÉ PHYSIQUE

1. **Analyser le contenu spectral** d'un signal électrique
2. **Déterminer les paramètres** d'une modulation d'amplitude
3. **Décoder** un signal modulé en amplitude

Institut d'Optique  
Graduate School, France  
<https://www.institutoptique.fr>

GitHub - Digital Methods

<https://github.com/IOGS-Digital-Methods>

## Livrables attendus

Pour valider cette session, il n'y a **pas de livrables** à fournir.

Cependant, vous devez **faire valider par un-e encadrant-e**, durant les séances, les différents travaux personnels à réaliser. Ces travaux sont repérés dans les fichiers PDF par les repères suivants :



**S'ENTRAINER**

Il sera en particulier évalué :

- Vos **bonnes pratiques en programmation**
  - Code propre / documenté
  - Utilisation de fonctions
- Pertinence des **résultats présentés**
- Analyse et **critiques des résultats** (aspect physique/mathématique)

Les figures générées devront être pertinentes et légendées (axes, titres...). Les différentes fonctions devront être documentées selon la norme PEP 257.

## Données de départ

Dans cette séquence, vous serez amenés à utiliser des fichiers contenant des **notions de Python à tester**. Ces fichiers se trouvent dans le répertoire EXEMPLES.

Les fichiers sont nommés selon un ordre chronologique. Des **notions plus théoriques** ainsi que des **astuces de programmation** sont proposées dans les fichiers au format PDF B1\_SN\_EXXX\_NNNN.PDF (où N correspond au numéro de la séance et XX au numéro de l'exemple).

## Ressources

Cette séquence est basée sur le langage Python.

Vous pouvez utiliser l'environnement **JupyterHub@Paris-Saclay** - <https://jupyterhub.ijclab.in2p3.fr/> ou l'environnement **Spyder 5** inclus dans *Anaconda 3*.

Des tutoriels Python (et sur les bibliothèques classiques : Numpy, Matplotlib or Scipy) sont disponibles à l'adresse : <http://lense.institutoptique.fr/python/>. Parmi ces tutoriels, nous vous suggérons de lire les suivants :

- Python / Bases du langage
  -
- Python / Bases scientifiques
  - Premiers pas avec Numpy
  - Premiers pas avec Matplotlib
  - Premiers pas avec Scipy

## Outils Numériques

Fonctions et bibliothèques conseillées :

- **Python**
  - type, size
  - print
- **Numpy** Gestion de matrices
  - linspace, logspace
  - zeros, ones
  - shape
- **Matplotlib** Affichage de données
  - plotly
  - figure, plot
  - subplot
  - legend, title
  - xlabel, ylabel
  - show
- **Sympy** Calcul symbolique
  - symbols, Functions
  - Derivative
  - dsolve, lambdify
- **Scipy** Fonctions scientifiques
  - integrate (sous-bibliothèque)
  - solve\_ivp
- **Control** Étude de systèmes
  - tf
  - bode\_plot
  - step\_response
  - impulse\_response

Outils avancés :

- **rcParams** de Matplotlib.pyplot pour l'amélioration de l'affichage de courbes