

# Outils Numériques pour l' Ingénieur.e en Photonique

Outils Numériques / Semestre 5  
Bloc Intro

## <https://lense.institutoptique.fr/ONIP/>



### Semestre 5

Ce module s'intéresse aux méthodes numériques utiles à tout ingénieur.e. L'idée est de construire une **boîte à outils de méthodes numériques** pour les étudiant.es en physique, en se basant sur le langage **Python** et les **bibliothèques standards en science**.

Une série de tutoriels pour Python en suivant le lien ci-après.

[Python For Science / LEnsE.tech](#)

Ce module est décomposé en **3 thèmes** de 4 séances chacun :

- Bloc Intro – Python scientifique
- Bloc AM – Traitement de données 1D
- Bloc Laser – Traitement de données 2D



Introduction ONIP-1

VERSION 2024



Intro Python

VERSION 2024



Python Scientifique

Bloc Intro

Données 1D

Bloc AM

Données 2D

Bloc Laser



Grille Evaluation Module ONIP

VERSION 2023



[github.com/IOGS-Digital-Methods](https://github.com/IOGS-Digital-Methods)

Pinned

semester-5 Public

TeX

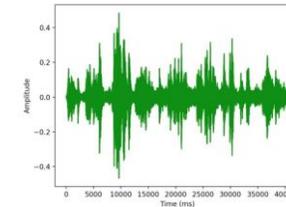
3 séances introductives (2h/séance)

2 blocs de 4 et 5 séances (2h/séance)

- Sur machine
- En binôme
- 2 encadrant.es par séance

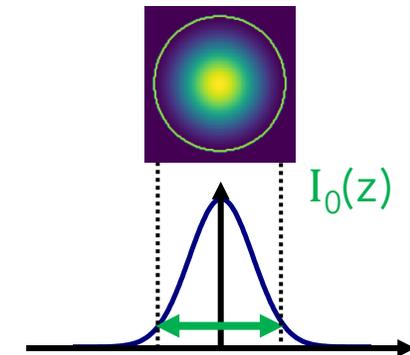
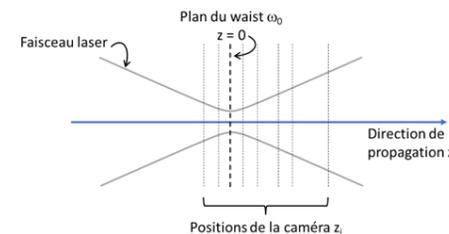
*Bloc AM : Traitement de données 1D*

**Problème 1** : signal modulé en amplitude / acquisition numérique



*Bloc Laser : Traitement de données 2D*

**Problème 2** : images d'un faisceau LASER en différents points d'un chemin optique



# Acquis d'Apprentissage Visés / ONIP

▶ UC dans l'UE Traitement de l'Information

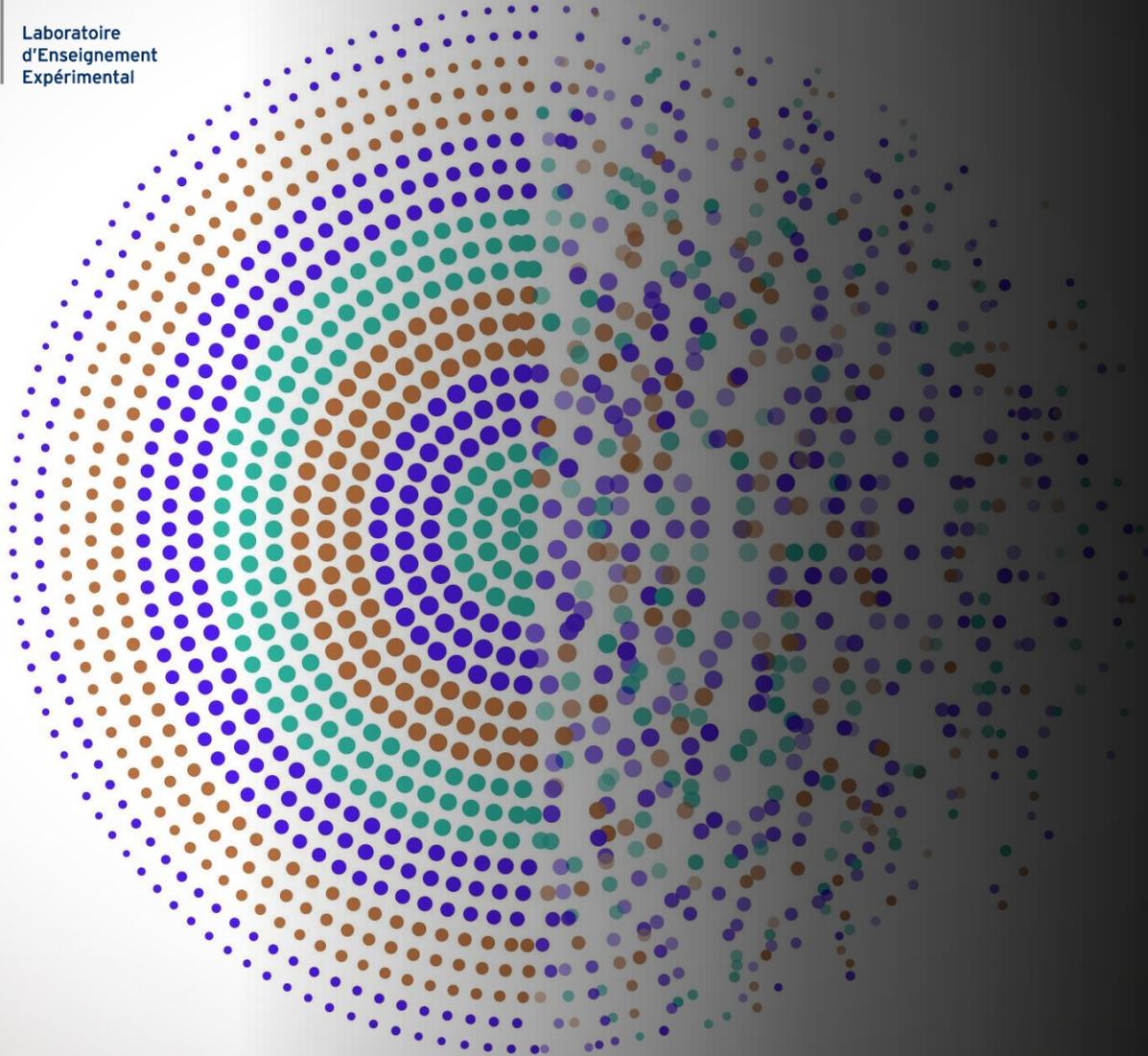
12 séances de TD Machine

▶ **Acquis d'Apprentissage Visés**

Être capable de **valider un modèle physique simple et fourni** à l'aide d'un outil de calcul scientifique

Être capable de **générer des graphiques scientifiques** légendés

Être capable de **d'écrire un script réutilisable dans un langage de haut niveau** (à but scientifique)



# Outils et méthode de travail

---

Outils Numériques / Semestre 5  
Bloc Intro

# Outils de développement

- Utilisation de **Python**

- *Python 3.9* (ou supérieur) via Anaconda 3



IDE

- **PyCharm** Community Edition



- **LEnsE.tech**

LEnsE.TECH



- <https://iogs-lense-training.github.io/python-for-science/>

- Travail seul ou en binôme
- Exercices prévus par séance
  - *Chacun son rythme*
  - *Correction « magistrale » pour certains exercices*
- « Au secours ! », je suis perdu·e

**Merci de rendre les supports imprimés et plastifiés des sujets aux encadrant·es**

## <https://lense.institutoptique.fr/ONIP/>

### Semestre 5

Ce module s'intéresse aux méthodes numériques utiles à tout ingénieur.e. L'idée est de construire une boîte à outils de méthodes numériques pour les étudiant.es en physique, en se basant sur le langage Python et les bibliothèques standards en science.

Une série de tutoriels pour Python en suivant le lien ci-après.

[Python For Science / LEnsE.tech](#)

Ce module est décomposé en 3 thèmes de 4 séances chacun :

- Bloc Intro – Python scientifique
- Bloc AM – Traitement de données 1D
- Bloc Laser – Traitement de données 2D

	<b>Introduction ONIP-1</b> VERSION 2024		
	<b>Intro Python</b> VERSION 2024		
	<b>Python Scientifique</b> <b>Bloc Intro</b>	<b>Données 1D</b> <b>Bloc AM</b>	<b>Données 2D</b> <b>Bloc Laser</b>
	<b>Grille Evaluation Module ONIP</b> VERSION 2023		

### Contenu

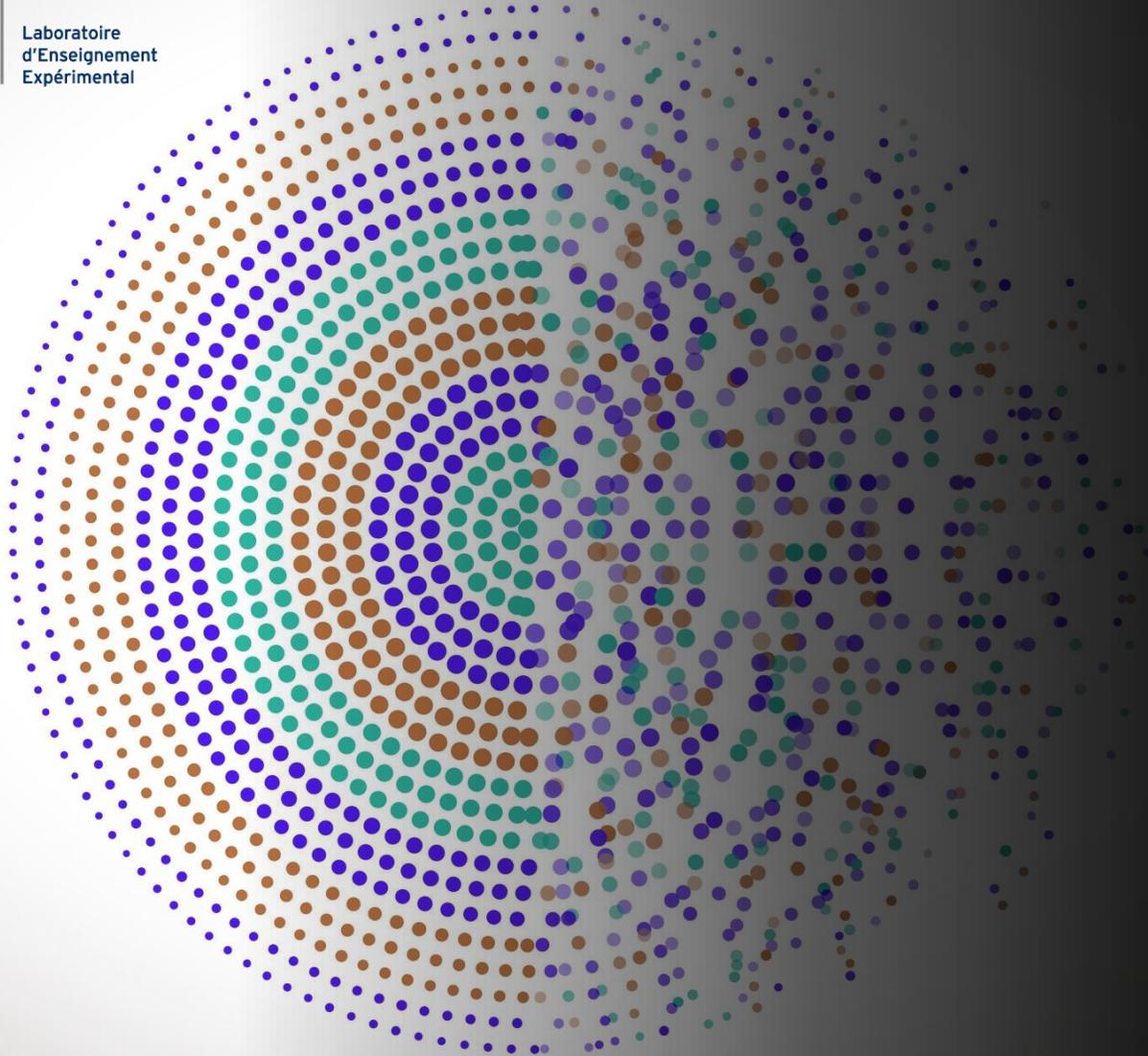
	<b>Sujet Bloc 1</b> VERSION 2024
<b>Python</b>	
	<b>Intro Python</b> VERSION 2024
<b>PyCharm et les méthodes de débogage</b>	
	<b>Répertoire vers les codes à tester</b> VERSION 2023
<b>Notions de base</b>	
	<b>Numpy / Meshgrid</b> VERSION 2024

**Merci de rendre les supports imprimés et plastifiés des sujets aux encadrant·es**

## Exercice 4 / Génération de signaux et affichage

Notions : *Functions* | *Numpy Basics - Using Vectors* | *Matplotlib Basics - Scientist figure*

1. Créer un vecteur *temps* de 101 points régulièrement répartis entre 0 et 1 s. Quelle est la période d'échantillonnage ?
2. Importer la bibliothèque `matplotlib.pyplot`.
3. Tracer une sinusoïde de période 50 ms en rouge. Ajouter un titre, des axes et une légende au graphique.
4. Que pensez-vous du résultat ? Améliorer le résultat.
5. Tracer sur le même graphique une sinusoïde de période 20 ms en bleu.
6. Faire un zoom pour n'afficher que quelques périodes des deux sinusoïdes.
7. Faire une fonction qui prend comme argument la période de la sinusoïde, tester-la avec une période de 30 ms.



# Bloc Intro Séance 1 PyCharm

---

Outils Numériques / Semestre 5  
Bloc Intro

# PyCharm et le débogage



The screenshot displays the PyCharm IDE interface. On the left, the 'Project' view shows a folder named 'first\_project' containing a file 'main.py'. The main editor window shows the content of 'main.py' with a red dot indicating a breakpoint on line 9. The code in the editor is as follows:

```
1 # This is a sample Python script.
2
3 # Press Maj+F10 to execute it or replace it with your code.
4 # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7 def print_hi(name):
8     # Use a breakpoint in the code line below to debug your script.
9     print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17
```

The status bar at the bottom indicates 'Scanning files to index...', 'Show all (3)', 'CRLF', 'UTF-8', '4 spaces', and 'Python 3.10'.

# Séance 1 / first\_try.py

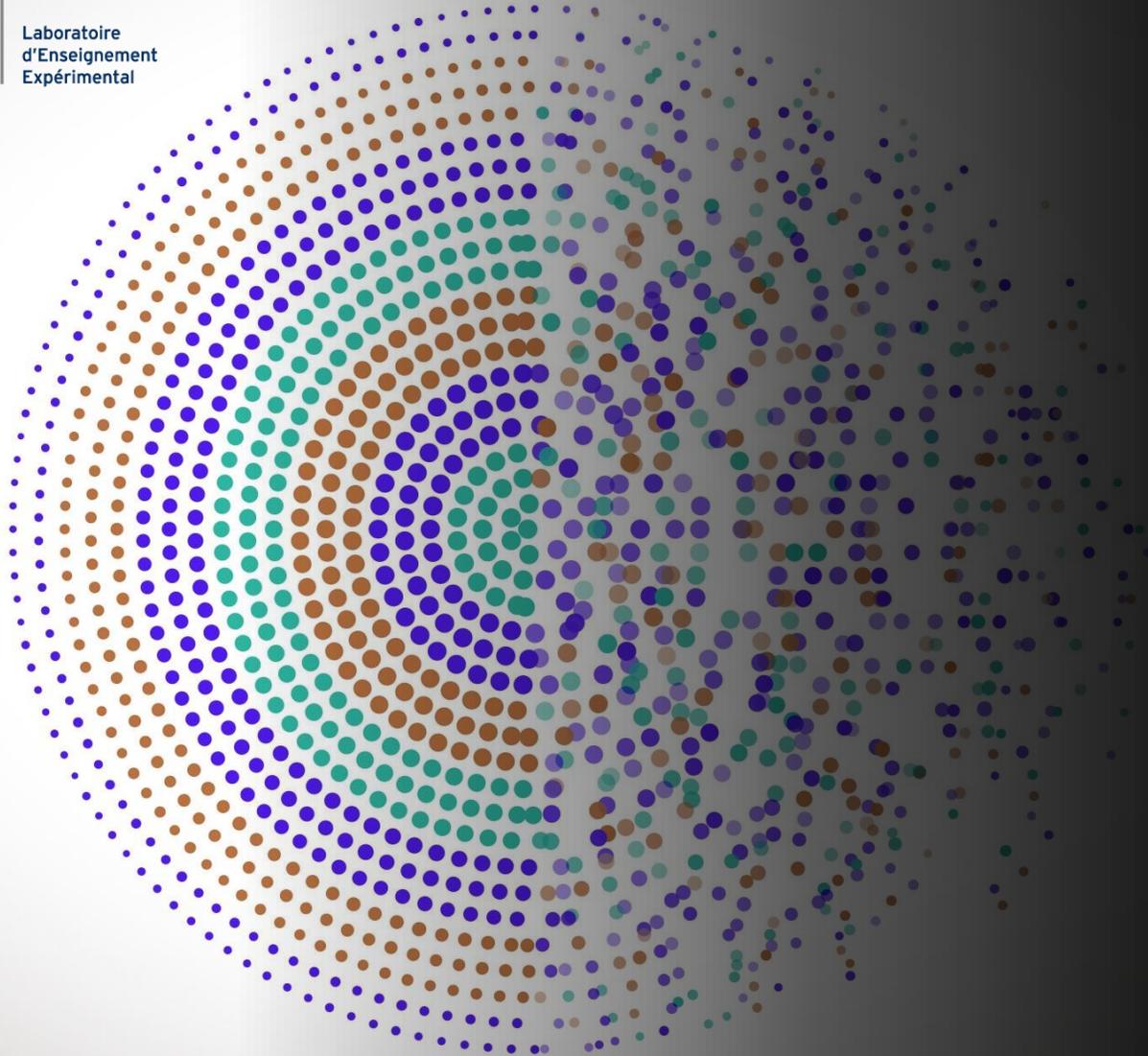
```
import numpy as nu

def fonctionA(a, b, c):
    return a * nu.sin(2*3.14 * b * c)

if __name__ == '__main__':
    a1=20
    b1=850
    c1=nu.linspace(0,50, 101)

    r1 = fonctionA(a1, b1, c1)
    print(r1)

    plt.figure()
    plt.plot(c1, r1)
    plt.show()
```



---

# Bloc Intro Séances 2/3 Python Basics

---

Outils Numériques / Semestre 5  
Bloc Intro