

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

5N-028-PHY / ONIP-1

Bloc AM - Modulation AM ( 50%)

### Concepts étudiés

[PHYS] Modulation d'amplitude  
[MATH] Transformée de Fourier  
[NUM] Signaux numériques  
[NUM] Figures scientifiques

### Mots clefs

Fichier CSV ; Graphique scientifique ;  
Transformée de Fourier ; Modulation  
d'Amplitude ; Démodulation

### Sessions

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

### Travail

En binôme

## Afficher et traiter des données provenant d'instruments de mesure

Les **expériences scientifiques**, les **essais industriels** sur des systèmes ou bien encore des **résultats de simulation** produisent énormément de **données**. Ces données sont souvent sauvegardées sous forme de **fichiers formatés** (format normalisé ou interne aux entreprises/laboratoires).

Il est alors indispensable de pouvoir **afficher les données** contenues dans ce type de fichier de manière claire et sans ambiguïté, avant d'en **extraire des informations pertinentes** par un traitement adapté.

### Données à traiter

Dans cette séquence, vous serez amenés à utiliser des données provenant d'un fichier de points issu d'un **oscilloscope**. Le fichier se nomme B3\_DATA\_01.CSV (modulante sinusoïdale).

Le signal qu'il contient est un enregistrement d'une **transmission d'informations modulées en amplitude** par un signal porteur sinusoïdal.

Deux autres fichiers vous sont également proposés :

- B3\_DATA\_02.TXT contenant un signal sonore modulé en amplitude à déchiffrer...
  - Format de données binaire 64 / Modulante sinusoïdale / Fichier sonore : 24 kHz / 16 bits
- B3\_DATA\_03.TXT contenant un ensemble de signaux modulés en amplitude à l'aide de différentes porteuses.
  - Format de données binaire 64 / Modulantes sinusoïdales / Fichier sonore : 160 kHz / 16 bits

### Institut d'Optique

Graduate School, France

<https://www.institutoptique.fr>

### GitHub - Digital Methods

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

### Ressources

Cette séquence est basée sur le langage Python. Vous pouvez utiliser l'environnement **PyCharm**.

Des tutoriels Python (et sur les bibliothèques classiques : Numpy, Matplotlib or Scipy) sont disponibles à l'adresse : <http://lense.institutoptique.fr/python/>.

## Acquis d'Apprentissage Visés

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

1. **valider un modèle physique** simple et fourni à l'aide d'un outil de calcul scientifique
  - Transcrire/Traduire un modèle physique donné (sous forme d'équations) en algorithme numérique
  - Choisir les paramètres de tests adaptés et réfléchis (discrétisation du signal, échantillonnage correct...)
  - Analyser la pertinence des résultats obtenus (erreurs de calcul, divergence...)
2. **générer des graphiques scientifiques** légendés
  - Réaliser le graphique
  - Décrire les axes avec les grandeurs et les unités associées
  - Légender le graphique (titre, légende des courbes...)
3. **écrire un script réutilisable dans un langage de haut niveau** (à but scientifique)
  - Utiliser des fonctions du langage avec des paramètres adaptés
  - Ecrire des fonctions dans un langage de haut niveau afin de rendre des parties du code réutilisable
  - Fournir un code lisible et réutilisable (convention d'écriture dans le langage, commentaires, documentation...)
4. **calculer, afficher et utiliser la transformée de Fourier discrète** d'un signal
  - Représenter l'axe des fréquences
  - Savoir repérer graphiquement les composantes fréquentielles d'un signal dans un spectre
  - Lister les contraintes de la FFT / Hypothèses et propriétés (signaux périodiques, symétrie hermitienne...)

## Livrables attendus

**Vous aurez 10 minutes lors de la séance 4 pour présenter l'ensemble de vos résultats et de vos analyses.**

**Vos scripts devront être déposés sur la plateforme eCampus la veille de votre passage à l'oral.**

Le nom du fichier doit être de la forme suivante : *Gx\_NOM1\_NOM2.py* où *x* est le numéro de votre groupe de TD.

Lors de l'oral, vous serez amené à **présenter la démarche** ayant mené à vos résultats. Vous devrez **exécuter au moins une fois votre code** afin de valider son bon fonctionnement. *Vous pouvez également générer des graphiques légendés permettant de faciliter votre présentation.*

## Outils Numériques

**Fonctions et bibliothèques conseillées :**

- **Numpy** gestion de matrices :
  - **arange**
  - **linspace**
  - **logspace**
- **Matplotlib** affichage de données :
  - **plotly**
  - **figure, plot**
  - **subplot**
  - **legend, title**
  - **xlabel, ylabel**
  - **show**
- **Scipy** fonctions scientifiques :
  - **fftpack** sublibrary
  - **fft, ifft**
  - **fftshift**
  - **fftfreq**

**Outils avancés :**

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

# Étapes

## Étape 1 Lecture d'un fichier de points

- *Fichier* : *B3\_data\_01.csv* provenant d'un enregistrement sur un oscilloscope numérique.
- Lire un fichier texte / tableur
- Récupérer les données dans un vecteur
- Afficher le signal contenu dans le fichier

## Étape 2 Simulation du phénomène de modulation/démodulation AM

- Générer un signal périodique sinusoïdal
- Générer plusieurs signaux de tests (sinusoïdaux)
- Afficher la FFT de ces signaux
- Générer un signal modulé en amplitude et afficher la FFT de ces signaux
- Démoduler le signal et afficher la FFT

## Étape 3 Démodulation du signal

- Afficher la FFT du signal contenu dans le fichier de l'étape 1
- Identifier la fréquence de la porteuse
- Générer un signal sinusoïdal à cette fréquence et multiplier le signal initial avec cette porteuse
- Afficher la FFT de ce nouveau signal
- Déterminer une méthode pour démoduler l'information contenue dans le signal d'origine

## Étape 4 Démodulation d'un signal quelconque

- *Fichier* : *B3\_data\_02.txt* provenant d'une génération d'un fichier en base 64 (fichier sonore initial en 24 kHz et 16 bits).
- Lire un fichier et le décoder depuis le format base 64
- Récupérer les données dans un vecteur
- Afficher les signaux contenus dans le fichier
- Afficher la transformée de Fourier du signal et en déduire la fréquence de la porteuse
- Démoduler le signal sonore
- Jouer le son décodé

## Bonus 1 Démodulation multi-porteuse

- *Fichier* : *B3\_data\_03.txt* provenant d'une génération d'un fichier en base 64 (fichier sonore initial en 160 kHz et 16 bits). Multi-porteuses sinusoïdales.

## Bonus 2 Génération de fichiers modulés