

# Installation Matlab & Prise en Main

L'installation de MATLAB R2023a avec une licence étudiante nécessite de suivre quelques étapes. Voici la liste des instructions pour installer MATLAB R2023a avec les boîtes à outils spécifiques que vous avez mentionnées (Signal Processing Toolbox, Communications Toolbox, DSP System Toolbox) :

## Étape 1 : Préparation des Prérequis

Assurez-vous d'avoir les éléments suivants :

- Votre licence étudiante MATLAB valide.
- Un compte MathWorks (si vous n'en avez pas, vous devrez en créer un sur le site de MathWorks).
- Accès à Internet pour télécharger le logiciel.

## Étape 2 : Connexion à votre Compte MathWorks

1. Rendez-vous sur le site Web de MathWorks :  
[<https://www.mathworks.com/>](<https://www.mathworks.com/>)
2. Connectez-vous à votre compte MathWorks en utilisant vos identifiants.

## Étape 3 : Téléchargement de MATLAB R2023a

1. Une fois connecté, accédez à la section de téléchargement de MATLAB.
2. Sélectionnez la version R2023a à télécharger.
3. Choisissez votre système d'exploitation (Windows, macOS, Linux) et la version (32 bits ou 64 bits) en fonction de votre configuration.
4. Téléchargez le fichier d'installation de MATLAB R2023a.

**NB : Vous pouvez également télécharger des versions antérieures de MATLAB R2023a.**

## Étape 4 : Installation de MATLAB R2023a

1. Exécutez le fichier d'installation téléchargé.
2. Suivez les instructions de l'assistant d'installation.
3. Lorsque vous arrivez à l'étape de la sélection des produits à installer, assurez-vous de cocher les produits suivants :
  - MATLAB
  - Signal Processing Toolbox
  - Communications Toolbox
  - DSP System Toolbox
4. Terminez le processus d'installation en suivant les étapes restantes de l'assistant.

## Étape 5 : Activation avec la Licence Étudiante

1. Au cours du processus d'installation, vous serez invité à entrer la clé de licence étudiante que vous avez reçue de MathWorks.
2. Saisissez la clé de licence dans les champs prévus à cet effet.

3. Suivez les instructions pour activer votre licence.

## Étape 6 : Finalisation de l'Installation

1. Une fois l'installation terminée, vous pouvez lancer MATLAB R2023a.
2. Assurez-vous que les boîtes à outils que vous avez sélectionnées (Signal Processing Toolbox, Communications Toolbox, DSP System Toolbox) sont disponibles et fonctionnent correctement.

## Étape 7 : Prise en main de Matlab

Les étapes 1 à 6 devraient vous permettre d'installer MATLAB R2023a avec une licence étudiante, ainsi que les boîtes à outils spécifiques (Toolbox). Assurez-vous de suivre attentivement les instructions fournies lors du processus d'installation et d'activation de la licence.

Voici trois (03) exemples de code MATLAB pour la mise en œuvre de fonctions de communication numérique. Ces exemples vous permettront de vous familiariser avec les concepts de base de la communication numérique en utilisant MATLAB.

### Exemple 1 : Modulation d'Amplitude (AM)

L'exemple 1 illustre la modulation d'amplitude (AM) en utilisant MATLAB. L'AM est une technique de modulation dans laquelle l'amplitude du signal porteur est modulée en fonction du signal modulant (message). Le code MATLAB est donné ci-dessous.

Dans cet exemple :

1. Nous définissons les paramètres de base tels que la fréquence de la porteuse ( $f_c$ ), la fréquence d'échantillonnage ( $f_s$ ), et nous créons une séquence temporelle ( $t$ ).
2. Le signal modulant (message) est créé en tant que sinusoïde de 5 Hz.
3. Le signal modulé en AM est obtenu en multipliant la porteuse par une version modulée du message. Dans cet exemple, nous utilisons une modulation AM simple avec un indice de modulation de 0.5.
4. Les signaux sont tracés pour visualiser le signal modulant, la porteuse, et le signal modulé.

Vous pouvez exécuter ce code dans MATLAB pour voir comment la modulation d'amplitude fonctionne et comment les signaux sont affectés. Cela vous aidera à mieux comprendre ce concept fondamental de la communication numérique.

```
% Paramètres de base
fc = 1000; % Fréquence de la porteuse
fs = 10000; % Fréquence d'échantillonnage
t = 0:1/fs:1; % Création d'une séquence temporelle

% Signal modulant (message)
message = sin(2*pi*5*t); % Un signal sinusoïdal de 5 Hz

% Signal modulé en AM
signal_module = (1 + 0.5*message) .* cos(2*pi*fc*t);

% Tracé des signaux
subplot(3,1,1);
```

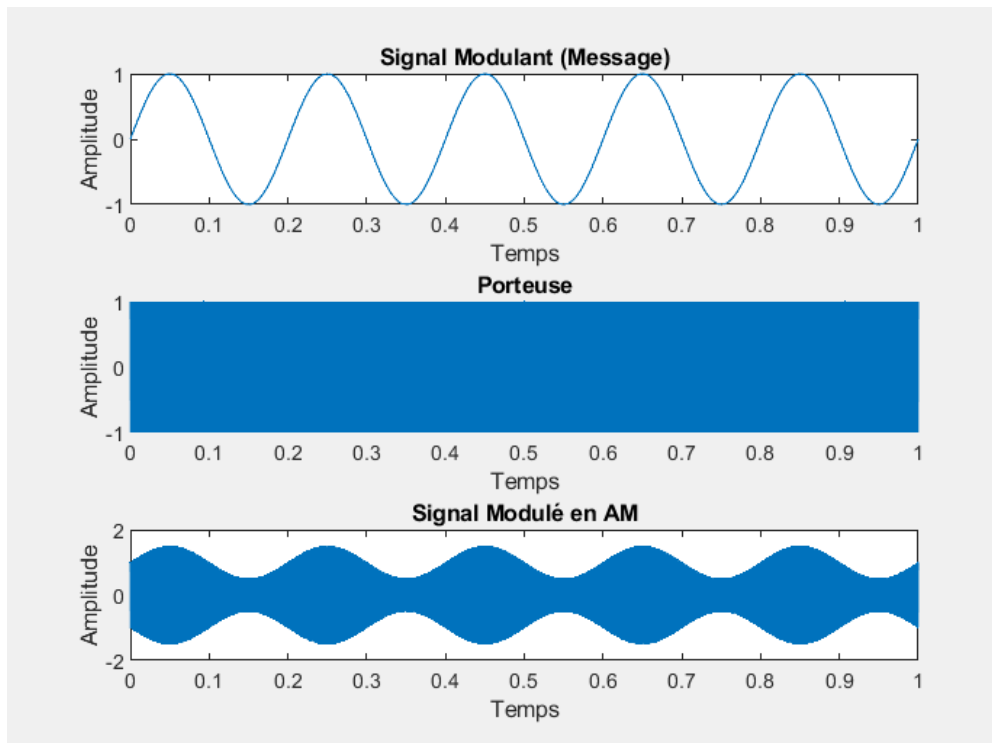
```

plot(t, message);
title('Signal Modulant (Message)');
xlabel('Temps');
ylabel('Amplitude');

subplot(3,1,2);
plot(t, cos(2*pi*fc*t));
title('Porteuse');
xlabel('Temps');
ylabel('Amplitude');

subplot(3,1,3);
plot(t, signal_module);
title('Signal Modulé en AM');
xlabel('Temps');
ylabel('Amplitude');

```



### Exemple 2 : Modulation de Phase (PSK)

L'exemple 2 concerne la modulation de phase (PSK) en utilisant MATLAB. La modulation de phase est un processus où la phase d'une onde porteuse est modifiée en fonction des données binaires d'entrée pour transmettre de l'information. Le code MATLAB pour illustrer la modulation de phase (PSK) est donné ci-dessous.

Dans cet exemple, nous générons un signal binaire aléatoire représentant les données d'entrée (0 ou 1). Les valeurs de phase (0 et  $\pi$ ) sont associées à ces données binaires. Le signal modulé en PSK est obtenu en modifiant la phase de l'onde porteuse en fonction des valeurs binaires.

**Vous pouvez exécuter ce code dans MATLAB pour voir comment la modulation de phase (PSK) fonctionne et comment les signaux modulés en PSK sont générés en fonction des données binaires. Vous observerez que le signal modulé varie en phase selon les valeurs binaires du message.**

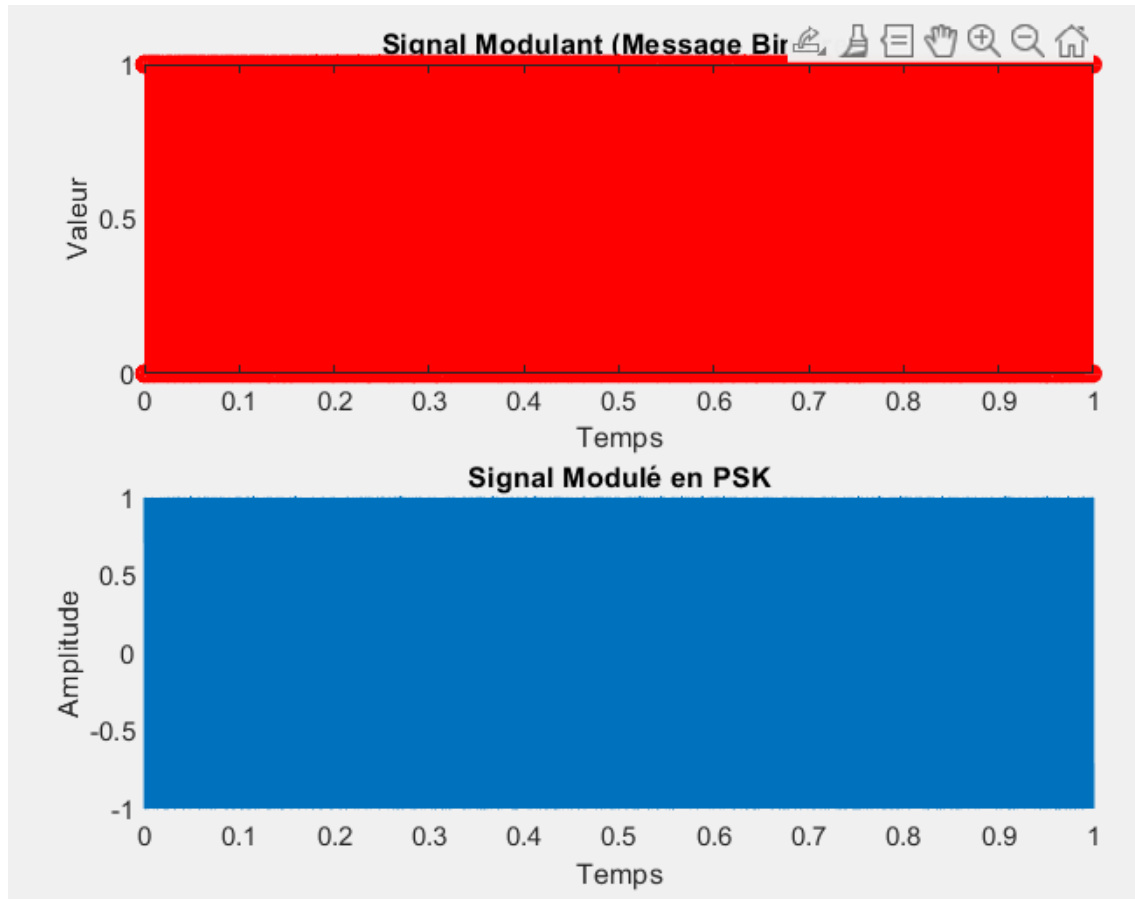
```
% Paramètres de base
fs = 10000; % Fréquence d'échantillonnage
t = 0:1/fs:1; % Création d'une séquence temporelle

% Signal modulant (message)
message = randi([0,1], 1, length(t)); % Signal binaire aléatoire

% Signal modulé en PSK
signal_module = cos(2*pi*fs*t + pi*message);

% Tracé des signaux
subplot(2,1,1);
stem(t, message, 'r');
title('Signal Modulant (Message Binaire)');
xlabel('Temps');
ylabel('Valeur');

subplot(2,1,2);
plot(t, signal_module);
title('Signal Modulé en PSK');
xlabel('Temps');
ylabel('Amplitude');
```



### Exemple 3 : Modulation de Fréquence (FM)

Pour l'exemple de modulation de fréquence (FM) en MATLAB. Le code qui génère un signal modulé en FM à partir d'un signal modulant sinusoïdal est donné ci-dessous.

Ce code génère un signal modulant (message) sinusoïdal de 5 Hz, puis le module en utilisant la modulation de fréquence (FM). La constante de modulation 'kf' contrôle l'ampleur de la modulation. Vous pouvez exécuter ce code dans l'environnement MATLAB pour visualiser le signal modulant, la porteuse et le signal modulé en FM.

L'exemple illustre comment la fréquence de la porteuse est modulée en fonction du signal modulant. La modulation de fréquence est couramment utilisée dans les communications radio, notamment dans les émissions FM.

```
% Paramètres de base
fs = 10000; % Fréquence d'échantillonnage
t = 0:1/fs:1; % Création d'une séquence temporelle

% Signal modulant (message)
message = sin(2*pi*5*t); % Signal sinusoïdal de 5 Hz

% Signal modulé en FM
kf = 50; % Sensibilité de la fréquence (constante de modulation)
signal_module = cos(2*pi*100*t + kf*message);
```

```
% Tracé des signaux
subplot(3,1,1);
plot(t, message);
title('Signal Modulant (Message)');
xlabel('Temps');
ylabel('Amplitude');

subplot(3,1,2);
plot(t, cos(2*pi*100*t));
title('Porteuse');
xlabel('Temps');
ylabel('Amplitude');

subplot(3,1,3);
plot(t, signal_module);
title('Signal Modulé en FM');
xlabel('Temps');
ylabel('Amplitude');
```

