

Etape 5 : Loi d'Ohm (épisode 2)

Notions abordées

- Création d'un tableau à l'aide de la bibliothèque numpy
- Appel et utilisation de la fonction polyfit de la bibliothèque numpy
- Création d'un tableau numpy à partir d'un calcul faisant intervenir d'autres tableaux numpy
- Affichage d'un tableau numpy
- Appel et affichage des valeurs d'un tableau numpy
- Affichage de deux courbes sur un même graphique
- Pointage et zoom sur une courbe

Référence pyspc

- [Les tableaux numpy](#)
- [Les modélisations](#)
- [Les fonctions](#)
- [Les graphiques - première partie](#)
- [Les graphiques - zoom et pointeur](#)

Consigne :

Etudier le programme ci-dessous puis effectuer la mise en situation présentée dans la dernière cellule.



On souhaite modéliser la caractéristique d'un dipôle ohmique, c'est-à-dire la courbe donnant les valeurs de la tension aux bornes du dipôle ohmique en fonction des valeurs de l'intensité du courant qui le traverse.

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

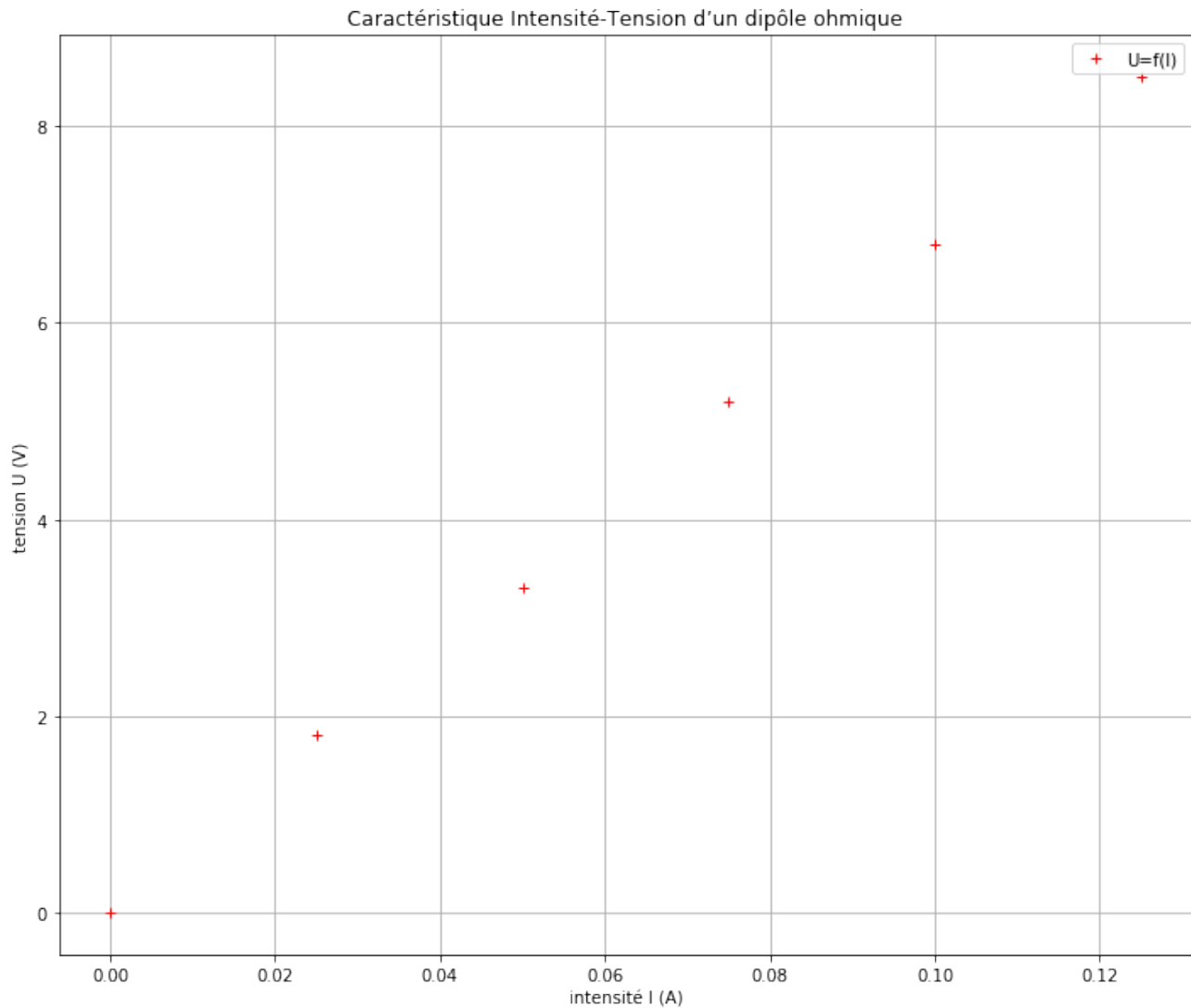
```
# Création des tableaux de valeurs avec la bibliothèque numpy

I=np.array([0,25e-3,50e-3,75e-3,100e-3,125e-3])
U=np.array([0,1.8,3.3,5.2,6.8,8.5])
```

In [3]:

```
# Reprise du programme vu précédemment et permettant d'afficher
# la caractéristique d'un dipole ohmique.

fig = plt.figure(figsize=(12,10))
plt.plot(I,U,'r+',label='U=f(I)')
plt.legend()
plt.xlabel("intensité I (A)")
plt.ylabel("tension U (V)")
plt.grid()
plt.title("Caractéristique Intensité-Tension "
          "dun dipôle ohmique")
plt.show()
```



Il s'agit maintenant de modéliser la courbe obtenue.

Exécutez le programme ci-dessous permettant de modéliser la courbe obtenue par une droite.

In [4]:

```
coeff=np.polyfit(I, U,1)
# La fonction polyfit (inclue dans la bibliothèque numpy)
# retourne un tableau numpy a une dimension (nommée ici coeff)
# contenant les coefficients d'un polynome U=f(I) de degré 1 (ici).

# Affichage du tableau coeff
print(coeff)

# Affichage des valeurs contenues dans le tableau avec une décimale
print ('{0:.1f}'.format(coeff[0]),
      '{0:.1f}'.format(coeff[1]))

# Création puis affichage d'un tableau numpy Umodel regroupant
```

```

# les valeurs de la tension modélisée par une fonction affine.
# Cette opération dite vectorisée ne peut pas se faire sur des listes,
# d'où la nécessité de créer des tableaux numpy.
Umodel = coeff[0]*I+coeff[1]

print(Umodel)

# Affichage de l'équation de la droite modélisée
print('U={0:.1f}'.format(coeff[0]), 'x I')

```

```

[6.78857143e+01 2.38095238e-02]
67.9 0.0
[0.02380952 1.72095238 3.41809524 5.1152381 6.81238095 8.50952381]
U=67.9 x I

```

1. Que représentent les trois paramètres ordonnés de la fonction polyfit ?
2. Que représente coeff[0] ?
3. Que représente coeff[1] ?

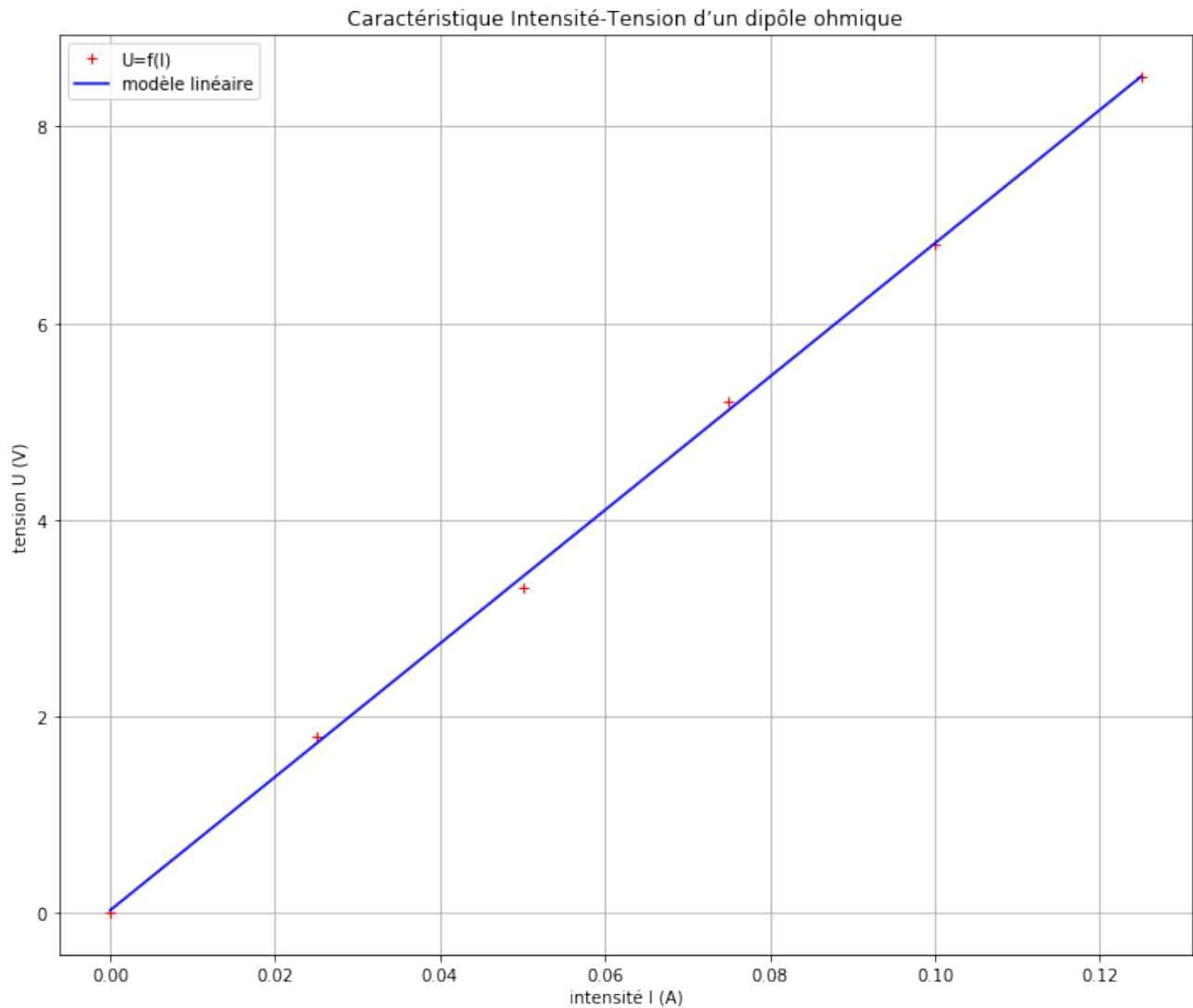
Affichez la droite modélisée grâce au programme ci-dessous.

In [5]:

```

fig = plt.figure(figsize=(12,10))
plt.plot(I,U,'r+',label='U=f(I)')
plt.plot(I,Umodel,'b',label='modèle linéaire')
plt.legend()
plt.xlabel("intensité I (A)")
plt.ylabel("tension U (V)")
plt.grid()
plt.title("Caractéristique Intensité-Tension "
          "dun dipôle ohmique")
plt.show()

```



Si vous désirez faire un zoom ou utiliser un pointeur sur la courbe cela est possible (uniquement dans le logiciel Jupyter Notebook (ENT, Anaconda) mais pas dans le logiciel JupyterLab) en important la bibliothèque ipywidgets qui permet d'afficher une fenêtre interactive.

Attention, ne pas écrire la ligne de code `%matplotlib inline` dans le programme pour voir s'afficher la fenêtre interactive. Ici, il sera nécessaire d'exécuter la cellule deux fois de suite afin de voir s'afficher cette fenêtre et la barre d'outil qui permet d'interagir avec la courbe.

In [7]:

```
import ipywidgets as widgets
%matplotlib widget

fig = plt.figure(figsize=(6,5))
plt.plot(I,U,'r+',label='U=f(I)')
plt.plot(I,Umodel,'b',label='modèle linéaire')
plt.legend()
plt.xlabel("intensité I (A)")
plt.ylabel("tension U (V)")
```

```
plt.grid()
plt.title("Caractéristique Intensité-Tension "
         "dun dipôle ohmique")
plt.show()
```

Canvas(toolbar=Toolbar(toolitems=[('Home', 'Reset original view', 'home', 'home'), ('Back', 'Back to

Mise en situation

- Ouvrir le programme *Activité : Modélisation d'un mouvement parabolique* et puis suivre les instructions pour compléter le programme.
- Vérifier votre travail si besoin avec le notebook *Correction : Modélisation d'un mouvement parabolique*