

Exercice : Linéarisation

```
# Import des modules nécessaires et définition de fonctions pour tracer les courbes
import numpy as np
import matplotlib.pyplot as plt
import seaborn
from sklearn.linear_model import LinearRegression
from IPython.display import display, Markdown

def plotIt(x,y, title="sortie", xlab="X", ylab="Y"):
    fig, ax = plt.subplots()
    ax.plot(x, y)
    ax.set_xlabel(xlab)
    ax.set_ylabel(ylab)
    ax.set_title(title)
    ax.grid(True, which='both')

    seaborn.despine(ax=ax, offset=0)

def plotScat(x,y, title="sortie", xlab="X", ylab="Y"):
    fig, ax = plt.subplots()
    ax.scatter(x, y)
    ax.set_xlabel(xlab)
    ax.set_ylabel(ylab)
    ax.set_title(title)
    ax.grid(True, which='both')

    seaborn.despine(ax=ax, offset=0)

def plotIt2(x1,y1,x2,y2,title="sortie", xlab="X", ylab="Y", legends=""):
    fig, ax = plt.subplots()
    ax.plot(x1, y1, x2,y2)
    ax.set_xlabel(xlab)
```

```

    ax.set_ylabel(ylab)
    ax.set_title(title)
    ax.grid(True, which='both')
    ax.legend(legends)

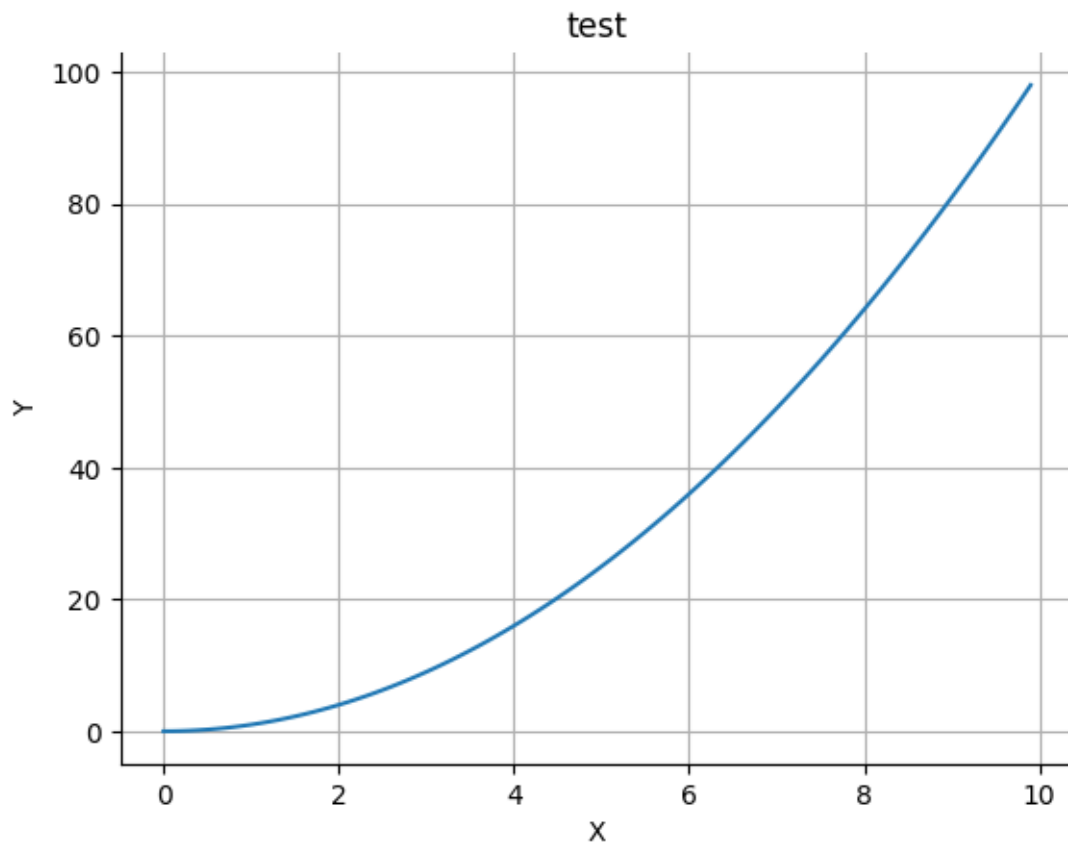
    seaborn.despine(ax=ax, offset=0)

def plotIt2s(x1,y1,x2,y2,title="sortie", xlab="X", ylab="Y", legends=""):
    fig, ax = plt.subplots()
    ax.plot(x1, y1,'rx')
    ax.plot(x2, y2)
    ax.set_xlabel(xlab)
    ax.set_ylabel(ylab)
    ax.set_title(title)
    ax.grid(True, which='both')
    ax.legend(legends)

    seaborn.despine(ax=ax, offset=0)

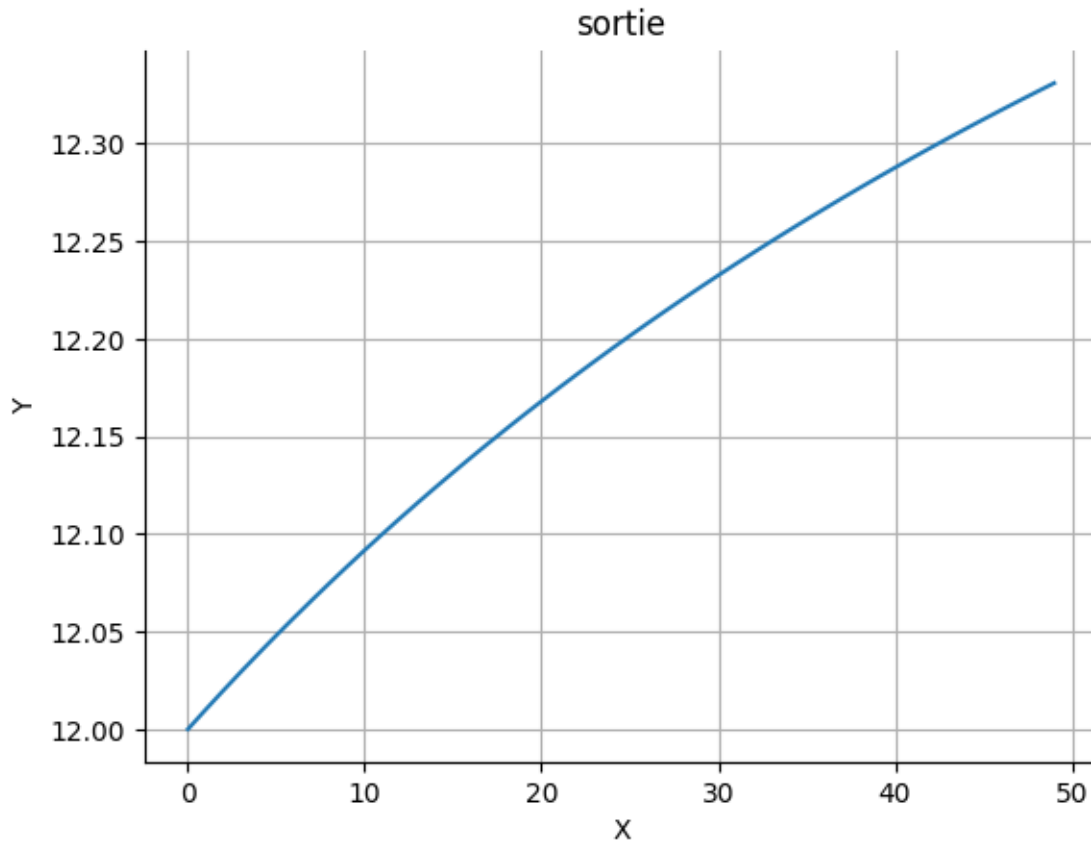
a=np.arange(0,10,0.1)
b=a**2
plotIt(a,b, title="test")

```



On génère des données de base sur la base d'un modèle avec une caractéristique non linéaire.

```
X=np.arange(0,50)  
X0=12  
X1=99  
Y=X0+X/(X+X1)  
plotIt(X,Y)
```



Identification avec une droite

On utilise la fonction `polyfit(X,Y,N)` pour calculer les coefficient du polynôme. Utilisez le manuel de la fonction pour savoir comment utiliser les résultats.

<https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html>

- Calculez les coefficient
- Définissez une fonction pour évaluer les valeurs
- Calculer les points
- Tracer la fonction

```
# Utilisez la fonction poly=np.polyfit(X, Y, 1), puis calculez les points obtenus avec le
```

```
# Solution
```

```
# Visualisation des résultats de l'identification  
poly
```

```
array([6.65509730e-03, 1.20241086e+01])
```

Comparaison avec les données

Plotter un graphique avec la différence entre les données de base et la courbe identifiée.

Identification avec une courbe quadratique

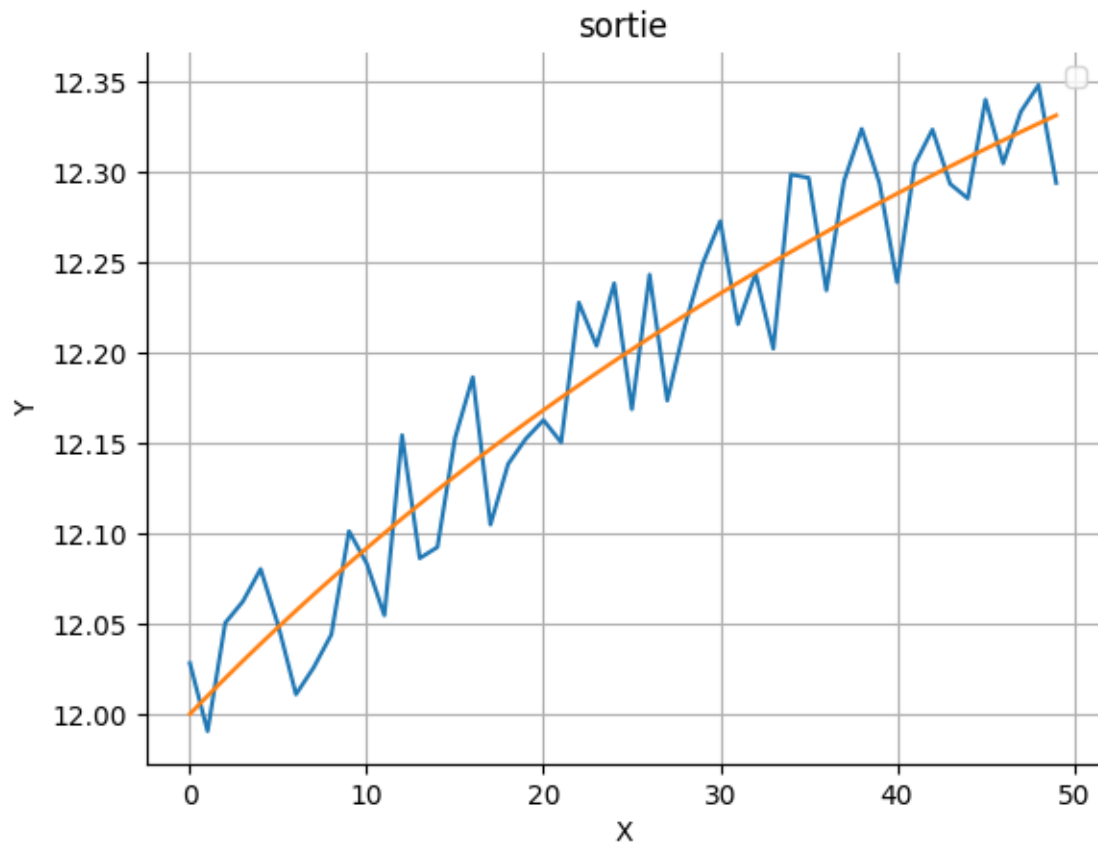
Recommencez l'identification avec une identification du 2e ordre

```
#Comparez les résultats
```

Influence du bruit

On ajoute du bruit aux données mesurées

```
Yn=X0+X/(X+X1)+(np.random.rand(len(X))-0.5)*0.1  
plotIt2(X,Yn,X,Y)
```



Refaire une identification pour évaluer l'influence du bruit sur l'identification. Tracer la courbe identifiée en la comparant à la courbe originale et à la courbe avec le bruit.