

Linéarisation d'un sonde de température NTC

```
# 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")

```

La linéarisation d'une caractéristique consiste à calculer l'équation de la tangente à la courbe autour d'un point donné. Le cas échéant, l'inversion de la courbe est plus simple.

Exemple avec une thermo-résistance NTC

La caractéristique de la sonde NTC est donnée par une relation de la forme suivante :

$$R(T) = R_0 \left(e^{\beta \left(\frac{1}{T} - \frac{1}{T_0} \right)} \right)$$

Attention : Les températures sont exprimées en Kelvin !

Le coefficient β peut être calculé pour deux points connus d'une sonde par la relation

$$\beta = \frac{T_1 \cdot T_2}{T_2 - T_1} \cdot \ln \left(\frac{R_1}{R_2} \right)$$

En pratique, le coefficient β est donné par le fabricant pour une plage de température donnée. (cf. datasheet NTCS0603)

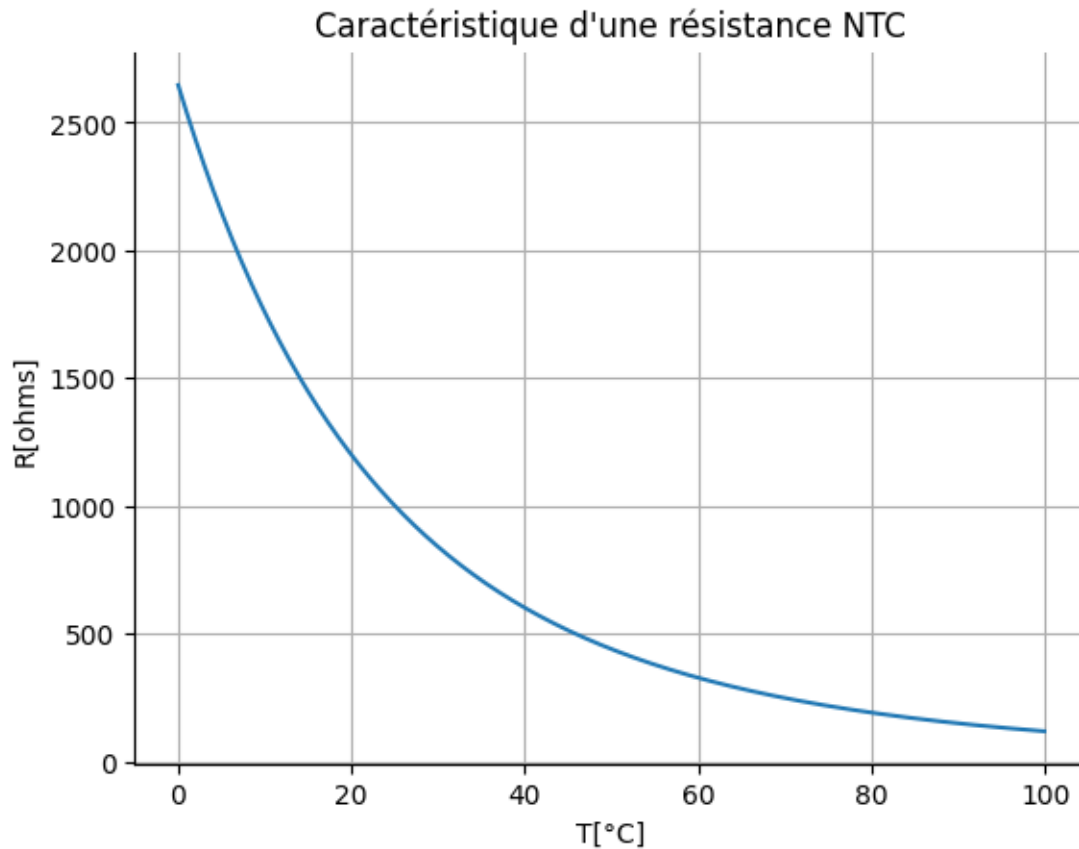
Courbe de la thermorésistance

Pour tracer la courbe de la thermorésistance, on définit la relation et on calcule la valeur de la résistance pour une plage de températures.

```
def Rntc(T, T0, R0, B):
    K0=273.15
    Tk=T+K0
    T0k=T0+K0
    return R0*np.exp(B*(1/Tk-1/T0k))

# Définition d'une plage de température et calcul des valeurs de résistance.
vT=np.arange(0, 100, 0.1)
vRt = Rntc(vT, T0=25, R0=1000, B=3170)

plotIt(vT,vRt, xlabel="T[°C]", ylab="R[ohms]", title="Caractéristique d'une résistance NTC")
plt.savefig("../images/caracteristiqueNTC.png")
```



Linéarisation

On veut linéariser cette caractéristique autour de 20°C en utilisant la *tangente*. On part de la relation qui donne la valeur de la résistance comme fonction de la température.

On utilise *sympy* pour nous aider à résoudre le problème. La fonction qui nous intéresse peut être définie algébriquement comme suit :

```
import sympy as sp
T, T0, R0, B = sp.symbols('T T0 R0 B')
eqR = R0*sp.exp(B*(1/T-1/T0))
```

```
eqR
```

$$R_0 e^{B\left(-\frac{1}{T_0} + \frac{1}{T}\right)}$$

Pour faire une linéarisation, on calcule la pente de la courbe à un point de la courbe. Les valeurs proches peuvent être calculées avec une relation de la forme ci-dessous.

$$y = f(x), \quad y_{lin} = f(x_0) + S \cdot (x - x_0), \quad S = \left. \frac{df(m)}{dm} \right|_{m_0}$$

S est la pente de la fonction à un point x_0 .

Le module *sympy* permet de trouver la dérivée d'une expression assez simplement.

- On obtient la dérivée de la fonction qu'on a défini *eqR* avec l'expression *eqR.diff(T)*
- On peut évaluer la valeur du résultat avec *res.subs(variable, valeur)*. Pour remplacer plusieurs valeurs, il suffit d'itérer (*res.subs().subs()...*)

EX1 : Calculez la fonction linéarisée y_{lin} autour d'un point $T = 20^\circ\text{C}$.

```
dR=eqR.diff(T)
```

EX2 : Tracez la nouvelle fonction sur un graphique avec la caractéristique de la résistance NTC

EX3 : Quelle est l'erreur maximum sur une plage de 20 à 30°C ?

EX4 : Calculez quelle sera l'erreur d'estimation de la température si on utilise la courbe linéarisée inverse au lieu de la caractéristique réelle

EX5 : Comment réduire l'erreur due à la fonction de linéarisation ?