Adja Condé — **22222207 - L3 ST CPES**
Nina Rubin — **22220808 - L3 ST CPES**

Manuel d'utilisation

Conductimètre ADNI

Équipe : Adja Condé et Nina Rubin

François Métivier, Lorette Drique, Corentin Feray, Olivier Lumembe Kibangu

Université Paris Cité - Licence Sciences de la Terre

UE - Mesures automatisées en physique et chimie de l'environnement

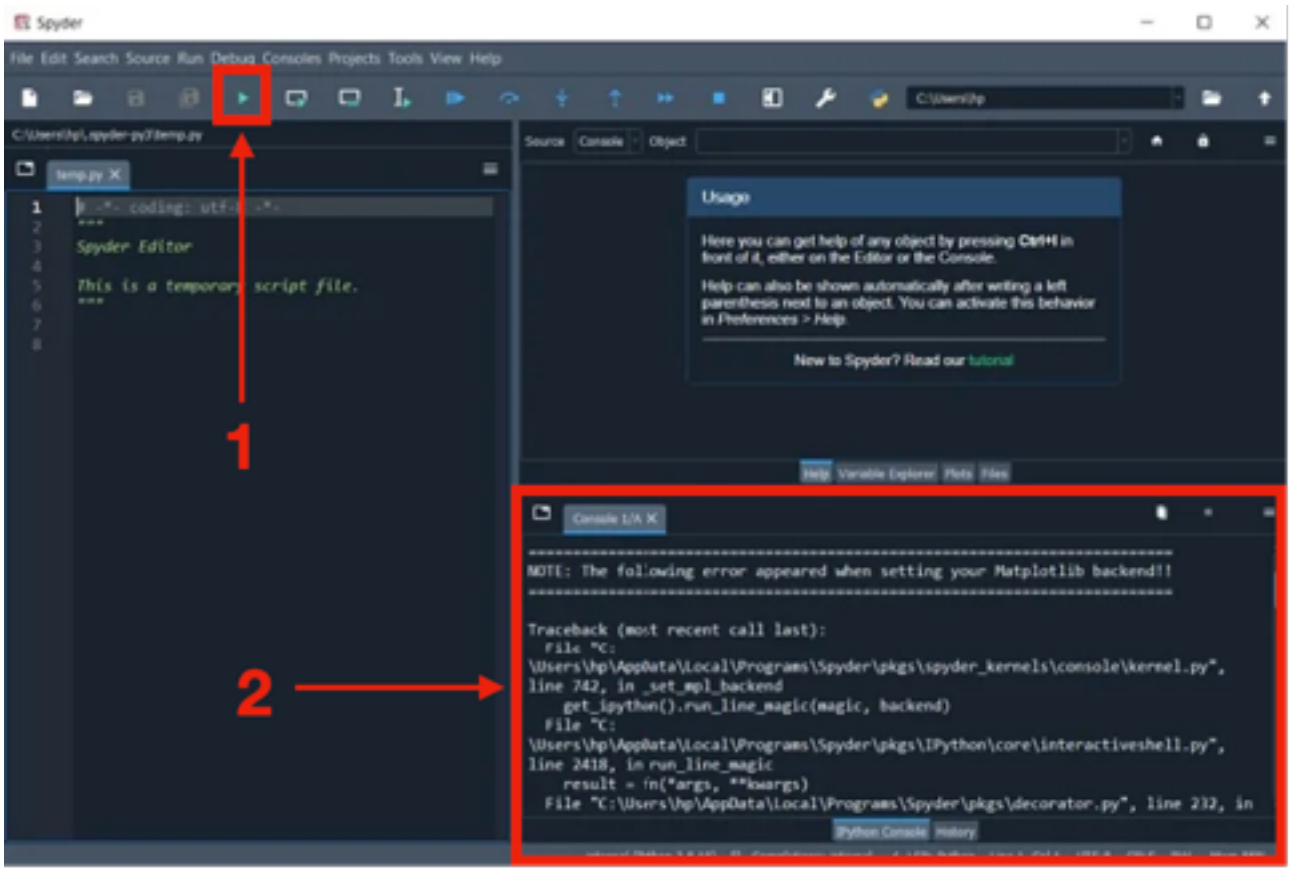
Année universitaire 2024 - 2025

SOMMAIRE

I - Mise en place et modification des valeurs par défaut

Pour commencer à utiliser le conductimètre, branchez-le à votre ordinateur à l'aide d'un câble arduino.

Dans le logiciel Spyder sur l'ordinateur, lancez le fichier 'ADNI_ProgrammePython.py'. Ensuite, appuyez sur le logo ▶ en haut à gauche (indiqué par le 1 ci-dessous), le programme devrait ensuite se lancer dans la console (case en bas à gauche indiquée par le 2 ci-dessous).



Il vous suffit ensuite de suivre les instructions dictées par le programme en rentrant vos choix directement à la suite du programme puis en appuyant sur Entrée.

Le mode d'emploi pour l'étalonnage, les mesures et la gestion des problèmes se trouvent dans la suite de ce manuel.

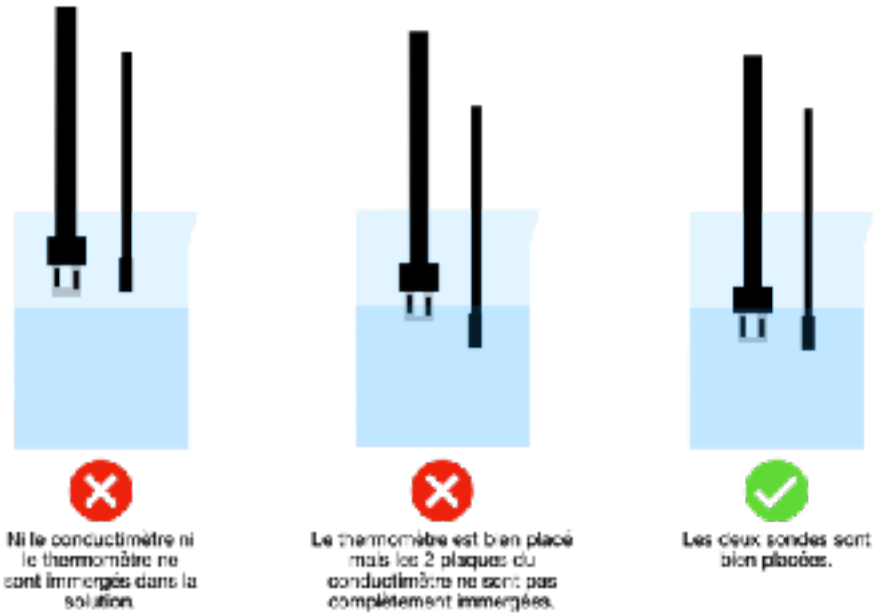
Des valeurs par défaut pour le nombre de valeurs prises en compte lors des calculs de conductivité (par étalon ou par mesure), sont définies, mais vous pouvez les modifier au début en entrant '3' dans l'interface d'accueil. Attention, les valeurs modifiées ne s'enregistrent pas si vous relancez le programme.

II - Étalonnage

Un nouveau calibrage est conseillé à chaque nouvelle utilisation du conductimètre. Cependant, le programme enregistre à chaque fois le dernier étalonnage en date et le réutilise si vous n'en refaites pas.

Les instructions pour l'étalonnage sont les suivantes :

1. Entrez le nombre d'étalon que vous voulez mesurer, il est conseillé d'en faire au moins 3.
2. Indiquez la conductivité de votre premier étalon.
   Remarque : Si cette mesure correspond à 12,8 mS/cm, 1413 µS/cm ou 5000 µS/cm, les conductivités seront corrigées selon la température de la solution. Il vous faudra alors plonger le thermomètre dans la solution.
3. Versez votre étalon dans un bécher de 50mL.
4. Plongez le conductimètre et le thermomètre dans la solution en les immergeant bien comme sur le schéma suivant :



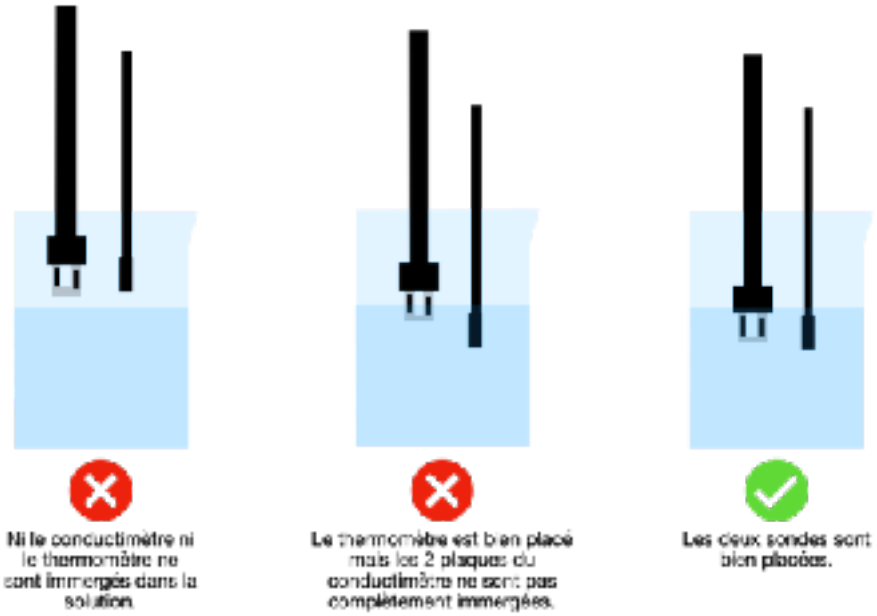
5. Appuyez sur Entrée pour lancer la mesure.
   Cette étape peut prendre du temps.
6. L'écart type ainsi qu'un graphique vous sont affichés, il vous faut indiquer à l'ordinateur si la mesure est assez stable.
   Si la mesure n'est pas stable, elle sera effectuée une nouvelle fois.
7. Une fois que vous avez indiqué la série de mesure comme stable, vous pouvez passer à la solution étalon suivante en répétant les étapes 2 à 6.
8. Une fois tous vos étalons faits, la courbe d'étalonnage devrait s'afficher sur votre ordinateur. Ce calibrage est ainsi enregistré dans votre ordinateur sous le nom 'dernier_étalonnage.csv' en écrasant le dernier fichier du même nom. Si vous voulez le conserver même si vous refaite un étalonnage, il vous suffit de le renommer.

Une fois toutes ces étapes effectuées, vous avez fini le calibrage. Vous pouvez passer aux mesures.

III - Mesures

Vous pouvez à présent mesurer la conductivité de vos différents échantillons. Pour cela vous devez :

1. Saisir le nombre d'échantillons à mesurer.
2. Introduire dans un bécher de 50 mL une quantité suffisante de solution à mesurer afin d’assurer l'immersion totale des deux électrodes du conductimètre, comme sur le schéma suivant :



3. Plonger le conductimètre dans la solution.
4. Appuyer sur Entrée pour lancer la mesure.
5. Attendre que la conductivité moyenne s'affiche sur votre écran. Cette étape peut être assez longue et dépend du nombre de valeurs prise pour calculer la conductivité (modifiable depuis l’accueil).
6. Nettoyer l'électrode à l'eau distillée et la sécher.
7. Passer à l'échantillon suivant.

Vos mesures ont été effectuées avec succès. Vous pouvez réaliser une nouvelle série de mesures ou procéder au rangement du matériel. N'oubliez pas de rincer les électrodes et le thermomètre à l'eau distillée et de tout sécher avant de ranger le matériel.

Annexes

```
import numpy as np
import time
import matplotlib.pyplot as plt
import serial


def accueil():
    """
    Fonction d'accueil du conductimetre, lance le programme complet.

    Returns
    -------
    None.

    """

    #Initialisation des valeurs par défaut
    nbr_mesure_par_echantillon = 100
    nbr_mesure_par_etalon = 200

    try :
        droite = np.loadtxt("./dernier_etalonnage.csv", delimiter = ';')
        print('Le dernier étalonnage a été enregistré, il sera réutilisé par défaut si vous n\'en ref
    except Exception :
        print('Aucun calibrage n\'est enregistré, il vous faut en faire un.')
        nbr_etalon = int(input("Combien d'étalons voulez-vous mesurer ? (au moins 3) : "))
        droite = Etalonnage(nbr_etalon, nbr_mesure_par_etalon)
        np.savetxt('./dernier_etalonnage.csv', droite, delimiter = ';', header = 'Droite étalonnage')
        print('- Vous avez fini le calibrage.')

    while(True) :
        reponse = input('\nQue voulez vous faire ?\n1 : Mesures \n2 : Nouvel étalonnage\n3 : Modifier
        droite = np.loadtxt("./dernier_étalonnage.csv", delimiter = ';')

        if reponse == '1' : # Mesures
            nbr_echantillon = int(input('Combien d\'échantillons voulez-vous mesurer ? : '))
            Mesures(nbr_echantillon, droite, nbr_mesure_par_echantillon)
            print('- Vous avez fini vos mesures.\n')

        elif reponse == '2' : # Calibrage
            nbr_etalon = int(input("Combien d'étalons voulez-vous mesurer ? (au moins 3) : "))
            droite = Etalonnage(nbr_etalon, nbr_mesure_par_etalon)
            np.savetxt('./dernier_étalonnage.csv', droite, delimiter = ';', header = 'Droite étalonna
            print('- Vous avez fini le calibrage.\n')

        elif reponse == '3' : # Modification des valeurs par défaut
            choix_modif = input('\nQuelle valeur voulez-vous modifier ?\n1 : Nombre de valeurs par éc
            if choix_modif == '1' :
                print('Actuellement, votre nombre de valeurs par échantillon est de : ', nbr_mesure_p
                nbr_mesure_par_echantillon = int(input('Combien de mesures voulez-vous effectuer ? :
            elif choix_modif == '2' :
                print('Actuellement, votre nombre de valeurs par étalon est de : ', nbr_mesure_par_et
                nbr_mesure_par_etalon = int(input('Combien de mesures voulez-vous effectuer pour chaq
            else :
                print('Veuillez répondre uniquement 1 ou 2')

        elif reponse == '4' : # Arrêt du programme
            print('  Merci, et bonne journée !')
            break

        else :
            print('Merci de répondre uniquement 1, 2, 3 ou 4\n')

    return
```

```
def setup():
    """
    Parametre d'initialisation de la carte Arduino

    Returns
    -------
    arduino : TYPE
        Localisation de la carte arduino du conductimetre.

    """
    try:
        arduino = serial.Serial(port='/dev/ttyACM0', baudrate = 115200, timeout = 5)
        time.sleep(1)  # Laisser le temps à l'Arduino
        arduino.reset_input_buffer()
        return arduino
    except Exception as e:
        print(f"Erreur de connexion à l'Arduino : {e}")
        return None
```

```
def Mesures(nbr_echantillon, droite, nbr_mesure_par_echantillon):
    """
    Fonction pour faire les mesures de conductivité et de température et les stocker dans un fichier.

    Parameters
    ----------
    nbr_echantillon : int
        Nombre d'échantillons à mesurer dans la série.
    droite : numpy.poly1d
        Equation de droite du calibrage permettant de lier la tension mesurer par le conductimètre et
    nbr_mesure_par_echantillon : int
        Nombre de mesure sur lesquelles les conductivité et température moyennes seront calculées.

    Returns
    -------
    None.

    """
    conductimeter = setup()
    list_temp = []
    list_conductivite = []
    numerotation = []
    donnees = []

    for k in range(nbr_echantillon):
        print('\n[ Échantillon',k+1,']')
        input('Veuillez preparer votre échantillon, puis appuyer sur Entrée pour lancer la mesure.')
        print('Vos mesures sont en cours, patientez s\'il vous plait.']
        conductimeter.flushInput()
        for i in range(nbr_mesure_par_echantillon):
            data = conductimeter.readline().decode().strip('\r\n').split(',')
            list_temp.append(float(data[0]))
            list_conductivite.append(droite[0]*(float(data[1])) + droite[1])
            numerotation.append(k*0.01)
            time.sleep(0.0)
        print('-> Resultat : La température moyenne est : ', np.mean(list_temp), '°C, et la conductiv
        donnees.append([k, np.mean(list_temp), np.mean(list_conductivite)])
    np.savetxt('./data_conductivité.csv', donnees, delimiter = ';',fmt = '%.2f', header="Mesure n°,Te
    print("Vos mesures sont désormais stockées dans le fichier data_conductivité.csv.\nAttention, si
    return
```

```
def graph_conductimeter():
    """
    Fonction qui permet de tracer à partir du fichier data_conductivité.csv la température et la conc

    Returns
    -------
    None.

    """
    data = np.loadtxt("./data_conductivité.csv", delimiter = ';')
    plt.figure()
    plt.plot(data[0], data[1], 'o', color = 'r')
    plt.xlabel('Temps (s)')
    plt.ylabel('Température (°C)')
    plt.axis([0, np.max(data[0]), 0, np.max(data[1])+5]) #définition du domaine des axes
    plt.figure()
    plt.plot(data[0], data[2], 'o', color = 'b')
    plt.xlabel('Temps (s)')
    plt.ylabel('Conductivité (uS/cm)')
    plt.axis([0, np.max(data[0]), 0, np.max(data[2])+5]) #définition du domaine des axes

    return
```

```
def mesure_etalonnage(nbr_mesure_par_etalon):   # confirmation que l'étalonnage est bon en faisant pe
    """
    Fonction permettant de faire plein de mesure à haute fréquance puis les mettant dans un graphique

    Parameters
    ----------
    nbr_mesure_par_etalon : int
        nombre de mesures par étalon, décidé dans les valeurs par défaut.

    Returns
    -------
    list_tension_etalon : list
        Liste des tensions mesurées pendant l'étalonnage.

    """
    conductimeter = setup()
    list_tension_etalon = []
    numerotation = []
    print('Les mesures sont en cours, attendez s\'il vous plait.')
    conductimeter.flushinput()
    for k in range(nbr_mesure_par_etalon):
        lect = conductimeter.readline().decode().strip('\r\n').split(',')
        list_tension_etalon.append(float(lect[1]))
        numerotation.append(k*0.01)
        time.sleep(0.01)
    plt.figure()
    plt.plot(numerotation, list_tension_etalon, 'o')
    plt.xlabel('Temps (s)')
    plt.ylabel('Tension (V)')
    plt.show()

    return list_tension_etalon
```

```
def Etalonnage(nbr_etalon, nbr_mesure_par_etalon):
    """
    Fonction qui, pour le nombre d'étalon indiqué en argument, mesure la tension, trouve la corrélati

    Parameters
    ----------
    nbr_etalon : int
        Nombre d'étalon pour l'étalonnage.
    nbr_mesure_par_etalon : int
        Nombre de mesure à faire pour chaque étalon.

    Returns
    -------
    droite : numpy.poly1d
        Droite d'étalonnage sous la forme ax + b.

    """
    conductimeter = setup()
    list_tension = []
    list_conductivite=[]

    #Correction de la conductivité par la température
    Temperature = [0,5,10,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]
    Conductivite_12880 = [7150,8220,9330,10480,10720,10950,11190,11430,11670,11910,12150,12390,12640,
    Conductivite_1413 = [776,896,1020,1147,1173,1195,1225,1251,1278,1305,1332,1359,1386,1413,1440,145
    Conductivite_5000 = [2760,3180,3515,4063,4155,4245,4337,4429,4523,4617,4711,4805,4902,5000,5096,5

    reg_12880 = np.polyfit(Temperature,Conductivite_12880,1)
    droite_12880 = np.poly1d(reg_12880)

    reg_1413= np.polyfit(Temperature,Conductivite_1413,1)
    droite_1413 = np.poly1d(reg_1413)

    reg_5000= np.polyfit(Temperature,Conductivite_5000,1)
    droite_5000 = np.poly1d(reg_5000)

    for k in range(nbr_etalon):
        print('\n[ Étalon', k+1,']')

        conductivite = float(input('Quelle est la valeur de conductivité de votre étalon à 25°C ? (en
        if conductivite == 12880 :
            print('Remarque : Votre mesure sera corrigée avec la température mesurée dans l\'échantil
            input('Appuyez sur Entrée après avoir plongé le thermomètre dans la solution.')
            lect = conductimeter.readline().decode().strip('\r\n').split(",")
            T = float(lect[0])
            conductivite = droite_12880(T)
        elif conductivite == 5000 :
            print('Remarque : Votre mesure sera corrigée avec la température mesurée dans l\'échantil
            input('Appuyez sur Entrée après avoir plongé le thermomètre dans la solution.')
            lect = conductimeter.readline().decode().strip('\r\n').split(",")
            T = float(lect[0])
            conductivite = droite_5000(T)
        elif conductivite == 1413 :
            print('Remarque : Votre mesure sera corrigée avec la température mesurée dans l\'échantil
            input('Appuyez sur Entrée après avoir plongé le thermomètre dans la solution.')
            lect = conductimeter.readline().decode().strip('\r\n').split(",")
            T = float(lect[0])
            conductivite = droite_1413(T)
        else :
            print('Remarque : Votre mesure ne pourra pas être corrigée selon la température ')

        input('\nAppuyez sur Entrée pour lancer la mesure.')
        list_conductivite.append(conductivite)
```

```
    else :
        print('Remarque : Votre mesure ne pourra pas être corrigée selon la température.')

    input('\nAppuyez sur Entrée pour lancer la mesure.')
    list_conductivite.append(conductivite)
    list_tension_etalonnage = mesure_etalonnage(nbr_mesure_par_etalon)
    ecart_type = np.std(list_tension_etalonnage)
    print('L\'écart-type des mesures de l\'étalon est :', ecart_type,'V, un graphique s\'est aussi
    a = input('\nEst ce que la mesure est stable ? (y/n) : ')
    moy_etalon = stabilite_mesure(a, list_tension_etalonnage, nbr_mesure_par_etalon)
    print('La conductivité de votre étalon est de', conductivite, ' uS/cm avec une tension enregi
    list_tension.append(float(moy_etalon))
    print('Cette mesure est enregistrée, passez à la suite.\n')

plt.figure()
plt.plot(list_tension,list_conductivite, 'o', color = 'r')
plt.xlabel('Tension (V)')
plt.ylabel('Conductivité (uS/cm)')
plt.axis([0, np.max(list_tension), 0, np.max(list_conductivite) +5])
plt.title('Courbe d\'étalonnage')
reg = np.polyfit(list_tension, list_conductivite, 1)
droite = np.poly1d(reg)
plt.plot(list_tension, droite[1] * np.array(list_tension) + droite[0])
plt.show()

return droite
```

```
def stabilite_mesure(a, nbr_mesure_par_etalon):
    while a == 'n' :
        list_tension_etalonnage = mesure_etalonnage(nbr_mesure_par_etalon)
        ecart_type = np.std(list_tension_etalonnage)
        print('L\'écart-type des mesures de l\'étalon est :', ecart_type,'V, un graphique s\'est aus
        a = input('Est ce que la mesure est stable ? (y/n) : ')
    if a =='y' :
        moy_etalon = np.mean(list_tension_etalonnage[int(nbr_mesure_par_etalon *0.25):])
    else:
        print('Merci de ne répondre que \'y\' pour yes et \'n\' pour no. )
        stabilite_mesure(a, list_tension_etalonnage, nbr_mesure_par_etalon)
    return moy_etalon




if __name__ == '__main__':

    accueil()
```