

ExoNote2_2024_gr1B_sujet

March 17, 2024

1 Exercice noté

1.1 groupe 1B– mercredi 20 mars

1.2 NOM :

1.3 Prénom :

On a fait un certain nombre de mesures successivement toujours dans les mêmes conditions. On a 15 valeurs comprises entre 0.1 et 1 dans un tableau nommé xref. Les différentes mesures sont effectuées pour ces valeurs là.

Ces mesures sont mises dans un second tableau appelé mesures qui se compose de 10 séries de 15 valeurs qui correspondent donc à 10 séries de mesures correspondants aux valeurs de xref.

```
[1]: import numpy as np
import matplotlib.pyplot as plt
```

```
[2]: xref=np.array([0.1          , 0.16428571, 0.22857143, 0.29285714, 0.35714286,
0.42142857, 0.48571429, 0.55          , 0.61428571, 0.67857143,
0.74285714, 0.80714286, 0.87142857, 0.93571429, 1.          ])

mesures=np.array([[ 0.75479356,  0.62472619,  0.450314   ,  0.21230411, -0.
↪10758782,
-0.51130372, -0.97283841, -1.42144237, -1.73776484, -1.8023336 ,
-1.5225087 , -0.96058387, -0.36467317, -0.10859138, -0.47368007],
[ 0.75409624,  0.62519105,  0.45047013,  0.21203858, -0.10764305,
-0.51138594, -0.97250361, -1.42112839, -1.73824807, -1.8021314 ,
-1.52118442, -0.96145969, -0.36452806, -0.10842284, -0.47355031],
[ 0.7540591 ,  0.62494635,  0.45078532,  0.21216349, -0.10760606,
-0.51158744, -0.97247599, -1.41988751, -1.73892838, -1.79877176,
-1.52342637, -0.96153561, -0.36434846, -0.10862795, -0.47287967],
[ 0.75424915,  0.62426608,  0.45077518,  0.21226789, -0.10773574,
-0.51147247, -0.97356636, -1.42038341, -1.73881914, -1.79894048,
-1.5221951 , -0.96132793, -0.36397861, -0.10850696, -0.47293642],
[ 0.75460585,  0.6242309 ,  0.45040047,  0.2122364 , -0.1076606 ,
-0.51182517, -0.97278277, -1.42000311, -1.74106031, -1.79918139,
-1.52314976, -0.96105537, -0.36397926, -0.10843636, -0.47349404],
[ 0.75517932,  0.6252301 ,  0.45021131,  0.21198192, -0.10769042,
-0.51154723, -0.97301572, -1.42132521, -1.73928168, -1.80019725,
```

```

-1.52240838, -0.96147968, -0.36396327, -0.1085195 , -0.47367581],
[ 0.75463506, 0.62481934, 0.45037456, 0.21203034, -0.10770719,
-0.51190454, -0.97343855, -1.42127494, -1.73975395, -1.80019924,
-1.52142373, -0.96099088, -0.36448489, -0.10857041, -0.47311828],
[ 0.75411707, 0.62526902, 0.45023469, 0.21223322, -0.10773661,
-0.51131978, -0.97330938, -1.41942442, -1.73893917, -1.80081052,
-1.52315007, -0.96078771, -0.36396075, -0.10851716, -0.47344307],
[ 0.75460532, 0.62518605, 0.45076301, 0.21220984, -0.10769284,
-0.51146597, -0.97332902, -1.42023359, -1.7391413 , -1.79989116,
-1.52309739, -0.96020262, -0.36419854, -0.10845816, -0.4729474 ],
[ 0.75521003, 0.62528451, 0.45061085, 0.21232567, -0.10771718,
-0.51154436, -0.97317738, -1.41969239, -1.73883741, -1.80162995,
-1.52330017, -0.96039282, -0.36425818, -0.10859955, -0.47356825]])

```

1. Calculer le tableau tm qui contient les 15 valeurs moyennes des 10 séries (sans boucle).

[3]: # à compléter

```

tableau des moyennes [ 0.75455507 0.62491496 0.45049395 0.21217915
-0.10767775 -0.51153566
-0.97304372 -1.42047953 -1.73907743 -1.80040867 -1.52258441 -0.96098162
-0.36423732 -0.10852503 -0.47332933]

```

2. Calculer le tableau des écarts entre les mesures et les moyennes par colonnes.

En déduire combien de valeurs du tableau mesures s'éloignent de plus de 0.001 de la moyenne

[4]: # à compléter

```

tableau des écarts [[ 2.384900e-04 -1.887690e-04 -1.799520e-04 1.249640e-04
8.993100e-05
2.319420e-04 2.053090e-04 -9.628360e-04 1.312585e-03 -1.924925e-03
7.570900e-05 3.977480e-04 -4.358510e-04 -6.635300e-05 -3.507380e-04]
[-4.588300e-04 2.760910e-04 -2.382200e-05 -1.405660e-04 3.470100e-05
1.497220e-04 5.401090e-04 -6.488560e-04 8.293550e-04 -1.722725e-03
1.399989e-03 -4.780720e-04 -2.907410e-04 1.021870e-04 -2.209780e-04]
[-4.959700e-04 3.139100e-05 2.913680e-04 -1.565600e-05 7.169100e-05
-5.177800e-05 5.677290e-04 5.920240e-04 1.490450e-04 1.636915e-03
-8.419610e-04 -5.539920e-04 -1.111410e-04 -1.029230e-04 4.496620e-04]
[-3.059200e-04 -6.488790e-04 2.812280e-04 8.874400e-05 -5.798900e-05
6.319200e-05 -5.226410e-04 9.612400e-05 2.582850e-04 1.468195e-03
3.893090e-04 -3.463120e-04 2.587090e-04 1.806700e-05 3.929120e-04]
[ 5.078000e-05 -6.840590e-04 -9.348200e-05 5.725400e-05 1.715100e-05
-2.895080e-04 2.609490e-04 4.764240e-04 -1.982885e-03 1.227285e-03
-5.653510e-04 -7.375200e-05 2.580590e-04 8.866700e-05 -1.647080e-04]
[ 6.242500e-04 3.151410e-04 -2.826420e-04 -1.972260e-04 -1.266900e-05
-1.156800e-05 2.799900e-05 -8.456760e-04 -2.042550e-04 2.114250e-04
1.760290e-04 -4.980620e-04 2.740490e-04 5.527000e-06 -3.464780e-04]
[ 7.999000e-05 -9.561900e-05 -1.193920e-04 -1.488060e-04 -2.943900e-05

```

```

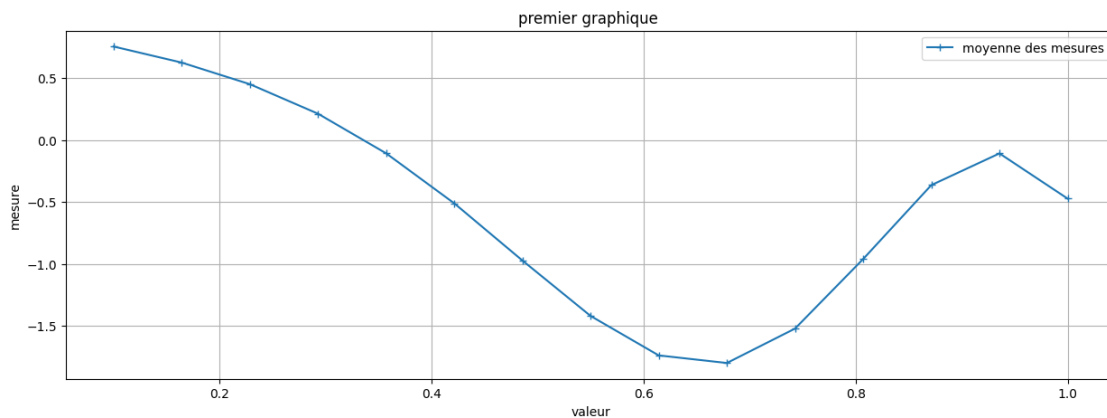
-3.688780e-04 -3.948310e-04 -7.954060e-04 -6.765250e-04  2.094350e-04
 1.160679e-03 -9.262000e-06 -2.475710e-04 -4.538300e-05  2.110520e-04]
[-4.380000e-04  3.540610e-04 -2.592620e-04  5.407400e-05 -5.885900e-05
 2.158820e-04 -2.656610e-04  1.055114e-03  1.382550e-04 -4.018450e-04
-5.656610e-04  1.939080e-04  2.765690e-04  7.867000e-06 -1.137380e-04]
[ 5.025000e-05  2.710910e-04  2.690580e-04  3.069400e-05 -1.508900e-05
 6.969200e-05 -2.853010e-04  2.459440e-04 -6.387500e-05  5.175150e-04
-5.129810e-04  7.789980e-04  3.877900e-05  6.686700e-05  3.819320e-04]
[ 6.549600e-04  3.695510e-04  1.168980e-04  1.465240e-04 -3.942900e-05
-8.698000e-06 -1.336610e-04  7.871440e-04  2.400150e-04 -1.221275e-03
-7.157610e-04  5.887980e-04 -2.086100e-05 -7.452300e-05 -2.389180e-04]]
il y a 11 valeurs qui s'écartent de plus de 0.001 de la moyenne

```

3. Faire un graphique comme sur la figure ci-dessous: ce schéma doit comporter les points correspondants à xref et au tableau des moyennes

```
[5]: # à compléter
```

```
[5]: <matplotlib.legend.Legend at 0x7f4c9415f760>
```



4. On cherche à avoir une fonction dont la représentation graphique doit se rapprocher du dessin des moyennes.

On hésite entre deux fonctions:

Soit f_1 de la forme $a * \sin(bx + c)$

Soit f_2 de la forme $a_1 * \cos(b_1 x^2 + c_1)$

Utilisez la fonction `scipy.optimize.curve_fit` pour déterminer les paramètres de ces deux fonctions.

```
[6]: import scipy.optimize
```

```
# à compléter
```

```
pour f1, a= -1.423492677710676 b= -4.457024389614511 c= 4.4769288625723505
pour f2, a1= 1.5873059303872035 b1= 3.7893092072392687 c1= 1.189473841951293
```

5. On veut évaluer ces deux fonctions trouvées. Calculer et afficher les résidus (écart entre modèle et données) au moyen de la formule $r = \frac{\sqrt{\sum (f(t) - t_i)^2}}{N}$.

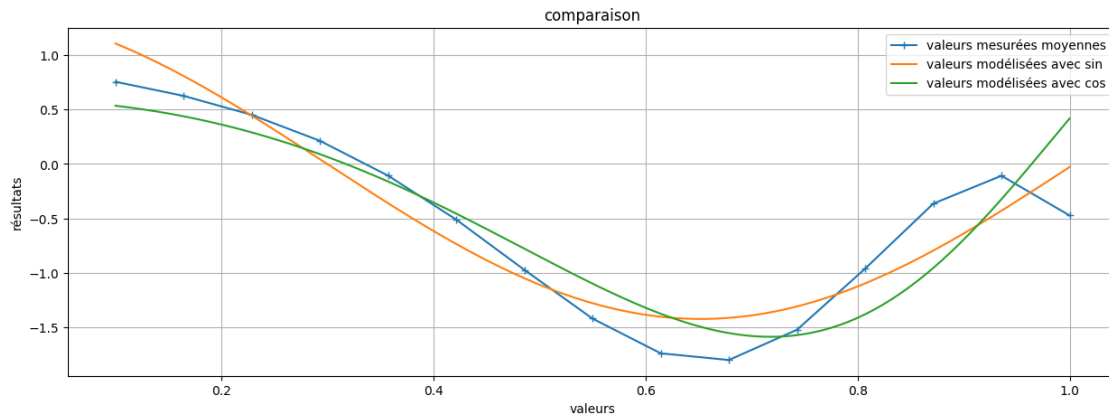
```
[7]: # à compléter
```

```
résidu moyen avec les moyennes pour f1: 0.07116039128290448
résidu moyen avec les moyennes pour f2: 0.09023527675910722
```

6. Tracer sur la même courbe la moyenne des données relevées et les courbes obtenues pour f_1 et f_2 avec la fonction d'estimation ci-dessus.

```
[8]: # à compléter
```

```
[8]: <matplotlib.legend.Legend at 0x7f4c7aa27fd0>
```

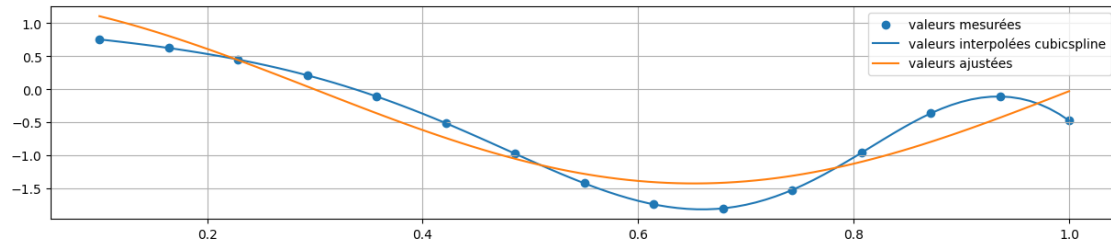


7. On cherche à interpoler entre les points pour estimer les valeurs entre ces points.
- Interpoler au moyen de `scipy.interpolate.CubicSpline`
 - Tracer la courbe interpolée et les points sur le même graphique.
 - Tracer également la courbe obtenue pour f_1 avec la méthode d'ajustement de la première partie.

```
[9]: import scipy.interpolate
```

```
# à compléter
```

```
[10]: # à compléter
```



Grace à ces estimations, déterminer des valeurs approchées de la valeur qui devrait être obtenue pour $x=0.5$

[11]: *# à compléter*

avec f1 -1.1089959261397957

avec f2 -0.851215348930826

avec le cubicSpline -1.0772174629492048