

Pour aller plus loin

numériser · échantillonner · spectre · filtrer · diffusion · hasard · Student

NUMÉRISER · BUT MESURES PHYSIQUES

pas d'un CAN	<code>p = U_max / (2 ** n - 1)</code>
entier lu	<code>N = np.round(u / p)</code>
tension restituée	<code>N * p</code> ; erreur $\leq p / 2$
plus grand écart	<code>np.max(np.abs(a - b))</code>

ÉCHANTILLONNER · PC, MP, PSI

instants des mesures	<code>np.arange(0, duree, 1 / fe)</code>
Nyquist-Shannon	$f_e > 2 \times f_{\max}$
repliement	f entre $f_e/2$ et f_e vue à $f_e - f$

SPECTRE · PSI

transformée	<code>X = np.fft.rfft(x)</code>
amplitudes	<code>np.abs(X) * 2 / len(x)</code>
fréquences	<code>np.fft.rfftfreq(N, 1 / fe)</code>
résolution	$f_e / N = 1 / \text{durée}$
pic le plus haut	<code>freq[np.argmax(A)]</code>

FILTRE · MP, TSI2

moyenne glissante	<code>np.mean(T[i - k : i + k + 1])</code>
ajouter à une liste	<code>liste.append(v)</code>
passe-bas (Euler)	<code>y[i] = y[i-1] + Te/tau * (x[i-1] - y[i-1])</code>

TABLEAUX 2D · PC, MP, TSI2

créer	<code>T = np.zeros((nl, nc))</code> · <code>T.shape</code>
un élément	<code>T[j, i]</code> : ligne j , colonne i
ligne, colonne	<code>T[j, :]</code> · <code>T[:, i]</code>
sans les bords	<code>T[j, 1:-1]</code> ; dernier : <code>-1</code>
stabilité	$r = D \Delta t / \Delta x^2 \leq 0,5$

HASARD ET STATISTIQUES · PC, BUT CHIMIE

pas au hasard	<code>np.random.choice([-1, 1], size=(n, p))</code>
somme des lignes	<code>np.sum(M, axis=1)</code>
marche aléatoire	écart-type $\propto \sqrt{t}$
Student, 95 %	<code>stats.t.ppf(0.975, n - 1)</code>
LD · LQ	$3 s_{\text{blanc}} / k \cdot 10 s_{\text{blanc}} / k$

EXEMPLE COMMENTÉ 1 · LE SPECTRE D'UNE NOTE

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

fe = 8000 # fréquence d'échantillonnage, en Hz : fe > 2 * fmax
t = np.arange(0, 1.0, 1 / fe) # 1 s de mesures : spectre précis au hertz
x = np.sin(2 * np.pi * 220 * t) + 0.5 * np.sin(2 * np.pi * 440 * t) # fondamentale + rang 2

N = len(x)
A = np.abs(np.fft.rfft(x)) * 2 / N # amplitudes : module, puis * 2 / N
freq = np.fft.rfftfreq(N, 1 / fe) # 2e paramètre : Te = 1 / fe, pas fe

print('fondamentale :', freq[np.argmax(A)], 'Hz') # fréquence du pic le plus haut
print('pics :', freq[A > 0.1], 'Hz') # fréquences où A dépasse 0,1
plt.plot(freq, A)
plt.xlabel('f (Hz)')
plt.ylabel('amplitude')
plt.show()
```

```
fondamentale : 220.0 Hz
pics : [220. 440.] Hz
```

EXEMPLE COMMENTÉ 2 · LA DIFFUSION THERMIQUE DANS UN MUR

```
import numpy as np

L, nx, D = 0.20, 41, 8e-7          # épaisseur (m), nombre de points, diffusivité (m²/s)
dx = L / (nx - 1)                  # pas d'espace : 5 mm
dt = 12.5                           # pas de temps, en s
r = D * dt / dx ** 2                # doit rester ≤ 0,5, sinon le calcul explose
print('r =', round(r, 2))
nt = int(12 * 3600 / dt)            # nombre de pas pour 12 h

T = np.zeros((nt + 1, nx))         # une LIGNE par instant, une COLONNE par point du mur
T[0, :] = 20                       # condition initiale : tout le mur à 20 °C
T[:, 0] = 0                         # face extérieure à 0 °C, à tout instant
T[:, -1] = 20                      # face intérieure à 20 °C, à tout instant

for j in range(nt):                 # chaque point intérieur se rapproche de la moyenne de ses voisins
    T[j + 1, 1:-1] = T[j, 1:-1] + r * (T[j, 2:] - 2 * T[j, 1:-1] + T[j, :-2])
    #                               le point      voisins de droite    lui-même      voisins de gauche

print('milieu après 1 h :', round(T[288, 20], 1), '°C')    # 1 h = 288 pas ; colonne 20 = milieu
print('milieu après 12 h :', round(T[-1, 20], 1), '°C')    # dernière ligne : régime permanent
```

```
r = 0.4
milieu après 1 h : 16.2 °C
milieu après 12 h : 10.0 °C
```

En régime permanent, le profil est une droite de 0 à 20 °C : le milieu est à 10 °C.

EXEMPLE COMMENTÉ 3 · STUDENT ET LIMITES D'UN DOSAGE

```
import numpy as np
from scipy import stats              # scipy : téléchargé à la première utilisation

c = np.array([0.0, 1.0, 2.0, 3.0, 4.0, 5.0])    # gamme étalon, en mg/L
A = np.array([0.001, 0.126, 0.247, 0.373, 0.495, 0.621])
k, b = np.polyfit(c, A, 1)                # pente k, ordonnée à l'origine b

blanc = np.array([0.002, -0.001, 0.003, 0.000, 0.001,
                  -0.002, 0.002, 0.001, -0.001, 0.000])    # 10 mesures du blanc
s_blanc = np.std(blanc, ddof=1)            # écart-type expérimental : ddof=1
LD = 3 * s_blanc / k                      # limite de détection : sous LD, rien n'est affirmé
LQ = 10 * s_blanc / k                    # limite de quantification : au-dessus, résultat fiable
print('LD =', round(LD, 3), 'mg/L   LQ =', round(LQ, 3), 'mg/L')

mes = np.array([0.471, 0.468, 0.475, 0.466, 0.472])    # 5 mesures seulement
n = len(mes)
u = np.std(mes, ddof=1) / np.sqrt(n)          # incertitude-type de la moyenne (type A)
t95 = stats.t.ppf(0.975, n - 1)              # Student : 95 %, n - 1 degrés de liberté → 2,78
print('A =', round(np.mean(mes), 4), '±', round(t95 * u, 4))
```

```
LD = 0.038 mg/L   LQ = 0.128 mg/L
A = 0.4704 ± 0.0044
```

Avec $2u$, on aurait écrit $\pm 0,0031$: avec cinq mesures, Student élargit l'intervalle.

LES PIÈGES

pas d'un CAN	<code>2 ** n - 1</code>	2^n valeurs, mais $2^n - 1$ intervalles entre 0 et $U_{\max}$.
rfftfreq	<code>1 / fe</code>	Le second paramètre est la période d'échantillonnage T_e , pas f_e .
amplitudes	<code>* 2 / N</code>	Sans ce facteur, un pic d'amplitude 0,8 s'affiche à 3 200.
pic intrus	$f_e - f$	Une composante au-delà de $f_e / 2$ se replie : aucun message, un spectre faux.
tranche	<code>T[a : b]</code>	S'arrête AVANT <code>b</code> : pour aller jusqu'à <code>i + k</code> , écrire <code>i + k + 1</code> .
tableau 2D	<code>T[j, i]</code>	La ligne d'abord, la colonne ensuite. <code>shape</code> donne (lignes, colonnes).
somme	<code>axis=1</code>	Somme de chaque ligne ; <code>axis=0</code> somme chaque colonne.
stabilité	$r \leq 0,5$	Au-delà, le schéma explicite explose (10^{49} °C), sans message d'erreur.
Student	<code>0.975, n - 1</code>	2,5 % de chaque côté pour 95 % ; $n - 1$ degrés de liberté.
LD, LQ	<code>/ k</code>	Une absorbance divisée par la pente donne une concentration.

D'OÙ VIENNENT CES NOTIONS

numériser, CAN	BUT Mesures physiques (R2.08)
échantillonnage, repliement	CPGE PC, MP, PSI (2e année) ; BUT Mesures physiques
transformée de Fourier discrète	CPGE PSI ; en PC, « non exigible »
filtrage numérique	CPGE MP, TSI2
différences finies, tableaux 2D	CPGE PC, MP, TSI2
marche aléatoire	CPGE PC
Student, limites de détection	BUT Chimie (R3.08)

CE QUE JE SAIS FAIRE

- ☐ Calculer le pas d'un CAN et l'erreur de numérisation.
- ☐ Choisir f_e avec Nyquist-Shannon ; reconnaître un repliement.
- ☐ Calculer et lire le spectre d'un son avec `rfft`.
- ☐ Lisser par moyenne glissante ; filtrer par récurrence.
- ☐ Résoudre la diffusion par différences finies, en restant stable.
- ☐ Simuler une marche aléatoire : étalement en $\sqrt{t}$.
- ☐ Écrire un intervalle de confiance avec Student.
- ☐ Calculer les limites de détection et de quantification.

Point à revoir :

.....

.....