

Concours blanc – informatique MP

mercredi 6 novembre 2024, durée 2 heures



Les candidats ne doivent faire usage d'aucun document : l'utilisation de toute calculatrice et de tout matériel électronique est interdite. L'épreuve est constituée d'un exercice et d'un problème.

Exercice

L est une liste d'entiers.

1. Écrire une fonction qui renvoie **True** si toutes les valeurs de L sont comprises entre 1 et 20, et **False** si ce n'est pas le cas. On s'interdit d'utiliser `min` et `max`.
2. Écrire une fonction qui renvoie le nombre de valeurs de L qui sont inférieures ou égales à 50.

3. Étant donné un tableau matriciel carré (liste de listes), écrire une fonction **Marges** qui renvoie ses sommes marginales par ligne et par colonne.

Par exemple, pour $M = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]$, visualisée matriciellement

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, M \text{ a pour marges}$$

$$L = [(1 + 4 + 7) \quad (2 + 5 + 8) \quad (3 + 6 + 9)] \text{ et } C = [(1 + 2 + 3) \quad (4 + 5 + 6) \quad (7 + 8 + 9)].$$

4. Écrire une fonction **Sont_egales** qui prend en paramètre un tableau matriciel carré et qui renvoie **True** si ses deux sommes marginales sont égales et **False** sinon.
-

Problème : Mesures de Houle

Le sujet comporte des questions de programmation. Le langage à utiliser est Python. **Il est conseillé de feuilleter le document avant de débiter.**

On s'intéresse à des mesures de niveau de la surface libre de la mer effectuées par une bouée (représentée sur la figure 1)¹. Cette bouée contient un ensemble de capteurs incluant un accéléromètre vertical qui fournit, après un traitement approprié, des mesures à étudier².

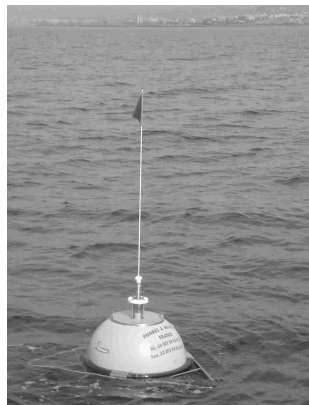


FIGURE 1 – Bouée de mesure de houle.

Partie I. Stockage interne des données

Une campagne de mesures a été effectuée. Les caractéristiques de cette campagne sont les suivantes :

- durée de la campagne : 15 jours ;
- durée d'enregistrement : 20 min toutes les demi-heures ;
- fréquence d'échantillonnage : 2Hz.

Les relevés de la campagne de mesure sont écrits dans un fichier texte dont le contenu est défini comme suit :

- Les informations relatives à la campagne sont rassemblées sur la première ligne du fichier, séparées par des points-virgules (";"). On y indique différentes informations importantes comme le numéro de la campagne, le nom du site, le type du capteur, la latitude et la longitude de la bouée, la date et l'heure de la séquence.
- Les lignes suivantes contiennent les mesures du déplacement vertical (en mètre). Chaque ligne comporte 8 caractères (dont le caractère de fin de ligne). Par exemple, on trouvera dans le fichier texte les trois lignes suivantes :

```
+0.4256
+0.3174
-0.0825
...
```

1. Cette étude utilise des résultats extraits de la base de données du Centre d'Archivage National des Données de Houle In Situ. Les acquisitions ont été effectuées par le Centre d'Etudes Techniques Maritimes et Fluviales.

2. L'ensemble des paramètres des états de mer présent dans la base CANDHIS est calculé par les logiciels :

- Houle5 (CETMEF) : analyse vague par vague (temporelle) ;
- PADINES (EDF/LNHE) : analyse spectrale et directionnelle (fréquentielle).

1. On suppose que chaque caractère est codé sur 8 bits. En ne tenant pas compte de la première ligne, déterminer le nombre d'octets correspondant à 20 minutes d'enregistrement à la fréquence d'échantillonnage de 2Hz.
2. En déduire le nombre approximatif (un ordre de grandeur suffira) d'octets contenus dans le fichier correspondant à la campagne de mesures définie précédemment. Une carte mémoire de 1Go est-elle suffisante ?
3. Si, dans un souci de réduction de la taille du fichier, on souhaitait ôter un chiffre significatif dans les mesures, quel gain relatif d'espace mémoire obtiendrait-on ?
4. Les données se trouvent dans le répertoire de travail sous forme d'un fichier `donnees.txt`. Proposer une suite d'instructions permettant de créer à partir de ce fichier une liste de flottants `liste_niveaux` contenant les valeurs du niveau de la mer. On prendra garde à ne pas insérer dans la liste la première ligne du fichier.

Deux analyses sont effectuées sur les mesures : l'une est appelée « vague par vague », l'autre est appelée « spectrale ».

Partie II. Analyse « vague par vague »

On considère ici que la mesure de houle est représentée par un signal $\eta(t) \in \mathbb{R}$, $t \in [0, T]$, avec η une fonction \mathcal{C}^1 .

On appelle niveau moyen m la moyenne de $\eta(t)$ sur $[0, T]$.

On définit Z_1, Z_2, \dots, Z_n l'ensemble (supposé fini) des Passages par le Niveau moyen en Descente (PND, voir figure 2). À chaque PND, le signal traverse la valeur m en descente.

On suppose $\eta(0) > m$ et $\frac{d\eta}{dt}(0) > 0$. On en déduit que $\eta(t) - m \geq 0$ sur $[0, Z_1]$.

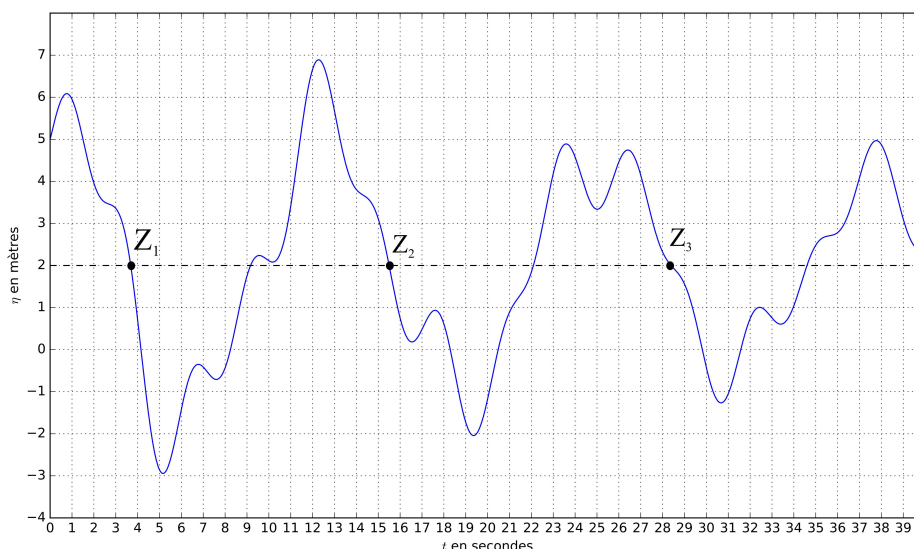


FIGURE 2 – Passage par le Niveau moyen en Descente (PND). Ici la moyenne m vaut 2.

Les hauteurs des vagues H_i sont définies par les différences

$$\begin{cases} H_1 = \max_{t \in [0, Z_1]} \eta(t) - \min_{t \in [Z_1, Z_2]} \eta(t) \\ H_i = \max_{t \in [Z_{i-1}, Z_i]} \eta(t) - \min_{t \in [Z_i, Z_{i+1}]} \eta(t) \quad \text{pour } 2 \leq i < n \end{cases}$$

On définit les *périodes de vagues* par $T_i = Z_{i+1} - Z_i$.

5. Pour le signal représenté sur la figure 2, que valent approximativement H_1 , H_2 et H_3 ? Que valent approximativement T_1 et T_2 ?

On adopte désormais une représentation en temps discret du signal. On appelle *horodate* un ensemble (fini) des mesures réalisées sur une période de 20 minutes à une fréquence d'échantillonnage de 2Hz. Les informations de niveau de surface libre d'un horodate sont stockées dans une liste de flottants `liste_niveaux`. On suppose qu'aucun des éléments de cette liste n'est égal à la moyenne.

6. Proposer une fonction `moyenne` prenant en argument une liste non vide `liste_niveaux` et renvoyant sa valeur moyenne.
7. Rappeler la formule de la valeur moyenne d'une fonction sur un segment à l'aide d'une intégrale. On suppose que l'on dispose d'une fonction `integrale_precise` prenant en argument une liste non vide `liste_niveaux` et renvoyant une valeur approchée de l'intégrale de η sur une période de 20 minutes.
En déduire une fonction `moyenne_precise` prenant en argument une liste non vide `liste_niveaux` et renvoyant une estimation de la moyenne de η sur une période de 20 minutes.
8. Proposer une fonction `ind_premier_pnd(liste_niveaux)` renvoyant, s'il existe, l'indice du premier élément de la liste tel que cet élément soit supérieur à la moyenne et l'élément suivant soit inférieur à la moyenne. Cette fonction devra renvoyer -1 si aucun élément vérifiant cette condition n'existe.
9. Proposer une fonction renvoyant l'indice i du *dernier* élément de la liste tel que cet élément soit supérieur à la moyenne et l'élément suivant soit inférieur à la moyenne. Cette fonction devra retourner -2 si aucun élément vérifiant cette condition n'existe. On cherchera à proposer une fonction de complexité $O(1)$ dans le meilleur des cas.
10. On souhaite stocker dans une liste `successeurs` les indices des points succédant (strictement) aux PND (voir figure 3).
On propose la fonction `construction_successeurs` en annexe (algorithme 1). Elle renvoie la liste `successeurs`. Compléter (sur la copie) les lignes 6 et 7.
11. Proposer une fonction `decompose_vagues(liste_niveaux)` qui permet de décomposer une liste de niveaux en liste de vagues. On omettra les données précédant le premier PND et celles succédant au dernier PND. Ainsi `decompose_vagues([1,-1,-2,2,-2,-1,6,4,-2,-5])` (noter que cette liste est de moyenne nulle) renverra `[[-1,-2,2],[-2,-1,6,4]]`.

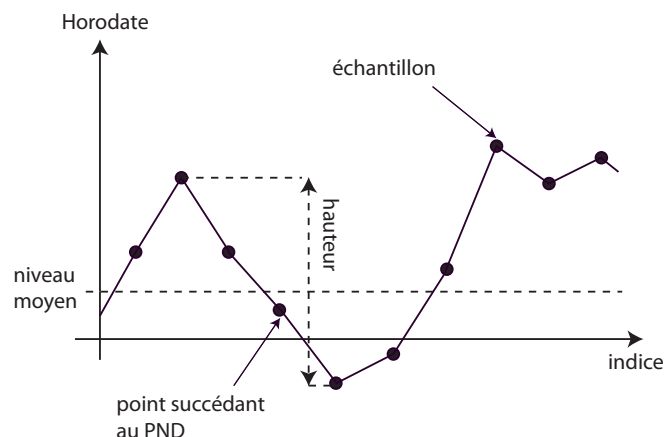


FIGURE 3 – Propriétés d'une vague

12. On désire maintenant caractériser les vagues.
Ainsi, on cherche à concevoir une fonction `proprietes(liste_niveaux)` renvoyant une liste de listes à deux éléments `[Hi, Ti]` permettant de caractériser *chacune des vagues* i par ses attributs :

- H_i , sa hauteur en mètres (m) (voir figure 3) ;
- T_i , sa période en secondes (s).

Proposer une fonction `proprietes(liste_niveaux)` réalisant cet objectif. On pourra utiliser les fonctions `max(L)` et `min(L)` de Python qui renvoient le maximum et le minimum d'une liste `L`, respectivement.

Partie III. Contrôle des données

Plusieurs indicateurs sont couramment considérés pour définir l'état de la mer. Parmi eux, on note :

- H_{max} : la hauteur de la plus grande vague observée sur l'intervalle d'enregistrement $[0, T]$;
- $H_{1/3}$: la valeur moyenne des hauteurs du tiers supérieur des plus grandes vagues observées sur $[0, T]$;
- $T_{H_{1/3}}$: la valeur moyenne des périodes du tiers supérieur des plus grandes vagues observées sur $[0, T]$.

13. Proposer une fonction prenant en argument la liste `liste_niveaux`, utilisant la question 12 et renvoyant H_{max} .

La distribution des hauteurs de vague (voir figure 4) lors de l'analyse *vague par vague* est réputée être gaussienne. On peut contrôler cela par des tests de *skewness* (variable désignée par S) et de *kurtosis* (variable désignée par K) définis ci-après. Ces deux tests permettent de quantifier respectivement l'asymétrie et l'aplatissement de la distribution.

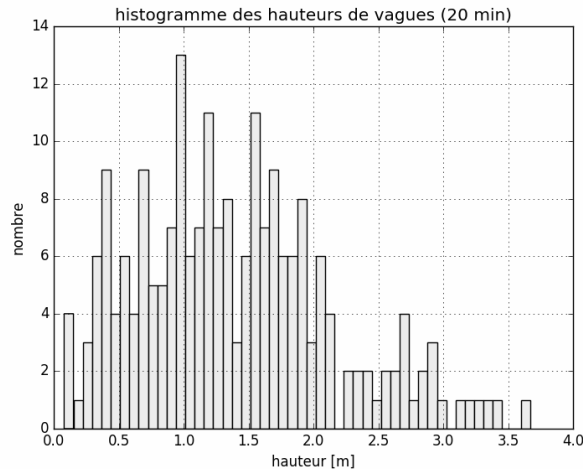


FIGURE 4 – Histogramme des hauteurs de vague

On appelle \bar{H} et σ^2 les estimateurs non biaisés de l'espérance et de la variance, n le nombre d'éléments H_1, H_2, \dots, H_n . On définit alors

$$S = \frac{n}{(n-1)(n-2)} \times \frac{1}{\sigma^3} \times \sum_{i=1}^n (H_i - \bar{H})^3$$

$$K = \frac{n}{(n-1)(n-2)(n-3)} \times \frac{1}{\sigma^4} \times \sum_{i=1}^n (H_i - \bar{H})^4 - \frac{3(n-1)^2}{(n-2)(n-3)}$$

Le test suivant est appliqué :

- si la valeur absolue de S est supérieure à 0,3 alors l’horodate est déclaré non valide ;
- si la valeur de K est supérieure à 5 alors l’horodate est déclaré non valide.

On utilise la fonction `moyenne` pour estimer la valeur de \overline{H} , et on suppose disposer de la fonction `ecartType` qui permet de renvoyer la valeur de l’écart-type non biaisé σ .

14. Un codage de la fonction `skewness` pour une liste ayant au moins trois éléments est donné en annexe (algorithme 2). Le temps d’exécution est anormalement long. Proposer une modification simple de la fonction pour diminuer le temps d’exécution (sans remettre en cause l’implémentation des fonctions `ecartType` et `moyenne`).
15. Doit-on s’attendre à une différence de type de la complexité entre une fonction évaluant S et une fonction évaluant K ?

Partie IV. Base de données relationnelle

On dispose d’une base de données relationnelle `Vagues`.

La première table est `Bouee`. On se limite aux attributs suivants : le numéro d’identification `idBouee`, le nom du site `nomSite`, le nom de la mer ou de l’océan `localisation`, le type du capteur `typeCapteur` et la fréquence d’échantillonnage `frequence`.

<code>idBouee</code>	<code>nomSite</code>	<code>localisation</code>	<code>typeCapteur</code>	<code>frequence</code>
831	Porquerolles	Mediterranee	Datawell non directionnelle	2.00
291	Les pierres noires	Mer d’iroise	Datawell directionnelle	1.28
...

La seconde table est `Campagne`. On se limite aux attributs suivants : le numéro d’identification `idCampagne`, le numéro d’identification de la bouée `idBouee`, la date de début `debutCampagne` et la date de fin `finCampagne`.

<code>idCampagne</code>	<code>idBouee</code>	<code>debutCampagne</code>	<code>finCampagne</code>
08301	831	01/01/2010 00h00	15/01/2010 00h00
02911	291	15/10/2005 18h30	18/10/2005 08h00
...

La troisième table est `Tempete`. Les informations fournies relatives à un événement « tempête » sont les suivantes :

- date de début et fin de tempête ;
- évolution des paramètres $H_{1/3}$ et H_{max} en fonction du temps
- Le détail de certains paramètres non définis ici, obtenus au pic de tempête.

On se limite aux attributs suivants : le numéro d’identification de la tempête `idTempete`, le numéro d’identification de la bouée `idBouee`, la date de début `debutTempete`, la date de fin `finTempete`, la valeur maximale de hauteur de vague `Hmax`.

<code>idTempete</code>	<code>idBouee</code>	<code>debutTempete</code>	<code>finTempete</code>	<code>Hmax</code>
083010	831	07/01/2010 20h00	09/01/2010 15h30	5.3
029012	291	16/10/2005 08h30	18/10/2005 09h00	8.5
...

16. Formuler les requêtes SQL permettant de répondre aux questions suivantes :

- « Quels sont le numéro d'identification et le nom de site des bouées localisées en Méditerranée ? »
- « Combien de campagnes ont été menées sur la bouée n° 300 ? »
- « Pour chaque bouée, quelle est la hauteur maximale des hauteurs H_{max} ? »
- « Quelles sont les localisations, par ordre alphabétique, et les sites, eux aussi par ordre alphabétique, des bouées ? »

Partie V. Analyse « spectrale »

L'analyse spectrale (fréquentielle) du niveau permet elle aussi de caractériser l'état de la mer qui peut, en première approximation, être modélisé par une superposition linéaire d'ondes sinusoïdales indépendantes. Des coefficients estimateurs de l'état de la mer issus de l'analyse

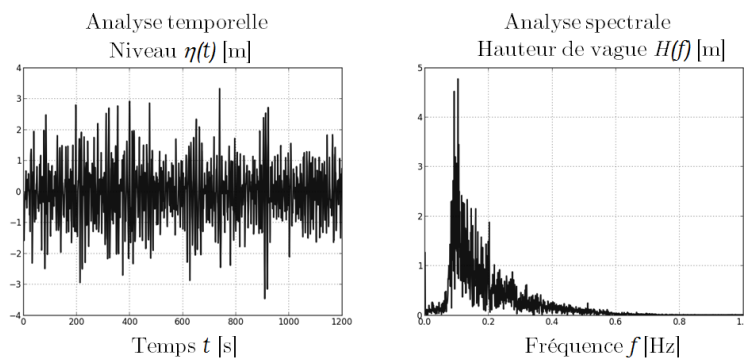


FIGURE 5 – Analyses temporelle et spectrale

spectrale ont donc été définis. Parmi eux, on note par exemple H_{m0} la hauteur significative spectrale des vagues ou T_P la période de pic barycentrique.

Pour leur calcul, il est nécessaire d'introduire la Transformation de Fourier Discrète (TFD). Sa définition pour un signal numérique x de N échantillons, est la suivante :

$$X_k = \sum_{i=0}^{N-1} x_i \times \exp(-2\pi jk \frac{i}{N}), \quad 0 \leq k < N \quad \text{et} \quad j^2 = -1 \quad (1)$$

Il existe plusieurs méthodes dites de « transformée de Fourier rapide ». On étudie dans la suite l'algorithme de Cooley–Tukey adapté de celui de Gauss. On propose ici une réécriture de (1) appelée entrelacement temporel (DIT decimation-in-time).

Dans toute la suite, on suppose que N est une puissance de 2.

On note $w = \exp(-2\pi j/N)$ (qui est une racine N -ième de l'unité).

On pose P_k (TFD des indices pairs) et I_k (TFD des indices impairs) :

$$P_k = \sum_{i=0}^{N/2-1} x_{2i} \times \exp(-2\pi jk \frac{i}{N/2})$$

$$I_k = \sum_{i=0}^{N/2-1} x_{2i+1} \times \exp(-2\pi jk \frac{i}{N/2})$$

On montre alors que pour $0 \leq k < \frac{N}{2}$,

$$X_k = P_k + w^k I_k$$

$$X_{k+N/2} = P_k - w^k I_k$$

L'algorithme est de type « diviser pour régner » : le calcul d'une TFD pour N éléments se fait à l'aide de deux TFD de $N/2$ éléments.

17. Quelle est la complexité en temps de cet algorithme récursif en fonction de N ? Justifier en une ou deux lignes.
18. Écrire une fonction *récursive* prenant en argument la liste de données x et renvoyant la liste X obtenue par transformée de Fourier discrète rapide. La longueur de x est une puissance de 2.

Annexe

Instructions diverses

Le nombre complexe j tel que $j^2 = -1$ s'obtient en Python par : `1j`.

<pre>f = open('nom_fichier', 'r') f = open('nom_fichier', 'w') f = open('nom_fichier', 'a') f.close() f.readlines()</pre>	<p>Ouvre le fichier en lecture. Le fichier doit exister et seule la lecture est autorisée.</p> <p>Ouvre le fichier en écriture. Si le fichier n'existe pas, il est créé. S'il existe, il est effacé.</p> <p>Ouvre le fichier en ajout (append). Si le fichier n'existe pas, il est créé. S'il n'existe pas, ce qu'on écrit est ajouté à la fin du fichier.</p> <p>Ferme le fichier ouvert précédemment.</p> <p>Lit tout le fichier d'un coup et le renvoie sous forme d'une liste de chaîne de caractères. Chaque élément de la liste correspond à une ligne. Attention, le saut de ligne <code>/n</code> est présent à la fin de chaque chaîne.</p>
---	--

Algorithme 1

```
1 def construction_succeurs(liste_niveaux):
2     n = len(liste_niveaux)
3     successeurs = []
4     m = moyenne(liste_niveaux)
5     for i in range(n-1):
6         if # Completer
7           # Completer
8     return successeurs
```

Algorithme 2

```
1 def skewness(liste_hauteurs):
2     n = len(liste_hauteurs)
3     et3 = (ecartType(liste_hauteurs))**3
4     S = 0
5     for i in range(n):
6         S += (liste_hauteurs[i] - moyenne(liste_hauteurs))**3
7     S = n/(n-1)/(n-2) * S/et3
8     return S
```