DS 3
Informatique
Extrait Concours Mines-Ponts
Mesures de houle

Simon Dauguet
simon.dauguet@gmail.com

Mercredi 12 Février 2025

Le devoir dure 2h.

La qualité de la rédaction et de la présentation seront prises en compte dans la notation. On prendra bien garde à la justesse et la précision des justifications.

Si un candidat repère ce qui lui semble être une erreur d'énoncé, il l'identifiera clairement sur la copie et explicitera les décisions qu'il sera amené à prendre.

La calculatrice n'est pas autorisée.

Le sujet comporte 7 pages.

Le sujet comporte des questions de programmation. Le langage à utiliser est le langage Python.

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 étudier[2].

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).

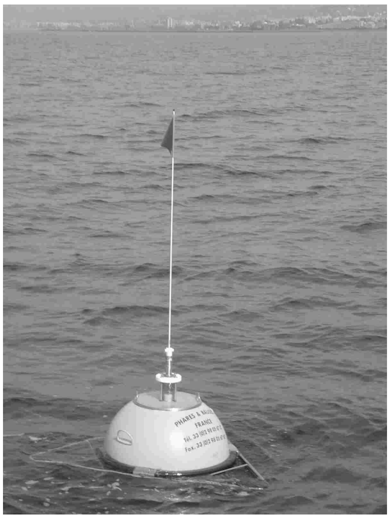

FIGURE 1 – Bouée de mesure de houle.

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 : $2\ Hz$.

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. 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, ..., 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 $[0, Z_1]$.

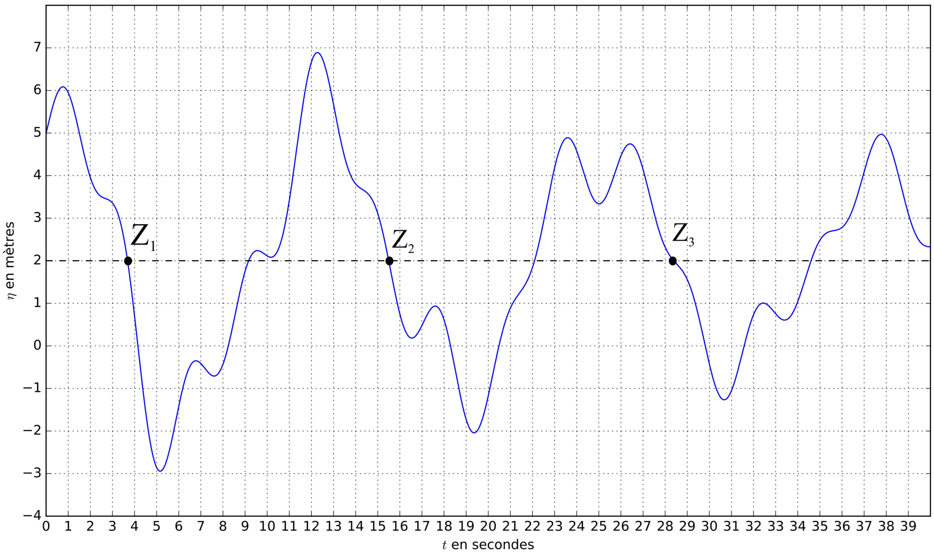

FIGURE 2 – Passage par le Niveau moyen en Descente (PND). Ici la moyenne m vaut 2.

Les hauteurs des vagues H_i sont définies par les différences

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

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(liste_niveaux:list) -> float` prenant en argument une liste non vide **`liste_niveaux`** et renvoyant sa valeur moyenne.

7. Proposer une fonction `ind_premier_pzd(liste_niveaux:list) -> int` 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.

8. 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.

On souhaite stocker dans une liste **successeurs** les indices des points succédant (strictement) aux PND (voir figure 3).

9. On propose la fonction `construction_successeurs` en annexe (algorithme 1). Elle renvoie la liste `successeurs`. Compléter (sur la copie) les lignes 6 et 7.
10. Proposer une fonction `decompose_vagues(liste_niveaux:list) -> list` 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]]`.

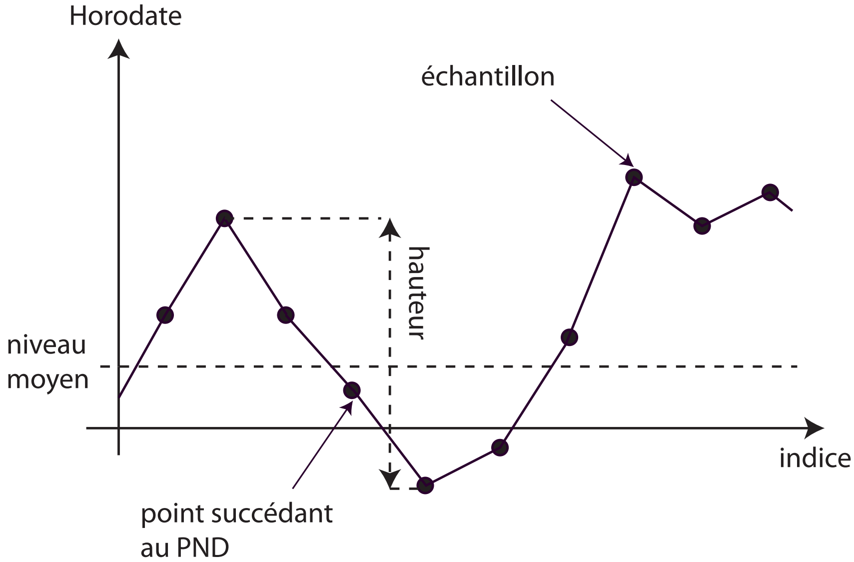

FIGURE 3 – Propriétés d'une vague

On désire maintenant caractériser les vagues.

Ainsi, on cherche à concevoir une fonction `proprietes(liste_niveaux:list) -> list` renvoyant une liste de listes à deux éléments `[Hi,Ti]` permettant de caractériser *chacune des vagues* i par ses attributs :

— `Hi`, sa hauteur en mètres (m) (voir figure 3) ;
— `Ti`, sa période en secondes (s).

11. Proposer une fonction `proprietes(liste_niveaux:list) -> list` 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 $[0, T]$.

12. Proposer une fonction prenant en argument la liste `liste_niveaux` de 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 ceci par des tests de *skewness* (variable désignée par S) et de *kurtosis*

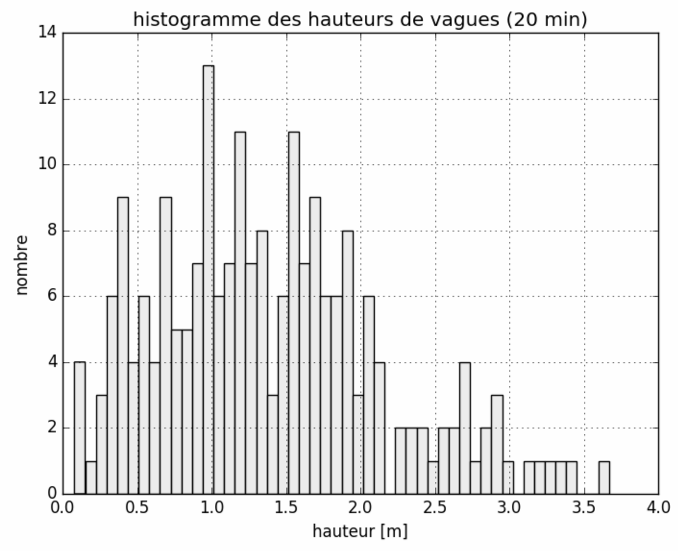

FIGURE 4 – Histogramme des hauteurs de vague

(variable désignée par K) définis ci-après. Ces deux tests permettent de quantifier respectivement l'asymétrie et l'aplatissement de la distribution.

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

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

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

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.

13. Un codage de la fonction **`skewness`** pour une liste ayant au moins trois éléments est donné en annexe (algorithme 4). 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`**).

14. Doit-on s'attendre à une différence de type de la complexité entre une fonction évaluant S et une fonction évaluant K ?

Partie IV. 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 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).

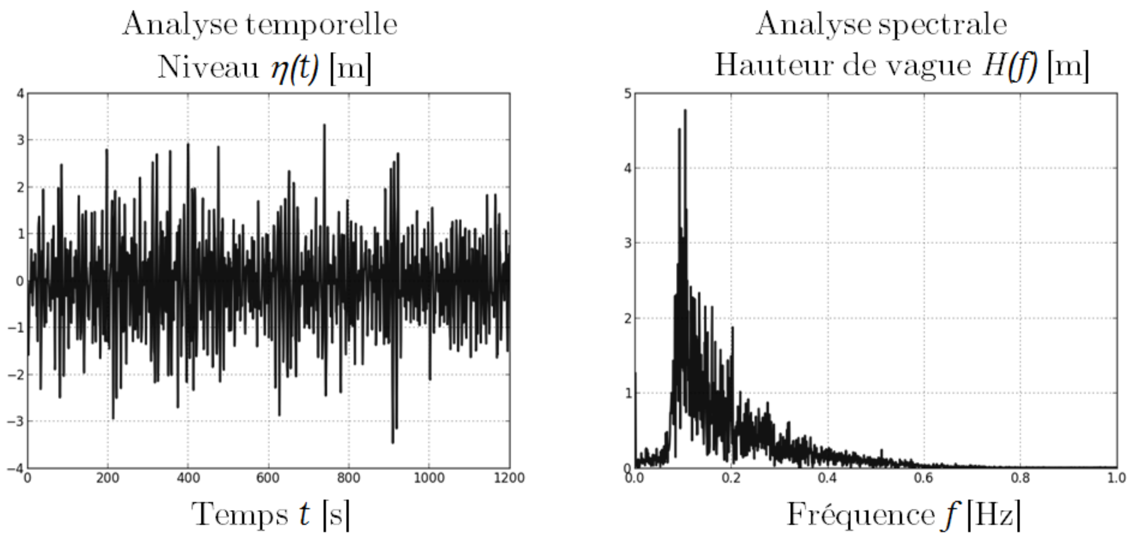

FIGURE 5 – Analyses temporelle et spectrale

Sa définition pour un signal numérique x de N échantillons, est la suivante :

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

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) appelé entrelacement temporel (DIT decimation-in-time).

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

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

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

$$\begin{aligned} P_k &= \sum_{i=0}^{N/2-1} x_{2i} \times e^{-2\pi jk\,\frac{i}{N/2}} \\ I_k &= \sum_{i=0}^{N/2-1} x_{2i+1} \times e^{-2\pi jk\,\frac{i}{N/2}} \end{aligned}$$

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

$$\begin{aligned} X_k &= P_k + w^k\, I_k \\ X_{k+N/2} &= P_k - w^k\, I_k \end{aligned}$$

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.

15. Quelle est la complexité en temps de cet algorithme en fonction de N ? Justifier en une ou deux lignes.
16. 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.

Annexes

‘

Algorithme 1

```python
def construction_successeurs(liste_niveaux):
    n = len(liste_niveaux)
    successeurs = []
    m = moyenne(liste_niveaux)
    for i in range(n-1):
        if                  # À compléter
                            # À compléter
    return successeurs
```

Algorithme 4

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

Fin de l'épreuve.