

# Concours Blanc : épreuve d'Informatique

PTSI Lycée Eiffel

5 juin 2018

## Durée : 2H.

Dans tout ce problème, on s'intéresse à des mesures de houle (déplacement vertical du niveau de la mer) effectuées par une bouée située à une certaine distance de la côte. Cette bouée contient un ensemble de capteurs incluant un accéléromètre vertical qui fournit, après traitement approprié, des mesures à étudier. Une campagne de mesure est effectuée en respectant les paramètres suivants :

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

### Première partie : fichier de stockage des mesures.

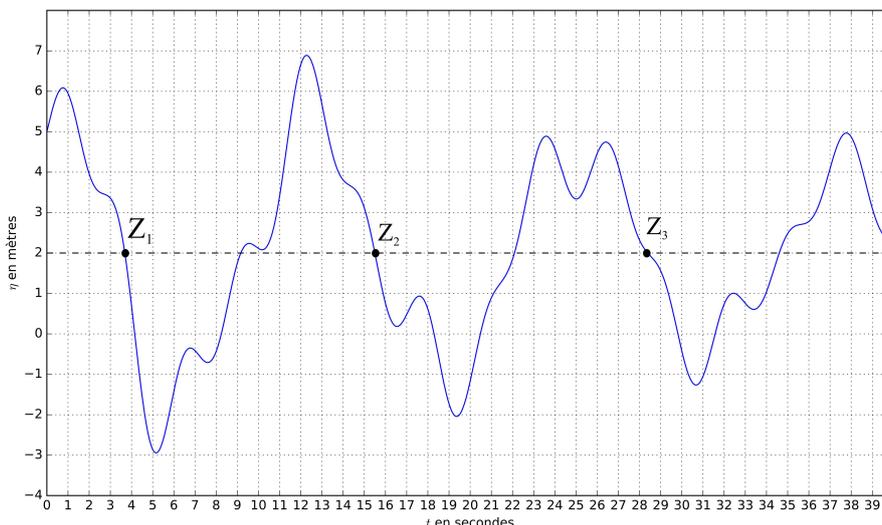
Les relevés effectués lors d'une campagne de mesure sont stockés dans un fichier texte dont le contenu est défini comme suit : les informations générales relatives à la campagne (numéro de campagne, nom du site, type de capteur, latitude et longitude de la bouée, date et heure de la séquence) sont toutes rassemblées sur la première ligne du fichier, séparées par des points-virgules. Les lignes suivantes contiennent chacune une mesure de déplacement vertical (en mètres). Chacune de ces lignes est constituée de huit caractères, dont un signe et un caractère de passage à la ligne, ce qui laisse donc quatre chiffres significatifs après le séparateur décimal (on suppose que la houle ne dépassera jamais les 10 mètres), par exemple  $+0.4257$  ou  $-0.0824$ .

1. On suppose que chaque caractère du fichier est stocké sur 8 bits. En ne tenant pas compte de la première ligne, déterminer le nombre d'octets nécessaire pour stocker une série de mesures sur 20 minutes à une fréquence d'échantillonnage de 2Hz.
2. En déduire un ordre de grandeur de l'espace mémoire nécessaire pour stocker toutes les données correspondant à une campagne de mesures complète. Une carte-mémoire de 1Go sera-t-elle suffisante ?
3. Pour réduire la taille du fichier, on souhaite supprimer deux chiffres significatifs aux mesures effectuées. Quel sera le gain relatif d'espace mémoire ?
4. Les données sont stockées dans un fichier nommé **donnees.txt**. Proposer une suite d'instructions en Python permettant de créer à partir de ce fichier une liste des valeurs numériques du niveau de la mer (on prendra garde à ne pas insérer la première ligne du fichier de données, et à bien gérer les signes).

## Deuxième partie : analyse des données par vague.

On souhaite étudier une série de valeurs obtenue lors d'une période de 20 minutes (on supposera pour simplifier que le temps est systématiquement mesuré à partir de 0), et on suppose ces valeurs stockées dans une liste Python nommée **listeniveaux**, comme demandé à la dernière question de la première partie du sujet. On notera  $m$  la moyenne des hauteurs mesurées lors de cette série, et on notera  $Z_1, Z_2, \dots, Z_n$  les Passages par le Niveau moyen en Descente (PND), c'est-à-dire les instants correspondant à une mesure de houle inférieure à la moyenne  $m$ , mais pour lesquels la mesure précédente était supérieure à  $m$  (autrement dit, entre la mesure précédente et la mesure concernée, on a traversé la valeur moyenne en descendant, d'où le nom). Pour simplifier, on supposera que la première valeur obtenue en début de mesure est systématiquement supérieure à  $m$ . On appellera *période de vague* l'écart  $T_i = Z_{i+1} - Z_i$  entre deux PND successifs, et *hauteur de vague* la valeur  $H_i = \max_{t \in [Z_{i-1}, Z_i]} h(t) - \min_{t \in [Z_i, Z_{i+1}]}$ , où  $h(t)$  désigne la hauteur de la houle. Dans le cas où  $i = 1$ , on remplacera dans cette définition  $Z_0$  par 0.

Sur la figure ci-dessous (où on a représenté  $h(t)$  comme une fonction continue, alors qu'on ne dispose en fait que d'un échantillon discret de valeurs de la fonction), la moyenne  $m$  est égale à 2 mètres, et les PND  $Z_1, Z_2$  et  $Z_3$  sont indiqués :



1. Estimer les valeurs de  $H_1, H_2$  et  $H_3$ , ainsi que celles de  $T_1$  et  $T_2$ , pour la figure précédente.
2. Écrire une fonction Python **moyenne** calculant la moyenne des valeurs contenues dans une liste (on évitera de recourir à la fonction **sum**).
3. Écrire une fonction Python **intliste** prenant comme argument une liste de valeurs (telle que la liste **listeniveaux** décrite ci-dessus), et renvoyant une valeur approchée de l'intégrale de la fonction  $h(t)$  correspondant à cette échantillon, en utilisant la méthode des trapèzes. En déduire une nouvelle fonction **moyint** déterminant une valeur approchée de la moyenne des valeurs de la liste (en exploitant la valeur de l'intégrale qu'on vient de calculer).
4. Écrire une fonction **premierpnd** retournant l'indice du premier élément de notre liste correspondant à un PND. S'il n'y a aucun PND dans la liste, on retournera la valeur  $-1$ .
5. Proposer de même une fonction déterminant l'indice du dernier PND dans la liste (s'il n'y en a pas, on retournera à nouveau  $-1$ ).
6. Écrire désormais une fonction **constructionpnd** permettant de stocker dans une liste tous les indices correspondant à un PND.
7. Écrire une fonction **decomposevagues** permettant de découper la liste **listeniveaux** en vagues. On retournera une liste de listes en omettant toutes les valeurs précédant le premier

PND et toutes celles suivantes le dernier PND. Ainsi, `decomposevagues`([1,-1,-2,2,-2,-1,6,4,-2,-5]) renverra la liste [[-1,-2,2],[-2,-1,6,4]] (la liste proposée étant de moyenne nulle).

- Écrire enfin une fonction `proprietesvagues` qui crée deux listes `periodes` et `hauteurs` permettant de stocker les hauteurs et périodes de toutes les vagues correspondant à la liste `listevaleurs`. On pourra ici utiliser les fonctions Python `max` et `min` pour alléger le code.

## Troisième partie : Contrôle des données.

Pour définir l'état de la mer à partir des données obtenues lors d'une série de mesures, on récupère la liste `hauteurs` des différentes hauteurs de vagues mesurées, et on définit les paramètres suivants :

- $H_{max}$  est simplement la hauteur de la plus grande vague mesurée dans cette série.
- $H_{1/3}$  est la moyenne des hauteurs du tiers supérieur des plus grandes vagues observées. Ainsi, si on a observé 60 vagues lors de la série de mesure, on ne retient que les 20 vagues les plus hautes, et on calcule la moyenne des 20 hauteurs de vagues correspondantes.
- $S = \frac{n}{(n-1)(n-2)} \times \frac{1}{\sigma^3} \sum_{i=1}^n (H_i - m)^3$  est un facteur appelé *skewness* qui mesure l'asymétrie de la distribution des différentes hauteurs de vagues (qui est théoriquement censée être gaussienne). Dans la formule, les  $H_i$  sont toutes les hauteurs de vagues mesurées (on a donc repéré  $n$  vagues sur cette série de mesures),  $m$  désigne la hauteur moyenne de toutes ces vagues, et  $\sigma$  leur écart-type.

- Écrire une fonction Python prenant comme argument la liste `listeniveaux`, et retournant la valeur de  $H_{max}$  (on n'utilisera pas la fonction Python `max` pour cette question).
- Pour déterminer  $H_{1/3}$ , on a d'abord besoin de trier les éléments de la liste. On se propose pour cela d'utiliser un algorithme de tri par insertion : on crée une liste initialement vide dans laquelle on insère successivement chaque élément de la liste initiale en le positionnant à sa place (dans l'ordre croissant).
  - Écrire une fonction Python `insereliste(l,a)` prenant comme argument une liste `l` triée dans l'ordre croissant, et un nombre `a`, et insérant la valeur `a` dans la liste `l` en conservant la liste triée.
  - En déduire une fonction `triinsertion(l)` effectuant un tri par insertion d'une liste `l`.
- On suppose qu'on dispose déjà de fonctions `moyenne` et `ecarttype` calculant respectivement la moyenne et l'écart-type d'une liste de valeurs. Écrire une fonction Python `skewness` prenant comme argument une liste `l` et retournant la valeur du paramètre de skewness correspondant.
- On considère que la liste de mesures observée est « normale » si elle a une skewness inférieure ou égale à 0.3. Écrire une fonction Python déterminant si une liste de hauteurs donnée correspond à une série de mesures normale (on renverra `True` ou `False`).

## Quatrième partie : Modélisation numérique des vagues.

La houle mesurée par une bouée semble se prêter à une modélisation numérique, où elle vérifierait une équation différentielle du type :

$$h''(t) = Kh(t) + a(t)h'(t)$$

où  $K$  est une constante réelle, et  $a$  une fonction de perturbation supposée connue (et déjà implémentée en Python).

1. Écrire un programme Python **Euler** permettant d'effectuer la résolution approchée d'une équation différentielle d'ordre 1, en traçant l'allure de la courbe de la solution approchée à l'aide du module **matplotlib.pyplot**.
2. Que vaut ici la fonction  $F$  représentant l'équation différentielle dans le programme précédent ? Quelles valeurs donnera-t-on aux différents paramètres si on veut pouvoir comparer les valeurs obtenues à celles de nos séries de mesures effectuées sur une période de 20 minutes avec une fréquence de 2Hz ?
3. On suppose qu'on dispose de la liste **listevaleurs** des mesures expérimentales, et d'une liste **valeurstheoriques** contenant les valeurs données par la simulation numérique définie ci-dessus. Pour mesurer l'adéquation du modèle aux mesures effectuées, on souhaite calculer l'écart relatif maximal obtenu entre les valeurs des deux listes. Proposer une fonction Python effectuant ce calcul (on a le droit d'utiliser la commande Python **max** pour cette question).

## Cinquième partie : exploitation d'une base de données.

Certaines données concernant les mesures effectuées sont stockées dans une base de données contenant les trois tables suivantes :

- une table **Bouee** contenant les attributs idbouee (numéro d'identification de la bouée), localisation (chaîne de caractères contenant le nom de la mer ou de l'Océan où est située la bouée), nomSite (chaîne de caractères contenant le nom précis du site de positionnement de la bouée), typecapteur (chaîne de caractères), fréquence (flottant représentant la fréquence d'échantillonnage de la bouée).
- une table **Campagne** contenant les attributs idcampagne (numéro d'identification), idbouee. (clé secondaire), datedébut (date de début de la campagne de mesures, chaîne de caractères au format jjmmaaaa), datefin.
- une table **Tempetes** contenant des données plus précises concernant les tempêtes : un identifiant idtempete, idbouee (clé secondaire, comme dans la table précédente), datetempete, hmax (valeur maximale de la houle pendant le jour de tempête, nombre flottant).

1. Écrire une requête SQL donnant le numéro d'identifiant et le nom de site de toutes les bouées situées en mer Méditerranée.
2. Écrire une requête SQL donnant les noms des sites où a été effectuée une campagne de mesures en juillet 2015.
3. Écrire une commande SQL permettant d'ajouter à la table **Bouee** une colonne Age permettant de stocker l'âge de la bouée (nombre d'années depuis sa mise en service).
4. Écrire une commande SQL permettant d'afficher le nom du site où a été observée la vague la plus haute le 28 avril 1981 (en supposant qu'il y a eu des tempêtes ce jour-là).