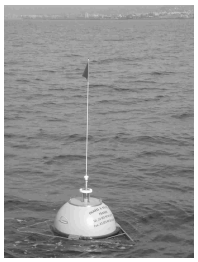
# DS 06



## Mesures de houle

### Question de cours – Résolution d'équations différentielles

On cherche à résoudre l'équation différentielle $y' = y$ sur l'intervalle $[0,4]$ avec $y(0) = 1$.

**Question 1** *Écrire la fonction de Cauchy* `f1(y:float,t:float)-> float` *associée à cette équation différentielle.*

**Question 2** *Écrire la fonction* `euler(a:float,b:float,y0:float,h:float,f:function)-> list,list` *permettant de résoudre cette équation différentielle.*

**Question 3** *Après avoir importé les bibliothèques nécessaires, donner les instructions permettant de tracer la solution de l'équation différentielle en fonction du temps. On prendra* $h = 0,001$.

On cherche à résoudre l'équation différentielle $y'' + y = 0$ sur l'intervalle $[0,10]$ avec $y(0) = 0$ et $y'(0) = 1$.

**Question 4** *Écrire la fonction de Cauchy* `f2(y:np.array,t:float)-> np.array` *associée à cette équation différentielle.*

**Question 5** *La définition de votre fonction* `f2` *peut-elle être utilisée avec la fonction* `euler` *? Si non, expliquer pourquoi et proposer des modifications.*

## 1 Introduction

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

Les mesures réalisées à bord de la bouée sont envoyées par liaison radio à une station à terre où elles sont enregistrées, contrôlées et diffusées pour servir à des études scientifiques. Pour plus de sûreté, les mesures sont aussi enregistrées sur une carte mémoire interne à la bouée.

## 2 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 (m). Chaque ligne comporte 8 caractères dont le caractère de fin de ligne. Par exemple, on trouvera dans le fichier texte les 3 lignes suivantes :

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

**Question 6** *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* 2 Hz.

**Question 7** *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* 1 Go *est-elle suffisante ?*

**Question 8** *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 ?*

**Question 9** *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 ».

## 3 Analyse « vague par vague »

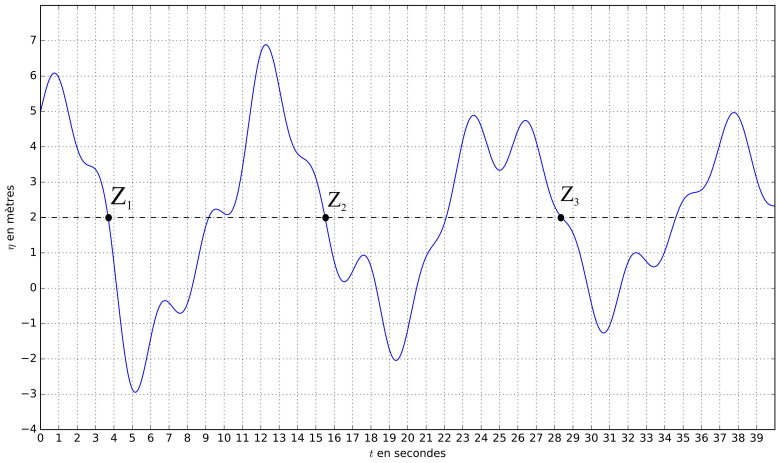
On considère ici que la mesure de houle est représentée par un signal $\eta(t) \in \mathbb{R}$, $t \in [0, T]$, avec $\eta$ une fonction $C^1$. On appelle niveau moyen $m$ la moyenne de $\eta(t)$ sur $[0, T]$. On définit $Z_1$, $Z_2$, ..., $Z_n$ l'ensemble (supposé fini) des Passages par le Niveau moyen en Descente (PND) (voir Figure suivante). À chaque PND, le signal traverse la valeur m en descente. On suppose $\eta(0) > m$ et $\dfrac{\mathrm{d}\eta}{\mathrm{d}t}(0) > 0$. On en déduit que $\eta(t) - m \geq 0$ sur $[0, Z_1]$. Les hauteurs de vagues $H_i$ sont définies par les différences :

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

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

**Question 10** *Pour le signal représenté sur la figure suivante, 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 2 Hz. 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.




*Passages par le Niveau moyen en Descente (PND). Ici la moyenne m vaut 2*

On précise que la liste `liste_niveaux` contient l'ensemble des valeurs ($\eta$) du niveau de la mer.

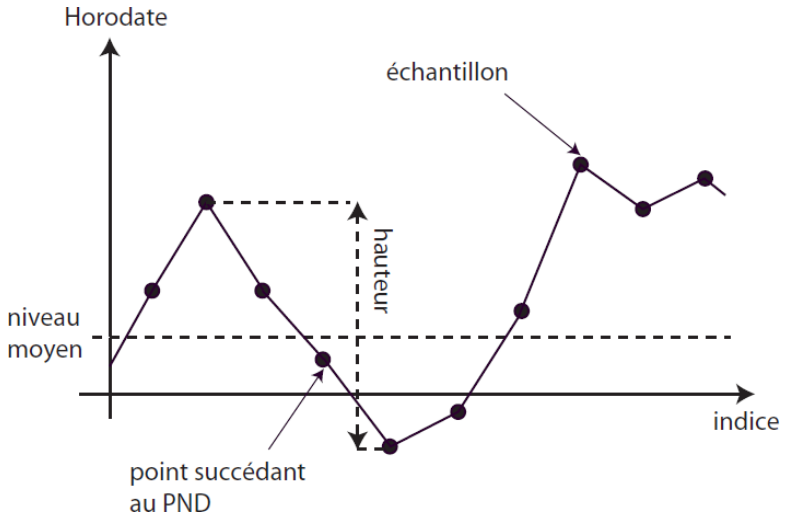
**Question 11** *Proposer une fonction* `moyenne` *prenant en argument une liste non vide* `liste_niveaux`, *et retournant sa valeur moyenne.*

**Question 12** *Proposer une fonction* `integrale_precise` *prenant en argument une liste non vide* `liste_niveaux`, *et retournant la valeur approchée de l'intégrale de* $\eta$ *sur une période de 20 minutes. On demande d'utiliser la méthode des trapèzes (on supposera que l'échantillonnage est constant). En déduire une fonction* `moyenne_precise` *prenant en argument une liste non vide* `liste_niveaux` *et retournant une estimation de la moyenne de* $\eta$ *sur une période de 20 minutes.*

**Question 13** *Proposer une fonction* `ind_premier_pzd(liste_niveaux:list) -> int` *retournant, 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 retourner -1 si aucun élément vérifiant cette condition n'existe.*

**Question 14** *Proposer une fonction* `ind_dernier_pzd(liste_niveaux:list) -> int` *retournant 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 suivante).




*Propriétés d'une vague*

**Question 15** *On propose la fonction* `construction_successeurs`. *Elle retourne la liste* `successeurs`. *Compléter (sur la copie) les lignes à compléter.*

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

**Question 16** *Donner la complexité algorithmique de la fonction* `construction_successeurs` *dans le pire des cas.*

**Question 17** *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) retournera* `[[-1,-2,2], [-2,-1,6,4]]`.

On désire maintenant caractériser les vagues. Ainsi, on cherche à concevoir une fonction `proprietes(liste_niveaux)` retournant 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 précédente),
- `Ti`, sa période en secondes (s).

**Question 18** *Proposer une fonction* `proprietes(liste_niveaux)` *réalisant cet objectif. On pourra utiliser les fonctions de Python* `max(L)` *et* `min(L)` *qui retournent le maximum et le minimum d'une liste* `L`, *respectivement.*

**Fin du sujet**