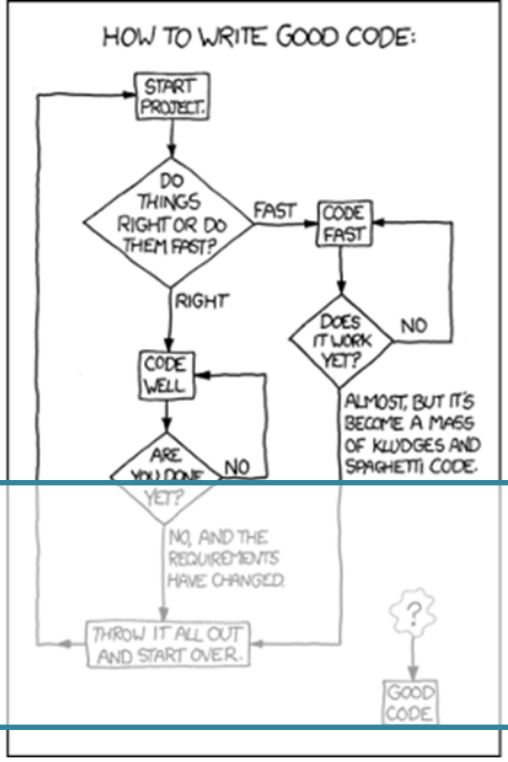
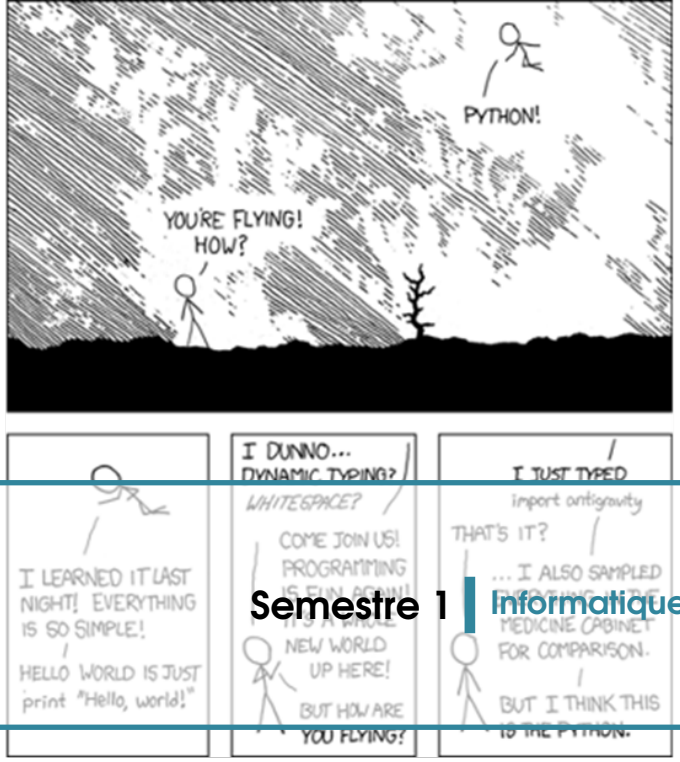




Thèmes d'étude

Recherche séquentielle
Structures imbriquées
Utilisation Modules

Exercice 1 – Surfing Porquerolles

TODO : corrigé

D'après Concours Mines Ponts 2018.

Objectif

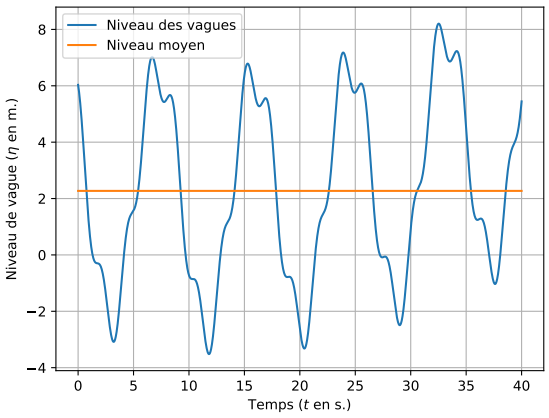
- Lire un fichier texte.
- Analyser les données d'un fichiers.

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 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 :

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

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



Le fichier `vagues.txt` contient un relevé des niveaux d'eau mesurés par une bouée au large de Porquerolles. (Pour ne pas se mentir, on a plutôt généré un profil qui pourrait vaguement ressembler à un tel relevé.)

Il est constitué de deux colonnes, séparées par une virgule, la première colonne correspondant à une mesure de temps (en secondes), la seconde colonne correspondant à une mesure de niveau de hauteur d'eau (en mètres).

Question 1 *Écrire une fonction* `lire_fichier(file: str) -> list,list` *prenant comme argument le nom d'un fichier et renvoyant la liste des temps que l'on notera* `les_t` *et la liste des niveaux de vagues que l'on notera* `liste_niveaux`.

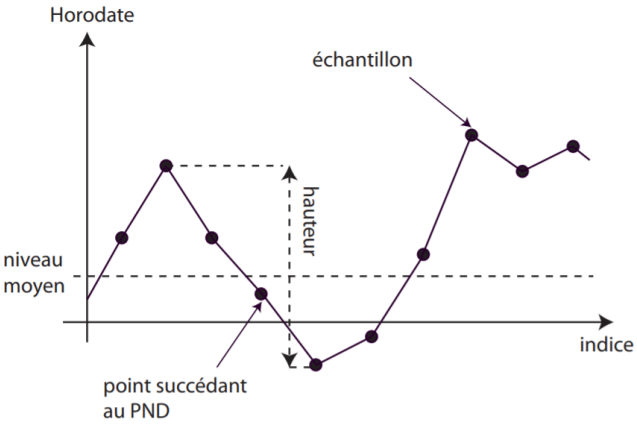
Question 2 *Écrire une fonction* `trace_vagues(file: str) -> None` *prenant comme argument le nom d'un fichier affichant le profil des vagues en fonction du temps.*

Question 3 *Écrire une fonction* `moyenne(liste_niveaux: list) -> float` *prenant comme argument une liste non vide* `liste_niveaux`*, et retournant sa valeur moyenne.*

Question 4 *Écrire 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 5 *Écrire 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 souhaite stocker dans une liste successeurs, les indices des points succédant (strictement) aux PND (voir figure suivante).



Question 6 *Écrire une fonction* `construction_successeurs(liste_niveaux: list) -> list` *retournant la liste successeurs.*

Question 7 *Écrire 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) retournera* `[[-1, -2, 2], [-2, -1, 6, 4]]`.

On désire maintenant caractériser les vagues. Ainsi, on cherche à concevoir une fonction

`proprietes(liste_niveaux: list) -> list` 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) ;
- `Ti`, sa période en secondes (s).

Question 8 *Écrire une fonction* `proprietes(liste_niveaux: list) -> list` *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.*

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 $[0, T]$;
- $T_{H1/3}$: la valeur moyenne des périodes du tiers supérieur des plus grandes vagues observées sur $[0, T]$.

Question 9 *Écrire une fonction* *** *prenant en argument la liste liste_niveaux de la question 12*** *et renvoyant* H_{max}.

Afin de déterminer $H_{1/3}$ et $T_{H1/3}$, il est nécessaire de trier la liste des propriétés des vagues.

****

La distribution des hauteurs de vague (voir figure suivante) 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 (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 :

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

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

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 retourner la valeur de l'écart type non biaisé σ.

Question 10 *Un codage de la fonction* `skewness` *pour une liste ayant au moins 3 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 le codage des fonctions* `ecartType` *et* `moyenne`*).*

Algorithmes dichotomiques
Fonctions récursives
Algorithmes gloutons
Traitement d'images
Tris