TP 04 — Activités pratiques

1 Randonnée

On souhaite réaliser différentes opérations sur un parcours de randonnée effectué lors d'un weekend à la montagne.

Pour s'entraîner nous allons nous entrainer sur un profil fictif.

1.1 Analyse du profil global

1.2 Question préliminaire

La variable `les_x` contient l'abscisse d'un profil. La variable `les_y` contient l'altitude d'un profil. Saisir les instruction suivantes.

```
plt.ylabel("Altitude [m]")
plt.plot(les_x,les_y,label = "Profil")
plt.legend()
plt.grid()
plt.show()
```

Question 1 *Vérifier qu'un profil de montagne s'affiche.*

Question 2 *Écrire la fonction* `maximum(L:list) -> int` *permettant de déterminer le maximum d'une liste (la fonction* `max` *sera ici interdite).*

Question 3 *Vérifier que* `maximum(les_y[0:300])` *renvoie* `1855.99`.

Question 4 *Écrire la fonction* `plus_haut_indice(L:list) -> float` *permettant de déterminer l'indice de l'altitude la plus haute atteinte lors de la randonnée.*

Question 5 *Vérifier que* `plus_haut_indice(les_y[300:500])` *renvoie* `199`.

Question 6 *Écrire la fonction* `deniveles(alt:list) -> list` *qui calcule les dénivelés cumulés positif et négatif (en mètres) de la randonnée, sous forme d'une liste de deux flottants. Le dénivelé positif est la somme des variations d'altitude positives sur le chemin, et inversement pour le dénivelé négatif.*

Question 7 *Vérifier que* `deniveles(les_y)` *renvoie* `[3438.100, -2746.747]`.

1.3 Découpage du profil

Dans cette partie, nous allons chercher à découper le profil de la randonnée en tentant de retrouver les différentes montagnes franchies par le randonneur.

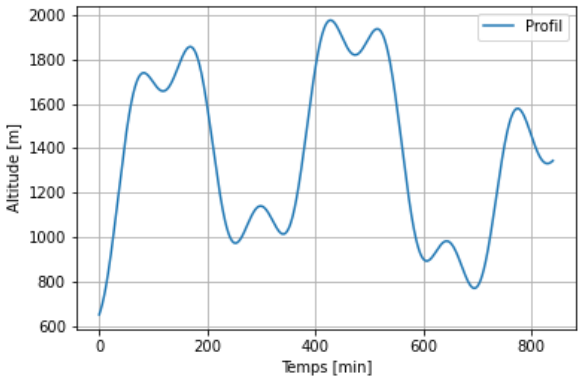

FIGURE 1 – Profil de la randonnée

Question 8 *Écrire la fonction* `moyenne(alt:list) -> float` *permettant de calculer la moyenne des altitudes mesurées par le GPS (l'utilisation de* `sum` *est interdite).*

Question 9 *Vérifier que* `moyenne(les_y)` *renvoie* `1376.35`.

On note PND les points de passages par le niveau moyen en descente et PNM les points de passage par le niveau moyen en montée.

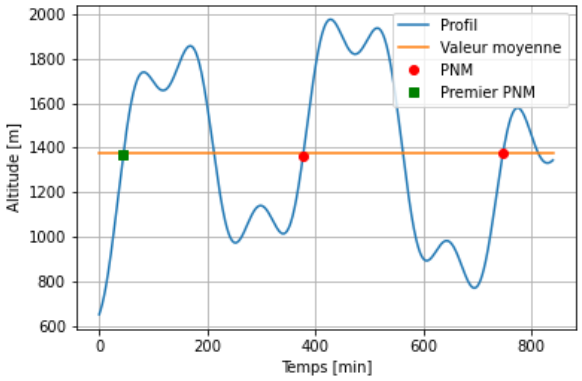

FIGURE 2 – Points de passage par le niveau moyen en montée [PNM]

Question 10 *Écrire la fonction* `indice_premier_PNM(alt:list) -> int` *renvoyant, s'il existe, l'indice* `i` *du premier élément de la liste tel que cet élément soit inférieur à la moyenne et l'élément suivant soit supérieur à la moyenne. Cette fonction devra renvoyer* `-1` *si aucun élément vérifiant cette condition n'existe.*

Question 11 *Vérifier* `indice_premier_PNM(les_y)` *renvoie* `53` *et* `indice_premier_PNM(les_y[200:250])` *renvoie* `-1`.

Question 12 *Écrire la fonction* `indices_PNM(alt:list) -> list` *retournant la liste des indices de tous les PNM.*

Question 13 *Vérifier* `indices_PNM(les_y)` *renvoie* `[53, 449, 889]`.

Question 14 *Dans le but de séparer les différents profils, nous allons chercher les indices des altitudes minimales entre deux PNM successifs. Écrire la fonction* `liste_alt_mini(alt:list) -> list` *qui répond à ce besoin. En utilisant le profil donné, cette fonction renverrait la liste* `[300,826]`.

On appelle `pam` la liste des indices des points ayant une altitude minimale.

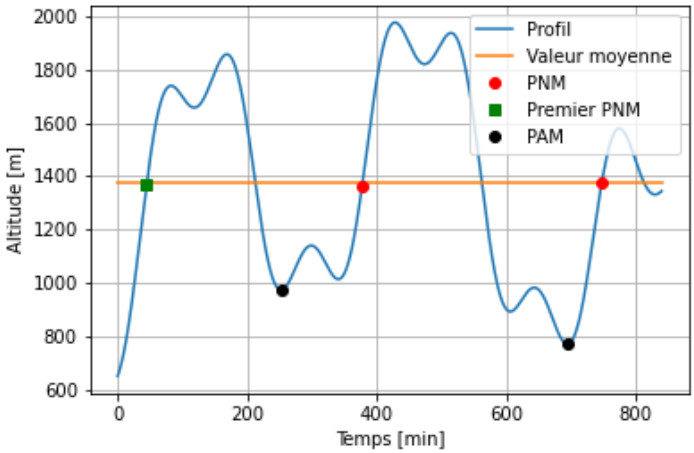

FIGURE 3 – Points avec altitude minimale [pam]

On souhaite maintenant décomposer le profil mesuré en plusieurs « montagnes ». Une montagne est une liste constituée d'altitudes successives. La première montagne ira de la première altitude mesurée au premier `pam`. On aura ensuite une montagne entre chaque `pam`. La dernière montagne ira du dernier `pam` à la dernière altitude mesurée.

Question 15 *Écrire la fonction* `creer_montagnes(alt) -> list` *renvoyant une liste constituée de la liste des montagnes élémentaires.*

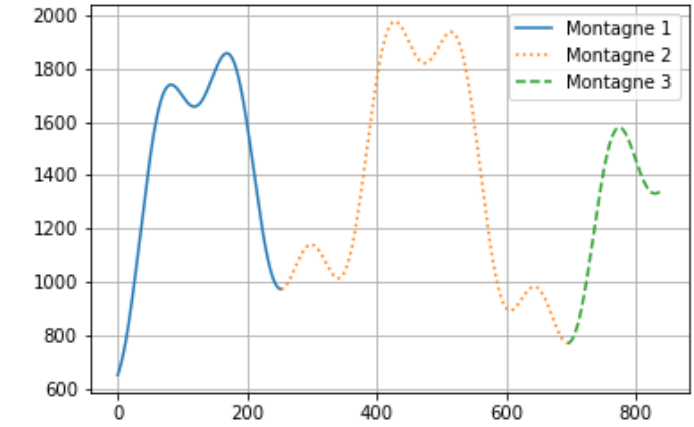

FIGURE 4 – Découpage en montagnes