

# Rendu semaines 7-8

David Wiedemann

12 novembre 2020

## 1.5

Il y a plusieurs problèmes avec ce programme.

- `n` est un entier, tandis que `v.size()` est de type `size_t`.
- On ne vérifie pas que le tableau contient un élément, il se pourrait qu'il est vide, ce qui donnerait une erreur.
- Dans la dernière étape de la boucle `for`, on accède à un élément qui n'est pas dans le tableau (les tableaux sont numérotés de 0 à  $n - 1$  pas jusqu'à  $n$ )
- Bien qu'on initialise `m` à la valeur d'un double, `m` prendra toujours des valeurs entières car `s` est un entier et `v.size()` l'est aussi.

## 2.1.2

L'erreur maximale qu'on peut faire est de 0.2 car  $6 - 5.8 = 0.2$ , donc l'erreur relative est

$$\frac{0.2}{5.8} \simeq 0.0344$$

## 2.3.1

On a

$$10011100_2 = 156_{10}$$

## 4.1

La fonction `f` regarde si un élément `a` est contenu dans un tableau.

On regarde si la différence entre les éléments est plus petite qu'une certaine précision donnée car l'opérateur `==` a des comportements mal définis quand on compare des doubles.

Un meilleur nom pour la fonction serait donc "contient".

La premiere fonction **g** prend en argument deux tableaux **v1**, **v2** de double et un entier **i** et retourne un tableau qui contient tous les elements du tableau **v1** qui sont pas dans **v2** et dont l'indice est inferieur a **i**.

La deuxieme fonction **g** prend deux tableaux **v1**, **v2** et retourne tous les elements qui sont dans **v1** mais pas dans **v2**.

## 4.2

La fonction  $g$  est recursive et fait toujours un appel a soi meme. La fonction **f** a une complexite constante , pour **v2** fixe. On a donc une formule recursive qui definit la complexite de l'algorithme  $C(n)$  :

$$C(n) = 6C(n - 1)$$

de plus on voit que  $C(0) = 1$  et donc la complexite est en  $\Theta(n)$ .

La complexite de **g** en fonction de la taille de **v2** ( qu'on notera  $m$ ) est en  $\Theta(m)$ .

En effet, notons la taille de **v1**  $n$ , alors  $n$  est fixe, on doit donc parcourir le tableau  $m$  fois dans l'appel de la fonction  $g$ .

## 1.4

On voit que la periode d'echantillonnage etait de  $\frac{1}{30}$ , donc la frequence d'echantillonnage  $f_e$  etait de 30.

On sait de plus que

$$X_I(m\frac{1}{30}) = a_m = 7\sin(\frac{2\pi}{3}m + \frac{\pi}{6}) + 3\sin(\frac{4\pi}{5}m + \frac{\pi}{4}) + 2\sin(\frac{2\pi}{6}m)$$

On sait que  $f$  peut etre reconstruite car la frequence d'interpolation est assez elevee ( clairement 2 fois plus grande que la bande passante.), on a donc,

$$f(t) = 7\sin(\frac{2\pi}{3}30t + \frac{\pi}{6}) + 3\sin(\frac{4\pi}{5}30t + \frac{\pi}{4}) + 2\sin(\frac{2\pi}{6}30t)$$

en simplifiant, on trouve

$$f(t) = 7\sin(\frac{20\pi}{3}t + \frac{\pi}{6}) + 3\sin(24\pi t + \frac{\pi}{4}) + 2\sin(\pi t)$$

## 6.1

Le but est clairement d'encoder un fichier texte, en remplaçant chaque lettre par un code binaire.

## 6.2

L'erreur est ( en tout cas selon google) que l'operateur « n'arrive pas a lire le type string.

## 6.3

Voici la liste des erreurs que j'ai pu trouver :

- On modifie la valeur de *c* dans la boucle a l'interieur d'une boucle for, ce qui rend le code plus difficilement lisible.
- Les fonctions n'ont pas ete prototypee
- Si la valeur de *c* ( qui est un char mais qui est represente comme entier) est 16, dans la boucle while, on accederait au 16 element du tableau, qui n'existe pas.
- Si le texte est tres grand, la taille de la variable "resultat" dans la fonction encodage pourrait devenir trop grande et produire des problemes d'execution.