

CS221 - TP1

Vincent MOUCADEAU - Rémi MAZZONE — 2A

23/11/2022

Table des matières

1	Introduction	2
2	Préparation	2
2.1	Pseudo code du tri "Bubble sort"	2
2.2	Makefile	2
3	Tri bulle	3
4	Conclusion	5

1 Introduction

Dans ce TP, maintenant que nous avons pris en main les outils de bases, nous allons passer à l'étude d'un système un peu plus complexe, qui traduit un vrai problème. En effet, nous allons étudier le mouvement d'un pendule (sans frottements) avec deux approches différentes : nous utiliserons Simulink dans la première partie et uniquement Matlab dans la deuxième.

2 Préparation

2.1 Pseudo code du tri "Bubble sort"

Listing 1: Pseudo code du tri "Bubble sort" optimisé

```
1  input: int *tab, int n
2  output: int nb_swaps
3  nb_swaps = 0
4  pour i = 0 a n-1 faire
5      bool swapped = false
6      pour j = 0 a n-i-1 faire
7          si tab[j] > tab[j+1] alors
8              echanger tab[j] et tab[j+1]
9              nb_swaps++
10             swapped = true
11         fin si
12     fin pour
13     si swapped == false alors
14         retourner nb_swaps
15     fin si
16 fin pour
```

2.2 Makefile

Listing 2: Makefile du projet

```
1  main:main.o fonctions.o
2      gcc -o $@ $^
3
4  main.o: main.c fonctions.h
5      gcc -c $<
6
7  fonctions.o: fonctions.c fonctions.h
8      gcc -c $<
9
10 clean:
11     rm -rf *.o
12     rm -rf main
```

3 Tri bulle

1. Avec le makefile écrit précédemment, le programme se compile et s'exécute correctement. Le programme affiche bien le contenu du tableau `tab1`.
2. Implémentation de la fonction `swap` en C. On utilise une variable temporaire pour échanger les valeurs des deux variables.

Listing 3: Implémentation de la fonction swap

```
1 void swap(int *a, int *b) {  
2     int temp = *a;  
3     *a = *b;  
4     *b = temp;  
5 }
```

3. Implémentation du tri bulle en C. On utilise la fonction `swap` pour échanger les valeurs des deux variables. On utilise une variable `swapped` pour savoir si un échange a eu lieu. Si aucun échange n'a eu lieu, on peut arrêter le tri.

Listing 4: Implémentation optimisée du tri bulle

```
1 int bubbleSort(int *tab, int n) {  
2     int nb_swaps = 0;  
3     int i, j;  
4     bool swapped;  
5     for (i = 0; i < n-1; i++) {  
6         trie = true;  
7         for (j = 0; j < n-i-1; j++) {  
8             if (tab[j] > tab[j+1]) {  
9                 swap(&tab[j], &tab[j+1]);  
10                nb_swaps++;  
11                trie = false;  
12            }  
13        }  
14        if (trie) {  
15            break;  
16        }  
17    }  
18    return nb_swaps;  
19 }
```

4. On vérifie que le tri fonctionne correctement à l'aide de la fonction `compare` qui compare élément par élément le tableau trié avec le tableau de référence. On affiche le nombre d'échanges effectués.
5. Résultats du tri bulle pour les tableaux donnés :
 - `tab1` : Comparaison OK, 682 échanges

- `tab2` : Comparaison OK, 1216 échanges
- `tab3` : Comparaison OK, 63 échanges
- `ref` : Comparaison OK, 0 échanges

4 Conclusion

Lors de ce TP, nous avons pu simuler le comportement d'un circuit $RC + R//C$ à l'aide de Matlab. Nous avons pu faire une analyse temporelle et fréquentielle du circuit. Nous avons pu observer que la tension U_c est déphasée par rapport à la tension U_s .