

melody.txt

f .wav f. → output.wav - f

extrait

1 nom de fichier wav de melody.txt

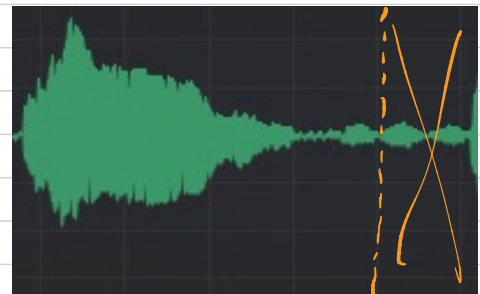
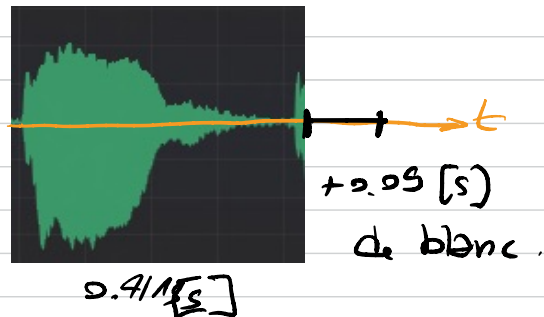
↳ numSamples (du .wav)

realloc du tableau final avec + numSample

transfert de .wav → tableau final.

durée de chaque wav ajoutée et fixe: 0.5[s].

- ① si numSamples < 0.5[s] → rallonger avec des 0  
 si numSamples > 0.5[s] → tronquer le signal.



→ lecture data .wav à stocker dans un tableau intermédiaire

→ realloc du tableau final: + 0.5[s] (fixe)

Si numSamples < 0.5[s] copie dans le tableau final  
 Si numSamples > 0.5[s] copie "tronquée"

TD20210506 (copie de 0523)

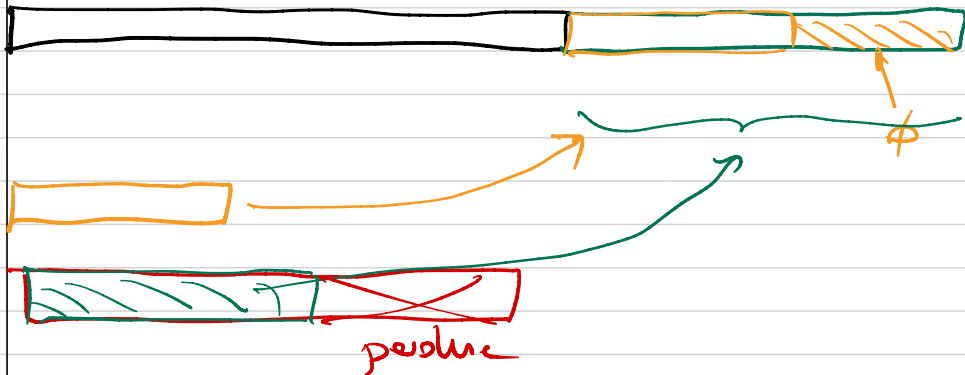
Pour la lecture du .wav → tableau dynamique temporaire.

realize "to S[S]"

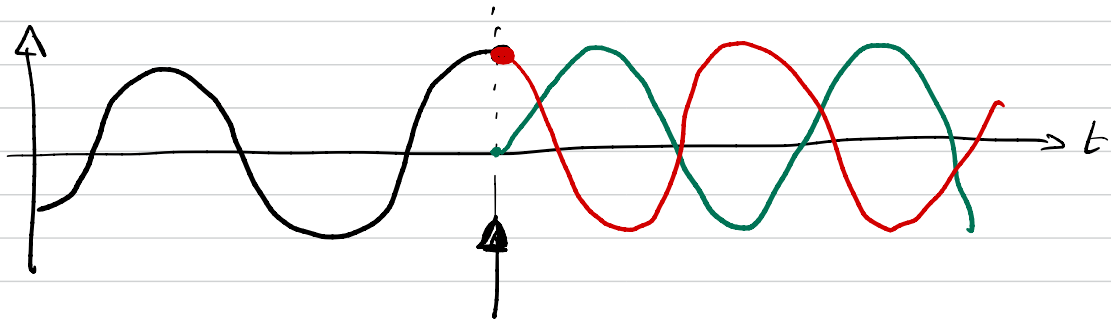
tableau final.

WAV < 0.5s

WAV > 0.5s

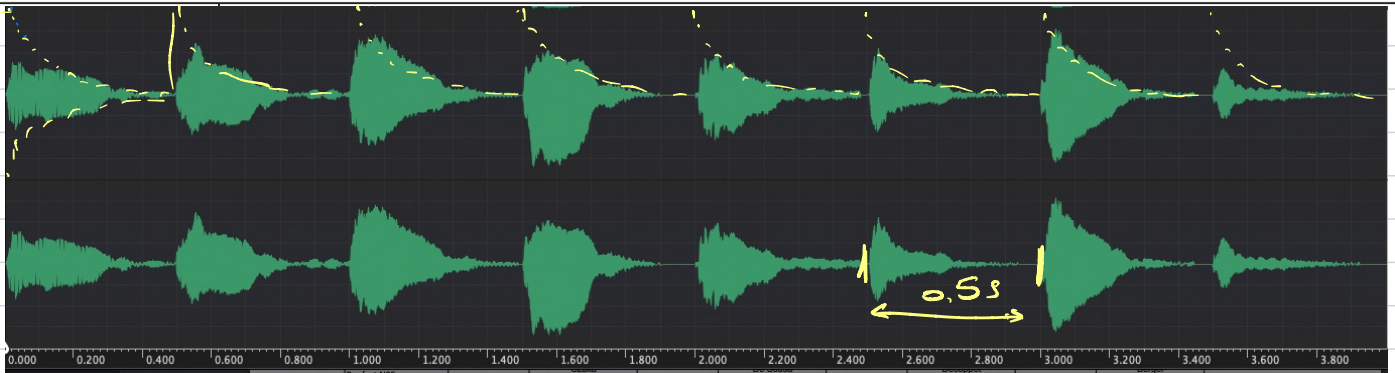


16<sup>3</sup>10.

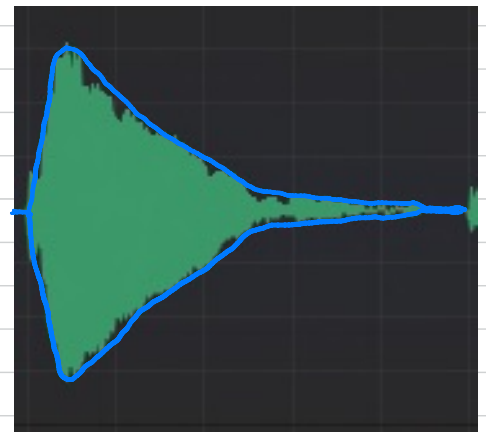
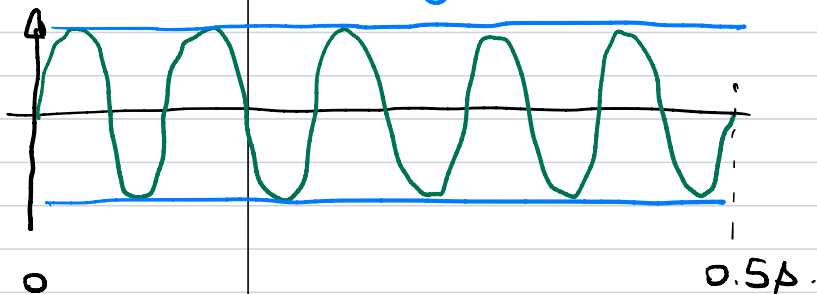


$$x(t) = A \sin(\omega t) \quad \text{at final.}$$

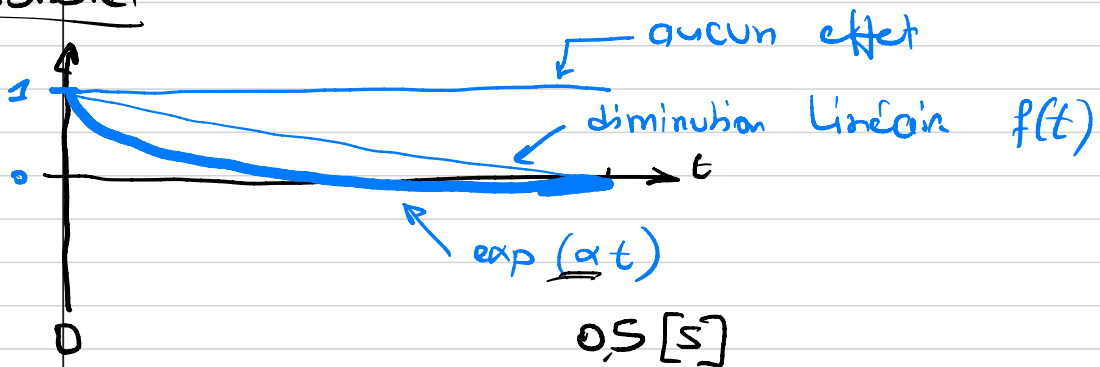
$$x(t) = A \sin(\omega t + \phi)$$



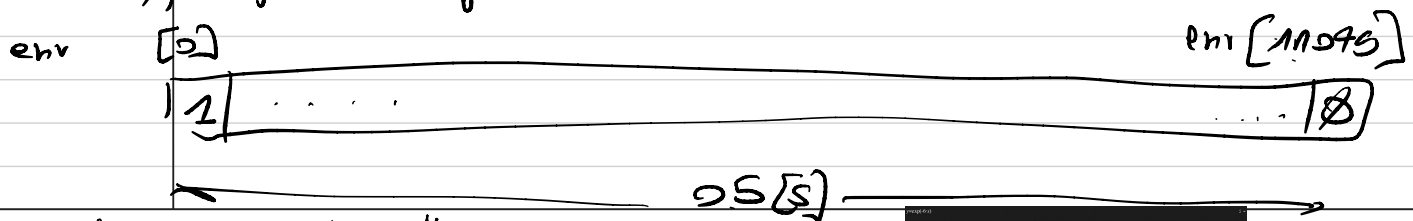
Enveloppe du signal.



Effet sonore:

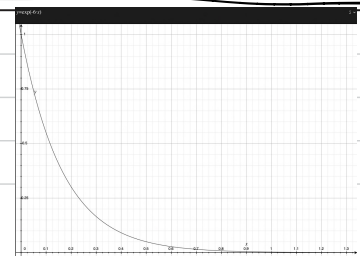


Un fois que le tableau final est plein on va appliquer l'enveloppe sur le signal.



① Créer le tableau "env"

② le remplir avec  $f(t)$



```
char *t = (char *) malloc(10 * sizeof(char));  
char *s = t;
```

```
char *newt = (char *) realloc(t, 20 * sizeof(char));
```

```
if (newt != NULL) {
```

```
    t = newt;
```

```
    // s = t  
    // t[0] .. t[19]  
    t++...
```

```
}
```

```
else {
```

```
    // t[0] .. t[9]
```

```
}
```

```
free(s);
```