

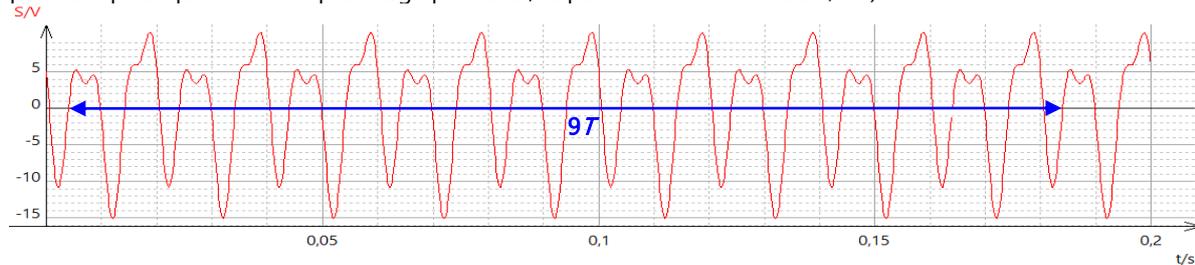
Décomposition et recomposition d'une onde périodique

Éléments de correction

I. Décomposition d'une onde périodique.

2. Pour mesurer la période avec précision :

- On choisit un point où la courbe a une pente assez raide et où elle coupe une ligne horizontale (tel que l'axe du 0 V) ;
- On zoome sur une période pour que ce soit le plus large possible (ou on mesure la durée d'un nombre N de périodes pour que ce soit le plus large possible ; la période est alors durée / N).



$$9T = 0,180 \text{ s} \quad \text{donc} \quad T = \frac{0,180 \text{ s}}{9} = 0,0200 \text{ s}$$

$$f = \frac{1}{T} = \frac{1}{0,0200 \text{ s}} = 50,0 \text{ Hz}$$

3. - Fondamental (harmonique de rang 1) : amplitude = 3 V et fréquence = 50 Hz ;

- harmonique de rang 2 : amplitude = 9 V et fréquence = 100 Hz ;
- harmonique de rang 3 : amplitude = 0,3 V et fréquence = 150 Hz ;
- harmonique de rang 4 : amplitude = 4 V et fréquence = 200 Hz ;

4. Ceci veut dire que le signal se porte comme la somme de 4 sinusoïdes dont les caractéristiques sont celles de la question précédente.

5. fréquence du fondamental = fréquence du signal (ici 50 Hz)

les fréquences des autres harmoniques sont des multiples de la fréquence du fondamental (ici, $2 \times 50 \text{ Hz}$, $3 \times 50 \text{ Hz}$ et $4 \times 50 \text{ Hz}$).

II. Recomposition de l'onde périodique à partir de sinusoïdes.

6. Pour créer la liste des dates t :

- Choisir *Fichier/Nouveau/Simulation* ;
- La durée permettant de visualiser 5 oscillations est celle de 5 période soit

$$5T = 5 \times \frac{1}{f} = 5 \times \frac{1}{50 \text{ Hz}} = 0,1 \text{ s} \quad (\text{à saisir dans } \textit{Maxi}) ;$$

- Choisir un nombre de points d'environ 10 000.

Pour créer les autres grandeurs :

- Cliquer sur l'icône *ajouter* (et préciser *grandeur calculée*) et saisir, pour la grandeur S_1 l'expression $3 \times \cos(360 \times 50 \times t + 0)$;

- Cliquer sur l'icône *ajouter* (et préciser *grandeur calculée*) et saisir, pour la grandeur S_2 l'expression $9 \times \cos(360 \times 100 \times t + 100)$;

- Cliquer sur l'icône *ajouter* (et préciser *grandeur calculée*) et saisir, pour la grandeur S_3 l'expression $0,3 \times \cos(360 \times 150 \times t + 20)$;

- Cliquer sur l'icône *ajouter* (et préciser *grandeur calculée*) et saisir, pour la grandeur S_4 l'expression $4 \times \cos(360 \times 200 \times t + 35)$.

Puis visualiser ces 4 courbes.

Puis créer la grandeur S : cliquer sur l'icône *ajouter*  (et préciser *grandeur calculée*) et saisir, pour la grandeur S l'expression $S_1 + S_2 + S_3 + S_4$.

Puis visualiser la courbe représentant S en fonction du temps. On vérifie qu'il s'agit bien du même signal que dans la partie I.

7.

```
import matplotlib.pyplot as plt
from math import*

# creation d'une liste contenant les 10 000 valeurs du temps
list_t = []
for n in range(10000):
    list_t.append(n*0.1/10000)

# creation d'une liste contenant les valeurs de S1 pour chacune des 10 000 valeurs du temps
list_S1 = []
for n in range(10000):
    list_S1.append(3 * cos(2*3.1416 * 50 * list_t[n] + 0))

# affichage du graphique
plt.clf()

plt.plot(list_t, list_S1, "b-", label="S1")

plt.legend()
plt.xlabel("t (s)")
plt.ylabel("U (V)")
plt.grid(False)
plt.show()
```

Le programme ci-dessus affiche 5 oscillations de l'onde S_1 .

Le programme suivant affiche les ondes S_1 à S_4 :

```
import matplotlib.pyplot as plt
from math import*

# creation d'une liste contenant les 10 000 valeurs du temps
list_t = []
for n in range(10000):
    list_t.append(n*0.1/10000)

# creation d'une liste contenant les valeurs de S1 pour chacune des 10 000 valeurs du temps
list_S1 = []
for n in range(10000):
    list_S1.append(3 * cos(2*3.1416 * 50 * list_t[n] + 0))

# creation d'une liste contenant les valeurs de S2 pour chacune des 10 000 valeurs du temps
list_S2 = []
for n in range(10000):
    list_S2.append(9 * cos(2*3.1416 * 100 * list_t[n] + 1.75))

# creation d'une liste contenant les valeurs de S3 pour chacune des 10 000 valeurs du temps
list_S3 = []
for n in range(10000):
    list_S3.append(0.3 * cos(2*3.1416 * 150 * list_t[n] + 0.35))

# creation d'une liste contenant les valeurs de S4 pour chacune des 10 000 valeurs du temps
list_S4 = []
for n in range(10000):
    list_S4.append(4 * cos(2*3.1416 * 200 * list_t[n] + 0.61))

# affichage du graphique
plt.clf()

plt.plot(list_t, list_S1, "b-", label="S1")
plt.plot(list_t, list_S2, "g-", label="S2")
plt.plot(list_t, list_S3, "c-", label="S3")
plt.plot(list_t, list_S4, "y-", label="S4")
```

```
plt.legend()
plt.xlabel("t (s)")
plt.ylabel("U (V)")
plt.grid(False)
plt.show()
```

Le programme suivant affiche les ondes S_1 à S_4 et l'onde S :

```
import matplotlib.pyplot as plt
from math import*

# creation d'une liste contenant les 10 000 valeurs du temps
list_t = []
for n in range(10000):
    list_t.append(n*0.1/10000)

# creation d'une liste contenant les valeurs de S1 pour chacune des 10 000 valeurs du temps
list_S1 = []
for n in range(10000):
    list_S1.append(3 * cos(2*3.1416 * 50 * list_t[n] + 0))

# creation d'une liste contenant les valeurs de S2 pour chacune des 10 000 valeurs du temps
list_S2 = []
for n in range(10000):
    list_S2.append(9 * cos(2*3.1416 * 100 * list_t[n] + 1.75))

# creation d'une liste contenant les valeurs de S3 pour chacune des 10 000 valeurs du temps
list_S3 = []
for n in range(10000):
    list_S3.append(0.3 * cos(2*3.1416 * 150 * list_t[n] + 0.35))

# creation d'une liste contenant les valeurs de S4 pour chacune des 10 000 valeurs du temps
list_S4 = []
for n in range(10000):
    list_S4.append(4 * cos(2*3.1416 * 200 * list_t[n] + 0.61))

# creation d'une liste contenant les 10 000 valeurs de S, somme des 10 000 valeurs de S1 + S2
# + S3 + S4
list_S = []
for n in range(10000):
    list_S.append(list_S1[n] + list_S2[n] + list_S3[n] + list_S4[n])

# affichage du graphique
plt.clf()

plt.plot(list_t, list_S1, "b-", label="S1")
plt.plot(list_t, list_S2, "g-", label="S2")
plt.plot(list_t, list_S3, "c-", label="S3")
plt.plot(list_t, list_S4, "y-", label="S4")
plt.plot(list_t, list_S, "r-", label="S")

plt.legend()
plt.xlabel("t (s)")
plt.ylabel("U (V)")
plt.grid(False)
plt.show()
```