

Séance 17:

Corde vibrante

Le but est de visualiser la corde vibrante $y = f(x, t) = \frac{\omega(x-ct) + \omega(x+ct)}{2}$ où :

$$\begin{cases} \text{cest la vitesse de propagation de la déformation} \\ y = \omega(x) \text{ est la forme initiale } (t = 0) \text{ de la corde} \end{cases}$$

1. Init

Ecrire une fonction `init(x)` qui donne la valeur de $\omega(x)$ pour $x \in [-10, 10]$. Pour dessiner une grille dans le graphe :

```
1 plt.xticks(np.arange(-10, 11, 1))
2 plt.grid()
```

> main.py

```
1 def init(x):
2     if (x < 0):
3         return -(init(-x))
4     elif (x <= 7):
5         return x/7
6     return -(x-10)/3
```

2. Omega

(a) *Ecrire une fonction `omega(x)` qui prolonge par période de 20 la fonction `init` à \mathbb{R} .*

> main.py

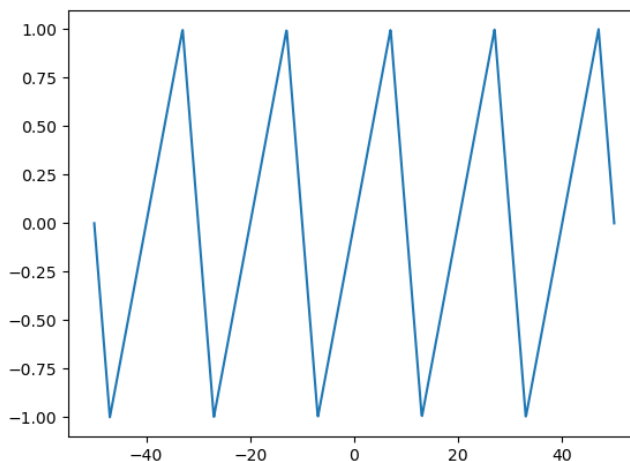
```
1 def omega(x):
2     k = np.floor((x+10)/20)
3     return init(x-k*20)
```

(b) *Tester en traçant la courbe sur cinq périodes.*

> main.py

```
1 def Trace(T):
2     x = np.linspace(-10*T, 10*T, 1200)
3     y = [omega(k) for k in x]
4     plt.plot(x, y)
5     plt.show()
```

En exécutant `Trace(5)`, j'obtiens :



3. Chaîne

Tracer la courbe $y = f(x, t)$ pour une vitesse $c = 2$ et des temps espacés de $t = 0.2 * k$ avec $t \leq 5$.

> main.py

```
1 def vibrante(k):
2     x = np.linspace(0, 10, 1000)
3     z = [omega(val) for val in x]
4     zz = [omega(val-10) for val in x]
5     c = 2
6     t = np.floor(k*100)/100
7     plt.figure()
8     plt.axis([0, 10, -1, 1])
9     y = [(omega(val+c*t)+omega(val-c*t))/2 for val in x]
10    plt.plot(x, y, "-", color=(0, 0, 0), linewidth=2.5, label="t="+str(t))
11    plt.plot(x, zz, ":", color=(1, 0, 0), linewidth=2, label="t=5")
12    plt.plot(x, z, ":", color=(1, 0, 0), linewidth=2, label="t=0")
13    plt.legend()
14    plt.show()
```

En exécutant :

```
1 for k in range(1, 5):
2     vibrante(k)
```

J'obtiens :

