Activité numérique

Chapitre 15, p. 331

2 Propagation d'une onde périodique

Script à compléter

Fichiers Python
Scripts à compléter
Fiches d'accompagnement
hatier-clic.fr/pc1331

L'objectif de cette activité est l'étude de la propagation d'une onde périodique à la surface d'une cuve à ondes à l'aide d'une simulation.
Cela contribue à la découverte des notions de longueur d'onde et de points en phase.

Prérequis théoriques

- **Notions de période et de signal périodique**

Deux programmes sont utilisés :

- l'un (15_onde-temporel.py) affichant deux graphiques représentant l'évolution temporelle de l'altitude de deux points au cours du temps,
- l'autre (15_onde-spatial.py) affichant deux graphiques représentant l'aspect de la surface de l'eau à deux dates.

Les seules modifications des programmes sont celles des données :

- période, amplitude, célérité des ondes pour les deux programmes ;
- distance du deuxième point à la source pour le programme 15_onde-temporel ;
- date de la vue du deuxième graphique pour le programme 15_onde-spatial.

Activité numérique

Chapitre 15, p. 331

Script à compléter (Programme « 15_onde-temporel.py »)

```
from pylab import *

print("")
print("************************************")
print("* Propagation d'une onde périodique *")
print("*  Mouvements en fonction du temps  *")
print("************************************")
print("")

##########################
### Données modifiables ###
##########################
A=5      # amplitude en mm (entre 0 et Amax)
T=50     # période en ms (entre 10 et 100)
v=40     # célérité en cm/s (entre 10 et 100)
xM=7     # abscisse de M en cm (entre 0 et L)

### Valeurs initiales
#A=5
#T=50
#v=40
#xM=7

### Données non modifiables
L=30             # longueur totale de la cuve en cm
tmax=250         # date de fin en ms
Amax=10          # amplitude maximale en mm

### Test pour les valeurs
probleme=[]
if T<10 or T>100:
        probleme.append("T doit être comprise entre 10 ms et 100 ms")
if v<10 or v>100:
        probleme.append("v doit être comprise entre 10 m/s et 100 m/s")
if xM<=0 or xM>L:
        probleme.append("xM doit être comprise entre 0 m et",L,"cm")
if A<=0 or A>Amax:
        probleme.append("A doit être comprise entre 0 mm et",Amax,"mm")
if len(probleme)!=0:
        for i in range(len(probleme)):
                print(probleme[i])
        sys.exit("Modifiez les paramètres et réessayez")

### Affichage des paramètres
print("Amplitude de l'onde : A =",round(A,1),"mm")
print("Période de l'onde : T =",round(T,1),"ms")
print("  d'où la fréquence f =",round(1000/T,1),"Hz")
print("Célérité de l'onde : v =",round(v,1),"cm/s")
print("  d'où la longueur d'onde de l'onde : lambda =",round(v*T/1000),"cm")
print("")

### Définition de t (500 points)
t=linspace(0,tmax,500)
### Définition de la fonction périodique
def y(t):
        return 0.56*A*(sin(2*pi*t/T)-sin(4*pi*t/T))
### Définition des graphiques
fig=figure()
gcf().subplots_adjust(hspace=0.7)
```

Modules

Le module `pylab` permet à la fois les fonctionnalités mathématiques et les tracés de courbes de manière souple.

Données modifiables

Attention aux unités, qui ne sont pas les unités de base du système international mais sont adaptées à la géométrie réaliste d'une cuve à ondes.

Données et tests (ne pas modifier)

Les dimensions sont réduites pour que les calculs restent dans des valeurs raisonnables. Les tests réalisés permettent de s'assurer que les valeurs ne sortent pas de ces limites.

Affichage des paramètres

Les grandeurs sont affichées avec leurs unités, arrondies à la première décimale. La longueur d'onde et la fréquence sont calculées.

```
60  ### Graphique à la source
61  fig.add_subplot(2,1,1)
62  plot(t,y(t))
63  title("y(t) à la source")
64  xlabel("t (en ms)")
65  ylabel("y (en mm)")
66  grid(True)
67  ylim(-Amax,Amax)
68  xlim(0,tmax)
69  ### Graphique en xM
70  fig.add_subplot(2,1,2)
71  plot(t,y(t-1000*xM/v))
72  title("y(t) à xM = "+str(xM)+" cm de la source")
73  xlabel("t (en ms)")
74  ylabel("y (en mm)")
75  grid(True)
76  ylim(-Amax,Amax)
77  xlim(0,tmax)
78  ### Affichage
79  show()
```

Fonction $y(t)$ à la source

Il s'agit d'une fonction périodique de période T, non symétrique, d'amplitude entre la valeur moyenne et la crête réglée pour valoir A.

Graphiques

Un graphique de $y(t)$ à la source, un autre au point M à la distance x_M fixée.

Script à compléter (Programme « 15_onde-spatial.py »)

```
1   from pylab import *
2
3   print("")
4   print("*************************************")
5   print("* Propagation d'une onde périodique *")
6   print("*     Aspect spatial du milieu      *")
7   print("*************************************")
8   print("")
9
10  ############################
11  ### Données modifiables ###
12  ############################
13  A=5      # amplitude en mm (entre 0 et Amax)
14  T=50     # période en ms (entre 10 et 100)
15  v=40     # célérité en cm/s (entre 10 et 100)
16  t1=0.12 # date t1 en s (positive)
17
18  ### Valeurs initiales
19  #A=5
20  #T=50
21  #v=40
22  #xM=7
23
24  ### Données non modifiables
25  L=30             # longueur totale de la cuve en cm
26  tmax=250         # date de fin en ms
27  Amax=10          # amplitude maximale en mm
28
29  ### Test pour les valeurs
30  probleme=[]
31  if T<10 or T>100:
32          probleme.append("T doit être comprise entre 10 ms et 100 ms")
33  if v<10 or v>100:
34          probleme.append("v doit être comprise entre 10 m/s et 100 m/s")
```

```
35  if t1<=0:
36          probleme.append("t1 doit être positive")
37  if A<=0 or A>Amax:
38          probleme.append("A doit être comprise entre 0 mm et",Amax,"mm")
39  if len(probleme)!=0:
40          for i in range(len(probleme)):
41                  print(probleme[i])
42          sys.exit("Modifiez les paramètres et réessayez")
43
44  ### Affichage des paramètres
45  print("Amplitude de l'onde : A =",round(A,1),"mm")
46  print("Période de l'onde : T =",round(T,1),"ms")
47  print("  d'où la fréquence f =",round(1000/T,1),"Hz")
48  print("Célérité de l'onde : v =",round(v,1),"cm/s")
49  print("  d'où la longueur d'onde de l'onde lambda =",round(v*T/1000),"cm")
50  print("")
51
52  ### Définition de x (500 points)
53  x=linspace(0,L,500)
54  ### Définition de la fonction périodique
55  def y(t):
56          return 0.56*A*(sin(2*pi*t/T)-sin(4*pi*t/T))
57  ### Définition des graphiques
58  fig=figure()
59  gcf().subplots_adjust(hspace=0.7)
60  ### Graphique à la source
61  fig.add_subplot(2,1,1)
62  plot(x,y(-1000*x/v))
63  title("y(x) à t0 = 0 s")
64  xlabel("x (cm)")
65  ylabel("y (mm)")
66  ylim(-Amax,Amax)
67  xlim(0,L)
68  grid(True)
69  ### Graphique en xM
70  fig.add_subplot(2,1,2)
71  plot(x,y(1000*t1-1000*x/v))
72  title("y(x) à t1 = "+str(t1)+" s")
73  xlabel("x (cm)")
74  ylabel("y (mm)")
75  ylim(-Amax,Amax)
76  xlim(0,L)
77  grid(True)
78  ### Affichage
79  show()
```

Graphiques

Un graphique de $y(x)$ à la date $t = 0$, un autre à la date t_1 définie dans les données.